TOPICS ON SPATIALLY HIGH-ORDER ACCURATE METHODS AND PRECONDITIONING FOR THE NAVIER-STOKES EQUATIONS WITH FINITE-RATE CHEMISTRY

by

Andrew Grady Godfrey

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Aerospace Engineering

APPROVED:

R. W. Walters
Robert W. Walters, Chairman

B Grossman
Bernard Grossman

Rakesh K. Kapania
Rakesh K. Kapania

William H. Mason
William H. Mason

Joseph A Schetz
Joseph A. Schetz

Bram van Leer

December, 1992
Blacksburg, Virginia

TOPICS ON SPATIALLY HIGH-ORDER ACCURATE METHODS AND PRECONDITIONING FOR THE NAVIER-STOKES EQUATIONS WITH FINITE-RATE CHEMISTRY

by

Andrew Grady Godfrey

Committee Chairman: Robert W. Walters

Aerospace Engineering

(ABSTRACT)

This dissertation discusses two aspects of computational fluid dynamics: high-order spatial accuracy and convergence-rate acceleration through system preconditioning. Concerning high-order accuracy, the computational qualities of various spatial methods for the finite-volume solution of the Euler equations are presented. The two-dimensional essentially non-oscillatory (ENO), k-exact, and dimensionally split ENO reconstruction operators are discussed and compared in terms of reconstruction and solution accuracy and computational cost. Standard variable extrapolation methods are included for completeness. Inherent steady-state convergence difficulties are demonstrated for adaptive-stencil algorithms. Methods for reconstruction error analysis are presented and an exact solution to the heat equation is used as an example. Numerical experiments presented include the Ringleb flow for numerical accuracy and a shock-reflection problem. A vortex-shock interaction demonstrates the ability of the ENO scheme to excel in capturing unsteady high-frequency flow physics.

Concerning convergence-rate acceleration, characteristic-wave preconditioning is extended to include generalized finite-rate chemistry with non-equilibrium thermodynamics. Additionally, the proper preconditioning for the one-dimensional

Navier-Stokes equations is presented. Eigenvalue stiffness is resolved and convergence-rate acceleration is demonstrated over the entire Mach-number range from the incompressible to the hypersonic. Specific benefits are realized at low and transonic flow speeds. The extended preconditioning matrix accounts for thermal and chemical non-equilibrium and its implementation is explained for both explicit and implicit time marching. The effects of high-order spatial accuracy and various flux splittings are investigated. Numerical analysis reveals the possible theoretical improvements from using preconditioning at all Mach numbers. Numerical results confirm the expectations from the analysis. The preconditioning matrix is applied with dual time stepping to obtain arbitrarily high-order accurate temporal solutions within an implicit formulation. Representative test cases include flows with previously troublesome embedded high-condition-number regions.

ACKNOWLEDGEMENTS

Before you, I sit in my humble apartment reflecting upon those who were crucial to the completion of my doctoral degree. Outside my window, snow sifts down from a frozen layer of stratus above, blanketing a sleepy, yet familiar, Virginia town which has been my home for the previous nine years. An aroma of Christmas juniper mists my senses as a winter white conceals Blacksburg's familiarity.

First, I must thank God for blessing me with a seemingly endless number of gifts. Patience, endurance and perseverance are His qualities, and not always mine. I am fortunate to have a very warm and loving wife, Gwen, who sacrificed her time and uplifted my spirits during the most intense academic challenge of my life. For his enduring patience and unnerving management, I graciously thank my advisor, Professor Bob Walters. His positive influence will endure within me throughout my aerospace career. For their friendship and support, I thank my co-workers in the sun room. My fellow graduate students were always interested and never lacking of opinion. Without a thanks for their support and sharing, this thesis would not be complete. A particular thanks is due Curtis Mitchell for many interactive discussions on non-linear reconstruction algorithms and for enduring the agony of second-shift work sessions. Finally, I must thank Eric Fuller who always seemed to have the right words to say.

The insight and inspiration of Bram van Leer proved pivotal in my education. With his friendly concern, my graduate career became a reflection of pride. For most of my aerodynamics education, I am indebted to Professor Bernard Grossman. Any mistakes I make are in no way a reflection upon him. This thesis would not have been possible without the financial support of Ajay Kumar under NASA grant NAG-1-776 and the two generous fellowships awarded by the Virginia Space Grant Consortium in Hampton, Virginia.

Finally I wish to thank my parents, Lee and Emily Godfrey. Their example of love, support, and guidance will forever be my continuing goal.

TABLE OF CONTENTS

PART ONE

SPATIALLY HIGH-ORDER ACCURATE METHODS

PART TWO

PRECONDITIONING WITH FINITE-RATE CHEMISTRY

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Part I

$c_{i,j,l}$	k-exact polynomial coefficients
$\epsilon_{\Delta x_i}$	solution error for grid spacing Δx_i
h_j	width of cell j
h^n	truncation error
i_m	least cell index in the support stencil
k	degree of reconstruction polynomial
n	order of spatial accuracy
p	thermodynamic pressure
$\bar{q}$	cell average fluid dynamic variable
r_c	measured computational accuracy
r_t	order of time accuracy
$\bar{w}$	general cell average value
$w(x)$	pointwise function w
$\tilde{w}$	pointwise reconstruction of q
x, y	Cartesian coordinates
$G_{i,j,l}$	integral of basis functions
$H_{k+1}(x)$	$k+1$ degree interpolation polynomial for discrete primitive function
$J(x, y)$	Jacobian of transformation
L_p	integral p-norm
L_p^s	semi-norm
N	number of Gauss points
Q^2	square of velocity magnitude

$R(x;\bar{w})$ –	reconstruction polynomial
S_m –	stencil set for cell m
$\bar{T}$ –	cell average temperature
$T(x,y)$ –	temperature field
$TV[\cdot]$ –	total variation
$W(x)$ –	continuous primitive function
ρ –	density
ϕ, κ –	MUSCL differencing high-order accuracy parameters
ξ, η –	Computational coordinates
Ω_m –	domain for cell m

Part II

a –	Sound speed
A, B –	Inviscid flux Jacobians
A_v –	Viscous flux Jacobian
c_v –	Specific heat at constant volume per unit mass
e –	Mixture internal energy per unit mass
e_0 –	Mixture internal stagnation energy per unit mass
e_n –	Mixture non-equilibrium energy per unit mass
f, g, h –	Inviscid flux vectors
h_0 –	Mixture internal stagnation enthalpy per unit mass
$\hat{f}$ –	Generalized coordinate flux vector
G –	Gain matrix
$\vec{k}_q, \vec{l}_q, \vec{m}_q$ –	Unit vectors
L –	Characteristic length
m –	Wave number
M –	Mach number
n, N –	Iteration time levels

p	Thermodynamic pressure
P	Preconditioning matrix for
	symmetrization variables
$\mathcal{P}$	Preconditioning matrix for
	numerical implementation
$\mathbf{p}$	3 × 3 velocity submatrix in $\mathcal{P}$
Pr	Prandtl number
q	Primitive variables, velocity magnitude
Q	Conservative variables
R	Flux balance residual
Re	Reynolds number
t	Time
T	Temperature
u, v, w	Cartesian velocity components
$\bar{u}$	Velocity normal to cell face
U	Symmetrization variables
	aligned coordinate system
Vol	Cell volume
x, y, z	Cartesian coordinates
Z	Viscous symmetrization variables
α	Angle of attack
β_i	Defined thermodynamic variable
β_x	Fourier frequency
γ	Ratio of specific heats
ϵ^n	Error in solution at time level n
θ	Cell face normal angle
$\theta_{\bar{x}}, \theta_{\bar{y}}, \theta_{\bar{z}}$	Direction cosine angles
λ	Courant number, wave length, and eigenvalue
ν	Courant number, kinematic viscosity

ξ, η, ζ –	Generalized and stream-aligned coordinates
$\mathbf{\Pi}$ –	3×3 velocity submatrix in $\mathcal{P}^{-1}$
ρ –	Mixture density
ρ_i –	Species density
ρ_s –	Spectral radius of the gain
τ, β –	Preconditioning factors dependent
–	upon Mach number

Subscripts

$(\cdot)_i$ –	Species value
L –	Left state variable
R –	Right state variable
x –	Component in the x direction
y –	Component in the y direction
z –	Component in the z direction
0 –	Stagnation quantity

Accents

$(\hat{\cdot})$ –	Unit vector or Generalized coordinate flux vector
$(\tilde{\cdot})$ –	Roe averaged quantity
$(\bar{\cdot})$ –	Stream aligned quantity

Operators

$\Delta(\cdot)$ –	Forward difference operator
$[\![\cdot]\!]$ –	$(\cdot)_R - (\cdot)_L$

"It was another baffling case, but then,
you don't hire a private eye for the easy ones."

— Bill Watterson's *Calvin and Hobbes*

PREFACE

A curious trait of human nature is the almost endless pursuit of orderly explanations. Perhaps no endeavor is more indicative than the fields of classical science and engineering. Pure mathematicians strive to constrain our untamed nature into a little box – bound, controlled, and predicted. In fact, the challenge of understanding may be reason for any scientific discipline's appeal.

Fundamentally, all scientific endeavor begins with first principles. In the field of fluid dynamics, the ideal fluid continuum is governed by the laws of conservation. Written in their mathematical form, these laws become a system of partial differential equations known as the Navier-Stokes equations. Assuming an inviscid fluid, the laws are termed the Euler equations. Hopefully, nature's gurgles of fluid are modeled adequately by these idealistic equations.

Unfortunately, a computer's humming thicket of wires and silicon operates only upon discrete numbers, and not upon a continuum. For simulation purposes the flowing expanse of fluid motion must be severed into small, finite units. Likewise, the continuum of time must be discretely approximated. After segregating the flow domain in both space and time, the computational fluid dynamicist has constructed his numerical wind-tunnel.

Basis functions in numerical fluid dynamics

Fundamentally, four categories lend themselves to representing a solution on a finite domain. These are the finite-difference, finite-element, spectral, and finite-volume methods. Their differences lie in how they recover an analytical, continuous solution from discrete data. For example, any solution can be approximated as

$$u(x, t^n) = \sum_i c_i \, \phi_i(x).$$

where ϕ_i are known as the basis functions and c_i are their corresponding coefficients. The above basis functions have different forms for each of the four categories.

In the finite-difference method, the basis functions are discrete delta functions and an analytic solution is represented as shown in Fig. (1a). Data is known only at specific points in the domain. Interpolation is a straight-forward procedure on a given grid, but generalization to arbitrary geometries has its own unique intricacies – among them maintaining a free stream. Specifically, the metric terms must be differenced in the identical fashion as the flux function to negate grid-induced errors.

The finite-element method utilizes the hat function with compact support as its basis function and a solution is represented as in Fig. (1b). This formulation attractively allows for the placement of grid points at any desired location in the flow. However, an inherent neighboring cell structure is lost and additional connectivity information is needed. As a result, implicit time marching strategies, which are attractive because of their high time-step capabilities, require the inversion of sparse matrices with large bandwidths. A finite-element researcher can expect both large memory requirements and high computational time per iteration.

The spectral method attempts to reproduce the solution with globally, rather than locally, defined basis functions and a solution is modeled as in Fig. (1c). The trial functions may be either Fourier expansions or Chebyshev polynomials. Grid generation in the sense of finite-difference methods is not necessary because a specific choice of collocation points is crucial to the spectral method's accuracy. Attractively, spectral methods exhibit "super-quadratic" convergence upon grid refinement. However, Gibbs-like oscillations occur around the discontinuities inherent to transonic and supersonic flows.

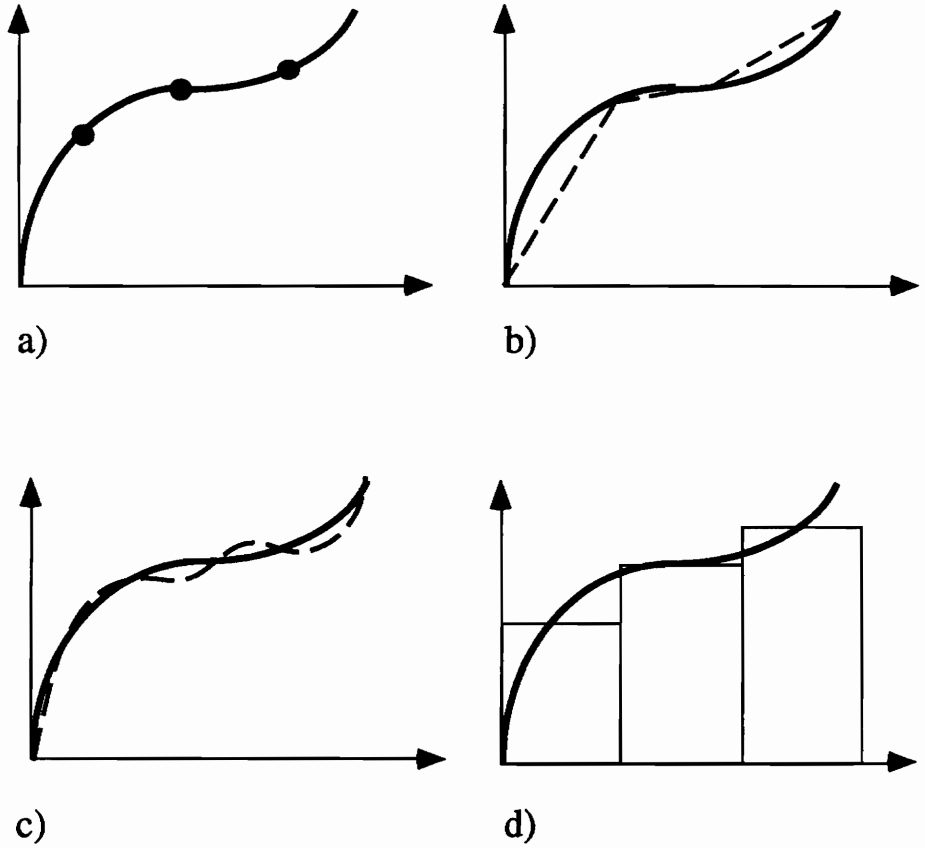

Figure 1. Four categories for representing a function on a finite domain.

In the finite-volume method data is stored as a cell average over the mesh, not at any specific point. The solution is cell averaged as in Fig. (1d). The basis function is then a step with compact support, and the grid lines serve as the boundaries for each connecting cell. The finite-volume method originates from the integral equations of fluid flow and has enjoyed unsurpassed popularity since Godunov's ingenious introduction of the Riemann solution into the numerical algorithm [23].

The finite-element, finite-difference and finite-volume methods approximate the solution locally. To control oscillations in high-gradient regions such as shocks and contact surfaces, these methods must either intelligently choose from the available data or limit the information used to describe the flow field. Either process then presents areas of research for controlling the dissipation to capture crisp discontinuities, but yet not allowing numerical wiggles. Additionally, the control of the damping at large-gradient regions must not be so diffusive as to lose the details of the flow field. Part I of this thesis addresses high-order reconstruction, the finite-volume analog to interpolation.

Accuracy and Monotonicity

Two issues are truly at hand regarding high-order spatial methods: accuracy and monotonicity. One may expect that high-order algorithms give better solutions. Indeed, for smooth flow fields this is true; however, if a flow field has discontinuities or steep gradients, a high-order algorithm faithfully collides with Godunov's proof that a monotonicity-preserving linear scheme is necessarily first-order accurate [23]. Consequently, high-order, constant-coefficient schemes generate spurious oscillations. After a single time step a monotonicity-preserving scheme propagates an initially monotonic solution downstream without generating new extrema. The solution for all successive time steps remains monotone. Unfortunately, high-order numerical schemes are not so obedient.

Oscillation generation is easily understood with the initial profile shown in Fig. (2). Specifically, a second-order method reconstructs the given discontinuous data as a piecewise-linear fluctuation. Subsequently, an undershoot and overshoot are generated upstream and downstream of the discontinuity, respectively. This reconstructed solution then propagates downstream a distance $a\Delta t$, and new cell averages evolve from this new "solution". Consequently, the upstream cell average decreases and the downstream cell average increases. This process effectively generates oscillations at every sufficiently steep gradient in the flow field. Reconstruction using rapidly changing cell-average data is the fertilizer for these growing oscillations.

To achieve both high-order accuracy and monotonicity preservation, we must incorporate non-linearity even if the model equation is linear. Traditional methods simply reconstruct the solution to high-order accuracy over the entire domain and bound the result in high gradient regions with limiters. Commonly used limiters include the minimum-modulus (min-mod) limiter, Van Albada's smooth limiter [63], or Roe's SuperBee limiter [51], among others. With any limiter, uniform accuracy is lost at the expense of monotonicity. Researchers have continually desired the perfect algorithm – high accuracy *and* monotonicity. To achieve this lofty goal, we require that the support stencil ebb with the computational solution. With this background knowledge, the objective of a general non-linear class of schemes called the essentially non-oscillatory (ENO) schemes is the quest for the most monotone data for reconstructing the local solution to high order. Stencil-adaptive, high-order schemes are the topic of Part I.

Part II of this thesis addresses convergence-rate acceleration for steady-state flows by eliminating eigenvalue stiffness. An enormous engineering effort has gone into developing computational codes for simulating modern compressible flows. These codes are limited by only the size and speed of the processors on which they run. Because computing time is relatively cheap, fluid dynamic simulation has

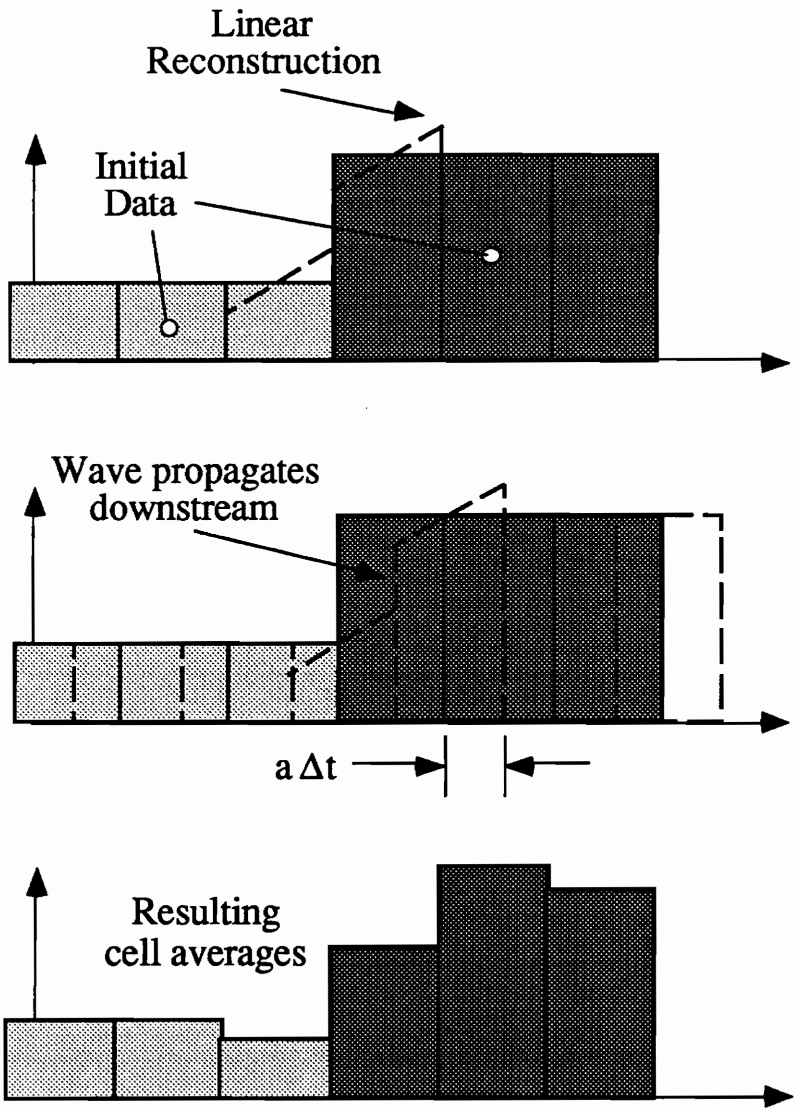

Figure 2. Cell averages around a discontinuity during the three stages of a numerical simulation.

grown with the economy of micro-chips in the research, application and development arenas. However, an unforeseen weakness lurks in the darkness for compressible algorithms⋯.

High-speed test cases are routinely run everyday; however, the low-speed performance of compressible algorithms is alarmingly poor. Why does a code that successfully simulates sophisticated high-speed flows fail so miserably for low speeds? The answer lies within the model equations. Steady-state calculations march in time until the state vector solves the steady terms in the unsteady formulation. The maximum time step is inversely proportional to the maximum wave speed in the system. At very low speeds, the fastest wave is a pressure disturbance moving at the speed of sound. The slow-moving convective waves propagate with the slow-moving fluid. To travel the same distance as the acoustic waves, the convective waves must pass through thousands of time steps. This is an elementary concept called stiffness and makes low-speed calculations prohibitively impractical. To fix the problem, we must accelerate the convective waves so that the convective and acoustic waves travel equivalent distances after a single time step. This process is called characteristic time stepping where each characteristic wave receives its own time step. An alternative explanation incorporates the condition number which is the ratio of the largest to the smallest characteristic wave in the system. A large departure of the condition number from unity means poorly matched characteristic speeds. The process of equalizing wave speeds is called preconditioning because the system of ill-conditioned waves is altered or preconditioned to yield a system with an optimal condition number.

Three primary reasons exist for using a compressible code to simulate nearly incompressible flows. Fundamentally, the compressible formulation is closer to reality. Many flows which are predominantly low-speed contain regions where compressibility is important. For example, a high-contraction-ratio wind tunnel has very-low-speed flow in the settling chamber which may then accelerate to nearly sonic conditions in the test section. A high-α fighter flies at its minimum controllable airspeed but has transonic flow around the leading edges of its wings. Many other examples exist where a compressible formulation is necessary but a characteristic-wave-speed disparity hampers convergence.

Secondly, today's aerospace computer banks and engineering minds are filled with years of research knowledge on the compressible formulation. The bill-paying efforts of a generation of numerical scientists should not go wasted. Rather, the weaknesses of the compressible codes should be improved to complement their abilities for the high-speed regime.

Even at low speeds, significant density gradients may result from large temperature variations of a gas in thermal non-equilibrium. We see this phenomenon in every-day life. Examples range from the induced velocity field around a paraffin wax candle to the boiling convective waves melting off of a hot summer's blacktop. The elimination of eigenvalue stiffness enhances the aerospace engineer's tool-box so that he can robustly model all realistic flows with reduced complication.

The preconditioned formulation is conceptually simple. A standard upwind flux-balance residual is calculated and multiplied by a filter called the preconditioning matrix. To not interrupt the physics of the flow, this matrix must be positive definite. The preconditioning matrix must not interfere with the natural wave propagation directions of the flow. For example, if we multiplied the residual by a negative-definite preconditioning matrix, the modeled numerical flow physics would move backwards in time. Energy methods assure that a preconditioning matrix derived from a symmetric hyperbolic system will be positive definite. Subsequently, the steady-state solution admitted by the preconditioned equations is identical to those of the original equations assuming the use of the same flux functions.

Part II of this thesis presents the analytical and numerical details of the preconditioning procedure. The objective here is to present the framework for deriving the matrix and demonstrate its implementation for three-dimensional flow with finite-rate chemistry. The derivations of all the matrices were accomplished in a timely and practical manner by using **Mathematica** programmed by Stephen Wolfram [82]. Both explicit and implicit time marching strategies are documented. The framework is set for viscous-flow preconditioning. The resulting matrices accelerate convergence at all Mach numbers and at all cell Reynolds numbers. However, since the hypersonic flow regime has negligible convective eigenvalue stiffness, particular benefits are realized at incompressible, subsonic and transonic flow speeds.

PART ONE

SPATIALLY HIGH-ORDER ACCURATE METHODS

CHAPTER

ONE

INTRODUCTION

The domain of high-order accurate numerics has been broadened with the development of uniformly accurate essentially non-oscillatory (ENO) schemes for structured rectangular grids by Ami Harten, Bjorn Engquist, Sukumar Chakravarthy and Stanley Osher [29]. The ENO schemes are a proposed class of algorithms which capture physically relevant high-frequency flow phenomenon with a uniform reconstruction procedure. An extension of the reconstruction-via-primitive-function principle to generalized coordinates was accomplished as a doctoral project by Jay Casper [8]. Throughout Part I of the present dissertation, an adequate Riemann solver is assumed. Of principal concern is the interpolation for the left and right fluid-dynamic states. Note that the Riemann solver used in this study is directionally split and not genuinely multi-dimensional. However, the reconstructions *are* multi-dimensional.

When numerically modeling a hyperbolic system, two methods are available for controlling oscillations around discontinuities for higher-order accuracy. Either the available stencil is reduced or limited, or the stencil shifts to use only a monotonic set for interpolation. An ENO scheme is a class of schemes (like TVD) which is uniformly accurate and whose stencil moves depending upon the data. The defining property of an ENO scheme is that the total variation of the solution is allowed to increase on the order of the truncation error. For fixed-stencil, higher-order methods, the total variation will increase with $\mathcal{O}(1)$ oscillations forming around discontinuities and high-gradient regions.

Many methods are available for reconstructing the piecewise continuous function from cell averages in the finite-volume formulation. With the ***primitive-function approach*** the cell averages are transformed into data which is known at discrete points. Interpolation with Lagrange polynomials can then be used to determine the reconstruction. By using the ***k-exact approach***, a degree-k polynomial whose cell averages are mapped onto a grid is subsequently reconstructed exactly. The k-exact reconstruction uses enough cell averages to determine the unknown coefficients in the reconstruction polynomial. The ***dimensionally split ENO approach*** overlaps two one-dimensional reconstructions which shift non-linearly with the data. This is not a true two-dimensional reconstruction and is limited to second order. A fixed stencil variable-extrapolation method coined monotonic-upstream centered scheme for conservation laws (MUSCL) provides a one-parameter family of reconstructions. MUSCL differencing is fast, simple, and is a directionally split method incorporated in most modern-day, preduction-level codes.

In a finite-volume formulation, an ENO scheme specifies requirements upon the reconstruction polynomial used for interpolating the state variables to a control-volume's cell faces. A general reconstruction polynomial approximates to high-order accuracy a function which generated the cell averages at the current time level. An ENO reconstruction is uniformly accurate meaning that all cell face interpolations are of the same order of accuracy (*i.e.* no local reductions in accuracy). Non-oscillatory behavior and uniform accuracy are accomplished in a hyperbolic system only if the stencil used in the reconstruction changes from cell to cell and time level to time level. Additionally, the moving stencil eliminates the necessity for traditional limiters which decrease accuracy at extrema in order to diminish the total variation of the solution.

An ENO scheme chooses the "best" cells locally in an asymptotic sense to reconstruct the pointwise solution at a cell face. In this way the algorithm incorporates gradient information to match the Taylor series. A straight-forward extension to unstructured grids of the essentially non-oscillatory behavior is not formally apparent. Efforts of Barth and Frederickson [3] have succeeded in reconstructing pointwise data with k-exact polynomials on general discretizations applicable to smooth fluid physics. More recently Abgrall [2] has designed an ENO scheme for application within the finite-element formulation. Although a reconstruction must only satisfy conservation on the cell in question, all of the reconstruction methods discussed in this thesis (two-dimensional reconstruction via primitive, k-exact, dimensionally split ENO, and traditional variable extrapolation MUSCL differencing) satisfy conservation for all the cells used to obtain the reconstruction. Interpolation algorithms should use reconstruction polynomials which satisfy conservation for not only the main parent cell, but also all the supporting cells to get a minimum reconstruction error for a given order of accuracy.

To achieve arbitrary high-order accuracy in two dimensions, the reconstruction via primitive incorporated in some ENO schemes or the k-exact reconstruction should formally be used. Both reconstructions satisfy the mean for the cells in their support stencil. However, these methods are computationally expensive both in terms of operation count and memory. As a compromise, a less rigorous, yet computationally simple ENO method is proposed which overlaps two one-dimensional interpolations giving an improvement to the MUSCL formulation. Because cross-derivative terms are neglected, this simpler method will achieve, at best, second-order accuracy, but will remain uniformly accurate, is less computational work, and the overlap of a pair of second-degree reconstructions has less reconstruction error than the overlap of a pair of first-degree reconstructions.

Chapter 2 contains a summary of popular reconstruction methods. The numerics of Casper's two-dimensional ENO reconstruction with arbitrary accuracy, k-exact reconstruction for general grids, and the dimensionally split ENO reconstruction are described as well as the reconstruction that the $\phi-\kappa$ formulation represents. We demonstrate a method for conserving the mean using k-exact reconstructions. A method is given for obtaining the smoothest stencil in generalized discretizations. The various algorithms' strengths and weaknesses are discussed for multi-dimensional applications. Steady-state convergence will always be a weakness for moving-stencil algorithms because a chattering stencil consistently changes the local truncation-error constants on every time step. Residual plots are shown for several steady-state test-cases which demonstrate this behavior.

Chapter 3 provides the necessary framework of integral error analysis. As a numerical example, reconstruction errors of an exact solution to the heat equation on a square grid are presented. Results are given for all of the reconstruction methods considered. The computational accuracy is determined from grid refinement and operations per grid cell show the CPU intensity of each scheme. The issues of efficiency and operations count are addressed in this dissertation for the first time. The dimensionally split ENO scheme is a proposed high-order uniformly accurate scheme which decreases the CPU intensity.

Numerical solution error to the Ringleb flow is given in Chapter 4. Results of the dimensionally split ENO scheme are also presented with a modification to reduce near-wall oscillations. The time history of a strong vortex moving through a normal shock in a diverging section is also presented. Conclusions and recommendations are given in Chapter 5.

CHAPTER TWO

RECONSTRUCTION RECIPES

Various methods for generating pointwise values at a cell face from cell averages are available in the literature. In this chapter, we will introduce four distinct methods and discuss their relative merits. First, consider the starting point of any aerodynamic simulation – the governing equations of fluid motion. In their one-dimensional differential form, the equations are

$$\frac{\partial Q}{\partial t} + \frac{\partial f}{\partial x} = 0, \tag{2–1}$$

where Q is the set of conserved variables and f contains the flux of of mass, momentum and energy. If we integrate over a discrete space-time domain $[t^n, t^{n+1}] \times [x_{i-1/2}, x_{i+1/2}]$, the fluid-dynamic equations in Eqn. (2–1) can be written exactly as

$$\frac{\bar{Q}^{n+1} - \bar{Q}^n}{\Delta t} + \frac{\hat{f}_{i+1/2} - \hat{f}_{i-1/2}}{\Delta x} = 0 \tag{2–2}$$

where

$$\bar{Q}^n \equiv \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} Q(x, t^n)\, dx \tag{2–3}$$

and

$$\hat{f}_{i+1/2} \equiv \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f\left[Q(x_{i+1/2}, t)\right] dt. \tag{2–4}$$

Any numerical procedure begins here and follows three distinct steps:

1) Reconstruction of the pointwise variable field from the cell-average values defined by Eqn. (2–3). Note that from this point onward, the governing equations of fluid motion are ***approximated***;

2) Numerical evaluation of the flux in Eqn. (2–4) using the cell-face values and an adequate Riemann solver for the flux;

3) Time evolution of the cell averages using a time-stepping procedure.

The objective of this portion of the research is precisely the first step described above. With a continuous, smooth function, $w(x)$, and a discretized domain, we can calculate the cell averages

$$\bar{w}(x_j) = \frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} w(\xi)\, d\xi. \tag{2–5}$$

We are concerned with the inverse problem. Given cell-average data, we want to accurately approximate the function, $w(x)$. This process is called reconstruction. We are reconstructing an unknown solution from its known cell averages using a polynomial basis set. We will denote the reconstruction polynomial $R(x; \bar{w})$, which reads "R of x given values of $\bar{w}$".

2.1 Total variation control

Ami Harten provided a necessary tool for analyzing non-linear schemes when he introduced, from applied-mathematics circles, the total variation of a discrete function [30]. The total variation at a specific time level is defined as

$$TV(\, w^n \,) \equiv \sum_{j=1}^{J} \left| w_j^n - w_{j-1}^n \right| . \tag{2–6}$$

A total variation diminishing (TVD) scheme updates the solution such that the solution's total variation diminishes as numerical time progresses. In notational form, the TVD requirement is

$$TV(w^{n+1}) \leq TV(w^n). \tag{2–7}$$

More accurately, a scheme that satisfies this inequality is a total variation non-increasing scheme.

The total variation monitors the oscillatory nature of a discrete data set. For example, a series of monotone non-decreasing numbers will have as its total variation the difference between the endpoint values. All interior values telescope in the summation. If after a single time step, the total variation increases, then new internal extrema have been generated. For a hyperbolic conservation law, the total variation of any initial solution is non-increasing, and we would hope that a numerical approximation could share the same quality. Unfortunately, this preservation restricts the accuracy, even for a non-linear scheme, to being uniformly second order [30]. A relaxation in the TVD requirement is necessary to enable higher-order accuracy. We must allow the total variation to increase, but only within the bounds of the truncation error. Schemes with such a relaxation are the essentially non-oscillatory (ENO) schemes [29]. The "essentially" reflects that the total variation is allowed to increase on the order of the numerical truncation error.

2.2 Simple One-dimensional Reconstructions

The group of cell-average data used to compute a reconstruction is known as the support set or support stencil. If a reconstruction is analytically integrated over the domain of the finite-volume cell and yields the known discrete cell average, the reconstruction is said to satisfy or conserve the mean. So, the solitary goal of a reconstruction polynomial is to satisfy the mean for every cell in the support stencil.

Perhaps it is fruitful to begin with a commonly used reconstruction. Current finite-volume production-level codes commonly incorporate Van Leer's variable-extrapolation method [68,69,70]. Van Leer's method (the $\phi - \kappa$ formulation or MUSCL approach) matches different cell averages for different combinations of ϕ and κ. For simplicity we adopt a locally defined coordinate, ξ, which is zero at the center of the j^{th} cell. With cells of constant width, the linear reconstruction satisfying the mean for the $j-1^{st}$ and j^{th} cells is

$$R_{-1}(\xi;\bar{w}) = \bar{w}_j + (\frac{\bar{w}_j - \bar{w}_{j-1}}{\Delta x})\xi. \tag{2--8}$$

The linear reconstruction which satisfies the mean for the $\{j,\, j+1\}$ stencil is

$$R_0(\xi;\bar{w}) = \bar{w}_j + (\frac{\bar{w}_{j+1} - \bar{w}_j}{\Delta x})\xi. \tag{2–9}$$

The third-order, quadratic reconstruction is determined by satisfying the mean for three cell averages. This reconstruction is obtained just like all others – by satisfying conservation for the support set. The unique quadratic reconstruction that uses the $\{j-1,\, j,\, j+1\}$ support stencil is

$$R(\xi;\bar{w}) = \bar{w}_j + \frac{\bar{w}_j - \bar{w}_{j-1}}{\Delta x}\xi + \frac{\bar{w}_{j-1} - 2\,\bar{w}_j + \bar{w}_{j+1}}{2\Delta x^2}(\xi^2 + \xi\Delta x - \frac{\Delta x^2}{12}). \tag{2–10}$$

As will be shown later in Section 2.2.1, this reconstruction is third-order accurate at every location, ξ. Specifically, this polynomial at the right cell face ($\xi = \Delta x/2$) gives a third-order accurate value for the left state, $w^-_{j+1/2}$, which is identical to that obtained from the $\phi-\kappa$ formulation with $\kappa = 1/3$.

The MUSCL-differencing approach gives a one-parameter family of reconstruction polynomials evaluated at the cell face. For the left state on the right cell face, the values of the piecewise-constant, linear, and quadratic reconstruction are kappa dependent and are evaluated from

$$w^-_{j+\frac{1}{2}} = \bar{w}_j + \frac{\phi}{4}\Big[(1+\kappa)\Delta + (1-\kappa)\nabla\Big]\bar{w}_j, \tag{2–11}$$

where ϕ is an on/off switch for high-order accurate interpolation. For the right state on the left face (*i.e.* $\xi = -\Delta x/2$) the $\phi-\kappa$ formulation is

$$w^+_{j-\frac{1}{2}} = \bar{w}_j - \frac{\phi}{4}\Big[(1+\kappa)\nabla - (1-\kappa)\Delta\Big]\bar{w}_j. \tag{2–12}$$

Engineers are accustomed to fitting data points with Lagrange polynomials because of their training in finite-difference methods and courses in numerical methods. However, with a reconstruction the data is known only on the average and *not* at a discrete point. In the finite-volume formulation which uses cell average data, a reconstruction from the cell averages should be utilized for high-order accurate methods on non-uniform grids.

2.3 Essentially Non-Oscillatory Reconstruction

The new twist associated with the essentially non-oscillatory schemes is a nonlinear method for selecting the support stencil. The reconstruction is then obtained by conserving the mean. The designers of the ENO schemes have specifically set down three defining conditions for their algorithms [29,30]. The three conditions are:

1. At all points where $w(x)$ is smooth, a reconstruction from cell averages must be high-order accurate, or

$$R(x;\bar{w}) = w(x) + \mathcal{O}(h^n). \tag{2–13}$$

The truncation error $\mathcal{O}(h^n)$ can be written as $e(x)\,h^n$ and can lose an order of accuracy at a discontinuity.

2. The reconstruction must be conservative in the sense that

$$\bar{R}(x_j;\bar{w}) = \bar{w}_j, \tag{2–14}$$

or,

$$\frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} R(\xi;\bar{w})\,d\xi = \bar{w}(x_j). \tag{2–15}$$

This means that the reconstruction itself must satisfy the same cell average as the original function.

3. The reconstruction must be essentially non-oscillatory, which in terms of the total variation means

$$TV\big[R(x;\bar{w})\big] \leq TV\big[w(x)\big] + \mathcal{O}(h^n). \tag{2–16}$$

This simply means that oscillations are allowed only on the level of the truncation error. Arbitrary accuracy and elimination of $\mathcal{O}(1)$ oscillations around discontinuities are achieved at the expense of a slightly increasing total variation.

2.3.1 The Primitive Function

A mathematically rigorous approach to reconstruction incorporates the *primitive* function [9,29,30]. The primitive function simply transforms cell-average data into discrete values at a point where classical numerical methods apply. By design, the function conserves the mean and helps to determine the truncation error. The foundation for interpolating with cell averages begins with the one-dimensional reconstruction of a scalar variable.

Given cell averages, $\bar{w}_j$, of a piecewise smooth function, $w(x)$, we can immediately evaluate discrete values of the *primitive* function, denoted $W(x)$. The analytic form is defined starting from an arbitrary point, j_0, as

$$W(x) \equiv \int_{x_{j_0-\frac{1}{2}}}^{x} w(\xi)\, d\xi. \tag{2-17}$$

The discrete values follow from a summation

$$W(x_{j+\frac{1}{2}}) = \sum_{i=j_0}^{j} \Delta x_i\, \bar{w}_i. \tag{2-18}$$

From Eqn. (2-17), the desired reconstructed function, $w(x)$, is the derivative of the analytic primitive function, or

$$w(x) = \frac{dW}{dx}. \tag{2-19}$$

We can apply a Lagrange interpolation polynomial to the discrete values of the primitive function, and obtain an ENO interpolation polynomial if we use the "smoothest" stencil. We can then determine an approximation to $w(x)$, which we call $R(x; \bar{w})$, by differentiating the Lagrange polynomial. This procedure conserves the mean and does not require uniformity of the mesh. Note that the primitive function, $W(x)$, is by one derivative smoother than $w(x)$.

The primitive function is particularly useful in determining the truncation error of a reconstruction. To this end, the following theorem taken from Burden and Faires [5] is important to any discussion on interpolation and reconstruction.

<u>**THEOREM :**</u> If $x_j, x_{j+1}, \ldots, x_{j+k}$ are distinct numbers in the interval $[a,b]$ and $f \in C^{k+1}[a,b]$, then, for each x in $[a,b]$, a number, $\xi(x)$, exists in (a,b) with

$$f(x) = P_k(x) + \frac{f^{(k+1)}\{\xi(x)\}}{(k+1)!} \prod_{i=0}^{k}(x - x_{j+i}),$$

where $P_k(x)$ is the Lagrange polynomial

$$P_k(x) = \sum_{m=0}^{k} f(x_m)\, L_{k,m}(x)$$

and

$$L_{k,m} = \prod_{\substack{i=0 \\ i\neq m}}^{k} \frac{(x - x_{j+i})}{(x_{m+i} - x_{j+i})}.$$

This theorem states that if $k+1$ discrete data points are fit with a degree-k Lagrange polynomial through the points, then the leading term in the interpolation is $k+1^{st}$ order. In this dissertation, a degree-k interpolation is denoted as n^{th}-order where $n = k+1$. For example, if we have data at x_{j-1}, x_j, and x_{j+1} called f_{j-1}, f_j, and f_{j+1}, then the Lagrange polynomial which passes through the three data points is

$$\begin{aligned} P_2(x) = & f_{j-1}\, \frac{(x - x_j)(x - x_{j+1})}{(x_{j-1} - x_j)(x_{j-1} - x_{j+1})} + f_j\, \frac{(x - x_{j-1})(x - x_{j+1})}{(x_j - x_{j-1})(x_j - x_{j+1})} \\ & + f_{j+1}\, \frac{(x - x_{j-1})(x - x_j)}{(x_{j+1} - x_{j-1})(x_{j+1} - x_j)}. \end{aligned} \tag{2–20}$$

Or, if we define $\xi \equiv x - x_j$ and assume constant cell widths, the polynomial is

$$P_2(x) = f_{j-1}\frac{\xi(\xi - \Delta x)}{2\Delta x^2} - f_j\frac{(\xi + \Delta x)(\xi - \Delta x)}{\Delta x^2} + f_{j+1}\frac{\xi(\xi + \Delta x)}{2\Delta x^2}. \tag{2–21}$$

The cell-face value of the function is then

$$P_2(x_{j+1/2}) = f_j + \frac{3}{8}\Delta f_j + \frac{1}{8}\nabla f_j + \mathcal{O}(\Delta x^3). \tag{2–22}$$

If we use the MUSCL differencing as in Eqn. $(2-11)$, but with data known at a point as in a *finite-difference* code, the above polynomial corresponds to a kappa value of $\kappa = 1/2$. In a finite-volume method, a third-order reconstruction which satisfies the mean for the three cells in the support stencil results from $\kappa = 1/3$. The $\phi - \kappa$ formulation may be used in a finite-difference code to reconstruct from point data or in a finite-volume code to reconstruct from cell-average data. However, the corresponding third-order values of kappa (κ) differ. Satisfaction of the mean in finite-volume formulations is the analog to Lagrange polynomials passing through data points in the finite-difference approach.

Thus, a question arises. How can we use the primitive function to determine the truncation error for the three most commonly incorporated MUSCL reconstructions: first-order, second-order upwind and third-order upwind-biased?

2.3.2 Determining Truncation Error

First-order Truncation Error

The first-order reconstruction is the piecewise-constant value used in Godunov's original scheme [23]. This is a frivolous way of saying that the reconstruction is the cell average itself. However, at the cell *center*, the cell average is a *second*-order approximation to the function that determined the cell average. To see this, we expand the Taylor series of $W(x_{j+1/2})$ about an arbitrary point, x, which gives

$$W(x_{j+\frac{1}{2}}) = W(x) + \frac{dW}{dx}(x_{j+\frac{1}{2}} - x) + \sum_{n=2}^{\infty} \frac{d^n W}{dx^n} \frac{(x_{j+\frac{1}{2}} - x)^n}{n!} \qquad (2-23)$$

From Eqn. (2-18), we see that

$$\bar{w}_j = \frac{W(x_{j+\frac{1}{2}}) - W(x_{j-\frac{1}{2}})}{\Delta x} \qquad (2-24)$$

Since $w(x)=dW/dx$, Eqns. (2–23) and (2–24) immediately imply that

$$\begin{aligned}\bar{w}_j = w(x) + \frac{d^2W}{dx^2}\frac{(\xi - \frac{\Delta x}{2})^2 - (\xi + \frac{\Delta x}{2})^2}{2\Delta x} \\ + \sum_{n=3}^{\infty}\frac{d^nW}{dx^n}\frac{(-1)^n}{n!}\frac{[(\xi - \Delta x/2)^n - (\xi + \Delta x/2)^n]}{\Delta x}\end{aligned} \qquad (2–25)$$

So, at the cell center, $\xi=0$, the cell average is a second-order approximation to $w(x)$ at $x=x_j$. Of course, everywhere else the reconstruction is first-order accurate. The truncation error of high-order reconstructions can be determined similarly.

Second-order Upwind Truncation Error

For the upwind case, a second-order approximation to $w(x)$ which uses the cell averages at j and $j-1$ will require *three* values of the primitive function: $W_{j+1/2}$, $W_{j-1/2}$, and $W_{j-3/2}$. A quadratic passing through these three points results in a third-order Lagrange polynomial, which is

$$\begin{aligned}P_2(\xi) = W_{j+1/2}\frac{(\xi + \Delta x/2)(\xi + 3\Delta x/2)}{2\Delta x^2} - W_{j-1/2}\frac{(\xi - \Delta x/2)(\xi + 3\Delta x/2)}{\Delta x^2} \\ + W_{j-3/2}\frac{(\xi - \Delta x/2)(\xi + \Delta x/2)}{2\Delta x^2} + \frac{d^3W}{dx^3}\frac{\xi(\xi + \Delta x/2)(\xi - \Delta x/2)}{3!}.\end{aligned} \qquad (2–26)$$

Obviously, the derivative of $P_2(x)$ will be one order less accurate. At the cell face, the derivative is the second-order reconstruction to the cell-average field. After some trivial algebra, we find the derivative to be

$$\begin{aligned}\frac{dP_2}{dx} = \frac{W_{j+1/2} - W_{j-1/2}}{\Delta x} \\ + \frac{1}{2}\left[\frac{W_{j+1/2} - W_{j-1/2}}{\Delta x} - \frac{W_{j-1/2} - W_{j-3/2}}{\Delta x}\right] \\ + \frac{d^3W}{dx^3}\frac{\Delta x^2}{2\cdot 3!},\end{aligned} \qquad (2–27)$$

which reduces to

$$R(x;\bar{w}) = \bar{w}_j + \frac{1}{2}(\bar{w}_j - \bar{w}_{j+1}) + \frac{d^3W}{dx^3}\frac{\Delta x^2}{12}. \qquad (2–28)$$

Third-order Upwind-biased Truncation Error

The upwind-biased third-order reconstruction must use the four primitives: $W_{j+3/2}$, $W_{j+1/2}$, $W_{j-1/2}$ and $W_{j-3/2}$. After differentiation of the Lagrange polynomial which passes through these data points, we obtain a reconstruction which incorporates the cell averages at $j-1$, j, and $j+1$. This reconstruction is

$$\begin{aligned} R(\xi;\bar{w}) = \bar{w}_j &+ \frac{\bar{w}_j - \bar{w}_{j-1}}{\Delta x}\xi \\ &+ \frac{\bar{w}_{j-1} - 2\,\bar{w}_j + \bar{w}_{j+1}}{2\Delta x^2}\,(\xi^2 + \xi\Delta x - \frac{\Delta x^2}{12}) \\ &+ \frac{d^4W}{dx^4}\frac{d}{d\xi}\left[\frac{\xi(\xi+\Delta x/2)(\xi-\Delta x/2)(\xi-3\Delta x/2)}{4!}\right]. \end{aligned} \tag{2–29}$$

The $\phi-\kappa$ formulation with $\kappa = 1/3$ is the above polynomial evaluated at the cell face. Including the truncation error term, this interpolation is

$$R(\xi = \Delta x/2; \bar{w}) = \bar{w}_j + \frac{1}{4}\Big[(1+1/3)\Delta + (1-1/3)\nabla\Big]\bar{w}_j - \frac{d^4W}{dx^4}\frac{\Delta x^3}{2\cdot 4!}. \tag{2–30}$$

2.3.3 A ENO Definition of "Smoothness"

The ENO algorithm chooses the "smoothest" stencil in a necessarily non-linear fashion. The smoothness of the discrete data, $W(x_{j+1/2})$, is determined with a divided-difference table. If $W(x)$ is $C^n[x_{j+1/2}, x_{j+1/2+n}]$, then

$$W[x_{j+\frac{1}{2}}, x_{j+\frac{3}{2}}, \ldots, x_{j+\frac{1}{2}+n}] = \frac{1}{n!}W^{(n)}(\xi), \tag{2–31}$$

for $\xi \in [x_{j+1/2}, x_{j+1/2+n}]$. So the absolute value of the divided differences gives a measure of the smoothness of $W(x)$ in the interval.

The procedure to determine the smoothest support set is as follows. Define the stencil of $m+1$ points, $S_m(j)$,

$$S_m(j) \equiv \{x_{j+\frac{1}{2}}, x_{j+\frac{3}{2}}, \ldots, x_{j+\frac{1}{2}+m}\}. \tag{2–32}$$

Then for reconstruction on the interval $[x_{j-1/2}, x_{j+1/2}]$, set

$$i_1(j) = j. \tag{2–33}$$

For high-order reconstruction, we determine the best cells by comparing the magnitudes of the divided differences. Consider adding a cell to the left and right of the current cell(s) as in Fig. (3) and compare the corresponding two divided differences. We choose the smaller difference in magnitude to obtain the stencil that is smoothest in an asymptotic sense. Explicitly, identify

$$S_m^L \equiv S_m(i_{m-1}(j) - 1), \tag{2–34}$$

and

$$S_m^R \equiv S_m(i_{m-1}(j)). \tag{2–35}$$

Now build the stencil from its preceding stencil according to

$$i_m(j) = \begin{cases} i_{m-1}(j) - 1, & \text{if } \left|W[S_m^L]\right| \leq \left|W[S_m^R]\right|; \\ i_{m-1}(j), & \text{otherwise.} \end{cases} \tag{2–36}$$

The stencil is built for $m = 1, 2, 3, \ldots, n$, where n is the order of the scheme.

Particularly note that the reconstruction does not necessarily pass through the cell centers, but must, instead, satisfy conservation of all the cell averages in the stencil.

2.3.4 Arbitrarily High-Order Accurate Two-Dimensional ENO

The application of the ENO scheme to multi-dimensions is documented by Harten *et.al.* [29] and Casper [7,8]. The two-dimensional ENO schemes based upon the primitive function operate upon data in one dimension at a time. Assume we have a Cartesian grid $jdim \times kdim$ in the (x, y) plane. Consider a reconstruction for the pointwise values of $w(x, y)$ on constant-x faces shown in Fig. (4).

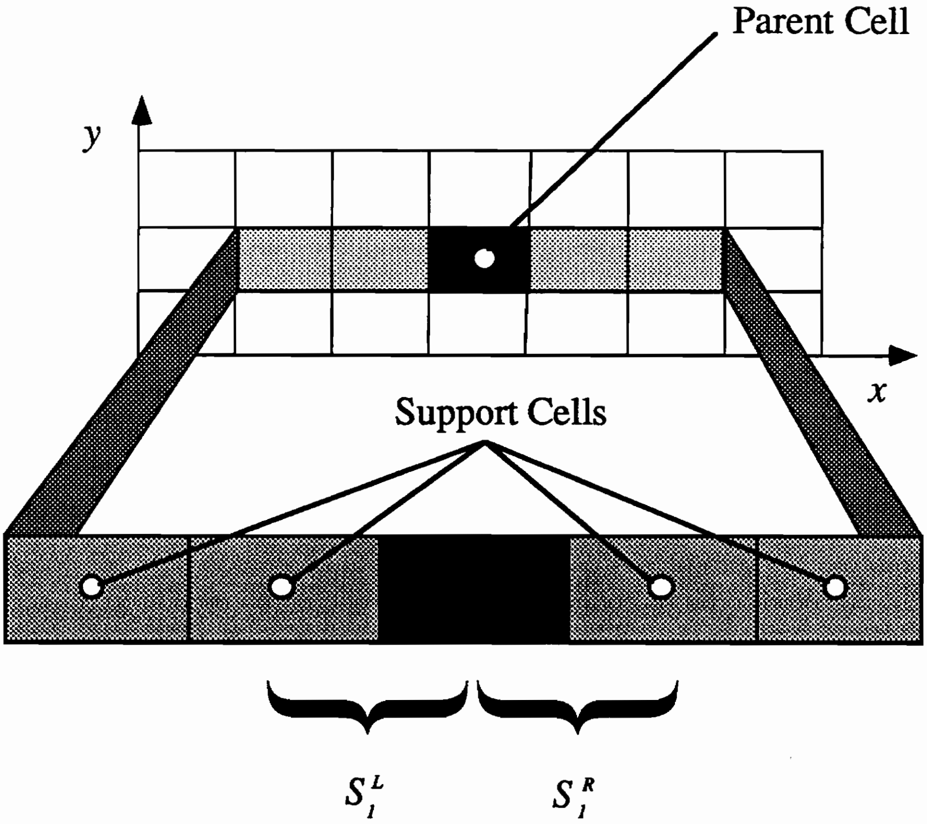

Figure 3. Dimensionally split support stencil centered around the first-order parent cell containing all contiguous sets of three cells used for a degree-two reconstruction.

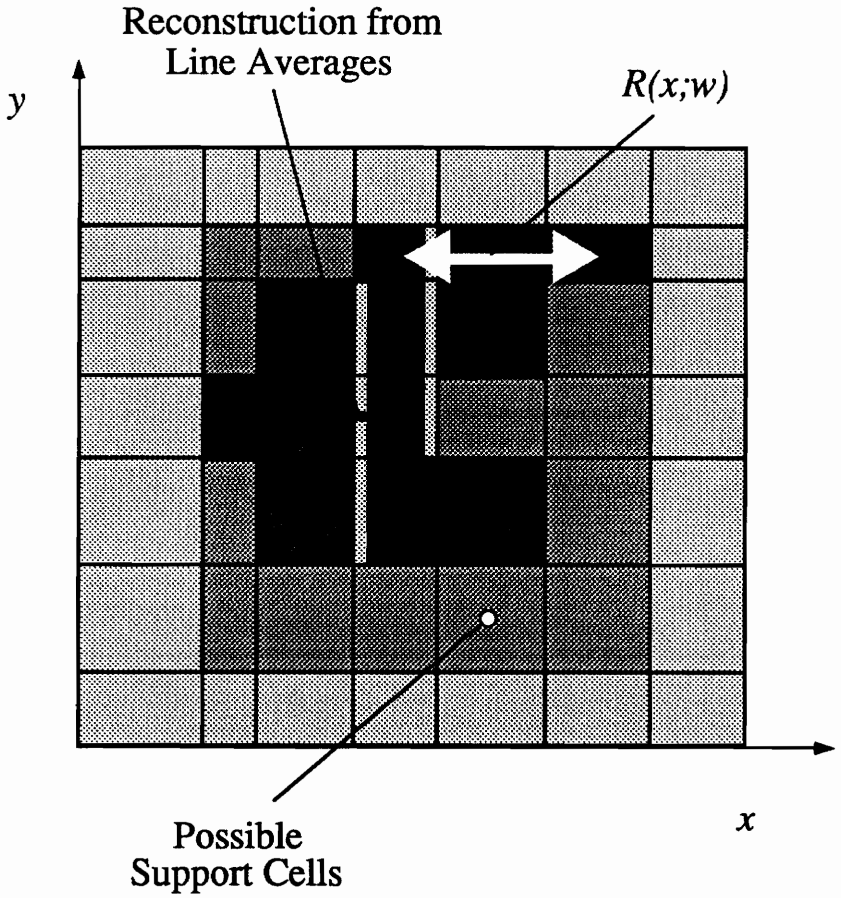

Figure 4. Parent cell (black), the multi-dimensional support stencil (darkest gray), set of all contiguous support cells (medium gray) and the physical domain (lightest gray) for a full two-dimensional ENO reconstruction of order two. Shown is the reconstruction procedure for the constant-x faces of the parent cell.

First, from a set of $jdim$ cell averages (i.e. constant k row)

$$\bar{w}_{j,k} = \frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{1}{\Delta y_k} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} w(x,y)\, dy\, dx, \tag{2–37}$$

accurate line averages are reconstructed via the primitive function. This procedure is repeated for all $jdim$ cells and all $kdim$ rows. We then have a reconstruction for the line averages in the y-direction

$$R(x;\bar{w}) = \frac{1}{\Delta y_k} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} w(x,y)\, dy. \tag{2–38}$$

Another data set for reconstruction is generated by evaluating $R(x;\bar{w})$ at specific values of x, namely at both vertical cell faces ($x = x^+_{j-1/2}, x^-_{j+1/2}$) for every cell. Then with accurate line averages, pointwise data is obtained by a reconstruction in the y-direction (*i.e.* with the reconstructed line averages, $R(x^-_{j+1/2};\bar{w})$, we obtain the pointwise reconstruction $R^2(y; R(x^-_{j+1/2};\bar{w}))$, which is a polynomial in the y-direction of degree, k, at the right face). Here the notation $x^-_{j+1/2}$ denotes the left state at $x_{j+1/2}$ and $x^+_{j-1/2}$ denotes the right state at $x_{j-1/2}$. R^2 is the pointwise reconstruction operating on R and is evaluated at the Gauss points of the flux integral. The notation at the the left face, $x = x_{j-1/2}$, is then $R^2(y; R(x^+_{j-1/2};\bar{w}))$. Two similar reconstructions are done for the constant-y faces. From these left and right states the flux integral can be accurately evaluated at the Gauss points with any flux-splitting technique.

Although notationally nasty, the reconstruction using line averages is crucial for arbitrary accuracy in two dimensions. For arbitrary three-dimensional accuracy, area averages, then line averages, and finally pointwise data must be reconstructed. The ENO scheme above logically chooses the smoothest stencil on each level from divided-difference tables of cell averages, area averages, and line averages in that order. The reconstruction satisfies the mean for all the cells in the support stencil and is of the form of a tensor-product polynomial, or more rigorously,

$$R(x,y,z;\bar{w}) = \sum_{i=0}^{k} \sum_{j=0}^{k} \sum_{l=0}^{k} c_{i,j,l}\, x^i y^j z^l \tag{2–39}$$

Note that for the reconstruction in two dimensions the ENO scheme will include $(k+1)^2$ cells and in three dimensions $(k+1)^3$ cells. This is not for each cell, but for each cell *face*, and the computation is just beginning.

Application to Mesh Curvature

The previous reconstruction using the primitive function is applicable to stretched rectangular grids. However, a differentiable mapping from (x, y) space to (ξ, η) space is required for the extension to meshes with curvature. If the transformation is not known, this requires a polynomial of degree-k passing through the grid points. The approximation to the pointwise function, $w(x, y)$, in generalized coordinates is then

$$\tilde{w}(x,y) = \frac{R^2(x,y;A\bar{w})}{R^2(x,y;A)} \tag{2–40}$$

where $R^2(x, y; A\bar{w})$ and $R^2(x, y; A)$ are reconstructions for the pointwise functions $w(x, y)\, J(x, y)$ and $J(x, y)$, respectively. $J(x, y)$ is the transformation Jacobian from (x, y) space to (ξ, η) space.

$$A = \frac{1}{\Delta x \Delta y} \int J(x,y)\, dx\, dy \tag{2–41}$$

This formulation will maintain the free stream to the order of the truncation error in the finite-volume setting.

In review, what computations are necessary for reconstruction in generalized coordinates? We must calculate the primitive function from cell averages. A nonlinear selection algorithm chooses the best cells from which to reconstruct the solution at a cell face. The reconstruction is determined by differentiating a polynomial passing through discrete values of the primitive function. A reconstruction of the transformation Jacobian from cell areas (or volumes) is necessary. Finally, high-order accurate metrics are needed at each flux-integration Gauss point. This entire process is far too impractical for routine use on modern-day computers.

2.4 *K*-exact Reconstruction

K-exact reconstruction is a simpler approach which makes no promises about the smoothness of the reconstructed solution. The method is multi-dimensional by nature and remains in the physical solution domain. The goal is simple – directly satisfy the mean for every cell in the stencil using the minimum number of cells necessary. *K*-exact reconstructions must satisfy three design criteria as specified by Barth and Frederickson [3].

1. The reconstruction must satisfy conservation of the mean of the parent and support cells as in Eqn. (2–3). If we consider a general polynomial $R(x,y,z;\bar{w})$,
$$R(x,y,z;\bar{w}) = \sum_{i=0}^{k} \quad \sum_{j=0}^{k-i} \quad \sum_{l=0}^{k-(i+j)} c_{i,j,l}\; x^i y^j z^l, \qquad (2\text{–}42)$$
then the reconstruction must satisfy the mean for $(k+1)(k+2)(k+3)/6$ cells. For one and two-dimensional polynomials, $k+1$ and $(k+1)(k+2)/2$ cell means must be conserved, respectively. The support set must be in the neighborhood of the parent cell.
2. $R(x,y,z;\bar{w})$ must be k-exact, which means that for polynomial functions, $w(x,y,z)$, of degree k or less the reconstruction is exact.
3. The reconstruction process should be computationally efficient.

In contrast to the described ENO reconstruction, the same k-exact reconstruction polynomial is used for all Gauss points on the cell's faces. While the coefficients in $R(x,y,z;\bar{w})$ depend upon the support stencil, the integral of $x^i\, y^j\, z^l$ depends only upon the geometry of the grid. Applying the averaging operation of Eqn. $(2-15)$ in three dimensions, the mean of the reconstruction is

$$\bar{R}(x,y,z;\bar{w}) = \sum_{i=0}^{k} \quad \sum_{j=0}^{k-i} \quad \sum_{l=0}^{k-(i+j)} c_{i,j,l} \frac{1}{\Omega} \int x^i y^j z^l \, d\Omega. \qquad (2\text{–}43)$$

Notice that the integral is strictly grid dependent while the coefficients are solution dependent.

A simple example can clarify the k-exact reconstruction procedure. If we want a quadratic reconstruction, $k=2$, in two dimensions, the corresponding polynomial is

$$R(x, y, \bar{w}) = c_{0,0} + c_{1,0}\,x + c_{0,1}\,y + c_{2,0}\,x^2 + c_{1,1}\,x\,y + c_{0,2}\,y^2. \tag{2–44}$$

Note that the two-dimensional ENO reconstruction would include three additional terms in the reconstruction: $x^2\,y$, $x\,y^2$, and $x^2\,y^2$. A comparison of the k-exact and ENO polynomial basis set for degree-two reconstruction is shown in Fig. (5).

We require that this reconstruction satisfy the mean for six cells as shown in Fig. (6). In equation form, this requirement gives a system of equations, $A\,x = b$, where A is

$$\begin{bmatrix} \frac{1}{\Omega_1}\int d\Omega & \frac{1}{\Omega_1}\int x\,d\Omega & \frac{1}{\Omega_1}\int y\,d\Omega & \frac{1}{\Omega_1}\int x^2\,d\Omega & \frac{1}{\Omega_1}\int x\,y\,d\Omega & \frac{1}{\Omega_1}\int y^2\,d\Omega \\ \frac{1}{\Omega_2}\int d\Omega & \frac{1}{\Omega_2}\int x\,d\Omega & & & & \vdots \\ \frac{1}{\Omega_3}\int d\Omega & & \frac{1}{\Omega_3}\int y\,d\Omega & & & \vdots \\ \frac{1}{\Omega_4}\int d\Omega & & & \frac{1}{\Omega_4}\int x^2\,d\Omega & & \vdots \\ \frac{1}{\Omega_5}\int d\Omega & & & & \frac{1}{\Omega_5}\int x\,y\,d\Omega & \vdots \\ \frac{1}{\Omega_6}\int d\Omega & \cdots & \cdots & \cdots & \cdots & \frac{1}{\Omega_6}\int y^2\,d\Omega \end{bmatrix} \tag{2–45}$$

and the vectors of unknown coefficients and known mean values, x and b, are

$$x = \left\{ \begin{matrix} c_{0,0} \\ c_{1,0} \\ c_{0,1} \\ c_{2,0} \\ c_{1,1} \\ c_{0,2} \end{matrix} \right\} \quad \text{and} \quad b = \left\{ \begin{matrix} \bar{w}_1 \\ \bar{w}_2 \\ \bar{w}_3 \\ \bar{w}_4 \\ \bar{w}_5 \\ \bar{w}_6 \end{matrix} \right\}. \tag{2–46}$$

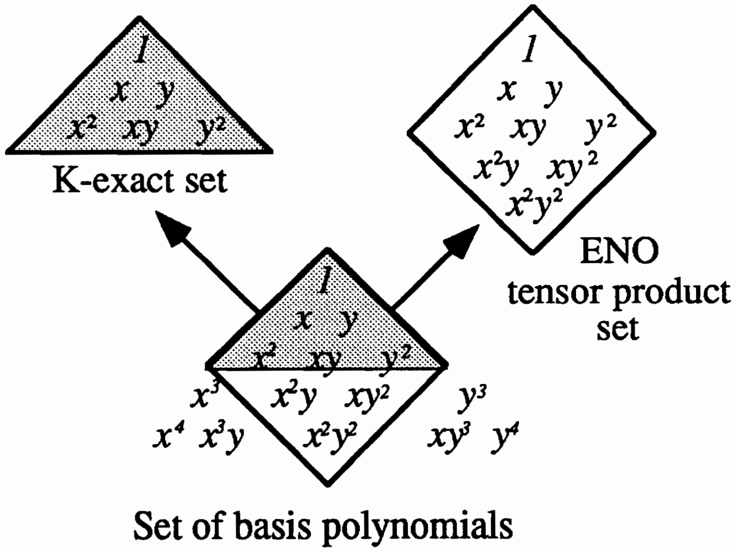

Figure 5. Degree-two polynomial basis set for k-exact and ENO reconstructions.

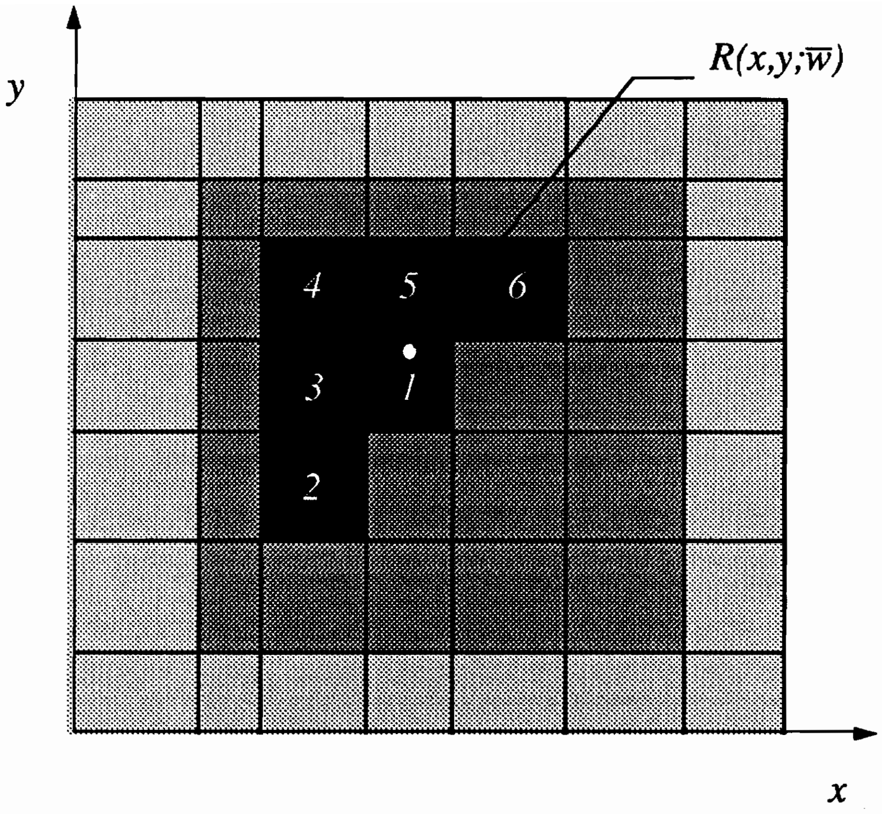

Figure 6. Parent cell (black) and K-exact reconstruction support stencil (darkest gray) for degree-two reconstruction in two dimensions.

Inverting the left-hand side, we obtain the reconstruction for the field of six cells. This reconstruction is then evaluated at the (x, y) locations of the cell-face Gauss points.

If the stencil remains *fixed* during the time-evolution process, the square matrix, A, can be decomposed into lower and upper triangular matrices prior to all time integration. Then, the system of equations can be solved for the coefficients, $c_{i,j}$, by forward and backward substitution given a time level's cell averages, $\bar{w}$. If the stencil should have to change from iteration to iteration as with ENO schemes, an LU decomposition at every iteration is required since the support set changes.

2.5 Dimensionally Split ENO Reconstruction

A one-dimensional reconstruction polynomial, $R(x; \bar{w})$, has $k + 1$ constants which are determined by satisfying conservation for $k + 1$ cells in the stencil. Specifically, if the lowest cell number in a contiguous stencil of cells is i, then the reconstruction, $R(x; \bar{w})$, must satisfy the cell averages of cells i through $i + k$. An example support set is shown in Fig. (7).

For two-dimensional meshes we apply two such reconstructions to cell averages. The generated set of line averages is not further reduced to pointwise values as before and, subsequently, arbitrary accuracy is not achieved. The hope is to decrease CPU time of the ENO scheme while giving a better interpolation algorithm than the traditional fixed-stencil interpolations. Dimensionally split ENO accentuates physically real extrema and retains uniform high-order accuracy. The dimensionally split ENO reconstruction may be summarized in formula as

$$R_{DS}(x, y; \bar{w}) = R(x; \bar{w}) + R(y; \bar{w}) \tag{2–47}$$

For a rectangular grid the polynomial in the x-direction, $R(x; \bar{w})$, is applied to the vertical faces. Likewise, the polynomial in the y-direction is applied to the horizontal cell faces.

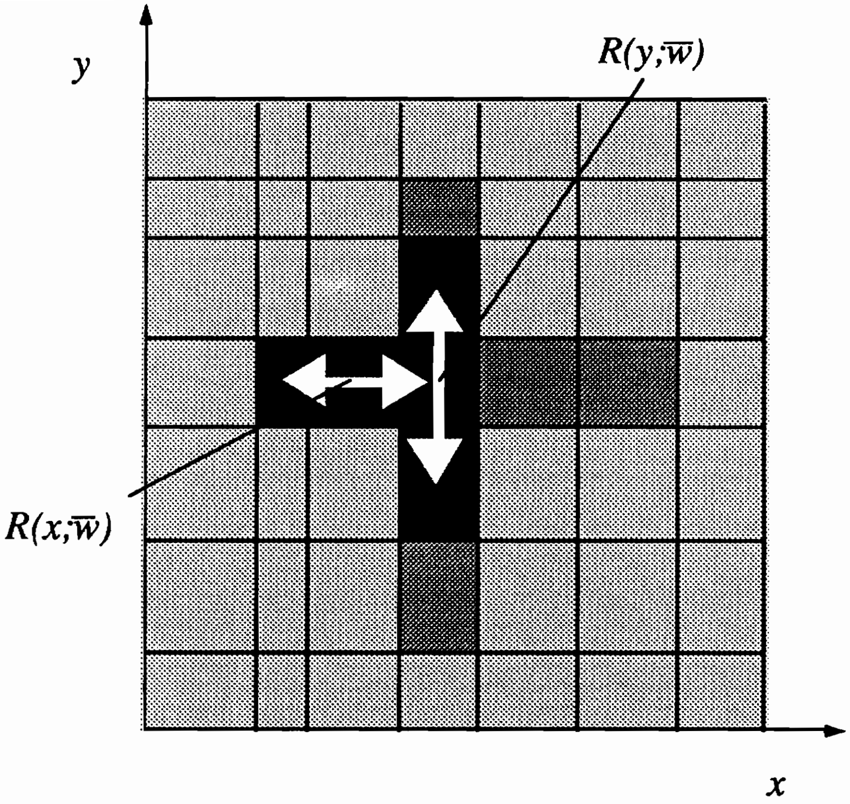

Figure 7. Parent cell (black) and dimensionally split ENO reconstruction support stencil (arrows) for degree-two reconstruction in two dimensions.

2.6 Merits and Deficiencies

The adaptive-stencil, high-order accurate methods inherently will have poor high-frequency damping characteristics associated with the spatial differencing operator. A steady-state solution will only be possible if the time-evolution operator damps high-frequency noise. For a steady state, errors must propagate from the domain and not return. Noise generation is constantly being re-introduced by the time-dependent chatter of the adaptive stencils. Even with perfectly absorbing boundary conditions, which would not reflect outgoing waves, the adaptive-stencil algorithms will constantly bring in truncation-error noise from the changing interpolations and prevent steady-state convergence to machine zero. For this reason, steady-state solutions obtained by convergence to machine zero and adaptive-stencil algorithms used to obtain arbitrarily high-order uniform accuracy are incompatible. We do not mean to convey that p-refinement (increasing accuracy on the same mesh) gives worse solutions, but merely that the sufficient number of iterations required to reflect a steady-state solution for a non-zero residual is not apparent. High-order uniformly accurate schemes lend themselves best to time-dependent problems where fluxes are inherently unbalanced. Typical residual histories for several steady-state problems are shown in Fig. (8). Notice that the Ringleb solutions are converged to machine zero because the monotonicity of the solution fixes the stencil for all time. The details of Ringleb's flow are further discussed in Chapter 4.

Fixed-stencil algorithms do not suffer the same woes as do the moving-stencil algorithms in the march to a steady state. However, arbitrarily high-order accurate algorithms are not frequently coded in current production-level codes. The k-exact reconstruction method is by far the easiest to implement for general discretizations. All that is necessary is the precise flow domain and the inverted space matrix of polynomials to satisfy a mean quantity. We can control oscillations at discontinuities by either using a fixed-stencil algorithm which constrains the accuracy yet attains steady-state convergence or use a moving-stencil algorithm which allows for

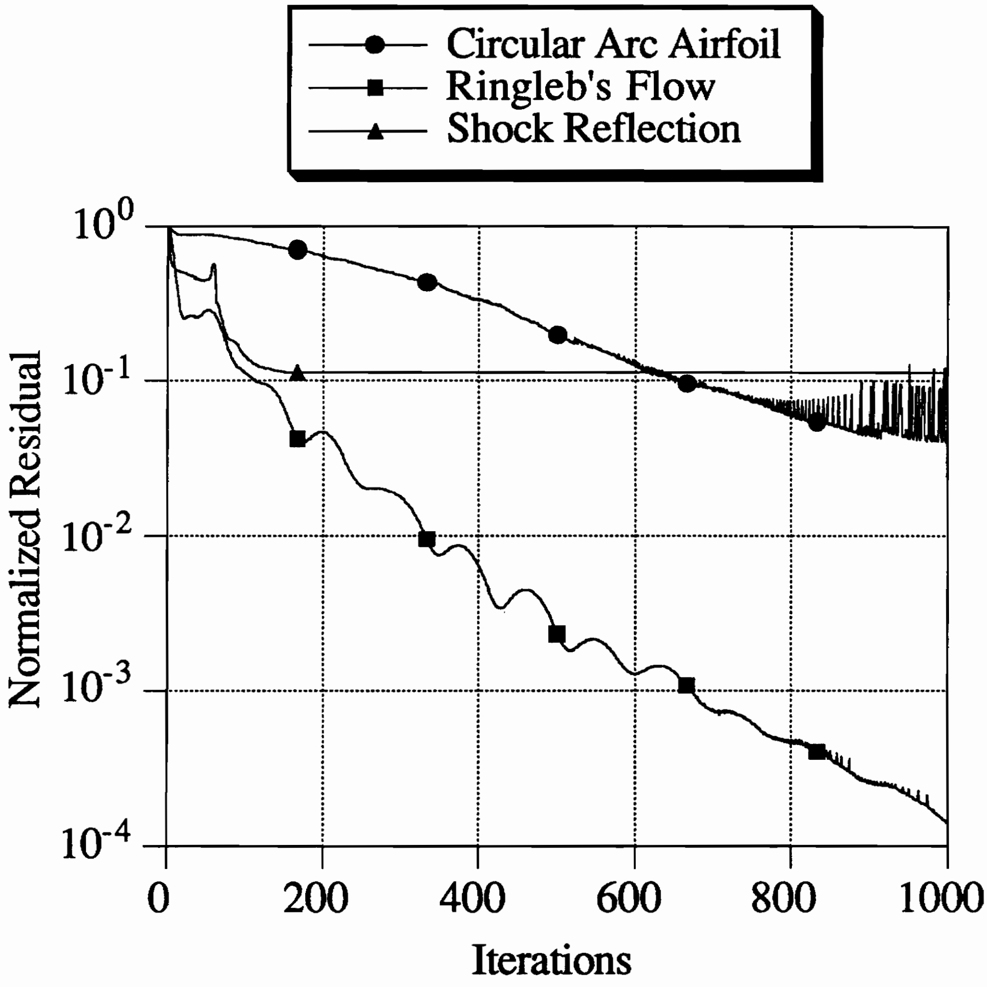

Figure 8. Typical residual histories for problems using a chattering interpolation stencil.

arbitrarily high-order accuracy but does not achieve steady-state convergence in the sense of a machine-zero residual. A fixed-stencil algorithm is computationally much cheaper than its moving-stencil counterpart. For a fixed stencil, the support cells are decreased sufficiently until the reconstruction is monotonic with all the surrounding neighbors.

Oh, but there are other concerns as well. All the classical literature on high-order accuracy methods assumes an abundance of support stencils and the numerical algorithm simply makes the proper choice. For a real flow simulation, this is rarely possible. A discontinuity near a boundary severely restricts the available number of monotonic reconstruction stencils. Inevitably, this simple pitfall will generate oscillations in perhaps the most crucial region of any flow domain – the boundary.

A simple compromise

A dimensionally split ENO scheme will attain uniform accuracy but not arbitrary accuracy. Dimensionally split ENO hopefully requires less CPU time and recovers the time accuracy qualities of the moving-stencil algorithms, and the solution will still be essentially non-oscillatory. The time level for steady-state convergence will be questionable, and the strategy at boundaries is to limit the stencils in a TVD fashion. The results for this algorithm are given in Chapter 4.

CHAPTER

THREE

RECONSTRUCTION ERROR

How do the algorithms of Chapter 2 perform on a simple numerical problem? In this chapter we discuss methods to determine the error in a numerical reconstruction and in a steady-state solution. Through grid refinement we can extract the computational order of accuracy of the scheme.

3.1 Integral Error

Even though a flux evaluation only requires the fluid-dynamic state at a cell face, reconstructions give the solution field over the *entire* domain. Appropriately, the entire analytical reconstruction must be used along with an integral error norm to determine numerical accuracy and error. Suppose that a numerical solution has been evolved, and we wish to know the computational accuracy of the corresponding cell-average solution set. The integral L_p-norm of the error is

$$\|\epsilon(x)\|_p \equiv \left[\int \left|w_{num}(x) - w_{exact}(x)\right|^p dx\right]^{1/p} \tag{3–1}$$

where $w_{num}(x)$ and $w_{exact}(x)$ are the numerical and assumed known exact solutions. One final reconstruction from the numerical solution's cell averages is necessary. If the numerical solution is an n^{th}-order accurate approximation to the exact solution, then the numerical solution must be reconstructed to n^{th} order at the points where the integral is numerically evaluated. For example, if we have a second-order accurate solution, then we need to know the numerical solution field at the points of integration to second-order accuracy, or in equation we need

$$R(x;\bar{w}) = w_{num}(x) + \mathcal{O}(\Delta x^2) \tag{3–2}$$

If we naïvely use 1-point Gauss quadrature and the first-order piecewise constant cell-average reconstruction, then we will obtain a second-order slope upon grid refinement, but only through luck. The cell mid-point is indeed a second-order approximation to the numerical solution at the midpoint, and thus we will get second-order accuracy. If we used *higher-order* numerical integration, the result of applying the above procedure after several grid refinements would show *first*-order accuracy, regardless of the accuracy of the solution's cell averages! I'll state this again differently. The L_p norm of a piecewise-constant reconstruction, namely the cell average, will yield first-order accuracy if the integral is done exactly.

In general, if we have an n^{th}-order numerical solution in the form of cell averages and an m^{th}-order reconstruction from those cells, or specifically

$$\begin{aligned} w_{num}(x) &= w_{exact}(x) + \mathcal{O}(\Delta x^n) \\ R(x;\bar{w}) &= w_{num}(x) + \mathcal{O}(\Delta x^m), \end{aligned} \tag{3–3}$$

then

$$R(x;\bar{w}) - w_{exact}(x) = \mathcal{O}(\Delta x^n, \Delta x^m) \tag{3–4}$$

The solution error is then

$$\left[\int \left|w_{num}(x) - w_{exact}(x)\right|^p dx\right]^{1/p} = \left[\int \left|R(x;\bar{w}) - w_{exact}(x) + \mathcal{O}(\Delta x^n, \Delta x^m)\right|^p dx\right]^{1/p}$$

So n^{th}-order accuracy in the numerical scheme is obtained if the integration is done exactly with $m \geq n$. To finally clarify, suppose we obtained an infinitely accurate numerical solution where

$$w_{num}(x) = w_{exact}(x) \tag{3–5}$$

Then if we reconstruct the numerical solution to m^{th}-order accuracy and evaluate the error integral norm exactly, then the slope of the L_p-norm versus grid refinement will falsely reveal an m^{th}-order accurate solution.

3.2. A Numerical Example - The Heat Equation

Consider the integral form of the heat equation with constant thermal properties

$$\frac{d}{dt}\iint T(x,y)\,dA + \oint \nabla T \cdot \hat{n}\,ds = 0 \tag{3–6}$$

on the domain $I \in [0,1] \times [0,1]$ with the boundary conditions

$$T(x,0) = T(x,1) = T(0,y) = 0$$

and

$$T(1,y) = sin(\pi y).$$

The exact solution under these constraints is

$$T(x,y) = \frac{sinh(\pi x)\,sin(\pi y)}{sinh\pi}. \tag{3–7}$$

We can measure the reconstruction error and numerical accuracy by calculating cell averages on several square grids and then reconstruct an approximation to $T(x,y)$. The L_p-norm of the error in the temperature field is

$$\|\epsilon(x,y)\|_p \equiv \left[\int \left|R(x,y;\bar{T}) - T(x,y)\right|^p dA\right]^{1/p} \tag{3–8}$$

where $R(x,y;\bar{T})$ is a sufficiently accurate reconstruction using either the arbitrarily high-order accurate two-dimensional ENO scheme or the k-exact reconstruction. We compare degree-zero, degree-one, and degree-two reconstructions on several grids by evaluating the L_1-norm in Eqn. (3–8) with 4-point Gaussian quadrature for each cell. In general, the size of the problem domain must not change when refining the grid for accuracy-analysis studies. The slope of the L_p-norm of the error versus Δx on a log-log plot is the accuracy, r_c, of the scheme. The computational accuracy is calculated in a discrete sense according to

$$r_c \equiv \frac{\log(\epsilon_{\Delta x_1}/\epsilon_{\Delta x_2})}{\log(\Delta x_1/\Delta x_2)}. \tag{3–9}$$

Here, $\epsilon_{\Delta x_1}$ designates the norm of the temperature error on a grid with spacing, Δx_1.

The reconstruction errors for the heat equation using the arbitrarily high-order accurate methods are shown in Tables (1-4) and plotted in Fig. (9). To interpret the results, consider the basis functions for the k-exact and two-dimensional ENO reconstruction operators. The ENO polynomial includes the first $(k+1)^2$ set of basis polynomials, whereas the k-exact polynomial includes only enough to be k-exact. The additional terms in the ENO polynomial and a search for the smoothest data set account for the lower reconstruction error with the ENO method seen in Fig. (9).

The dimensionally split ENO scheme only gives the pointwise value of the temperature at the cell faces. We use the semi-norm which is

$$\|\epsilon(x,y)\|_p^s \equiv \left[\frac{1}{N}\sum_{j,k}\left|R(x_{j+\frac{1}{2}},y_{k+\frac{1}{2}};\bar{T}) - T(x_{j+\frac{1}{2}},y_{k+\frac{1}{2}})\right|^p\right]^{1/p}, \qquad (3-10)$$

where N is the number of interior faces. The L_1^s-norm of the error is evaluated for both dimensionally split ENO and the $\phi-\kappa$ formulation with $\kappa=1/3$. Results are given in Tables (5-7) and plotted in Fig. (10) with the semi-norm. Both $k=1$ and $k=2$ reconstructions with the dimensionally split ENO scheme give second-order accuracy in the reconstruction. Interestingly, the $\kappa=1/3$ reconstruction error is lower. In general, the $\phi-\kappa$ reconstruction is not uniformly accurate and the total variation of the solution will decrease in time, whereas the dimensionally split ENO scheme inherently accentuates extrema and the total variation is allowed to increase on the order of the truncation error.

Operation counts for the high-order methods are given in Table 8 and Fig. (11). The k-exact and two-dimensional ENO methods require operations per grid cell of the same order. The benefit of dimensionally split ENO for this scalar reconstruction exercise is seen by a 10:1 decrease in the required number of operations. It should be noted that the k-exact reconstruction requires a pre-processing step to generate the matrix of integrated basis functions. For a hyperbolic system the entries in this matrix change in an adaptive fashion.

Table 1. Two-dimensional ENO degree-one reconstruction error of the heat equation.

Δx	L_1	r_c	L_2	r_c
0.25000	$2.0312 \cdot 10^{-2}$		$3.4914 \cdot 10^{-2}$	
0.11111	$3.7970 \cdot 10^{-3}$	2.0680	$7.0296 \cdot 10^{-3}$	1.9764
0.05263	$8.3174 \cdot 10^{-4}$	2.0477	$1.5937 \cdot 10^{-3}$	1.9862
0.03448	$3.5418 \cdot 10^{-4}$	2.0189	$6.8635 \cdot 10^{-4}$	1.9922
0.02564	$1.9506 \cdot 10^{-4}$	2.0133	$3.8013 \cdot 10^{-4}$	1.9944

Table 2. Two-dimensional ENO degree-two reconstruction error of the heat equation.

Δx	L_1	r_c	L_2	r_c
0.25000	$7.1659 \cdot 10^{-3}$		$1.0760 \cdot 10^{-2}$	
0.11111	$7.8343 \cdot 10^{-4}$	2.7295	$1.2624 \cdot 10^{-3}$	2.6424
0.05263	$8.6513 \cdot 10^{-5}$	2.9488	$1.3941 \cdot 10^{-4}$	2.9488
0.03448	$2.4647 \cdot 10^{-5}$	2.9694	$3.9617 \cdot 10^{-5}$	2.9754
0.02564	$1.0196 \cdot 10^{-5}$	2.9792	$1.6374 \cdot 10^{-5}$	2.9823

Table 3. K-Exact degree-one reconstruction error of the heat equation.

Δx	L_1	r_c
0.25000	$2.2458 \cdot 10^{-2}$	
0.11111	$4.3807 \cdot 10^{-3}$	2.0155
0.05263	$9.7962 \cdot 10^{-4}$	2.0045
0.03448	$4.2307 \cdot 10^{-4}$	1.9856
0.02564	$2.3229 \cdot 10^{-4}$	2.0236

Table 4. K-Exact degree-two reconstruction error of the heat equation.

Δx	L_1	r_c
0.25000	$1.5814 \cdot 10^{-2}$	
0.11111	$1.0769 \cdot 10^{-3}$	3.3132
0.05263	$1.0160 \cdot 10^{-4}$	3.1596
0.03448	$2.7332 \cdot 10^{-5}$	3.1049
0.02564	$1.0253 \cdot 10^{-5}$	3.3095

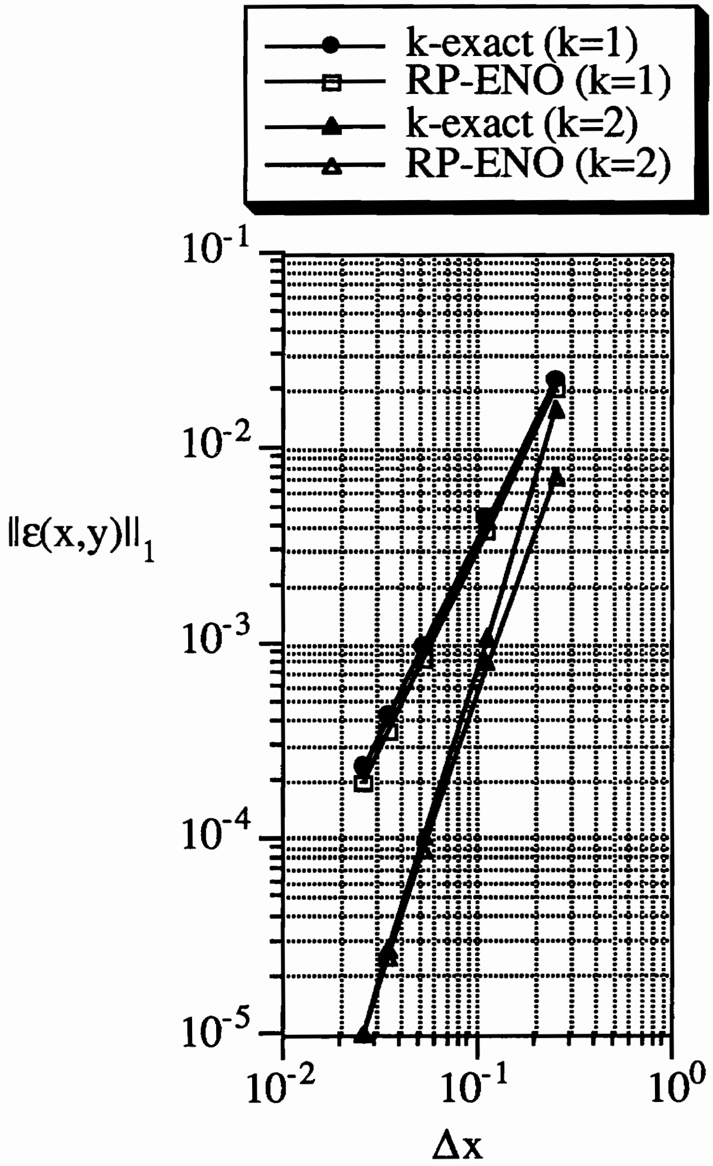

Figure 9. Two-dimensional ENO and k-exact reconstruction error on the heat equation. "RP-ENO" represents the full two-dimensional ENO reconstruction.

Table 5. Dimensionally split ENO degree-one reconstruction error of the heat equation.

Δx	L_1^s	r_c	L_2^s	r_c
0.25000	$2.6748 \cdot 10^{-2}$		$4.1140 \cdot 10^{-2}$	
0.11111	$6.5640 \cdot 10^{-3}$	1.7324	$1.0028 \cdot 10^{-2}$	1.7407
0.05263	$1.5295 \cdot 10^{-3}$	1.9495	$2.3647 \cdot 10^{-3}$	1.9335
0.03448	$6.6467 \cdot 10^{-4}$	1.9709	$1.0356 \cdot 10^{-3}$	1.8526
0.02564	$3.6980 \cdot 10^{-4}$	1.9790	$5.7878 \cdot 10^{-4}$	1.9634

Table 6. Dimensionally split ENO degree-two reconstruction error of the heat equation.

Δx	L_1^s	r_c	L_2^s	r_c
0.25000	$8.9873 \cdot 10^{-3}$		$1.3247 \cdot 10^{-2}$	
0.11111	$1.6972 \cdot 10^{-3}$	2.0555	$2.5310 \cdot 10^{-3}$	2.0411
0.05263	$3.0701 \cdot 10^{-4}$	2.2883	$4.5852 \cdot 10^{-4}$	2.2863
0.03448	$1.1946 \cdot 10^{-4}$	2.2322	$1.7807 \cdot 10^{-4}$	2.2367
0.02564	$6.2390 \cdot 10^{-5}$	2.1922	$9.2940 \cdot 10^{-5}$	2.1947

Table 7. MUSCL-reconstruction error of the heat equation with $\kappa = 1/3$.

Δx	L_1^s	r_c	L_2^s	r_c
0.25000	$5.4120 \cdot 10^{-3}$		$8.4762 \cdot 10^{-3}$	
0.11111	$7.6237 \cdot 10^{-4}$	2.4169	$1.2753 \cdot 10^{-3}$	2.3357
0.05263	$1.8348 \cdot 10^{-4}$	1.9062	$2.7966 \cdot 10^{-4}$	2.0517
0.03448	$8.2136 \cdot 10^{-5}$	1.9007	$1.2369 \cdot 10^{-4}$	1.9292
0.02564	$4.6493 \cdot 10^{-5}$	1.9208	$6.9817 \cdot 10^{-5}$	1.9303

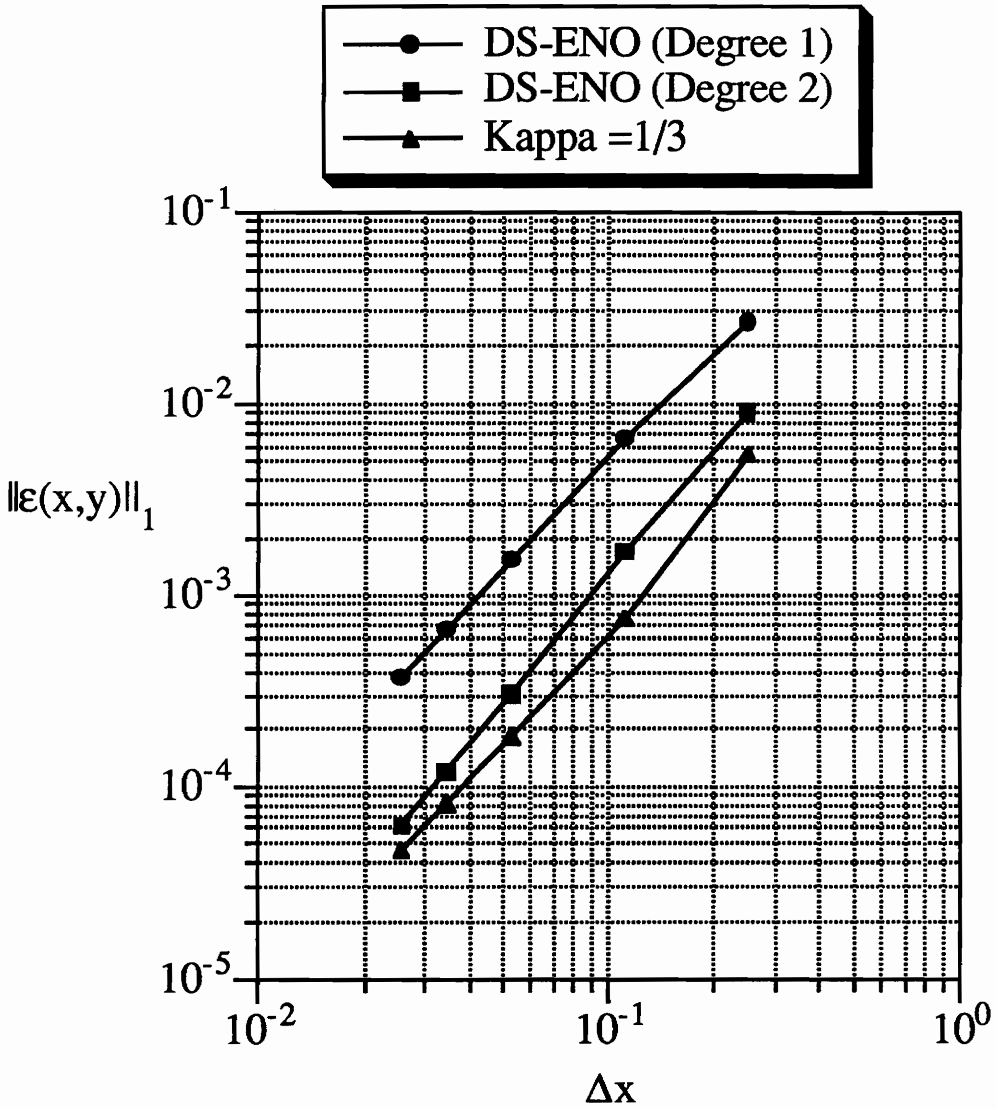

Figure 10. Reconstruction error for dimensionally split ENO and MUSCL differencing ($\kappa = 1/3$) applied to the heat equation. "DS-ENO" represents the dimensionally split ENO scheme.

Table 8. Operations required per cell for interpolation. N is the number of Gauss points and n is the order of the interpolation. For an n^{th}-order reconstruction, the polynomial is degree k where $k = n - 1$ and $n = k + 1$.

Scheme	Operations/Grid Cell
Two-dimensional ENO	$(n-1)(n)(2n-1)^2/6+$ $2[N+1](n-1)(n)(2n-1)/3+$ $[4N](n^2-1)$
Two-dimensional K-exact	$(n)(n-1)(2n^3+n^2+11n)/4$
Dimensionally split ENO	$(n-1)(n)(2n-1)/3+2(n^2-1)$
MUSCL	9

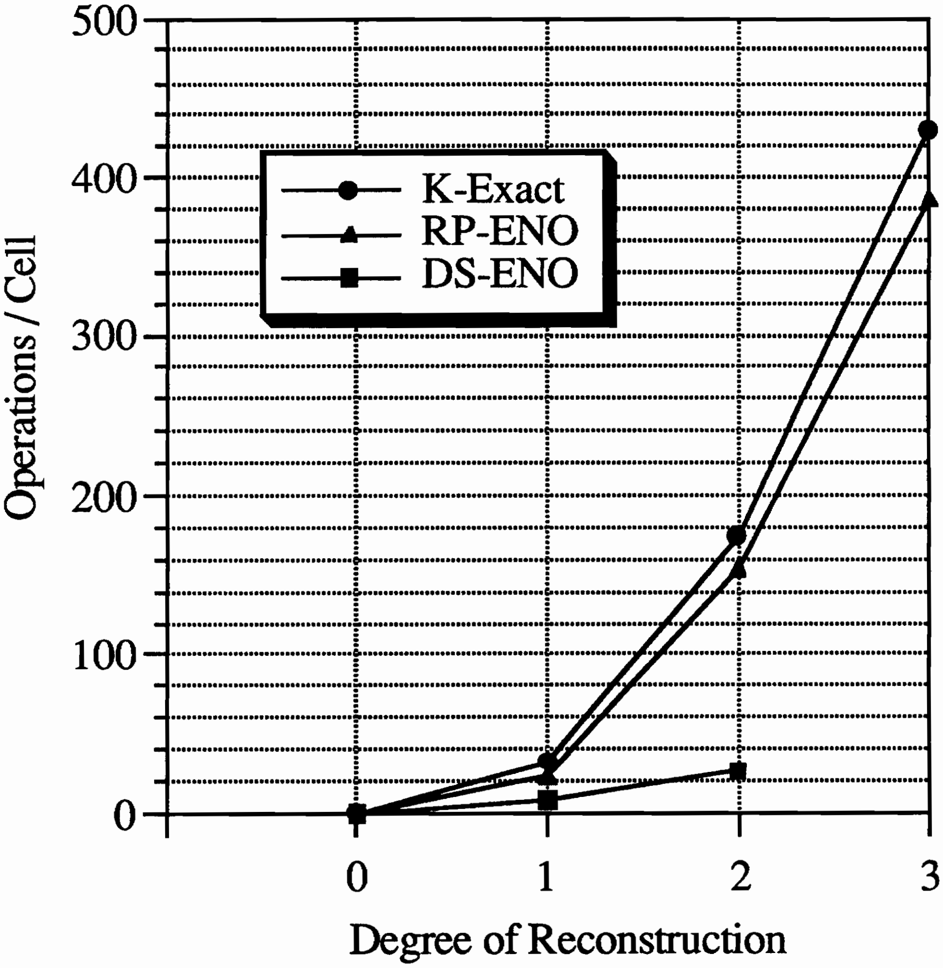

Figure 11. Algorithm-dependent operations required per cell for increasing accuracy. "RP-ENO" and "DS-ENO" represent the full two-dimensional reconstruction and the dimensionally split ENO scheme.

CHAPTER FOUR
NUMERICAL RESULTS

As a test for the dimensionally split and k-exact reconstruction algorithms, we provide several numerical test cases. An exact compressible solution in the absence of shocks is necessary to measure the steady-state spatial accuracy of the schemes. Typically, researchers choose a model equation which is periodic in nature and has a known wave speed. The solution is traversed through several periods and compared with the exact solution at some later time. We address the numerical error problem from a steady-state point of view. Assuming a solution can be converged to machine zero, a reconstruction of that steady-state solution can be compared with the exact solution to determine the numerical error. For this purpose, we select the Ringleb flow documented by Chiocchia [11] and used for a similar purpose by Barth and Frederickson [3].

Next, the oscillatory problems encountered when discontinuities and boundaries clash are shown. A test-case unsteady flow demonstrates the beautiful definition attainable by using high-order accurate numerics for time-dependent flows.

4.1 Ringleb's Flow

Ringleb's flow is a transonic solution to the two-dimensional potential equation solved in the hodograph plane and transformed into the physical plane. If we non-dimensionalize the appropriate flow quantities by stagnation values of the speed of sound and density, we can define

$$\begin{aligned} \bar{\rho} &\equiv \rho/\rho_0 \\ \bar{a} &\equiv a/a_0 \, . \qquad (4\text{--}1) \\ \bar{q} &\equiv q/a_0 \end{aligned}$$

The streamlines of Ringleb's hodograph solution in the physical plane are given by

$$
\begin{aligned}
x &= \frac{1}{2\bar{\rho}}\left(\frac{1}{\bar{q}^2} - \frac{2}{k^2}\right) + \frac{J}{2} \\
y &= \frac{\pm 1}{k\bar{\rho}\bar{q}}\sqrt{1 - \left(\frac{\bar{q}}{k}\right)^2}
\end{aligned}
\tag{4--2}
$$

where

$$
\begin{aligned}
k &= 1/\psi = \textit{constant on a streamline} \\
J &= \frac{1}{\bar{a}} + \frac{1}{3\bar{a}^3} + \frac{1}{5\bar{a}^5} - \ln\sqrt{\frac{1+\bar{a}}{1-\bar{a}}} \\
\bar{q} &= \sqrt{\frac{2}{\gamma - 1}\left(1 - \bar{a}^2\right)} \\
\bar{\rho} &= \bar{a}^{\left(\frac{2}{\gamma-1}\right)}.
\end{aligned}
$$

Equivalently, constant-property contours form circles given by

$$
\left(x - \frac{J}{2}\right)^2 + y^2 = \frac{1}{4\bar{\rho}^2\bar{q}^4}. \tag{4--3}
$$

Selecting values of $\bar{a}$ and streamline constant, k, determines the corresponding fluid state's position in the physical plane. The density field described by the above equations is shown in Fig. (12). The center of a contour circle is located at $(J/2, 0)$ and has radius $1/(2\bar{\rho}\bar{q}^2)$.

Numerically, the Ringleb solution was obtained using k-exact and dimensionally split ENO methods for degree-zero, degree-one, and degree-two reconstructions. Three grids of dimension 11×11, 21×21, and 41×41 were used with a streamline constant, k, which varies from 1.47 on the left wall to 0.8 on the right wall. Fig. (13) shows the 41×41 mesh. The Mach number for the symmetry line is shown in Fig. (14). Fig. (15) shows degree-two pressure contours on the finest mesh with dimensionally split ENO. Flow moves from the left through the symmetric outflow on the bottom. Our domain is chosen so that the right wall is always subsonic while the left wall is transonic. The k-exact solution looks identical. Note that the

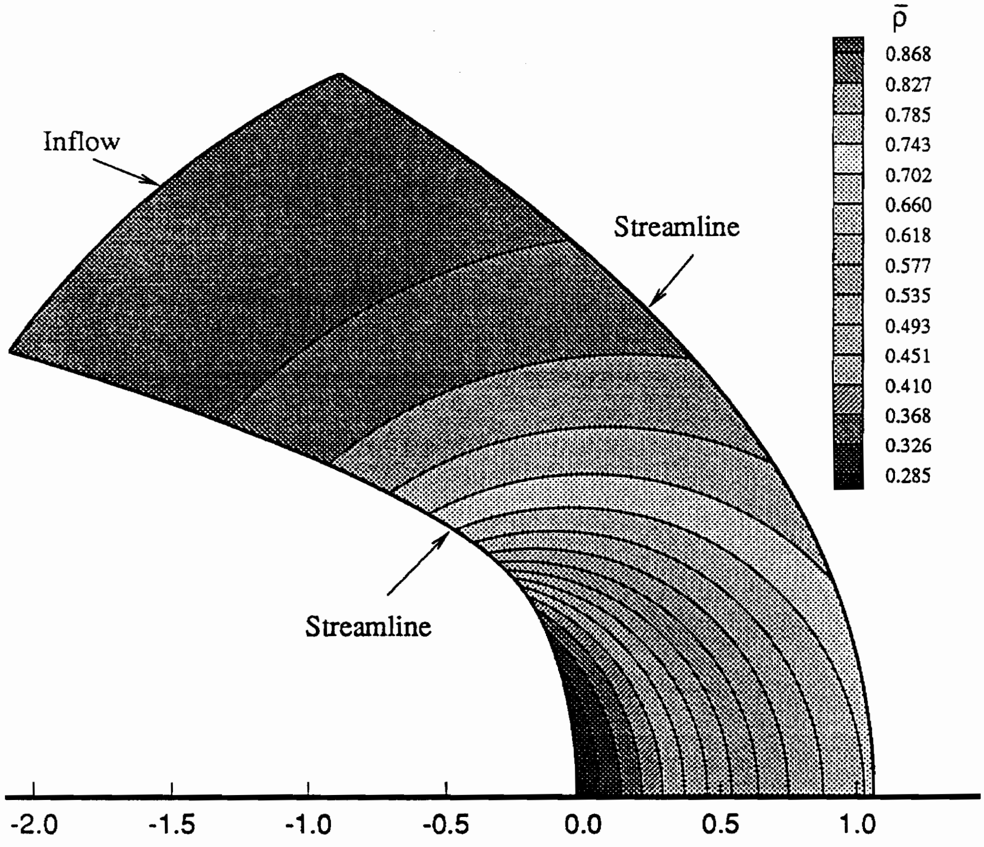

Figure 12. Exact Ringleb solution with mixed subsonic and supersonic outflow.

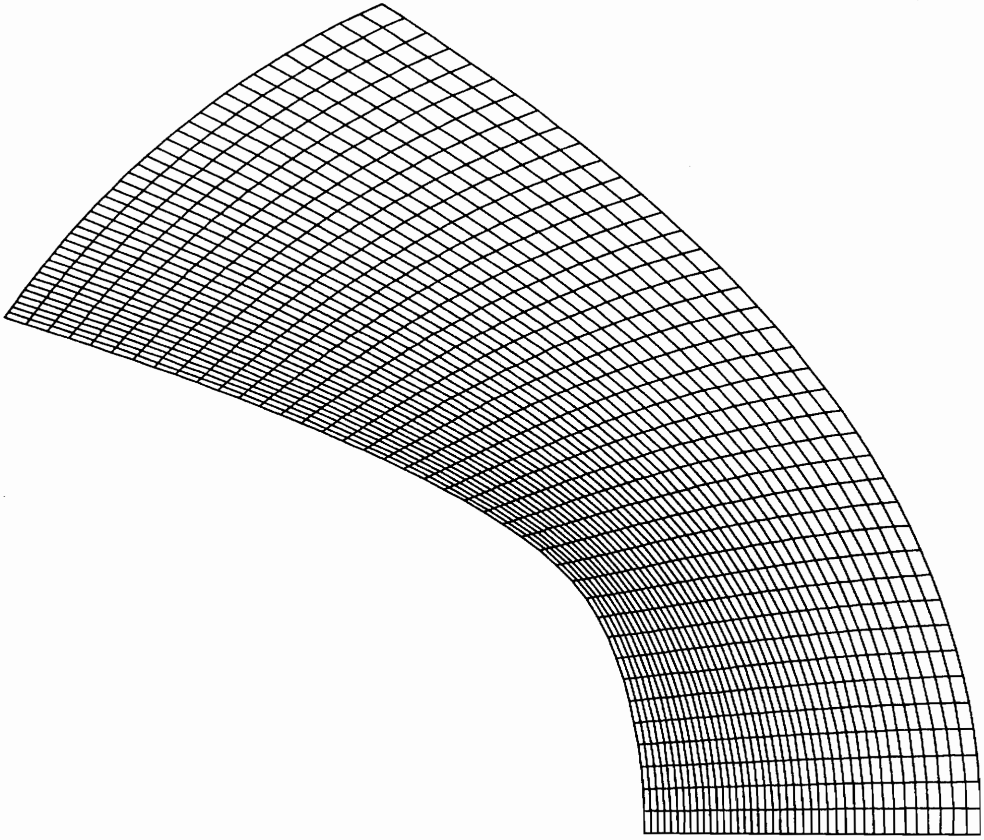

Figure 13. Finest mesh of dimension 41 × 41 used for Ringleb's problem.

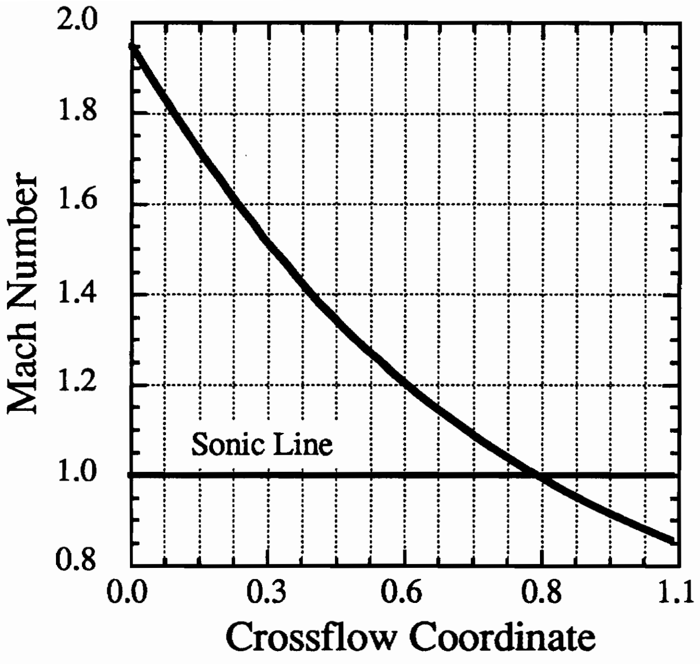

Figure 14. Exact outflow-boundary Mach number for Ringleb's problem.

plotting package averages the boundary values with the first interior cells, which causes "kinks" in the contours on the right wall.

Roe's approximate Riemann solver was used with the exact left and right states necessarily set on the inflow and outflow boundaries, respectively. The grid is generated from the equations for the streamlines, Eqn. (4–2). The tangency boundary conditions on the left and right walls satisfy the normal-momentum equation

$$\frac{\partial p}{\partial n} = \frac{\rho(\vec{V}\cdot\vec{V})}{R}, \tag{4–4}$$

where $\partial p/\partial n$ is the pressure gradient on the wall, ρ is the density, $\vec{V}$ is the velocity vector, and R is the local radius of curvature. Note that R is negative for the right wall and positive for the left wall. The pressure gradient was evaluated to the accuracy of the interior-point scheme.

The L_1 norm of the density error is shown in Fig. (16) for the dimensionally split scheme and in Fig. (17) for the k-exact scheme. As is expected the dimensionally split ENO scheme will achieve no better than second-order accuracy; however, the numerical-solution error for $k = 2$ is less than the error for $k = 1$. The k-exact scheme attains computational accuracies of 0.986, 2.15, and 2.92 for degree-zero, degree-one, and degree-two reconstructions. Pressure along the left and right wall are shown along with the exact solution in Figs. (18) and (19) for the dimensionally split scheme.

4.2 Shock Reflection

A simple Cartesian grid is used to simulate a $M_\infty = 2.9$ oblique shock with a wave angle $\beta = 29°$ reflecting off of a flat surface. A grid with 61×31 mesh points was used with constant spacing in both directions. A sweeping line Gauss-Seidel algorithm is used because the stencil-selection algorithm is elliptically defined.

Most ENO schemes assume that adequately smooth-data is available in some unknown direction. The logic in the scheme itself chooses the smoothest direction

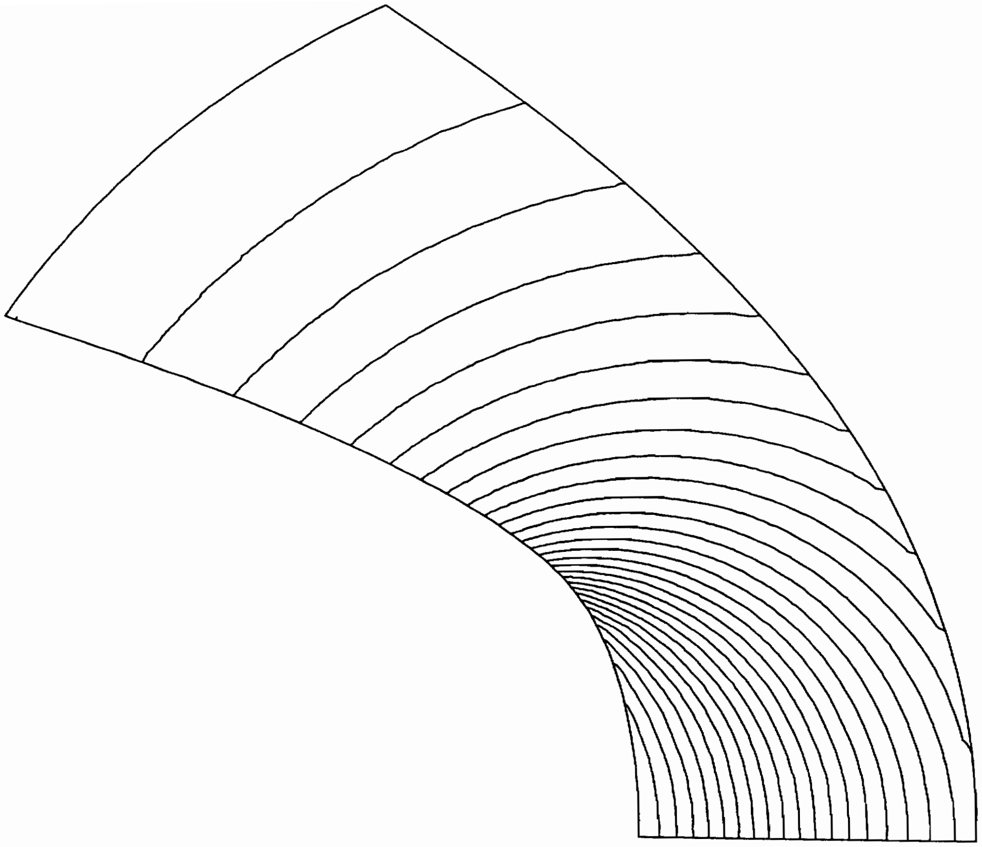

Figure 15. Degree-two dimensionally split ENO pressure contours for Ringleb's problem on the finest (41 × 41) mesh.

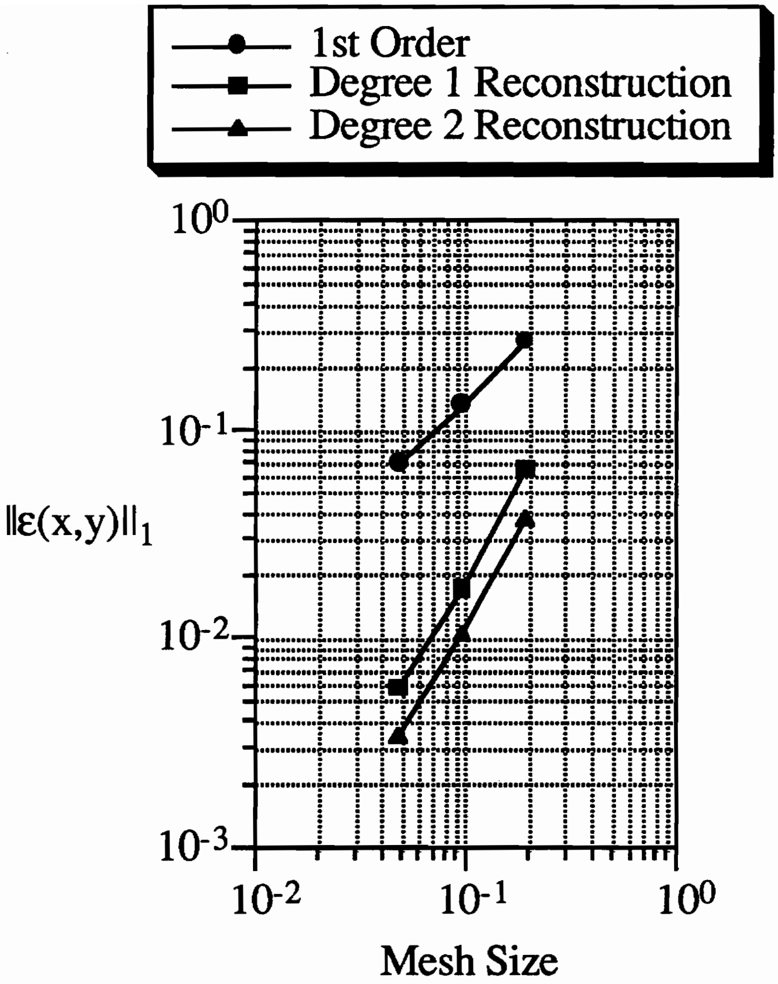

Figure 16. Error in density cell averages for Ringleb's problem with dimensionally split ENO.

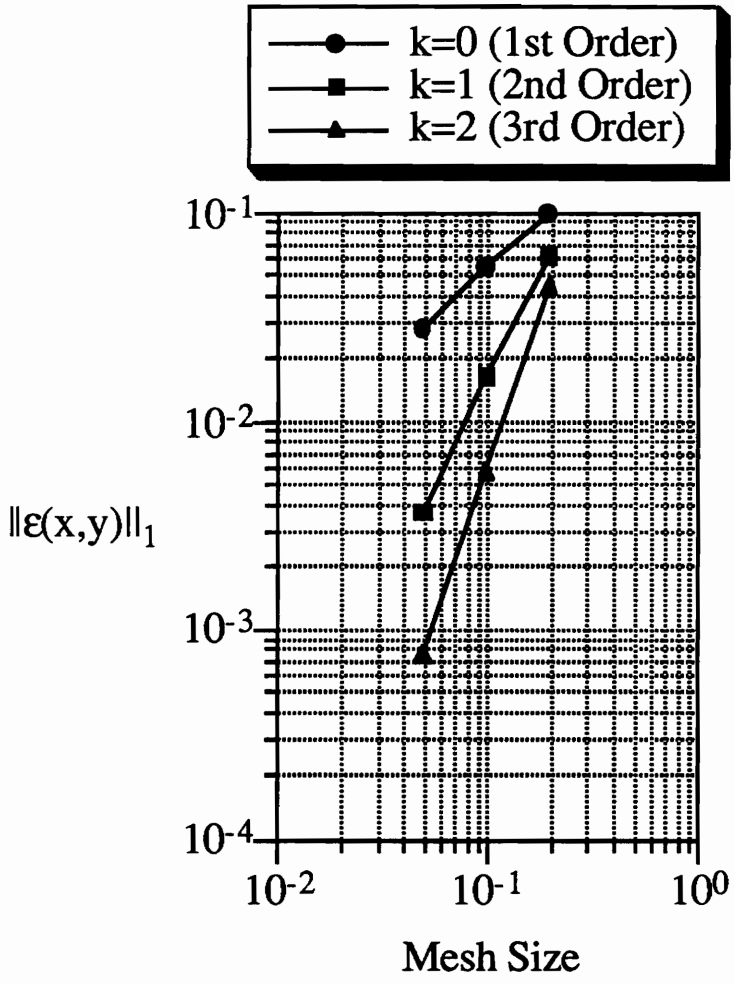

Figure 17. Error in density cell averages for Ringleb's problem with k-exact reconstruction.

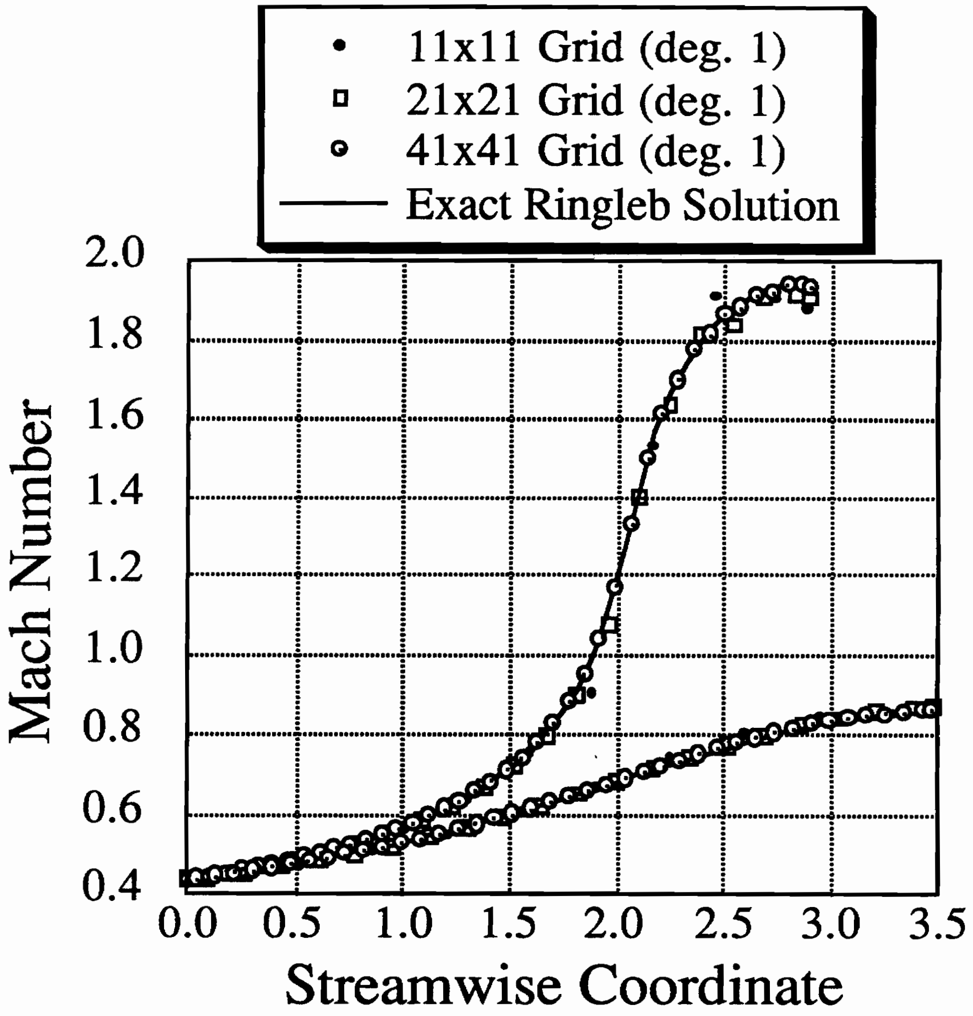

Figure 18. Second-order Mach-number distribution on channel walls for Ringleb's problem compared with the exact solution.

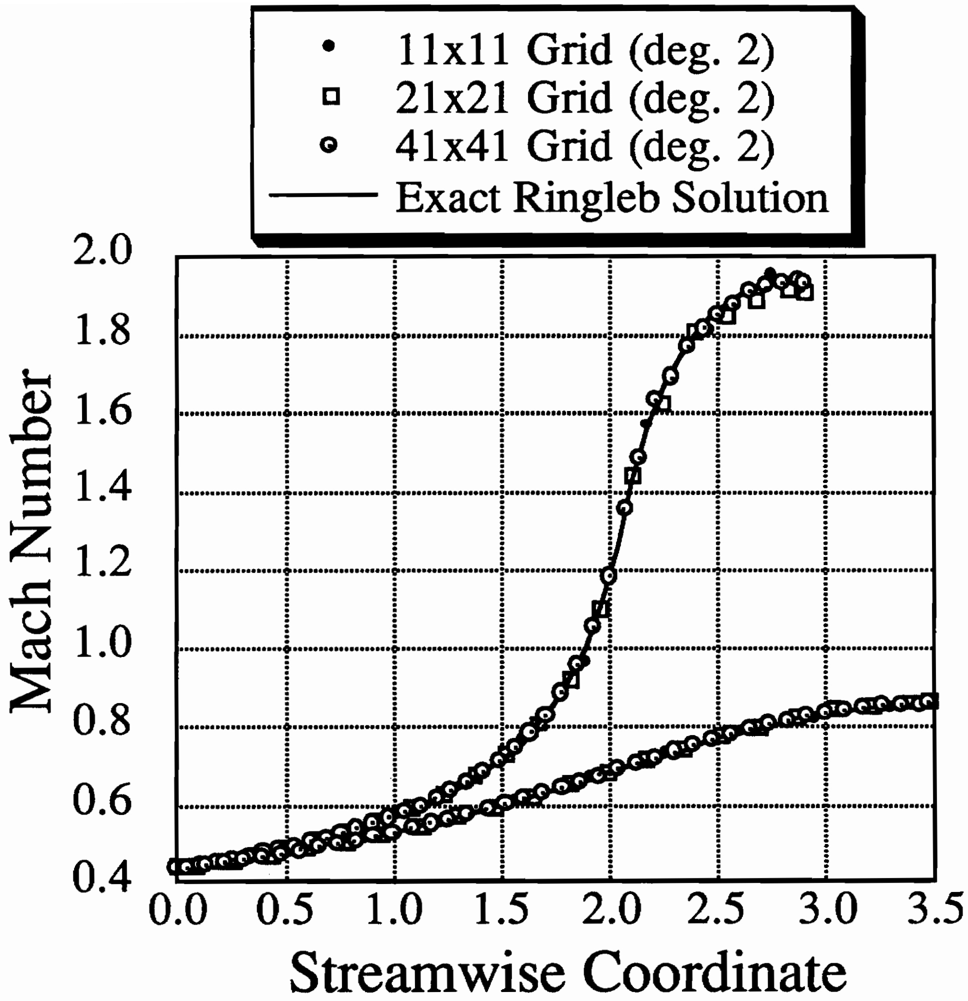

Figure 19. Third-order Mach-number distribution on channel walls for Ringleb's problem compared with the exact solution.

to build the reconstruction. Obviously, when discontinuities are close to boundaries, the underlying assumption of a smooth data-set breaks down. For the shock reflection, all boundary interpolations are forced to the interior, and the stencils are limited according to the magnitude of the next-highest divided difference. For example, if the third divided difference exceeds the second, the stencil accuracy is truncated at the second difference. In the shock-shock region on the wall surface, the numerical oscillations are expected and controlled with a reduction of accuracy at the cells where the norm of the divided-difference table increases. The table is constructed using characteristic-variables. Fig. (20) shows the centerline pressure with and without reduction in accuracy using the dimensionally split ENO scheme with $k=2$.

4.3 Vortex-Shock Interaction

The propagation of a wake vortex through a normal shock is investigated and compared to the experimental results of Naumann and Hermanns [45] and the numerical results of Meadows *et.al.* [43] and Casper [8]. The flow depicts two main fluid phenomena: a secondary sound wave that propagates from the initial interaction position and a spreading angle of this sound wave. The strengths of the secondary wave and the spreading angle depend upon the strengths of the shock and vortex, respectively. The interferogram at $58\,\mu s$ after the vortex strikes the shock is shown in Fig. (21).

To numerically simulate this interaction, the steady-state expansion through a $1/2^{\circ}$ diverging section is calculated with a standing shock at the mid-section. The inflow Mach number is $M=1.1$. The vortex model of Meadows, *et.al.* [43] is then superimposed onto the velocity field and the time evolution calculated according to Shu's TVD Runge-Kutta scheme [55] with third-order temporal accuracy and the dimensionally split ENO scheme. The 101×121 grid is shown in Fig. (22). Density contours are shown at 12.7, 54.7, 85.3 and 101.1 microseconds after the

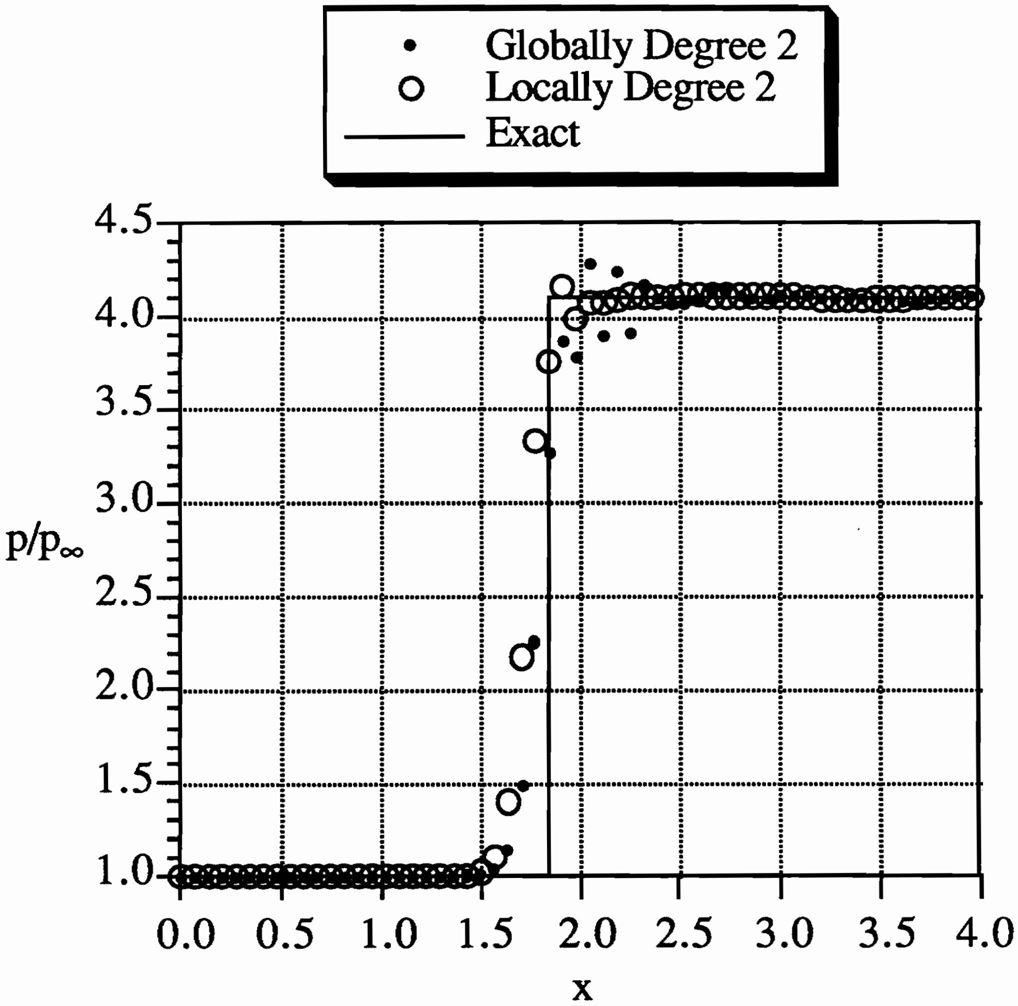

Figure 20. Degree-two dimensionally split ENO shock reflection centerline pressure.

superposition in Figs. (23) and (24). Recall that in the experiment, the time is zeroed when the vortex first strikes the normal shock. A rough estimate can be used to correlate the times by lagging the numerical computation a time of $\Delta t \approx 32\,\mu s$. For example, the interferogram of Fig. (21) occurs at $58\,\mu s$ according to the experiment and corresponds to a numerical time of around $90\,\mu s$. Quantitative results show improvement over the second-order results of Meadows and more dissipation than the arbitrarily high-order accurate results of Casper.

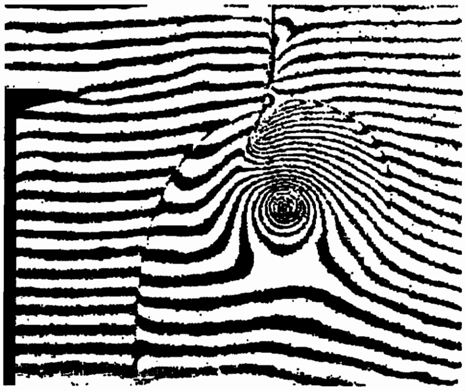

Figure 21. Interferogram of vortex-shock interaction showing the secondary wave and spreading angle 58 μs after the vortex strikes the normal shock.

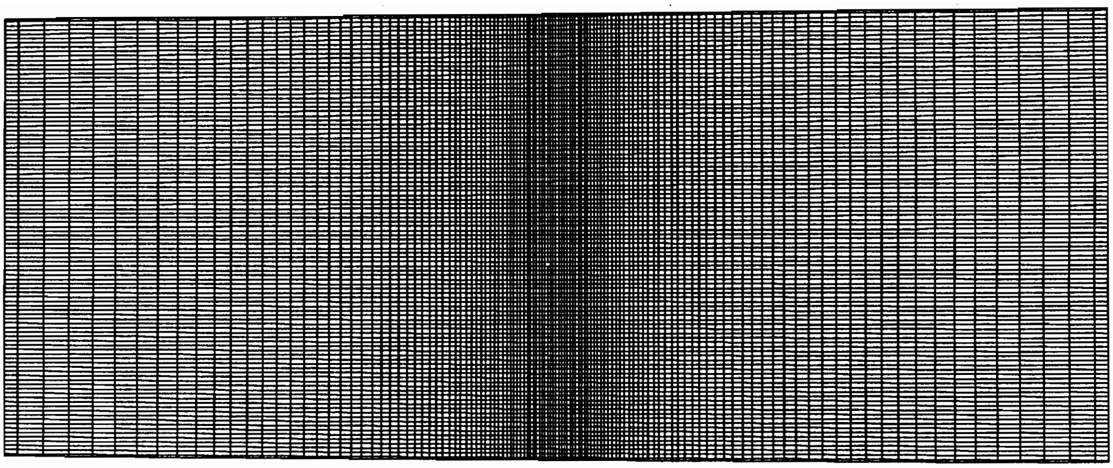

Figure 22. 101 × 121 grid used for vortex-shock interaction.

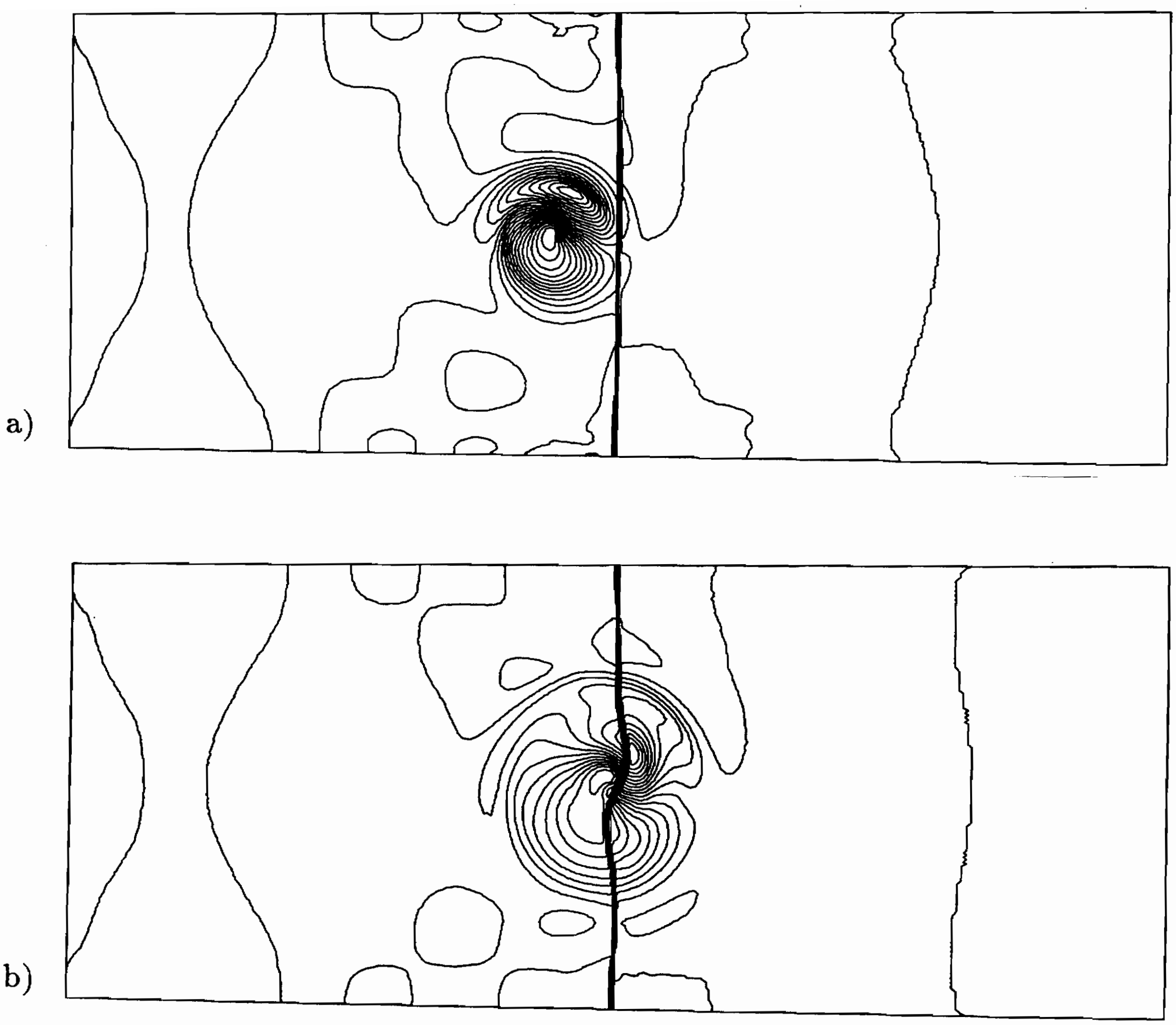

Figure 23. Degree-two dimensionally split ENO density contours for vortex-shock interaction at 12.7(a) and 54.7(b) μs after the ***superposition***.

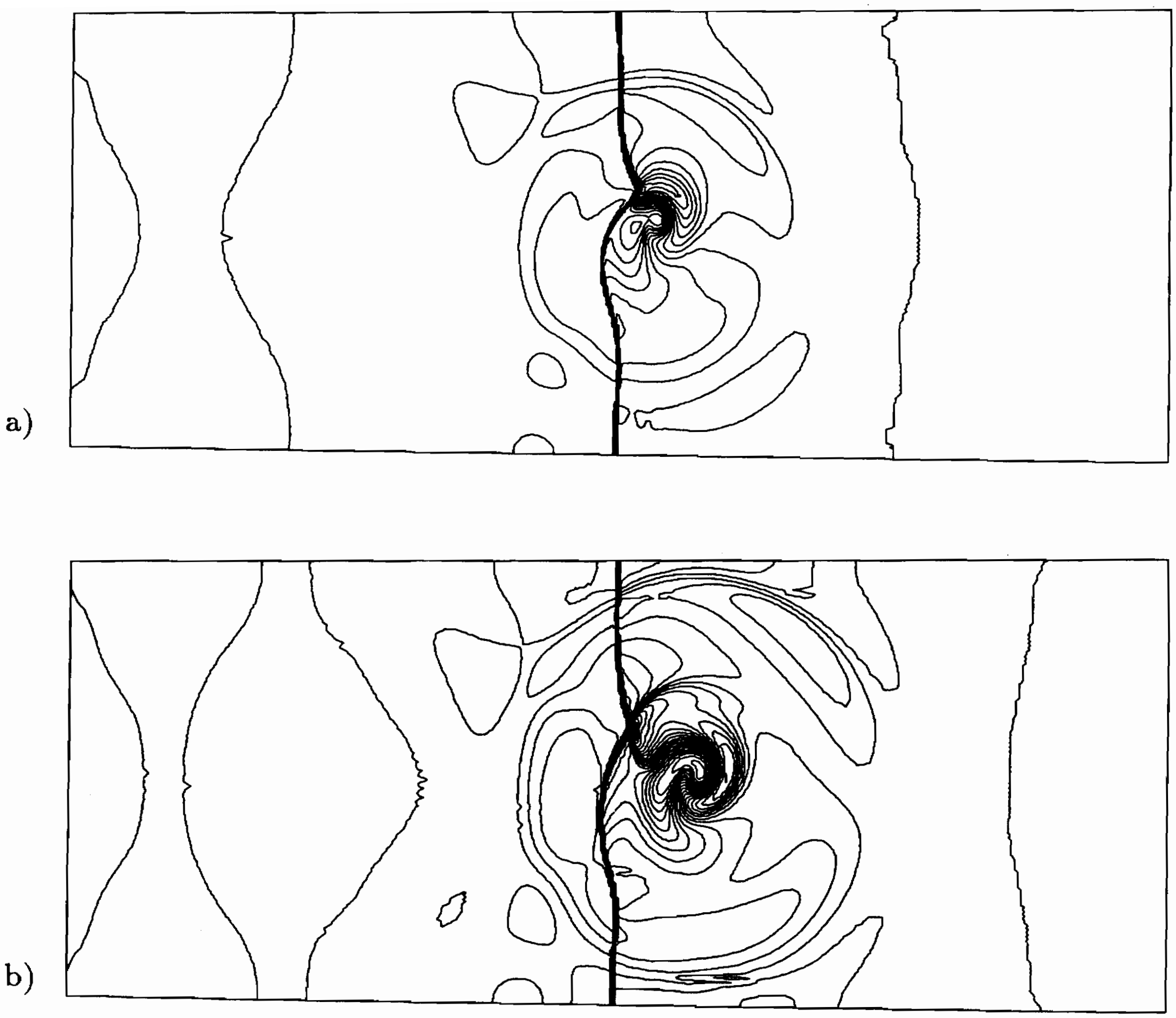

Figure 24. Degree-two dimensionally split ENO density contours for vortex-shock interaction at 85.3(a) and 101.1(b) μs after the ***superposition.***

CHAPTER
FIVE
CONCLUSIONS

Part I of this thesis documents four reconstruction algorithms for implementation in the finite-volume formulation. The methods for truncation-error and reconstruction-error analysis are emphasized. An ENO scheme is described which will achieve, at best, second-order accuracy in two and three dimensions, but is an order of magnitude cheaper than the arbitrarily high-order accurate two-dimensional ENO and k-exact schemes that we implemented. Results for the Ringleb flow verify the computational accuracy. The issues of efficiency and operations required per iteration reflect the severe computational intensity of these multi-dimensional reconstruction operators. A dimensionally split ENO scheme is discussed that controls the interpolation and the subsequent oscillations in shock-boundary regions. Results for a shock-reflection flow show the ability of the stencil-choice algorithm to avoid interpolation across a shock which impinges onto a boundary surface.

The numerics of high-order schemes are explained from an engineering viewpoint. The correct methods for analyzing high-order results are documented. Reconstruction error on the heat equation demonstrates the truncation error realized by each scheme. Finally, a vortex-shock interaction shows the ability of the dimensionally split ENO scheme to excel in capturing high-frequency physical phenomenon in a time accurate setting. Steady-state convergence will inevitably be difficult when using a chattering stencil algorithm. Practical application of the k-exact scheme will depend upon the ability to quickly solve small systems of equations for the polynomial coefficients. All ENO schemes and adaptive-stencil k-exact schemes will inevitably produce successful results for time-dependent flows.

PART

TWO

PRECONDITIONING WITH FINITE-RATE CHEMISTRY

CHAPTER
ONE
INTRODUCTION

The conservation equations for moderate and high Mach-number flows are well coupled, and standard numerical techniques perform adequately. However, in regions of low Mach-number flow, the energy and momentum equations decouple and the continuity equation is poorly posed since the density is nearly constant for all time. The fluid dynamics become stiff. The ***condition number*** quantifies the degree of stiffness and is the ratio of the largest to the smallest characteristic speed. When the smallest wave speed in a hyperbolic system approaches zero (*i.e.* at slow, essentially incompressible flow and at transonic flow speeds), the condition number becomes prohibitively large. Characteristic time stepping or preconditioning eradicates this difficulty and allows the simulation of incompressible flows with compressible flow algorithms.

Van Leer, *et.al.* [64] recently developed an optimal, analytic preconditioning technique to reduce eigenvalue stiffness over the full Mach-number range. They proposed the method for first-order, perfect-gas, inviscid flow simulations using explicit time integration and showed two-dimensional numerical results. Previously, preconditioning methods were proposed by, among others, Chorin [14], Turkel [61], Choi and Merkle [13], and Viviand [77]. While effective in their scope, preconditioning methods before that of Van Leer lacked both the physical connection with the fluid dynamics and the necessary robustness for all Mach-number flows.

Van Leer adopted an approach based upon wave propagation. The Euler equations in two dimensions exhibit a four wave structure consisting of entropy, vorticity, and acoustic waves. At supersonic speeds, these waves travel in predictable

Mach-number-dependent directions. With an understanding of these propagation directions, a preconditioning matrix can be determined that normalizes each wave's speed [64]. However, at subsonic speeds the acoustic waves travel omni-directionally, and in [64] Van Leer uses the structure of the supersonic preconditioning matrix to obtain the optimal subsonic matrix. By multiplying the flux-balance residual with the preconditioning matrix, we can scale the acoustic wave speeds so that all waves propagate at the same rate, an essential property to eliminate inherent eigenvalue stiffness.

The numerical elements of flows with thermal and chemical non-equilibrium have been developed by among others Grossman and Cinnella [24], Grossman and Walters [25], Vinokur and Liu [76], Glaister [22], and Liou *et.al.* [41]. They developed flux-split algorithms for fluid-dynamic simulations with chemical production and vibrational-energy relaxation processes. Walters *et.al.* [79,80] implemented these algorithms in developing a production-level computational code, the General Aerodynamic Simulation Program (GASP). In GASP, stiffness from competing chemical and fluid-dynamic time scales are effectively neutralized by treating the chemical source terms implicitly.

Part II of this thesis synthesizes the thermo-chemical non-equilibrium flux-splitting of Grossman and Cinnella and the characteristic wave preconditioning of Van Leer into a powerful tool for implicitly solving two and three-dimensional, inviscid flows with generalized finite-rate chemistry. Proof of its effectiveness in a real-gas flow regime is given for a very-high-temperature diverging nozzle. However, the majority of the results are for the incompressible, inviscid, perfect-gas regime where the most dramatic acceleration is attained.

The following chapter discusses three preliminary concepts to preconditioning. The principles of vector and matrix norms and the importance of the condition number are first introduced. Next, the stream-aligned coordinate system is developed from which the preconditioning matrix and flux function are derived. Finally, the Fourier transform of a numerical spatial discretization (the Fourier footprint) is established. In Chapter 3 the basics of wave propagation in one and two dimensions are discussed. From these ideas, the preconditioning matrix that optimally equalizes the characteristic wave speeds is derived. Also discussed in Chapter 3 is the preconditioning matrix for generalized finite-rate chemistry, the explicit and implicit formulations, and the numerical implementation of the modified Roe flux function. Insight is provided through an example of the implicit damping qualities of the one-dimensional Euler equations with and without preconditioning. In addition the penalty of using a standard flux formula with preconditioning is analytically illustrated.

Chapter 4 provides many steady-state numerical results of the preconditioning. An inviscid channel flow is used to determine the convergence-rate performance at all Mach numbers and for several Courant numbers. For the transonic channel-flow case, the effect of higher-order accuracy through MUSCL differencing is determined. Further results include a very-low-speed Eppler 387 airfoil. Space marching cases include a supersonic three-dimensional wedge and an axi-symmetric nozzle with chemical reactions. Chapter 5 describes the pitfalls and resolutions to preconditioning the one-dimensional Navier-Stokes equations. Viscous test cases include the damping of a pressure disturbance and the internal structure of a normal shock. Chapter 6 discusses dual-time stepping with the preconditioning matrix to obtain high-order temporal accuracy within an implicit formulation. Unsteady results for Sod's shock-tube problem and the evolution of the transonic channel flow of Chapter 4 are given. Conclusions and future work for the full Navier-Stokes equations appear in the closing chapter.

The work in this dissertation contributes to previous preconditioning research by providing and documenting numerical results for high-order spatial accuracy and various flux functions over the entire range of Mach numbers. The explicit and implicit formulations are determined and documented for finite-rate chemistry with non-equilibrium thermodynamics. A study is made for preconditioning at very low speeds using a compressible formulation. The preconditioning allows for accurate results in an efficient amount of time with no increase in CPU cost per iteration.

CHAPTER

TWO

PRELUDE TO PRECONDITIONING

This chapter serves as an introduction to the fundamental concepts which precede a discussion of system preconditioning. The mathematical definition of the condition number is introduced so the reader understands the concept of its control. Next, the stream-aligned coordinate system is determined in which much of the derivations in Part II of this thesis reside. Finally, the Fourier transform of a spatial discretization, known as the Fourier footprint is discussed as a subset of the commonly used Von Neumann stability analysis. Von Neumann's approach, which was a classified procedure in Los Alamos during World War II, combines both the temporal operator and the spatial discretization to determine stability. The Fourier footprint concerns only the spatial operator. As an example, the damping behavior of several model discretizations of the scalar wave and heat equations is given. Stability of a first-order ordinary differential equation is used in combination with the Fourier footprint to determine stability and wave damping qualities.

2.1 Norms

Use of vectors and matrices requires a method for determining their relative magnitudes. We would like to know whether one group of numbers is bigger than another group. The mathematical description of vector and matrix magnitude is the role of the norm [4].

2.1.1 Vector Norms

Three axioms which define the norm of a vector, X, are

$$\begin{aligned} &(a)\ \|X\| = 0 \text{ if } X = 0, \text{ otherwise } \|X\| > 0, \\ &(b)\ \|cX\| = |c|\,\|X\|, \text{ where } c \text{ is a scalar}, \\ &(c)\ \|X+Y\| \leq \|X\| + \|Y\|. \end{aligned} \qquad (2\text{--}1)$$

Using these axioms, we can also state that

$$\|X - Y\| \geq |\ \|X\| - \|Y\|\ |. \qquad (2\text{--}2)$$

The first axiom says that if all the elements of a vector are zero, then the norm of that vector is also zero; otherwise, the "size" of the vector is positive. The second axiom is the homogeneity condition and says that scaling a vector scales the norm equally. The third axiom is the triangle inequality and implies that the norm of a vector is no greater than the sum of the other two norms. A family of general vector norms known as p-norms conforms to these axioms. They are

$$\|X\|_p = \left(\sum_i |x_i|^p\right)^{1/p}, \quad p \geq 1. \qquad (2\text{--}3)$$

This definition is used later for induced or subordinate matrix norms. For specific values of p, we can determine the L_1, L_2 and L_∞-norms. They are defined as

$$\begin{aligned} \|X\|_1 &= \sum_i |x_i| \\ \|X\|_2 &= \left(\sum_i |x_i|^2\right)^{1/2} \\ \|X\|_\infty &= \max_i |x_i|. \end{aligned} \qquad (2\text{--}4)$$

The L_2-norm is the length of a vector and is called the Euclidean norm, named after the father of analytic geometry. For *continuous* functions on a closed interval $[a, b]$ the family of p-norms is defined as

$$\|f(x)\|_p = \left(\int_a^b |f(x)|^p\, dx\right),^{1/p} \quad p \geq 1. \qquad (2\text{--}5)$$

This integral norm was used in Part I for determining reconstruction error.

2.1.2 Matrix Norms

Similar to the vector norms, matrix norms also have defining axioms. These axioms are

$$\begin{aligned}
&(a)\ \|A\| = 0 \text{ if } A = 0, \text{ otherwise } \|A\| > 0,\\
&(b)\ \|c\,A\| = |c|\,\|A\|, \text{ where } c \text{ is a scalar,}\\
&(c)\ \|A + B\| \leq \|A\| + \|B\|.\\
&(d)\ \|A \cdot B\| \leq \|A\| \cdot \|B\|.
\end{aligned} \qquad (2\text{--}6)$$

The vector p-norms determine families of matrix norms. Suppose we are given a matrix, A, a vector, X, and their vector and matrix norms. If $\|A\,X\| \leq \|A\|\,\|X\|$, then the vector and matrix norms are said to be consistent. For any vector norm, a consistent matrix norm exists and is called the induced, natural or subordinate norm. By using the general family of vector p-norms, the induced matrix norm is defined as

$$\|A\|_p = \max_{x \neq 0} \frac{\|A\,X\|_p}{\|X\|_p}, \qquad (2\text{--}7)$$

where $p = 1$, 2 or ∞. Simple matrix norms induced by the L_1 and L_∞-norms are the maximum column-sum norm,

$$\|A\|_1 = \max_j \sum_i |a_{ij}|, \qquad (2\text{--}8)$$

and the maximum row-sum norm,

$$\|A\|_\infty = \max_i \sum_j |a_{ij}|. \qquad (2\text{--}9)$$

The spectral matrix norm is induced by the vector L_2-norm and is related to the spectral *radius.* The spectral radius represents the maximum eigenvalue and is the least upper bound of all the induced matrix norms. In mathematical form, the spectral radius is

$$\rho(A) = \max |\lambda_i|. \qquad (2\text{--}10)$$

No matrix norm is smaller than the spectral radius; (*i.e.* for any induced matrix norm, $\rho(A) \leq \|A\|$). The spectral norm is induced by the vector L_2-norm and is

$$\|A\|_2 = \sqrt{\rho(A^TA)}. \tag{2–11}$$

A simple explicit summation similar to the maximum row-sum or column-sum norms is not available for the spectral norm, and, in practice, Eqn. (2 – 11) is difficult to calculate. A simple norm to consider is the Frobenius norm and is defined as

$$\|A\|_F \equiv \left(\sum_{i,j} |a_{ij}|^2 \right)^{1/2} . \tag{2–12}$$

This norm is easy to calculate but is not induced by any vector norm.

An Example

As an example of the four matrix norms, consider the matrix

$$A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} . \tag{2–13}$$

Since the matrix is symmetric, the eigenvalues are real and are $\lambda = \{2, 4\}$. The matrix norms discussed above are

$$\begin{aligned}
||A\|_1 &= \text{max column}-\text{sum} = \max\{|3| + |-1|,\ |-1| + |3|\} = 4 \\
||A\|_\infty &= \text{max row}-\text{sum} = 4 \\
||A\|_2 &= \sqrt{\rho(A^TA)} = \sqrt{\max\{4,\ 16\}} = 4 \\
||A\|_F &= \left(3^2 + (-1)^2 + (-1)^2 + 3^2\right)^{1/2} = 2\sqrt{5}.
\end{aligned} \tag{2–14}$$

2.1.3 The Condition Number

The condition number is defined for a non-singular matrix and for any of the matrix norms above as

$$k(A) \equiv \|A\| \, \|A^{-1}\|. \tag{2–15}$$

Since $\|A\,A^{-1}\| = 1$, axiom (d) for matrix norms implies that the condition number is always greater than unity. In mathematical formula, the axiom implies that

$$1 = \|I\| = \|A\,A^{-1}\| \leq \|A\| \, \|A^{-1}\| = k(A) \tag{2–16}$$

which means that the condition number has unit lower bound. The size of the condition number is a measure of the singularity of A, or, equivalently, a large condition number means that a matrix is nearly singular. By using Eqn. $(2-16)$ and the spectral *radius*, instead of the spectral norm, the condition number is loosely the ratio of the largest to the smallest matrix eigenvalue. The condition number based upon spectral radius is the definition used in this thesis.

Condition Numbers and Solutions to a Linear Systems

Consider the linear system

$$A\,x = b. \tag{2–17}$$

A well-conditioned matrix, A, is one where the solution to the above system is well defined. Specifically, the matrix is well-behaved if the condition number is nearly unity, and, conversely, ill-conditioned if a large disparity exists between the minimum and maximum eigenvalues of A. With an understanding of the condition number, we would like to determine how the solution vector, x, is affected by small perturbations in the forcing vector, b. Denoting small perturbations as $\delta(\cdot)$, the perturbed system is

$$A\,(x + \delta x) = b + \delta b. \tag{2–18}$$

Since $A\,x = b$, the solution to the perturbed system is

$$\delta x = A^{-1}\delta b. \tag{2–19}$$

By axiom (d) we can measure the relative size of each term as

$$\|\delta x\| \leq \|A^{-1}\|\,\|\delta b\|. \tag{2–20}$$

Applying an equivalent operation to Eqn. (2–19) yields

$$\|A\,x\| = \|b\| \leq \|A\|\,\|x\|. \tag{2–21}$$

So combining the two equations in Eqn. (2–20) and Eqn. (2–21), we obtain a fundamental inequality,

$$\frac{\|\delta x\|}{\|x\|} \leq k(A)\frac{\|\delta b\|}{\|b\|}. \tag{2–22}$$

For a nearly singular matrix, A, small perturbations in the forcing vector, b, may cause extreme uncertainty in the solution vector, x. Eqn. (2–22) implies that the solution to a linear set of equations is no more accurate than the initial data. Strictly, the inequality gives an upper bound upon the error in the solution vector induced by an error in the forcing vector.

Alternatively, the condition number helps to analyze the solution of a linear system. If we have an approximate solution called x_a to the linear system in Eqn. (2–17), then the residual is

$$A\,x_a - b = R. \tag{2–23}$$

Hopefully, for small residuals the solution vector is sufficiently accurate. Utilizing Eqn. (2–22) we can determine that

$$\frac{\|x - x_a\|}{\|x\|} \leq k(A)\frac{\|R\|}{\|b\|}. \tag{2–24}$$

A small residual does not necessarily mean an accurate solution. During computations on finite-precision machines, perfectly solved solutions are unobtainable. This inaccuracy is the role of numerical round-off error. The condition number reveals the sincerity of a small residual. For example, let the system in Eqn. (2 − 17) be defined by

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2+\epsilon \end{bmatrix}, \quad b = \begin{Bmatrix} 2 \\ 4+\epsilon \end{Bmatrix}. \tag{2–25}$$

The condition number of A using the maximum row-sum norm is $k(A) = (4+\epsilon)(3+\epsilon)/\epsilon$, and the solution vector is

$$x = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}. \tag{2–26}$$

If the forcing vector, b, has a small perturbation

$$\delta b = \begin{Bmatrix} \epsilon \\ -\epsilon \end{Bmatrix}, \tag{2–27}$$

then the induced solution vector is perturbed by

$$\delta x = \begin{Bmatrix} 3+\epsilon \\ -3 \end{Bmatrix}. \tag{2–28}$$

With $\epsilon = 0.01$, the solution to the perturbed system is then

$$x + \delta x = \begin{Bmatrix} 4.01 \\ -2 \end{Bmatrix}, \tag{2–29}$$

and the corresponding uncertainties in the forcing and solution vectors are

$$\frac{\|\delta b\|_\infty}{\|b\|_\infty} = \frac{0.01}{4.01} \quad \text{and} \quad \frac{\|\delta x\|_\infty}{\|x\|_\infty} = \frac{3.01}{1}.$$

Remarkably, this implies that a 0.25% error in the residual yields a 300% error in the solution for a matrix condition number of $k \approx 1200$. These results for ill-conditioned systems should be recalled later in Chapter 5 during the calculation of high-condition-number flow fields.

2.2 Stream-Aligned Coordinate Systems

The underlying physics of fluid dynamics leap forward from the symmetric non-conservative form of the Euler equations. To obtain this perceptive form, we conduct coordinate transformations to change from Cartesian conservation variables to stream-aligned symmetrization variables. The resulting system quickly reveals the primary wave systems dictating the Euler equations. A secondary motive for the transformation is to exchange the cumbersome inviscid flux Jacobians, $\partial f/\partial Q$ and $\partial g/\partial Q$, for simpler forms. The resulting Jacobians are then much easier to manipulate.

First let's start at the conservation level. The well-known, two-dimensional Euler equations in conservation form are

$$\frac{\partial Q}{\partial t}+\frac{\partial f}{\partial x}+\frac{\partial g}{\partial y}=0 \qquad (2\text{–}30)$$

where

$$Q\equiv\left\{\begin{array}{c}\rho\\ \rho u\\ \rho v\\ \rho e_0\end{array}\right\},f\equiv\left\{\begin{array}{c}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uh_0\end{array}\right\},g\equiv\left\{\begin{array}{c}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vh_0\end{array}\right\}. \qquad (2\text{–}31)$$

For a perfect gas, the equations are closed by the perfect-gas equation of state

$$p=(\gamma-1)\rho e. \qquad (2\text{–}32)$$

In the equations above $\rho, u, v, e_0, p, h_0, \gamma$ are the density, (x, y) components of velocity, total internal energy per unit mass, pressure, total enthalpy per unit mass and the ratio of specific heats, respectively. The total energy and enthalpy are defined as $e_0=e+(u^2+v^2)/2$ and $h_0=e_0+p/\rho$.

Following Van Leer [64], the variables, U, that symmetrize the inviscid flux Jacobians are known in differential form as

$$dU\equiv\left\{\begin{array}{c}dp/(\rho a)\\ du\\ dv\\ dp-a^2d\rho\end{array}\right\}. \qquad (2\text{–}33)$$

A variable transformation of the Euler equations to the symmetrization variables yields

$$\frac{\partial U}{\partial t} + A'\frac{\partial U}{\partial x} + B'\frac{\partial U}{\partial y} = 0 \tag{2–34}$$

where

$$\begin{aligned} A' &\equiv \frac{\partial U}{\partial q}\frac{\partial q}{\partial Q}\frac{\partial f}{\partial Q}\frac{\partial Q}{\partial q}\frac{\partial q}{\partial U}, \\ B' &\equiv \frac{\partial U}{\partial q}\frac{\partial q}{\partial Q}\frac{\partial g}{\partial Q}\frac{\partial Q}{\partial q}\frac{\partial q}{\partial U}. \end{aligned} \tag{2–35}$$

Note that from Eqn. (2–33) we can identify the Jacobian of U with respect to the primitive variables, $q=(\rho,u,v,p)^T$ which is

$$\frac{\partial U}{\partial q} = \begin{bmatrix} 0 & 0 & 0 & 1/(\rho a) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a^2 & 0 & 0 & 1 \end{bmatrix}. \tag{2–36}$$

The Jacobian of the conservative variables with respect to the primitive variables assuming a perfect-gas law are

$$\frac{\partial Q}{\partial q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ u & \rho & 0 & 0 \\ v & 0 & \rho & 0 \\ (u^2+v^2)/2 & \rho u & \rho v & 1/(\gamma-1) \end{bmatrix}. \tag{2–37}$$

The converse Jacobian is simply the inverse of the above matrix, or

$$\frac{\partial q}{\partial Q} = \left(\frac{\partial Q}{\partial q}\right)^{-1}. \tag{2–38}$$

All of these transformation matrices are fairly trivial, and using them, the inviscid flux Jacobians transform into the matrices A' and B' of Eqn. (2–35) as

$$A' = \begin{bmatrix} u & a & 0 & 0 \\ a & u & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{bmatrix} \tag{2–39}$$

and

$$B' = \begin{bmatrix} v & 0 & a & 0 \\ 0 & v & 0 & 0 \\ a & 0 & v & 0 \\ 0 & 0 & 0 & v \end{bmatrix}. \tag{2–40}$$

The state vector's velocities are with respect to a Cartesian coordinate system, so we need to rotate the axes to the stream-aligned coordinate system which is denoted as (ξ, η) with ξ in the direction of the velocity vector. The state vector, $\bar{U}$, designates the symmetrization variables with *stream-aligned* velocities. A transformation from the (x, y) system to the (ξ, η) system through an angle, θ, yields the transformation matrix

$$\frac{\partial \bar{U}}{\partial U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2–41}$$

The angle, θ, is determined from the velocity components as $\tan\theta = v/u$. We also need to differentiate with respect to the stream-aligned coordinates. Applying the chain rule to $\partial U/\partial x$ and $\partial U/\partial y$, the final metamorphosis of the Euler equations is complete. The equations become

$$\frac{\partial \bar{U}}{\partial t} + \bar{A}\frac{\partial \bar{U}}{\partial \xi} + \bar{B}\frac{\partial \bar{U}}{\partial \eta} = 0 \tag{2–42}$$

where

$$\bar{A} = \begin{bmatrix} q & a & 0 & 0 \\ a & q & 0 & 0 \\ 0 & 0 & q & 0 \\ 0 & 0 & 0 & q \end{bmatrix}, \quad \text{and} \quad \bar{B} = \begin{bmatrix} 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2–43}$$

The eigenvalues and underlying wave structure are evident in this simple form, and the flux Jacobians are now trivial. The derivations of the preconditioning matrix and modified flux function start with this stream-aligned system. For numerical implementation, the preconditioning matrix must be transformed back to update the desired state variables which are usually the conserved variables with Cartesian velocities. Later, in this thesis the preconditioning matrix for updating primitive variables is documented. The artificial-viscosity matrix in the modified flux function must be transformed to a cell-face-aligned system with conservation variables for finite-volume implementation.

2.3 Fourier Footprints

In Chapters 3 and 4, analysis of inviscid and one-dimensional viscous flows governed by the Euler and Navier-Stokes equations will utilize the Fourier footprint. In this subsection the basic principles of Fourier-frequency analysis is explained. As a foundation, the scalar wave and heat equation are analyzed as model problems.

For a linearized equation, the Fourier transform of the temporal and spatial operators combine to determine the stability and damping behavior of a specific space-time discretization. This is a Von Neumann stability analysis [33]. A more descriptive approach determines the effects of each operator individually. The Fourier footprint is the Fourier transform of strictly the spatial discretization and provides a visual portrait of the error frequencies in the complex plane. For example, consider the scalar wave equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0. \tag{2–44}$$

With Euler explicit time integration and first-order upwind differencing a Von Neumann stability analysis yields

$$g - 1 = -\lambda(1 - e^{-i\beta_x}) \tag{2–45}$$

where g is the scalar gain or amplification factor, λ is the Courant number ($\lambda = a\Delta t/\Delta x$) and β_x is the Fourier frequency. Of course, the gain must be bounded by unity for absolute stability. The Fourier footprint is the right-hand side of Eqn. (2–45). More rigorously, the footprint is determined by multiplying Eqn. (2–44) by a time step, Δt, which yields

$$\begin{aligned}\mathcal{F}\left\{\Delta t\frac{\partial u}{\partial t}\right\} &= -\lambda\left(1 - e^{-i\beta_x}\right) \\ &= -\lambda\left[2\sin^2(\beta_x/2) + i\sin\beta_x\right].\end{aligned} \tag{2–46}$$

In the complex plane, the above footprint yields a circle with radius, λ, located in the negative real half-plane as shown in Fig. (25). The low frequencies ($\beta_x \rightarrow 0$)

converge onto the origin and the high frequencies ($\beta_x \rightarrow \pi$) meet at $\Re\{\mathcal{F}\} = -2\lambda$ on the negative real axis.

For central differencing the Fourier footprint of the scalar wave equation is

$$\mathcal{F}\left\{\Delta t \frac{\partial u}{\partial t}\right\} = -2\lambda\, i \sin\beta_x \tag{2--47}$$

and is included in Fig. (25). For all frequencies, β_x, the footprint lies along the imaginary axis corresponding to waves with no dissipation (*i.e.* no real part).

The unstable first-order downwind scheme has the footprint

$$\mathcal{F}\left\{\Delta t \frac{\partial u}{\partial t}\right\} = -\lambda\left(-2\sin^2\beta_x/2 + i\sin\beta_x\right). \tag{2--48}$$

This footprint is located in the positive real half-plane and mirrors the first-order upwind Fourier footprint.

Stability restrictions for the footprints follow from the stability of a first-order linear ordinary differential equation with a complex constant, μ,

$$\frac{du}{dt} = \mu\, u. \tag{2--49}$$

With initial data $u(0) = u_0$, the analytical solution has exponential form

$$u(t) = u_o e^{\mu t}. \tag{2--50}$$

Obviously, stability of the ordinary differential equation requires that $\Re\{\mu\} \leq 0$. So, for a perfect temporal operator, a Fourier footprint lying in the negative real half-plane satisfies the stability requirements of the ordinary differential equation. However, explicit numerical operators are more restrictive than the differential equation.

Consider the first-order Euler explicit time discretization

$$\frac{u^{n+1} - u^n}{\Delta t} = \mu\, u^n, \tag{2--51}$$

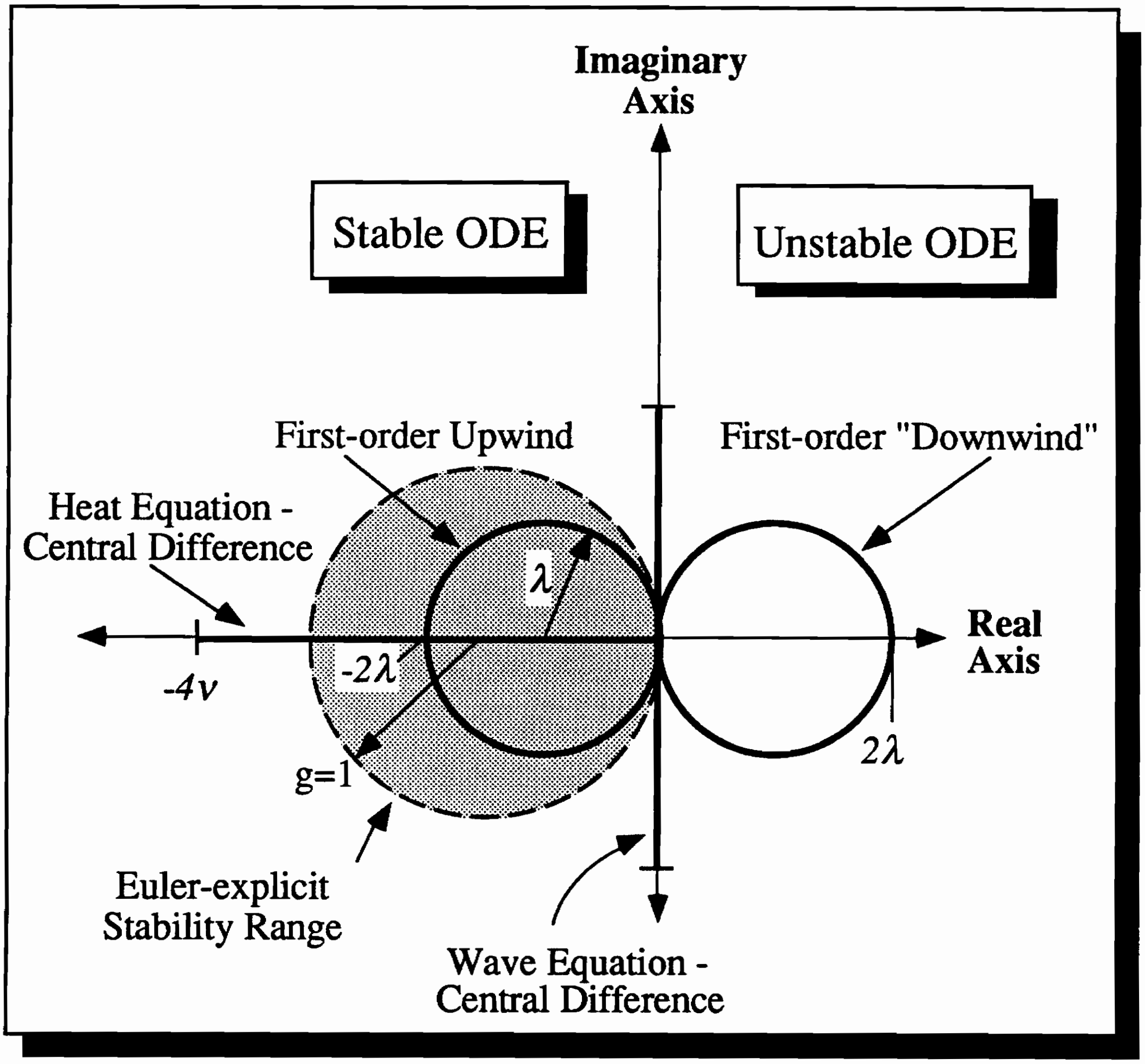

Figure 25. Fourier footprints for the scalar wave equation with first-order upwind, central differencing, and first-order "downwind" and for the scalar heat equation with second-order central differencing.

which reduces to $u^{n+1} = (1+\mu\Delta t)u^n$. The term "$(1+\mu\Delta t)$" represents the amplification factor of the scheme. Introducing $z = \mu\Delta t$, the polynomial of the Euler-explicit temporal operator is

$$P_1(z) = 1 + z. \tag{2–52}$$

The temporal operator, in general, reflects the gain of the error from time level to time level, and for a stable scheme using Euler explicit time integration, $|P_1| \leq 1$. Likewise, all frequencies in the spatial discretization visualized with the Fourier footprint must be engulfed by the unit contour of the temporal operator as seen in Fig. (26). The Courant number scales the footprint to accomplish the task of numerical stability.

More interestingly, individual frequencies in the Fourier footprint can be perfectly damped by strategically matching the temporal and spatial operators. With the first-order upwind discretization, a Courant number of $\lambda = 1/2$ perfectly overlaps the highest-frequency wave ($\beta_x = \pi$) of the footprint with the root of the temporal polynomial, $z = -1$, as seen in Fig. (26). Recall that the highest frequency wave is located on the negative real axis at $z = -2\lambda$. However, since the low frequencies in the Fourier footprint are placed near the origin where the gain is unity, the longest waves are always poorly damped. This is true for all numerical schemes and is not specific to the preconditioned equations. Van Leer *et.al.* [65] design optimally smooth multi-stage schemes for the Euler equations by overlapping the Fourier footprint and the temporal operator. By choosing appropriate stage weights and Courant numbers, they achieve optimal damping of specific waves in the high-frequency spectrum. Immediate applications for multi-grid algorithms are evident.

For viscous flows, the heat equation serves as a model equation for the dissipative behavior inherent to second-order derivatives. The heat equation in differential form with constant thermal properties is

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}. \tag{2–53}$$

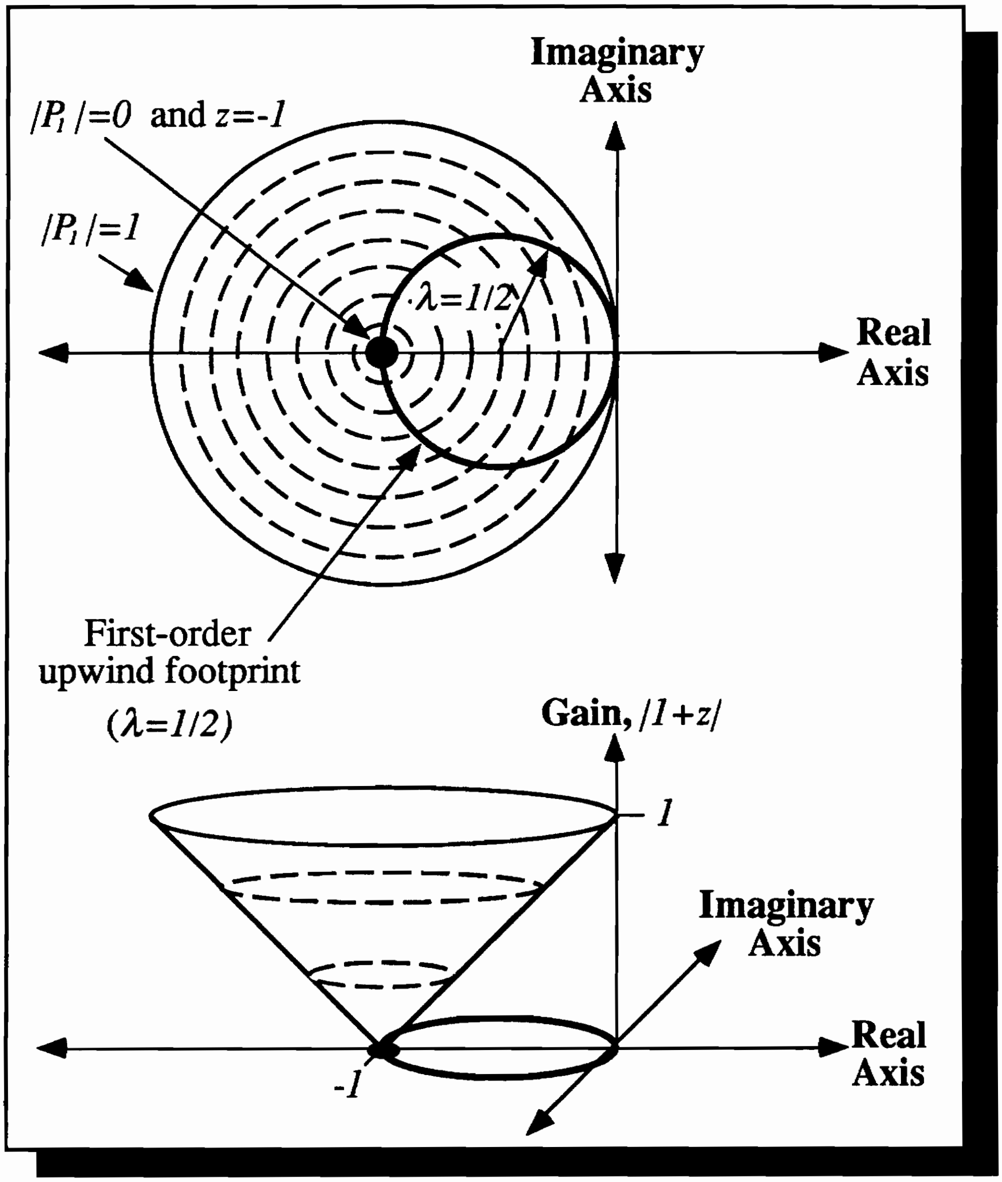

Figure 26. Overlap of the first-order upwind Fourier footprint of the scalar wave equation with the amplification polynomial from Euler explicit time integration. A CFL of $\lambda = 1/2$ yields perfect damping of the highest frequencies.

Second-order central differencing yields the purely dissipative footprint shown previously in Fig. (25). The footprint is

$$\mathcal{F}\left\{\Delta t\frac{\partial u}{\partial t}\right\} = -4\nu\sin^2(\beta_x/2). \tag{2–54}$$

Note that stability using Euler explicit time integration requires $\nu \leq 1/2$. This time step scales the Fourier footprint inside the stability circle of unit radius. We may expect that $\nu = 1/4$ yields the optimal damping of the high-frequency error waves.

CHAPTER
THREE
PRECONDITIONING THEORY

3.1 Simple Wave Analysis

For a general hyperbolic conservation law of the form

$$\frac{\partial Q}{\partial t} + \sum_i \frac{\partial f_i}{\partial x_i} \qquad (3-1)$$

a systematic simple wave analysis follows four steps: determining the non-conservative form, the wave speeds, the plane wave fronts or wave crests, and finally, the wave front emitted by a point disturbance which is the envelope of all the plane wave fronts. The following two subsections describe the underlying simple wave structure of the Euler equations. A knowledge of wave propagation behavior is fundamental to the derivation of the preconditioning matrix which normalizes the characteristic waves' speeds.

3.1.1 One-dimensional Analysis

The Euler equations in Eqn. $(2-30)$ written in one-dimensional form and differentiated with respect to primitive variables, q, are

$$\frac{\partial q}{\partial t} + A\frac{\partial q}{\partial x} = 0, \qquad (3-2)$$

where the flux Jacobian is

$$A = \begin{bmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \rho a^2 & u \end{bmatrix}. \qquad (3-3)$$

For systems of two independent variables (*i.e.* one flux Jacobian), a diagonalized form of the Euler equations is possible. For systems of two or more independent variables, diagonalization is limited to systems with commuting flux Jacobians (*i.e.* $AB = BA$). When the Jacobians commute, they have the same left and right eigenvectors, and consequently both directions have the same characteristic variables. The diagonalized form of the one-dimensional Euler equations is determined by transforming from primitive to characteristic variables by using the left eigenvectors of A. The flux Jacobian can be decomposed as $A = T\Lambda T^{-1}$ where T and T^{-1} are the matrices of the right and left eigenvectors, respectively. The eigenvalues of the Jacobian rest along the diagonals of Λ. The characteristic variables, dw, are the elements of

$$dw = T^{-1} dq, \tag{3-4}$$

and the diagonalized equations are then

$$\frac{\partial w}{\partial t} + \Lambda \frac{\partial w}{\partial x} = 0. \tag{3-5}$$

The inverse operation of Eqn. (3−4) implies that the primitive variables may be considered as a linear combination of simple wave perturbations as

$$\delta q = \sum_{\alpha} r_{\alpha} \delta w^{\alpha} = \sum \delta q^{\alpha}. \tag{3-6}$$

Each perturbation is proportional to a right eigenvector of A, so any perturbation of the primitive variables can be projected onto the eigenvectors of A with the characteristic variables as weights. The spatial variation of a characteristic variable is the wave strength or intensity of a simple wave perturbation. The wave strength is independent of the choice of differentiation variables (*e.g.* conservative, primitive, or symmetrization variables). The speed, λ, with which the simple wave perturbations propagate is called the ***wave speed*** and is also independent of the differentiation variables. However, the eigenvectors translate the effect of the simple wave intensities onto the chosen set of non-conservative variables and, therefore, depend upon the choice. Plane wave fronts propagate normal to the wave speed vectors like the ocean's wave crests crashing onto the shores of the East coast.

One-dimensional Entropy Waves

The diagonalized system in Eqn. (3–5) has three-degrees of freedom consisting of one entropy wave and two acoustic waves. For the entropy wave, the convective eigenvalue, characteristic variable and right eigenvector are the wave speed, strength, and projection respectively. These are

$$\lambda_u = u, \quad \delta w_u = -\frac{\delta p}{a^2} + \delta\rho, \quad r_u = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}. \qquad (3\text{–}7)$$

Since the entropy is proportional to this characteristic variable, these waves are called entropy waves. The wave decomposition says that entropy-like disturbances, δw_u, propagate in the x-direction with a wave speed, λ_u, equal to the speed of the flow. The right eigenvector reveals that entropy disturbances cause variations in the density and do not affect the velocity or the pressure.

One-dimensional Acoustic Waves

The two acoustic waves share symmetry around the velocity vector. The forward-moving acoustic wave is described by

$$\lambda_+ = u + a, \quad \delta w_+ = \frac{1}{2}\left(\delta p + \rho a\, \delta u\right), \quad r_+ = \begin{Bmatrix} 1/a^2 \\ 1/(\rho a) \\ 1 \end{Bmatrix}. \qquad (3\text{–}8)$$

The backward moving wave is

$$\lambda_- = u - a, \quad \delta w_- = \frac{1}{2}\left(\delta p - \rho a\, \delta u\right), \quad r_- = \begin{Bmatrix} 1/a^2 \\ -1/(\rho a) \\ 1 \end{Bmatrix}. \qquad (3\text{–}9)$$

For a stationary gas, pressure and velocity fluctuations propagate with the speed of sound symmetrically away from a point disturbance at the origin. Variations are translated onto the primitive variables according to the right eigenvectors.

3.1.2 Two-dimensional Analysis

The two-dimensional Euler equations admit simple wave solutions of the form

$$q = q(x\cos\theta + y\sin\theta - \lambda t), \tag{3-10}$$

where $q(x, y, t)$ is a constant on a plane surface in (x, y, t) space. Inserting this solution into the non-conservative form of the governing equations with primitive variables and stream-aligned coordinates,

$$\frac{\partial q}{\partial t} + A\frac{\partial q}{\partial x} + B\frac{\partial q}{\partial y} = 0, \tag{3-11}$$

yields

$$\left(-I\lambda t + A\cos\theta + B\sin\theta\right) q = 0. \tag{3-12}$$

A non-trivial solution is possible if the characteristic wave speed, $\lambda(\theta)$, is an eigenvalue of $A\cos\theta + B\sin\theta$. Convective plane wave fronts translate normal to themselves with speed, $\lambda(\theta)$, as shown in Fig. (27). For example, a particle that initially lies at the origin and on a convective plane wave front will propagate a distance $\lambda\Delta t$ after a time, $t = \Delta t$. Equivalently, a line of particles at $t = 0$ defined by $x\cos\theta + y\sin\theta = 0$ will lie along the line $x\cos\theta + y\sin\theta = \lambda\Delta t$.

The wave decomposition of Eqn. $(3-11)$ reveals four simple wave structures. Three waves are common to one-dimensional flow: the entropy and acoustic waves. The second dimension allows for a new wave, the shear or vorticity wave.

Two-dimensional Entropy Waves

The entropy wave propagates like the convective wave in Fig. (27). The eigenvalue, entropy-like characteristic variable, and right eigenvector are

$$\lambda_{\vec{V}} = |\vec{V}|\cos\theta, \quad \delta w_{\vec{V}} = -\frac{\delta p}{a^2} + \delta\rho, \quad r_{\vec{V}} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix}. \tag{3-13}$$

This simple wave disturbance causes changes in the density in an identical manner as the one-dimensional case.

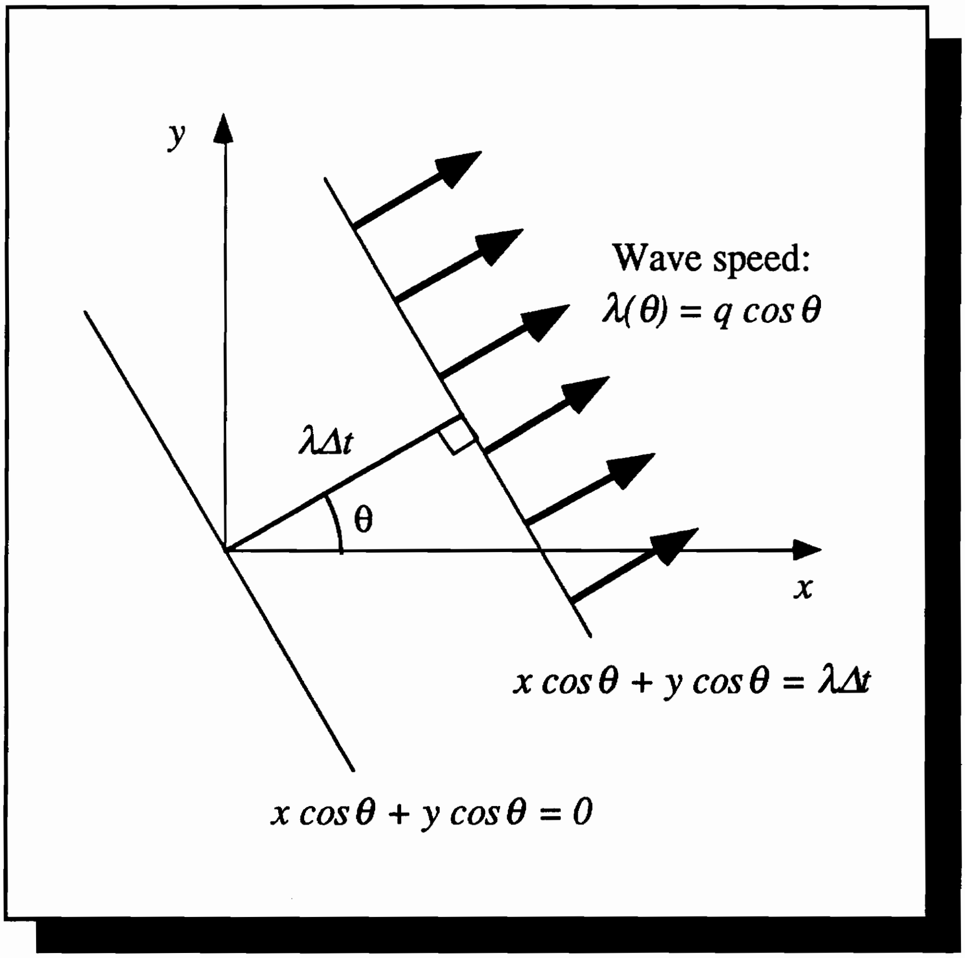
y
Wave speed:
$\lambda(\theta) = q\cos\theta$
$\lambda\Delta t$
θ
x
$x\cos\theta + y\cos\theta = \lambda\Delta t$
$x\cos\theta + y\cos\theta = 0$

Figure 27. Propagation of the convective plane wave front after time step $t = \Delta t$.

Two-dimensional Shear Waves

For the shear wave, the wave decomposition is

$$\lambda_\perp = |\vec{V}|\cos\theta, \quad \delta w_\perp = \hat{s}\cdot\vec{V}, \quad r_\perp = \left\{ \begin{array}{c} 0 \\ -\sin\theta \\ \cos\theta \\ 0 \end{array} \right\}, \tag{3–14}$$

where $\hat{s} = -\hat{\imath}\cos\theta + \hat{\jmath}\sin\theta$. For a plane wave, disturbances in the *cross-flow* velocity, $\delta w_\perp$, are propagated with the fluid velocity in the *stream-wise* direction causing variations in only the velocity components. This is the shearing or vortical motion shown in Fig. (28).

Two-dimensional Acoustic Waves

The two acoustic waves share symmetry about the flow speed. The forward moving acoustic wave is

$$\lambda_+ = |\vec{V}|\cos\theta + a, \quad \delta w_+ = \frac{1}{2}\left(\delta p + \rho a\,\delta|\vec{V}|\cos\theta\right), \quad r_+ = \frac{1}{\rho a}\left\{ \begin{array}{c} \rho/a \\ \cos\theta \\ \sin\theta \\ \rho a \end{array} \right\}, \tag{3–15}$$

and the backward moving wave is

$$\lambda_- = |\vec{V}|\cos\theta - a, \quad \delta w_- = \frac{1}{2}\left(\delta p - \rho a\,\delta|\vec{V}|\cos\theta\right), \quad r_- = \frac{1}{\rho a}\left\{ \begin{array}{c} \rho/a \\ -\cos\theta \\ -\sin\theta \\ \rho a \end{array} \right\}. \tag{3–16}$$

A particle initially located at the origin on the acoustic plane wave front moves tangent to a translating and growing circle of radius, $a\Delta t$, at $t = \Delta t$. This is the acoustic wave originating from a point disturbance shown in Fig. (29).

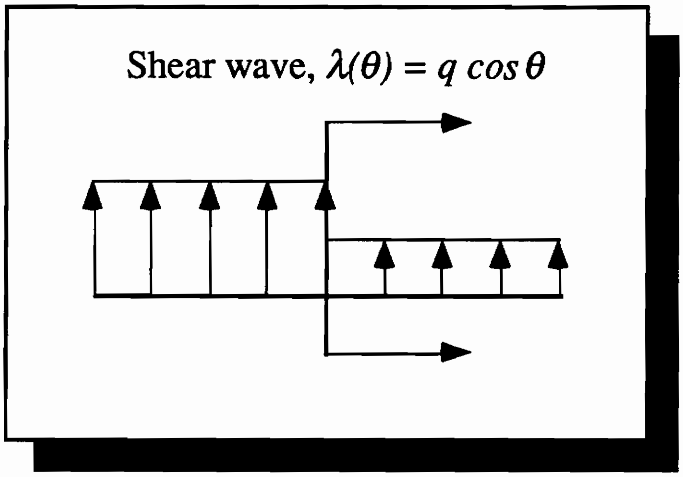

Figure 28. Effect of the simple shear wave on the velocity components.

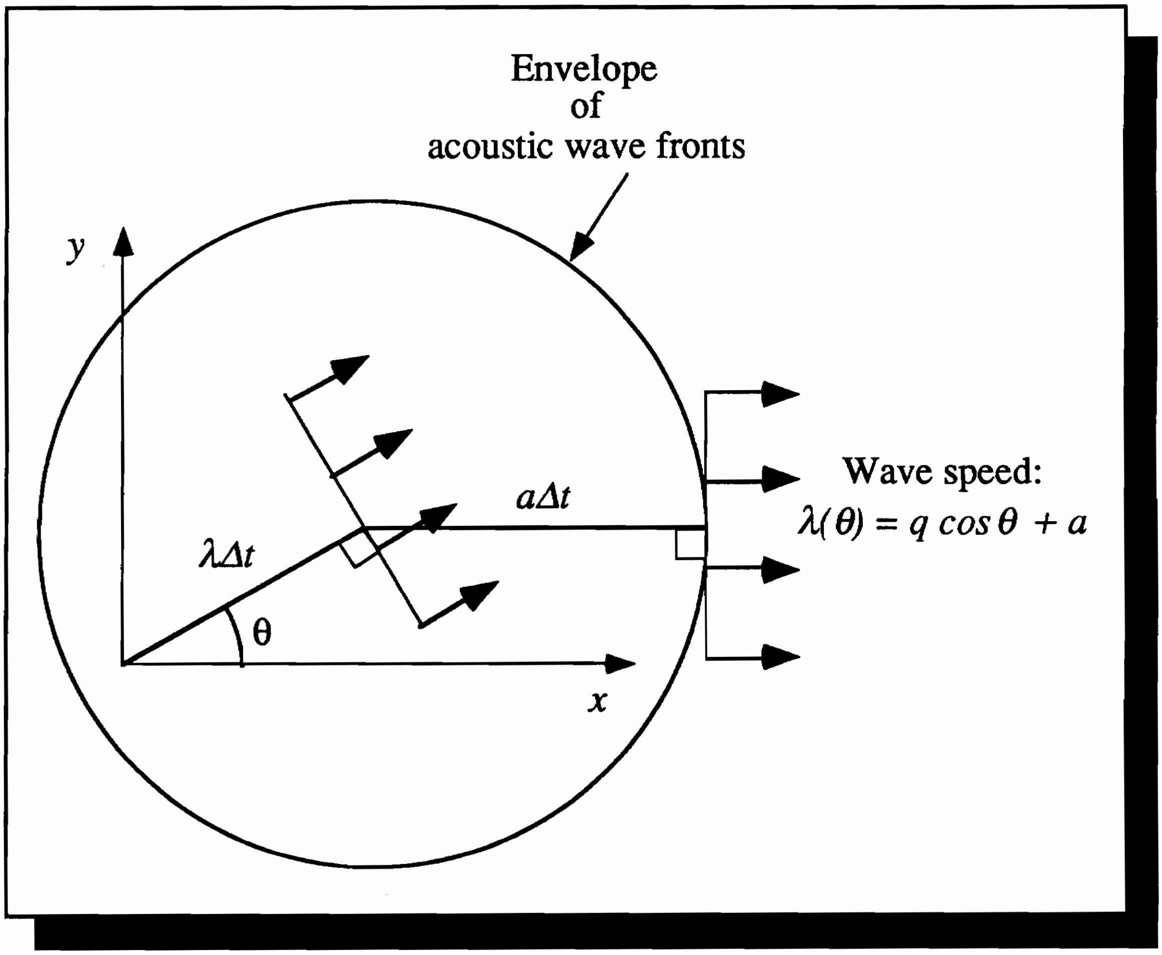

Figure 29. Circular acoustic plane wave-front pattern propagating away from the tip of the velocity vector.

3.2 Normalizing the Wave Speeds

This section begins by visualizing the previous mathematical depiction and summarizes much of the wave analysis of Van Leer *et.al* [64] and Lee [39]. Specifically, their derivation of the perfect-gas preconditioner for supersonic and subsonic flow speeds is discussed beginning in the present section and continuing through Section 3.2.4. In addition, the necessary modifications to the Roe flux is paraphrased in Section 3.5.

The wave speeds for the Euler equations are rewritten collectively as

$$\lambda(\theta) = a\left\{M\cos\theta - 1,\, M\cos\theta,\, M\cos\theta,\, M\cos\theta + 1\right\}. \tag{3–17}$$

The mathematical formula for determining the envelope of plane wave fronts from a general wave speed is

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} \lambda(\theta) \\ \lambda'(\theta) \end{Bmatrix}, \tag{3–18}$$

where $(x,\, y)$ is the planar position of the wave front due to the wave speed, $\lambda(\theta)$.

For supersonic flow, the convective wave speeds, $\lambda(\theta) = q\cos\theta$, form a circle with radius $M/2$. The convective plane wave fronts collapse to a point at the tip of the velocity vector. The θ-dependent acoustic wave speed forms a cardioid as shown in Fig. (30) for $M=2$. A polar plot of the eigenvalue, $\lambda(\theta)=M\cos\theta+1$, sweeps in the counter-clockwise direction starting at $\theta=0$ radians. Consider the wave speed at this starting point. The wave speed non-dimensionalized by the speed of sound maps to a point on the x-axis at $\lambda=3$. At $\theta=\pi/2$, the wave speed moving in the cross-flow direction yields a point on the y-axis with $\lambda=1$. The cardioid continues around as the angle traces through angles $\theta\in[0,2\pi]$.

Also shown in Fig. (30) are two wave speeds separated by a differential angle, $d\theta$, and their corresponding acoustic plane wave fronts. The envelope of plane wave fronts is simply the intersection of all of the wave crests. The acoustic plane wave fronts trace a circular envelope forming the Mach wedge.

For the convective waves the plane wave-front envelope collapses onto a point at the tip of the velocity vector. In fact, any wave-speed pattern which forms a circle will collapse onto a wave-front envelope which is a point. This underlying wave structure is crucial to determining the proper characteristic wave preconditioning matrix.

3.2.1 Convergence Difficulties

The Euler equations exhibit convergence difficulties when the wave speeds and corresponding plane wave envelopes are drastically mismatched between the waves. The optimal preconditioning matrix would alter the plane wave-front envelope so that all the waves propagate at identical speeds. The wave-front envelopes for the Euler equations at several subsonic and supersonic Mach numbers are shown in Fig. (31). At the sonic Mach number the upstream acoustic wave moves at zero speed. Similarly, at the very-low-speed regime the convective waves are nearly stagnant while the acoustic waves propagate with the speed of sound. Numerically, these two regimes exhibit the worst convergence rates.

3.2.2 Two-dimensional Preconditioning

The preconditioning matrix attempts to optimally alter the wave speeds in the governing equations without affecting the steady-state solution. To derive the preconditioning matrix, we begin with supersonic flow. The non-conservative form of the Euler equations including the preconditioning matrix is

$$\frac{\partial U}{\partial t} = -P\big(A\frac{\partial U}{\partial x} + B\frac{\partial U}{\partial y}\big) \qquad (3-19)$$

The first step to determining the matrix, P, is called stream-wise preconditioning. This step amounts to setting P to the inverse of A which is a positive-definite matrix

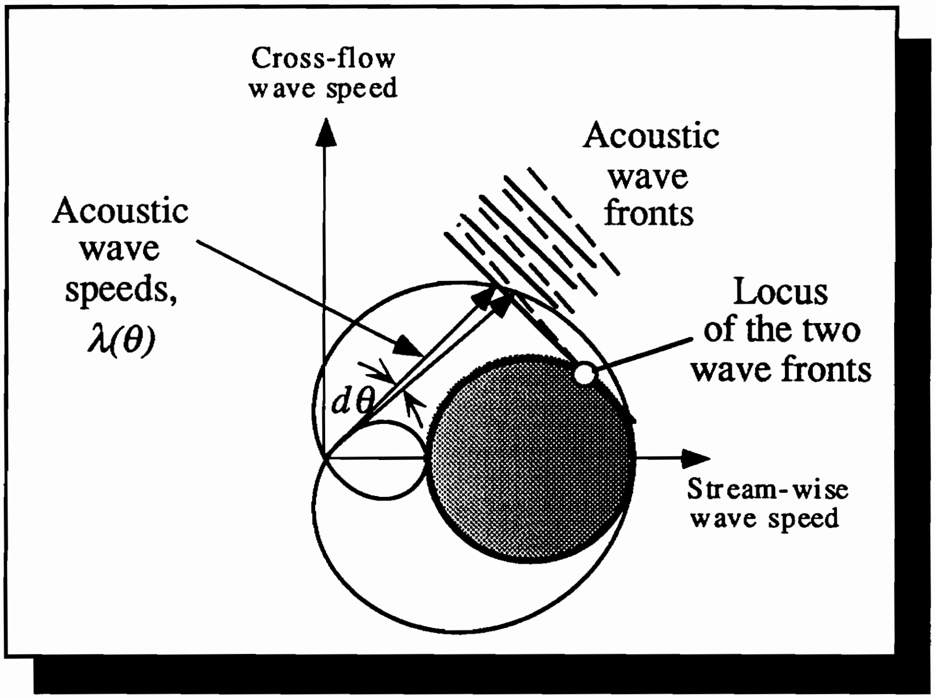

Figure 30. Supersonic acoustic wave speeds, plane wave fronts and envelope of acoustic plane wave fronts.

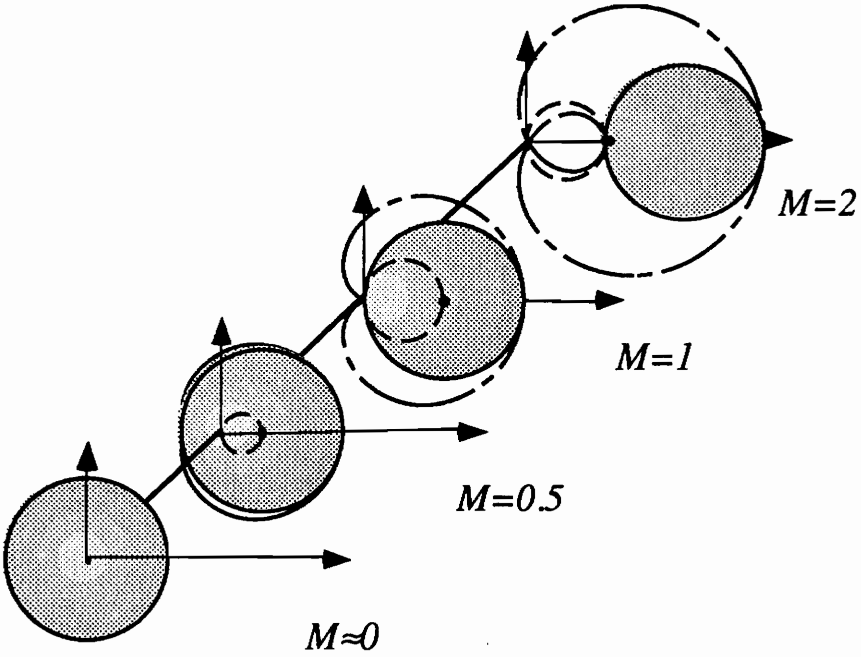

Figure 31. Convective and acoustic wave patterns for the two-dimensional Euler equations at Mach numbers of $M=0.1$, 0.5, 1.0 and $M=2.0$.

for the supersonic-flow condition. Specifically,

$$P = A^{-1}. \tag{3–20}$$

We have now altered the governing equations but a steady-state solution to the preconditioned equations remains a solution to the unmodified Euler equations. The wave speeds admitted by the stream-wise preconditioned system are the eigenvalues of

$$I\cos\theta + A^{-1}B\sin\theta, \tag{3–21}$$

which are

$$\begin{aligned}\lambda(\theta) = a\{M\cos\theta - \frac{M}{\sqrt{M^2-1}}\sin\theta,\ M\cos\theta,\ M\cos\theta,\\ M\cos\theta + \frac{M}{\sqrt{M^2-1}}\sin\theta\}.\end{aligned} \tag{3–22}$$

The preconditioned system may be simultaneously diagonalized with the change of variables

$$dV = T^{-1}dU, \tag{3–23}$$

where

$$A^{-1}B = T\Lambda T^{-1}. \tag{3–24}$$

The diagonal system is now

$$\frac{\partial V}{\partial t} = -(I\frac{\partial V}{\partial x} + \Lambda\frac{\partial V}{\partial y}) \tag{3–25}$$

The acoustic waves form a circle and the corresponding envelopes collapse onto points symmetric about the stream-wise axis. The acoustic and convective patterns for the stream-wise preconditioning are shown in Fig. (32).

The dots represent the envelopes of the acoustic and convective plane wave fronts. We want to normalize the wave speeds which correspond to these wave-front envelopes. As shown in the figure, the ratio of the convective wave speed to the acoustic wave speed is $M/\sqrt{M^2-1}$. Therefore, the acoustic waves must

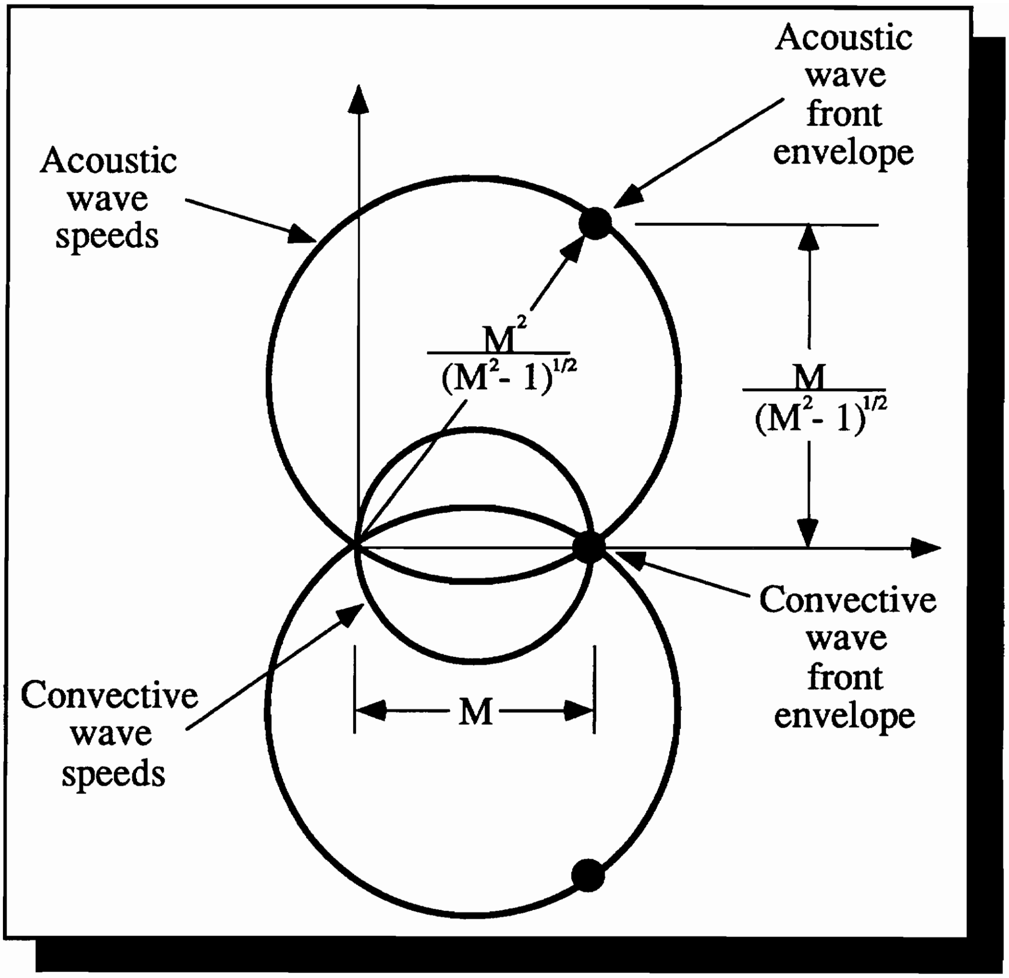
Acoustic wave front envelope
Acoustic wave speeds
$\frac{M^2}{(M^2-1)^{1/2}}$
$\frac{M}{(M^2-1)^{1/2}}$
Convective wave front envelope
Convective wave speeds
M

Figure 32. Streamwise preconditioning wave pattern at $M=2$. The heavy circles are the convective and acoustic wave speeds. The dots represent the envelope of the plane wave fronts.

be scaled by the inverse of this magnitude so that the waves corresponding to the convective and acoustic wave-front envelopes propagate at equivalent speeds. The matrix, X, which changes the acoustic speeds in the stream-wise preconditioned system without modifying the convective waves is

$$X = diag\,\{\tau,\ 1,\ 1,\ \tau\} \tag{3–26}$$

where

$$\tau = \frac{\sqrt{M^2-1}}{M} \tag{3–27}$$

and scales the acoustic waves.

This matrix multiplies the diagonal form of the stream-wise preconditioned equations as

$$\frac{\partial V}{\partial t} = -X\big(I\frac{\partial V}{\partial x} + \Lambda\frac{\partial V}{\partial y}\big). \tag{3–28}$$

This is the last step in a two-step process that generates the supersonic preconditioning matrix. Now, the fully preconditioned system has wave speeds

$$\begin{aligned}\lambda(\theta) = a\big\{\sqrt{M^2-1}\cos\theta - \sin\theta,\ M\cos\theta,\ M\cos\theta,\\ \sqrt{M^2-1}\cos\theta + \sin\theta\big\}.\end{aligned} \tag{3–29}$$

These wave speeds and associated plane wave fronts are shown in Fig. (33) for $M=1$. Notice that the envelopes are all equally positioned away from the origin yielding unit condition number. Theoretically, we will obtain infinitely better convergence rate at sonic flow conditions.

Working backwards to the original preconditioned form in Eqn. (3–19) yields the supersonic preconditioning matrix

$$P = TXT^{-1}A^{-1}. \tag{3–30}$$

Performing all the algebra gives the supersonic wave-preconditioning matrix

$$P = \begin{bmatrix} \frac{\tau}{\beta^2}M^2 & -\frac{\tau}{\beta^2}M & 0 & 0 \\ -\frac{\tau}{\beta^2}M & \frac{\tau}{\beta^2}+1 & 0 & 0 \\ 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3–31}$$

where $\tau = \sqrt{M^2 - 1}/M$ and $\beta = \sqrt{M^2 - 1}$.

Unfortunately, the subsonic preconditioning matrix does not follow the same two-step process that generates the supersonic preconditioning matrix. For subsonic flow the waves travel omni-directionally and the stream-wise preconditioning step with $P = |A|^{-1}$ does not reveal the same insightful propagation directions as do the acoustic waves for supersonic flow. To derive the subsonic matrix, assume the matrix shares the same sparseness as the supersonic preconditioning matrix. The general subsonic matrix is then

$$P = \begin{bmatrix} A & D & 0 & 0 \\ D & B & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3–32}$$

The task is to determine the entries A, B, C and D for the optimal subsonic characteristic wave pattern. The convective waves should remain unchanged and the acoustic waves should maintain their symmetry around the velocity vector. If the shear and entropy waves should remain unmodified, their speeds are $\lambda(\theta) = q\cos\theta$. The stream-wise acoustic waves must be symmetric about the velocity vector, and at the incompressible limit, $M \to 0$, they must travel at the flow speed in opposite directions. With these constraints on the eigenvalues, the unknown matrix entries become

$$\begin{aligned} A &= M^2/\sqrt{1 - M^2} \\ B &= 1/\sqrt{1 - M^2} + 1 \\ C &= \sqrt{1 - M^2} \\ D &= -M/\sqrt{1 - M^2} \end{aligned} \tag{3–33}$$

The resulting wave pattern is then

$$\begin{aligned} \lambda(\theta) = a\{&-M\sqrt{(1 - M^2)\cos^2\theta + \sin^2\theta},\ M\cos\theta,\ M\cos\theta, \\ &M\sqrt{(1 - M^2)\cos^2\theta + \sin^2\theta}\}. \end{aligned} \tag{3–34}$$

The plane wave fronts and plane wave-front envelopes at several subsonic and supersonic Mach numbers are shown in Fig. (34). Notice that the envelope of the

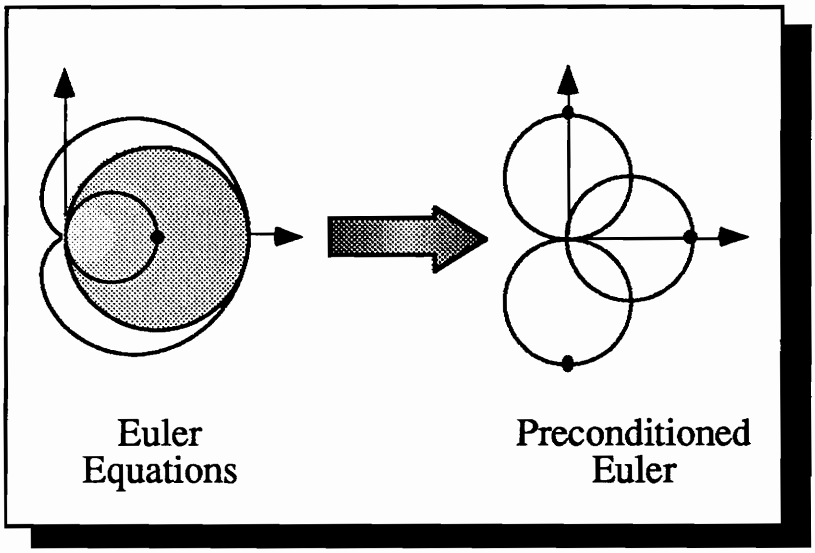

Figure 33. Comparison of acoustic and convective plane wave-front envelopes at the sonic Mach number, $M=1$.

plane wave fronts smoothly transition into the supersonic regime at $M=1$. For the low-speed region the envelope of acoustic plane wave fronts become circles. In the incompressible limit, the wave speeds corresponding to the acoustic and convective fronts are perfectly matched. Specifically, the "acoustic" wave speeds adopt the convective speed rather than the sonic speed.

A weakness exists at a stagnation point where the convective wave fronts for the preconditioned equations never reduce to points which should occur at exactly the zero flow condition. A smoothing procedure is suggested by Van Leer based upon physical reasoning, but is not employed in this dissertation. The preconditioning calculations are smoothed with a minimum threshold set for the offending terms.

In the stream-aligned system with symmetrization variables as derived in Chapter 2, the two-dimensional preconditioning matrix, P, that optimally normalizes the acoustic and convective wave fronts at all Mach numbers is

$$P = \begin{bmatrix} \frac{\tau}{\beta^2}M^2 & -\frac{\tau}{\beta^2}M & 0 & 0 \\ -\frac{\tau}{\beta^2}M & \frac{\tau}{\beta^2}+1 & 0 & 0 \\ 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-35}$$

The switching parameters, τ and β, are determined from the Mach number as

$$\begin{aligned} \beta &= \begin{cases} \sqrt{M^2-1}, & \text{if } M \geq 1; \\ \sqrt{1-M^2}, & \text{if } M < 1, \end{cases} \\ \tau &= \begin{cases} \sqrt{1-M^{-2}}, & \text{if } M \geq 1; \\ \sqrt{1-M^2}, & \text{if } M < 1. \end{cases} \end{aligned} \tag{3-36}$$

3.2.3 Three-dimensional Preconditioning

The three-dimensional preconditioning matrix must equalize all the acoustic waves in an identical way. This requires that the wave speeds, which are now propagating in three dimensions instead of the restricted two-dimensional plane, have no

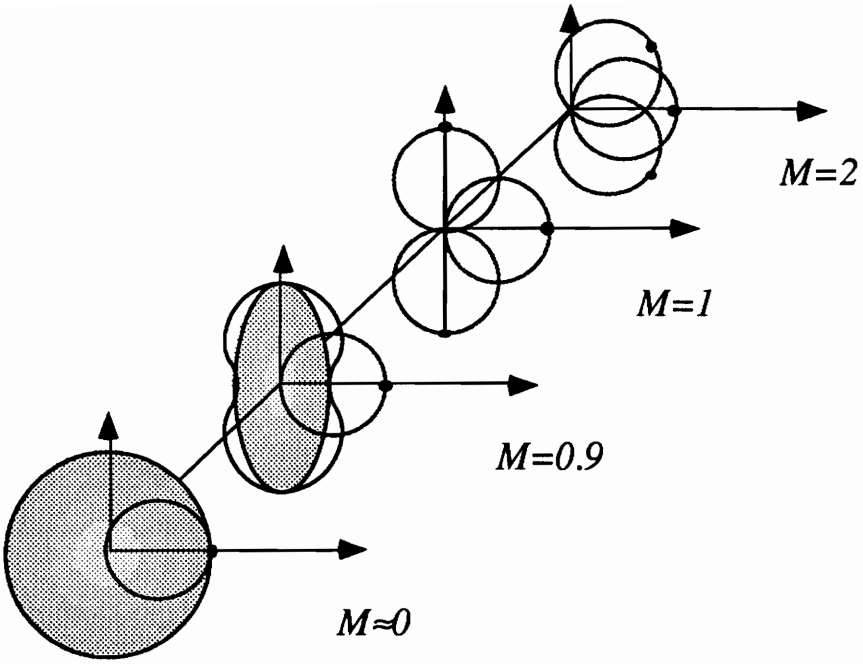

Figure 34. Convective and acoustic wave patterns for the two-dimensional *preconditioned* Euler equations at Mach numbers of $M=0.1$, 0.9, 1.0 and $M=2.0$.

angular dependence, ϕ, in the cross-flow plane, Fig. (35). The only preconditioning matrix with the form of Eqn. (3−35) that satisfies this independence criterion is

$$P = \begin{bmatrix} \frac{\tau}{\beta^2}M^2 & -\frac{\tau}{\beta^2}M & 0 & 0 & 0 \\ -\frac{\tau}{\beta^2}M & \frac{\tau}{\beta^2}+1 & 0 & 0 & 0 \\ 0 & 0 & \tau & 0 & 0 \\ 0 & 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (3\text{–}37)$$

Unfortunately, now τ multiplies and shortens a stream-wise shear wave speed penalizing the supersonic condition number.

The theoretical acceleration in convergence rate is the ratio of the condition numbers with and without preconditioning. A summary of the subsonic and supersonic condition numbers and corresponding theoretical acceleration rates is given in Table (9). The minimum theoretical speed-up for subsonic flow is $(3\sqrt{3})/2$ and occurs at a Mach number of $M = 0.5$. For supersonic flow, the benefit of preconditioning decreases as the Euler waves grow closer together with increasing Mach number.

3.2.4 Preconditioned-Euler Fourier Footprints

The wave structure is also evident in the Fourier footprint introduced in Chapter 2. For the preconditioned Euler equations, consider low-speed flow at $M = 0.1$, first-order modified Roe flux (discussed in Section 3.5) and a Courant number of $\nu = 1$. The full Fourier footprint is shown in Fig. (36).

This footprint combines all four waves into a single Fourier transform. Each wave contribution may be visualized by showing only the convective and acoustic eigenvalues individually. The convective entropy and shear waves form circles just like the scalar wave equation as shown in Fig. (37). The magnitudes of the two convective waves are controlled by the unit terms in elements (4, 4) and (2, 2) of the preconditioning matrix in Eqn. (3−35). Replacing unity by a variable allows

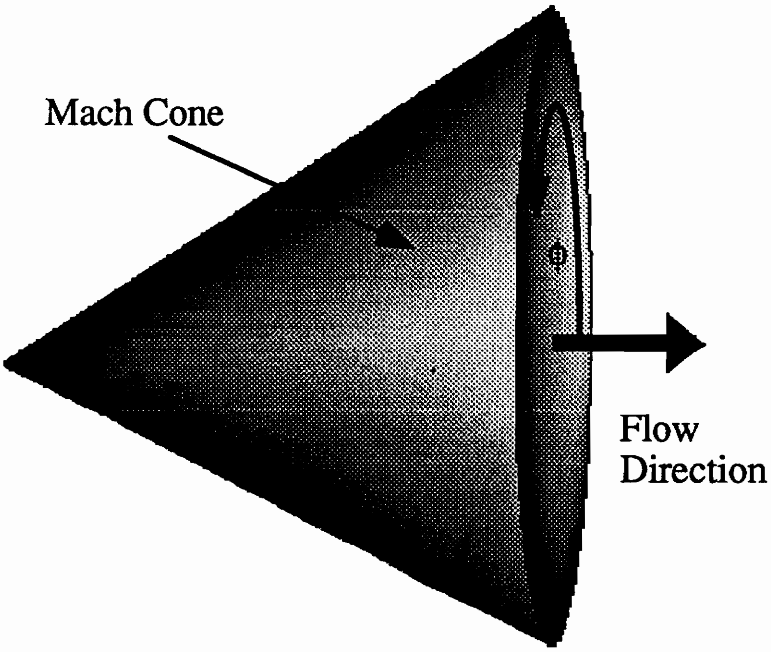

Figure 35. Supersonic three-dimensional Mach cone illustrating angular variable, ϕ.

Table 9. Theoretical convergence-rate acceleration based on the partial differential equations' condition numbers.

Mach No. Range	Cond. No. (Euler)	Cond. No. (Pre. Euler)	Cond. No. Ratio
$0 < M \leq 1/2$	$\frac{1+M}{M}$	$\frac{1}{\sqrt{1-M^2}}$	$\frac{1+M}{M}\sqrt{1-M^2}$
$1/2 \leq M < 1$ (2-D or 3-D)	$\frac{1+M}{1-M}$		$\frac{1+M}{1-M}\sqrt{1-M^2}$
$M > 1$ (2-D)	$\frac{M+1}{M-1}$	1	$\frac{M+1}{M-1}$
$M > 1$ (3-D)		$\frac{M}{\sqrt{M^2-1}}$	$\frac{M+1}{M}\sqrt{\frac{M+1}{M-1}}$

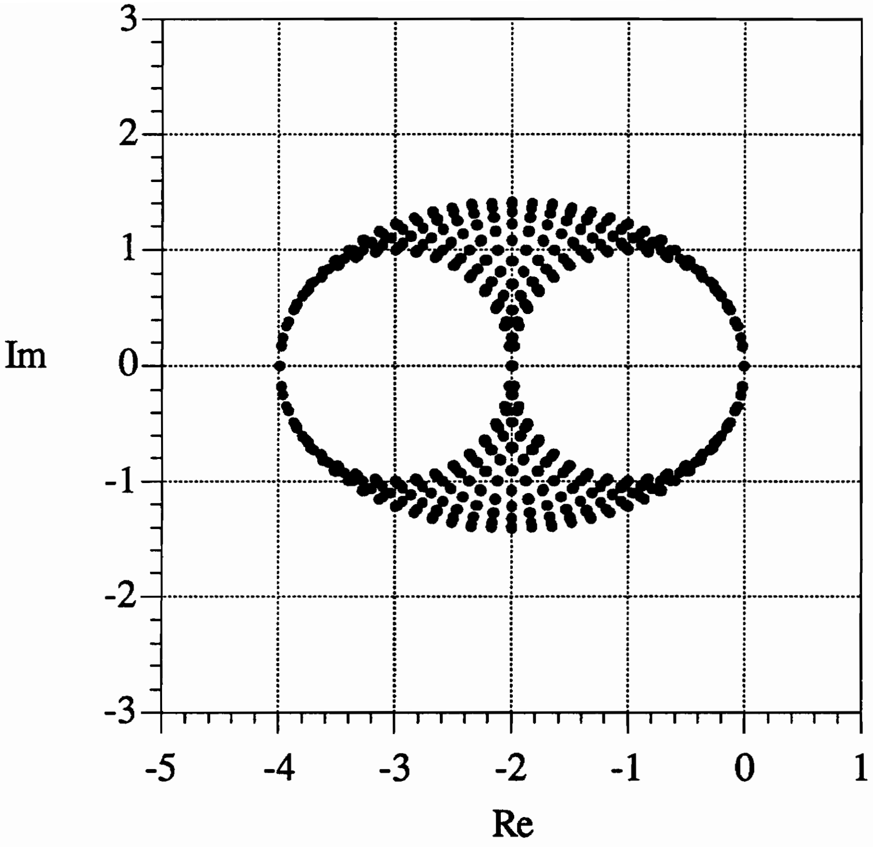

Figure 36. Full Fourier footprint of the preconditioned Euler equations at Mach number, $M = 0.1$.

us to linearly stretch or compress the convective waves up and down the negative real axis. For stability these waves may be multiplied by twice their current speed. This affects the wave damping qualities of the high frequency waves since they are moved down the negative real axis. After doubling the convective wave speeds, the high-frequency errors in both the convective waves and acoustic waves are equally damped.

The individual acoustic footprints are shown in Figs. (38) and (39). Error waves in both directions, β_x and β_y, are zero at the origin. The low-frequency wave ($\beta_y = 0$) in the cross-flow direction and the high-frequency wave in the stream-wise direction ($\beta_x = \pi$) fall on the negative real axis at negative two. The high-high frequency region ($\beta_x = \beta_y = \pi$) collapses onto the negative real axis at negative four.

3.2.5 Extension to Generalized Chemistry

For the extension of the preconditioning matrix to include finite-rate chemistry, we adopt the formulation proposed by Grossman and Cinnella [24]. They assume a system of N species and M diatomic molecules considered to have contributions to the internal energy from non-equilibrium vibration. The modeled system is assumed to have J reactions of the form

$$\nu'_{1,j}X_1 + \nu'_{2,j}X_2 + \cdots + \nu'_{N,j}X_N \rightleftharpoons \nu''_{1,j}X_1 + \nu''_{2,j}X_2 + \cdots + \nu''_{N,j}X_N, \quad j = 1, 2, \ldots, J, \tag{3–38}$$

where $\nu'_{i,j}$ and $\nu''_{i,j}$ are the stoichiometric coefficients of the reactants and products respectively of species i in the j^{th} reaction. For non-equilibrium chemistry, the rate of production of species i is determined by summing the contribution from each J reaction. The production term is written

$$\begin{aligned} \dot{w}_i &\equiv \frac{d\rho_i}{dt} \\ &= \mathcal{M}_i \sum_{j=1}^{J} (\nu''_{i,j} - \nu'_{i,j}) \left[k_{f,j} \prod_{n=1}^{N} \left(\frac{\rho_n}{\mathcal{M}_n} \right)^{\nu'_{n,j}} - k_{b,j} \prod_{n=1}^{N} \left(\frac{\rho_n}{\mathcal{M}_n} \right)^{\nu''_{n,j}} \right], \\ &\qquad i = 1, 2, \ldots, N, \end{aligned} \tag{3–39}$$

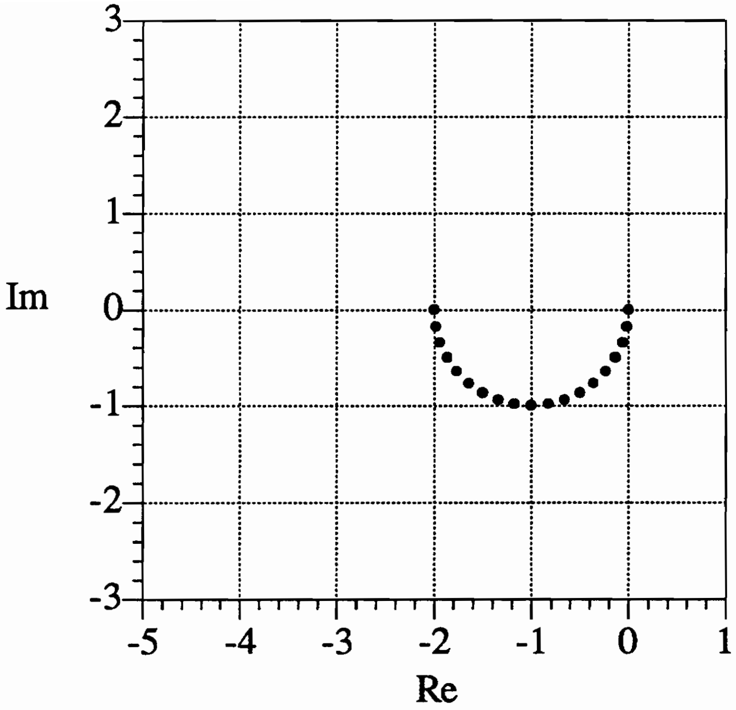

Figure 37. Convective shear and entropy waves of the full Fourier footprint at Mach number of $M = 0.1$.

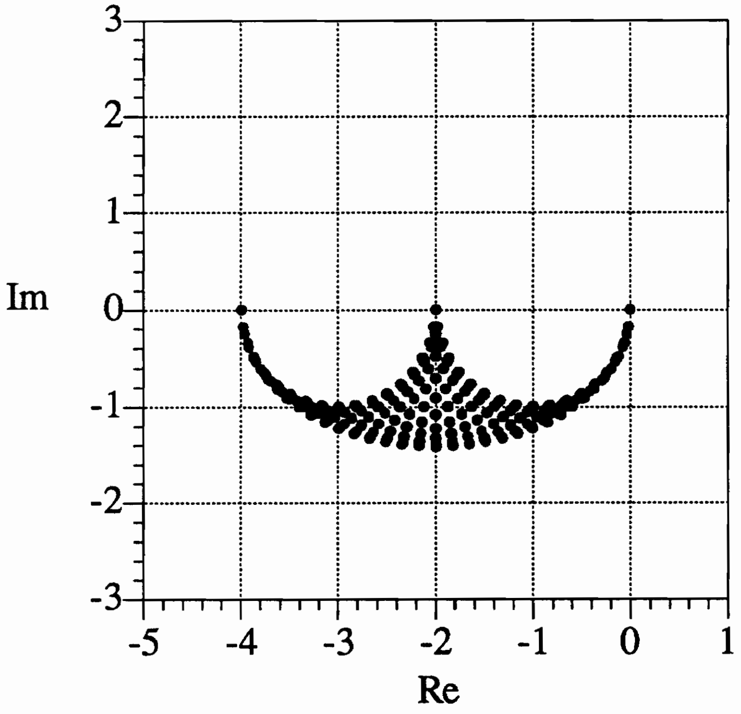

Figure 38. Downstream-moving acoustic-wave portion of the full Fourier footprint at Mach number of $M = 0.1$.

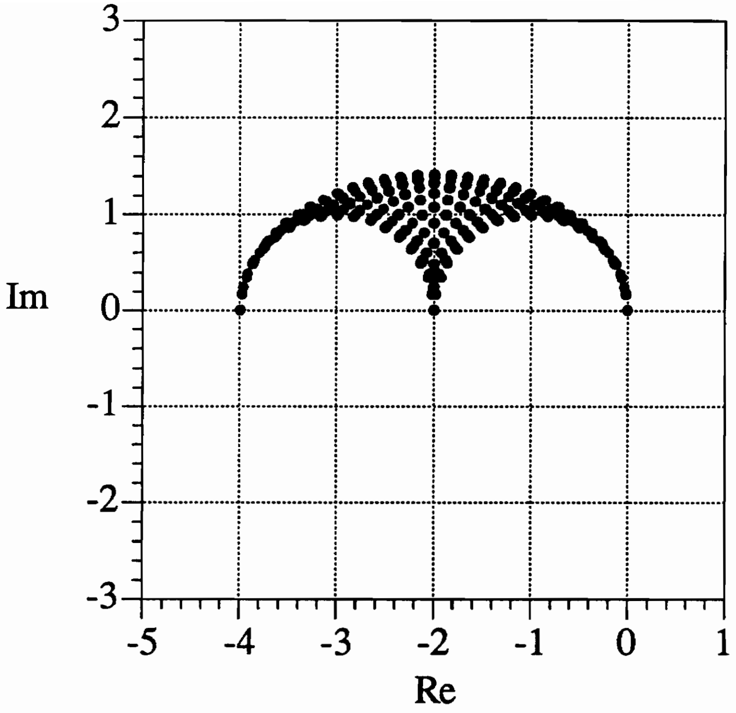

Figure 39. Upstream-moving acoustic-wave portion of the full Fourier footprint at Mach number of $M = 0.1$.

where ρ_n is the mass density and $\mathcal{M}_i$ is the molecular weight of species i. The forward and backward reaction rates of reaction j are denoted $k_{f,j}(T)$ and $k_{b,j}(T)$, respectively, and are assumed to be known functions of the temperature.

The thermodynamic model assumes all the species are of nearly the same molecular mass which allows for a single translational temperature (*i.e.* $T_i = T$ for all species i). Note, for example, that a very-small-mass species like electrons are not modeled adequately. With a single translational temperature and assuming that each species behaves as a thermally perfect gas, thermodynamic equilibrium allows

$$e_i = e_i(T) \quad \text{and} \quad p_i = \rho_i R_i T. \tag{3–40}$$

The model for the non-equilibrium assumes that the internal energy is determined from two contributions. One portion of the internal energy for each species is considered in thermodynamic equilibrium, and the other portion is in a non-equilibrium state. The internal energy per unit mass of species i is defined as

$$e_i \equiv \tilde{e}_i(T) + e_{n_i}, \tag{3–41}$$

where $\tilde{e}_i$ is the equilibrium portion and e_{n_i} is the portion in thermodynamic non-equilibrium. The equilibrium portion is defined by

$$\tilde{e}_i \equiv \int_{T_{ref}}^{T} \tilde{c}_{v_i}(\tau)\, d\tau + h_{f_i} \tag{3–42}$$

where $\tilde{c}_{v_i} = d\tilde{e}_i/dT$, and h_{f_i} is the heat of formation of species i. The internal energy per unit mass of the mixture may be written as

$$e = \sum_{i=1}^{N} \frac{\rho_i}{\rho} e_i = \tilde{e} + \sum_{j=1}^{M} \frac{\rho_j}{\rho} e_{n_j}. \tag{3–43}$$

With a simplified vibration model

$$\tilde{e}_i = \tilde{c}_{v_i} T + h_i^0 \tag{3–44}$$

and

$$e_{n_j} = e_{v_j},$$

where h_i^0 is defined to be

$$h_i^0 \equiv h_{f_i} + \tilde{c}_{v_i} T_{ref}. \tag{3–45}$$

For this model we assume translational and rotation equilibrium which would mean that for monatomic gases, $\tilde{c}_{v_i} = \frac{3}{2} R_i$, and for diatomic gases, $\tilde{c}_{v_i} = \frac{5}{2} R_i$. The production of vibrational energy per unit mass, e_{v_j}, for diatomic gases is determined through a vibrational relaxation process for a harmonic oscillator [75]. The vibrational energy production rate is then

$$\dot{e}_{v_i} = \frac{e^*_{v_i} - e_{v_i}}{\tau_i(T)}, \tag{3–46}$$

where $e^*_{v_i}$ is the equilibrium contribution,

$$e^*_{v_i} = \frac{R_i \Theta_{v_i}}{e^{(\Theta_{v_i}/T)} - 1},$$

and Θ_{v_i} is the characteristic temperature for vibration of species i. We use a Landau-Teller model to determine the local relaxation time, τ_i, of species i. The relaxation time is then

$$\tau_i = \frac{k_{1_i} T^{(5/6)} e^{(k_{2_i}/T)^{1/3}}}{p\left[1 - e^{(\Theta_{v_i}/T)}\right]}. \tag{3–47}$$

where k_{1_i} and k_{2_i} for several elementary particles are available in Vicenti and Kruger [75].

For a inviscid, non-heat-conducting, three-dimensional flow with non-equilibrium chemistry and non-equilibrium internal energy, the governing equations modeled in differential conservation-law form are

$$\frac{\partial Q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = W. \tag{3–48}$$

The state vector of conserved quantities and flux vectors are of length $N+M+4$ where N is the number of chemical species, M is the number of diatomic species which are consider in vibrational non-equilibrium. The remaining four equations appear from the momentum and energy equations. The conservation variables, flux vectors, and source term are

$$Q = \left\{ \begin{array}{c} \rho_1 \\ \vdots \\ \rho_N \\ \rho u \\ \rho v \\ \rho w \\ \rho_1 e_{n_1} \\ \vdots \\ \rho_M e_{n_M} \\ \rho e_0 \end{array} \right\}, \quad f = \left\{ \begin{array}{c} \rho_1 u \\ \vdots \\ \rho_N u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho_1 e_{n_1} u \\ \vdots \\ \rho_M e_{n_M} u \\ \rho u h_0 \end{array} \right\}, \quad g = \left\{ \begin{array}{c} \rho_1 v \\ \vdots \\ \rho_N v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho_1 e_{n_1} v \\ \vdots \\ \rho_M e_{n_M} v \\ \rho v h_0 \end{array} \right\}, \tag{3–49}$$

and

$$h = \left\{ \begin{array}{c} \rho_1 w \\ \vdots \\ \rho_N w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho_1 e_{n_1} w \\ \vdots \\ \rho_M e_{n_M} w \\ \rho w h_0 \end{array} \right\}, \quad W = \left\{ \begin{array}{c} \dot{w}_1 \\ \vdots \\ \dot{w}_N \\ 0 \\ 0 \\ 0 \\ \rho_1 \dot{e}_{n_1} + e_{n_1} \dot{w}_1 \\ \vdots \\ \rho_M \dot{e}_{n_M} + e_{n_M} \dot{w}_M \\ 0 \end{array} \right\}. \tag{3–50}$$

In the above equations, ρ is the mixture density, (u, v, w) are the velocities in the (x, y, z) directions respectively, p is the pressure, e_0 is the total energy per unit mass, h_0 is total enthalpy per unit mass, ρ_i is the mass density of species i, and e_{n_i} is the non-equilibrium portion of the internal energy per unit mass of species i. The total enthalpy is defined as $h_0 \equiv e_0 + p/\rho$. The equations may be written in generalized coordinates by standard techniques available in Hirsch [34].

The system is completed by the definition of density

$$\rho = \sum_{i=1}^{N} \rho_i \tag{3–51}$$

and the equation of state which is known implicitly through

$$p = \sum_{i=1}^{N} \rho_i R_i T \tag{3–52}$$

and

$$e = \sum_{i=1}^{N} \frac{\rho_i}{\rho} \left[\int_{T_{ref}}^{T} \tilde{c}_{v_i}(\tau)\, d\tau + h_{f_i} \right] + \sum_{j=1}^{M} \frac{\rho_j}{\rho} e_{n_j}. \tag{3–53}$$

The mixture specific heats and gas constant are determined through

$$\tilde{c}_v = \sum_{i=1}^{N} \frac{\rho_i}{\rho} \tilde{c}_{v_i}, \quad \tilde{c}_p = \sum_{i=1}^{N} \frac{\rho_i}{\rho} \tilde{c}_{p_i}, \tag{3–54}$$

and

$$\tilde{R} = \sum_{i=1}^{N} \frac{\rho_i}{\rho} \tilde{R}_i = \tilde{c}_p - \tilde{c}_v. \tag{3–55}$$

Following Grossman and Cinnella [24], the ratio of specific heats is conveniently defined

$$\tilde{\gamma} \equiv \frac{\tilde{c}_p}{\tilde{c}_v}$$

so that the frozen speed of sound becomes simply

$$a^2 = \frac{\tilde{\gamma} p}{\rho}. \tag{3–56}$$

The preconditioning matrix extended to generalized finite-rate chemistry must accommodate a flexible system of moving waves. Fortunately, no new kind of wave appears in the system. We have only entropy and vorticity waves moving with the fluid and acoustic waves propagating in Mach-number dependent directions. The non-equilibrium vibrational energies and mass fractions act strictly as convective waves. The variables, U, which symmetrize the inviscid flux Jacobians are defined for generalized finite-rate chemistry in differential form as

$$dU = \left\{ \begin{matrix} dp/(\rho a) \\ dp - (\rho a^2/\rho_2)d\rho_2 \\ \vdots \\ dp - (\rho a^2/\rho_N)d\rho_N \\ du \\ dv \\ dw \\ dp - (\rho a^2/\rho_1)d\rho_1 - (\rho a^2/e_{n_1})de_{n_1} \\ \vdots \\ dp - (\rho a^2/\rho_M)d\rho_M - (\rho a^2/e_{n_M})de_{n_M} \\ dp - (\rho a^2/\rho_1)d\rho_1 \end{matrix} \right\}. \tag{3–57}$$

Note that this matrix reduces to the symmetrization variables of Eqn. (2−33) for two-dimensional flow of a perfect gas. The same derivation that generates the preconditioning matrix in Eqn. (3–35) yields the generalized finite-rate preconditioner.

The preconditioning matrix for finite rate chemistry in a stream-aligned coordinate system is then

$$P = \begin{bmatrix} \frac{\tau}{\beta^2}M^2 & 0 & \cdots & 0 & -\frac{\tau}{\beta^2}M & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 1 & & \vdots & 0 & \vdots & \vdots & \vdots & & & \vdots & \vdots \\ \vdots & & \ddots & 0 & \vdots & \vdots & \vdots & \vdots & & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ -\frac{\tau}{\beta^2}M & 0 & \cdots & 0 & \frac{\tau}{\beta^2}+1 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & \cdots & \cdots & 0 & 0 & \tau & 0 & \vdots & & & \vdots & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 0 & \tau & 0 & \cdots & \cdots & 0 & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & \vdots & \vdots & \vdots & \vdots & 0 & \ddots & & \vdots & \vdots \\ \vdots & & & \vdots & \vdots & \vdots & \vdots & \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}. \tag{3–58}$$

Here τ and β are dependent upon the Mach number through

$$\begin{aligned} \beta &= \begin{cases} \sqrt{M^2-1}, & \text{if } M \geq 1; \\ \sqrt{1-M^2}, & \text{if } M < 1, \end{cases} \\ \tau &= \begin{cases} \sqrt{1-M^{-2}}, & \text{if } M \geq 1; \\ \sqrt{1-M^2}, & \text{if } M < 1. \end{cases} \end{aligned} \tag{3–59}$$

3.3 Explicit Implementation

As stated previously, the preconditioning matrix is derived in a coordinate system that is not appropriate for numerical implementation. To use the above preconditioning matrix, we must transform back to a conservative system and update the state variables of our choice. Note that the state variables may be the conservative or primitive variables or any other system which contains the velocity components and two thermodynamic variables. This includes both the pressure and density based formulations. It's simply a matter of choice.

We choose to transform from stream-aligned, symmetrization variables to Cartesian, conservation variables. This requires a rotation of the velocity axes and a variable transformation. Preconditioning the governing equations and using standard Euler explicit time integration results in the following numerical procedure.

$$\Delta Q = -\frac{\Delta t}{Vol}\frac{\partial Q}{\partial q}\frac{\partial q}{\partial U}\frac{\partial U}{\partial \bar{U}}P\frac{\partial \bar{U}}{\partial U}\frac{\partial U}{\partial q}\frac{\partial q}{\partial Q}R(Q^n) \tag{3-60}$$

To determine the time step, Δt, the CFL-like relaxation factor, ν, is defined as

$$\nu \equiv \frac{q\Delta t}{\Delta L}. \tag{3-61}$$

The characteristic length, ΔL, is any length that reduces to the one-dimensional definition for grid-aligned flow. Notice the absence of the sound speed in the above CFL definition. For the preconditioned equations the spectral radius is simply the convective velocity, and does not include the sound speed. This new definition of the CFL number is important for the proper time-step calculation in slow-flow regions. Simplifying the notation we define

$$\mathcal{P} \equiv \frac{\partial q}{\partial U}\frac{\partial U}{\partial \bar{U}}P\frac{\partial \bar{U}}{\partial U}\frac{\partial U}{\partial q}. \tag{3-62}$$

The matrix, $\mathcal{P}$, for finite rate chemistry in three dimensions is

$$\mathcal{P} = \begin{bmatrix} 1 & 0 & \cdots & 0 & -\frac{\tau}{\beta^2}\frac{\rho_1}{a^2}\vec{V}^T & 0 & \cdots & \cdots & 0 & \frac{\rho_1}{\rho}\left(\frac{\tau M^2-\beta^2}{a^2\beta^2}\right) \\ 0 & \ddots & & 0 & -\frac{\tau}{\beta^2}\frac{\rho_2}{a^2}\vec{V}^T & \vdots & \ddots & & \vdots & \frac{\rho_2}{\rho}\left(\frac{\tau M^2-\beta^2}{a^2\beta^2}\right) \\ \vdots & & \ddots & 0 & \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & -\frac{\tau}{\beta^2}\frac{\rho_N}{a^2}\vec{V}^T & 0 & \cdots & \cdots & 0 & \frac{\rho_N}{\rho}\left(\frac{\tau M^2-\beta^2}{a^2\beta^2}\right) \\ 0 & \cdots & \cdots & 0 & \mathbf{p} & 0 & \cdots & \cdots & 0 & -\frac{\tau}{\beta^2}\frac{\vec{V}}{\rho a^2} \\ 0 & \cdots & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots & \vdots & 0 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & -\frac{\tau}{\beta^2}\rho\vec{V}^T & 0 & \cdots & \cdots & 0 & -\frac{\tau}{\beta^2}M^2 \end{bmatrix}, \qquad (3\text{--}63)$$

where

$$\mathbf{p} = \frac{1}{q^2}\left[(1+\frac{\tau}{\beta^2}-\tau)\vec{\mathbf{V}}^T\vec{\mathbf{V}}+\tau q^2\mathbf{I}\right]. \qquad (3\text{--}64)$$

If we update primitive variables, instead of conservative variables as in Eqn. (3–60), the Euler explicit algorithm becomes

$$\Delta q = -\frac{\Delta t}{Vol}\mathcal{P}\frac{\partial q}{\partial Q}R(q^n). \qquad (3\text{--}65)$$

3.4 Implicit Implementation

For implicit time integration, we move the preconditioning matrix of Eqn. (3–19) to the left-hand side. Then, the system can be solved using any direct or indirect solver in the literature. Note that with very high time steps the effect of the preconditioning is canceled out. However, we can achieve significant improvements for low-speed flows and for simulations using approximate factorization. Standard Euler implicit time integration for the preconditioned equations yields the following formulation to update primitive variables:

$$\left[\frac{Vol}{\Delta t}\frac{\partial Q}{\partial q}\mathcal{P}^{-1}+\frac{\partial R}{\partial q}\right]^n \Delta q = -R(q^n). \qquad (3\text{--}66)$$

The matrix, $\mathcal{P}^{-1}$, for generalized finite-rate chemistry in three dimensions is

$$\mathcal{P}^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 & \rho_1 \vec{V}^T/q^2 & 0 & \cdots & \cdots & 0 & \frac{\rho_1}{\rho q^2}\left(1-M^2+\frac{\beta^2}{\tau}\right) \\ 0 & \ddots & & 0 & \rho_2 \vec{V}^T/q^2 & \vdots & \ddots & & \vdots & \frac{\rho_2}{\rho q^2}\left(1-M^2+\frac{\beta^2}{\tau}\right) \\ \vdots & & \ddots & 0 & \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & \rho_N \vec{V}^T/q^2 & 0 & \cdots & \cdots & 0 & \frac{\rho_N}{\rho q^2}\left(1-M^2+\frac{\beta^2}{\tau}\right) \\ 0 & \cdots & \cdots & 0 & \mathbf{\Pi} & 0 & \cdots & \cdots & 0 & \vec{V}/(\rho q^2) \\ 0 & \cdots & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots & \vdots & 0 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & \rho \vec{V}^T/M^2 & 0 & \cdots & \cdots & 0 & (\tau+\beta^2)/(\tau M^2) \end{bmatrix}, \tag{3–67}$$

where

$$\mathbf{\Pi} = \frac{1}{q^2}\left[\frac{(\tau-1)}{\tau}\vec{\mathbf{V}}^T\vec{\mathbf{V}} + \frac{q^2}{\tau}\mathbf{I}\right]. \tag{3–68}$$

The matrices $\partial Q/\partial q$ and $\partial q/\partial Q$ are given in Grossman and Cinnella [24].

3.5 Modified Flux Formulas

In developing flux formulas for the preconditioned Euler equations, the form of the numerical scheme must consistently represent the modeled partial differential equations. Therefore, the artificial viscosity for an approximate Riemann solver must be modified for preconditioning. For the original Roe algorithm, the artificial-viscosity matrix [39] is

$$\mathcal{A.V.} = |\hat{A}| = |A\cos\theta + B\sin\theta|, \tag{3–69}$$

which consistently represents the non-conservative form of the Euler equations. Here, $\cos\theta$ and $\sin\theta$ are the cell-face-normal direction cosines. The equivalent artificial-viscosity matrix for preconditioning must be

$$|P\hat{A}| = |P(A\cos\theta + B\sin\theta)| \tag{3–70}$$

and not

$$P|\hat{A}| = P|A\cos\theta + B\sin\theta| \tag{3–71}$$

to consistently depict the preconditioned Euler equations. For a robust artificial-viscosity matrix, a slight modification is still necessary. The matrix $|P\hat{A}|$ contains singular terms for certain flow angles, particularly the free-stream direction. An alternative stems from the multi-dimensional Riemann solver research [52] and conservatively sums the fluxes from multiple Riemann problems. The alternative artificial-viscosity matrix is then

$$|P\hat{A}| = |PA\cos\theta_{\bar{x}}| + |PB\cos\theta_{\bar{y}})| + |PC\cos\theta_{\bar{z}})|, \tag{3–72}$$

where $\theta_{\bar{x}}$, $\theta_{\bar{y}}$, and $\theta_{\bar{z}}$ are the direction cosines between the flow direction orthonormals and the cell face normal. The full equations are documented below. The inverse of P must premultiply the artificial-viscosity matrix so that the modified Roe flux is preserved upon multiplication of the residual by the preconditioning matrix. The resulting modified flux becomes

$$\hat{f}_{j+1/2} = \frac{1}{2}\left(\hat{f}_j + \hat{f}_{j+1}\right) - \frac{1}{2}\hat{P}^{-1}|\hat{P}\hat{A}|(Q_{j+1} - Q_j). \tag{3–73}$$

Grossman and Cinnella [24] proposed a Roe algorithm for finite-rate chemistry which employs the jumps in the mass fractions. The method proposed below incorporates jumps in the species densities which is more efficient to code. The modified Roe's scheme for three-dimensional implementation with finite-rate chemistry may be written

$$\hat{f}_{j+1/2} = \frac{1}{2}\left(\hat{f}_j + \hat{f}_{j+1}\right) - \frac{1}{2}\sum_{i=1}^{3}[\![f]\!]_i, \tag{3–74}$$

where

$$\llbracket f \rrbracket_1 = \frac{-\llbracket p \rrbracket}{\tilde{a}^2} |\tilde{\tilde{u}}| \left\{ \begin{array}{c} \tilde{\rho}_1/\tilde{\rho} \\ \vdots \\ \tilde{\rho}_N/\tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \widetilde{\rho_1 e_{n_1}}/\tilde{\rho} \\ \vdots \\ \widetilde{\rho_M e_{n_M}}/\tilde{\rho} \\ \Theta_1 \end{array} \right\} \quad + \quad |\tilde{\tilde{u}}| \left\{ \begin{array}{c} \llbracket \rho_1 \rrbracket \\ \vdots \\ \llbracket \rho_N \rrbracket \\ \tilde{u}\llbracket \rho \rrbracket + \tilde{\rho}\llbracket u \rrbracket \\ \tilde{v}\llbracket \rho \rrbracket + \tilde{\rho}\llbracket v \rrbracket \\ \tilde{w}\llbracket \rho \rrbracket + \tilde{\rho}\llbracket w \rrbracket \\ \llbracket \rho_1 e_{n_1} \rrbracket \\ \vdots \\ \llbracket \rho_M e_{n_M} \rrbracket \\ \Theta_2 \end{array} \right\}, \qquad (3\text{--}75)$$

$$\llbracket f \rrbracket_2 = \frac{1}{\tilde{q}^2} \left(\llbracket p \rrbracket + \tilde{\beta}^2 \llbracket p \rrbracket + \tilde{\rho}\tilde{q}\llbracket u_q \rrbracket \right) |\tilde{\tilde{u}}| \left\{ \begin{array}{c} \tilde{\rho}_1/\tilde{\rho} \\ \vdots \\ \tilde{\rho}_N/\tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \widetilde{\rho_1 e_{n_1}}/\tilde{\rho} \\ \vdots \\ \widetilde{\rho_M e_{n_M}}/\tilde{\rho} \\ \tilde{h}_0 \end{array} \right\} \quad + \quad \frac{\llbracket p \rrbracket}{\tilde{q}^2} |\tilde{\tilde{u}}| \left\{ \begin{array}{c} 0 \\ \vdots \\ 0 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ 0 \\ \vdots \\ 0 \\ \tilde{q}^2 \end{array} \right\}, \qquad (3\text{--}76)$$

$$\begin{aligned}\llbracket f \rrbracket_3 = \frac{\tilde{\beta}}{\tilde{q}} \llbracket p \rrbracket \left(|\cos\theta_{\bar{y}}| + |\cos\theta_{\bar{z}}| \right) &\left\{ \begin{matrix} \tilde{\rho}_1/\tilde{\rho} \\ \vdots \\ \tilde{\rho}_N/\tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \widetilde{\rho_1 e_{n_1}}/\tilde{\rho} \\ \vdots \\ \widetilde{\rho_M e_{n_M}}/\tilde{\rho} \\ \tilde{h}_0 \end{matrix} \right\} \\ + \frac{\tilde{\rho}\tilde{q}}{\tilde{\beta}} &\left\{ \begin{matrix} 0 \\ \vdots \\ 0 \\ \hat{l}_{q_x} \llbracket v_q \rrbracket |\cos\theta_{\bar{y}}| + \hat{m}_{q_x} \llbracket w_q \rrbracket |\cos\theta_{\bar{z}}| \\ \hat{l}_{q_y} \llbracket v_q \rrbracket |\cos\theta_{\bar{y}}| + \hat{m}_{q_y} \llbracket w_q \rrbracket |\cos\theta_{\bar{z}}| \\ \hat{l}_{q_z} \llbracket v_q \rrbracket |\cos\theta_{\bar{y}}| + \hat{m}_{q_z} \llbracket w_q \rrbracket |\cos\theta_{\bar{z}}| \\ 0 \\ \vdots \\ 0 \\ 0 \end{matrix} \right\}, \end{aligned} \tag{3–77}$$

where

$$\Theta_1 \equiv \tilde{h}_0 - \frac{\tilde{a}^2}{\tilde{\tilde{\gamma}} - 1} \tag{3–78}$$

$$\Theta_2 \equiv \sum_{i=1}^{N} \tilde{\beta}_i \llbracket \rho_i \rrbracket + \sum_{j=1}^{M} \llbracket \rho_j e_{n_j} \rrbracket + \tilde{\rho}\tilde{u} \llbracket u \rrbracket + \tilde{\rho}\tilde{v} \llbracket v \rrbracket + \tilde{\rho}\tilde{w} \llbracket w \rrbracket . \tag{3–79}$$

Two sets of direction cosines are involved in the modified Roe algorithm. One set, $(\cdot)_{q_x}, (\cdot)_{q_y}, (\cdot)_{q_z}$, transforms from Cartesian coordinates to the Roe average stream-aligned coordinate system. This transformation may be determined by many different axis rotation sequences and is therefore somewhat arbitrary. The velocity vector in Cartesian coordinates has components $\{\tilde{u}, \tilde{v}, \tilde{w}\}$, so the unit vector along the stream-aligned coordinate is

$$\vec{k}_q \equiv \{\tilde{u}/\tilde{q},\ \tilde{v}/\tilde{q},\ \tilde{w}/\tilde{q}\}. \tag{3–80}$$

The transformation used here is

$$\begin{bmatrix} \hat{k}_{q_x} & \hat{k}_{q_y} & \hat{k}_{q_z} \\ \hat{l}_{q_x} & \hat{l}_{q_y} & \hat{l}_{q_z} \\ \hat{m}_{q_x} & \hat{m}_{q_y} & \hat{m}_{q_z} \end{bmatrix} = \begin{bmatrix} \cos\phi & \cos\theta\sin\phi & \sin\theta\sin\phi \\ -\sin\phi & \cos\theta\cos\phi & \sin\theta\cos\phi \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}. \tag{3–81}$$

The angles ϕ, θ are determined from the velocity components as $\tan\theta = w/v$ and $\tan\phi = \sqrt{v^2+w^2}/u$. Note especially that a singularity occurs for $u = q$, (*i.e.* the flow is perfectly aligned with the Cartesian x-axis). This case represents the only flow condition that does not require a similarity transformation or equivalently the similarity transformation should be the identity matrix. To accommodate all flow cases a small perturbation on the order of the machine's accuracy should be added to the y component of the velocity. With this change, $\cos\theta = 1$ and $\sin\theta = 0$ for the case when $u = q$, and the identity matrix is the similarity transformation. The jumps in contravariant velocity are constructed as

$$\begin{aligned} [\![u_q]\!] &\equiv \hat{k}_{q_x}[\![u]\!] + \hat{k}_{q_y}[\![v]\!] + \hat{k}_{q_z}[\![w]\!] \\ [\![v_q]\!] &\equiv \hat{l}_{q_x}[\![u]\!] + \hat{l}_{q_y}[\![v]\!] + \hat{l}_{q_z}[\![w]\!] \\ [\![w_q]\!] &\equiv \hat{m}_{q_x}[\![u]\!] + \hat{m}_{q_y}[\![v]\!] + \hat{m}_{q_z}[\![w]\!]. \end{aligned} \tag{3–82}$$

The other set are the afore-mentioned direction cosines between the stream-aligned coordinate unit vectors and the cell face normal, $\vec{n}$. These are obtained with the dot product as

$$\begin{aligned} \cos\theta_{\bar{x}} &\equiv \vec{k}_q \cdot \vec{n} \\ \cos\theta_{\bar{y}} &\equiv \vec{l}_q \cdot \vec{n} \\ \cos\theta_{\bar{z}} &\equiv \vec{m}_q \cdot \vec{n}. \end{aligned} \tag{3–83}$$

The flow variables $\tilde{\rho}, \tilde{u}, \tilde{v}, \tilde{w}, \tilde{\rho}_i, \widetilde{\rho_i e_{n_i}}, \tilde{h}_0, \tilde{\beta}_i, \tilde{\tilde{e}}_i, \tilde{\tilde{R}}, \tilde{T}, \tilde{\tilde{\gamma}}, c^*_{v_i}, \tilde{c}^*_v$, and $\tilde{a}^2$ are determined from Roe averages as follows.

$$\tilde{\rho} = \sqrt{\rho_R\,\rho_L}, \tag{3–84}$$

$$\tilde{u} = \frac{< u\sqrt{\rho} >}{< \sqrt{\rho} >}, \quad \tilde{v} = \frac{< v\sqrt{\rho} >}{< \sqrt{\rho} >}, \quad \tilde{w} = \frac{< w\sqrt{\rho} >}{< \sqrt{\rho} >}, \tag{3–85}$$

$$\tilde{\rho}_i = \frac{\sqrt{\rho_R}\,\rho_{i_L} + \sqrt{\rho_L}\,\rho_{i_R}}{\sqrt{\rho_R} + \sqrt{\rho_L}}, \quad i = 1, \ldots, N \tag{3–86}$$

$$\widetilde{\rho_j e_{n_j}} = \frac{\sqrt{\rho_R}\,(\rho_j e_{n_j})_L + \sqrt{\rho_L}\,(\rho_j e_{n_j})_R}{\sqrt{\rho_R} + \sqrt{\rho_L}}, \quad j = 1, \ldots, M \tag{3–87}$$

$$\tilde{h}_0 = \frac{< h_0\sqrt{\rho} >}{< \sqrt{\rho} >}, \quad \tilde{\beta}_i = \tilde{\tilde{e}}_i - \frac{R_i \tilde{T}}{\tilde{\tilde{\gamma}} - 1} + \frac{\tilde{q}^2}{2}, \tag{3–88}$$

$$\tilde{\tilde{e}}_i = \frac{< \tilde{e}_i\sqrt{\rho} >}{< \sqrt{\rho} >}, \quad \tilde{\tilde{R}} = \frac{< \tilde{R}\sqrt{\rho} >}{< \sqrt{\rho} >}, \quad \tilde{T} = \frac{< T\sqrt{\rho} >}{< \sqrt{\rho} >}, \tag{3–89}$$

$$\tilde{\tilde{\gamma}} = 1 + \frac{\tilde{\tilde{R}}}{\tilde{c}_v^*} \tag{3–90}$$

$$c_{v_i}^* = \frac{1}{[\![T]\!]} \int_{T_L}^{T_R} c_{v_i}(\tau)\, d\tau, \quad \tilde{c}_v^* = \sum_{i=1}^{N} \frac{\tilde{\rho}_i c_{v_i}^*}{\tilde{\rho}} \tag{3–91}$$

$$\tilde{a}^2 = (\tilde{\tilde{\gamma}} - 1)\left(\tilde{h}_0 - \frac{\tilde{q}^2}{2} + \tilde{c}_v^* \tilde{T} - \sum_{i=1}^{N} \frac{\tilde{\rho}_i \tilde{\tilde{e}}_i}{\tilde{\rho}} - \sum_{j=1}^{M} \frac{\widetilde{\rho_j e_{n_j}}}{\tilde{\rho}}\right). \tag{3–92}$$

The contravariant velocity is

$$\tilde{\tilde{u}} \equiv \hat{k}_x \tilde{u} + \hat{k}_y \tilde{v} + \hat{k}_z \tilde{w}. \tag{3–93}$$

Both the Roe and modified Roe algorithm have the property that for a stationary discontinuity, the numerical flux function returns a full flux. This means that for left and right states which satisfy the Rankine-Hugoniot jump conditions, the numerical fluxes sum to yield machine-zero residual. This is also true for contact discontinuities and, unfortunately, for non-physical entropy shocks. A well-designed entropy fix (cf. Harten [31]) ensures that the second law of thermodynamics is satisfied without disturbing the flux function's ability to perfectly resolve stationary discontinuities.

3.6 Euler Implicit Damping Characteristics

In this section, we retreat briefly to the one-dimensional Euler equations. From within this simple framework, the deficiencies of a typical compressible code can be seen. We analyze Euler implicit time integration with the preconditioning algorithms proposed above. Of course, we include the standard non-preconditioned scheme as a comparison.

Recall that a discrete grid restricts the representative range of wavelengths and frequencies. The smallest wavelength and corresponding highest frequency that can be depicted on a finite discretization are $\lambda = 2\Delta x$ and $\beta_x = \pi$, respectively. The wavelength, wave number, and frequency are related through the following:

$$\lambda = \frac{2\pi}{m} = \frac{2\pi}{\beta_x}\Delta x. \tag{3-94}$$

We are interested in the spectral radius at both long and short wavelengths, (*i.e.* $\beta_x \to 0$ and $\beta_x \to \pi$) because typically, during convergence to the steady state, short waves decay quickly while long waves persist, expiring gradually.

3.6.1 Local Time Stepping

We want to determine the damping behavior of the implicit time operator over the entire Mach number range. The damping is determined by the gain matrix from a Von Neumann stability analysis. We want to determine the spectral radius, and thus the amplification factor, of the update matrix. This matrix is a function of the Mach number and the Fourier frequency.

Performing a Von Neumann stability analysis of the Euler implicit procedure with Roe flux-difference splitting produces an amplification matrix whose inverse is given below.

$$G^{-1} = I + \frac{2\lambda}{M+1}|A|\sin^2\frac{\beta_x}{2} + i\frac{\lambda}{M+1}A\sin\beta_x. \tag{3-95}$$

The Courant number is defined as $\lambda \equiv (u+a)\Delta t/\Delta x$. The smallest eigenvalue will be the inverse of the spectral radius. The three analytic eigenvalues of the inverse gain matrix are

$$
\begin{aligned}
\lambda_- &= 1 + \frac{2\lambda|M-1|}{M+1}\sin^2\frac{\beta_x}{2} + i\frac{\lambda|M-1|}{M+1}\sin\beta_x, \\
\lambda_u &= 1 + \frac{2\lambda M}{M+1}\sin^2\frac{\beta_x}{2} + i\frac{\lambda M}{M+1}\sin\beta_x, \\
\lambda_+ &= 1 + 2\lambda\sin^2\frac{\beta_x}{2} + i\lambda\sin\beta_x.
\end{aligned}
\qquad (3\text{–}96)
$$

A carpet plot of the spectral radii over a Mach number range of 10^{-2} to 10 and Fourier frequencies from lowest to highest is given in Fig. (64). Notice that the spectral radius approaches unity in the limit of $M \to 0$ for all wavelengths and equals unity at exactly Mach 1. A nearly unitary gain results in minimal damping of a solution error in these flow regimes. Notice that significant damping occurs for supersonic and moderate subsonic Mach numbers for the high frequency waves. However, as the grid points in a simulation increase in number, longer wavelengths are numerically represented (*i.e.* $\beta_x \to 0$) and their gain approaches unity.

3.6.2 Characteristic Time Stepping

For analysis, we use the one-dimensional equivalent of the two-dimensional preconditioning matrix by replacing the wave scale, τ, with unity. A Von Neumann stability analysis with preconditioning and the modified Roe flux produces an amplification matrix whose inverse is

$$
G^{-1} = I + \frac{2\nu}{M}|PA|\sin^2\frac{\beta_x}{2} + i\frac{\nu}{M}PA\sin\beta_x \qquad (3\text{–}97)
$$

where $\nu \equiv u\Delta t/\Delta x$. The three eigenvalues of the inverse are identical and are

$$
\lambda_- = \lambda_u = \lambda_+ = 1 + 2\nu\sin^2\frac{\beta_x}{2} + i\nu\sin\beta_x. \qquad (3\text{–}98)
$$

Notice the absence of the Mach number in the above equation. More specifically, we have the same damping characteristics at all Mach numbers as the scalar

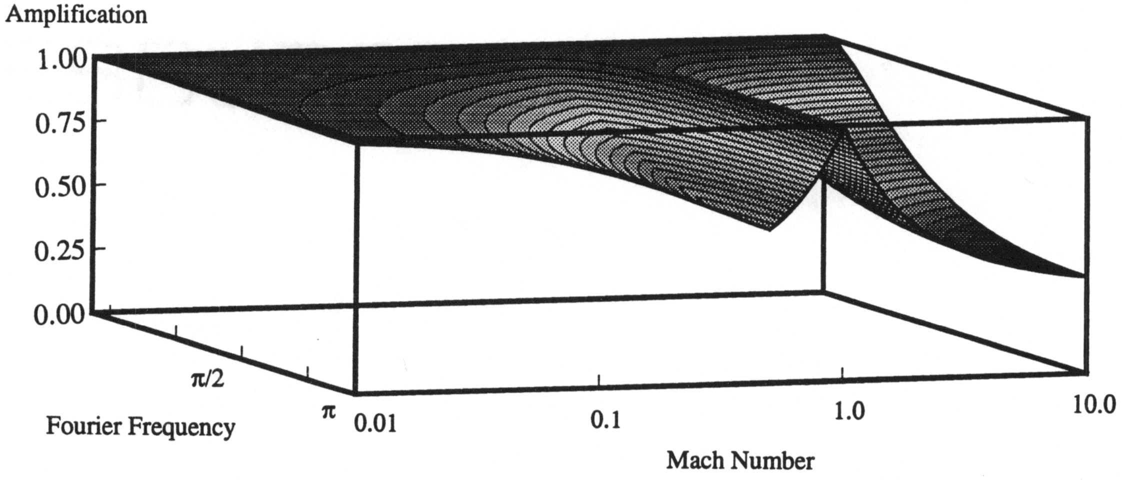

Figure 40. Gain over a range of Mach numbers for the Euler equations with first-order Roe flux-difference splitting using Euler implicit time integration and $\lambda = 1.0$.

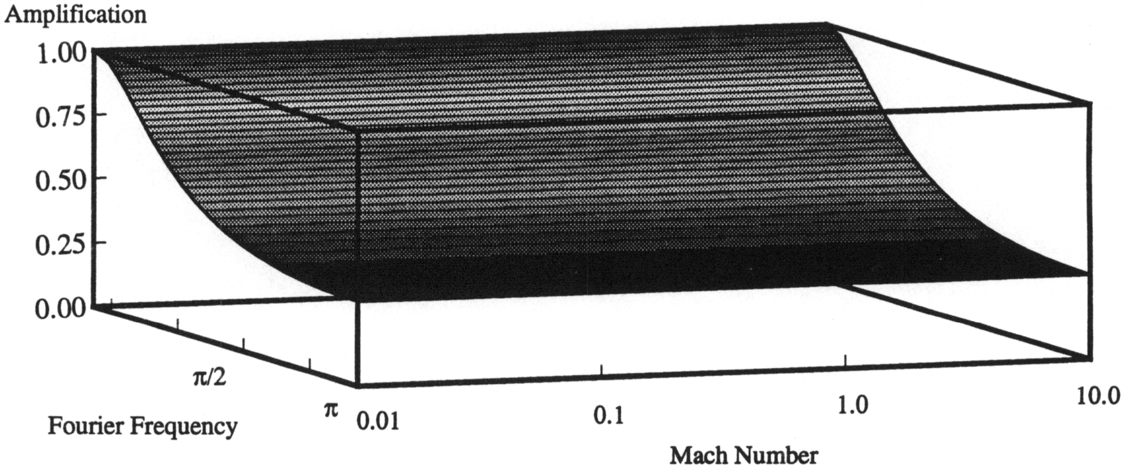

Figure 41. Gain over a range of Mach numbers for the preconditioned Euler equations with first-order *modified* Roe flux-difference splitting using Euler implicit time integration and $\nu = 1.0$.

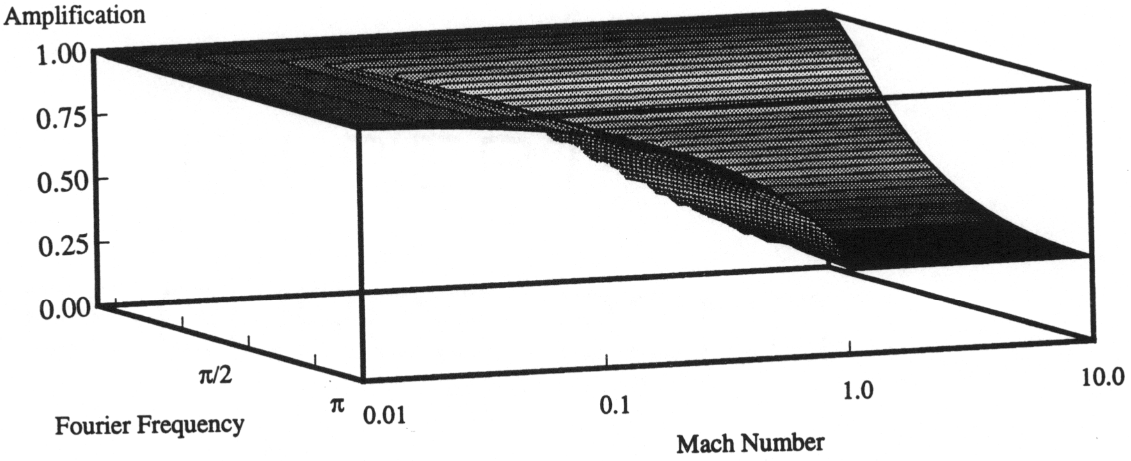

Figure 42. Gain over a range of Mach numbers for the preconditioned Euler equations with first-order *standard* Roe flux-difference splitting using Euler implicit time integration and $\nu = 1.0$.

wave equation. This is precisely our goal. A carpet plot of the gain for the range of frequencies and the same representative range of Mach numbers is illustrated in Fig. (41). We see significant improvement from the damping shown in Fig. (40), which is the current state of the art.

One should be careful. If the standard Roe algorithm is preconditioned without any flux modification as in Eqn. (3–46), the spectral radius degrades in the subsonic region for all frequencies as shown in Fig. (42). When treated implicitly, this formulation is unconditionally stable, yet the gain approaches unity as $M \rightarrow 0$, and, thus, we have poor subsonic damping. Using an explicit time marching strategy, the same algorithm requires a prohibitively small time step for stability [40]. The modified Roe flux is necessary for preconditioning subsonic flows.

3.7 A Note on Boundary Conditions

This section describes the underlying finite non-linear waves for the preconditioned Euler equations emphasizing future work on characteristic boundary conditions. The characteristic compatibility relations appear in this dissertation as an original contribution. The one-dimensional preconditioned Euler equations differentiated with respect to primitive variables yields the following wave structure.

3.7.1 Riemann Invariants

Without preconditioning the characteristic compatibility equations are

$$du \pm \frac{dp}{\rho a} = 0 \qquad (3\text{–}99)$$

along the characteristic defined by

$$\frac{dx}{dt} = u \pm a. \qquad (3\text{–}100)$$

The Riemann invariants are determined by integrating along the characteristic path and are

$$R^{\pm} = u \pm \frac{2a}{\gamma - 1}. \qquad (3\text{–}101)$$

The above quantities are invariant along their characteristic paths. The same analysis of the preconditioned equations gives fundamental concepts to formulate the boundary conditions. After some minor algebraic manipulation, the compatibility equations are

$$dp = 0 \tag{3–102}$$

on the characteristic line defined by

$$\frac{dx}{dt} = u\frac{M-1}{|M-1|}. \tag{3–103}$$

Notice that the sound speed is now absent and that the wave moves upstream for subsonic flow. The other "acoustic" wave equation is

$$dp + \rho u \, du = 0 \tag{3–104}$$

on

$$\frac{dx}{dt} = u. \tag{3–105}$$

Integration of the characteristic compatibility equations gives the Riemann invariants. Since

$$\frac{p}{p_0} = \left(\frac{a}{a_0}\right)^{\frac{2\gamma}{\gamma-1}} \tag{3–106}$$

and

$$\rho = \frac{\gamma p}{a^2} \tag{3–107}$$

the Riemann invariants are

$$\begin{aligned} h_0 = \frac{a^2}{\gamma-1} + \frac{u^2}{2} &= constant \text{ on } \frac{dx}{dt} = u \\ p &= constant \text{ on } \frac{dx}{dt} = u\frac{M-1}{|M-1|} \\ s &= constant \text{ on } \frac{dx}{dt} = u \end{aligned} \tag{3–108}$$

The above analysis encourages a subsonic inflow boundary condition to specify entropy and total enthalpy from the free stream and extrapolate pressure from the interior. However, we have found that the most consistent boundary conditions incorporate a ghost cell and utilize the numerical flux function on the boundary faces.

3.7.2 Curvature-corrected Symmetry Technique

The preconditioned flux function is very sensitive to the boundary conditions applied. For the studies of this dissertation the curvature-corrected symmetry techniques of Dadone and Grossman [16] proved beneficial. Ghost cell boundary conditions are applied for all flow boundaries and the modified Roe flux extracts the appropriate upwind information from the surrounding data. Additionally with regards to limiters, boundary and interior cell interpolations are treated identically. Past researchers have stressed the importance of using a characteristic boundary condition in the sense of implicitly solving for the update state vector at the boundaries by using the preconditioning matrix in the left-hand-side mass matrix. The present calculations specify the free-stream state for far-field boundaries and the impermeability boundary approach of Dadone and Grossman [16].

The tangency condition specifies the symmetric state which enforces tangency at a cell face. The boundary interface nomenclature is defined in Fig. (43). Neglecting surface curvature, the symmetry technique valid for steady flows over planar surfaces, is

$$\begin{aligned} \rho_{1,i} &= \rho_1 \\ \bar{u}_{1,i} &= -\bar{u}_1 \\ \bar{v}_{1,i} &= \bar{v}_1 \\ p_{1,i} &= p_1 \end{aligned} \quad , \quad \text{and} \quad \begin{aligned} \rho_{2,i} &= \rho_2 \\ \bar{u}_{2,i} &= -\bar{u}_2 \\ \bar{v}_{2,i} &= \bar{v}_2 \\ p_{2,i} &= p_1 \end{aligned} \quad . \tag{3-109}$$

where

$$\begin{aligned} \bar{u} &= \hat{n}_x u + \hat{n}_y v \\ \bar{v} &= -\hat{n}_y u + \hat{n}_x v \end{aligned} \tag{3-110}$$

and the cell-face unit normal is $(\hat{n}_x, \hat{n}_y)$.

The curvature-corrected symmetry technique accounts for normal pressure gradients by satisfying the normal momentum equation and enforcing an impermeable boundary. The pressure is not symmetric on either side of the wall as is the case for a flat wall. With interpolated values at the wall of the velocity magnitude and

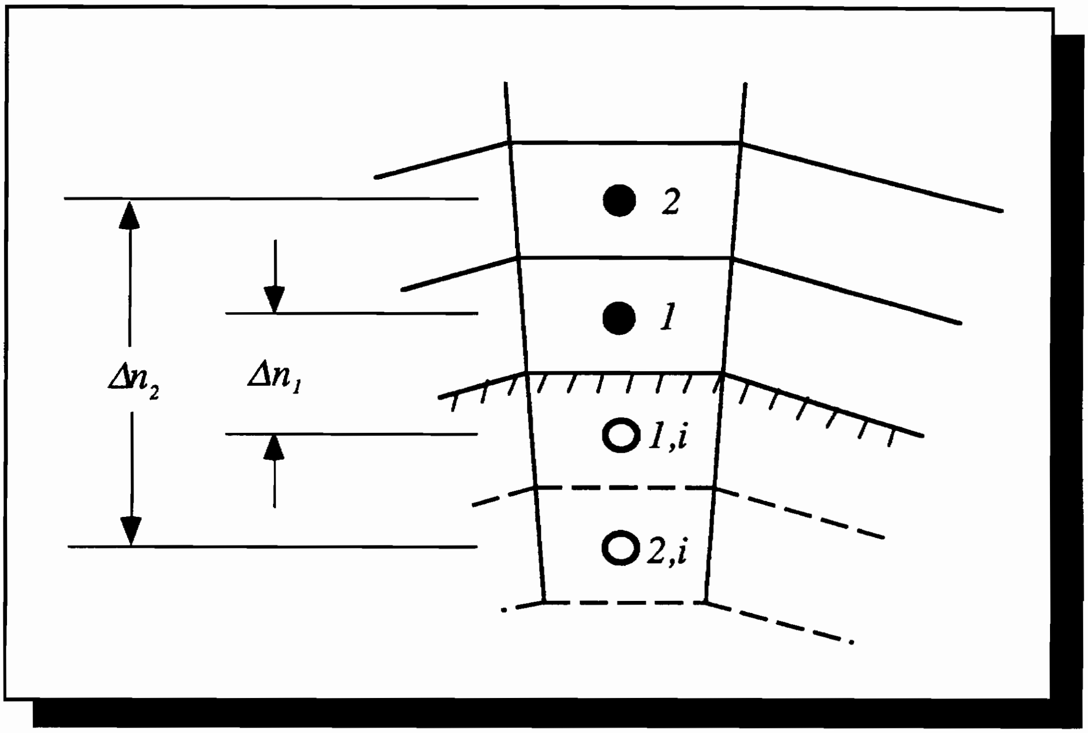

Figure 43. Nomenclature for curvature-corrected symmetry technique of Dadone and Grossman.

density, the pressure at the image cells is determined using the radius of curvature, R, and the mesh spacing, Δn_1 and Δn_2, as

$$\begin{aligned} p_{1,i} &= p_1 - \frac{\rho_w q_w^2}{R}\Delta n_1 \\ p_{2,i} &= p_2 - \frac{\rho_w q_w^2}{R}\Delta n_2 \end{aligned} \tag{3–111}$$

Assuming symmetry in entropy and total enthalpy and anti-symmetry in the contravariant velocity, then

$$\begin{aligned} s_{1,i} &= s_1 \\ (h_0)_{1,i} &= (h_0)_1 \, . \\ \bar{u}_{1,i} &= -\bar{u}_1 \end{aligned} \tag{3–112}$$

The stagnation enthalpy, pressure and density give the velocity magnitude and thus $\bar{v}_{1,i}$. The Cartesian velocities are then easily found through matrix transformation

$$\left\{ \begin{matrix} u \\ v \end{matrix} \right\}_{1,i} = \begin{bmatrix} \hat{n}_x & -\hat{n}_y \\ \hat{n}_y & \hat{n}_x \end{bmatrix} \left\{ \begin{matrix} \bar{u} \\ \bar{v} \end{matrix} \right\}_{1,i} \tag{3–113}$$

The second image cell is determined from symmetric entropy and stagnation enthalpy

$$\begin{aligned} s_{2,i} &= s_2 \\ (h_0)_{2,i} &= (h_0)_2 \end{aligned} \, . \tag{3–114}$$

However, the contravariant velocity, $\bar{u}$, is determined such that $\bar{u}_l\sqrt{\rho_l} + \bar{u}_r\sqrt{\rho_r} = 0$ at the cell face. With linear extrapolations of the left and right states, the velocity is

$$\bar{u}_{2,i} = 3\bar{u}_{1,i} + 2\left[\bar{u}_1 + (\bar{u}_1 - \bar{u}_2)/2\right]\sqrt{\frac{\rho_1 + (\rho_1 - \rho_2)/2}{\rho_{1,i} + (\rho_{1,i} - \rho_{2,i})/2}} \, . \tag{3–115}$$

The Cartesian velocities for cell $(2, i)$ are then easily determined with $\bar{u}_{2,i}$ and $q_{2,i}$.

CHAPTER
FOUR
STEADY-STATE RESULTS

4.1 Channel Flow

A simple geometry is used to demonstrate the abilities of the preconditioning procedure to accelerate convergence to a steady state. The transonic flow through a channel with a circular-arc disturbance on the lower wall was the subject of a GAMM workshop in 1979. The proceedings were then published in 1981 [47]. The transonic flow is slow to converge because the transonic shock which forms on the recompression side of the arc is reluctant to find its final steady state position.

Using this problem, we can determine the answers to many questions: How well does the preconditioning accelerate the convergence? What happens to convergence for different flux functions? What is the optimal Courant number to use with approximate factorization and, in particular, is this optimal value the same with and without preconditioning? What is the effective spectral radius of the iteration scheme? How does higher-order differencing change the performance of the preconditioned equations? How does the preconditioning perform across the entire subsonic Mach-number range? And, finally, how accurate are the computed incompressible solutions, which we can compare to the small-disturbance potential equation?

4.1.1 Numerical Synopsis

The circular arc is nestled in the lower wall and has unit chord. The inflow and outflow boundaries are two chords away from the leading and trailing edges, and the top wall is 2.073 chords above the lower wall. The curvature-corrected symmetry technique of Dadone and Grossman [16] is used as our tangency boundary condition on the upper and lower walls. At the inflow and outflow boundaries, the corresponding left and right states are set at the free-stream value. A numerical flux is calculated at all boundaries which alleviates boundary condition stiffness from the newly modified Roe flux function. We integrate in time using approximate factorization, and use both Van Leer flux-vector splitting and Roe flux-difference splitting.

Four different grids of dimension 31×16, 61×31, 91×61, and 181×91 are used. All pressure contour plots are given on the 61×31 mesh shown in Fig. (44). The Courant-number study below was done on the coarsest mesh to conserve excessive CPU time.

4.1.2 Courant-Number Study

The transonic flow solution has a free-stream Mach number of $M = 0.85$. Convergence of a flux balance residual to machine zero on a Cray YMP is obtained using MUSCL differencing ($\kappa = 1/3$) and the Van Albada limiter. The results for the range of time steps is shown in Fig. (45).

First, notice the typical behavior of the approximate factorization algorithm (*i.e.* gains that approach unity for both small and large Courant numbers and an optimal Courant number in the middle on the order of 10). The Courant number corresponding to the minimum number of iterations appears to be different for local time stepping than for characteristic time stepping; however, the definition of Courant number differs between the two. The lowest number of iterations required to solve the transonic flow using flux-difference splitting is 496 with local time

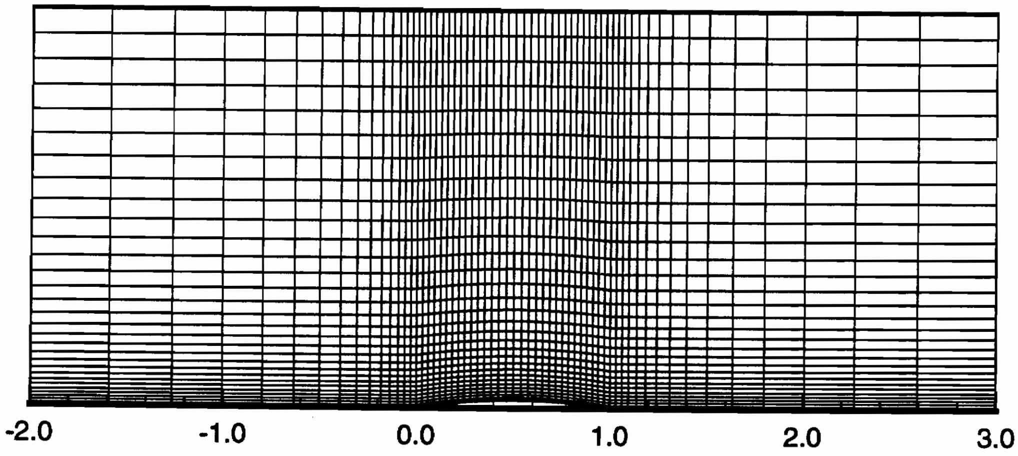

Figure 44. Typical mesh used to simulate the flow through a channel. All pressure contour plots are presented on this 61×31 grid.

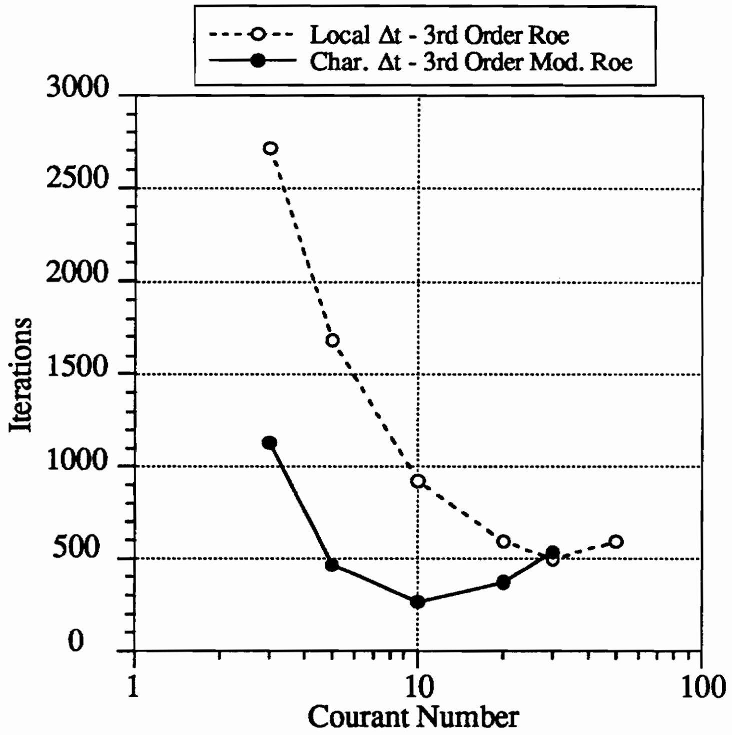

Figure 45. Courant-number study using approximate factorization and Roe flux-difference splitting for the transonic channel.

stepping and 263 with characteristic time stepping. The optimal Courant number for local time stepping is not necessarily the proper selection for characteristic time stepping. The first-order results are similar.

Pressure contours are shown in Fig. (46), and the lower wall distribution is shown in Fig. (47). The result from Rizzi's shock-fitting code [47] is included as a nearly exact solution. Careful examination will show that the peak pressure using the modified Roe algorithm is lower than for standard flux-difference splitting. This can be attributed to a larger artificial viscosity associated with the modified flux algorithm.

4.1.3 Numerical or Effective Spectral Radius

In this subsection, the residual histories of a grid-convergence cycle are used to determine the spectral radius of the numerical algorithm. After initial short wavelengths are smoothed, the residual history reflects the gain of the longest numerically achievable wavelength. The dominant long wave makes a residual history on a semi-log plot appear linear. The spectral radius for this one wave can be estimated strictly by examining the residual history of this final damping region. If the error at iteration number n is a fraction (*i.e.* the spectral radius) of the error at iteration number $n+1$, or if

$$\epsilon^{n+1} = \rho_s \, \epsilon^n, \qquad (4-1)$$

then the spectral radius over several iterations numbered n to N is

$$\rho_s = [\frac{||R^N||_2}{||R^n||_2}].^{\frac{1}{N-n}}. \qquad (4-2)$$

A typical residual history for the transonic channel flow is shown in Fig. (48)., and the corresponding spectral radii over several mesh refinements are shown in Fig. (49). As the grid spacing becomes finer, the long waves are linearly less damped,

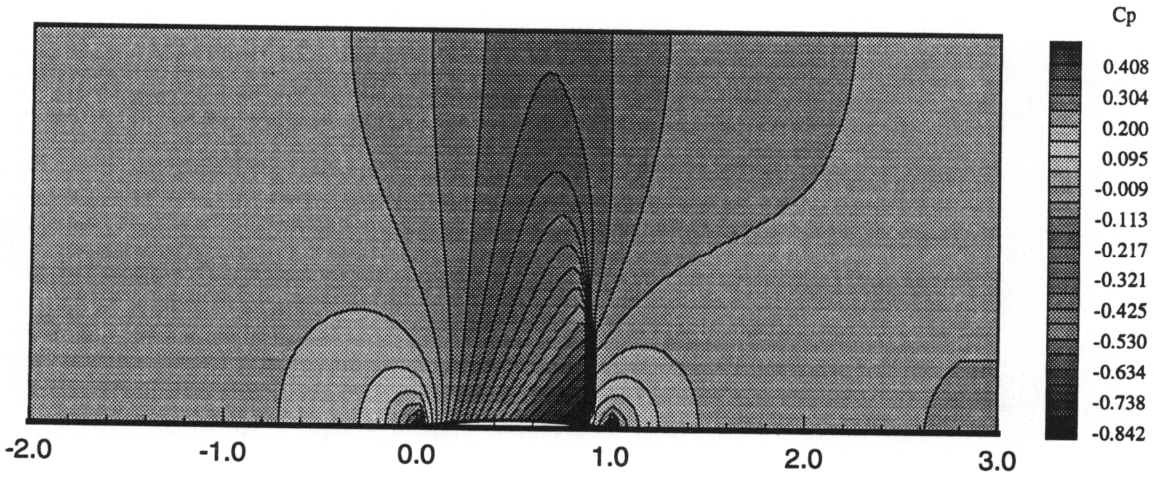

Figure 46. Pressure coefficient contours for the transonic $M = 0.85$ solution.

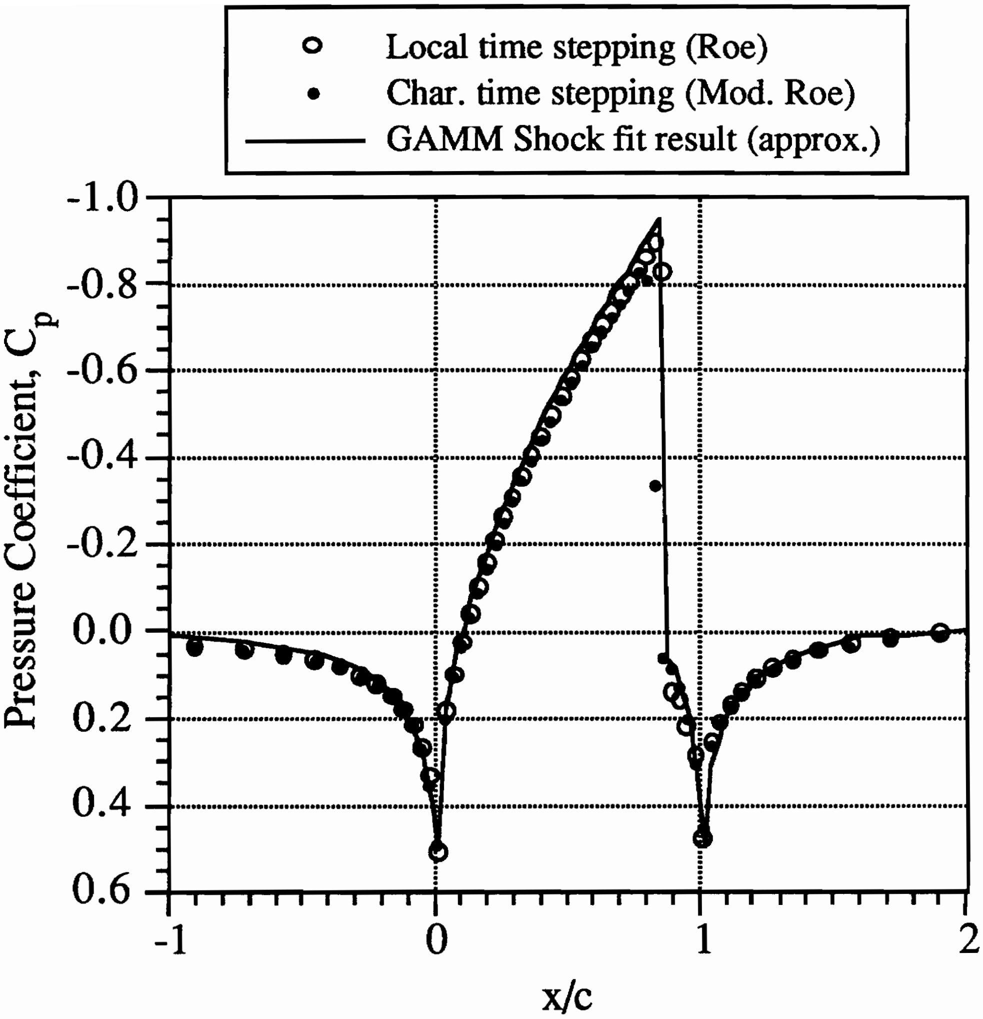

Figure 47. Lower wall pressure coefficient for the transonic flow through a channel compared to the shock-fit results of Rizzi [47].

and consequently iterations and computational time increase. Note that the spectral radius for the characteristic time-stepping algorithm is lower for all mesh sizes in comparison to local time stepping. This indicates that a solution using preconditioning may be obtained on a grid with roughly twice as many points in each direction in the same number of iterations (*i.e.* same spectral radius).

4.1.4 Mach-Number Study

In this subsection we investigate the performance of the preconditioning algorithm over a range of subsonic Mach numbers. Performance for preconditioning and local time stepping using Roe's scheme is shown for first and higher order in Fig. (50).

The preconditioning algorithm gives convergence in fewer iterations for all free-stream Mach numbers. We have successfully run at Mach numbers as low as $M = 10^{-8}$ while the solution using the standard Roe algorithm with local time stepping was entirely inaccurate, as will be demonstrated in the next subsection.

Performance for the flux-vector splitting scheme is shown for first and higher order in Fig. (51). The mass splitting technique poorly models the Riemann problem for low Mach numbers and thus mathematically confuses the wave-based preconditioning matrix. Consequently, preconditioning with flux-vector splitting at low Mach numbers is unwise. The cross-over Mach number appears to be around $M \approx 0.5$.

To determine the result of preconditioning with higher-order interpolation, we calculate the percent savings over the Mach-number range for different accuracies and compare with the theoretical savings. Results are shown in Fig. (52). The savings were calculated as the difference between the required number of iterations divided by the iterations required without preconditioning. Preconditioning with higher-order interpolation is more efficient simply because the partial differential equations are modeled more accurately. Previous researchers have hypothesized

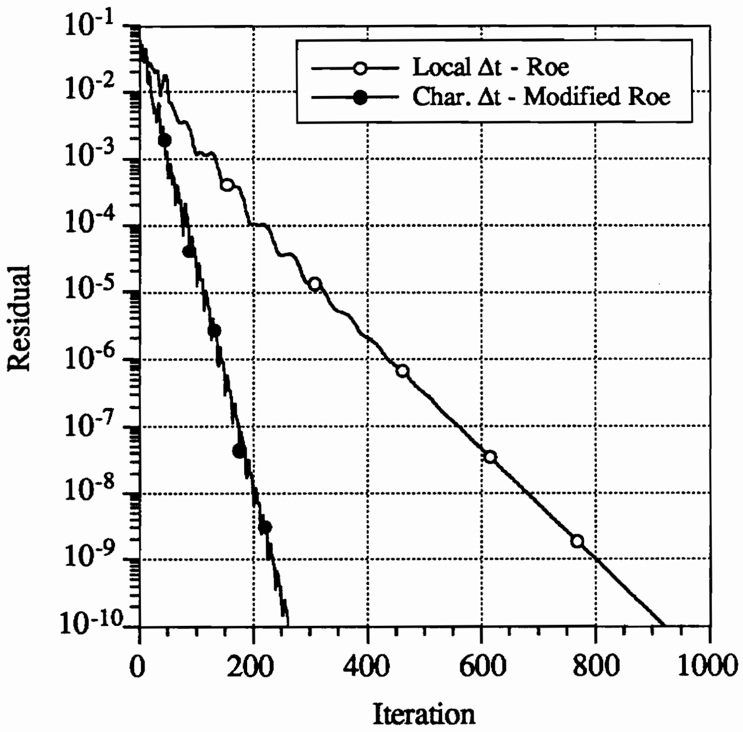

Figure 48. Residual history using approximate factorization with Roe flux-difference splitting for a transonic channel.

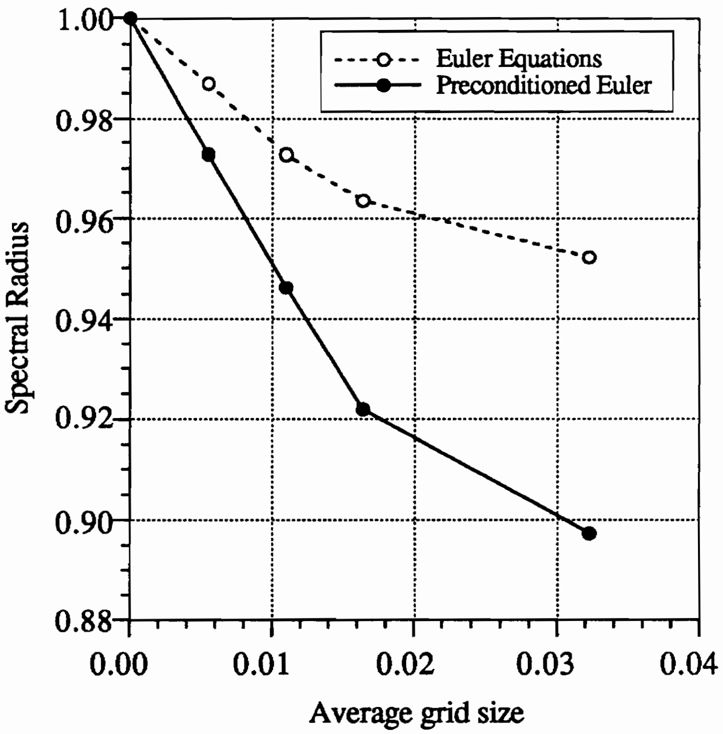

Figure 49. Spectral radius using approximate factorization with Roe flux-difference splitting for a transonic channel. The average grid size is one over the stream-wise grid dimension.

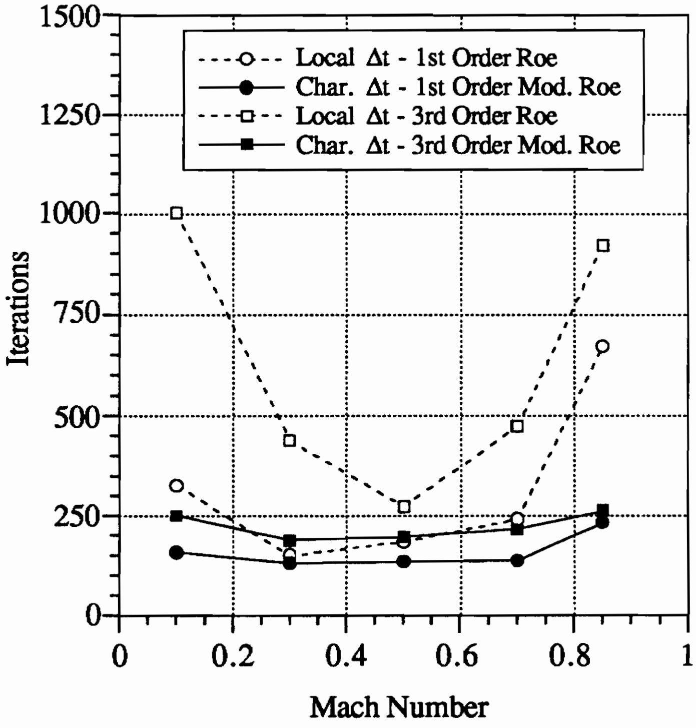

Figure 50. Convergence profile over subsonic Mach-number range using first and third-order MUSCL differencing with Roe flux-difference splitting.

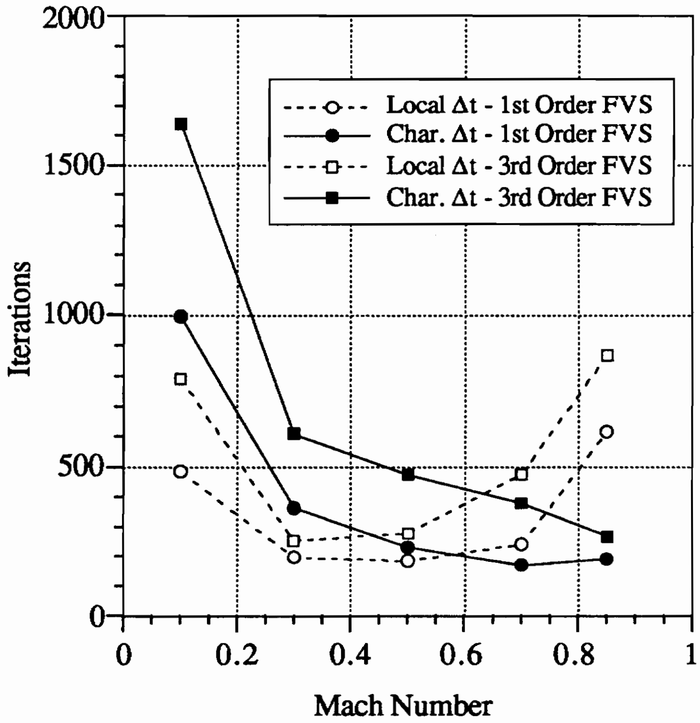

Figure 51. Convergence profile over subsonic Mach-number range using first and third-order MUSCL differencing with Van Leer flux-vector splitting.

that higher-order spatial accuracy would yield better percent savings compared to the first-order method. This dissertation confirms for the first time that the preconditioning *does* accelerate the damping of the long waves which are better represented with higher-order differencing.

4.1.5 Very-Low-Speed ($M=10^{-3}$) Flow

An essential consideration for low Mach-number flows is accurate pressure interpolation. The ratio of the convective to pressure terms is $1/(\gamma M^2)$. This ratio approaches infinity at low Mach numbers. To accurately balance the flux of momentum, the pressure must be scaled to be on the order of the convective terms. Past researchers have incorporated a gauge pressure referenced to the free-stream pressure. Interpolation and flux evaluation is done with this small pressure perturbation. Summing the gauge pressure and free-stream pressure recovers the absolute pressure through

$$p = p_\infty + p' \tag{4–3}$$

We have successfully utilized this strategy modeling very-low-speed flows. Note that to be perfectly consistent, we must scale the internal energy as suggested by Feng and Merkle [19]. For a perfect gas this is simply

$$\rho e = (\rho e)_\infty + (\rho e)', \tag{4–4}$$

where

$$(\rho e)_\infty = \frac{p_\infty}{\gamma - 1}. \tag{4–5}$$

However, for a flow code which models both perfect-gas and finite-rate chemistry, a gauge energy is impractical and was not done in the present computations. The consequence of this is seen in the residual history shown in Fig. (53). For very small residuals the energy balance exceeds the machine's precision. Increasing the machine precision or using a gauge energy allows convergence to higher orders.

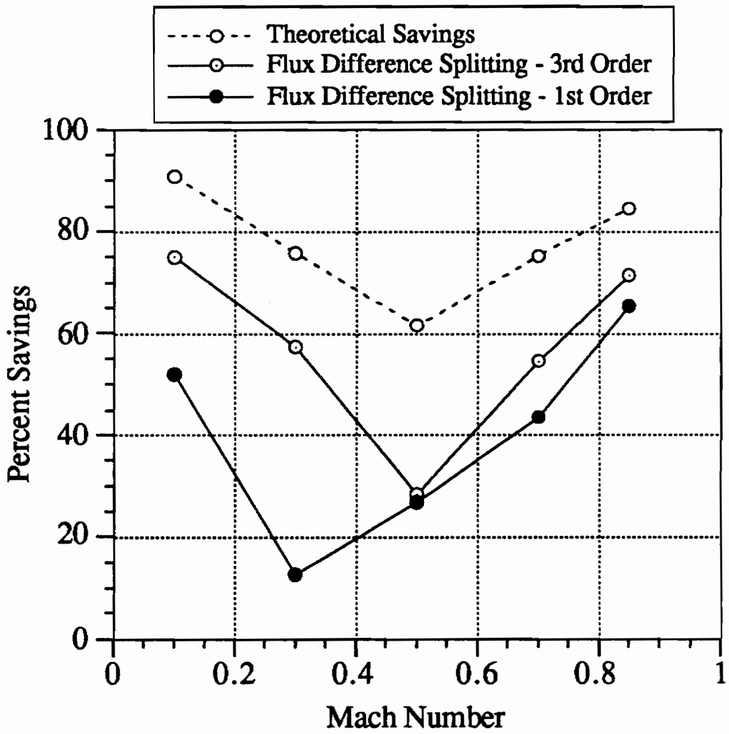

Figure 52. Percent savings using preconditioning over subsonic Mach-number range using Roe flux-difference splitting.

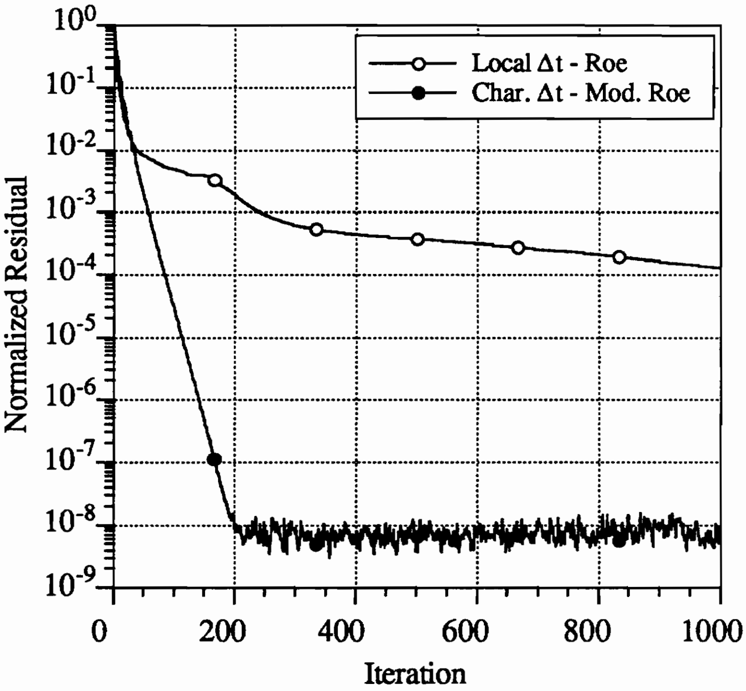

Figure 53. Very-low-speed channel flow residual history using approximate factorization and Roe flux-difference splitting.

The same channel geometry was run with a free-stream Mach number of $M = 0.001$. The pressure coefficient on the lower wall is compared to the small-disturbance solution in Fig. (54). Thirty levels of pressure coefficient contours are shown in Fig. (55). Notice the symmetry in the data and in the contours stemming from the curvature-corrected symmetry boundary technique. This dissertation shows the first successfully accurate simulation at such a low Mach number. A comparable incompressible result calculated with a compressible code has never before been obtained.

An alternative solution using preconditioning with a standard Roe scheme is shown in Fig. (56). Obviously the modified Roe flux is essential for reasons of both accuracy *and* stability. Recall from the discussion in Section 2.1.3, that the uncertainty of a solution to a high-condition number system may be large even with a small residual. This is clearly seen in the contours of Fig. (56). The culprit is the singular dissipation matrix in the standard Roe flux function.

4.2 Eppler 387 Airfoil

An Eppler 387 airfoil is simulated at a free-stream Mach number of $M = 0.05$ and angle of attack, $\alpha = 2°$. There is little hope of accurately and efficiently calculating this inviscid flow-field with a standard compressible Euler code without including the preconditioning technique. The computation using preconditioning is compared to the experiment performed by McGhee [42] at NASA Langley Research Center in the low-turbulence pressure tunnel (LTPT). Their experiment was performed at a free-stream Reynolds number of $Re_c = 60,000$. A predominant phenomenon of the low-Reynolds number flow is the laminar separation bubble over most of the upper surface.

This case typifies where we are with preconditioning and where we want to go. At present, we cannot simulate the complete Navier-Stokes equations and the laminar separation bubble using preconditioning. The reasons will be discussed in

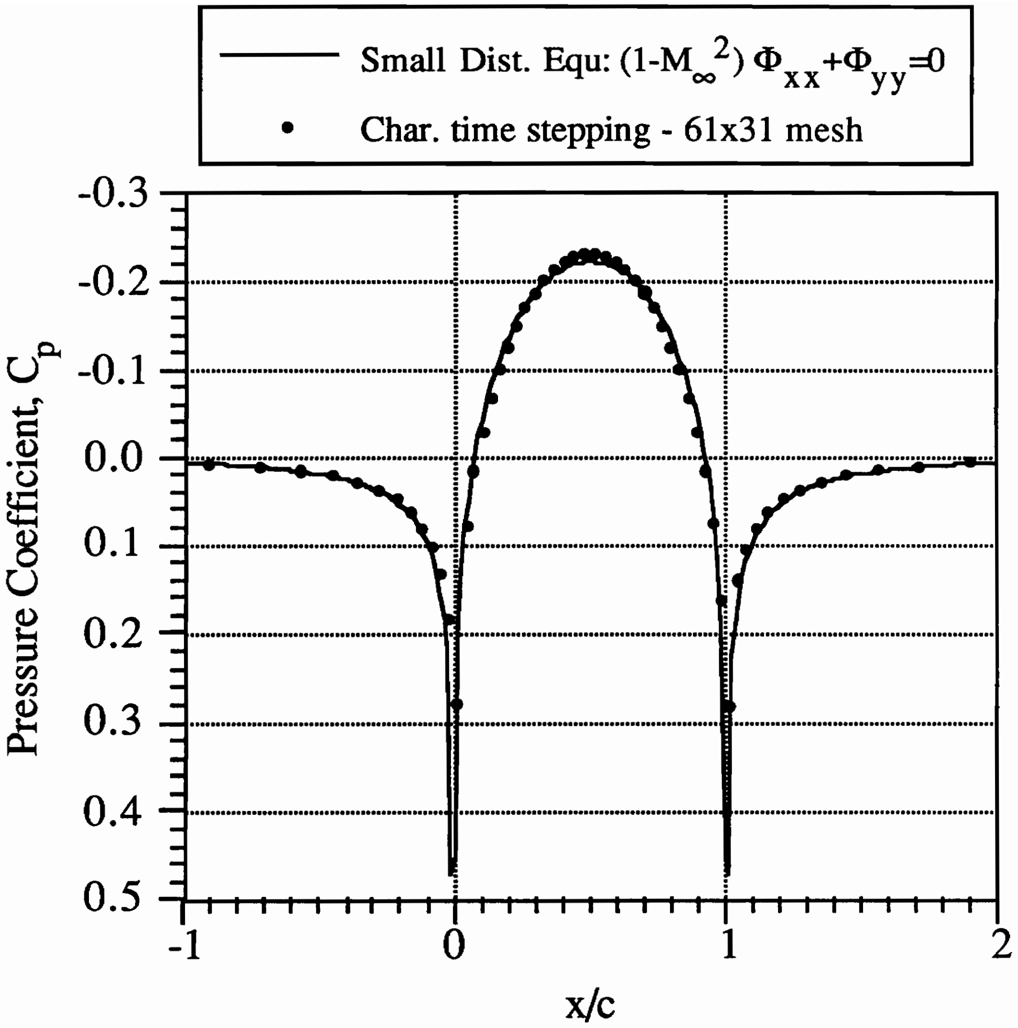

Figure 54. Pressure coefficient for the very-low-speed flow over a circular-arc airfoil.

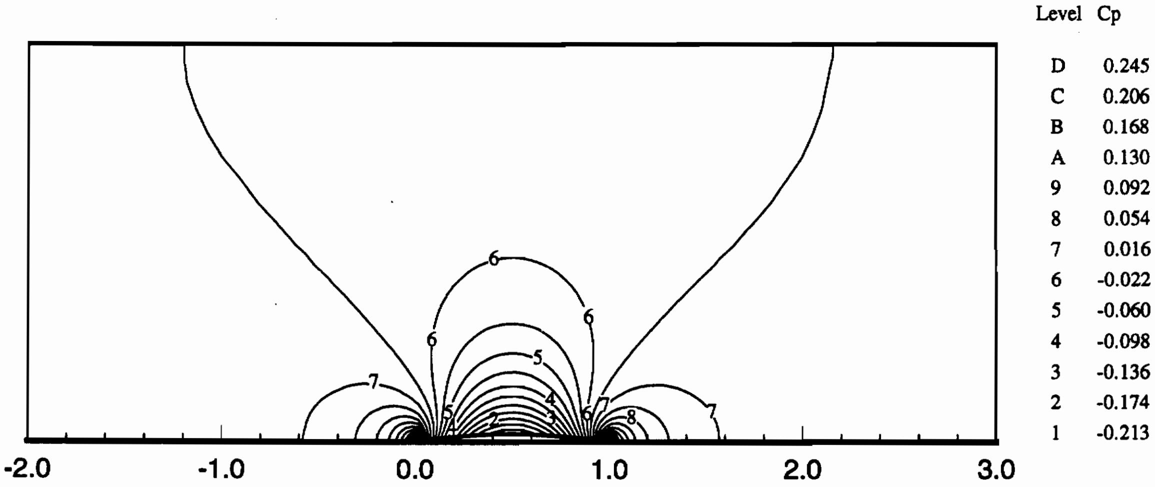

Figure 55. Pressure coefficient contours for very-low-speed flow through a channel using preconditioning and a *modified* Roe flux.

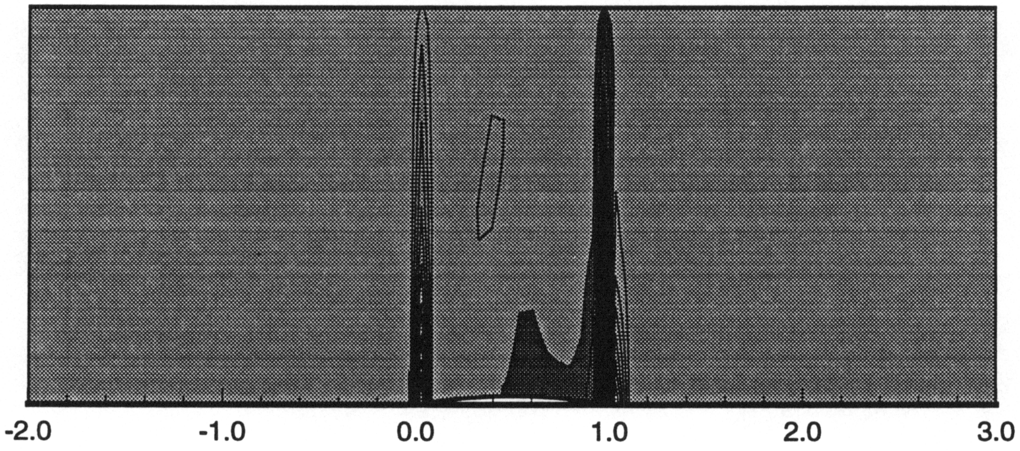

Figure 56. Pressure coefficient contours for very-low-speed flow through a channel using preconditioning and a *standard* Roe flux.

Chapter 5 as well as a possible solution which is effective for one-dimensional Navier-Stokes equations. In fact, obtaining accurate experimental data for this extremely sensitive flow-field is difficult also. Experimental data varies among different facilities because of the data's sensitivity to free-stream turbulence and model surface roughness. Additionally, the model forces and pressure differences from the upper and lower surface are difficult to accurately measure. CFD can make great strides to contribute to the understanding of laminar separation and low Reynolds number flows.

The inviscid computation with preconditioning used the 61×41 mesh shown in Fig. (57). Convergence of the normalized flux residual to a tolerance of six orders was obtained in 708 iterations and 81 CPU seconds. We iterated using the mesh sequencing capabilities of GASP and three levels of grid refinement. The pressure coefficient contours of the flow field are shown in Fig. (58), and the surface pressure is compared to McGhee's wind tunnel data [42] in Fig. (59). To verify our inviscid results, we compared to the results from Drela's panel-method code [15]. The panel code also performs an integral boundary-layer calculation with the pressure field as an input. The boundary-layer solution is included for comparison and as a validation of this simpler method.

4.3 Space Marching

4.3.1 Three-dimensional Wedge

As a three-dimensional test case for the flux function and preconditioning matrix we simulate a skewed wedge that was proposed by Rumsey, *et. al.* [52] for their five-wave multi-dimensional Riemann solver. The inflow Mach number is 2.8, and the geometry with pressure contours is shown in Fig. (60). The number of iterations required to converge an absolute residual by six orders per cross-flow plane is shown in Fig. (61). Approximate factorization is used in the cross flow plane with

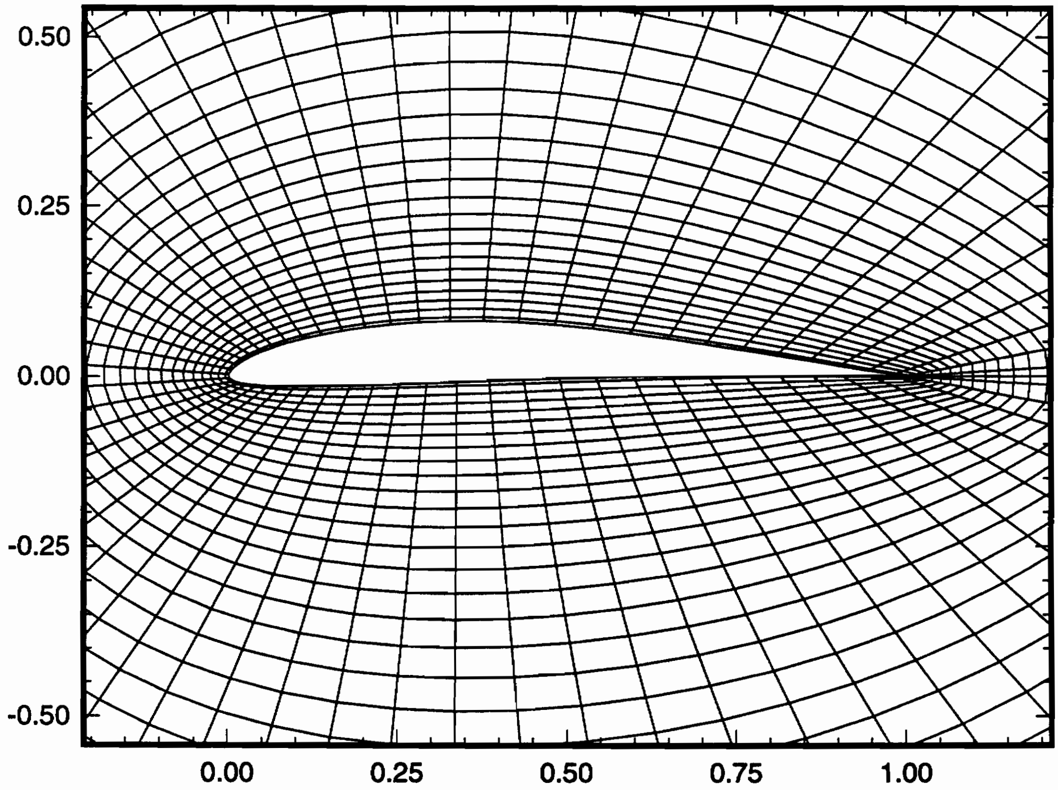

Figure 57. Close-up of the 61 × 41 O-mesh surrounding the Eppler 387 airfoil.

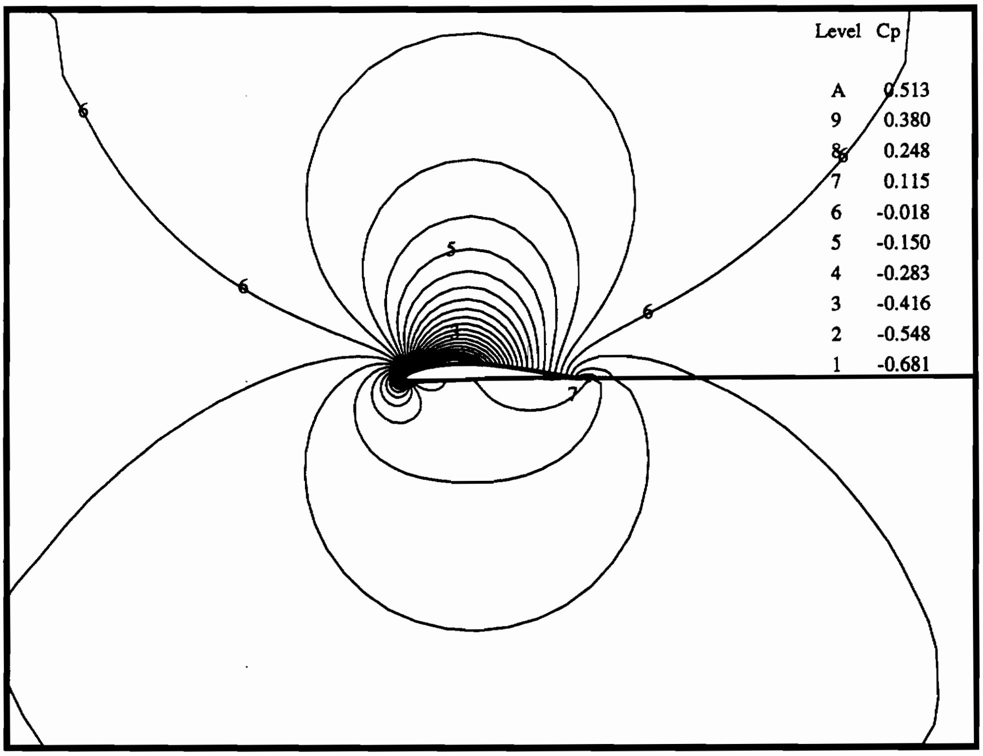

Figure 58. Pressure coefficient contours surrounding the Eppler 387 airfoil.

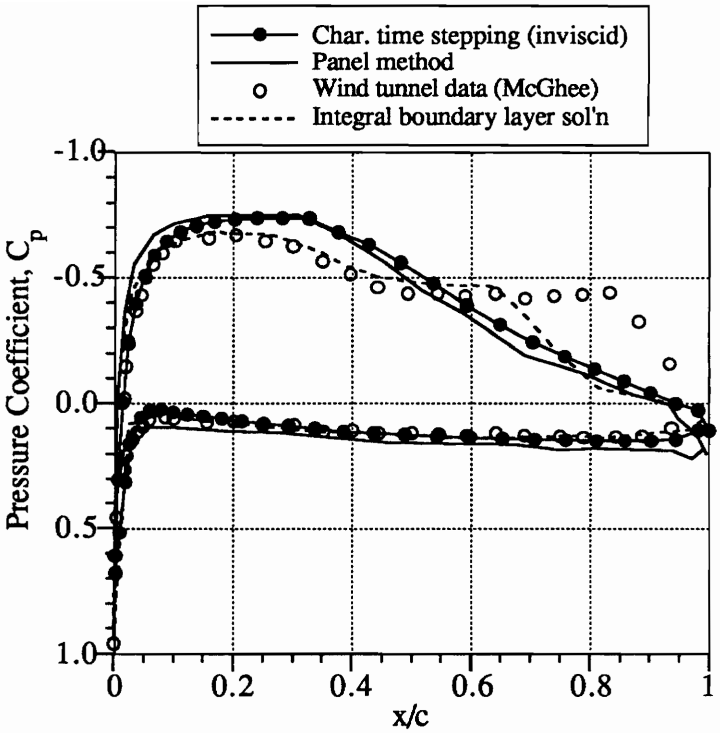

Figure 59. Pressure coefficient on the Eppler 387 airfoil surface using preconditioned Euler, experimental and potential methods.

a non-linear residual update in the marching direction and a Courant number of five. Based on the condition numbers for three-dimensional $M=2.8$ flow, we would expect 1.97 times better convergence rate.

In practice, the preconditioned case converges 0.93 to 1.33 times faster than without preconditioning depending upon the cross-flow plane. So, for a few cross-flow planes the preconditioning is actually worse. The explanation is simple. The equations modeled in a marching calculation are only a subset of the full partial differential equations. A global iteration would represent a more fair comparison and, most probably (based on the experience of past test cases), convergence in fewer iterations.

4.3.2 Axi-symmetric Nozzle

The second marching case is the axi-symmetric reacting flow through a diverging nozzle. This case helps to determine the effect of preconditioning for flow with equilibrium and finite-rate chemistry as well as non-equilibrium vibrational relaxation. Four cases are run with different assumptions upon the thermodynamics. All cases use a Courant number of ten.

The inflow temperature is a $9000\,K$ and the inflow velocity is $3000\,m/s$ yielding an approximately sonic inflow Mach number. We expect large benefits from preconditioning within this sonic flow regime. Perfect gas, equilibrium, and finite-rate chemistry with and without non-equilibrium vibration were run using the Kang *et.al.* [37] five-species air chemistry model. The Mach number, temperature and mass fraction of diatomic nitrogen are shown in Figs. (62), (63) and (64). Iterations per plane are shown for the above cases in Figs. (65), (66), (67) and (68). As the supersonic flow accelerates the condition number disparity becomes less, and the benefits of preconditioning decrease. Convergence rate is accelerated for all cases at the nearly sonic inflow condition.

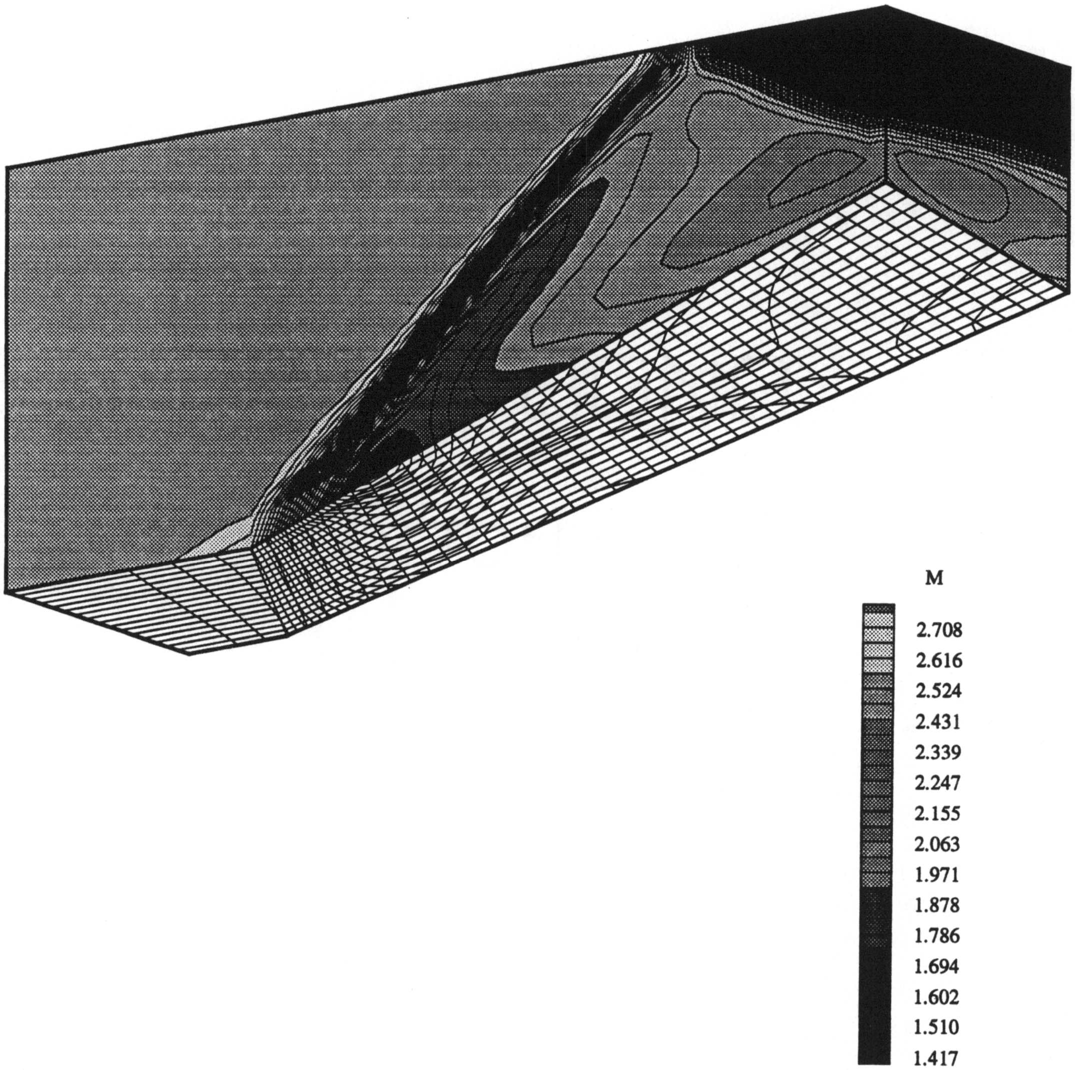

Figure 60. Flooded and line contours of pressure and a representative section of the mesh for the three-dimensional wedge with inflow $M = 2.8$. Flooded contours are shown on the left wall and the outflow plane. Line contours and the mesh rest upon the compression ramp.

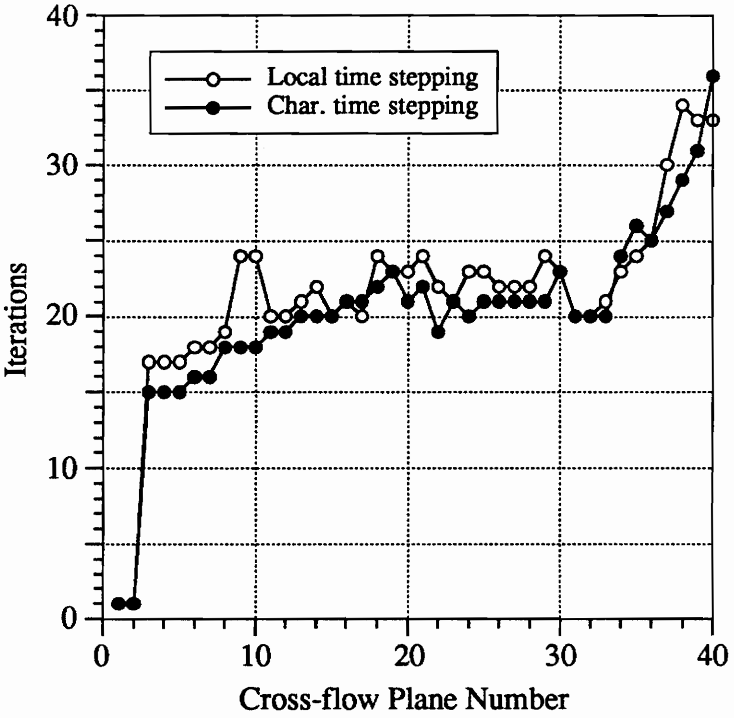

Figure 61. Iterations per plane for three-dimensional wedge with inflow Mach number of $M = 2.8$ and approximate factorization in the cross flow plane.

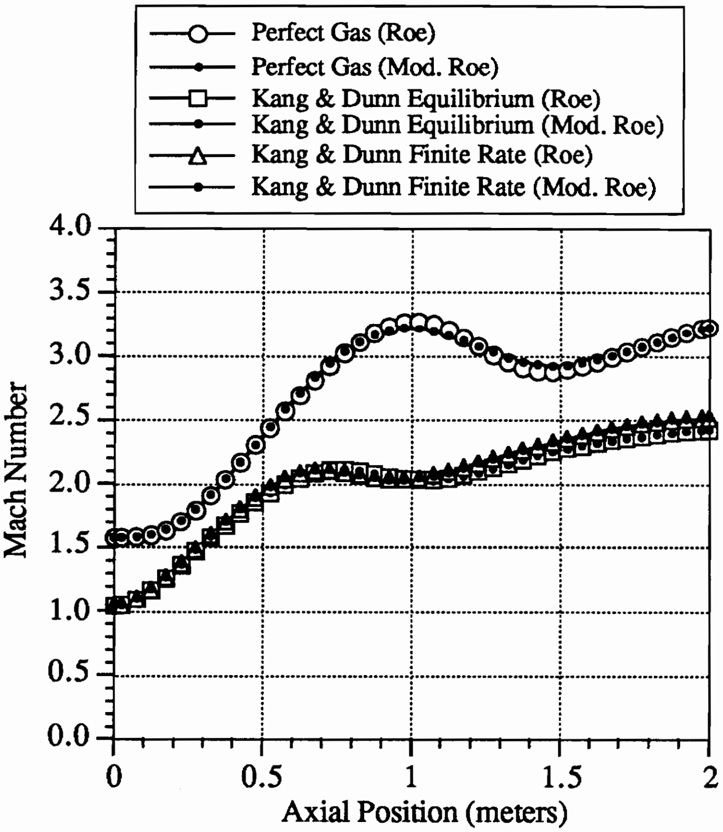

Figure 62. Stream-wise Mach-number distribution on the centerline of the axisymmetric nozzle.

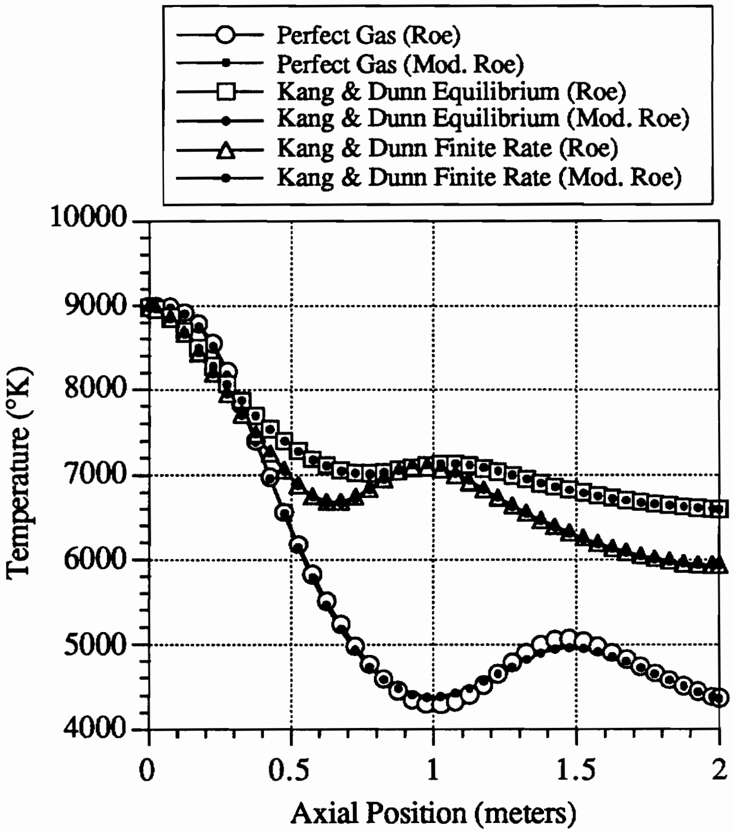

Figure 63. Stream-wise temperature distribution on the centerline of the axisymmetric nozzle.

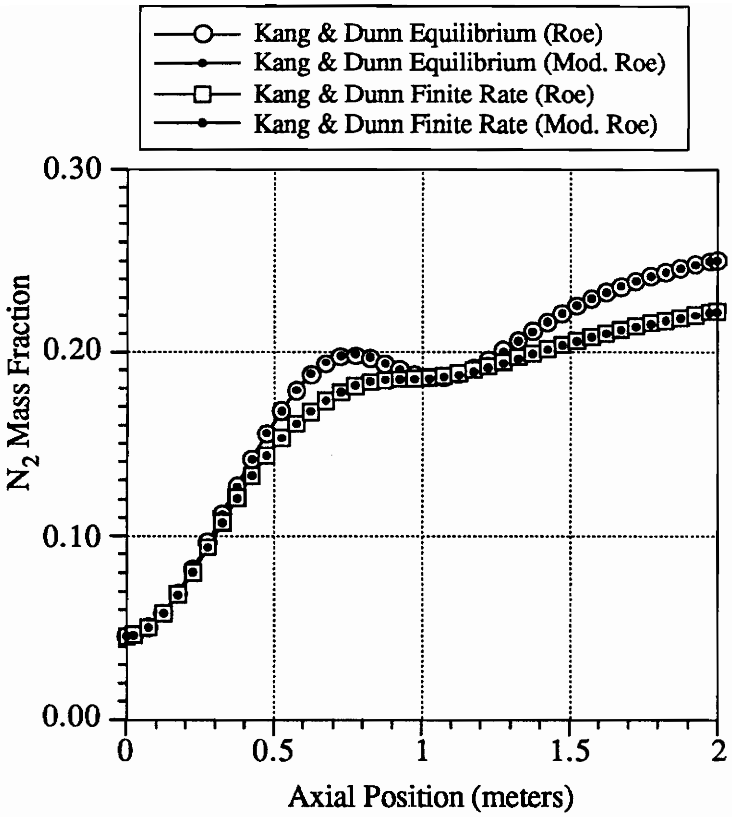

Figure 64. Stream-wise diatomic nitrogen mass-fraction distribution on the centerline of the axi-symmetric nozzle.

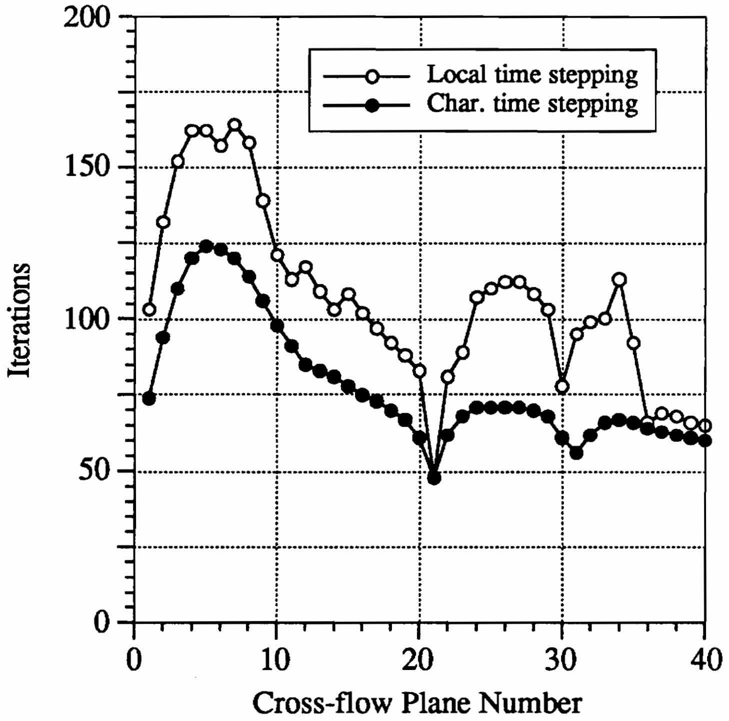

Figure 65. Iterations per plane for reacting-nozzle case assuming air as a perfect gas.

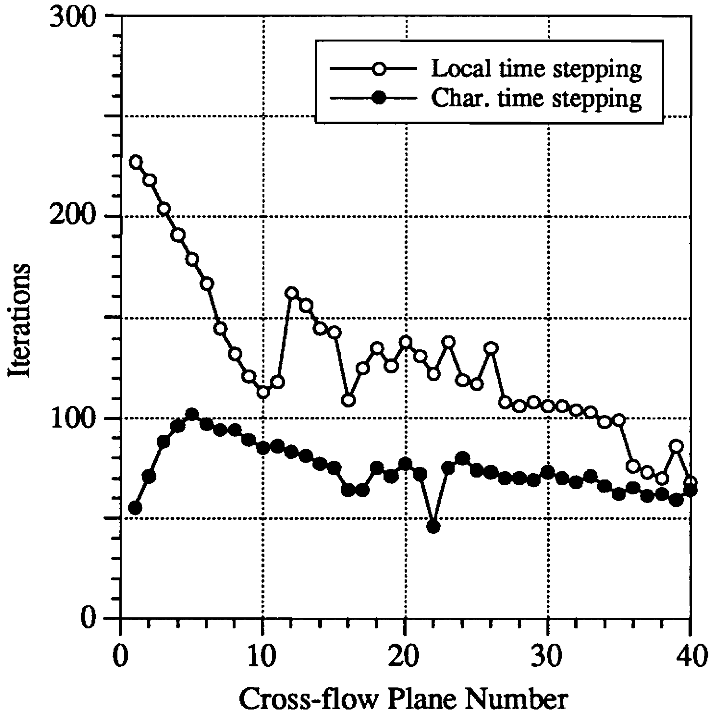

Figure 66. Iterations per plane for diverging nozzle with a real gas assuming five-species equilibrium flow.

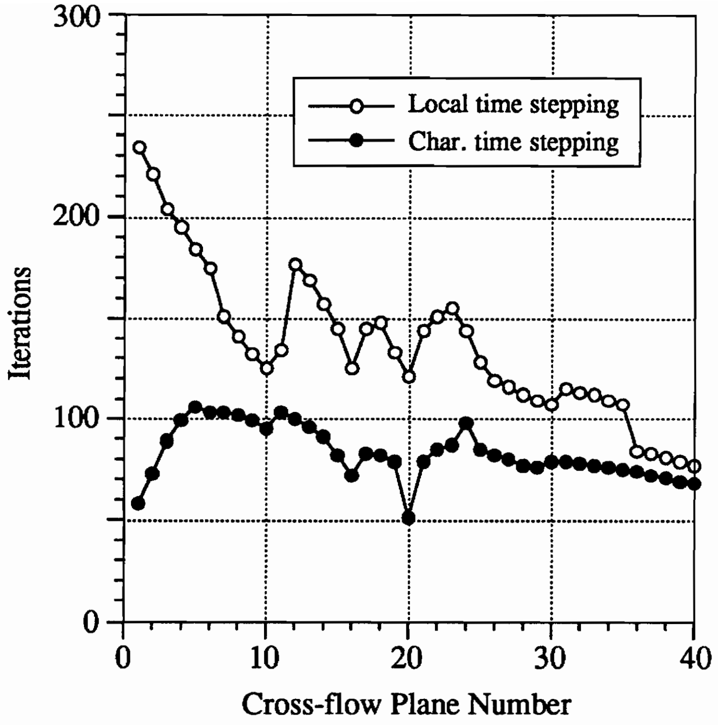

Figure 67. Iterations per plane for reacting-nozzle case assuming finite-rate reactions with species in vibrational equilibrium.

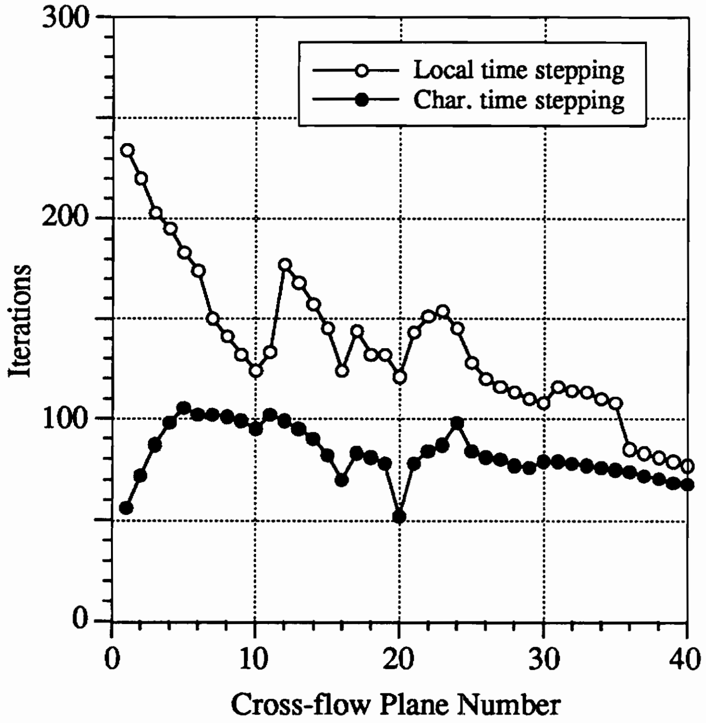

Figure 68. Iterations per plane for reacting-nozzle case assuming finite-rate reactions with three non-equilibrium vibration species.

4.4 Tabular Summary

The theoretical and actual convergence-rate acceleration for the above test cases are summarized in Table (10). The N/A indicates test cases that would give poor solutions after a large number of iterations. The theoretical speed-up for the nozzle case is generous because stiffness from finite-rate reactions is neglected. The actual speed-up is actually somewhat lower since the reacting nozzle flow is a marching calculation.

Table 10. Theoretical and actual convergence-rate acceleration for the presented test cases. N/A indicates test cases that were impractical without preconditioning.

Test Case	Mach No.	Theory	Actual
V.L.S. Channel	0.001	1001	N/A
T. Channel	0.85	6.50	3.56
Eppler 387	0.05	21.0	N/A
3-D Ramp	2.80	1.97	0.92-1.33
Nozzle	1.046	44.5	4.27

CHAPTER

FIVE

VISCOUS PRECONDITIONING

The idea behind characteristic time stepping is to transform as closely as possible the behavior of a system of equations into that of a scalar equation. The progress of this transition can be visualized with the Fourier footprint. The Fourier footprint reflects the stability of a first-order ordinary differential equation and thus any positive real part causes disturbances to grow exponentially.

The footprint results from determining the eigenvalues of the Fourier transform of a discrete time step, Δt, times the semi-discrete form of the residual. For the first-order scalar wave equation, the Fourier footprint is a circle with a Courant-number radius resting in the negative real half of the complex plane. The final criterion for numerical stability is set down by the method of time integration. The footprint and the characteristic polynomial of the time evolution scheme determines which waves are most effectively damped [65].

The one-dimensional preconditioned Euler equations share the same Fourier footprint as the scalar wave equation. However, if we include the viscous terms which comprise the Navier-Stokes equations, the Fourier footprint becomes prohibitively skewed as the cell Reynolds number approaches zero, or

$$Re_{\Delta x} \equiv \frac{\rho u \Delta x}{\mu} \rightarrow 0. \tag{5-1}$$

This footprint with the inviscid preconditioning is shown in Fig. (69). Note that the axes are not one-to-one and that the footprints for high Reynolds numbers are nearly circular. The conclusion is that the convective and acoustic waves in the low Reynolds number viscous limit cause more stiffness than the original equations.

What do we want the footprint to look like? The one-dimensional heat equation yields a footprint which lies on the negative real axis with no imaginary part. The footprint is bounded by zero and twice the Courant number. We would hope that as the cell Reynolds number approaches the viscous limit, the Fourier footprint would approach the footprint of the heat equation for all three waves in the system. To accomplish this, we must derive a new preconditioning matrix that is designed for the task.

To determine the viscous preconditioning matrix, we adopt the symmetrization variables proposed by Abarbanel, *et.al.* [1]. These variables symmetrize both the inviscid and viscous flux Jacobians, which is beneficial because we can then determine a positive definite preconditioner. These variables are also known in differential form as

$$dZ \equiv \left\{ \begin{matrix} \frac{a}{\sqrt{\gamma}} \frac{d\rho}{\rho} \\ du \\ \frac{a}{\sqrt{\gamma(\gamma-1)}} \frac{dT}{T} \end{matrix} \right\}. \tag{5-2}$$

The non-conservative form of the Navier-Stokes equations are then

$$\frac{\partial Z}{\partial t} = -\left(A\frac{\partial Z}{\partial x} + A_v \frac{\partial^2 Z}{\partial x^2} \right), \tag{5-3}$$

where

$$A = \begin{bmatrix} u & a\frac{1}{\sqrt{\gamma}} & 0 \\ a\frac{1}{\sqrt{\gamma}} & u & a\sqrt{\frac{\gamma-1}{\gamma}} \\ 0 & a\sqrt{\frac{\gamma-1}{\gamma}} & u \end{bmatrix}, \tag{5-4}$$

and

$$A_v = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{4}{3}\nu & 0 \\ 0 & 0 & \frac{\gamma}{Pr}\nu \end{bmatrix}. \tag{5-5}$$

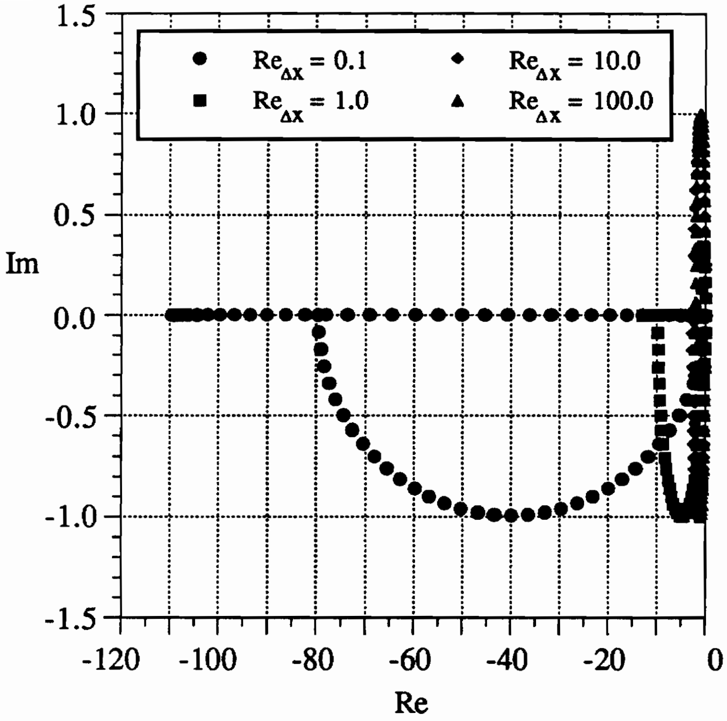

Figure 69. Fourier footprint using the inviscid preconditioning on the one-dimensional Navier-Stokes equations with unit CFL, $\nu = 1$.

For convenience we define matrices, $\tilde{A}$ and $\tilde{A}_v$, which are non-dimensionalized by the velocity and kinematic viscosity, u and ν, respectively. These matrices are

$$\tilde{A} \equiv \frac{1}{u} A = \begin{bmatrix} 1 & \frac{1}{M}\frac{1}{\sqrt{\gamma}} & 0 \\ \frac{1}{M}\frac{1}{\sqrt{\gamma}} & 1 & \frac{1}{M}\frac{\sqrt{\gamma-1}}{\sqrt{\gamma}} \\ 0 & \frac{1}{M}\frac{\sqrt{\gamma-1}}{\sqrt{\gamma}} & 1 \end{bmatrix}, \tag{5–6}$$

and

$$\tilde{A}_v \equiv \frac{1}{\nu} A_v = diag\,\{0,\ 4/3,\ \gamma/Pr\}\,. \tag{5–7}$$

The Fourier footprint of the preconditioned Navier-Stokes equations with Roe's approximate Riemann solver and second order central differencing is

$$\begin{aligned} \mathcal{F}\left\{\Delta t\, P\frac{\partial Z}{\partial t}\right\} = &- i\,\nu_x\, P\tilde{A}\,\sin\beta_x \\ &- 2\nu_x\, P|\tilde{A}|\,\sin^2(\beta_x/2) \\ &- \frac{4\nu_x}{Re_{\Delta x}} P\tilde{A}_v\,\sin^2(\beta_x/2). \end{aligned} \tag{5–8}$$

This can be rearranged by separating the real and imaginary parts to give

$$\begin{aligned} \mathcal{F}\left\{\Delta t\, P\frac{\partial Z}{\partial t}\right\} = &- 2\nu_x\, P\left(|\tilde{A}| + \frac{2\tilde{A}_v}{Re_{\Delta x}}\right)\sin^2(\beta_x/2) \\ &- i\,\nu_x\, P\tilde{A}\sin\beta_x. \end{aligned} \tag{5–9}$$

To control the growing negative real part, we must choose the viscous preconditioning matrix to be

$$P = \left(|\tilde{A}| + \frac{2\tilde{A}_v}{Re_{\Delta x}}\right)^{-1}. \tag{5–10}$$

With this formulation, we recover the inviscid preconditioner in the inviscid limit of $Re_{\Delta x} \rightarrow \infty$. The Fourier footprint for this preconditioning matrix is shown for Mach numbers of $M = 0.1$, 0.5, 0.9 and 2.0 in Figs. (70), (71), (72) and (73), respectively, for several representative cell Reynolds numbers. Notice that as the cell Reynolds number becomes large, the footprint becomes circular. Also, notice the controlled behavior for decreasing cell Reynolds numbers. All three waves are slowly collapsing onto the negative real axis as totally dissipative waves. Finally, notice that the high-frequency acoustic and entropy waves are forced to meet at the real axis.

As an initial plausibility test case, we consider the simple damping of a 5% pressure disturbance in the center of a $M = 0.1$ flow field. A pressure disturbance couples both the linear convective and the non-linear acoustic waves. We use MUSCL differencing with Roe's approximate Riemann solver and second-order central differencing for the viscous fluxes and investigate the convergence rate at several cell Reynolds numbers. Since Euler explicit time integration with $\kappa = 1/3$ is unstable for some waves, we use the optimal 3-stage scheme of Van Leer, *et.al.* [65]. The recommended stage weights are $\alpha_1 = 0.2884$, $\alpha_2 = 0.5010$ and $\alpha_3 = 1.0$ with a Courant number of $\nu = 1.3254$.

We compare the proposed viscous preconditioning with the inviscid preconditioning of Eqn. (3–35) with $\tau = 1$ and include the standard scheme without preconditioning. The results are shown in Fig. (74). In the inviscid limit, the inviscid and viscous preconditioners converge the normalized residual to a tolerance of 10 orders in 30 iterations. Without preconditioning convergence is met in 195 iterations. At $M = 0.1$, the expected convergence rate acceleration for the one-dimensional Euler equations is 11.

Notice how the performance of the inviscid preconditioner deteriorates as the cell Reynolds number decreases. The threshold is around $Re_{\Delta x} \approx 2$. We have observed the same behavior for the two-dimensional preconditioning. The viscous preconditioning holds true to form for all cell Reynolds numbers and converges the lowest Reynolds number case in 572 iterations. This is compared to 5767 iterations without preconditioning. Running the inviscid preconditioning at the low end was impractical. We estimate convergence in more than 100,000 iterations for $Re_{\Delta x} = 0.01$.

The problem with preconditioning the viscous equations with the inviscid preconditioning matrix in Eqn. (3-35) was first discovered by solving the thin-layer

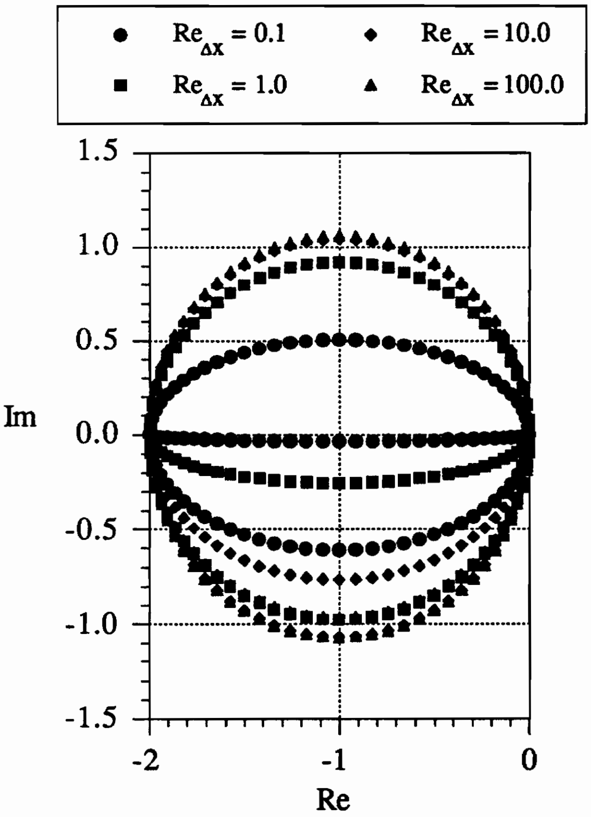

Figure 70. Fourier footprint using viscous preconditioning on the one-dimensional Navier-Stokes equations at $M=0.1$.

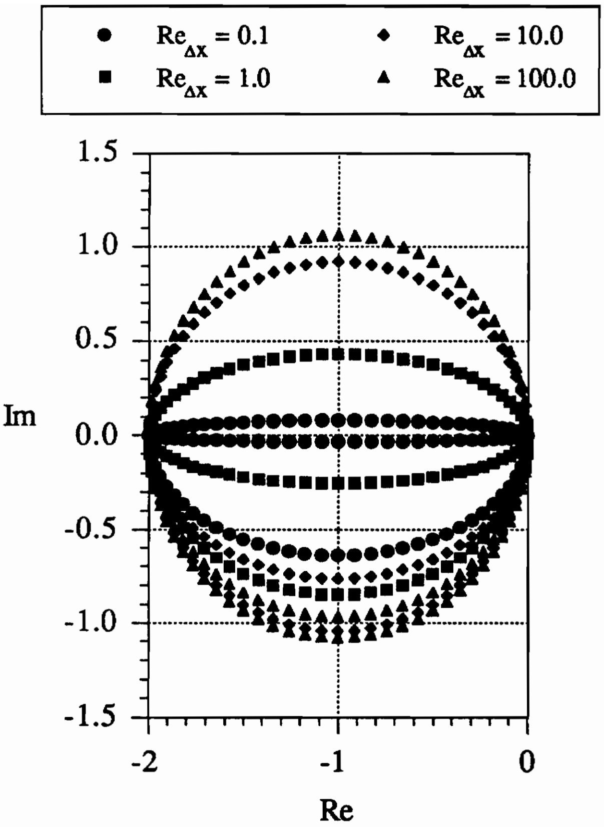

Figure 71. Fourier footprint using viscous preconditioning on the one-dimensional Navier-Stokes equations at at $M=0.5$.

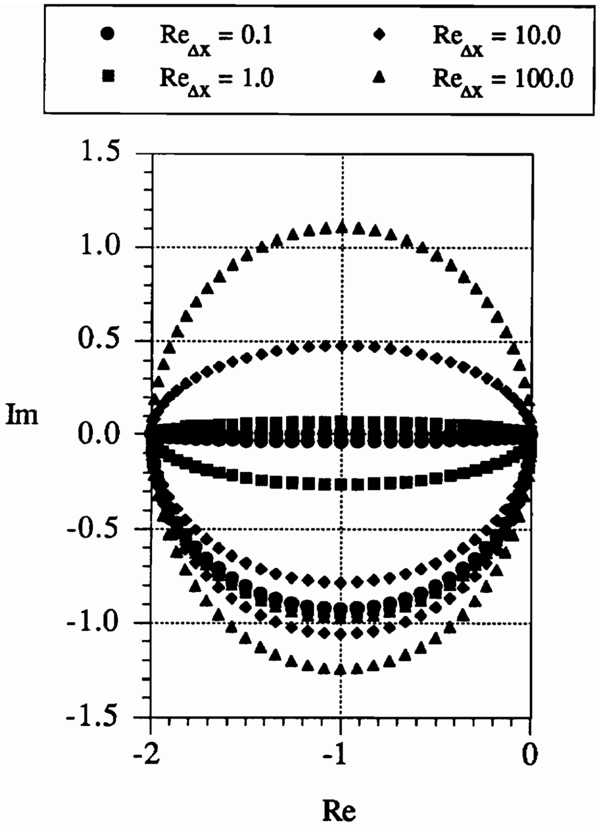

Figure 72. Fourier footprint using viscous preconditioning on the one-dimensional Navier-Stokes equations at $M = 0.9$.

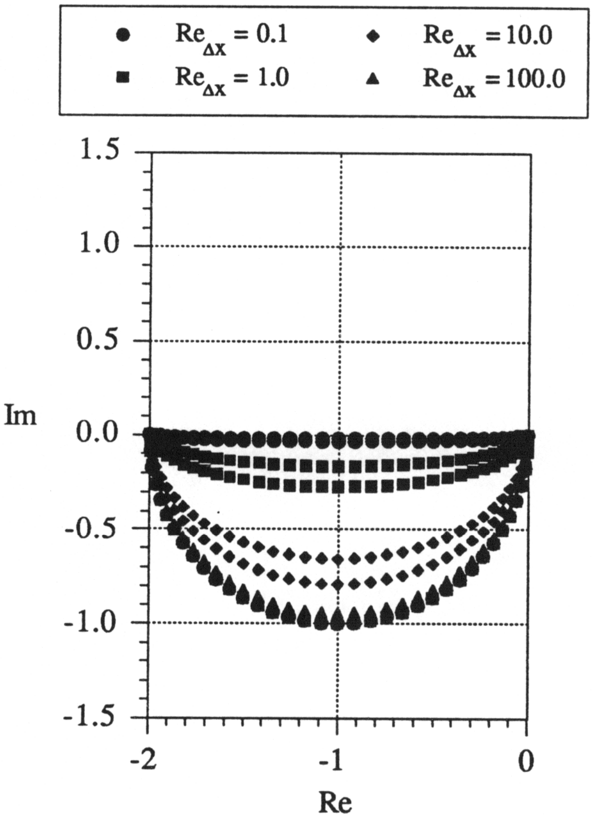

Figure 73. Fourier footprint using viscous preconditioning on the one-dimensional Navier-Stokes equations at $M=2.0$.

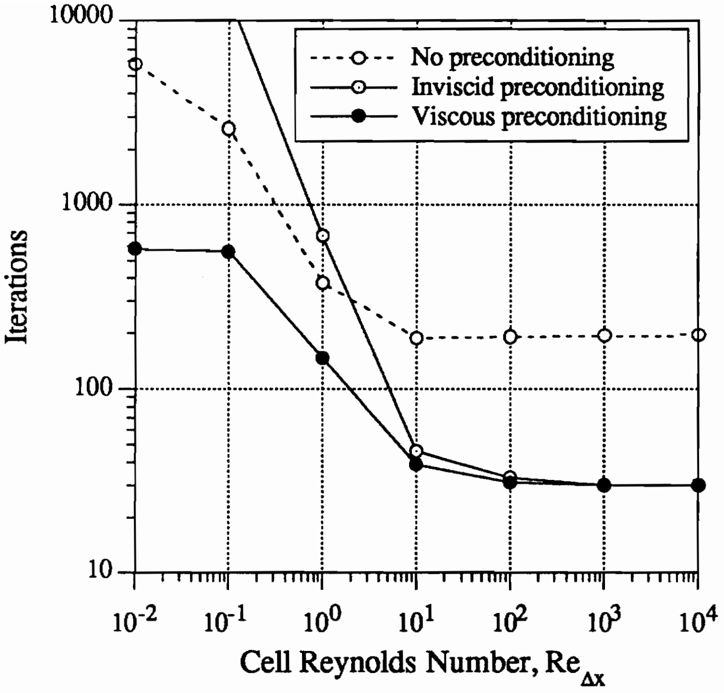

Figure 74. Iterations required to damp a 5% pressure disturbance in a $M = 0.1$ free stream at various cell Reynolds numbers.

Navier-Stokes equations over a flat plate. Note that a Blasius profile simulation has cell Reynolds numbers on the order of those in this pressure-damping example. The viscous equations with an inviscid preconditioning matrix are extremely stiff near the no-slip wall and converging the solution is prohibitively difficult. The velocity profile with preconditioning unrealistically tries to reverse its flow direction and separate. Applying the same idea as stated in Eqn. (5-10) with the proper inviscid preconditioning matrix and a thin-layer viscous flux Jacobian results in a cleaner footprint, but results to date have been unsuccessful. Lee and Van Leer [40] have successfully implemented the technique described in this Chapter for damping a small pressure disturbance in two dimensions.

To compute a more realistic problem, we consider the internal structure of a normal shock with upstream Mach number of $M = 2.0$ and a cell Reynolds number of $Re_{\Delta x} \approx 1.5$. Note that this problem is in the continuum transition regime where the Navier-Stokes equations are not strictly valid, but merely serves as a test problem for determining the numerical convergence-rate acceleration for the viscous preconditioning matrix. The domain is bounded by a distance of 45 mean free paths upstream and downstream of the shock. The mean free path assuming hard, elastic spheres may be determined from the upstream viscosity, density, and temperature (which are denoted μ_1, ρ_1, and T_1, respectively) through the expression

$$\lambda_1 = \frac{16\mu_1}{5\rho_1(2\pi R T_1)^{1/2}}. \qquad (5-11)$$

These properties and the mean free path are determined assuming sea-level, stagnation temperature and density and Sutherland's viscosity formula,

$$\mu(T) = C_1 \frac{T^{3/2}}{T + C_2}, \qquad (5-12)$$

where $C_1 = 1.458 \cdot 10^{-6} kg/(Ms\sqrt{K})$ and $C_2 = 110.4°K$.

This problem was proposed by Fisko and Chapman [20] and Grossman *et.al.* [26] during their examination of real-gas effects at hypersonic Mach numbers for monatomic and diatomic gases. The normal shock solution is shown along with the inviscid solution in Fig. (75). The viscous test case did not require a limiter. Residual histories with and without viscous preconditioning are shown in Fig. (76). Convergence is obtained in 2536 and 5091 iterations, respectively.

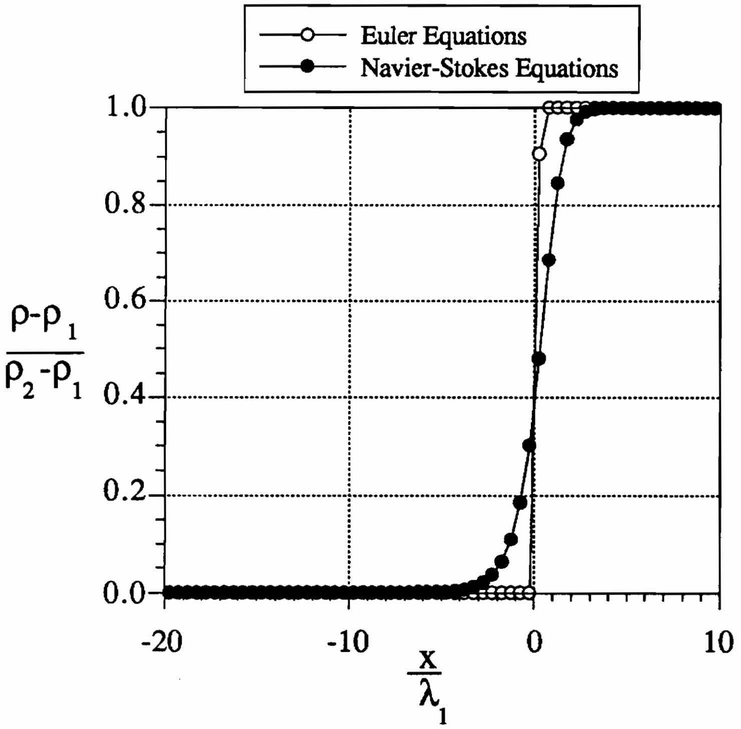

Figure 75. Internal normal shock structure. Shown is the inviscid and viscous normalized density distribution.

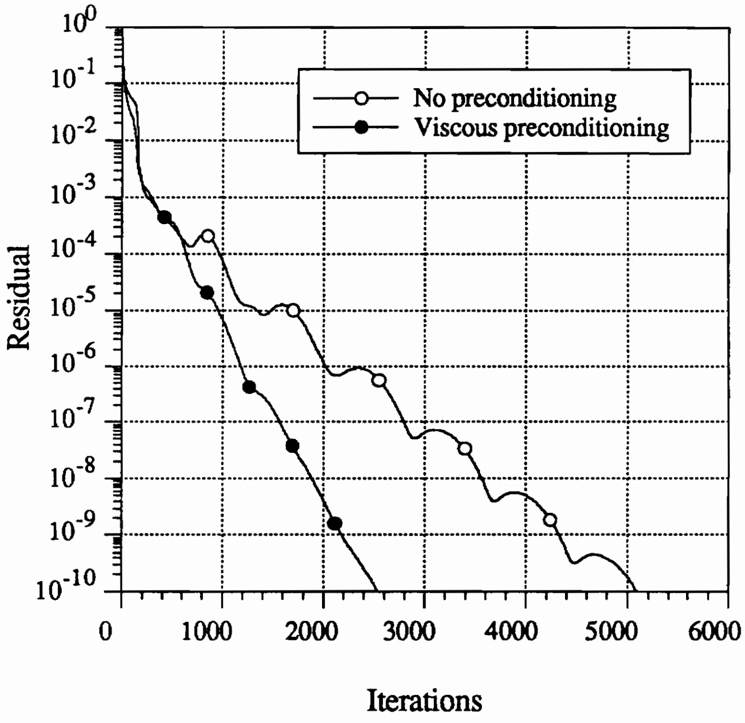

Figure 76. Residual histories for the computation of an internal normal shock structure.

CHAPTER

SIX

DUAL TIME STEPPING

6.1 Stiffness from Chemical Processes

Not surprisingly, the problem of stiffness with chemical processes boils down to a time-scale mismatch. For a fluid-dynamic state near equilibrium, the characteristic fluid-dynamic and chemical time scales brawl like prize-fighters. This can hardly be considered a fair fight since the characteristic time for the flow is much larger than for the reactions. To restore peace and order, a normalization similar to characteristic time stepping for the Euler equations is necessary; except, this time, for the chemical reaction rates.

The chemical time scales result from the reaction rates of chemical dissociations and recombinations. For example, the model equation for the rate of chemical-species production is

$$\frac{d\rho_i}{dt} = -k_f\,\rho_i, \tag{6–1}$$

where k_f is the rate at which forward reactions progress. This equation frequently models decaying processes and, for our application, yields a chemical time constant, τ_{chem}, of

$$\tau_{chem} = \frac{1}{k_f}. \tag{6–2}$$

For a gas close to equilibrium, the reaction rates are nearly instantaneous (*i.e.* $k_f \rightarrow \infty$ and $\tau_{chem} \rightarrow 0$). The classic text by Gear [21] discusses the time scale disparity in ordinary differential equations typically encountered in chemical kinetics and electrical network analysis. In CFD, analysis for systems of partial differential equations has been the subject of many papers with the recent funding for hypersonic flow research. Of renoun are the early works of Bussing and Murman [6] and Eklund *et.al.* [18].

As an example of the numerical ramifications of a time-scale disparity, consider the scalar wave equation with a "chemical" source term,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = -\frac{u}{\tau_{chem}}. \tag{6-3}$$

Using Euler explicit time integration and first-order upwind differencing yields a stability restriction of

$$a\Delta t/\Delta x \leq 1$$

$$\Delta t/\tau_{chem} \leq 2.$$

Fast chemical processes result in nearly instantaneous chemical time scales and the second equation for stability implies convergence to a steady-state solution in a prohibitively large number of iterations.

To determine the preconditioning for source term stiffness, consider the one-dimensional Euler equations and the reaction, $O_2 \rightleftharpoons 2O$. The equations are

$$\frac{\partial Q}{\partial t} + \frac{\partial f}{\partial x} = H, \tag{6-4}$$

where

$$Q = \begin{Bmatrix} \rho_{O_2} \\ \rho_O \\ \rho u \\ \rho e_0 \end{Bmatrix}, \quad f = \begin{Bmatrix} \rho_{O_2} u \\ \rho_O u \\ \rho u^2 + p \\ \rho u h_0 \end{Bmatrix}, \quad \text{and} \quad H = \begin{Bmatrix} \dot{w}_{O_2} \\ \dot{w}_O \\ 0 \\ 0 \end{Bmatrix}. \tag{6-5}$$

For this dissociation-recombination reaction, the production terms are

$$\begin{aligned} \dot{w}_{O_2} &= -\mathcal{M}_{O_2}\left[k_f \frac{\rho_{O_2}}{\mathcal{M}_{O_2}} - k_b \frac{\rho_O}{\mathcal{M}_O}\right] \\ \dot{w}_O &= \ 2\mathcal{M}_O\left[k_f \frac{\rho_{O_2}}{\mathcal{M}_{O_2}} - k_b \frac{\rho_O}{\mathcal{M}_O}\right], \end{aligned} \tag{6-6}$$

where k_f and k_b are the forward and backward reaction rates.

If we treat the convective terms explicitly and the source term *implicitly*, we obtain

$$\left[I - \Delta t \frac{\partial H}{\partial Q}\right] \frac{\Delta Q}{\Delta t} = -\left(\frac{\partial f}{\partial x} - H\right)^n . \tag{6–7}$$

By analogy to a preconditioned formulation, we can identify the preconditioning matrix for chemical reactions as

$$P^{-1} = I - \Delta t \frac{\partial H}{\partial Q}. \tag{6–8}$$

A formulation which treats the chemical source terms implicitly is a practical method for preconditioning the chemical reaction rates, but what is really happening to the time scales?

For simplicity of demonstration, consider the point implicit procedure with only the forward reaction, k_f. The source-term preconditioning yields

$$P^{-1} = \begin{bmatrix} 1 + k_f \Delta t & 0 & 0 & 0 \\ -2k_f \Delta t \mathcal{M}_{O_2}/\mathcal{M}_O & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6–9}$$

From the unit elements of the preconditioning matrix above, we see that the updates for momentum and energy-conservation quantities are based upon a fluid-dynamic time scale, Δt. However, the species continuity equation is preconditioned to give an update based upon τ_{chem}. Solving for the change in diatomic oxygen, we have

$$\Delta \rho_{O_2} = \left(\frac{-\Delta t}{1 + k_f \Delta t}\right) \left[-\frac{f_{j+1/2} - f_{j-1/2}}{\Delta x} - k_f \rho_{O_2}\right] \tag{6–10}$$

In the limit of infinite reaction rate, the effective time step becomes

$$\lim_{\left(\frac{\Delta t}{\tau_{chem}}\right) \to \infty} \left(\frac{\Delta t}{1 + \left(\frac{\Delta t}{\tau_{chem}}\right)}\right) \rightarrow \tau_{chem}. \tag{6–11}$$

The update equation in the limit is

$$\Delta \rho_{O_2} = -\tau_{chem} \left[-\frac{f_{j+1/2} - f_{j-1/2}}{\Delta x} - k_f \rho_{O_2}\right],$$

so the species equation receives a scaled time step on the order of its characteristic reaction rate. Modern production-level codes implement these ideas to effectively simulate chemically reacting flows.

6.2 Dual Time Stepping

A time accurate, low-speed combustion problem has two degrees of stiffness. The convective and acoustic waves are not only incompatible with each other, but also with the characteristic time scale of the chemical reactions. The preconditioning for normalizing the wave speeds is discussed in Chapter 3. From the previous discussion in Section 6.1, a point implicit procedure for the chemical source terms is theoretically an effective preconditioning for variable chemical reaction rates. Because of the chemical stiffness, temporal accuracy for reacting flow through an explicit algorithm is impractical. So, Withington *et.al.* [81] propose an implicit formulation known as *dual time stepping* which uses the preconditioning of Choi and Merkle [13] while treating the source terms implicitly. Dual time stepping allows for arbitrarily high-order time-accurate solutions at all Mach-number flows. This method incorporates a second temporal derivative (*i.e.* a "pseudo-time derivative") for converging to the root of a time-accurate discretization. Standard relaxation techniques apply in "pseudo-time", and the physical time derivative of the conserved state vector is calculated to high order.

In this dissertation, we incorporate dual time stepping with Van Leer's preconditioning matrix. The implicit governing equations with a pseudo-time derivative are

$$\frac{\partial Q}{\partial q}\mathcal{P}^{-1}\frac{\partial q}{\partial \tau} = -\left(\frac{\partial Q}{\partial t} + R(Q)\right)^{m+1} . \tag{6–12}$$

We wish to discretize the time derivative of the conservative variables to arbitrary order. The dual-time-stepping procedure iterates in "pseudo-time", τ, to find the solution at the $n+1^{st}$ time level in "real-time", t. For the purposes of linearization and arbitrary temporal accuracy, the general one-sided temporal derivative of the conserved variables is written as

$$\left(\frac{\partial Q}{\partial t}\right)^{n+1} = \frac{aQ^{n+1} - \phi(Q^n, Q^{n-1}, \ldots)}{\Delta t}, \tag{6–13}$$

where a is a constant coefficient and ϕ is a linear combination of the previous time-level's conservative variables. The state vector of conserved quantities at time level $n+1$ is unknown. This is the state that is iterated upon. So, replacing $n+1$ with $m+1$ in Eqn. (6-13) and linearizing in "pseudo-time", we have

$$\left(\frac{\partial Q}{\partial t}\right)^{m+1} = \frac{a(Q^m + \frac{\partial Q}{\partial q}\Delta q) - \phi(Q^n, Q^{n-1}, \ldots)}{\Delta t}, \tag{6–14}$$

where $\Delta q = q^{m+1} - q^m$. This is rearranged to become

$$\left(\frac{\partial Q}{\partial t}\right)^{m+1} = \frac{a}{\Delta t}\frac{\partial Q}{\partial q}\Delta q + \frac{aQ^m - \phi(Q^n, Q^{n-1}, \ldots)}{\Delta t}. \tag{6–15}$$

Substituting into the system in Eqn. (6-12), yields the system to numerically solve which is

$$\left[\frac{1}{\Delta\tau}\frac{\partial Q}{\partial q}\mathcal{P}^{-1} + \frac{a}{\Delta t}\frac{\partial Q}{\partial q} + \frac{\partial R}{\partial q}\right]\Delta q = -\left[\frac{aQ^m - \phi(Q^n, Q^{n-1}, \ldots)}{\Delta t} + R(q^m)\right]. \tag{6–16}$$

We may use any relaxation scheme we wish to determine the steady-state solution to the above equations, and upon convergence in "pseudo-time"

$$q^{m+1} = q^m \rightarrow Q^{n+1} = Q(q^m). \tag{6–17}$$

The results of this chapter are obtained by using approximate factorization in "pseudo-time" and second-order accuracy in "real-time". The cases presented serve only to demonstrate viability of the above proposed time-accurate implicit formulation and not as proof of its application to chemically reacting flows. Numerical one-sided temporal derivatives for Q_t are given in Table (11).

As an example, consider a second-order temporal approximation. The physical time derivative is discretized as

$$Q_t^{n+1} = \frac{3Q^{n+1} - 4Q^n + Q^{n-1}}{2\Delta t} + \mathcal{O}(\Delta t^2) \tag{6–18}$$

Table 11. Numerical coefficients for a high-order one-sided temporal derivative of the form $Q_t^{n+1} = \left[aQ^{n+1} - \phi(Q^n, Q^{n-1}, \ldots)\right] / \Delta t$.

Q_t	a	$\phi(Q^n, Q^{n-1}, \ldots)$
$\mathcal{O}(\Delta t)$	1	Q^n
$\mathcal{O}(\Delta t^2)$	$\frac{3}{2}$	$2Q^n - \frac{1}{2}Q^{n-1}$
$\mathcal{O}(\Delta t^3)$	$-\frac{11}{6}$	$-3Q^n + \frac{3}{2}Q^{n-1} - \frac{1}{3}Q^{n-2}$
$\mathcal{O}(\Delta t^4)$	$-\frac{25}{12}$	$-4Q^n + 3Q^{n-1} - \frac{4}{3}Q^{n-2} + \frac{1}{4}Q^{n-3}$

The corresponding second-order values of a and ϕ are

$$\begin{aligned} a &= 3/2 \\ \phi &= \frac{4Q^n - Q^{n-1}}{2}. \end{aligned} \tag{6-19}$$

The dual-time-stepping procedure iterates upon Q^{n+1} in "pseudo-time". Upon convergence, the second-order temporal solution results. This implicit procedure allows for larger physical time steps and arbitrary temporal accuracy for flows with stiff chemical processes.

6.3 Sod's Problem

As a test case for the dual-time stepping, a 10:1 driver-to-driven shock tube is calculated to second-order temporal and third-order spatial accuracy. The Venkat-Spekraize [59] spatial limiter is employed to control oscillations around discontinuities. This problem enjoys wide use in the high-order-accuracy literature. The initial conditions divide in half a uniform 100-cell mesh for a tube of unit length . The solution is determined after 20 dual-time steps and terminated at a physical time of $t = 0.0005$. The left and right states are specified as

$$\left\{ \begin{matrix} \rho \\ u \\ p \end{matrix} \right\}_l = \left\{ \begin{matrix} 1 \\ 0 \\ 1 \end{matrix} \right\}, \quad \text{and} \quad \left\{ \begin{matrix} \rho \\ u \\ p \end{matrix} \right\}_r = \left\{ \begin{matrix} 0.125 \\ 0 \\ 0.10 \end{matrix} \right\}. \tag{6-20}$$

The numerical and exact density distributions are depicted in Fig. (77). The numerical contact, shock and expansion fan faithfully propagate at the correct speeds. Convergence of the residual plus the time derivative to ten orders of magnitude in "pseudo-time" required between 36 and 68 iterations for the 20 physical time steps.

6.4 Instantaneously Perturbed Wall

A two-dimensional test case simulates the transonic evolution of flow around a suddenly perturbed channel wall. The above preconditioning is applied for a perfect

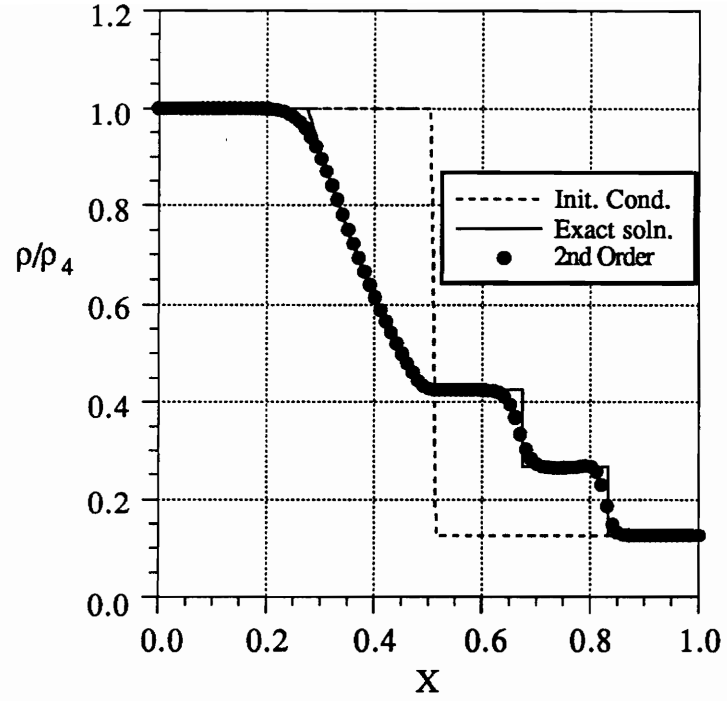

Figure 77. Density variation for Sod's shock-tube problem.

gas and the 61×31 mesh used in Section 4.1. The calculation begins with all cells initialized to the $M_\infty = 0.85$ free-stream state. The evolution progresses through 100 physical time steps with second-order temporal and third-order spatial accuracy. Convergence to ten orders in "pseudo-time" uses approximate factorization and the iterations required per outer loop are shown in Fig. (78). In separate cases, the calculation used the inviscid preconditioning of Chapter 3 and the simple conservative Jacobian as the mass matrix, P^{-1} in Eqn. $(6-16)$. Characteristic time stepping for this test problem shows eratic convergence-rate performance which is not understood. The flow evolution for the preconditioning case with modified Roe flux-difference splitting is shown in Figs. (79-83).

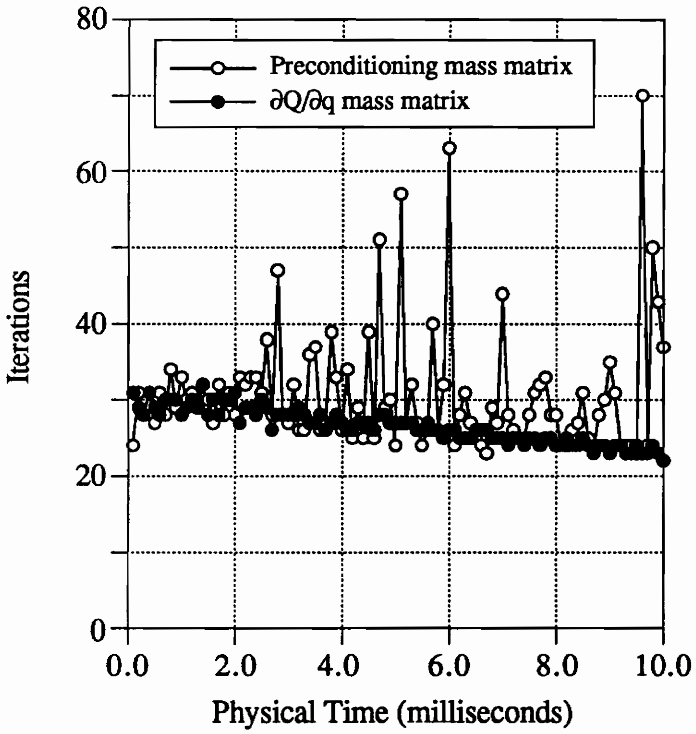

Figure 78. Iterations per *physical* time step for the evolution of the transonic channel flow.

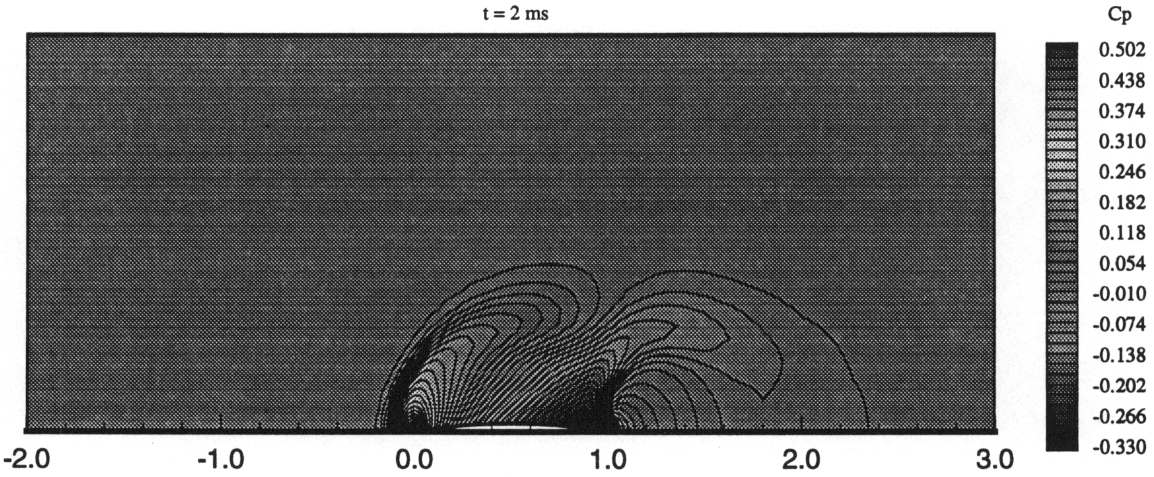

Figure 79. Evolution of the transonic channel flow from a free stream at $t = 2ms$.

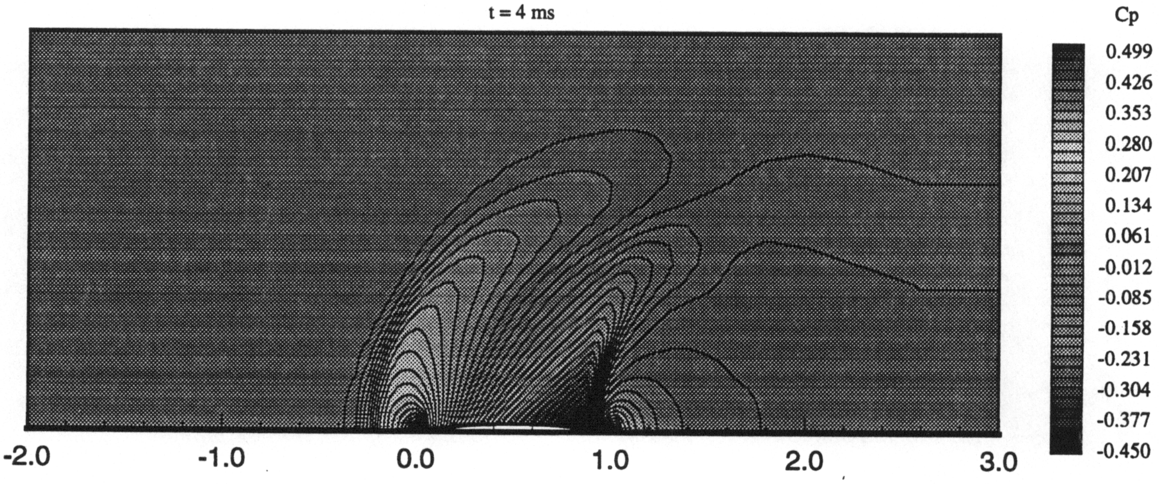

Figure 80. Evolution of the transonic channel flow from a free stream at $t = 4ms$.

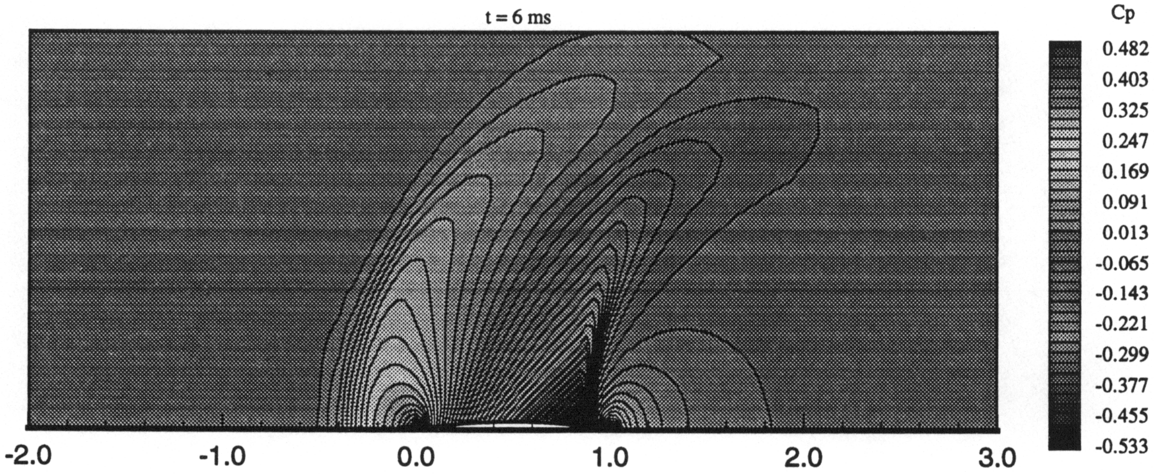

Figure 81. Evolution of the transonic channel flow from a free stream at $t = 6ms$.

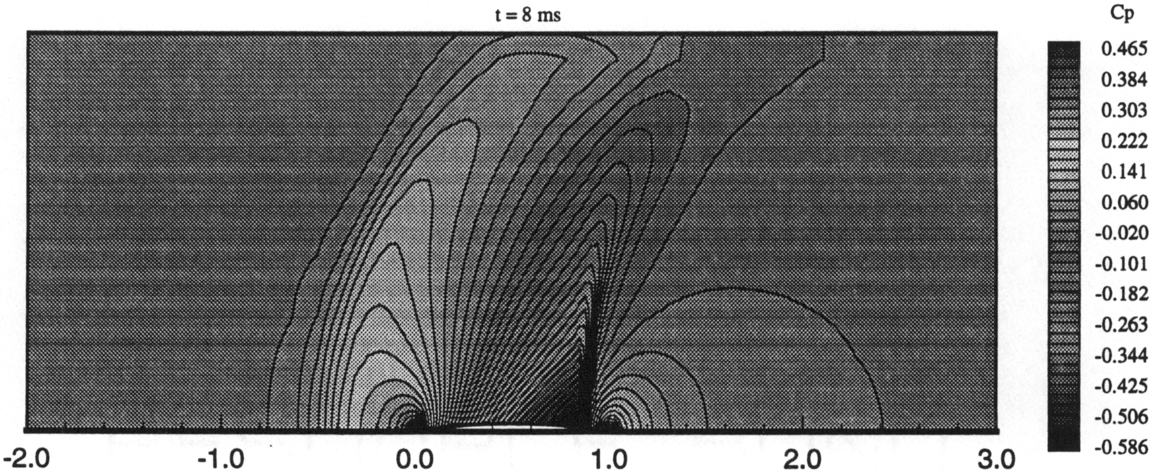

Figure 82. Evolution of the transonic channel flow from a free stream at $t = 8ms$.

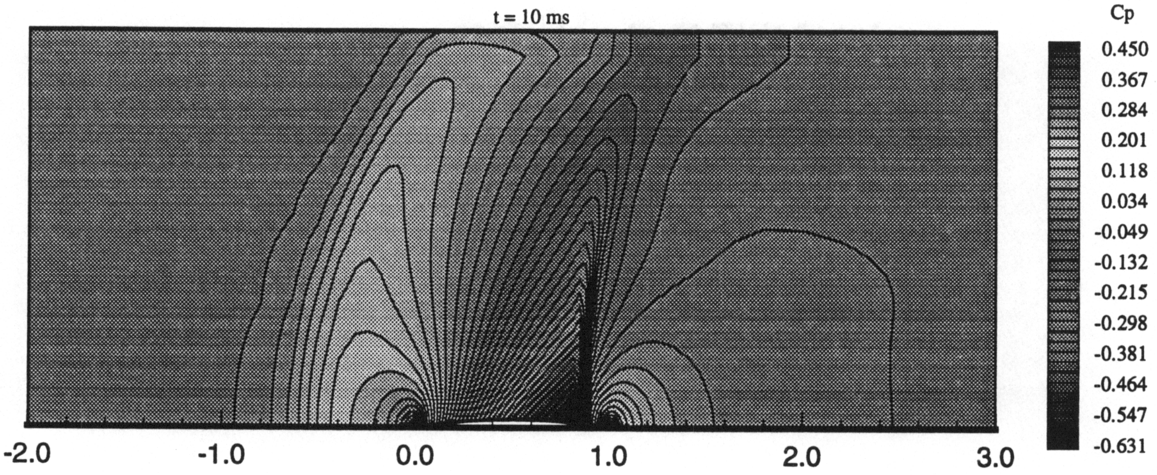

Figure 83. Evolution of the transonic channel flow from a free stream at $t = 10ms$.

CHAPTER SEVEN

CONCLUSIONS

Part II of this thesis documents the formulation for extending the matrix preconditioning of Van Leer to finite-rate chemistry and for implicit time integration. A significant effort was made to document the entire theoretical process of preconditioning from the underlying physical wave structure to the matrix that normalizes the wave speeds.

Implementation through perfect-gas experiments confirms convergence acceleration at all subsonic Mach numbers with significant improvement in the transonic and incompressible regimes. Inviscid incompressible flows are quickly and efficiently calculated using a compressible formulation and eigenvalue stiffness is effectively reduced. Results for marching a chemically reacting flow proves likewise beneficial at low supersonic speeds.

The dual time stepping procedure is documented and utilized for second-order temporal accuracy within an implicit formulation. The future application is for low-speed combustion problems (*e.g.* diffusion flames or the internal combustion engine).

In addition, the one-dimensional Navier-Stokes equations are effectively preconditioned. Future work must robustly simulate all flow regimes, namely the realm of the complete Navier-Stokes equations. A viscous preconditioning which controls the high-frequency errors seems the most profitable extension to multiple dimensions.

[1] Abarbanel, S., Dutt, P., and Gottlieb, D., "Splitting Methods for Low Mach Number Euler and Navier-Stokes Equations," *NASA Contractor Report 178297* and *ICASE Report No. 87-30*, May, 1987.

[2] Abgrall, R., "Design of an Essentially Non-Oscillatory Reconstruction Procedure on Finite-Element Type Meshes," *NASA Report No. 189574*, December, 1991.

[3] Barth, T.J. and Frederickson, P.O., "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," *AIAA 90-0013*, Reno, Nevada, January, 1990.

[4] Broyden, C.G., **Basic Matrices - An Introduction to Matrix Theory and Practice**, The Macmillan Press, 1975.

[5] Burden, R.L. and Faires, J.D., **Numerical Analysis - Fourth Edition**, PWS-KENT Publishing Company, 1989.

[6] Bussing, T.R.A. and Murman, E.M., "A Finite-Volume Method for the Calculation of Compressible Chemically Reacting Flow," *AIAA 85-0331*, Reno, Nevada, January, 1985.

[7] Casper, J., "An Extension of Essentially Non-Oscillatory Shock Capturing Schemes to Multi-Dimensional Systems of Conservation Laws," Doctoral Dissertation, Old Dominion University, December, 1990.

[8] Casper, J., "Finite Volume Application of High Order ENO Schemes to Two-Dimensional Boundary Value Problems," *AIAA 91-0631, submitted to Journal of Computational Physics*, 1991.

[9] Chakravarthy, S.R., "Some Aspects of Essentially Non-Oscillatory (ENO) Formulations for the Euler Equations," *NASA Contractor Report 4285*, May, 1990.

[10] Chakravarthy, S.R., "Euler Equations - Implicit Schemes and Implicit Boundary Conditions," *AIAA 82-0228*, January, 1982.

[11] Chiocchia, G., "Exact Solutions to Transonic and Supersonic Flows," *AGARD Advisory Report, AR-211*, 1985.

[12] Choi, Y.-H. and Merkle, C.L., "The Application of Preconditioning in Viscous Flows," to appear in *Journal of Computational Physics*, 1992.

[13] Choi, Y.-H. and Merkle, C.L., "Application of Time-Iterative Schemes to Incompressible Flows," *AIAA Journal* Vol. **23**, no. 10, p 1518-1524, 1985.

[14] Chorin, A.J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, **2**, 1967.

[15] Cruz, J. (private communication)

[16] Dadone, A. and Grossman, B., "Surface Boundary Conditions for the Numerical Solution of the Euler Equations," Virginia Tech ICAM Report *ICAM 92-10-04*, 1992.

[17] Deconinck, H, "Random Choice Solution of Hyperbolic Systems," *VKI Lecture Series 1987-04 on Computational Fluid Dynamics*, March, 1987.

[18] Eklund, D.R., Hassan, H.A., and Drummond, J.P., "The Efficient Calculation of Chemically Reacting Flow," *AIAA 86-0563*, Reno, Nevada, January, 1986.

[19] Feng, J. and Merkle, C.L., "Evaluation of Preconditioning Methods for Time-Marching Systems," *AIAA 90-0016* Reno, Nevada, 1990.

[20] Fisko, K.A. and Chapman, D.R., "Comparison of Burnett, Super-Burnett and Monte Carlo Solutions for Hypersonic Shock Structure," *16th International Symposium of Rarefied Gas Dynamics* 1988.

[21] Gear, C.W., **Numerical Initial Value Problems in Ordinary Differential Equations**, Prentice-Hall, pp. 209-223, 1971.

[22] Glaister, P., "An Approximate Linearised Riemann Solver for the Euler Equations for Real Gases," *Journal of Computational Physics*, **74**, pp. 382-408, 1988.

[23] Godunov, S.K., "A Finite-difference Method for the Numerical Computation and Discontinuous Solution of the Equations of Fluid Dynamics," *Matematicheskii Sbornik*, **74**, Vol. 47, pp. 271-306, 1959. (in Russian). (Translated by US Joint Publ. Res. Service as JPRS 7226, 1969).

[24] Grossman, B. and Cinnella, P., "Flux-Split Algorithms for Flows with Non-equilibrium Chemistry and Vibrational Relaxation," *Journal of Computational Physics*, **88**, pp. 131-168, 1990.

[25] Grossman, B. and Walters, R.W., "An Analysis of Flux-Split Algorithms for Euler's Equations with Real Gases," *AIAA Journal*, 27, No. 5, pp. 524-531, 1989. (also AIAA 87-1117)

[26] Grossman, B., Cinnella, P., and Eppard, W.M., "New Developments Pertaining to Algorithms for Non-equilibrium Hypersonic Flows," *CFD Journal*, Vol. 1, No. 2, pp. 175-186, July, 1992.

[27] Harten, A., "Short Course: Numerical Methods for Hyperbolic Conservation Laws," *ICASE Internal Report, Document No. 33.*, November 25, 1986.

[28] Harten, A., "On the Symmetric Form of Systems of Conservation Laws with Entropy," *ICASE Report No. 81-34.*, October, 1981.

[29] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S.R., "Uniformly High-Order Accurate Essentially Non-Oscillatory Schemes III," *Journal of Computational Physics*, **71**, pp. 231-323. 1987.

[30] Harten, A. and Osher, S., "Uniformly High-Order Accurate Essentially Non-Oscillatory Schemes I," *MRC Technical Summary Report* No. 2823, May, 1985.

[31] Harten, A., Lax, P.D., and Van Leer, B., "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," *Society for Industrial and Applied Mathematics Review*, Vol. 25, No. 1, January, 1983.

[32] Hindman, R.G., "Generalized Coordinate Forms of Governing Fluid Equations and Associated Geometrically Induced Errors," *AIAA Journal*, Vol. 20, No. 10, October, 1982.

[33] Hirsch, C. **Numerical Computational of Internal and External Flows**, *Volume I, Fundamentals of Numerical Discretization*, John Wiley and Sons, 1988.

[34] Hirsch, C. **Numerical Computational of Internal and External Flows**, *Volume II, Computational Methods for Inviscid and Viscous Flows*, John Wiley and Sons, 1988.

[35] Hsu, C.-H., Chen, Y.-M., and Liu, C.-H. "Preconditioned Upwind Methods to Solve 3-D Incompressible Navier-Stokes Equations for Viscous Flows," *AIAA 90-1496*, 1990.

[36] Hussaini, M.Y., Kopriva, D.A., Salas, M.D., and Zang, T.A. "Spectral Methods for the Euler Equations: Fourier Methods and Shock-Capturing," *NASA Contractor Report 172294*, NASA Langley Research Center, 1984.

[37] Kang, S.W., Dunn, M.G., and Jones, W.L., "Theoretical and Measured Electron-Density Distributions for the RAM Vehicle at High Altitudes," *AIAA 72-689*, Boston, Massachusetts, June, 1972.

[38] Lax, P. and Wendroff, B. "Systems of Conservation Laws," Communications on Pure and Applied Mathematics, Vol XIII, pp. 217-237, 1960.

[39] Lee, W.-T., "Local Preconditioning of the Euler Equations," Doctoral Dissertation, University of Michigan, 1991.

[40] Lee, D. and Van Leer, B., "Progress in Local Preconditioning of the Euler and Navier-Stokes Equations," *11th AIAA Computational Fluid Dynamics Conference*, 1993 (to appear).

[41] Liou, M.-S., Van Leer, B., and Shuen, J.-S., "Inviscid Flux Splitting Algorithms for Real Gases with Nonequilibrium Chemistry," *NASA TM-100856*, 1988.

[42] McGhee, R.J., Walker, B.S., and Millard, B.F., "Experimental Results for the Eppler 387 Airfoil at Low Reynolds Numbers in the Langley Low-Turbulence Pressure Tunnel," *NASA TM-4062*,1988.

[43] K. Meadows, A. Kumar, and M. Hussaini, "A Computational Study on the Interaction Between a Vortex and a Shock Wave," *AIAA 89-1043*, San Antonio, Texas, April, 1989.

[44] Merkle, C.L. and Choi, Y.-H., "Computation of Low-Speed Flow with Heat Addition," *AIAA Journal*, pp. 831-838, June, 1987.

[45] Naumann, A. and Hermanns, E., "On the Interaction Between a Shock Wave and a Vortex Field," *AGARD-CP-131*, 1973.

[46] Osher, S. and Chakravarthy, S., "Very High-Order Accurate TVD Schemes," *ICASE Report No. 84-44*, 1984.

[47] Rizzi, A., "Computation of Rotational Transonic Flow," *Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves, A GAMM Workshop*, pp. 153-166, 1981.

[48] Rogers, S.E., Kwak, D., and Kiris, C., "Numerical Solution of the Incompressible Navier-Stokes Equations for Steady- State and Time-Dependent Problems," *AIAA 89-0463*, Reno, Nevada, January, 1989.

[49] Roe, P.L., "Introduction to Computational Fluid Dynamics," *ICASE Internal Report, Document No. 35*, NASA Langley Research Center, August, 1987.

[50] Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, **43**, pp. 357-372, 1981.

[51] Roe, P.L. and Baines, M.J., "Algorithms for Advection and Shock Problems," *Proc. 4^{th} GAMM Conference on Numerical Methods in Fluid Mechanics* , Braunschweig Vieweg, 1982.

[52] Rumsey, C.L., Van Leer, B., and Roe, P.L., "Effect of a Multi-dimensional Flux Function on the Monotonicity of Euler and Navier-Stokes Computations," *AIAA 91-1530*, 1991.

[53] Shu, C.-W., "Numerical Experiments on the Accuracy of ENO and Modified ENO Schemes," *NASA Contractor Report No. 182083*, August, 1990.

[54] Shu, C.-W., "TVD Time Discretization II - Time Dependent Problems," *UCLA Preprint*, 1980.

[55] Shu, C.-W. and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Schemes," *Journal of Computational Physics*, **77**, pp. 439-471, 1988.

[56] Shu, C.-W. and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Schemes II," *Journal of Computational Physics*, **83**, 1989.

[57] Thomas, J.L., Van Leer, B., and Walters, R.W., "Implicit Flux-Split Schemes for the Euler Equations," *AIAA 85-1680*, Cincinnati, Ohio, July, 1985.

[58] Thomas, J.L. and Walters, R.W., "Upwind Relaxation Algorithms for the Navier-Stokes Equations," *AIAA 85-1501-CP*, Cincinnati, Ohio, July, 1985.

[59] Thomas, J.L., (private communication)

[60] Turkel, E., "Review of Preconditioning Methods for Fluid Dynamics," to appear in *Applied Numerical Mathematics*, 1992.

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

[62] Turkel, E., "Acceleration to a Steady State for the Euler Equations," *NASA Report No. 172398*, NASA Langley Research Center, 1984.

[63] Van Albada, G.D., Van Leer, B., and Roberts, W.W., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," *Astronomy and Astrophysics*, **108**, pp. 76-84, 1982.

[64] Van Leer, B., Lee, W.-T., and Roe, P.L., "Characteristic Time-Stepping or Local Preconditioning of the Euler Equations," *AIAA 91-1552*, Honolulu, Hawaii, June, 1991.

[65] Van Leer, B., Tai, C.-H., and Powell, K.G., "Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations," *AIAA 89-1933*, 1989.

[66] Van Leer, B., "Computational Methods for Ideal Compressible Flow," *NASA Contractor Report No. 172180*, July, 1983.

[67] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme II. Monotonicity and Conservation Combined in a Second-Order Scheme," *Journal of Computational Physics*, **14**, pp. 361-370, 1974.

[68] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme III. Upstream-Centered Finite-Difference Schemes for Ideal Compressible Flow," *Journal of Computational Physics*, **23**, pp. 263-275, 1977.

[69] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme IV. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, **32**, pp. 101-136, 1979.

[70] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V. A New Approach to Numerical Convection," *Journal of Computational Physics*, **23**, pp. 276-299, 1977.

[71] Van Leer, B., Tai, C.-H., and Powell, K.G., "Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations," *AIAA 89-1933*, 1989.

[72] Van Leer, B., "Flux-Vector Splitting for the Euler Equations," *Lecture Notes in Physics*, **Vol. 170**, pp. 507-512, 1982.

[73] Van Leer, B., "Class Notes for Numerical Fluid Dynamics, II," *Internal ICASE Report*, Winter , 1987.

[74] Van Leer, B., "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations," *AIAA Eighth CFD Conference*, pp. 156-161, Honolulu, Hawaii, June, 1987.

[75] Vincenti, W.G. and Kruger C.H., **Introduction to Physical Gas Dynamics**, Robert E. Kreiger Publishing Company, 1986.

[76] Vinokur, M. and Liu, Y., "Equilibrium Gas Flow Computations: II. An Analysis of Numerical Formulations of Conservation Laws," *AIAA 88-0127*, June, 1988.

[77] Viviand, H., "Pseudo-unsteady Systems for Steady Inviscid Flow Calculations," *Numerical Methods for the Euler Equations of Fluid Dynamics*, 1985.

[78] Volpe, G., "On the Use and Accuracy of Compressible Flow Codes at Low Mach Number," *AIAA 91-1662*, Honolulu, Hawaii, June, 1991.

[79] Walters, R.W., Cinnella, P., Slack, D.C., and Halt, D., "Characteristic-Based Algorithms for Flows in Thermo-Chemical Nonequilibrium," *AIAA Journal*, **Vol. 30**, No. 5, May, 1992.

[80] Walters, R.W., Cinnella, P., and Slack, D.C., "A Status Report on GASP - A General Aerodynamic Simulation Program," *Seventh National Aero-Space Plane Symposium*, Paper No. 9, NASA Lewis, 1989.

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

[82] Wolfram, S. **Mathematica: A System for Doing Mathematics by Computer**, *Second Edition*, Addison-Wesley Publishing Company, 1991.

VITA

The author was born to Lee and Emily Godfrey in Tampa, Florida on May 23, 1966. In 1984, he began his aerospace studies at Virginia Polytechnic Institute and State University (VPI & SU) after attending high school at The Baylor School for Boys. During his undergraduate studies, he spent two quarters in the engineering CO-OP program with NASA Langley Research Center. In 1988, he received his B.S. degree in Aerospace Engineering graduating *cum laude*. During his graduate education at VPI & SU, he married his bride, Gwendolyn Lee Hess, the daughter of Danny and Janice Hess on May 2, 1992. He then continued with his education in Aerospace Engineering to receive a Ph.D. Degree.

Andrew G. Godfrey