8-2-2003

A Global Preconditioning Method for the Euler Equations

B. Gazi Yildirim

A GLOBAL PRECONDITIONING METHOD FOR THE EULER EQUATIONS

By

B. Gazi Yildirim

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Aerospace Engineering
in the Department of Aerospace Engineering

Mississippi State, Mississippi

December 2003

A GLOBAL PRECONDITIONING METHOD FOR THE EULER EQUATIONS

By

B. Gazi Yildirim

Approved:

Pasquale Cinnella Associate Professor of Aerospace Engineering (Director of Thesis)	Z.U.A Warsi Professor Emeritus of Aerospace Engineering (Committee Member)
Jonathan Mark Janus Associate Professor of Aerospace Engineering (Committee Member)	
Pasquale Cinnella Associate Professor of Aerospace Engineering Graduate Coordinator	A. Wayne Bennett Dean of the Bagley College of Engineering

Name: B. Gazi Yildirim

Date of Degree: December 13, 2003

Institution: Mississippi State University

Major Field: Aerospace Engineering

Major Professor: Dr. Pasquale Cinnella

Title of Study: A GLOBAL PRECONDITIONING METHOD FOR THE EULER EQUATIONS

Pages in Study: 102

Candidate for Degree of Master of Science

This study seeks to validate a recently introduced global preconditioning technique for the Euler equations. Energy and enthalpy equations are nondimensionalized by means of a reference enthalpy, resulting in increased numerical accuracy for low-speed flows. A cell-based, finite volume formulation is used, with Roe flux difference splitting and both explicit and implicit time integration schemes. A Newton-linearized iterative implicit algorithm is implemented, with Symmetric Gauss-Seidel (LU/SGS) nested sub-iterations. This choice allows one to retain time accuracy, and eliminates approximate factorization errors, which become dominant at low speed flows. The linearized flux Jacobians are evaluated by numerical differentiation. Higher-order discretization is constructed by means of the MUSCL approach. Locally one-dimensional characteristic variable boundary conditions are implemented at the farfield boundary. The preconditioned scheme is successfully applied to the following traditional test cases used as benchmarks for local preconditioning techniques: point disturbance, flow angle disturbance, and stagnation point arising from the impingement of two identical jets. The flow over a symmetric airfoil and a convergent-divergent nozzle are then simulated for arbitrary Mach numbers. The preconditioned scheme greatly enhances accuracy and convergence rate for low-speed flows (all the way down to $M \approx 10E-4$). Some preliminary tests of fully unsteady flows are also conducted.

ACKNOWLEDGMENTS

I am very grateful for Dr. Cinnella for his guidance. Whenever I had a problem, lost my hope about the problem, his analytical approach was a great help. He also edited my thesis meticulously. I would like to express my appreciation to Dr. Warsi for his support. I especially thank Dr. Janus for his clear and complete instruction and his past works which I exploited a lot.

Most importantly, I am indebted to my parents for supporting me in every moment, and thank you to my sister also.

Thanks also to the Simcenter/MSU for providing the computing resources to make this work possible. Of course, I have spent almost two years in Starkville. I made some friends and have shared good times. Thanks for their help, support, and warm and sincere relationships.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

NOMENCLATURE

a, b, c, s	flux Jacobians in primitive variable
$a_\Gamma, b_\Gamma, c_\Gamma, s_\Gamma$	system matrix in preconditioned primitive formulation
c	speed of sound
A, B, C, K	flux Jacobians in conserved variable
C_p	Specific heat at constant pressure
C_v	Specific heat at constant volume
e	specific energy
E	total energy per unit volume
E_c	Eckert number
f, g, h	flux vectors (Cartesian coordinates)
F, G, H	flux vectors in computational space
h	perturbation
h_t	total specific enthalpy
J	Jacobian of transformation
L	length
M	transformation matrix to primitive variable, Mach number
p	pressure
q	primitive variable vector (curvilinear coordinates)
Q	conserved variable vector (curvilinear coordinates)
R	gas constant
$R_\xi, R_\eta, R_\zeta, R_q, R_Q$	eigenvectors
$\Re$	steady residual, residual
$\Re_U$	unsteady residual
t	physical time
T	temperature

u, v, w	Cartesian velocity components
U, V, W	velocity components in curvilinear coordinates
w	primitive variable vector (Cartesian coordinates)
W	conserved variable vector (Cartesian coordinates), characteristic variables
x, y, z	Cartesian axes
α	angle of attack
γ	Specific heat ratios C_p/C_v
Γ_q	global preconditioning matrix
$\Delta\tau$	time step
λ, Λ_k	eigenvalue, eigenvalue matrix
ξ, η, ζ, k	curvilinear axes
ρ	density
τ	computational time

Subscripts:

a	approaching boundary
b	boundary value
e	exit condition
i, j, k	location index
l	leaving boundary
L, R	left and right of an interface
o	frozen value, stagnation condition
r	reference quantity
∞	free stream quantity

Superscripts:

L, R	left and right of an interface
m	Newton iteration index
n	time level index

p	symmetric Gauss-Seidel iteration index
$\bar{}$	dimensional quantities
$\tilde{}$	Roe averaged
$'$	differentiation
$\hat{}$	normalized metrics

Mathematical Operators:

Max	maximum value
Min	minimum value
Sgn	sign function
$\nabla\cdot$	divergence
δ	central difference operator
Δ	difference
∇	gradient
$\cdot$	dot product
$\|\ \|$	Absolute value, determinant

CHAPTER I

INTRODUCTION

Any mutual motion between an object and a fluid will cause particles of the fluid to collide with the object. A fluid moving around a body is governed by the three conservation laws of mass, momentum, and energy. A large amount of practical applications occur in continuous media and are governed by the Navier-Stokes (N-S) equations. determination of flow properties around a body is highly desirable, in order to build and develop highly efficient vehicles or devices. Unfortunately, solving the N-S equations analytically seems impossible, except in some very simple problems. In many aerodynamics applications, the exclusion of viscous terms is a fairly good approximation for solving flowfields. The flow excluding viscous effects is called inviscid, and is governed by the Euler equations, which are a subset of the N-S equations. Nonetheless, this simplification does not mean that the Euler equations have analytical solutions for real world applications.

With the advent of high-speed computers, the solution of Euler and N-S equations has relied on numerical methods. Since the N-S equations consist of the Euler equations plus viscous and heat flux terms, the solution techniques for the Euler equations are equally important for viscous problems.

Presently, numerical methods for hyperbolic equations have played a crucial role in solving flow problems in an efficient and reliable manner. The hyperbolicity of Euler equations is guaranteed in their unsteady form; however, steady state problems, which are not fully hyperbolic, turn out to be solvable when using the unsteady form. Unfortunately, this approach cannot be extended to incompressible flows, where density changes are negligible. Modifications are necessary in order to remove ill-conditioned behavior for numerical methods at low-speed flows: hence, the development of preconditioning methods.

As mentioned earlier, traditional compressible algorithms fail at low Mach numbers in terms of accuracy and convergence. Attempts to correct this problem have resulted in

preconditioned schemes: the derivatives in the governing equations are modified by the introduction of a preconditioning matrix. The scheme is *local* when the matrix depends on local values of the flow, and *global* when it depends only a global (constant) reference values. Recent investigations of existing *local* preconditioning schemes are presented in [1] [2] [3]. In subsonic flow, the results generated by compressible algorithms deteriorate as the Mach number is reduced, mainly because there is a large disparity in convective and acoustic parts of the system eigenvalues; moreover the compressible equations are improperly scaled at low speed [4]. Preconditioning (either *local* or *global*) plays an important role in solving these problems.

Local preconditioning is designed to remove the arising problems due to low Mach numbers. The most two important benefits of preconditioning can be listed as: (1) balancing the order of magnitude of convective and acoustic parts of the eigenvalues; and, (2) scaling compressible equations properly for $M \rightarrow 0$. In addition to that, local preconditioners can provide some lesser important advantages which will be discussed shortly in this study. On the other hand, local preconditioning schemes suffer for vanishing Mach number. This problem has been attributed to (1) flow angle sensitivity [5], (2) eigenvector structure [6], and (3) lack of symmetrizability of preconditioning equations [2]. All of the existing preconditioners have been shown by Zaccanti [2] to be suffering in the vanishing Mach number limit. To resolve this issue, the same author proposed a hybrid scheme, which is combination of robust preconditioners (Turkel, Choi and Merkel) in low Mach number region with Van-Leer-Roe in the remaining flowfield.

Recently, a new global preconditioner was introduced by Briley, Taylor and Whitfield [7]. Unlike local preconditioning, this will change the governing equation globally. Preconditioned methods are designed to solve the compressible equations, while preserving accuracy and convergence behavior at low speeds. This new approach is inherently not local, thus the deficiencies of local preconditioning should not be encountered here.

The purpose of this study is to implement, validate, and test the global preconditioner and assess its performance. A characteristic-based finite volume upwind formulation with flux difference splitting is used to discretize the governing equations. The introduction

of a new term in the time derivative will change characteristics of the equations, hence face fluxes are in need to be modified in order to preserve accuracy at low Mach numbers [8]. Changing the time derivative term in the equation does not cause any problem in a steady-state problem, but it destroys time accuracy. In order to circumvent this problem, a Discretized-Newton formulation with Lower-Upper Symmetric Gauss-Seidel (LU/SGS) is chosen, because : (1) the Newton formulation takes into account unsteadiness; and (2)introducing one more subiteration level is an alternative, efficient solving method for the resulting linear system of equations, compared with approximate factorization (AF) methods, and it can eliminate AF errors.

Characteristic variable boundary conditions (CVBCs) are the most logical and accurate to impose on computational boundaries, and are crucial for successfully implementing numerical algorithms. CVBCs were developed for farfield boundary conditions [9]. The change of characteristic of the equations will results in a different set of eigenvectors, thus having different characteristic variables. CVBCs are presented for the new scheme, and their quality is shown by a comparison with other numerical results at the farfield boundaries.

Numerical experiments are conducted to assess the reliability and robustness of the new global preconditioner. Traditionally, local preconditioning schemes are evaluating using three test cases, namely point-disturbance, flow angle, and stagnation point tests. These tests try to minimize boundary condition effects in order to observe the behavior of preconditioned schemes [2]. In addition, the use of an explicit scheme is useful here, in order to circumvent computation complexities present in implicit schemes.

The new scheme is applied to steady-state (internal and external), and unsteady flowfields. External flows may have one or more stagnation points; therefore, external flowfields are good examples to test the new preconditioning ability near stagnation points. For airfoil tests, the code is applied to variable Mach numbers changing from $M = 0.001$ to $M = 1.2$ (using a NACA 0012 airfoil). In order to show that the new scheme is reliable and robust, the code should be applicable to a variety of flowfields. A nozzle problem is an example of internal flow which does not have a stagnation point;however, it may feature very low speeds locally. Thus it represents an excellent example of mixed

flow types. Several different flow conditions (subsonic and supersonic nozzle, shock in the diffuser) can be achieved by changing the exit pressure. Lastly, a shock tube problem is computed for various reference Mach numbers, as a simple test of the unsteady flow used to see capabilities of the present approach .

CHAPTER II

GOVERNING EQUATIONS

The governing equations for fluid flows are based on conservation of mass, Newton's second law of motion, and the first law of thermodynamics.

Momentum and energy equations are complicated for real flows due to the presence of viscosity, chemical reactions, thermal conduction, and gravity. In here, it is assumed that all these influences are negligible(a reasonable assumption for many practical applications).

Also, it is necessary to introduce one equation of state, which represents the fact that the local thermodynamic state is known when any two independent thermodynamics variables are known.

2.1 Euler Equations

The three laws of conservation, plus an equation of state, constitute the Euler equations (in a fixed coordinate system). The Euler equations can be written in the following form:

$$\frac{\partial \bar{\rho}}{\partial \bar{t}} \; + \; \nabla \cdot \bar{\rho}\bar{\mathbf{u}} \; = \; 0, \tag{2.1}$$

$$\frac{\partial \bar{\rho}\bar{\mathbf{u}}}{\partial \bar{t}} \; + \; \nabla \cdot \bar{\rho}\bar{\mathbf{u}}\bar{\mathbf{u}} \; + \; \nabla \bar{p} \; = \; 0, \tag{2.2}$$

$$\frac{\partial \bar{E}}{\partial \bar{t}} \; + \; \nabla \cdot (\bar{E} \; + \; \bar{p})\bar{\mathbf{u}} \; = \; 0. \tag{2.3}$$

Here , $\bar{\rho}$ is density, $\bar{p}$ is pressure, $\bar{\mathbf{u}}$ is velocity vector, $\bar{E}$ is total energy per unit volume, defined as

$$\bar{E} \; = \; \bar{\rho}(\frac{1}{2}\bar{U}^2 \; + \; \bar{e}), \tag{2.4}$$

and, $\bar{U}^2 = \bar{u}^2 + \bar{v}^2 + \bar{w}^2$ is the square of the magnitude of vector velocity.

In the above, $\bar{e}$ is the specific internal energy, given by a *caloric* equation of state,

$$\bar{e} = \bar{e}(\bar{\rho}, \bar{p}). \tag{2.5}$$

Assuming an *ideal gas*, specific internal energy $\bar{e}$ can be expressed in terms of pressure $\bar{p}$ and density $\bar{\rho}$ as

$$\bar{e} = \frac{\bar{p}}{(\gamma - 1)\bar{\rho}}. \tag{2.6}$$

For ideal gas, another important quantity: the speed of sound, $\bar{c}$, is given as follows

$$\bar{c} = \sqrt{\gamma \frac{\bar{p}}{\bar{\rho}}}. \tag{2.7}$$

2.2 Euler Equations in Vector Form

The system of equations is usually represented in vector form, in order to facilitate analysis. For the Euler equations shown in section 2.1, expanding gradient and divergent operators and collecting terms results in the vector form of the Euler equations. The final result reads

$$\frac{\partial \bar{W}}{\partial \bar{t}} + \frac{\partial \bar{f}}{\partial \bar{x}} + \frac{\partial \bar{g}}{\partial \bar{y}} + \frac{\partial \bar{h}}{\partial \bar{z}} = 0. \tag{2.8}$$

where

$$\bar{W} = \begin{Bmatrix} \bar{\rho} \\ \bar{\rho}\bar{u} \\ \bar{\rho}\bar{v} \\ \bar{\rho}\bar{w} \\ \bar{E} \end{Bmatrix}, \quad \bar{f} = \begin{Bmatrix} \bar{\rho}\bar{u} \\ \bar{\rho}\bar{u}^2 + \bar{p} \\ \bar{\rho}\bar{u}\bar{v} \\ \bar{\rho}\bar{u}\bar{w} \\ \bar{\rho}\bar{u}\bar{h}_t \end{Bmatrix}, \quad \bar{g} = \begin{Bmatrix} \bar{\rho}\bar{v} \\ \bar{\rho}\bar{u}\bar{v} \\ \bar{\rho}\bar{v}^2 + \bar{p} \\ \bar{\rho}\bar{v}\bar{w} \\ \bar{\rho}\bar{v}\bar{h}_t \end{Bmatrix}, \quad \bar{h} = \begin{Bmatrix} \bar{\rho}\bar{w} \\ \bar{\rho}\bar{u}\bar{w} \\ \bar{\rho}\bar{v}\bar{w} \\ \bar{\rho}\bar{w}^2 + \bar{p} \\ \bar{\rho}\bar{w}\bar{h}_t \end{Bmatrix}. \tag{2.9}$$

In here, $\bar{h}_t$ is the total specific enthalpy, related to the dependent variables as follows

$$\bar{h}_t = \frac{\bar{E} + \bar{p}}{\bar{\rho}}. \tag{2.10}$$

In this dimensional form, $\bar{W}$ is the solution vector for conservative variables, and $\bar{f}$, $\bar{g}$, $\bar{h}$ are flux vectors in Cartesian coordinates.

2.3 Nondimensionalization

Traditionally, fluid dynamics equations are cast into nondimensional form. Nondimensionalization makes flow variables normalized, so that their values are likely to fall in within reasonably small intervals. In dimensional form, the quantities may not be of the same order of magnitude, and thus can create arithmetic computation errors. In addition to that, the appropriate choice of nondimensional quantities can prevent the loss of accuracy in total specific enthalpy and energy terms, due to low speeds. The following reference quantities are proposed in [4]:

$$\begin{aligned} \rho &= \frac{\bar{\rho}}{\rho_r}, & u &= \frac{\bar{u}}{u_r}, & T &= \frac{\bar{T}}{T_r}, & p &= \frac{\bar{p}}{\rho_r u_r^2}, \\ L &= \frac{\bar{L}}{L_r}, & t &= \frac{\bar{t}}{L_r/u_r}, & e_t &= \frac{\bar{e_t}}{h_r}, & h_t &= \frac{\bar{h}_t}{h_r}. \end{aligned}$$

With the use of the above reference values, a nondimensional vector form of the Euler equations is obtained, similar to equation (2.8):

$$\frac{\partial W}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = 0, \tag{2.11}$$

The traditional way where non-dimensional expressions for w, f, g, h look the same as those given in (2.11) with non-dimensional values used for all variables to nondimensionalize the Euler equations is to use velocity as a reference value for nondimensionalization of energy and enthalpy terms. In contrast to the traditional way, the reference enthalpy is chosen here to nondimensionalize those terms.

The ratio of two different reference values is defined and named as the Eckert number (which is a very significant dimensionless number in compressible flow [10]). The introduction of the Eckert number changes slightly the nondimensional governing equations. The Eckert number is defined as

$$E_c = \frac{u_r^2}{h_r}. \tag{2.12}$$

The reference specific enthalpy, for calorically perfect gas, is written as

$$h_r \;=\; C_p T_r. \tag{2.13}$$

The expression of a C_p, for calorically perfect gas, is substituted into (2.13) to obtain h_r as function of T_r as follows

$$h_r \;=\; \frac{\gamma\ R\ T_r}{\gamma - 1}. \tag{2.14}$$

Now another useful relation for Eckert number is obtained by using Eq. (2.14) plus a reference speed of sound relation, $c_r^2 \;=\; \gamma R T_r$:

$$E_c \;=\; (\gamma - 1) M_r^2. \tag{2.15}$$

In here, $M_r \;=\; u_r / c_r$ is a reference Mach number.

Now, non-dimensional total specific energy and enthalpy can be obtained in terms of Eckert number and the reference Mach number. Starting from the well known relations for dimensional total specific energy and enthalpy:

$$\bar{e}_t \;=\; \frac{\bar{p}}{(\gamma - 1)\bar{\rho}} \;+\; \frac{\bar{u}^2 + \bar{v}^2 + \bar{w}^2}{2}, \tag{2.16}$$

$$\bar{h}_t \;=\; \bar{e}_t \;+\; \frac{\bar{p}}{\bar{\rho}}, \tag{2.17}$$

and dividing by the reference enthalpy h_r , the nondimensional total specific energy and enthalpy are obtained:

$$e_t \;=\; e \;+\; E_c\ \phi, \tag{2.18}$$

where

$$\phi \;=\; \frac{u^2 + v^2 + w^2}{2}, \qquad\qquad e \;=\; \frac{p}{\rho} M_r^2.$$

In a similar way, nondimensional total specific energy h_t can be found as

$$h_t \;=\; e_t \;+\; E_c \frac{p}{\rho}. \tag{2.19}$$

Another useful version of the dimensional equations (2.16) (2.17), introducing the speed of sound, reads:

$$\bar{e}_t \;=\; \frac{\bar{U}^2}{2} \;+\; \frac{\bar{c}^2}{\gamma\,(\gamma-1)}, \tag{2.20}$$

$$\bar{h}_t \;=\; \frac{\bar{U}^2}{2} \;+\; \frac{\bar{c}^2}{\gamma-1}. \tag{2.21}$$

In order to nondimensionalize this version of the equations, one follows the same procedure. As a result, the nondimensional total specific energy and enthalpy can be written as follows

$$e_t \;=\; M_r^2 \left(\frac{c^2}{\gamma} \;+\; (\gamma-1)\,\frac{1}{2}\,U^2 \right), \tag{2.22}$$

$$h_t \;=\; M_r^2 \left(c^2 \;+\; (\gamma-1)\,\frac{1}{2}\,U^2 \right), \tag{2.23}$$

where $U^2 = u^2 + v^2 + w^2$.

Note that as Eckert number approaches zero, the thermodynamic relations become $h_t \approx e_t \approx e$. This result is consistent with incompressible flow.

2.4 Curvilinear Transformation

In real world applications, geometries are too complex for rectangular grids: body-fitted coordinates, requiring transformation from Cartesian to curvilinear coordinates, are normally used. In contrast to Cartesian grids, curvilinear grids provide an excellent tool to represent general body shapes. The use of curvilinear coordinates requires transforming the governing equations to a somewhat more complicated version. The curvilinear coordinate

transformation can be expressed as follows:

$$\tau = \tau(t), \tag{2.24a}$$

$$\xi = \xi(x, y, z, t), \tag{2.24b}$$

$$\eta = \eta(x, y, z, t), \tag{2.24c}$$

$$\zeta = \zeta(x, y, z, t). \tag{2.24d}$$

The derivation of the transformed governing equations is given in detail in Appendix A. A legitimate coordinate transformation results in the same vector form of the governing equations

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0. \tag{2.25}$$

where

$$Q = J \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{Bmatrix}, \qquad F = J \begin{Bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ \rho h_t U - \xi_t p \end{Bmatrix},$$

$$G = J \begin{Bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ \rho h_t V - \eta_t p \end{Bmatrix}, \qquad H = J \begin{Bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ \rho h_t W - \zeta_t p \end{Bmatrix}. \tag{2.26}$$

and

$$U = \xi_x u + \xi_y v + \xi_z w + \xi_t, \tag{2.27a}$$

$$V = \eta_x u + \eta_y v + \eta_z w + \eta_t, \tag{2.27b}$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t. \tag{2.27c}$$

In addition to the above relations, the Jacobian of the transformation , J , and metrics relations are given in Appendix A.

2.5 Primitive Variable Formulation

The choice of dependent variables is somewhat arbitrary for a conservative finite volume scheme. In contrast to conserved variables, primitive or other sets of variables can be chosen as alternatives. There are some advantages to using primitive variables, such as easy implementation of boundary conditions. The use of primitive variables will be shown in Cartesian coordinates, and can be easily extended to curvilinear coordinates. The modeling equation (2.11) in a Cartesian frame can be written in the following form by introducing primitive variables, w:

$$M \frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = 0. \tag{2.28}$$

In here, $M = \partial W/\partial w$ is the transformation matrix, which can be written in generic form as

$$M = \begin{bmatrix} \frac{\partial W_1}{\partial w_1} & \frac{\partial W_1}{\partial w_2} & \frac{\partial W_1}{\partial w_3} & \frac{\partial W_1}{\partial w_4} & \frac{\partial W_1}{\partial w_5} \\ \frac{\partial W_2}{\partial w_1} & \frac{\partial W_2}{\partial w_2} & \frac{\partial W_2}{\partial w_3} & \frac{\partial W_2}{\partial w_4} & \frac{\partial W_2}{\partial w_5} \\ \frac{\partial W_3}{\partial w_1} & \frac{\partial W_3}{\partial w_2} & \frac{\partial W_3}{\partial w_3} & \frac{\partial W_3}{\partial w_4} & \frac{\partial W_3}{\partial w_5} \\ \frac{\partial W_4}{\partial w_1} & \frac{\partial W_4}{\partial w_2} & \frac{\partial W_4}{\partial w_3} & \frac{\partial W_4}{\partial w_4} & \frac{\partial W_4}{\partial w_5} \\ \frac{\partial W_5}{\partial w_1} & \frac{\partial W_5}{\partial w_2} & \frac{\partial W_5}{\partial w_3} & \frac{\partial W_5}{\partial w_4} & \frac{\partial W_5}{\partial w_5} \end{bmatrix}, \tag{2.29}$$

Choosing w as $w = \{\rho, u, v, w, p\}^T$, the conserved variables, $\{\rho\ ,\ \rho u\ ,\ \rho v\ ,\ \rho w\ ,\ E\}^T$ can be expressed in terms of primitive variables as follows

$$W \;=\; \left\{ \begin{array}{c} w_1 \\ w_1\ w_2 \\ w_1\ w_3 \\ w_1\ w_4 \\ w_5\ \frac{E_c}{(\gamma-1)} \;+\; E_c w_1 \frac{(w_2^2+w_3^2+w_4^2)}{2} \end{array} \right\}. \tag{2.30}$$

The elements of transformation matrix (2.29) can then be calculated by differentiating conserved variables in respect to primitive ones. The final form of the transformation matrix reads:

$$M \;=\; \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ w & 0 & 0 & \rho & 0 \\ E_c\ \phi & E_c\ \rho\ u & E_c\ \rho\ v & E_c\ \rho\ w & \frac{E_c}{(\gamma-1)} \end{array} \right]. \tag{2.31}$$

The use of primitive variables in curvilinear coordinates may be extended to the transformed governing equations. The final result reads:

$$M\ \frac{\partial q}{\partial \tau} \;+\; \frac{\partial F}{\partial \xi} \;+\; \frac{\partial G}{\partial \eta} \;+\; \frac{\partial H}{\partial \zeta} \;=\; 0. \tag{2.32}$$

where $q \;=\; J\{\rho\ ,\ u\ ,\ v\ ,\ w\ ,\ p\}^T$.

CHAPTER III

FLUX DIFFERENCE SPLITTING

Typical solution methods for conservation laws can be divided into two categories: flux approach and wave approach. The flux approach deals with only flux, whereas the wave approach models the flux and waves. In contrast to the flux approach, the wave approaches is more accurate but expensive [11].

Wave approaches are mainly split up into two subcategories: flux vector splitting, and reconstruction-evolution approaches. With flux vector splitting, one simply attempts to divide the flux into negative and positive components, by using various methods which are available in the literature. The reconstruction approach calculates the solution over the cell by an appropriate averaging procedure, then advancing to the next time step. This method is usually called flux difference splitting when the time evolution uses Roe's approximate Riemann solver [11].

3.1 The Riemann Problem

A hyperbolic conservation law with discontinuous piecewise constant data as initial conditions known as a Riemann problem. The Riemann problem remains important for solving fluid dynamics equations, because the governing equations include discontinuous solutions. A one-dimensional hyperbolic conservation law is written as

$$\frac{\partial Q}{\partial t} + A\frac{\partial Q}{\partial x} = 0. \tag{3.1}$$

with initial discontinuous (constant) data set:

$$Q(x,0) = \begin{cases} Q_L, & x < 0, \\ Q_R, & x > 0. \end{cases} \tag{3.2}$$

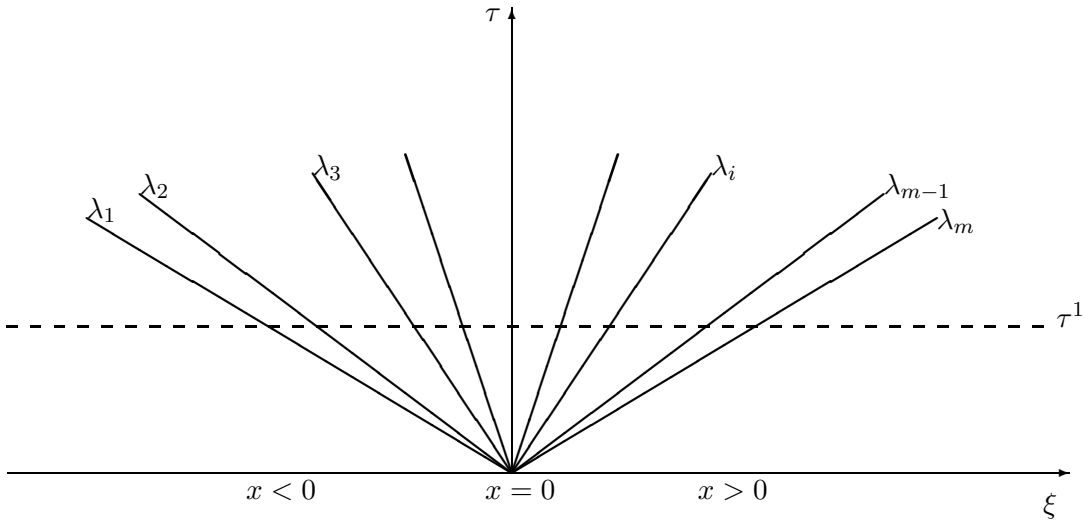

Figure 3.1: Structure of solution of the Riemann problem in the $\xi - \tau$ plane

The governing equation (3.1) and the above initial data form the Riemann problem. The flux Jacobian A has real and distinct eigenvalues, because it is assumed that the system is strictly hyperbolic. Its eigenvalues are ordered as follows

$$\lambda_1 < \lambda_2 < .... < \lambda_m.$$

The solution of the Riemann problem for *constant coefficient* system is explicitly known, and is very useful as a starting point for solving the nonlinear Riemann problem. The hyperbolic equations create a time-marching problem. In the following, it will be convenient to show the structure of the solution of the Riemann problem in the $x - t$ plane (see Fig. (3.1)).

Introducing right eigenvectors R of the constant flux Jacobians, A, into Eq. (3.1) and defining characteristic variables as $W = R^{-1}Q$, the system can be written as m scalar Riemann problems:

$$\frac{\partial W_i}{\partial t} + \lambda_i \frac{\partial W_i}{\partial x} = 0. \tag{3.3}$$

The solution is obtained for each scalar Riemann problem as follows

$$W_i(x,t) = W_i^0(x-\lambda_i t) = \begin{cases} \alpha_i, & x-\lambda_i t < 0, \\ \beta_i, & x-\lambda_i t > 0. \end{cases} \tag{3.4}$$

The constants α_i and β_i are initial data for the characteristic variables. The characteristic variables can be easily converted to dependent variables using right eigenvectors, as $Q = RW$, and the general solution of the Riemann problem becomes

$$Q(x,t) = \sum_i^m W_i^0(x-\lambda_i t)R^{(i)}. \tag{3.5}$$

The general solution can be written in different form by introducing p which is maximum index such as the condition $x - \lambda_i t > 0$ is satisfied and using the equation (3.4) results:

$$Q(x,t) = \sum_{i=1}^{p} \beta_i R^{(i)} + \sum_{i=p+1}^{m} \alpha_i R^{(i)}. \tag{3.6}$$

The solution can be now found at the next time step, τ_1, as seen in Fig. (3.1). As an illustration, the solution between waves is given in detail:

$$\frac{x}{t} < \lambda_1 \quad (p=0) \quad Q(x,t) = Q_L = \alpha_1 R^{(1)} + \alpha_2 R^{(3)} + ... + \alpha_{m-1} R^{(m-1)} + \alpha_m R^{(m)}, \tag{3.7a}$$

$$\lambda_1 < \frac{x}{t} < \lambda_2 \quad (p=1) \quad Q(x,t) = \beta_1 R^{(1)} + \alpha_2 R^{(3)} + ... + \alpha_{m-1} R^{(m-1)} + \alpha_m R^{(m)}, \tag{3.7b}$$

.....

$$\lambda_{m-1} < \frac{x}{t} < \lambda_m (p=m-1) Q(x,t) = \beta_1 R^{(1)} + \beta_2 R^{(2)} + ... + \beta_{m-1} R^{(m-1)} + \alpha_m R^{(m)}, \tag{3.7c}$$

$$\lambda_m < \frac{x}{t} \quad (p=m) \quad Q(x,t) = Q_R = \beta_1 R^{(1)} + \beta_2 R^{(3)} + ... + \beta_{m-1} R^{(m-1)} + \beta_m R^{(m)}. \tag{3.7d}$$

The jump in Q across all the waves can be stated as

$$\triangle Q = Q_R - Q_L = \sum_{i=1}^{m} \tilde{\alpha}_i R^{(i)}, \tag{3.8}$$

where $\tilde{\alpha_i}$ is called strength of the wave.

$$\tilde{\alpha_i} = \beta_i - \alpha_i. \tag{3.9}$$

The jump in Q is useful for solving the Riemann problem. The Riemann problem is investigated by Le Veque [12] and most recently by Toro [13]

3.2 Roe Flux Formulation

An effective approximate Riemann solver is proposed by Roe [14]. The Roe flux difference splitting algorithm employs an approximate solution for the Riemann problem in order to develop a formula for the numerical flux. A significant amount of work is eliminated if a linear approximation of the exact Riemann problem can be used. The one dimensional conservation-law equation (3.1) can be used to illustrate an procedure, where $A = \partial f/\partial Q$, is called the flux Jacobian.

Roe [14] attempted to solve an approximate Riemann problem instead of the exact one in Eq. (3.1) by replacing the true Jacobian matrix A with a *constant* Jacobian matrix $\tilde{A} = \tilde{A}(Q_L, Q_R)$. The Roe-averaged matrix $\tilde{A}$ is a function of the left state Q_L and the right state Q_R, which should be chosen so that a solution of the linear Riemann problem becomes an approximate solution of the nonlinear Riemann problem. Then, the original nonlinear equation is replaced by the (approximate) linear Riemann problem, as follows

$$\frac{\partial Q}{\partial t} + \tilde{A}\frac{\partial Q}{\partial x} = 0. \tag{3.10}$$

The linear Riemann problem was solved in the previous section. If we use the result of Eqs. (3.8) and (3.4), the solution $Q_{i+\frac{1}{2}}(x/t)$ can be evaluated in one of the following form:

$$Q_{i+\frac{1}{2}}(x/t) = Q_L + \sum_{\lambda_i<0} \tilde{\alpha}_i \tilde{R}^{(i)}, \tag{3.11}$$

or

$$Q_{i+\frac{1}{2}}(x/t) = Q_R - \sum_{\lambda_i>0} \tilde{\alpha}_i \tilde{R}^{(i)}, \tag{3.12}$$

where $i + \frac{1}{2}$ denotes face.

The above solution can be obtained provided that a *reasonable* approximate Jacobian matrix $\tilde{A}$ is determined. The choice of a legitimate Roe-averaged matrix $\tilde{A}$ depends on the following conditions, proposed by Roe : in [14]

(i) *It constitues a linear mapping from the vector space* Q *to the vector space* f .

(ii) *As* $Q_L - Q_R \to Q \quad \tilde{A}(Q_L, Q_R) \to A(Q) \quad$ *where* $A = \partial f / \partial Q$.

(iii) *For any* Q_L , Q_R , $\quad \tilde{A}(Q_L, Q_R) \cdot (Q_L - Q_R) = f_L - f_R$.

(iv) *The eigenvectors of* $\tilde{A}$ *are linearly independent.*

The conditions (i) and (iv) are simply the hyperbolicity requirements. This implies that the Roe matrix $\tilde{A}$ is required to have real eigenvalues, $\tilde{\lambda}_i = \tilde{\lambda}_i(Q_L, Q_R)$ and a complete set of linearly independent right eigenvectors, $\tilde{R}^{(i)}$, corresponding to $\tilde{\lambda}_i$. Property (ii) ensures consistency with the conservation laws.

The approximate Riemann problem can be interpreted as a modified conservation law:

$$\frac{\partial Q}{\partial t} + \frac{\partial \tilde{f}(\tilde{Q})}{\partial x} = 0, \tag{3.13}$$

where $\tilde{f} = \tilde{A}Q$ is the modified flux.

In order to be conservative, (see [12]), the modified equation flux must be related to the original Riemann problem flux, as follows

$$\tilde{f}(Q_R) - \tilde{f}(Q_L) = f(Q_R) - f(Q_L). \tag{3.14}$$

Integrating the approximate solution, $Q_{i+\frac{1}{2}}(0)$, in control a volume results in a numerical flux formulation [12] [13] as:

$$f_{i+\frac{1}{2}} = \tilde{f}(Q_{i+\frac{1}{2}}(0)) + f(Q_R) - \tilde{f}(Q_R). \tag{3.15}$$

Using $\tilde{f} = \tilde{A}Q$, one gets the following numerical flux expression:

$$f_{i+\frac{1}{2}} = \tilde{A}Q_{i+\frac{1}{2}}(0) + f(Q_R) - \tilde{A}Q_R. \tag{3.16}$$

The use of Eq. (3.12) allows us to write numerical flux in the following form:

$$f_{i+\frac{1}{2}} = f(Q_R) - \tilde{A} \sum_{\tilde{\lambda}_i > 0} \tilde{\alpha}_i \tilde{R}^{(i)} = f(Q_R) - \sum_{i=1}^{m} \tilde{\lambda}_i^{+} \tilde{\alpha}_i \tilde{R}^{(i)}. \tag{3.17}$$

Alternatively, numerical flux can be evaluated by using the relation (3.14) and Eq. (3.11) as:

$$f_{i+\frac{1}{2}} = f(Q_L) + \tilde{A} \sum_{\tilde{\lambda}_i < 0} \tilde{\alpha}_i \tilde{R}^{(i)} = f(Q_L) + \sum_{i=1}^{m} \tilde{\lambda}_i^{-} \tilde{\alpha}_i \tilde{R}^{(i)}, \tag{3.18}$$

where $\tilde{\lambda}_i^{-}$ and $\tilde{\lambda}_i^{+}$ are negative and positive eigenvalues, respectively. The third kind of expression for numerical flux can be obtained by averaging Eqs. (3.17) and (3.18), as follows:

$$f_{i+\frac{1}{2}} = \frac{1}{2}\Big[f(Q_R) + f(Q_L) - \sum_{i=1}^{m} |\tilde{\lambda}_i| \tilde{\alpha}_i \tilde{R}^{(i)} \Big]. \tag{3.19}$$

Recall that the jump in Q across all the waves was given in Eq. (3.8). Then, the numerical flux takes the following form, (see [15] for details):

$$f_{i+\frac{1}{2}} = \frac{1}{2}\Big[f(Q_R) + f(Q_L) - |\tilde{A}| \triangle Q \Big], \tag{3.20}$$

where

$$\triangle Q = Q_R - Q_L,$$

$$|\tilde{A}| = \tilde{A}^{+} - \tilde{A}^{-} = \tilde{R} |\tilde{\Lambda}| \tilde{R}^{-1},$$

with $\tilde{\Lambda} = diag(\ |\tilde{\lambda_1}|, |\tilde{\lambda_2}|, ..., |\tilde{\lambda_m}|\ )$.

In a practical application, the true Jacobian matrix, thus its eigenvectors, can easily be obtained for any hyperbolic system. The numerical flux is readily evaluated once dependent variables are replaced with the corresponding Roe average variables in the true Jacobian matrix, A, and its eigenvectors, R, according to Eq. (3.20). The remaining problem is to determine suitable Roe average variables via constructing the Roe average matrix, $\tilde{A}$.

3.2.1 Roe average variables for the Euler Equations

Constructing a matrix $\tilde{A}$ which satisfies the conditions mentioned is difficult, mostly due to requirement (*iii*): the obvious candidates $\tilde{A} = \frac{1}{2}(A_L + A_R)$ or $\tilde{A} = A(\frac{1}{2}(Q_L + Q_R))$ do not meet this condition.

However, a solution can be found, and results in $\tilde{A}$ having the same functional form as true Jacobian, A, with

$$\tilde{u} = \frac{\sqrt{\rho_L}\, u_L + \sqrt{\rho_R}\, u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \tag{3.21a}$$

$$\tilde{h}_t = \frac{\sqrt{\rho_L}\, h_{t_L} + \sqrt{\rho_R}\, h_{t_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \tag{3.21b}$$

$$\tilde{e}_t = \frac{\sqrt{\rho_L}\, e_{t_L} + \sqrt{\rho_R}\, e_{t_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}. \tag{3.21c}$$

Note that $\tilde{u}$ and *either* $\tilde{h}_t$ or $\tilde{e}_t$ are the only two variables that appear in $\tilde{A}$. In addition, a Roe averaged density variable can be introduced for convenience, as

$$\tilde{\rho} = \sqrt{\rho_L \rho_R}. \tag{3.22}$$

The Roe average speed of sound can be found by using either equations (3.21b) and (2.23) or (3.21c) and (2.22), with Eq. (3.21a)

$$\tilde{c}^2 = \frac{(\gamma-1)\tilde{h}_t}{E_c} - \frac{\gamma - 1}{2}\,\tilde{u}^2 \quad or \quad \tilde{c}^2 = \frac{\gamma(\gamma-1)}{E_c}\,\tilde{e}_t - \frac{\gamma\,(\gamma-1)}{2}\,\tilde{u}^2. \tag{3.23}$$

Moreover, using the speed of sound relation $c^2 = p\gamma/\rho$ leads one to define the average pressure as

$$\tilde{p} = \frac{\tilde{\rho}\,\tilde{c}^2}{\gamma}. \tag{3.24}$$

3.3 Primitive Variable Flux Formulation

The interface fluxes were evaluated in section 3.2 by means of conserved variables. The use of the primitive variable transformation, $M = \partial Q/\partial q$, allows one to write Eq. (3.1) in

a primitive variable formulation for conservation laws, as follows

$$M\ \frac{\partial q}{\partial x}\ +\ A\ M\ \frac{\partial q}{\partial t}\ =\ 0. \tag{3.25}$$

Multiplying Eq. (3.25) by M^{-1} and defining $a = M^{-1}AM$ gives a primitive variable formulation for conservation laws, as follows

$$\frac{\partial q}{\partial t}\ +\ a\frac{\partial q}{\partial x}\ =\ 0. \tag{3.26}$$

In particular, the two matrices, a and A , are said to be similar because they satisfy the relation $a = M^{-1}AM$. The matrix M is the transformation matrix, and the relationship between a and A is called a similarity transformation. The important point here is that similar matrices have the same eigenvalues. Similarity matrices have the property that if R_Q is an eigenvector of A, then $R_q = M^{-1}R_Q$ is the eigenvector of a corresponding to that same eigenvalue. The following steps prove the statement above:

$$\begin{aligned}
[A]\ \{R_Q\} &= \lambda\ \{R_Q\},\\
[M]^{-1}\ [A]\ \{R_Q\} &= \lambda\ [M]^{-1}\ \{R_Q\},\\
[M]^{-1}\ [A]\ [M]\ [M]^{-1}\{R_Q\} &= \lambda\ [M]^{-1}\ \{R_Q\},\\
[a]\ [M]^{-1}\ \{R_Q\} &= \lambda\ [M]^{-1}\{R_Q\},\\
[a]\{R_q\} &= \lambda\ \{R_q\}.
\end{aligned} \tag{3.27}$$

In the above, λ is an eigenvalue of matrix $[a]$ (as well as $[A]$), and $\{R_q\} = [M]^{-1}\{R_Q\}$ is the eigenvector of $[a]$ corresponding to the eigenvalue λ.

The matrix A is diagonalizable, due to hyperbolicity as $A = R_Q\Lambda R_Q^{-1}$. The use of the similarity transformation property allows to write the Jacobian matrix A as follows

$$A\ =\ M\ R_q\ \Lambda\ R_q^{-1}\ M^{-1}. \tag{3.28}$$

In addition to the above relation, an approximation to $\triangle Q$ can be made in terms of $\triangle q$ as

$$M \;=\; \frac{\partial Q}{\partial q} \qquad ; \qquad M\ \triangle q \;\cong\; \triangle Q. \tag{3.29}$$

Note that the relationship in Eq. (3.29) is *exact* when one uses Roe averages for the matrix M ($\tilde{M}\triangle q = \triangle Q$). The relation (3.28) and approximation of $\triangle Q$ in (3.29) allow to use the same numerical flux form Eq.(3.20) for primitive variables. Eq.(3.20) can now be rewritten as follows:

$$f_{i\,+\,\frac{1}{2}} \;=\; \frac{1}{2}(f_L \;+\; f_R) \;-\; \frac{1}{2}|\tilde{A}|\ \tilde{M}\triangle q. \tag{3.30}$$

The following relation is easily written from Eq.(3.28)

$$\begin{aligned} |\tilde{A}|\ \tilde{M} \;&=\; (\tilde{M}\ \tilde{R}_q\ |\tilde{\Lambda}|\ \tilde{R}_q^{-1}\ \tilde{M}^{-1}\ )\tilde{M}, \\ &=\; \tilde{M}(\tilde{R}_q\ |\tilde{\Lambda}|\ \tilde{R}_q^{-1}) \;=\; \tilde{M}\ |\tilde{a}|, \end{aligned} \tag{3.31}$$

where $|\tilde{a}| \;=\; \tilde{R}_q\ |\tilde{\Lambda}|\ \tilde{R}_q^{-1}$.

The primitive variable numerical flux can then be cast in the following form:

$$f_{i\,+\,\frac{1}{2}} \;=\; \frac{1}{2}(f_L \;+\; f_R) \;-\; \frac{1}{2}\tilde{M}\ |\tilde{a}|\ \triangle q. \tag{3.32}$$

As it is pointed out in [7], the traditional way of constructing R_Q usually uses the relation $R_Q \;=\; MR_q$. This primitive flux formulation is slightly simpler than the Roe flux formulation in conservative variables.

CHAPTER IV

PRECONDITIONING

The steady-state Euler equations are classified as elliptic in a subsonic flowfield and hyperbolic in supersonic regions. In order to overcome the difficulties with the mixed nature of the equations, the unsteady form is typically used for solving either unsteady or steady flowfields. In this case, hyperbolicity is guaranteed for all Mach numbers. Over the years, hyperbolic PDE solvers for fluid dynamics equations have been successfully developed and validated at medium to high Mach numbers. Applications for low-speed flows that are still compressible (i.e. combustion problems) have lagged behind, however. Moreover, attempts to extend compressible algorithms to low-speed problems ($M_\infty < 0.1$) and reach a *unified* approach have met some difficulties.

For compressible flow, the pressure term is a thermodynamic variable, which should depend only on local instantaneous values of density and internal energy. For incompressible flow, pressure is no longer a thermodynamic variable, and only depends on the velocity field [16]. Unlike the coupled compressible Euler equations, the incompressible equations drop the time derivative in the continuity equation and leave pressure only in the momentum equations. This is the main difficulty for solving incompressible equations. Traditionally, these uncoupled equations are solved by iterative methods [17] [18]. The pressure term can be obtained by taking the divergence of the momentum equations. The solution of the velocity field from the momentum equations is checked based on whether it satisfies a divergent-free condition. If it does not satisfy this condition, some corrections on pressure terms are made until the continuity equation is satisfied. These type of solutions are called pressure-correction methods and are expensive, due to the need of for a divergent-free velocity field. Another main approach, known as artificial compressibility, is to add pressure time derivative to the continuity equation, and couple the continuity and momentum equations[19]. The system becomes hyperbolic, and a time-marching scheme can be applied.

The unmodified unsteady Euler equations are singular as the Mach number approaches zero. At first, it was attempted to reformulate these equations in a nonsingular low Mach number formulation (see Briley, McDonald and Shamroth [4]). Later, Turkel [20] introduced a pressure time derivative to the momentum equation, extending Chorin's idea. If these equations are written in vector form, this is equivalent to introducing a matrix that premultiplies time derivatives. This opened a new approach for solving incompressible flow, which evolved into preconditioning methods.

The preconditioned methods have gained popularity over other methods designed for incompressible flow. A variety of flowfields are not purely incompressible or compressible, but of mixed types. Therefore, an arbitrary Mach number formulation is highly desirable.

4.1 Local Preconditioning Methods

Traditional compressible codes fail at low Mach number in terms of accuracy and convergence. In very low speed flows, the hyperbolic system will have a large disparity among eigenvalues, which causes serious problems in time-marching schemes. The time step size is limited by the propagation of the fastest acoustic wave ($u \pm c$), whereas the propagation of errors leaving the domain is restricted by the slowest wave, which is the entropy wave u in low-speed flows. Therefore, the ratio of the fastest and slowest eigenvalues (called the *condition number*) eventually governs the convergence rate [21]. One is forced to use a very small time step size, which makes the equation slow to convergence for time-marching schemes (either explicit or implicit). It is worth mentioning that approximate factorization (AF) errors become dominant at low Mach numbers in implicit schemes [22].

As accuracy is concerned, improperly scaled compressible equations can not represent well the virtually incompressible behavior at low speeds[4]. In order to see that, modified equation for the discretized algorithm can be used to show unevenly matched terms are present as $M \rightarrow 0$ (see [23], [5], [2]).

Local preconditioning is introduced to the time derivatives in order to alter the characteristics of the system of equations, thus equalizing eigenvalues The change does not effect the steady-state solution, only the convergence rate; however, it destroys time

accuracy for a time-dependent problem. This drawback can be circumvented in several ways. The first remedy is to use dual-time stepping (introducing a pseudo time derivative into the governing equations). This method has inner iteration in pseudo time, whereas the outer loop is advancing in physical time [24]. The other choice is a Newton formulation, adding preconditioning to the numerical terms, while converging to an unsteady residual which has the physical time term. The latter approach is taken as the solution method in this study.

4.1.1 Advantages of Local Preconditioning

Local preconditioning is designed to alleviate the problems due to low Mach numbers. The most important benefit of preconditioning is the removal of the stiffness, obtained by balancing the eigenvalues. As already mentioned, the condition number governing the convergence rate becomes larger as $M \rightarrow 0$. A preconditioner attempts to accelerate the convergence rate, but it also helps with accuracy preservation at low Mach numbers: local preconditioning properly scales compressible equations for $M \rightarrow 0$. A modified equation analysis proves that the order of magnitude of the elements in the equations is matched, including artificial viscosity elements, as $M \rightarrow 0$ [23].

Local preconditioning makes the Euler equations behave as a set of scalar equations, which is true also for the discretized equation. This property facilitates applying other acceleration techniques for low Mach number flows, such as multi-grid and residual smoothing [5]. Another use for preconditioning is to decouple the unsteady Euler equations, which is impossible to do in their original form. This will allow one to develop truly multi-dimensional discretizations [2] [5].

4.1.2 Issues with Local Preconditioning

Local preconditioning comes with some disadvantages. The various preconditioners yield the same optimal condition number. Nonetheless, numerical performance appears to be varying, somewhat distinctively [5]. Darmofal and Schmid [6] prove that the orthogonality of eigenvectors of a preconditioned system is crucial for the determination of transient amplification of errors, which may lead to an instability problem at the stagnation point.

Preconditioning is sensitive at the stagnation point, where flow is highly deflected and decelerated. Lee [5] attributed this instability to flow angle sensitivity and nearly parallel eigenvector of preconditioned equations. Note that the original equations do not suffer from the above problems. Another problem is the vorticity production near the stagnation point. The vorticity equation is obtained from the momentum equation: it results in vorticity being only convected, without production, destruction, or diffusion, in incompressible flows. In spite of that, preconditioned formulations may introduce some extra terms, which result in production and diffusion of vorticity.

Benefits and problems of local preconditioning have been briefly mentioned here. More detailed analysis is found in the Ph.D theses [2] and [5]. It is hard to have a local preconditioner which has all the advantages and overcomes all the problems at the same time. The perfect preconditioning algorithm seems impossible to find due to conflicts between the many required criteria. Detailed work by Zaccanti [25] shows that there is no *ideal* local preconditioning method.

4.2 Global Preconditioning

The use of global preconditioning, which adds a diagonal matrix to the isoenergic equations, was first put forward by Briley, McDonald and Shamroth [4]. In contrast to local preconditioning, global preconditioners are based on reference values, which are problem-dependent but are consistent throughout the flowfield. Recently, a constant preconditioning matrix was added to the full Navier-Stokes (N-S) equations by Briley, Taylor, and Whitfield [7].

The design of global preconditioning is based solely on rescaling the eigenvalues, reducing their disparity. Despite the fact that it is based on this sole criteria, the new global preconditioning, which is positive-symmetric and has zero elements except on the main diagonal, emerged as a reliable tool. The simple structure of the preconditioned matrix makes it simple to implement. Moreover, this new global preconditioner does not alter the momentum equations, thus it has no vorticity production or diffusion, but does have rotational invariance.

4.3 Preconditioned Formulation in Primitive Variables

The new preconditioning matrix is incorporated into the governing equations, cast in primitive variable form. Introducing preconditioning alters the time derivatives and changes the characteristic of the system of equations. The preconditioned matrix for 3-D problems is defined as follows:

$$\Gamma_q \;=\; Diag\;(1\;,\;1\;,\;1\;,\;1\;,\;\beta), \tag{4.1}$$

where $\beta = Min(1, M_r^2)$ and M_r is the reference Mach number (already encountered when introducing non-dimensionalization).

The matrix Γ_q is introduced into the-one dimensional primitive variable formulation, as follows

$$M\;\Gamma_q^{-1}\;\frac{\partial q}{\partial \tau}\;+\;\frac{\partial F}{\partial \xi}\;=\;0. \tag{4.2}$$

The above equation can be written as follows, after introducing the system matrix a ($a = M^{-1}AM$).

$$M\;\Gamma_q^{-1}\;\frac{\partial q}{\partial \tau}\;+\;M\;a\;\frac{\partial q}{\partial \xi}\;=\;0. \tag{4.3}$$

If Eq.(4.3) is multiplied by M^{-1} and Γ_q , successively, it takes the following form

$$\frac{\partial q}{\partial \tau}\;+\;a_\Gamma\;\frac{\partial q}{\partial \xi} = 0, \tag{4.4}$$

where the preconditioned system matrix $a_\Gamma \;=\; \Gamma_q\;a$ is introduced . The three-dimensional matrices are introduced in Appendix B.

The flux formula that is consistent with the preconditioned formulation shown above was derived by [7]. The final result in three-dimension reads:

$$K_{(\bullet)\;+\;\frac{1}{2}}\;=\;\frac{1}{2}\;(K_L\;+\;K_R)\;-\;\frac{1}{2}\;\tilde{M}\;\Gamma_q^{-1}\;|\tilde{s}_\Gamma|\;\triangle q, \tag{4.5}$$

where

$K \;=\; F\;\;,\;\;s_\Gamma \;=\; a\;\;for\;\;(\bullet)\;=\;i,$

$K \;=\; G\;\;,\;\;s_\Gamma \;=\; b\;\;for\;\;(\bullet)\;=\;j,$

$$K = H \quad , \quad s_\Gamma = c \quad for \quad (\bullet) = k.$$

In this formulation, there is a need to calculate eigenvalues and eigenvectors of the matrix s_Γ, in order to develop characteristic-based flux approximations. All necessary information for building interface fluxes are given in Appendices B and C.

CHAPTER V

NUMERICAL SOLUTION

Implicit numerical solution algorithms for the Euler equations fall largely into two categories: noniterative and iterative methods. The noniterative methods are primarily designed to solve steady-state cases, whereas iterative methods are used for both unsteady and steady-state flows. Incidentally, the main reason for using implicit algorithms is to widen stability constraints and improve convergence rates when compared with explicit schemes. Noniterative methods tend to factor out the system of equations, so that efficient direct matrix solvers can be applicable to each factor; on the other hand , the iterative approach consists of solving linearized implicit approximations iteratively at each time step. Subiterations can be incorporated into the schemes to eliminate factorization and linearization errors. Originally, iterative methods were applied for steady-state solutions. Nonetheless, they turn out to be an efficient approach for unsteady flows as well, because unsteady terms contribute to diagonal dominance [26]. The possible linearization strategies used for iterative implicit schemes can be divided into two main groups: (1) time linearization and (2) Newton-linearization [26]. A Newton scheme naturally accounts for unsteadiness in the residual, and is thus used in the present study.

One motivation of this work was to develop a *single* algorithm for arbitrary flow speeds. The choice of a Newton-linearized preconditioned scheme serves this purpose. Within the Newton algorithm, a nested subiteration eliminates the approximate factorization error when Symmetric Gauss-Seidel relaxation (LU/SGS) is used. As expected, the approximate factorization error starts dominating the unpreconditoned scheme at low Mach numbers and has an adverse effect on the convergence rate (this is the main mechanism responsible for slow convergence rate [22]).

Linearizing the scheme requires computing the flux Jacobian matrices. The analytical expression for flux Jacobians is extremely difficult to obtain, especially for three dimensional

flux-difference algorithms. Instead, numerical calculation of the flux Jacobians is used in this study.

5.1 Time-Linearized Implicit Scheme

As already mentioned, noniterative schemes are primarily used for steady-state flows. A time-linearized implicit scheme is summarized in this section, in order to highlight similarities and differences with the Newton-linearized formulation that is used in this study. The ultimate objective is to solve the following equation.

$$M\ \Gamma_q^{-1}\ \frac{\triangle q^n}{\triangle \tau}\ +\ \delta_i F\ +\ \delta_i G\ +\ \delta_i H\ =\ 0, \tag{5.1}$$

where $\triangle\tau$ is the time step, plus:

$$\begin{aligned}
\triangle q^n &= q^{n+1}\ -\ q^n, \\
\delta_i F &= F_{i\ +\ \frac{1}{2},j,k}\ -\ F_{i\ -\ \frac{1}{2},j,k}, \\
\delta_i G &= G_{i,j\ +\ \frac{1}{2},k}\ -\ G_{i,j\ -\ \frac{1}{2},k}, \\
\delta_i H &= H_{i,j,k\ +\ \frac{1}{2}}\ -\ H_{i,j,k\ -\ \frac{1}{2}}.
\end{aligned}$$

In the above, a first-order discretization of the time derivative has been used for simplicity. An implicit scheme for the equation (5.1) can be written as

$$M\ \Gamma_q^{-1}\ \frac{\triangle q^n}{\triangle \tau}\ +\ \delta_i F^{n+1}\ +\ \delta_j G^{n+1}\ +\ \delta_k H^{n+1}\ =\ 0. \tag{5.2}$$

The fluxes, $F(q^{n+1})$, $G(q^{n+1})$, $H(q^{n+1})$ can be linearized around q^n as

$$\begin{aligned}
q^{n+1} &= q^n\ +\ \triangle q^n, \\
F(q^n + \triangle q^n) &\approx F(q^n)\ +\ \left(\frac{\partial F}{\partial q}\right)^n \triangle q^n = F\ +\ A\triangle q^n, \\
G(q^n + \triangle q^n) &\approx G(q^n)\ +\ \left(\frac{\partial G}{\partial q}\right)^n \triangle q^n = G\ +\ B\triangle q^n, \\
H(q^n + \triangle q^n) &\approx H(q^n)\ +\ \left(\frac{\partial H}{\partial q}\right)^n \triangle q^n = H\ +\ C\triangle q^n.
\end{aligned}$$

A direct substitution of above linearization into the equation (5.1) results in

$$M\Gamma_q^{-1}\triangle q^n = -\triangle\tau\left[\delta_i(F^n \ + \ A\ \triangle q^n) + \delta_j(G^n \ + \ B\ \triangle q^n) + \delta_k(H^n \ + \ C\ \triangle q^n)\right]. \quad (5.3)$$

Rearranging the above equation, one obtains,

$$\left[M\ \frac{\Gamma_q^{-1}}{\triangle\tau} \ + \ \delta_i A \ + \ \delta_j B \ + \ \delta_k C\right]\ \triangle q^n \ = \ -\Re^n, \quad (5.4)$$

where

$$\Re^n \ = \ \delta_i F^n \ + \ \delta_j G^n \ + \ \delta_k H^n.$$

$\Re^n$ is the steady-state residual term. Numerous solution techniques for equation (5.4) are available in the literature [26] [17].

5.2 Newton Formulation

The main deficiency of the residual term appearing in Eq.(5.4) is that it does not contain the time derivative, therefore, it can not represent unsteady cases. Alternatively, a Newton formulation can be developed to solve Eq.(5.2) directly [27]. The Newton formulation has the undesirable feature of being computationally more expensive, however, it can be used for both steady and unsteady cases.

Newton's method can be developed for one-dimensional cases and extended to multi-dimensional cases readily. For simplicity, in the following the Newton formulation is going to be developed for first-order time integration in the one-dimensional case. The 1-D version of the equation (5.2) can be written as a non-linear equation for the variable q^{n+1}, as follows

$$\mathcal{F}(q^{n+1}) \ = \ M\ \Gamma_q^{-1}\frac{q_i^{n+1} \ - \ q_i^n}{\triangle\tau} \ + \ F_{i\ +\ \frac{1}{2}}(q^{n+1}) \ - \ F_{i\ -\ \frac{1}{2}}(q^{n+1}) \ = \ 0, \quad (5.5)$$

where

$$F_{i\ +\ \frac{1}{2}}(q^{n+1}) \ = \ F_{i\ +\ \frac{1}{2}}(q_i^{n+1}, q_{i+1}^{n+1}), \quad (5.6)$$

$$F_{i\ -\ \frac{1}{2}}(q^{n+1}) \ = \ F_{i\ -\ \frac{1}{2}}(q_{i-1}^{n+1}, q_i^{n+1}), \quad (5.7)$$

and a first-order space discretization as used for the time being. From equations (5.6) and (5.7), it is concluded that

$$\mathcal{F}(q^{n+1}) \;=\; \mathcal{F}(q_{i-1}^{n+1}, q_i^{n+1}, q_{i+1}^{n+1}). \tag{5.8}$$

In order to obtain the Newton formulation, $\mathcal{F}(q^{n+1,m+1})$ should be expanded in a Taylor series about $q^{n+1,m}$ to first order terms, where m is Newton iteration index:

$$\begin{aligned}\mathcal{F}\Big(q^{n+1,m+1}\Big) \;=\; & \mathcal{F}(q^{n+1,m}) + \left(\frac{\partial \mathcal{F}}{\partial q_{i-1}^{n+1}}\right)_m \triangle q_{i-1}^{n+1,m} \;+\; \left(\frac{\partial \mathcal{F}}{\partial q_i^{n+1}}\right)_m \triangle q_i^{n+1,m} \;+ \\ & \left(\frac{\partial \mathcal{F}}{\partial q_{i+1}^{n+1}}\right)_m \triangle q_{i+1}^{n+1,m} \;+\; \mathcal{O}[\big(\triangle q^{n+1,m}\big)^2],\end{aligned} \tag{5.9}$$

where

$$\triangle q^{n+1,m} \;=\; q^{n+1,m+1} \;-\; q^{n+1,m}.$$

Note that the current $\triangle$ operator is different from the one used in conjunction with time linearization. The goal of the Newton iteration is to make equation (5.9) zero, therefore the following equation is obtained

$$\left(\frac{\partial \mathcal{F}}{\partial q_{i-1}^{n+1}}\right)_m \triangle q_{i-1}^{n+1,m} + \left(\frac{\partial \mathcal{F}}{\partial q_i^{n+1}}\right)_m \triangle q_i^{n+1,m} + \left(\frac{\partial \mathcal{F}}{\partial q_{i+1}^{n+1}}\right)_m \triangle q_{i+1}^{n+1,m} = -\mathcal{F}(q^{n+1,m}). \tag{5.10}$$

Using equations (5.5), (5.6), and (5.7), one can build the derivatives with respect to dependent variables, as follows

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial q_{i-1}^{n+1}} \;&=\; -\frac{\partial F_{i-\frac{1}{2}}}{\partial q_{i-1}^{n+1}} = -F'_{i-\,\frac{1}{2},i-1}, \\ \frac{\partial \mathcal{F}}{\partial q_i^{n+1}} \;&=\; \frac{M\;\Gamma_q^{-1}}{\triangle \tau} + \frac{\partial F_{i+\frac{1}{2}}}{\partial q_i^{n+1}} - \frac{\partial F_{i-\frac{1}{2}}}{\partial q_i^{n+1}} = \frac{M\;\Gamma_q^{-1}}{\triangle \tau} + F'_{i+\frac{1}{2},i} - F'_{i-\frac{1}{2},i}, \\ \frac{\partial \mathcal{F}}{\partial q_{i+1}^{n+1}} \;&=\; \frac{\partial F_{i-\frac{1}{2}}}{\partial q_{i+1}^{n+1}} = F'_{i+\frac{1}{2},i+1},\end{aligned} \tag{5.11}$$

where the subscript m has been dropped from the derivation, for simplicity. Equation (5.10) then reads:

$$\begin{aligned} -F'_{i-\frac{1}{2}} \triangle q_{i-1}^{n+1,m} \quad &+ \quad \left(\frac{M\Gamma_q^{-1}}{\triangle \tau} + F'_{i+\frac{1}{2}} - F'_{i-\frac{1}{2}} \right) \triangle q_i^{n+1,m} + F'_{i+\frac{1}{2}} \triangle q_{i+1}^{n+1,m} \\ &= \quad - \left[M \frac{(q_i^{n+1,m} - q_i^n)}{\triangle \tau} + \delta_i F(q^{n+1,m}) \right]. \end{aligned} \tag{5.12}$$

In a similar way, a three dimensional form can be obtained, as follows:

$$\begin{aligned} &-F'_{i-\frac{1}{2},i-1} \triangle q_{i-1,j,k}^{n+1,m} - G'_{j-\frac{1}{2},j-1} \triangle q_{i,j-1,k}^{n+1,m} - H'_{k-\frac{1}{2},k-1} \triangle q_{i,j,k-1}^{n+1,m} \\ &+ \left(\frac{M\Gamma_q^{-1}}{\triangle \tau} + F'_{i+\frac{1}{2},i} + G'_{j+\frac{1}{2},j} + H'_{k+\frac{1}{2},k} - F'_{i-\frac{1}{2},i} - G'_{j-\frac{1}{2},j} - H'_{k-\frac{1}{2},k} \right) \triangle q_{i,j,k}^{n+1,m} \\ &+ F'_{i+\frac{1}{2},i+1} \triangle q_{i+1,j,k}^{n+1,m} + G'_{j+\frac{1}{2},j+1} \triangle q_{i,j+1,k}^{n+1,m} + H'_{k+\frac{1}{2},k-1} \triangle q_{i,j,k+1}^{n+1,m} = -\Re_U^{n+1,m}, \end{aligned} \tag{5.13}$$

where the *unsteady* residual has been introduced:

$$\Re_U^{n+1,m} = \left[M \frac{q_{i,j,k}^{n+1,m} - q_{i,j,k}^n}{\triangle \tau} + \delta_i F(q^{n+1,m}) + \delta_j G(q^{n+1,m}) + \delta_k H(q^{n+1,m}) \right].$$

As it is seen in the above equation, the residual, $\Re_U^{n+1,m}$ in Eq. (5.13) contains the time derivative terms, unlike Eq. (5.4), so, this formulation can accommodate unsteady cases as well as steady problems. Incidentally, this algorithm gives rise to a heptagonal solution matrix (band width of seven).

Thus for, the Newton formulation has just developed using a first-order space discretization. When a higher-order formulation is developed, it is obvious that the band width of the solution matrix will increase in proportion to the number of elements used to construct the numerical fluxes. On the other hand, using higher-order flux in the residual with a first-order Newton formulation on the left-hand side causes stability problems, because Jacobians are not evaluated correctly. Alternatively, differentiating fluxes with respect to left, q_L , and right, q_R , states rather than cell values $i, i+1, i-1$ achieves high-order formulation without the above problems [27]. The construction of high-order left and right states is detailed next.

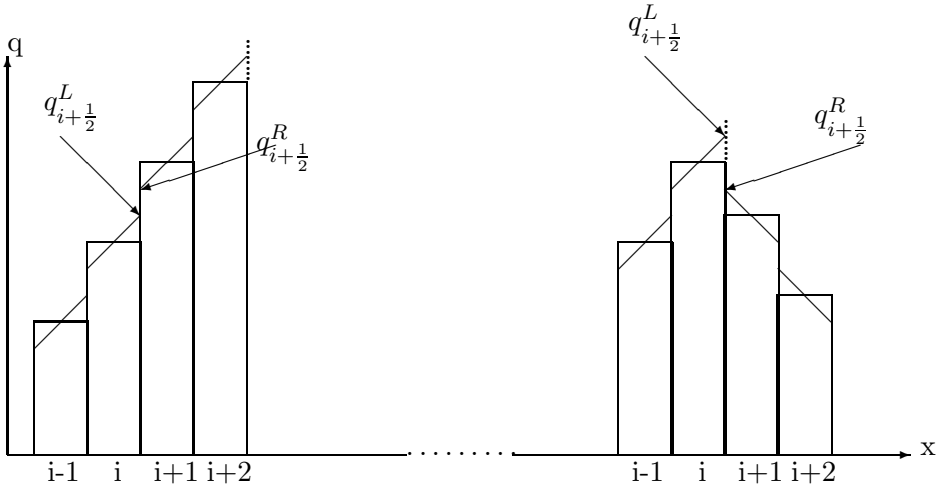

Figure 5.1: High-order solution construction

5.3 Anderson-Thomas-Van Leer Reconstruction-Evolution Methods

Two popular methods are available for creating high-order schemes, namely: *flux averaging* and *solution averaging* techniques. These approaches are investigated in [11]. In this study, the latter one will be used: the solution vector q is averaged, then the fluxes are evaluated. A second and third-order reconstruction can be written in single formulation, by introducing a parameter, ψ, as follows

$$q^L_{i+\frac{1}{2}} = q_i + \frac{1+\psi}{4}(q_{i+1} - q_i) + \frac{1-\psi}{4}(q_i - q_{i-1}), \tag{5.14}$$

$$q^R_{i+\frac{1}{2}} = q_{i+1} - \frac{1+\psi}{4}(q_{i+1} - q_i) - \frac{1-\psi}{4}(q_{i+2} - q_{i+1}). \tag{5.15}$$

Equations (5.14) and (5.15) are leftward-biased and rightward-biased, respectively, for $\psi \leq 1$, and the following values are typically used:

$$\begin{cases} \psi = -1 & \text{Second-order accurate scheme} \\ \psi = \frac{1}{3} & \text{Third-order accurate scheme.} \end{cases}$$

For consistency, the reconstructed solution $q^L_{i+\frac{1}{2}}$ and $q^R_{i+\frac{1}{2}}$ should vary between q_i and q_{i+1}. However, as seen in Fig. (5.1), where two possible configurations of a solution are shown, new maxima and minima can be created by the above formulas (see the case on

the right). This undesired inconsistency can be overcome if the following inequalities are satisfied:

$$q_i \leq q^L_{i+\frac{1}{2}} \leq q_{i+1} \quad and \quad q_i \leq q^R_{i+\frac{1}{2}} \leq q_{i+1}. \tag{5.16}$$

By using equations (5.14) and (5.15), one can prove that $q^L_{i+\frac{1}{2}}$ will lie between q_i and q_{i+1} if $q_{i+1} - q_i$ and $q_i - q_{i-1}$ have the same sign and

$$|q_i \ - \ q_{i-1}| \ \leq \ \frac{3 \ - \ \psi}{1 \ - \ \psi}|q_{i+1} \ - \ q_i|. \tag{5.17}$$

In a similar way, the condition for $q^R_{i+\frac{1}{2}}$ is that $q_{i+2} - q_{i+1}$ and $q_{i+1} - q_i$ have the same sign and

$$|q_{i+2} \ - \ q_{i+1}| \ \leq \ \frac{3 \ - \ \psi}{1 \ - \ \psi}|q_{i+1} \ - \ q_i|. \tag{5.18}$$

In order to satisfy these requirements, limiters are introduced in the extrapolation formulas. High order schemes without limiters can easily be subject to stability problems. The limiter based on the previous discussion is called the *minmod* limiter. Various other limiters are available in the literature. The following high-orderer construction of the solution can be created with the minmod limiter.

$$\begin{aligned} q^L_{i+\frac{1}{2}} \ = \ q_i \quad &+ \quad \frac{1+\psi}{4} \ minmod[q_{i+1} \ - \ q_i \ , \ b \ (q_i \ - \ q_{i-1})] \\ &+ \quad \frac{1-\psi}{4} \ minmod[q_i \quad - \ q_{i-1} \ , \ b \ (q_{i+1} \ - \ q_i)], \qquad (5.19) \\ q^R_{i+\frac{1}{2}} \ = \ q_{i+1} \ &- \quad \frac{1+\psi}{4} \ minmod[q_{i+1} \ - \ q_i \ , \ b \ (q_{i+2} \ - \ q_{i+1})] \\ &- \quad \frac{1-\psi}{4} \ minmod[q_{i+2} \ - \ q_{i+1} \ , \ b \ (q_{i+1} \ - \ q_i)], \qquad (5.20) \end{aligned}$$

where

$$1 \leq b \leq \frac{3-\psi}{1-\psi}. \tag{5.21}$$

Typically, the b value is taken as $\frac{3-\psi}{1-\psi}$. The function $minmod(x, y)$ can be written in the following form:

$$minmod(x, \ y) \ = \ Sgn(x) \ Max\Big\{0 \ , \ Min[|x|, \ y \ Sgn(x)]\Big\}. \tag{5.22}$$

5.4 Linearized Fluxes by Numerical Differentiation

The Newton formulation shown in Eq.(5.13) requires the evaluation of numerical fluxes. Analytical derivation of flux Jacobians is extremely involved for multidimensional cases. Instead, numerical differentiation is employed to evaluate them. The use of numerical differentiation with Newton's method is usually named the Discretized-Newton Iteration method. Numerical flux Jacobians in the ξ direction are written as

$$\hat{A}_i^+ \;=\; \frac{\partial F_{i+\frac{1}{2}}}{\partial q^L_{i+\frac{1}{2}}}, \qquad\qquad \hat{A}_i^- \;=\; \frac{\partial F_{i-\frac{1}{2}}}{\partial q^R_{i-\frac{1}{2}}}. \tag{5.23}$$

Finite differences are used to evaluate numerical Jacobians. For the kth column of the $\hat{A}^+$ Jacobian matrix, for example

$$\hat{A}_i^+ \;= \frac{F_{i+\frac{1}{2}}(q^R_{i+\frac{1}{2}}\;,\; q^L_{i+\frac{1}{2}} \;+\; he_k) \;-\; F_{i+\frac{1}{2}}(q^R_{i+\frac{1}{2}}\;,\; q^L_{i+\frac{1}{2}})}{h}. \tag{5.24}$$

In here , e^k is the kth unit vector and the perturbation can be selected as $h \approx \sqrt{\text{machine zero}}$.

For the three dimensional Newton formulation, $\hat{B}^+$, $\hat{B}^-$, $\hat{C}^+$, $\hat{C}^-$ are introduced in a similar way:

$$\hat{B}_j^+ = \frac{\partial G_{j+\frac{1}{2}}}{\partial q^L_{j+\frac{1}{2}}}, \qquad \hat{B}_j^- = \frac{\partial G_{j-\frac{1}{2}}}{\partial q^L_{j-\frac{1}{2}}}, \qquad \hat{C}_k^+ = \frac{\partial K_{i+\frac{1}{2}}}{\partial q^L_{k+\frac{1}{2}}}, \qquad \hat{C}_k^- = \frac{\partial K_{i-\frac{1}{2}}}{\partial q^L_{k-\frac{1}{2}}}. \tag{5.25}$$

The three dimensional form of the Newton formulation will be the same as in Eq. (5.13) using left , q_L, and right, q_R, states rather than cell values i, $i+1$, $i-1$ for the face fluxes (e.g. $F'_{i-\frac{1}{2},i-1}$ becomes $\hat{A}^+_{i-1}$, etc).

In equation (5.24) above, a vector representation was given for the finite difference calculation. In general, flux Jacobians are calculated column by column for a given perturbation to the dependent variables, in order to lower the cost of evaluation.

5.4.1 Complex Variable Approach

A complex variable approach is an alternative way to calculate derivative of real functions [28]. The flux function is expanded in a Taylor series using complex perturbations, as follows

$$K(q \ + \ ih) \ = \ K(q) \ + \ i \ h \ \frac{\partial K}{\partial q} \ - \ \frac{h^2}{2!} \ \frac{\partial^2 K}{\partial q^2} \ - \ \frac{i \ h^3}{3!} \ \frac{\partial^3 K}{\partial q^3} \ + \ \ldots \tag{5.26}$$

Then, using the imaginary part of the equation (5.26) gives the flux Jacobian, as follows

$$\frac{\partial K}{\partial q} \ = \ \frac{Im[K(q \ + \ i \ h)]}{h} + \mathcal{O}(h^2). \tag{5.27}$$

In contrast to first-order accuracy in the finite difference method, complex variable differentiation has second order accuracy. This approach is traditionally is used when high-order accuracy and step size selection are crucial. Moreover, finite difference methods are prone to roundoff errors when using very small step sizes. In contrast to finite differences, the complex variable method does not involve any roundoff error, thus it is insensitive to step size. The use of complex variables provides a high-order numerical Jacobian which uses very small perturbation with easy modifications in the existing codes. The disadvantage of complex variable approach is the increase in runtime. Newman [28] reported that the runtime can be on the order of three times at of the original solver.

5.5 Solution Algorithm

After the unsteady residual is discretized, and the flux Jacobian matrices are evaluated, following the procedure outlined, one is left with the following system of linear equations:

$$\begin{aligned} -\hat{A}^{+}_{i-1} \triangle q^{n+1,m}_{i-1,j,k} \ - \ \hat{B}^{+}_{j-1} \triangle q^{n+1,m}_{i,j-1,k} \ - \ \hat{C}^{+}_{k-1} \triangle q^{n+1,m}_{i,j,k-1} \ + \ \Bigg(\frac{M \ \Gamma^{-1}_{q}}{\triangle \tau} \ + \\ \hat{A}^{+}_{i} \ - \ \hat{A}^{-}_{i} \ + \ \hat{B}^{+}_{j} \ - \ \hat{B}^{-}_{j} \ + \ \hat{C}^{+}_{k} \ - \ \hat{C}^{-}_{k} \Bigg) \triangle q^{n+1,m}_{i,j,k} + \hat{A}^{-}_{i+1} \triangle q^{n+1,m}_{i+1,j,k} \ + \\ \hat{B}^{-}_{j+1} \triangle q^{n+1,m}_{i,j+1,k} \ + \ \hat{C}^{-}_{k+1} \triangle q^{n+1,m}_{i,j,k+1} \ = \ -\Re^{n+1,m}_{U}. \end{aligned} \tag{5.28}$$

The above system produces a large sparse matrix with a bandwidth of seven. Next, iterative solution techniques are used for this system. As discussed before, iterative solution methods are needed to solve the above equation if one wants to eliminate approximate factorization error. The large sparse matrix of equation (5.28) can be written as follows:

$$\Big[L \ + \ D \ + \ U\Big] \ \Big\{\triangle q_{i,j,k}\Big\} \ = \ -\Big\{\Re_{i,j,k}\Big\}. \tag{5.29}$$

In the above, the $\hat{A}^{+}$, $\hat{B}^{+}$, $\hat{C}^{+}$ terms contribute to the lower (L) and diagonal (D) matrices, and the $\hat{A}^{-}$, $\hat{B}^{-}$, $\hat{C}^{-}$ appear on the upper (U) and diagonal matrices.

Symmetric Gauss-Seidel iterations can be used to perform a forward and backward sweep through the computational scheme. In sweeping, the points behind i, j, k are known, but the definition of 'behind' changes, depending on whether the sweep is forward or backward. For clarification purposes, the full algorithm is given here as follows:

Forward sweep

$$\begin{aligned} &L_{i-1,j,k} \ \triangle q^{(2p-1)}_{i-1,j,k} \ + \ L_{i,j-1,k} \ \triangle q^{(2p-1)}_{i,j-1,k} \ + \ L_{i,j,k-1} \ \triangle q^{(2p-1)}_{i,j,k-1} \ + \ D_{i,j,k} \ \triangle q^{(2p-1)}_{i,j,k} \ + \\ &U_{i+1,j,k} \ \triangle q^{(2p-2)}_{i+1,j,k} \ + \ U_{i,j+1,k} \ \triangle q^{(2p-2)}_{i,j+1,k} \ + \ U_{i,j,k+1} \ \triangle q^{(2p-2)}_{i,j,k+1} \ = \ -\Re_{i,j,k}. \end{aligned} \tag{5.30}$$

Rearranging the above equation for solution $\triangle q_{i,j,k}$

$$\begin{aligned} \left\{\triangle q^{(2p-1)}_{i,j,k}\right\} \ = \ &\Big[D_{i,j,k}\Big]^{-1}\Bigg\{ - \Re_{i,j,k} \ - \ L_{i-1,j,k} \ \triangle q^{(2p-1)}_{i-1,j,k} \ - \ L_{i,j-1,k} \ \triangle q^{(2p-1)}_{i,j-1,k} \ - \\ &L_{i,j,k-1} \ \triangle q^{(2p-1)}_{i,j,k-1} \ - \ U_{i+1,j,k} \ \triangle q^{(2p-2)}_{i+1,j,k} \ - \ U_{i,j+1,k} \ \triangle q^{(2p-2)}_{i,j+1,k} \ - \\ &U_{i,j,k+1} \ \triangle q^{(2p-2)}_{i,j,k+1}\Bigg\}. \end{aligned} \tag{5.31}$$

In similar way, the backward sweep becomes

$$\left\{\triangle q_{i,j,k}^{(2p)}\right\} = \left[D_{i,j,k}\right]^{-1}\Big\{ -\Re_{i,j,k} - L_{i-1,j,k}\,\triangle q_{i-1,j,k}^{(2p-1)} - L_{i,j-1,k}\,\triangle q_{i,j-1,k}^{(2p-1)} - L_{i,j,k-1}\,\triangle q_{i,j,k-1}^{(2p-1)} - U_{i+1,j,k}\,\triangle q_{i+1,j,k}^{(2p)} - U_{i,j+1,k}\,\triangle q_{i,j+1,k}^{(2p)} - U_{i,j,k+1}\,\triangle q_{i,j,k+1}^{(2p)}\Big\}. \tag{5.32}$$

Here, $p = 1, 2, 3, \ldots$ is the subiteration counter.

Note that the general solution method is iterative but the solution $\left\{\triangle q_{i,j,k}\right\}$ is attained by direct methods at i, j, k points. More details can be found in [27].

The combination of forward and backward sweeping is called Lower-Upper Block Symmetric Gauss-Seidel (LU/SGS) relaxation. It is a very effective method. Experience shows that three or four full sweeps are sufficient to obtain a converged solution.

LU/SGS are subiterations nested within the Newton method. At convergence of the LU/SGS sweep one has an exact solution for $q^{n+1,m+1}$ in Eq. (5.28). At convergence of the Newton iteration, one has a time-accurate solution for q^{n+1}. In steady-state problem, the time step can be large enough that time accuracy loses meaning, but the steady-state solution can be found with very few Newton steps (in pseudo-time).

CHAPTER VI

COMPUTATIONAL BOUNDARY CONDITIONS

Boundary conditions play a very important role in the solution of practical flow problems. Flowfields governed by the same fluid equations differ from each other because of boundary conditions. Computational boundary conditions must be originated from physical boundary conditions: the correctness of boundary conditions has a large effect on the quality of the overall solution. Ill-posed boundary conditions can create a vicious cycle, where exchange of wrong information between interior domain and boundary can eventually lead to inaccurate results or loss of stability.

In this study, inviscid flows are of concern. Two types of boundary conditions are often used in aerodynamic applications: wall and farfield boundaries. The physical fact for wall boundary conditions is that the normal component of velocity to the surface is zero at the wall. As it is pointed out by Swafford [29], the interior domain requires more information at boundaries than the physics provides in the finite volume formulation. In a wall boundary condition, the physics tells one only about the velocity , but a numerical scheme still requires more information: the *Ghost Cell* approach is useful to overcome these problems. Ghost-cell methods come with the advantage of easy implementation and no alteration of the algorithm near boundaries, but they also have the disadvantage of requiring large amounts of memory usage in multidimensional problems [11]. The ghost cells are duplicated from the first one or two rows of cells within the domain, and this allows for the construction of high-order flux extrapolations, and boundary conditions.

As for farfield boundary conditions, specifying characteristic variables is a natural and appropriate way to implement boundary conditions. Characteristic-variable boundary conditions have been derived under a locally one-dimensional assumption, following the approach of Janus [9]. However, the introduction of the preconditioning matrix changes the characteristics of the hyperbolic equations, and results in new characteristic variables.

6.1 Characteristic Variables

The existence of independent eigenvectors for the flux Jacobians makes it possible to define a new set of dependent variables and decouple the hyperbolic conservation one-dimensional law. This property is the basis for establishing characteristic boundary conditions. The governing equation, Eq. (4.4), can be written in three dimensional space as follows:

$$\frac{\partial q}{\partial \tau} \; + \; a_\Gamma \, \frac{\partial q}{\partial \xi} \; + \; \; b_\Gamma \, \frac{\partial q}{\partial \eta} \; + \; \; c_\Gamma \, \frac{\partial q}{\partial \zeta} \; = \; 0. \tag{6.1}$$

The Jacobians matrices a_Γ , b_Γ , and c_Γ can be diagonalized due to hyperbolicity. A general expression for the Jacobians in all directions can then be written, as $s_\Gamma = R_k \Lambda_k R_k^{-1}$(where $k = \xi$, $k = \eta$, $k = \zeta$ corresponds a_Γ , b_Γ , c_Γ , respectively).
Using the above results, and multiplying equation (6.1) by R_ξ^{-1}, for example, yields:

$$R_\xi^{-1} \, \frac{\partial q}{\partial \tau} \; + \; \Lambda_\xi \; R_\xi^{-1} \, \frac{\partial q}{\partial \xi} \; + \; R_\xi^{-1} \; R_\eta \Lambda_\eta R_\eta^{-1} \, \frac{\partial q}{\partial \eta} \; + \; R_\xi^{-1} \; R_\zeta \Lambda_\zeta R_\zeta^{-1} \, \frac{\partial q}{\partial \zeta} \; = \; 0. \tag{6.2}$$

Assuming *locally frozen eigenvectors* and interpreting the last two terms a source term $S_{\eta,\zeta}$, one gets

$$\frac{\partial W_\xi}{\partial \xi} \; + \; \Lambda_\xi \, \frac{\partial W_\xi}{\partial \xi} \; + \; S_{\eta,\zeta} \; = \; 0, \tag{6.3}$$

where $W_\xi = R_\xi^{-1} q$ is the characteristic variable vector.

In a similar way, Eq. (6.1) can be multiplied by R_η^{-1} and R_ζ^{-1}, assumed locally constant. The above result can be generalized as

$$\frac{\partial W_k}{\partial \tau} \; + \; \Lambda_k \, \frac{\partial W_k}{\partial k} \; + \; S_{m,n} \; = \; 0, \tag{6.4}$$

where k , m , n are ξ, η, ζ, with the constraint $k \neq m \neq n$.

Neglecting the source terms $S_{m,n}$ will facilitate to the use of characteristic variables as a farfield boundary conditions. Introducing a total derivative for W_k, as follows, one has

$$\frac{dW_k}{d\tau} \; = \; \frac{\partial W_k}{\partial \tau} \, \frac{\partial \tau}{\partial \tau} \; + \; \frac{\partial W_k}{\partial k} \, \frac{\partial k}{\partial \tau} \; = \; \frac{\partial W_k}{\partial \tau} \; + \; \Lambda_k \, \frac{\partial W_k}{\partial k} \; = \; 0. \tag{6.5}$$

From the above, it is obvious that the system of equations is decoupled, W_k =const, and that Λ_k defines the characteristic direction $\partial k/\partial\tau = \lambda_k$. Each characteristic variable is associated with one specific eigenvalue, and it is constant along characteristic lines which correspond to the eigenvalues in a hyperbolic system. Characteristic variables can be constructed as

$$W_k \;=\; R_{k,o}^{-1} q. \tag{6.6}$$

Here, $R_{k,o}^{-1}$ is the *locally constant* inverse of the right eigenvector. The subscript "o" denotes frozen [9] value of dependent variables. Those values can be taken from the first cell inside or outside the computational domain, or as arithmetic averages of those cells.

The use of eigenvectors and the solution vector $q \;=\; J\Big\{\rho, u, v, w, P\Big\}^T$ which is detailed in Appendix A and Appendix C, results in a characteristic variable vector, as follows

$$W_k \;=\; J \left\{ \begin{array}{ccc} \rho\hat{k}_x & + & (-w\hat{k}_y + v\hat{k}_z) \;-\; \hat{k}_x\Omega \\ \rho\hat{k}_y & + & (\; w\hat{k}_x - u\hat{k}_z) \;-\; \hat{k}_y\Omega \\ \rho\hat{k}_z & + & (-v\hat{k}_x + u\hat{k}_y) \;-\; \hat{k}_z\Omega \\ \frac{P}{\rho_o\hat{\sigma}_o} & + & \frac{\hat{\Phi}(\hat{\sigma}_o + \hat{\theta}_{k,o}\beta^-)}{2\;\hat{\sigma}_o} \\ \frac{P}{\rho_o\hat{\sigma}_o} & - & \frac{\hat{\Phi}(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}{2\;\hat{\sigma}_o} \end{array} \right\}, \tag{6.7}$$

where

$$\begin{aligned} \Omega &= \frac{P \;+\; 2\rho_o(u\hat{k}_x + v\hat{k}_y + \hat{k}_z)\hat{\theta}_{k,o}\beta^-}{c_o^2\;\beta}, \\ \hat{\Phi} &= u\hat{k}_x + v\hat{k}_y + w\hat{k}_z. \end{aligned}$$

Recall from Appendix C that $\hat{\theta}_{k,o} \;=\; u_o\,\hat{k}_x + v_o\,\hat{k}_y + w_o\,\hat{k}_z + \hat{k}_t$ and $\hat{\sigma}_o \;=\; \Big[(\hat{\theta}_{k,o}\,\beta^-)^2 + \beta\,c_o^2\Big]^{\frac{1}{2}}$.

These characteristic variables are constant along characteristic lines, which correspond to the eigenvalues

$$\lambda_1 = \theta_k; \quad \lambda_2 = \theta_k; \quad \lambda_3 = \theta_k; \quad \lambda_4 = \theta_k\;\beta^+ \;+\; \sigma; \quad \lambda_5 = \theta_k\;\beta^+ \;-\; \sigma. \tag{6.8}$$

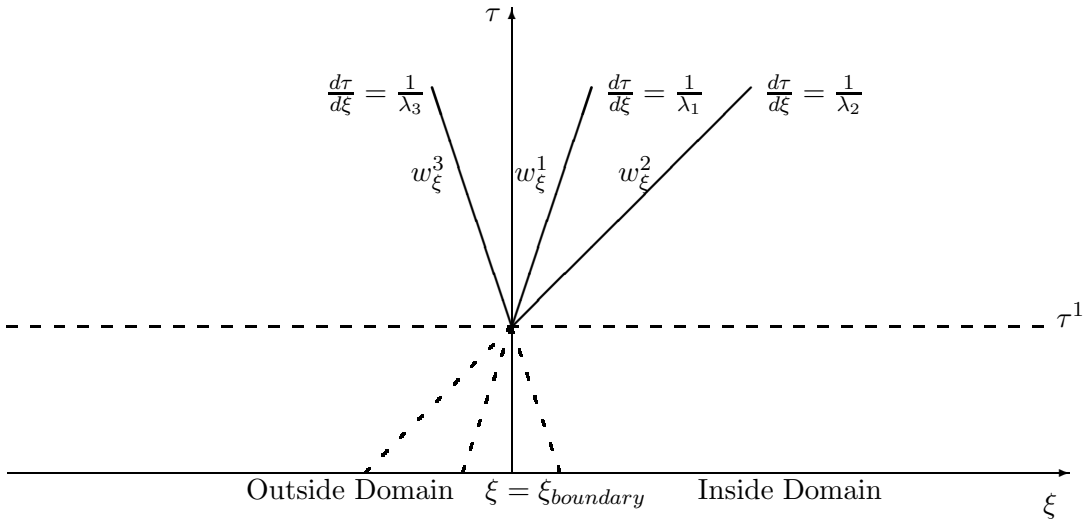

Figure 6.1: Characteristic lines in the $\xi - \tau$ plane

6.2 Characteristic Variable Boundary Conditions (CVBCs)

The CVBCs are going to be developed under a locally one-dimensional assumption (neglecting the source term $S_{m,n}$ in Eq. (6.4), as already mentioned). Characteristic variables W_k are constant along the characteristic lines. A representation of characteristic lines is given for illustration purpose in Fig. (6.1)

In Fig. (6.1), one piece of information comes from the inside domain and two other pieces from the outside domain (this is consistent with subsonic flow). Figure (6.2) shows how information propagates, and the importance of flow-grid orientation, in subsonic flow.

	Codirectional	Contradirectional
$\lambda_{k,1,2,3} > 0;$	$\left(W_k^{1,2,3}\right)_a = \left(W_k^{1,2,3}\right)_b,$	$\lambda_{k,1,2,3} < 0;$ $\left(W_k^{1,2,3}\right)_a = \left(W_k^{1,2,3}\right)_b,$
$\lambda_{k,4} > 0;$	$\left(W_k^{4}\right)_a = \left(W_k^{4}\right)_b,$	$\lambda_{k,4} > 0;$ $\left(W_k^{4}\right)_l = \left(W_k^{4}\right)_b,$
$\lambda_{k,5} < 0;$	$\left(W_k^{5}\right)_l = \left(W_k^{5}\right)_b,$	$\lambda_{k,5} < 0;$ $\left(W_k^{5}\right)_a = \left(W_k^{5}\right)_b.$

(6.9)

Here, 'a' and 'l' refer to the approaching and leaving boundary, respectively. The sign of the eigenvalues depends on flow-grid orientation as well as the flow regime. Codirectional

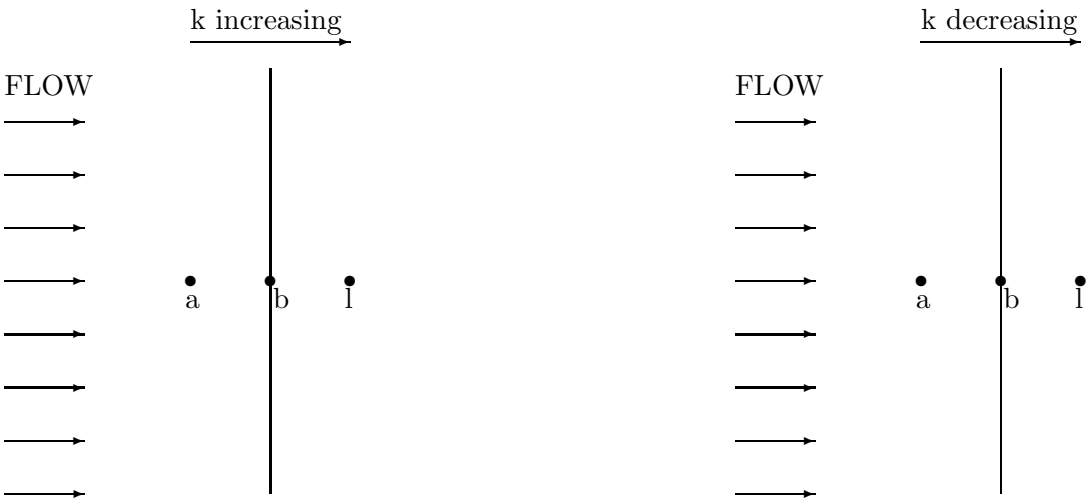

Figure 6.2: Codirectional and Contradirectional Flow

flow (in increasing direction of k) and Contradirectional flow (in decreasing direction of k), are the terms used for these two cases, see Janus [9].

6.3 Subsonic Codirectional Flow

In subsonic flow, four of the eigenvalues have the same sign, and one opposite. So, four pieces of information come from the same direction, in contrast to one that comes from the opposite direction. As mentioned before, codirectional flow is aligned with the direction of increasing k, which means that four of the eigenvalues are positive:

$$\lambda_1 \ , \ \lambda_2 \ , \ \lambda_3 \ , \ \lambda_4 \geq 0 \ \text{ and } \ \lambda_5 \leq 0.$$

Using the characteristic variables in Eq. (6.7), results in the following system of equations:

$$\left[J\left(\rho\hat{k}_x+(-w\hat{k}_y+v\hat{k}_z)-\hat{k}_x\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_x+(-w\hat{k}_y+v\hat{k}_z)-\hat{k}_x\Omega\right)\right]_b, \quad (6.10a)$$

$$\left[J\left(\rho\hat{k}_y+(\ w\hat{k}_x-u\hat{k}_z)-\hat{k}_y\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_y+(\ w\hat{k}_x-u\hat{k}_z)-\hat{k}_y\Omega\right)\right]_b, \quad (6.10b)$$

$$\left[J\left(\rho\hat{k}_z+(-v\hat{k}_x+u\hat{k}_y)-\hat{k}_z\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_z+(-v\hat{k}_x+u\hat{k}_y)-\hat{k}_z\Omega\right)\right]_b, \quad (6.10c)$$

$$\left[J\left(\frac{P}{\rho_o\hat{\sigma}_o}+\frac{\hat{\Phi}(\hat{\sigma}_o+\hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_a = \left[J\left(\frac{P}{\rho_o\hat{\sigma}_o}+\frac{\hat{\Phi}(\hat{\sigma}_o+\hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_b, \quad (6.10d)$$

$$\left[J\left(\frac{P}{\rho_o\hat{\sigma}_o}-\frac{\hat{\Phi}(\hat{\sigma}_o-\hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_l = \left[J\left(\frac{P}{\rho_o\hat{\sigma}_o}-\frac{\hat{\Phi}(\hat{\sigma}_o-\hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_b. \quad (6.10e)$$

In the ghost cell technique, boundary cells use the same metric information as ghost cells, therefore Jacobians cancel out in Eqs. (6.10). Then, the boundary values P_b , ρ_b, u_b , v_b , w_b are obtained as follows:

$$P_b = \frac{1}{2}\left\{P_a+P_l+\rho_o\hat{\sigma}_o\left[\hat{k}_x(u_a-u_l)+\hat{k}_y(v_a-vl)+\hat{k}_z(w_a-w_l)\right]\right\}- \frac{(P_a-P_b)}{2\hat{\sigma}_o}\hat{\theta}_{k,o}\beta^- - \frac{\rho_o[\hat{k}_x(u_a-u_b)+\hat{k}_y(v_a-v_b)+\hat{k}_z(w_a-w_l)]}{2\ \hat{\sigma}_o}(\hat{\theta}_{k,o}\beta^-)^2, \quad (6.11)$$

$$\rho_b = \rho_a+\frac{(P_b-P_a)(\hat{\sigma}_o-\hat{\theta}_{k,o}\beta^-)}{\beta c_o^2(\hat{\sigma}_o+\hat{\theta}_{k,o}\beta^-)}, \quad (6.12)$$

$$u_b = u_a+\hat{k}_x\frac{(P_a-P_b)}{\rho_o(\hat{\sigma}_o+\hat{\theta}_{k,o}\beta^-)}, \quad (6.13)$$

$$v_b = v_a+\hat{k}_y\frac{(P_a-P_b)}{\rho_o(\hat{\sigma}_o+\hat{\theta}_{k,o}\beta^-)}, \quad (6.14)$$

$$w_b = w_a+\hat{k}_z\frac{(P_a-P_b)}{\rho_o(\hat{\sigma}_o+\hat{\theta}_{k,o}\beta^-)}. \quad (6.15)$$

In the above, 'a' and 'b' should be interpreted according to whether the flow is leaving or entering the boundary.

6.4 Subsonic Contradirectional Flow

In contrast to the previous section, contradirectional flow is in the direction of decreasing k. It is easily seen that the first eigenvalue are then negative. Now the signs of eigenvalues is the following:

$$\lambda_1\ ,\ \lambda_2\ ,\ \lambda_3\ ,\ \lambda_5 \leq 0\ \ and\ \ \lambda_4 \geq 0$$

Using the characteristic variables, (in Eq. (6.7)) results in the following system of equations:

$$\left[J\left(\rho\hat{k}_x + (-w\hat{k}_y + v\hat{k}_z) - \hat{k}_x\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_x + (-w\hat{k}_y + v\hat{k}_z) - \hat{k}_x\Omega\right)\right]_b, \quad (6.16a)$$

$$\left[J\left(\rho\hat{k}_y + (\ w\hat{k}_x - u\hat{k}_z) - \hat{k}_y\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_y + (\ w\hat{k}_x - u\hat{k}_z) - \hat{k}_y\Omega\right)\right]_b, \quad (6.16b)$$

$$\left[J\left(\rho\hat{k}_z + (-v\hat{k}_x + u\hat{k}_y) - \hat{k}_z\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_z + (-v\hat{k}_x + u\hat{k}_y) - \hat{k}_z\Omega\right)\right]_b, \quad (6.16c)$$

$$\left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} + \frac{\hat{\Phi}(\hat{\sigma}_o + \hat{\theta}_{k,o}\beta^-)}{2\hat{\sigma}_o}\right)\right]_l = \left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} + \frac{\hat{\Phi}(\hat{\sigma}_o + \hat{\theta}_{k,o}\beta^-)}{2\hat{\sigma}_o}\right)\right]_b, \quad (6.16d)$$

$$\left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} - \frac{\hat{\Phi}(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}{2\hat{\sigma}_o}\right)\right]_a = \left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} - \frac{\hat{\Phi}(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}{2\hat{\sigma}_o}\right)\right]_b. \quad (6.16e)$$

Solving this system yields

$$P_b = \frac{1}{2}\left\{P_a + P_l - \rho_o\hat{\sigma}_o\left[\hat{k}_x(u_a - u_l) + \hat{k}_y(v_a - vl) + \hat{k}_z(w_a - w_l)\right]\right\} + \frac{(P_a - P_b)}{2\hat{\sigma}_o}\hat{\theta}_{k,o}\beta^- + \frac{\rho_o[\hat{k}_x(u_a - u_b) + \hat{k}_y(v_a - v_b) + \hat{k}_z(w_a - w_l)]}{2\hat{\sigma}_o}(\hat{\theta}_{k,o}\beta^-)^2, \quad (6.17)$$

$$\rho_b = \rho_a + \frac{(P_b - P_a)(\hat{\sigma}_o + \hat{\theta}_{k,o}\beta^-)}{\beta c_o^2(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}, \quad (6.18)$$

$$u_b = u_a - \hat{k}_x\frac{(P_a - P_b)}{\rho_o(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}, \quad (6.19)$$

$$v_b = v_a - \hat{k}_y\frac{(P_a - P_b)}{\rho_o(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}, \quad (6.20)$$

$$w_b = w_a - \hat{k}_z\frac{(P_a - P_b)}{\rho_o(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}. \quad (6.21)$$

There is a lot of similarity between the contradirectional and codirectional results. These results can be combined under one formulation by using the sign value of the first eigenvalue, as follows

$$
\begin{aligned}
P_b &= \frac{1}{2}\left\{ P_a + P_l + sgn(\lambda_k^1)\rho_o\hat{\sigma}_o\Big[\hat{k}_x(u_a-u_l)+\hat{k}_y(v_a-v_l)+\hat{k}_z(w_a-w_l)\Big]\right\} + \\
&\quad -sgn(\lambda_k^1)\frac{(P_a-P_l)}{2\ \hat{\sigma}_o}\hat{\theta}_{k,o}\beta^- + \\
&\quad -sgn(\lambda_k^1)\frac{\rho_o[\hat{k}_x(u_a-u_b)+\hat{k}_y(v_a-v_b)+\hat{k}_z(w_a-w_l)]}{2\ \hat{\sigma}_o}(\hat{\theta}_{k,o}\beta^-)^2, \qquad (6.22)\\
\rho_b &= \rho_a + \frac{(P_b-P_a)(\hat{\sigma}_o - sgn(\lambda_k^1)\hat{\theta}_{k,o}\beta^-)}{\beta c_o^2(\hat{\sigma}_o + sgn(\lambda_k^1)\hat{\theta}_{k,o}\beta^-)}, \qquad (6.23)\\
u_b &= u_a + sgn(\lambda_k^1)\hat{k}_x\frac{(P_a-P_b)}{\rho_o(\hat{\sigma}_o + sgn(\lambda_k^1)\hat{\theta}_{k,o}\beta^-)}, \qquad (6.24)\\
v_b &= v_a + sgn(\lambda_k^1)\hat{k}_y\frac{(P_a-P_b)}{\rho_o(\hat{\sigma}_o + sgn(\lambda_k^1)\hat{\theta}_{k,o}\beta^-)}, \qquad (6.25)\\
w_b &= w_a + sgn(\lambda_k^1)\hat{k}_z\frac{(P_a-P_b)}{\rho_o(\hat{\sigma}_o + sgn(\lambda_k^1)\hat{\theta}_{k,o}\beta^-)}. \qquad (6.26)
\end{aligned}
$$

The relations are generalized for subsonic flow (either contradirectional or codirectional). In the above, 'a' and 'l' is a kind of arbitrary position, depending on whether flow is entering or leaving the domain. In practical applications, the determination of what is 'a' and 'l' is automated for robustness.

6.5 Supersonic Flow

In supersonic flow, all eigenvalues have the same sign, so there is also no need to distinguish between codirectional and contradirectional flow. The characteristic equations

become:

$$\left[J\left(\rho\hat{k}_x + (-w\hat{k}_y + v\hat{k}_z) - \hat{k}_x\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_x + (-w\hat{k}_y + v\hat{k}_z) - \hat{k}_x\Omega\right)\right]_b, \quad (6.27a)$$

$$\left[J\left(\rho\hat{k}_y + (\ w\hat{k}_x - u\hat{k}_z) - \hat{k}_y\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_y + (\ w\hat{k}_x - u\hat{k}_z) - \hat{k}_y\Omega\right)\right]_b, \quad (6.27b)$$

$$\left[J\left(\rho\hat{k}_z + (-v\hat{k}_x + u\hat{k}_y) - \hat{k}_z\Omega\right)\right]_a = \left[J\left(\rho\hat{k}_z + (-v\hat{k}_x + u\hat{k}_y) - \hat{k}_z\Omega\right)\right]_b, \quad (6.27c)$$

$$\left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} + \frac{\hat{\Phi}(\hat{\sigma}_o + \hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_a = \left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} + \frac{\hat{\Phi}(\hat{\sigma}_o + \hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_b, \quad (6.27d)$$

$$\left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} - \frac{\hat{\Phi}(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_a = \left[J\left(\frac{P}{\rho_o\hat{\sigma}_o} - \frac{\hat{\Phi}(\hat{\sigma}_o - \hat{\theta}_{k,o}\beta^-)}{2\ \hat{\sigma}_o}\right)\right]_b. \quad (6.27e)$$

The solution of the above system is as follows

$$p_b = p_a, \qquad \rho_b = \rho_a, \qquad u_b = u_a, \qquad v_b = v_a, \qquad w_b = w_a.$$

These results should have been expected, due to the physics of supersonic flows.

6.6 Wall Boundary Conditions

According to inviscid flow theory, the flow is tangent to a solid surface. As already mentioned, a finite volume formulation requires more information than what physics can provide. Typically, zero-pressure gradient and constant enthalpy are used for obtaining pressure and density. Adjustments for improved inviscid boundary conditions have also proposed [30], which account for wall curvature effects, and result in slightly more complicated boundary conditions.

In the computational domain, contravariant velocity components can be written as

$$\begin{aligned} U &= \theta_\xi = \xi_t + u\xi_x + v\xi_y + w\xi_z, \\ V &= \theta_\eta = \eta_t + u\eta_x + v\eta_y + w\eta_z, \\ W &= \theta_\zeta = \zeta_t + u\zeta_x + v\zeta_y + w\zeta_z. \end{aligned} \quad (6.28)$$

In the above, the values U, V , W are taken from the interior of the domain, and are known. Normal velocity is zero at the wall. If applied to a surface η, this conditions

becomes:

$$\begin{aligned} \xi_t &+ u\xi_x + v\xi_y + w\xi_z = U, \\ \eta_t &+ u\eta_x + v\eta_y + w\eta_z = 0, \\ \zeta_t &+ u\zeta_x + v\zeta_y + w\zeta_z = W. \end{aligned} \tag{6.29}$$

The solution of equations (6.29) yields

$$u = J\Big\{\zeta_z\left(\eta_y\left(U-\xi_t\right)+\eta_t\xi_y\right)-\zeta_y\left(U\eta_z-\eta_z\xi_t+\eta_t\xi_z\right)+\left(W-\zeta_t\right)\left(\eta_z\xi_y-\eta_y\xi_z\right)\Big\}, \tag{6.30}$$

$$v = J\Big\{\zeta_z\left(\eta_x\left(U-\xi_t\right)+\eta_t\xi_x\right)-\zeta_x\left(U\eta_z-\eta_z\xi_t+\eta_t\xi_z\right)+\left(W-\zeta_t\right)\left(\eta_z\xi_x-\eta_x\xi_z\right)\Big\}, \tag{6.31}$$

$$w = J\Big\{\zeta_y\left(\eta_x\left(U-\xi_t\right)+\eta_t\xi_x\right)-\zeta_x\left(\eta_y\left(U-\xi_t\right)+\eta_t\xi_y\right)+\left(W-\zeta_t\right)\left(\eta_y\xi_x-\eta_x\xi_y\right)\Big\}, \tag{6.32}$$

where $J=\frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}=\left[x_\xi\left(y_\eta z_\zeta-z_\eta y_\zeta\right)+y_\xi\left(z_\eta x_\zeta-x_\eta z_\zeta\right)+z_\xi\left(x_\eta y_\zeta-y_\eta x_\zeta\right)\right]$ is the Jacobian of the inverse coordinate transformation.

Imposing a zero pressure gradient by setting $\partial p/\partial n = 0$, is equivalent to the statement that the pressure is constant in the normal direction to surface. Then, the computational boundary value for pressure is taken from the first cell inside the domain. For a solid η surface, this becomes:

$$P_b = P_{i,1,k}. \tag{6.33}$$

In order to find density, it can be assumed that the freestream total enthalpy is constant along the body surface

$$h_{t_\infty} = h_{t_{body}} = e_t + E_c\frac{p}{\rho}. \tag{6.34}$$

Computational boundary density can then be found as

$$\rho_b = \frac{\gamma E_c P_b}{(\gamma-1)(h_{t_\infty}-E_c\phi_b)}. \tag{6.35}$$

Note that ϕ_b and P_b have been found from the previous results.

CHAPTER VII

RESULTS

In this study, a computer code was implemented for solving the two-dimensional Euler equations. The code uses a finite volume formulation with the Roe flux difference splitting method, and can be applied to structured grids. Both explicit and implicit time integrations are employed. The Newton method is used for the linearization in the implicit scheme, with a symmetric Gauss-Seidel (LU/SGS) algorithm used for the solution of the linear equations. CVBCs and tangency conditions are implemented in the code for far-field and wall boundary, respectively.

A number of applications are performed to test the robustness of the preconditioning method, and its implementation. The first set of test cases is considered necessary to analyze preconditioning schemes. In order to reach a reliable conclusion, these test cases are designed to minimize boundary condition effects and to exclude the complexity of an implicit solver. These preliminary test cases are: (1) point disturbance, (2) flow angle sensitivity, and (3) stagnation point. The first one is considered a somewhat realistic case, because a wide range of linear and nonlinear waves is generated by the disturbance. The latter two test cases are closely related to possible instability at a stagnation point [5].

The next two sets of test cases involve steady-state and unsteady flows, respectively. The steady-state cases include both external and internal flows: an airfoil and a convergent-divergent nozzle, respectively. The symmetric airfoil NACA 0012 was chosen for this study. In particular, low Mach number flows are investigated in terms of accuracy and convergence rates. Moreover, the same airfoil is investigated at transonic speed to compare convergence rates for preconditioned and original schemes. Lastly, a supersonic case is also examined in order to test all range of Mach numbers. The other steady-state case is chosen to be an internal flow in a convergent-divergent (C-D) nozzle. This test case attempts to asses robustness and reliability in the code for internal flow application with a wide range of Mach

numbers. As for unsteady test cases, a shock tube problem is investigated for different reference Mach numbers.

7.1 Test Cases for Preconditioning Schemes

Several test cases involving both the original and preconditioned schemes are used to test reliability and capability of the new preconditioning method. The first test case is used to measure the ability to damp a pressure disturbance introduced into the domain. The second one is to observe the sensitivity of the preconditioner to a given flow angle disturbance. In the last test case, the preconditioner is studied at a stagnation point. This case is thought to be mandatory before a preconditioning scheme can be applied to any realistic configuration; however, a few local preconditioners will yield poor results here, or fail to converge altogether[2].

7.1.1 Point Disturbance Test

The first test case is applied to both the original and the preconditioned schemes. A given pressure disturbance, which is taken as $0.1M_\infty^2 p_\infty$, is introduced at the center of a 10×10 uniform grid. It will generate both linear and non-linear waves, thus representing a somewhat realistic case.

Boundary conditions are set as a uniform flow, which is not allowed to change. This will minimize boundary effects on the solution, and allow to examine convergence behavior safely. Furthermore, an *explicit* time-integration scheme is used, which will facilitate the analysis and be consistent with Zaccanti's results [2].

The explicit scheme is run for 3000 time steps with a CFL=0.3. In Fig.(7.1), the convergence history is given for the original scheme. As expected, the convergence of the original scheme tends to get slower at low Mach numbers. Two reasons have been identified for this behavior: (1) a large condition number (the ratio of the largest and smallest eigenvalues), and (2) and improperly scaled compressible equations, which can not represent well virtually incompressible behavior at low speeds [4]. The convergence behavior is similar in the case of high Mach numbers , and it deteriorates when the Mach number approaches zero.

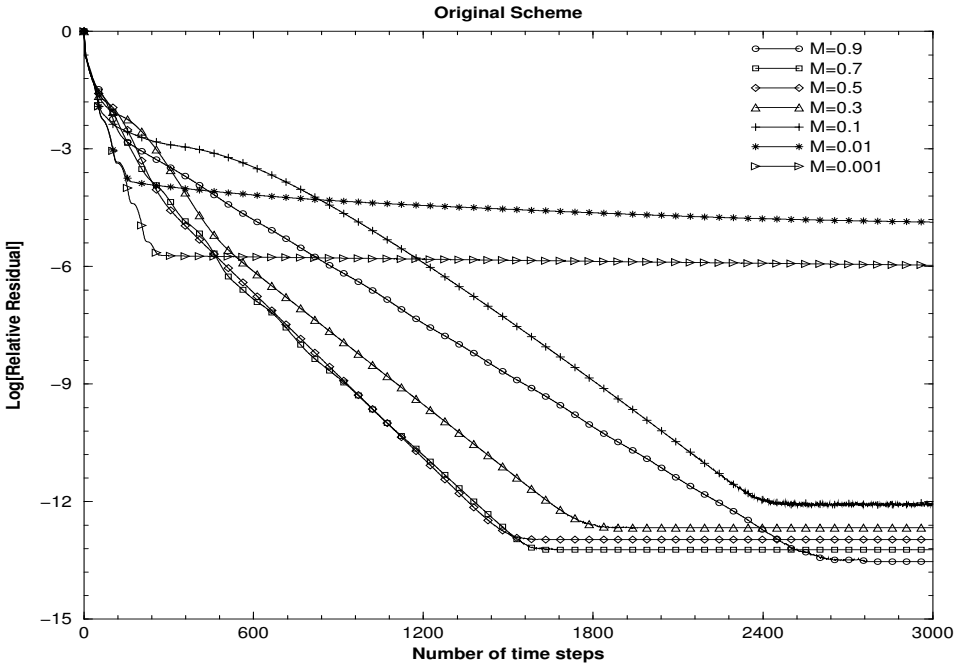

Figure 7.1: Original Scheme Convergence History for a given Pressure disturbance

The same case is run for the preconditioned scheme under the same conditions. The convergence history is presented in Fig.(7.2). One of the purposes of preconditioning is to make convergence Mach-number independent at low speed. The preconditioned scheme successfully achieves this objective, as seen in Fig.(7.2), although round-off errors appear to preclude convergence to machine zero for very low Mach numbers. The test conditions are taken from those of Zaccanti [2], and these results show that a new global preconditioning is not converging as fast as others. The number of iterations required to reach the same magnitude of residual reduction is approximately three times higher than most of the available preconditioned schemes. Note that the choice of local time stepping may vary slightly different implementations.

7.1.2 Flow Angle Test

The second test case may also be considered as a disturbance test: the velocity vector of the center cell is deviated from the uniform field. Deflected flows are usually encountered near the stagnation point in a realistic case, and are considered a possible reason for

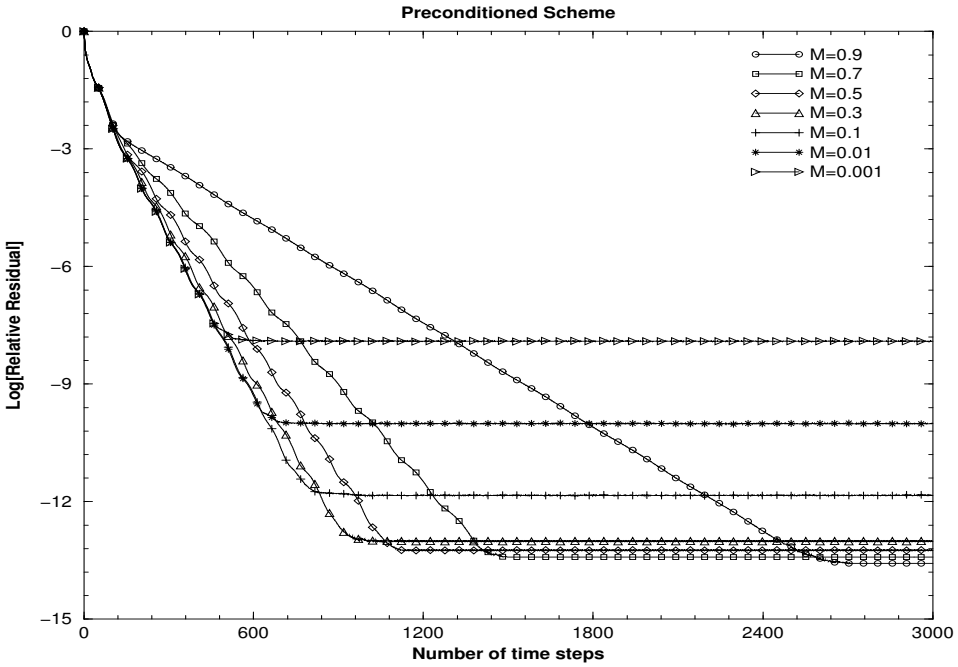

Figure 7.2: Preconditioned Scheme Convergence History for given Pressure disturbance

instability [5]. In traditional analysis, a preconditioning matrix may be written for any flow angle. In general, the preconditioning matrices are a function of flow angle and Mach number. As the Mach number approaches to zero, the preconditioning matrix can become sensitive to the flow angle. In order to remove this ill-conditioned situation, the flow dependence should be removed completely or alleviated to some extent [5]. Fortunately, the new global preconditioning is completely independent of flow angle, so the scheme should be able to recover from any disturbance all the way to the maximum deflection angle of 180^o. The maximum tolerated flow angle in the center cell is tested for various CFL numbers with a 10×10 square grid, at a fixed $M = 0.1$. Again, boundary conditions are consistent with a uniform flow and are not allowed to be changed. Convergence histories are given in Figs. (7.3) and (7.4) for both schemes for completeness. However, the traditional success criterion here is the value of the maximum perturbed angle tolerated by the scheme, regardless of convergence speed. The maximum perturbed angle is given as 180^o for the new preconditioned scheme, which successfully passes the convergence test for all CFL numbers.

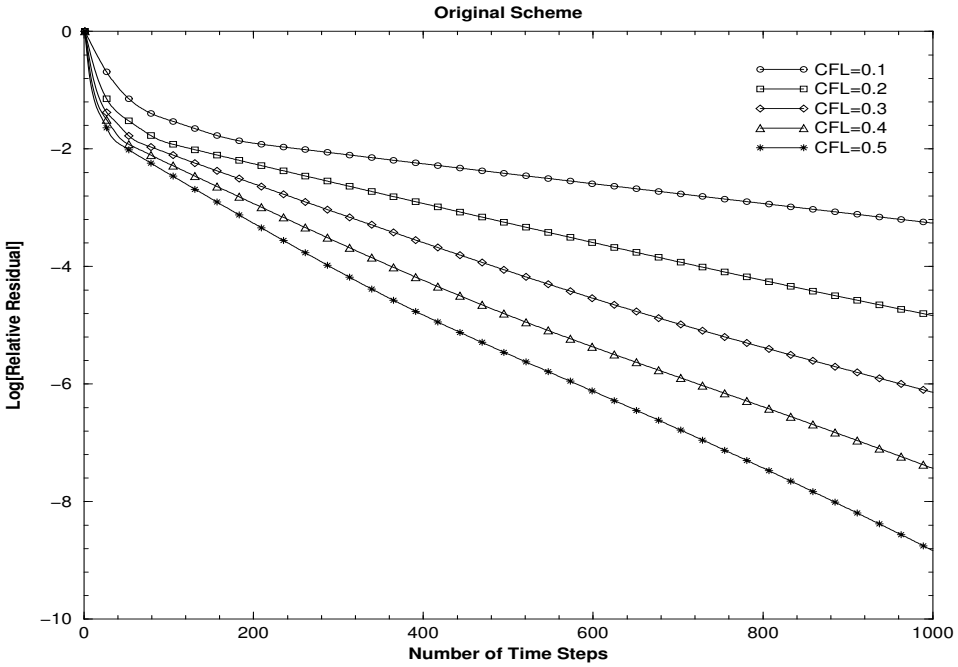

Figure 7.3: Original Scheme Convergence History for flow angle test case

According to Zaccanti's results [2], the Chorin/Turkel preconditioner is the *only one* to be as robust for the same conditions.

7.1.3 Stagnation Point Test

This last preliminary test case attempts to measure the ability of a preconditioning scheme to compute a stagnation point. Two identical jets moving in opposite directions are set to impinge in the middle of the computational domain, creating a stagnation point. The boundary conditions are set to the initial conditions, and are not allowed to change. The Mach number is set to 0.1 in a 22×23 uniform grid. The CFL number used for the solution is set to $CFL = 0.4$. The flow meets at *a cell face* in the middle of the domain. This makes the Roe-averaged velocity zero, and the flow deflection angle large at the face where the stagnation point resides, which is considered the worst-case scenario for a preconditioner. Some local preconditioners will not be able to deal with the aforementioned problems without some modifications of the scheme [25][5]. A detailed view of the simulation

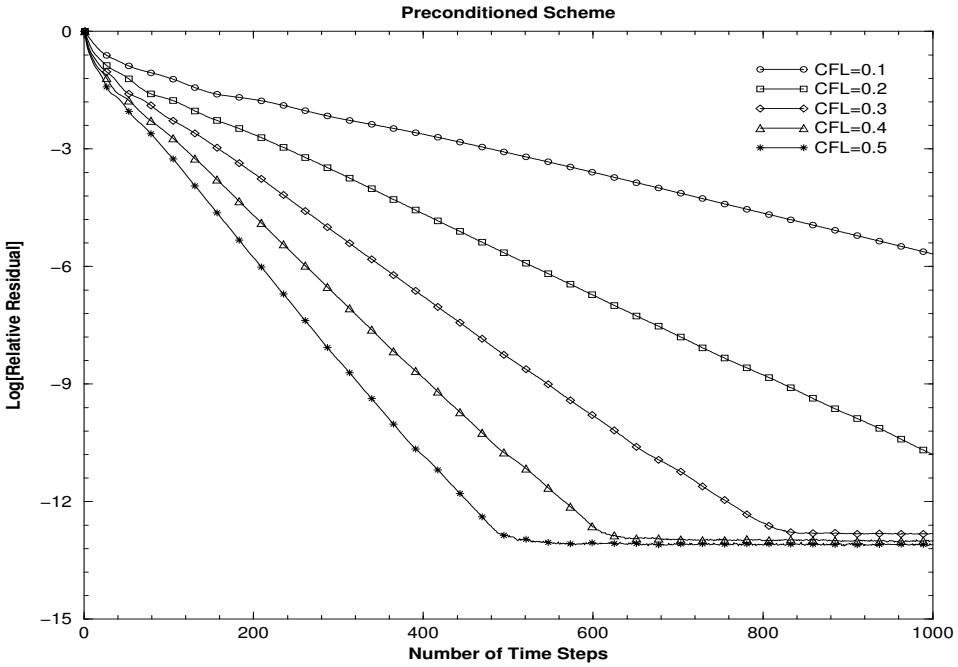

Figure 7.4: Preconditioned Scheme Convergence History for flow angle test case

results is given in Fig. (7.5). The new scheme captures the stagnation point perfectly, as does the original unpreconditioned one.

Convergence histories are given in Figs.(7.6) for both schemes, and agree with previous results [2]. In addition to giving accurate results, convergence is better at low Mach numbers when using the preconditioner.

7.2 Steady-State, External Flows: Airfoil Tests

In this section, comparisons of predictions from both original and preconditioned schemes for a NACA 0012 airfoil are presented, for a large range of Mach numbers. The preconditioned scheme is designed to make compressible algorithms usable at low Mach numbers, so that a single algorithm for all flow regimes can be employed. Consequently, subsonic, transonic, and supersonic cases are presented and compared for original and preconditioned schemes.

In subsonic cases, the NACA 0012 airfoil cases are run on a coarse grid (177×33 cells, with 112 cells on the airfoil) shown in Fig.(7.7), with a CFL number of $CFL = 100$. The

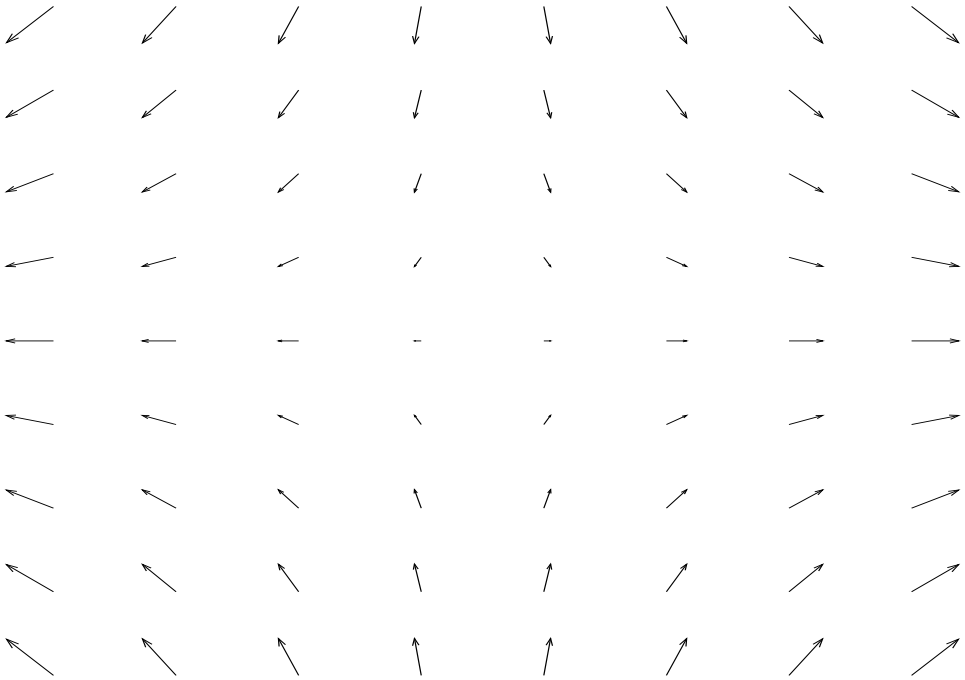

Figure 7.5: Stagnation point—detailed view

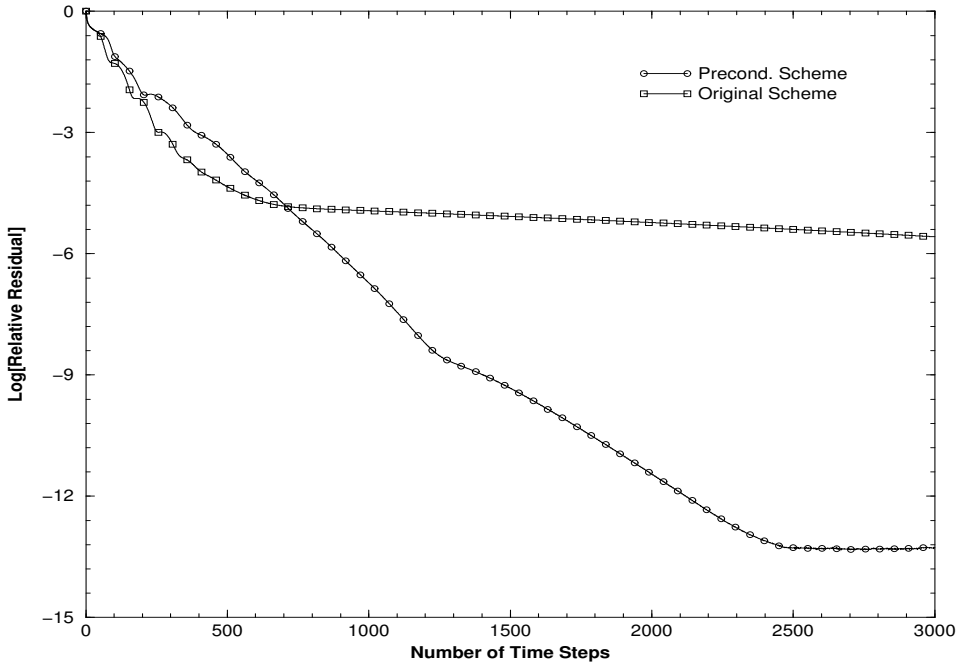

Figure 7.6: Stagnation point Convergence History for Original and Preconditioned Scheme

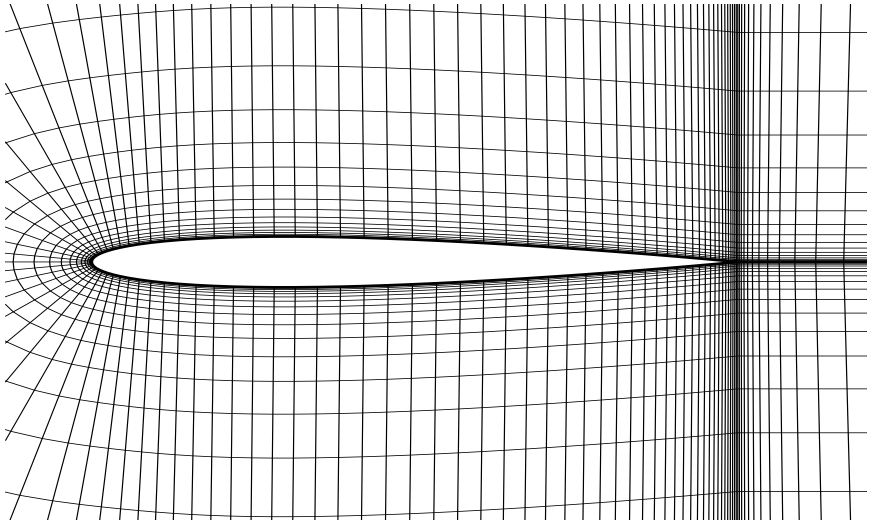

Figure 7.7: Coarser 177×33 C-grid for NACA0012 airfoil

implicit scheme is used, with 3 Newton iteration loops per time step and 4 LU/SGS iterative sweeps, on average. The numerical flux Jacobian matrix is computed twice per time step, which is considered sufficient even when using higher numbers of Newton iterations. Note that this test case is steady, so time accuracy is not required. The boundary conditions used are those already discussed: CVBCs at the farfield (located at a distance of 40 chord lengths away from the airfoil), and tangency at the wall, for all flow regimes.

In transonic and supersonic cases, a finer grid (353×65 cells, with 224 cells on the airfoil) given in Fig.(7.8) is used to achieve better accuracy of the solution. The use of high-order space discretization and the formation of strong shocks, may cause instability at the very beginning of the calculations. This can be fixed by using good initial values, or employing a lower CFL number and first-order space discretization at the early steps. Here, it is preferred to use a lower CFL number, while keeping the order of scheme the same, at the beginning of computation.

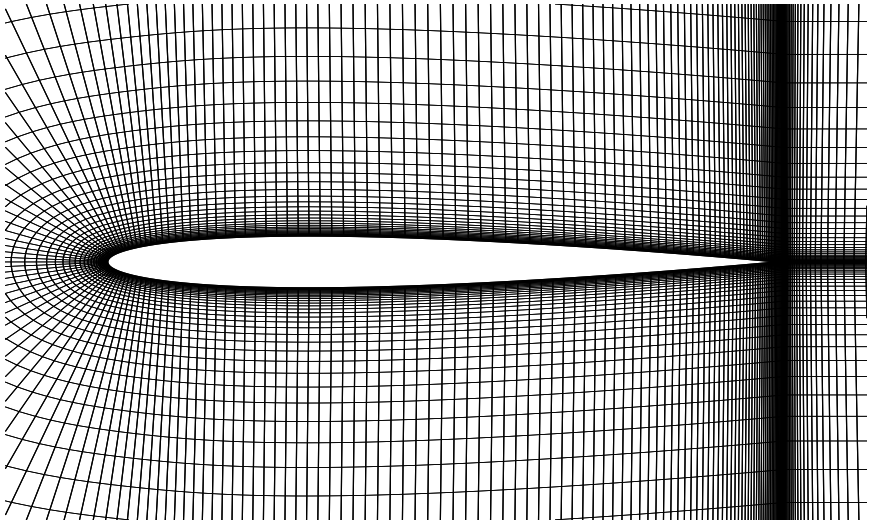

Figure 7.8: Finer 353×65 C-grid for NACA0012 airfoil

7.2.1 Subsonic Flow

For subsonic flow, test cases are run for various Mach numbers, ranging from 0.001 to 0.5, with a 2^o angle of attack. The original scheme experiences convergence difficulties, but eventually it does converge. However, the unpreconditioned compressible algorithm yields inaccurate results at very low Mach numbers (note that none of the previous test cases was designed to expose this problem). It is well known from classical thin airfoil theory for low-speed flow that the coefficient of lift is proportional to angle of attack, according to the Prandtl-Glauert relation $C_l = 2\pi\alpha/\sqrt{1 - M_\infty^2}$. This is a simple result that can be used to compare the predictions of the numerical schemes.

In Figs.(7.9) and (7.10), plots of coefficient of lift versus number of iterations are shown for variable Mach numbers for the first-order schemes. The prediction from the original scheme in Figs.(7.9) differ greatly from the theoretical results at $M = 0.01$. In practice, it is seen that the solution starts deteriorating at $M = 0.1$. The residual convergence plot of Fig.(7.11) shows a deterioration in the number of cycles it takes to obtain the (wrong) solution at low Mach numbers.

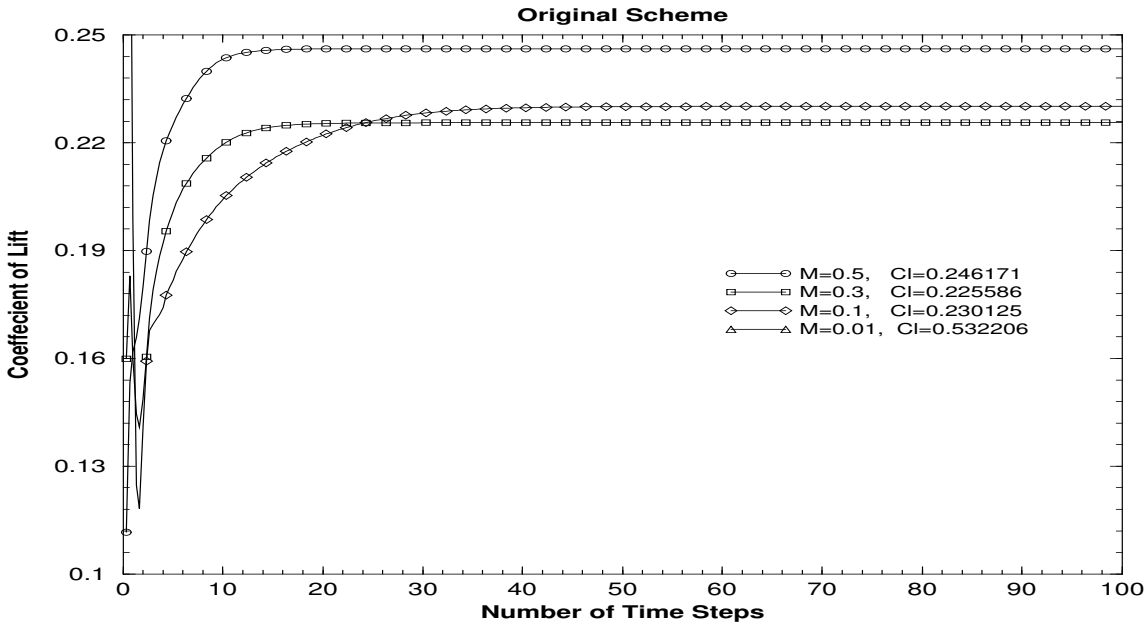

Figure 7.9: Coefficient of Lift for various Mach Numbers from 1^{st} Order Original Scheme at Subsonic Flow

The use of the preconditioning scheme greatly improves the solution, especially at very low Mach numbers. In Fig.(7.10), lift coefficients reach accurate values, consistent with those predicted by the theory. The convergence is independent of Mach number, as seen in Fig.(7.12), consistently with what was already seen in the point disturbance test case for low Mach numbers.

The same cases are run for second and third-order schemes, with the minmod limiter. As it is shown in Figs. (7.13), the coefficient of lift is more accurate than that in Fig. (7.9) at $M = 0.1$ for the original scheme. This is in line with the expectation that high order schemes are more accurate. As seen in Figs.(7.14), the preconditioned scheme also improves the solution at low Mach number. In contrast to Fig. (7.15), the second-order preconditioned scheme convergence given in Figs. (7.16) is seen to be independent of Mach number. As for third-order results, shown in Fig. (7.17), accuracy is better than the first-order scheme at $M = 0.1$. Very accurate results are recovered when using preconditioning schemes at low Mach numbers, as seen in Fig. (7.18). Convergence histories given in Figs. (7.19) and (7.20) follows the the same pattern as those of the second-order scheme. The

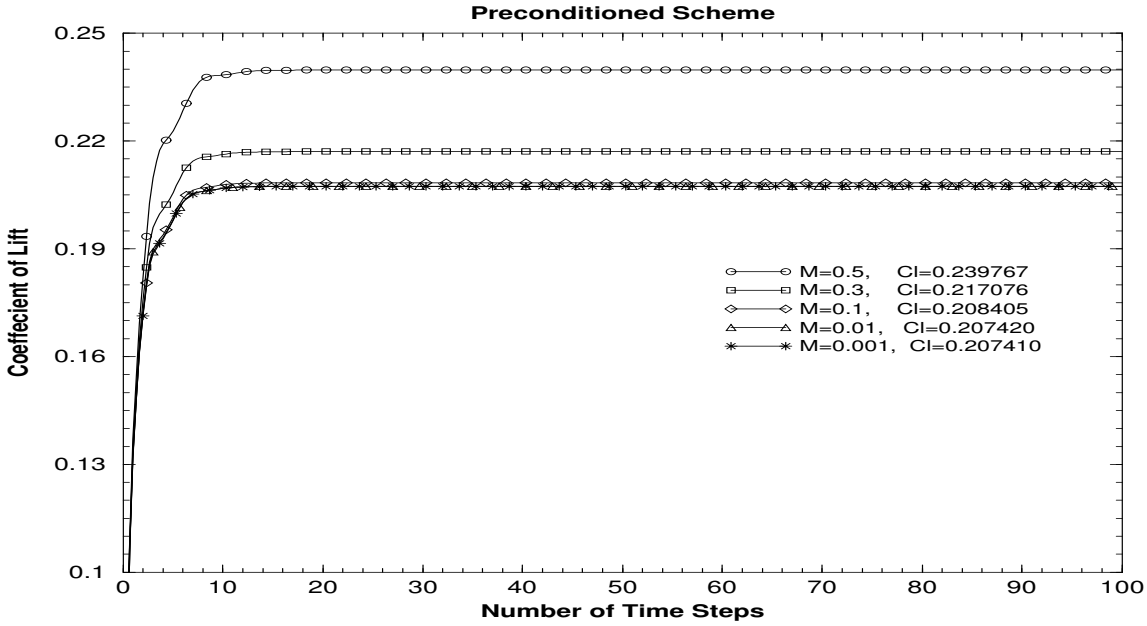

Figure 7.10: Coefficient of Lift for various Mach Numbers from 1^{st} Order Preconditioned Scheme at Subsonic Flow

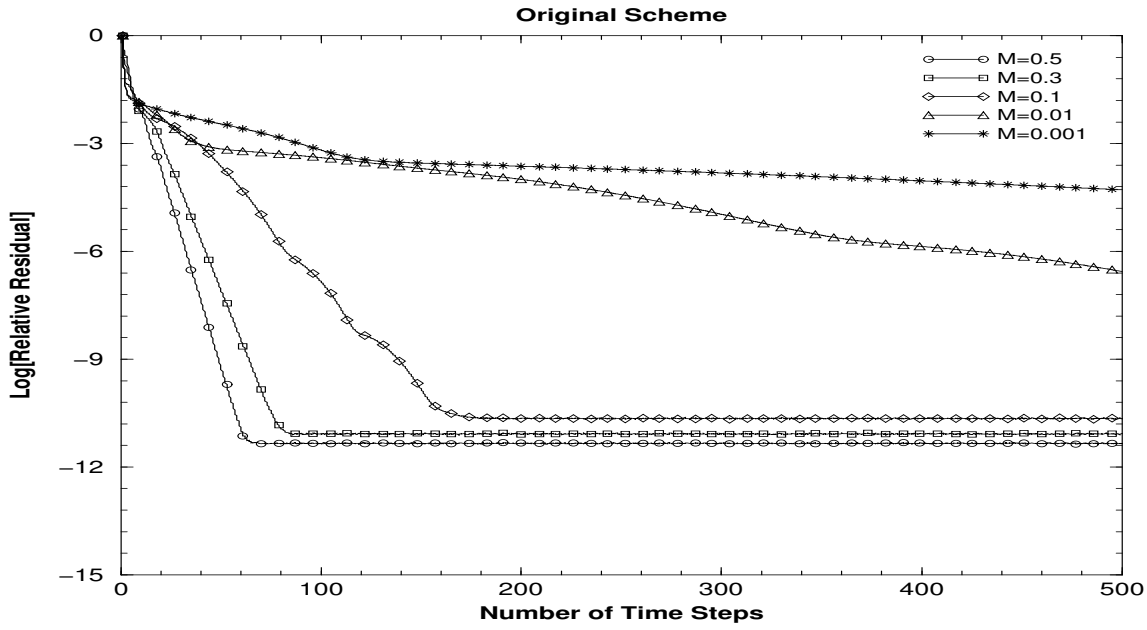

Figure 7.11: First-Order Original Scheme Convergence History for various Mach numbers at Subsonic Flow

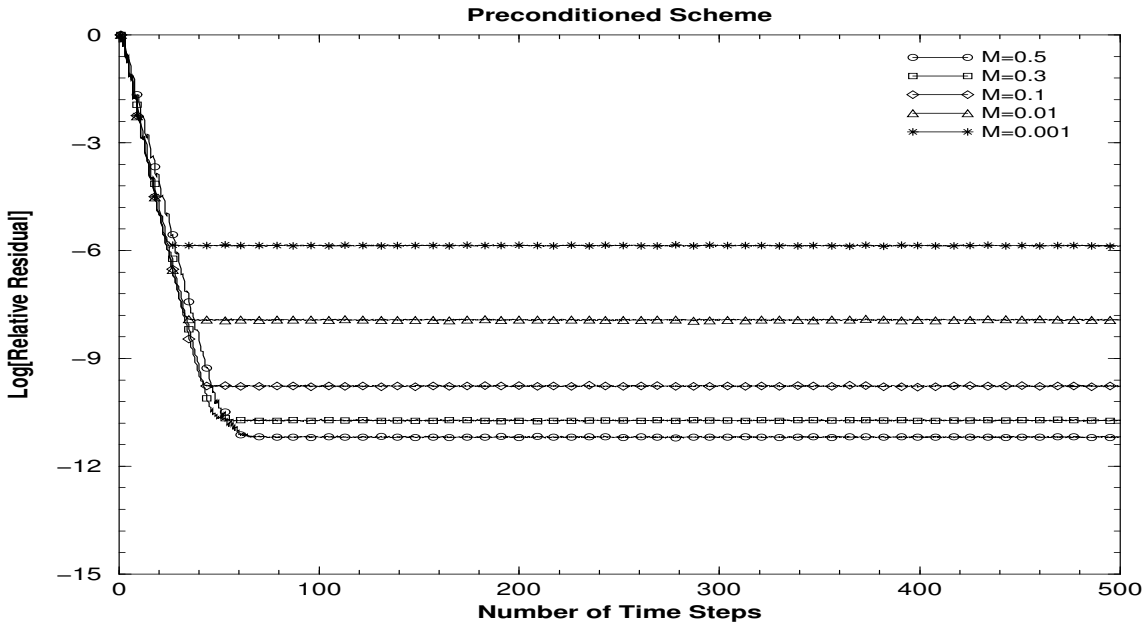

Figure 7.12: First-Order Preconditioned Scheme Convergence History for various Mach numbers at Subsonic Flow

high-order convergence histories are not able to go down to machine zero. Remarkably, the coefficient of lifts are leveled to the correct values after only few iterations. Traditionally, limiters such as minmod are blamed for this kind of behavior.

7.2.2 Transonic Flow

The next test case involves the same NACA 0012, at transonic speed. A finer grid (353×65 cells), is used for the numerical solution at $M_\infty = 0.85$ and $\alpha = 1^o$, due to presence of a strong shock on the upper surface and a relatively weaker shock on the lower surface. As already mentioned, high-order space discretization and the presence of shocks require the use of lower CFL numbers in the initial phase. Eventually, first and high-order schemes run with CFL=100 and CFL=30, respectively, after using a lower CFL for a few iterations at the very beginning.

The present results are compared to those from the AGARD test cases [31]. In particular, shock location, coefficient of lift and drag, and Mach contour lines will be compared. It is easily seen that large gradient occurs at the shock location (see Figs. (7.21) and (7.22), and

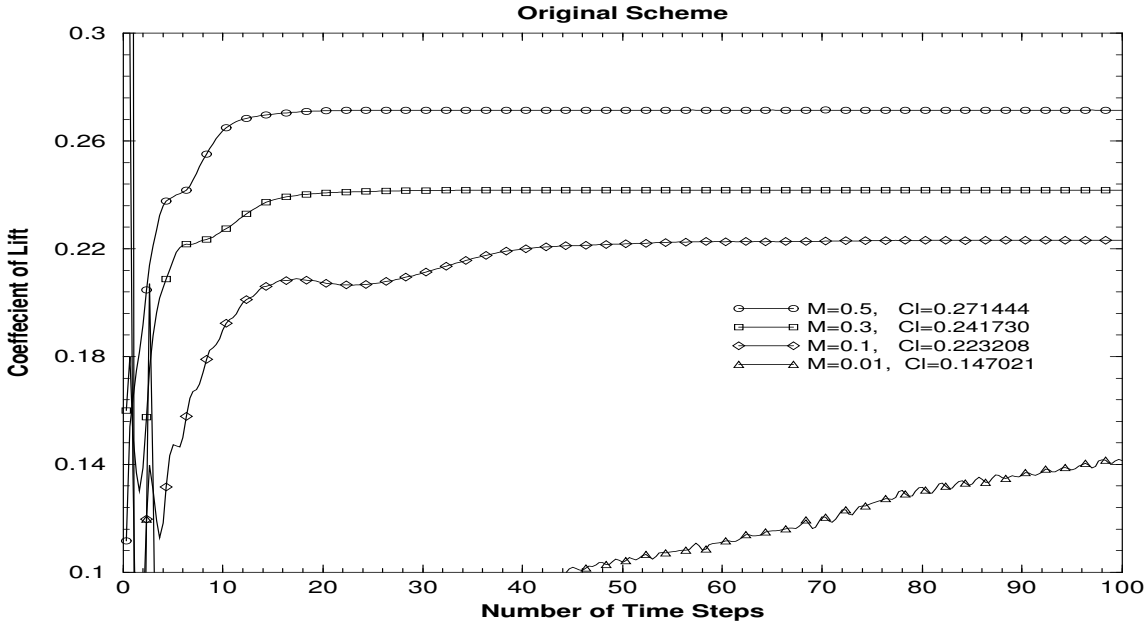

Figure 7.13: Coefficient of Lift for various Mach Numbers from 2^{nd} Order Original Scheme at Subsonic Flow

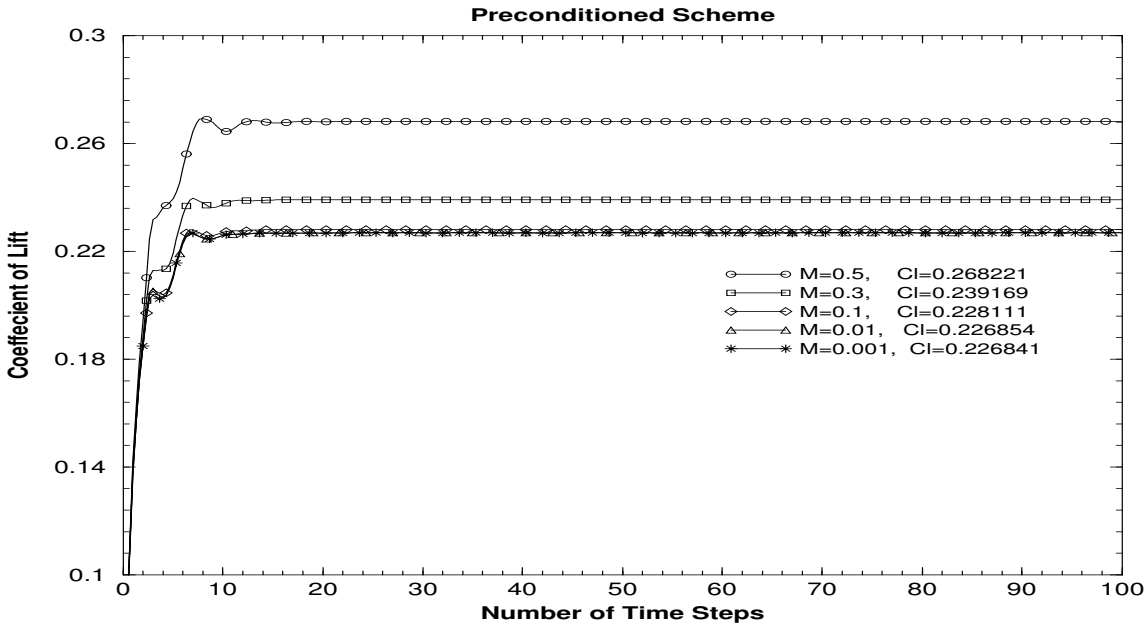

Figure 7.14: Coefficient of Lift for various Mach Numbers from 2^{nd} Order Preconditioned Scheme at Subsonic Flow

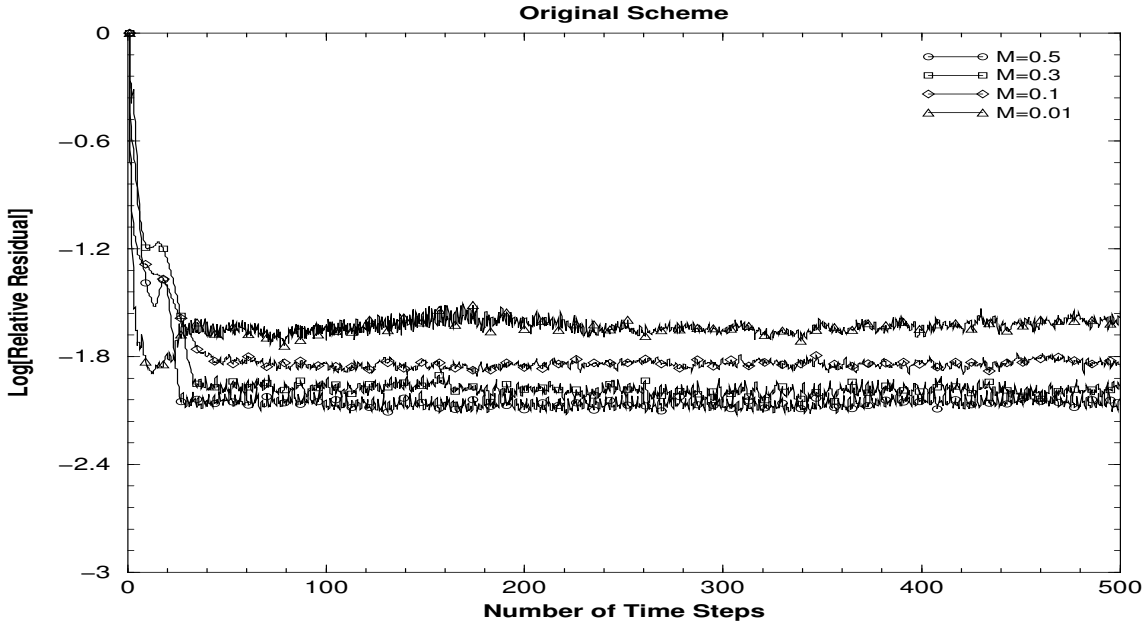

Figure 7.15: Second-Order Original Scheme Convergence History for various Mach numbers at Subsonic Flow

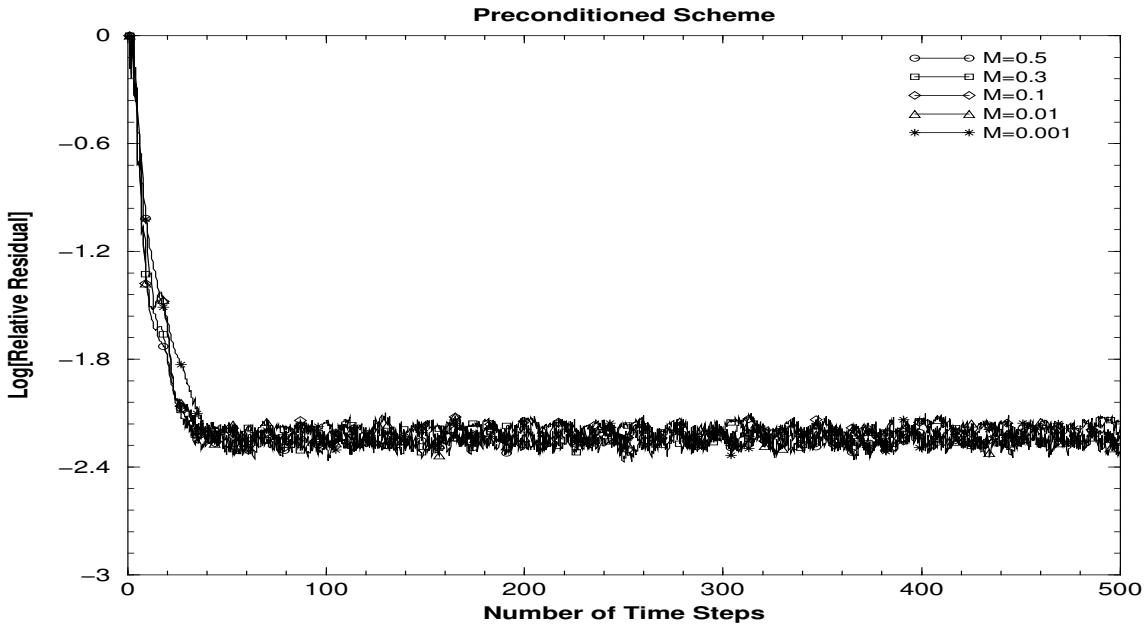

Figure 7.16: Second-Order Preconditioned Scheme Convergence History for various Mach numbers at Subsonic Flow

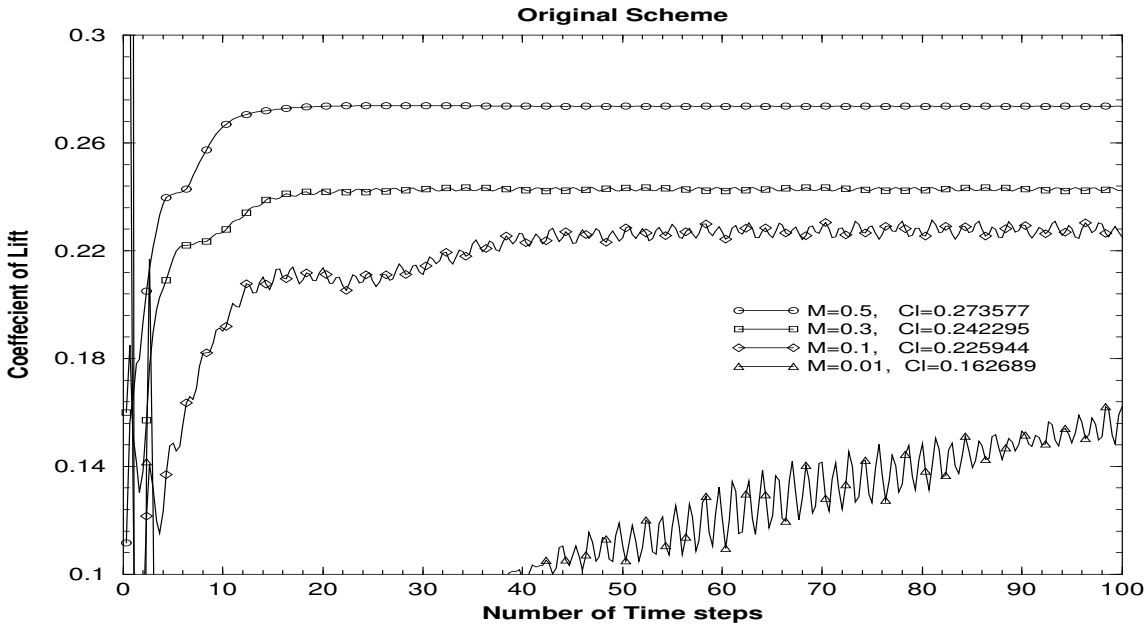

Figure 7.17: Coefficient of Lift for various Mach Numbers from 3^{rd} Order Original Scheme at Subsonic Flow

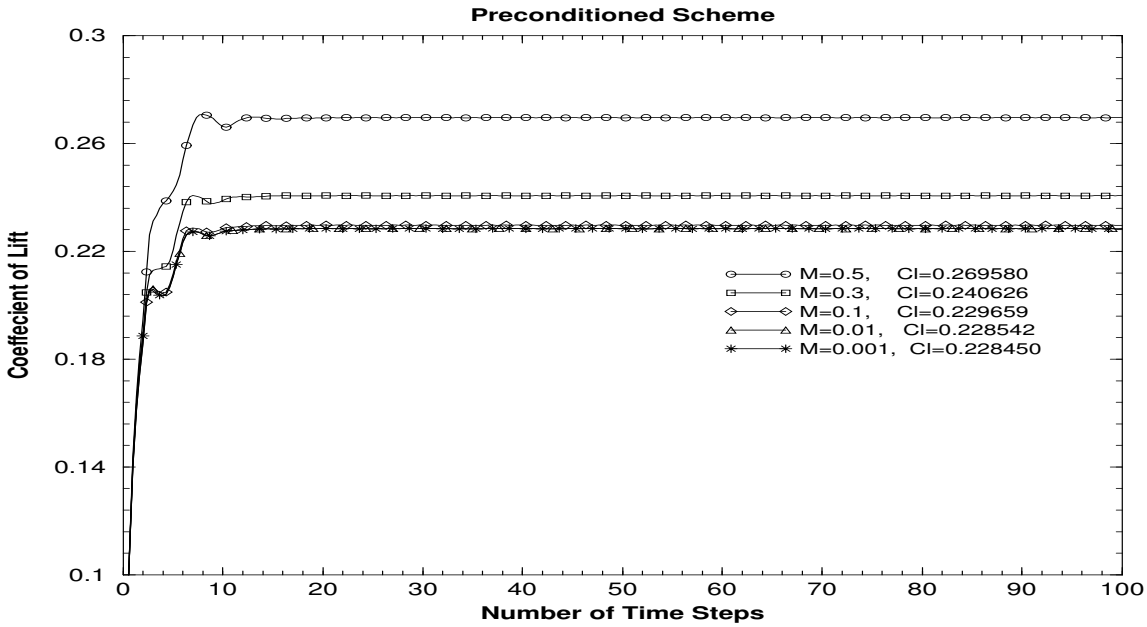

Figure 7.18: Coefficient of Lift for various Mach Numbers from 3^{rd} Order Preconditioned Scheme at Subsonic Flow

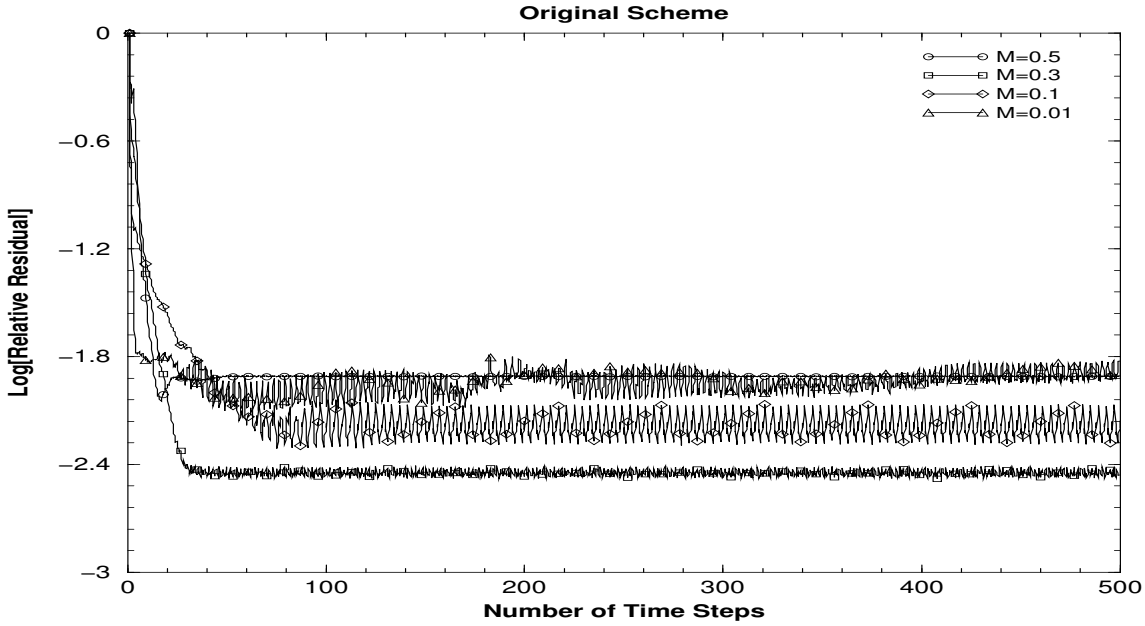

Figure 7.19: Third-Order Original Scheme Convergence History for various Mach numbers at Subsonic Flow

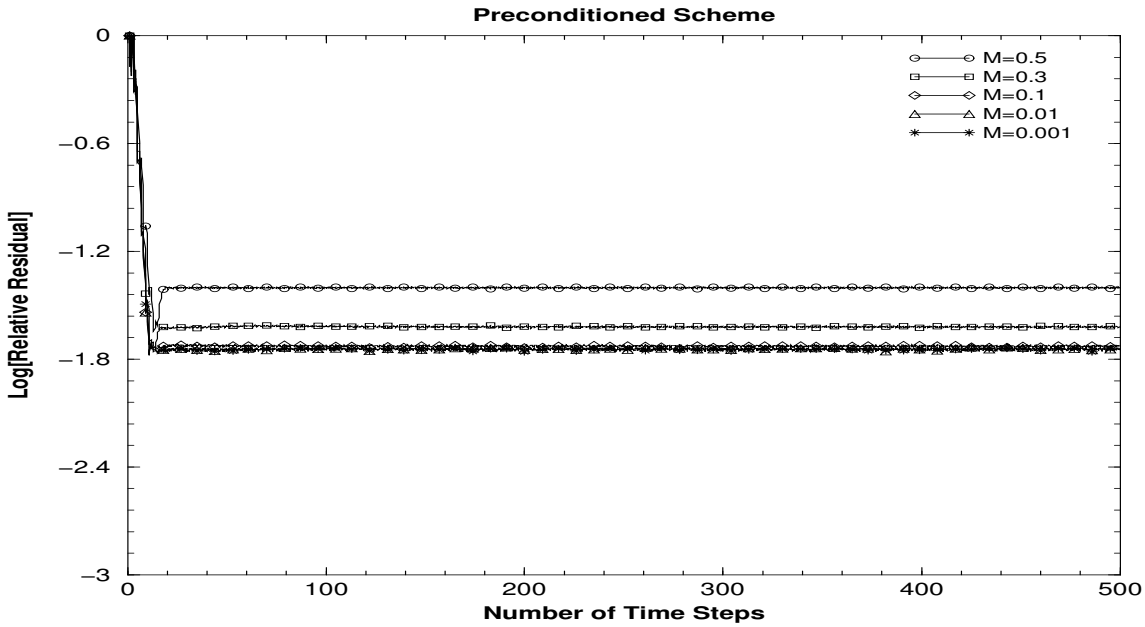

Figure 7.20: Third-Order Preconditioned Scheme Convergence History for various Mach numbers at Subsonic Flow

Table 7.1: Lift/Drag Coefficients comparisons at M=0.85 and $\alpha = 1^o$

	C_l	C_d
Original Scheme 1^{st} Order	0.3271	0.0689
2^{nd} Order	0.3600	0.0588
3^{rd} Order	0.3604	0.0586
Preconditioned Scheme 1^{st} Order	0.3230	0.0680
2^{nd} Order	0.3636	0.0592
3^{rd} Order	0.3599	0.0588
AGARD Test Cases	$0.330 \sim 0.3889$	$0.0464 \sim 0.0590$

the detailed view in Figs. (7.23) and (7.24)), and the results are in excellent agreement with those of AGARD. As seen in the detailed plots, higher-order schemes managed to capture the shock location better, and this will be shown to lead to more accurate results.

Integrating the pressure distribution in the flow and normal to flow directions results in lift and drag forces for this inviscid flow. The AGARD inviscid test cases are run for different mesh sizes and grid types, and different farfield locations. Lift/Drag comparisons of various order discretizations are given in Table (7.1). The values in the present study fall within the AGARD range for the high-order discretization. One should notice that first-order results are also reasonable, and they improve when using finer grid. As a final comparison, Mach contour lines are given in Fig.(7.26), and are in good agreement with the AGARD results.

For this study, convergence criteria are based on the behavior of the maximum residual and convergence rate suffers when using limiters in high-order schemes. A plot of the residual for the first order scheme is shown in Fig. (7.25) The convergence behavior for both original and preconditioned algorithms is virtually the same. It is worth mentioning that there is relatively minor gain occurring in transonic flow when using the preconditioned algorithm. Unlike first-order schemes, convergence rate for high-order discretizations shows some oscillatory, limit-cycle behavior. Second-order scheme converge three orders of magnitude 10^{-3} but third-order algorithms only convergence one order of magnitude. Incidentally, preconditioned high-order scheme has helped to reduce the magnitude of the limit-cycles.

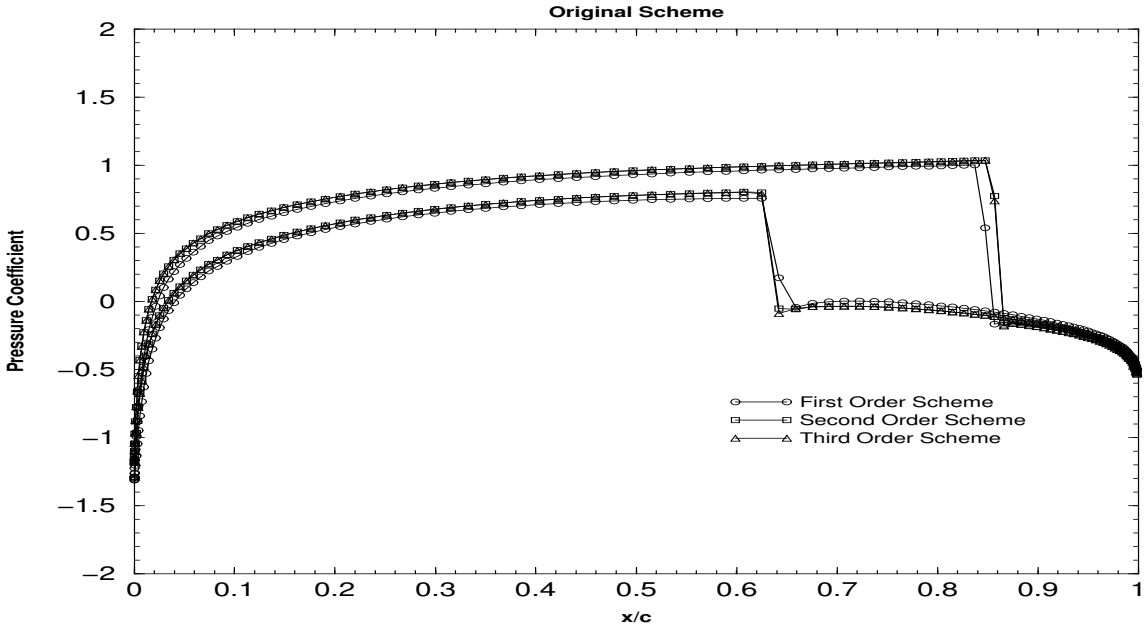

Figure 7.21: Pressure coefficient distribution over the transonic case at $M_\infty = 0.85$ and $\alpha = 1$, Original Scheme

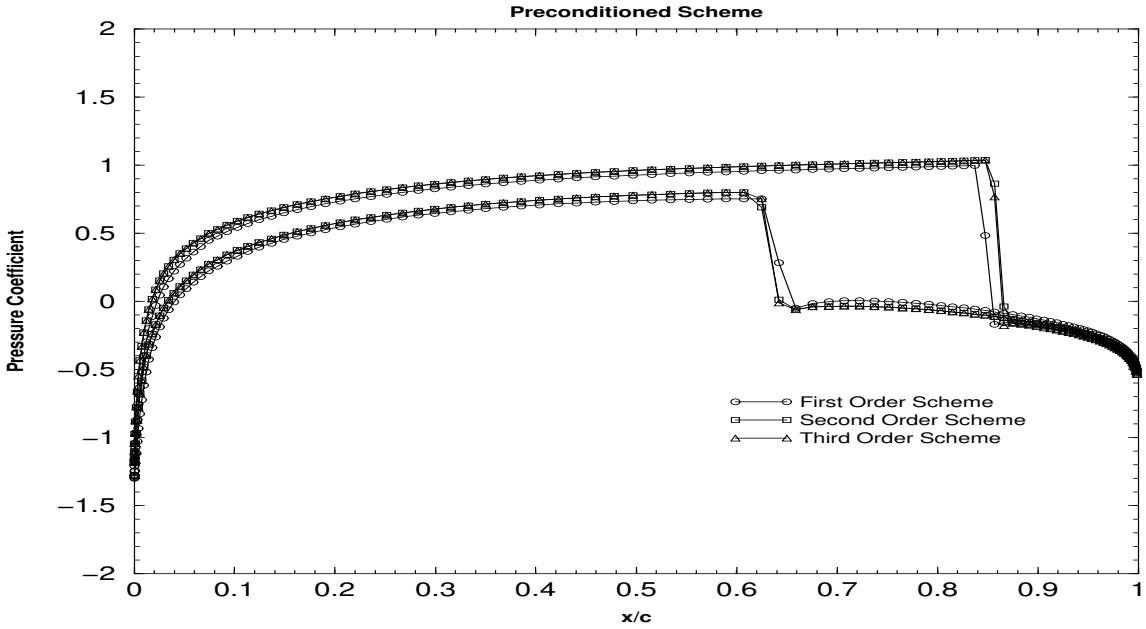

Figure 7.22: Pressure coefficient distribution over the transonic case at $M_\infty = 0.85$ and $\alpha = 1$, Preconditioned Scheme

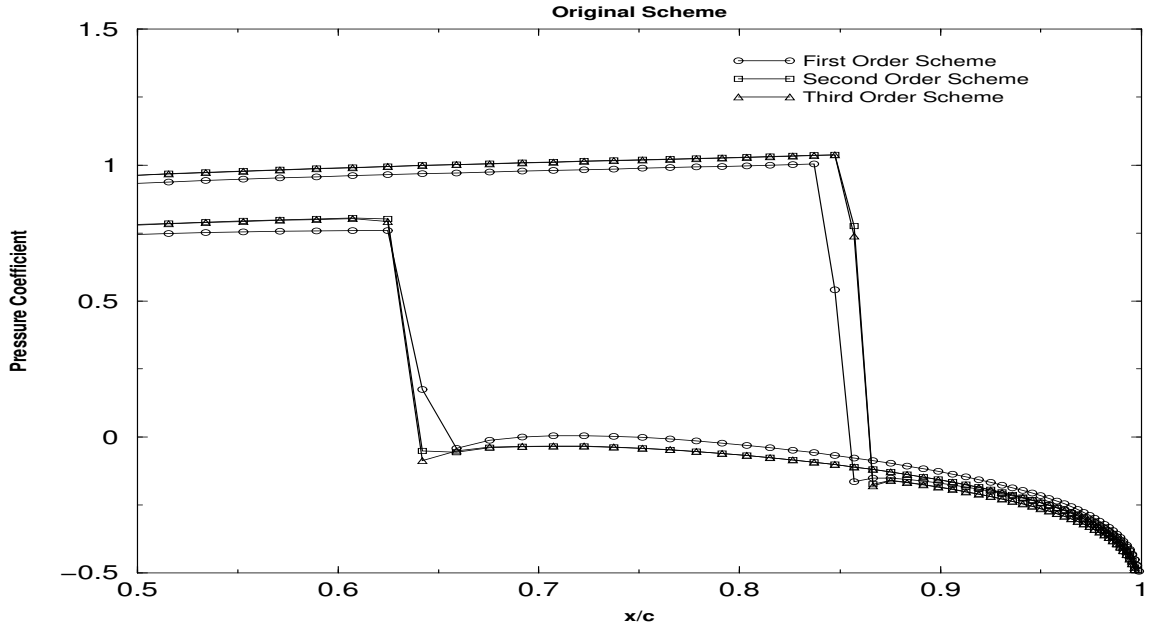

Figure 7.23: Detailed View of Pressure coefficient distribution over the transonic case at $M_\infty = 0.85$ and $\alpha = 1$, Original Scheme

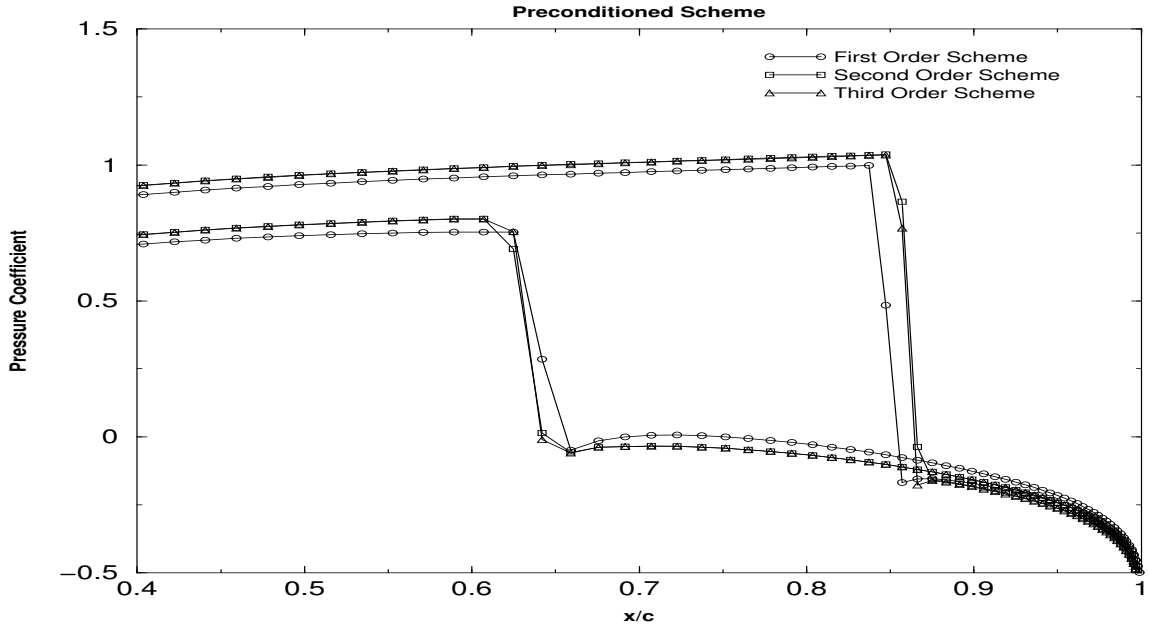

Figure 7.24: Detailed View of Pressure coefficient distribution over the transonic case at $M_\infty = 0.85$ and $\alpha = 1$, Preconditioned Scheme

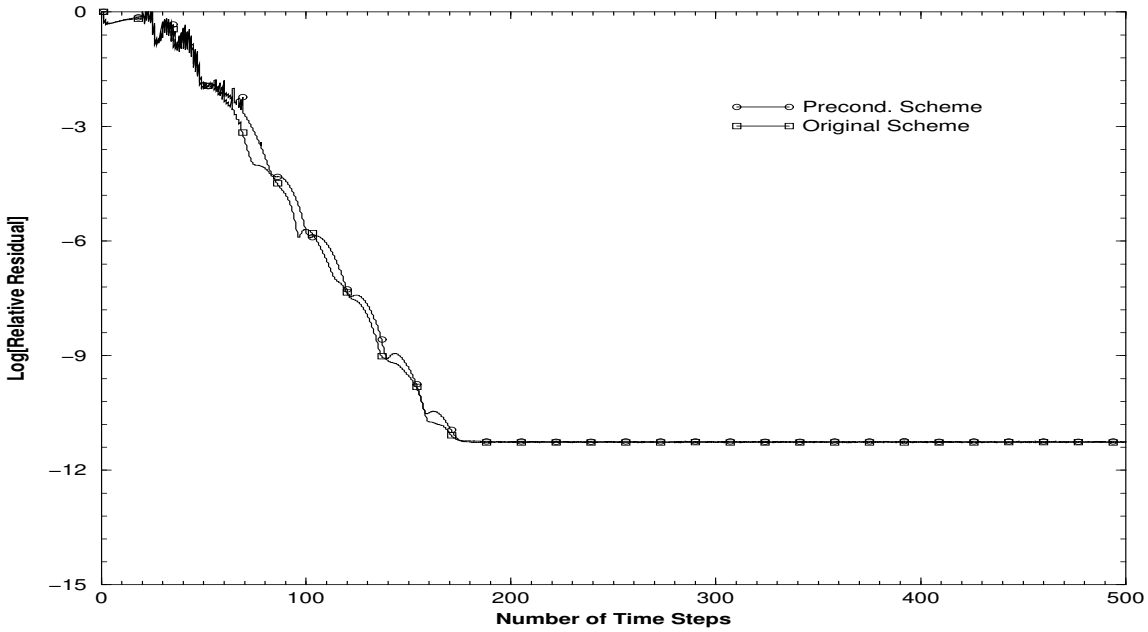

Figure 7.25: First-Order Original and Preconditioned scheme Convergence Histories for the transonic case ($M_\infty = 0.85$ and $\alpha = 1^o$)

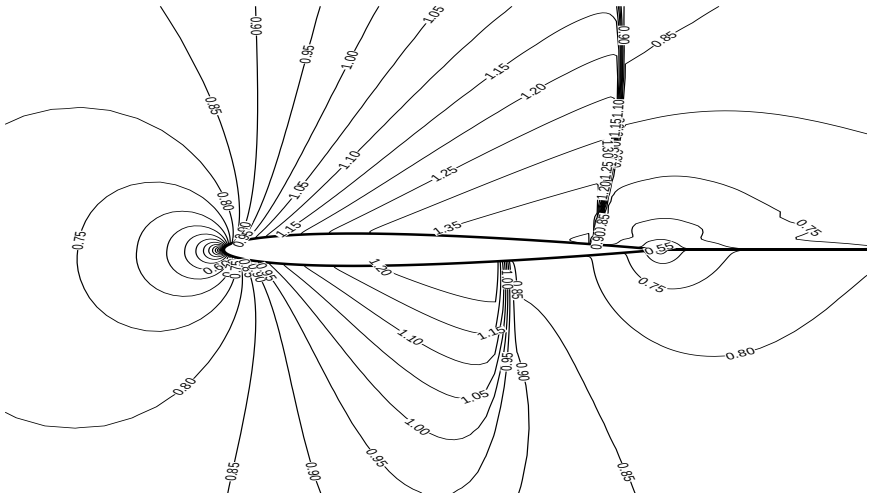

Figure 7.26: Mach contour lines for the transonic case ($M_\infty = 0.85$ and $\alpha = 1^o$)

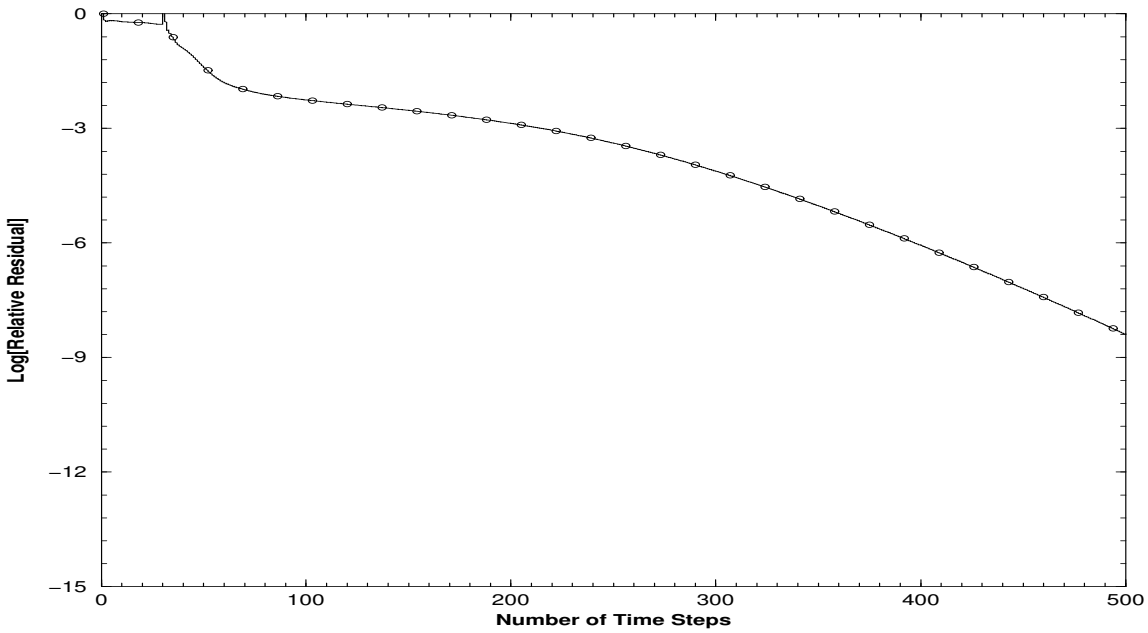

Figure 7.27: First-Order scheme Convergence History for the supersonic case ($M_\infty = 1.2$ and $\alpha = 0^o$)

7.2.3 Supersonic Test Case

This case is given for sake of completeness. The current preconditioning scheme is meant to work only for subsonic flows, and because it is a global preconditioner, it will not be active in the low-speed pockets near the stagnation region. This case is run at $M_\infty = 1.2$ with zero angle of attack on the finer grid. A plot of the residual reduction for the first-order scheme is given in Fig. (7.27). The first-order scheme is run at a CFL=60 after 50 iterations running at CFL=3.0, and the high-order scheme is run at a CFL=50 after 50 iterations running at CFL=2.0.

A comparison of drag coefficients is given in Table (7.2). It is noticed that the results are in good agreement with those in the AGARD report. Finally, Mach contour lines are shown in Fig. (7.28), and are in good agreement with the AGARD results.

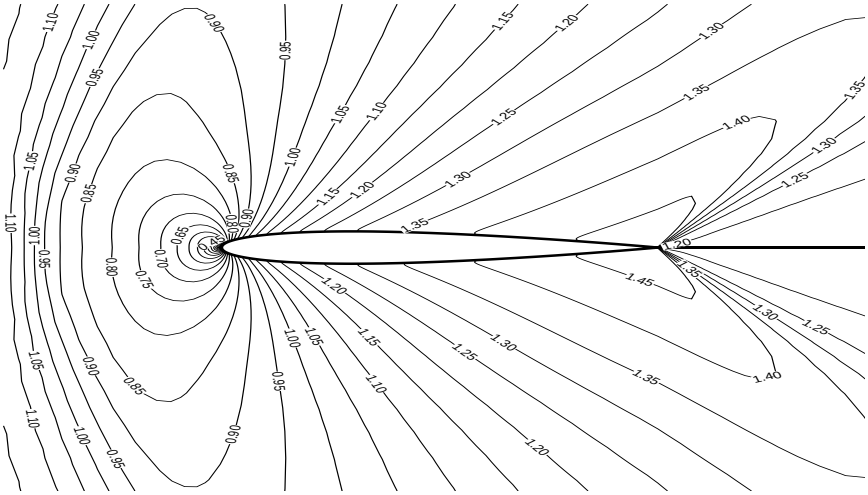

Figure 7.28: Mach contour lines for the supersonic case ($M_\infty = 1.2$ and $\alpha = 0^o$)

Table 7.2: Drag Coefficients comparisons at M=1.2 and $\alpha = 0^o$

	C_d
Original Scheme 1^{st} Order	0.1032
2^{nd} Order	0.0972
3^{rd} Order	0.0969
AGARD Test Cases	$0.0946 \sim 0.0960$

7.3 Steady-State, Internal Flows: Nozzle Tests

A Convergent-Divergent (C-D) nozzle is an excellent test case for internal flows, because of the availability of quasi one-dimensional analytical solutions. The nozzle geometry is from reference [32], with $A_{inlet}/A_{throat} = 2.5$ and $A_{exit}/A_{throat} = 1.5$. The nozzle is discretized by means of a 202×62 grid, shown in Fig. (7.29). CVBCs are imposed on exit and inlet of the nozzle.

Three major types of flow can be encountered in C-D Nozzle, depending on p_e/p_o ratios. The first one is fully subsonic flow for various, prescribed, exit pressures. Lowering the exit pressure, a shock standing in the diffuser can be formed, and this case can be used to test the performance of the schemes for mixed flow types. The third flow type is a fully

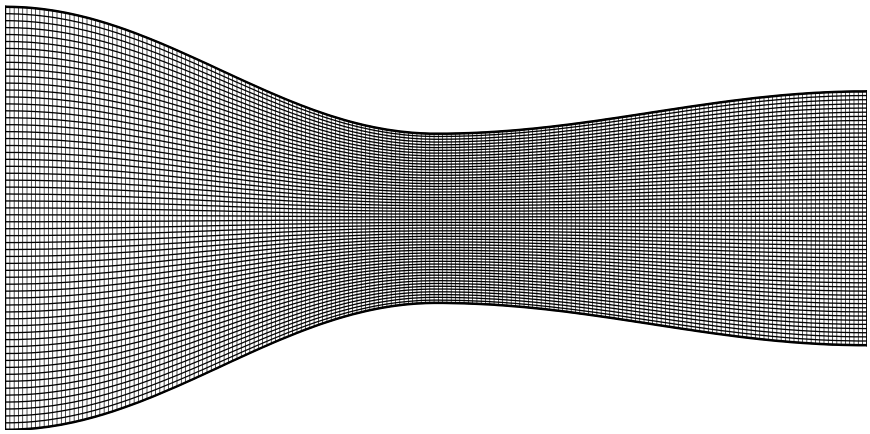

Figure 7.29: Grid used for the Convergent-Divergent Nozzle (202×62)

supersonic case. In all cases the reference Mach number, M_r, is taken to be the value at the inlet of the nozzle. The quasi-one-dimensional solution is also evaluated for these flow types.

7.3.1 Subsonic Nozzle

This case is designed to test the reliability and robustness of the preconditioning scheme for internal flow applications. The flow will accelerate/decelerate throughout the nozzle, according to the prescribed exit pressure. In this case, four different p_e/p_o ratios were imposed (0.999998, 0.9998, 0.98 and 0.89). It is a well-known result that the flow accelerates by further decreasing the exit pressure, until sonic conditions are established at the throat. Both schemes are run with a CFL=100, and comparisons are made between axial pressure and analytical solution. Note that the area-averaged results for each cross-section could be slightly closer to those of the exact solution.

The axial pressure distributions are compared to the analytical solution in Figs.(7.30), (7.31), (7.32), (7.33). It is seen that the preconditioned scheme features a better solution than the original first-order scheme. Notice that first order scheme results do not deviate

too much from the solution (the vertical scales in these plots are *extremely* enlarged at the higher pressures). However, if one looks closely at the lower the inlet Mach number cases, it is found that second-order original scheme solution shows some oscillations, which are not found in the preconditioned scheme. These oscillations actually grow larger as A_e/A_t is increased (this is now shown in the present study).

For the first-order original scheme the convergence histories deteriorate as the Mach number goes further down, as shown in Fig. (7.34). The preconditioned scheme makes the residual reduction roughly independent of Mach number at low speed flows, as seen in Fig.(7.35). However, an interesting result which was not observed in the airfoil case is that convergence is better for the original scheme when used for relatively high-speed subsonic nozzle (such as $p_e/p_o = 0.89$). As for high-order schemes, the same patterns are encountered for convergence histories, as seen in Figs. (7.36) and (7.37). Mach number independence in the residual history can be seen in Fig. (7.37) for the preconditioned scheme. Finally, Mach number contours are given for $p_e/p_o = 0.89$ in Fig. (7.38).

7.3.2 Nonisentropic Choked Nozzle

The flow in the convergent part of the nozzle will keep accelerating as p_e/p_o goes down until attaining sonic conditions at the throat. Further decreasing exit pressure will not change anything in convergent part, but will affect the divergent part. This flow is called a choked flow, and is no longer isentropic in the divergent section. A normal shock wave appears in the divergent part, an its location depends on the exit pressure. In this study, p_e/p_o is chosen as 0.75, in order to have a normal shock wave roughly halfway in the diffuser. Original and preconditioned schemes are run with a CFL=20. A lower CFL is used for a few iterations at the beginning. The analytical solution is set as the initial condition. Comparison are with values at the center line or area-averaged.

There is no expectation of superiority for the scheme in this case. In fact, this nozzle is a good example of a mixed type flow without a stagnation point. The first comparisons are made between axial pressure distribution and analytical solution in Fig. (7.39). All solutions are in very good agreement. The convergence graphs are shown in Fig. (7.40).

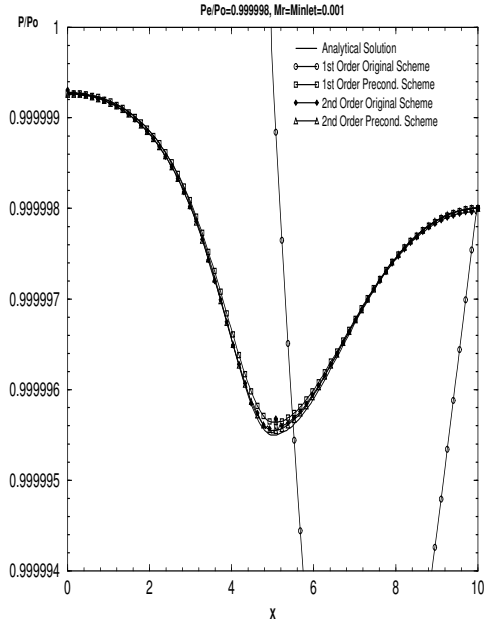

Figure 7.30: Pressure ratios, $\frac{p}{p_o}$, distribution along convergent-divergent nozzle for $M_r = 0.001$

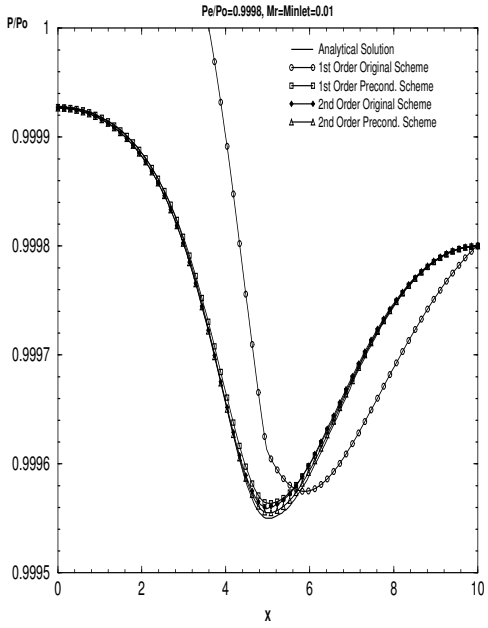

Figure 7.31: Pressure ratios, $\frac{p}{p_o}$, distribution along convergent-divergent nozzle for $M_r = 0.01$

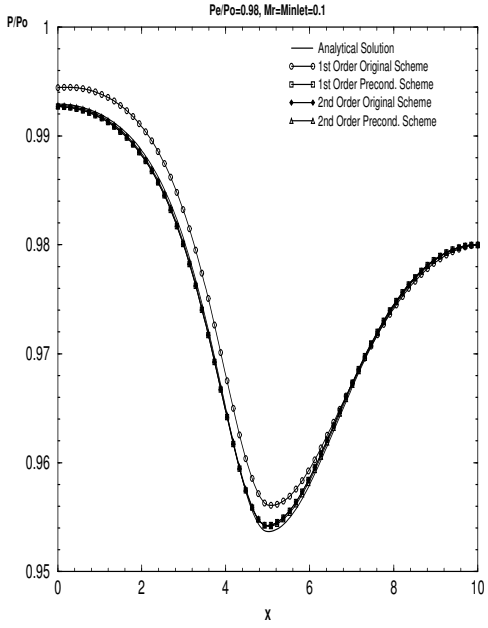

Figure 7.32: Pressure ratios, $\frac{p}{p_o}$, distribution along convergent-divergent nozzle for $M_r = 0.1$

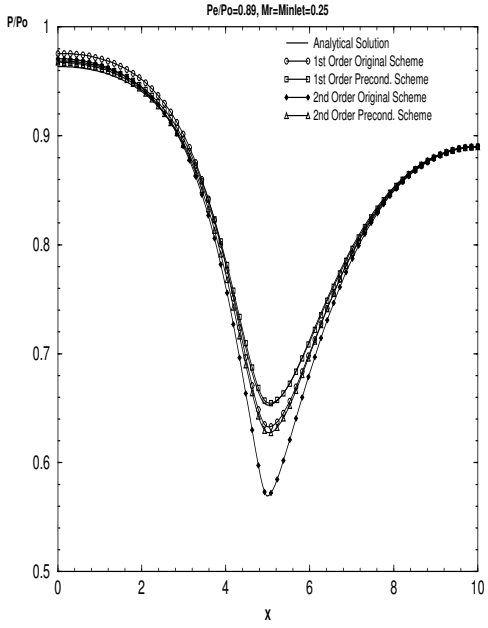

Figure 7.33: Pressure ratios, $\frac{p}{p_o}$, distribution along convergent-divergent nozzle for $M_r = 0.25$

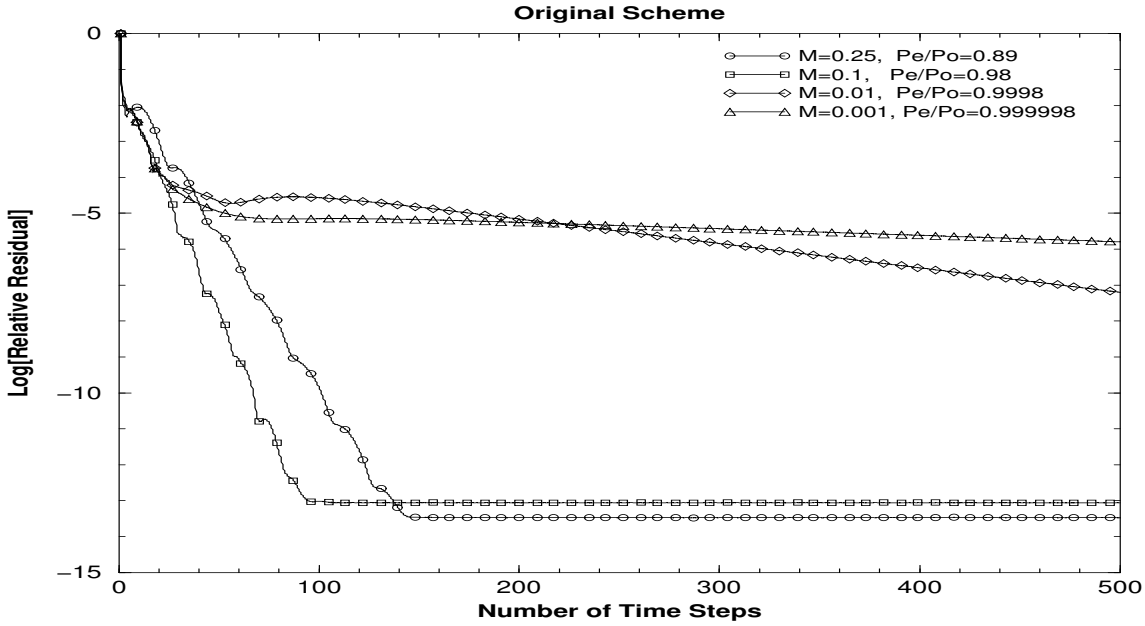

Figure 7.34: First-Order Scheme Convergence Histories for Subsonic Nozzle

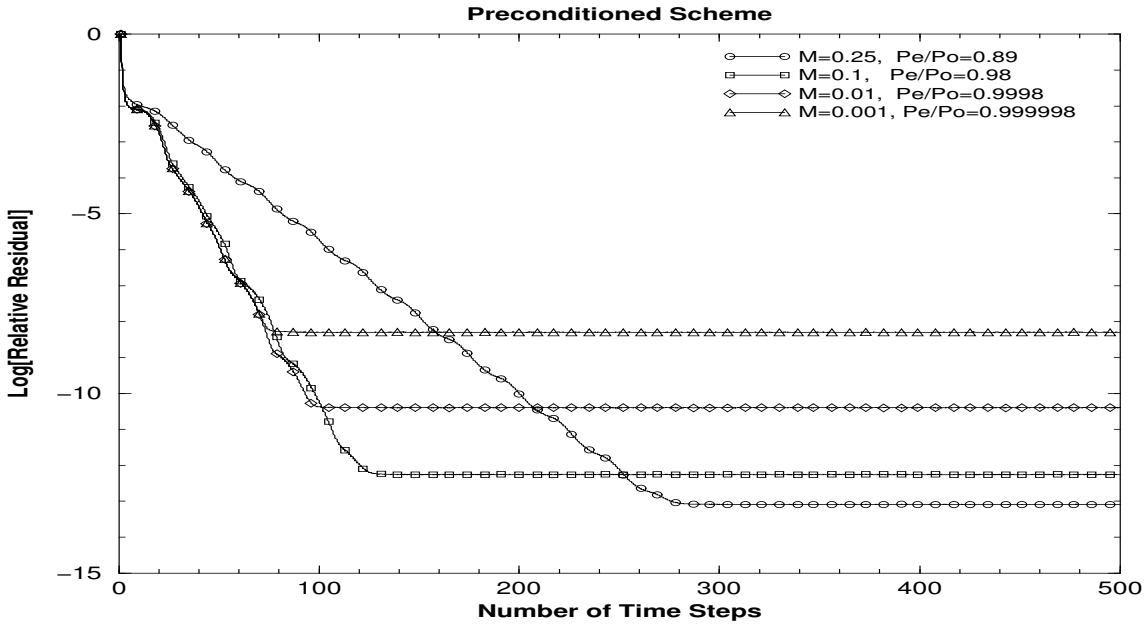

Figure 7.35: First-Order Scheme Convergence Histories for Subsonic Nozzle

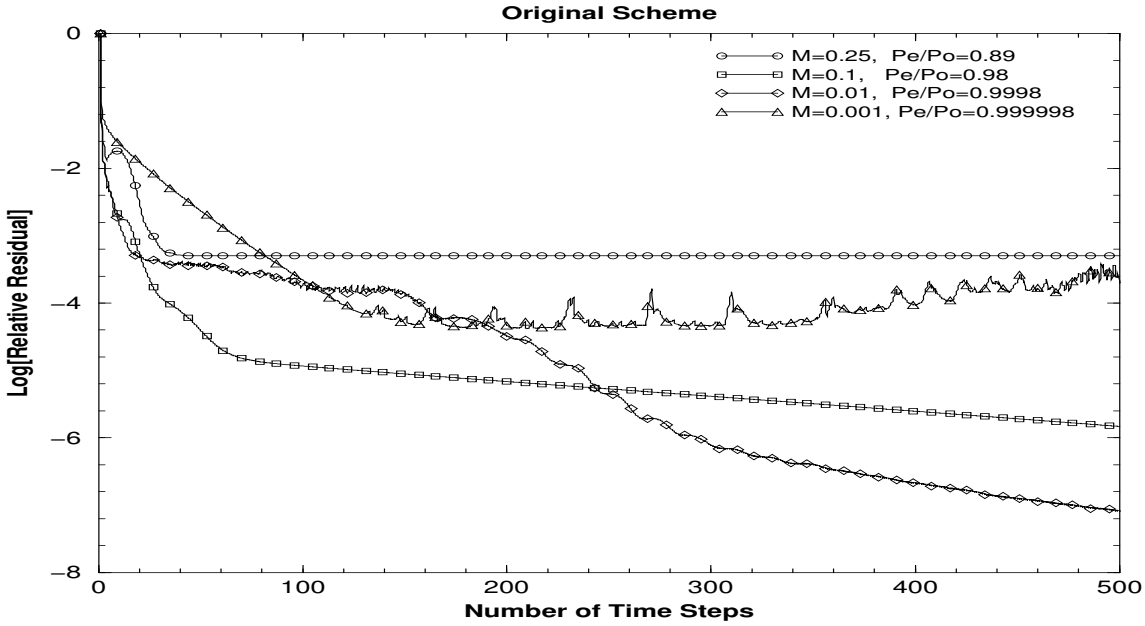

Figure 7.36: Second-Order Scheme Convergence Histories for Subsonic Nozzle

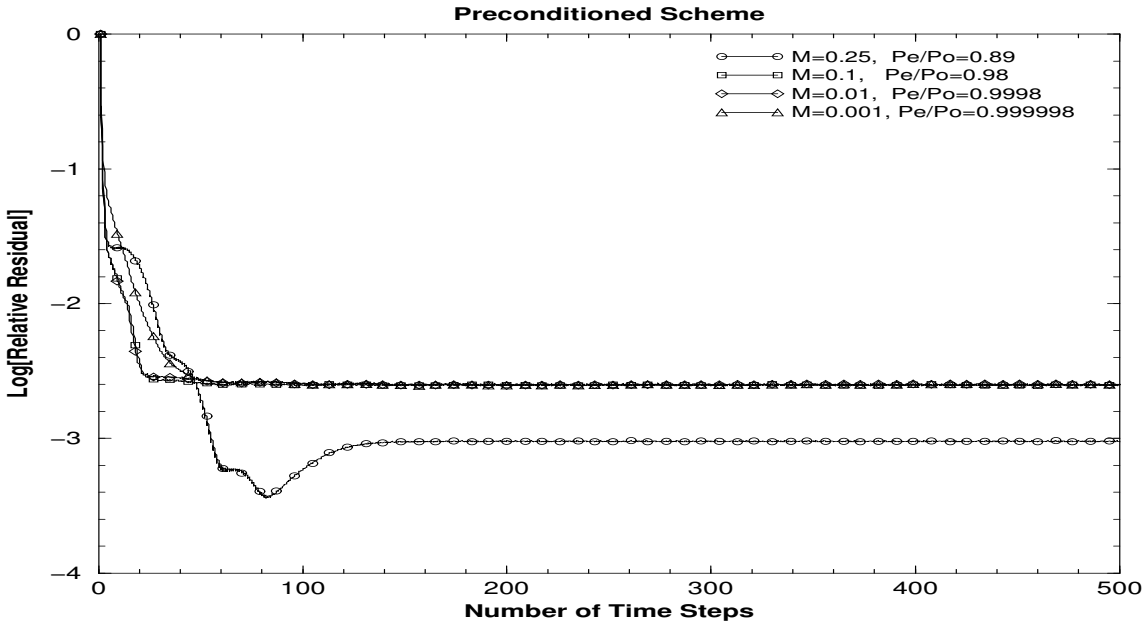

Figure 7.37: Second-Order Scheme Convergence Histories for Subsonic Nozzle

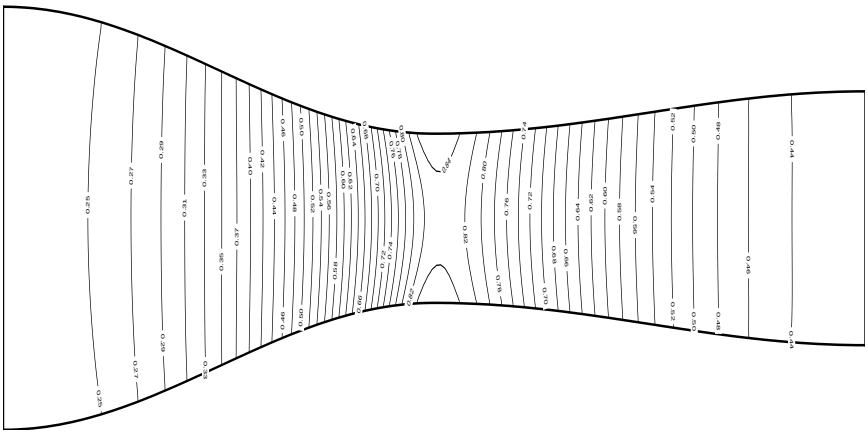

Figure 7.38: Mach contours for $\frac{p_e}{p_o} = 0.89$

The first-order original scheme is superior to the the preconditioned one in convergence rate, but the opposite is true for the high-order scheme.

This nozzle was also studied by using the WIND CFD solver at NASA [33]. The solution behind the shock there was largely disturbed. The author of the study blames that on the presence of interactions between the the normal shock and the pressure subsonic outflow. The present results for the given exit pressure, shown in Fig. (7.41), show excellent agreement with the analytical solution and no disturbances . This should be attributed to the implementation of characteristic variable boundary conditions (CVBCs). In external flow, farfield boundaries are located in regions where very small changes occur; consequently the importance of boundary conditions is somewhat diminished. Here, the importance of CVBCs is more clearly demonstrated. Finally, Mach contours are given in Fig.(7.42).

7.3.3 Supersonic Nozzle

When the exit pressure is reduced further, one can finally obtain a fully supersonic isentropic solution (at the design exit Mach number). The exit pressure ratio is set to 0.16 to attain this condition. Both schemes are run at CFL=100, using the analytical solution

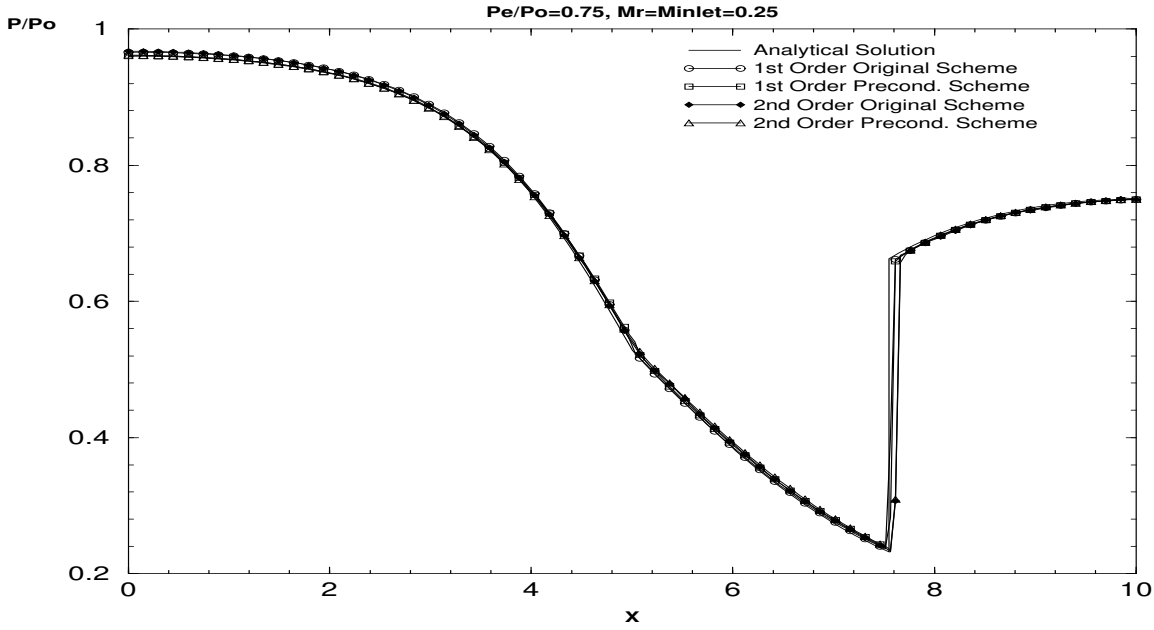

Figure 7.39: Pressure ratios, $\frac{p}{p_o}$, distribution along convergent-divergent nozzle for choked flow

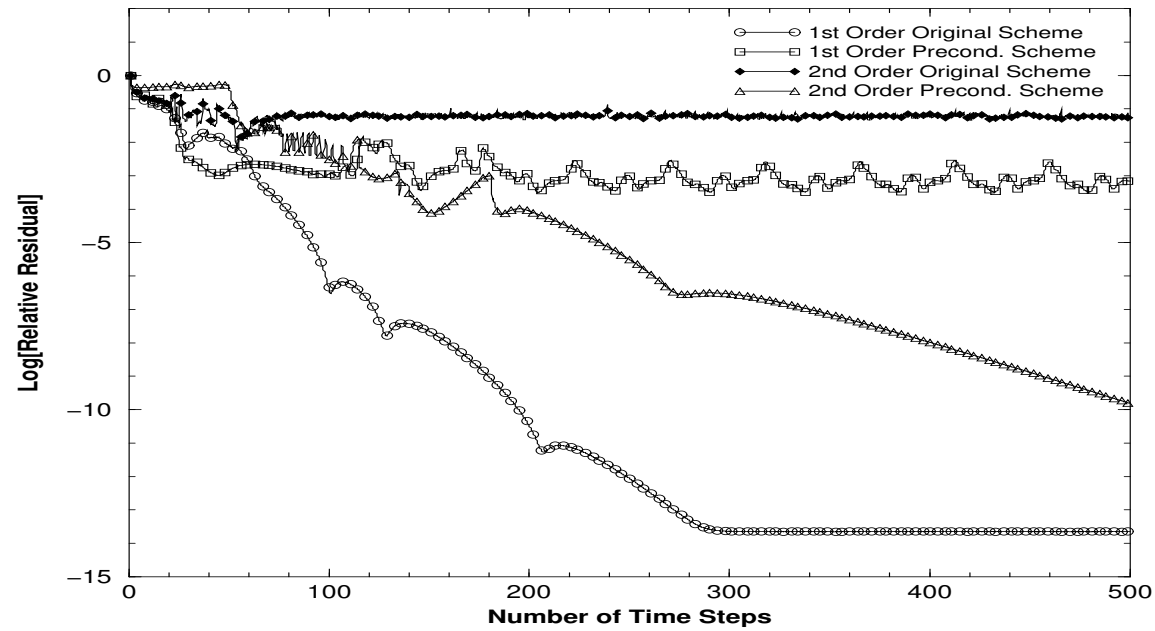

Figure 7.40: Convergence Histories for nonisentropic nozzle $\frac{p_e}{p_o} = 0.75$

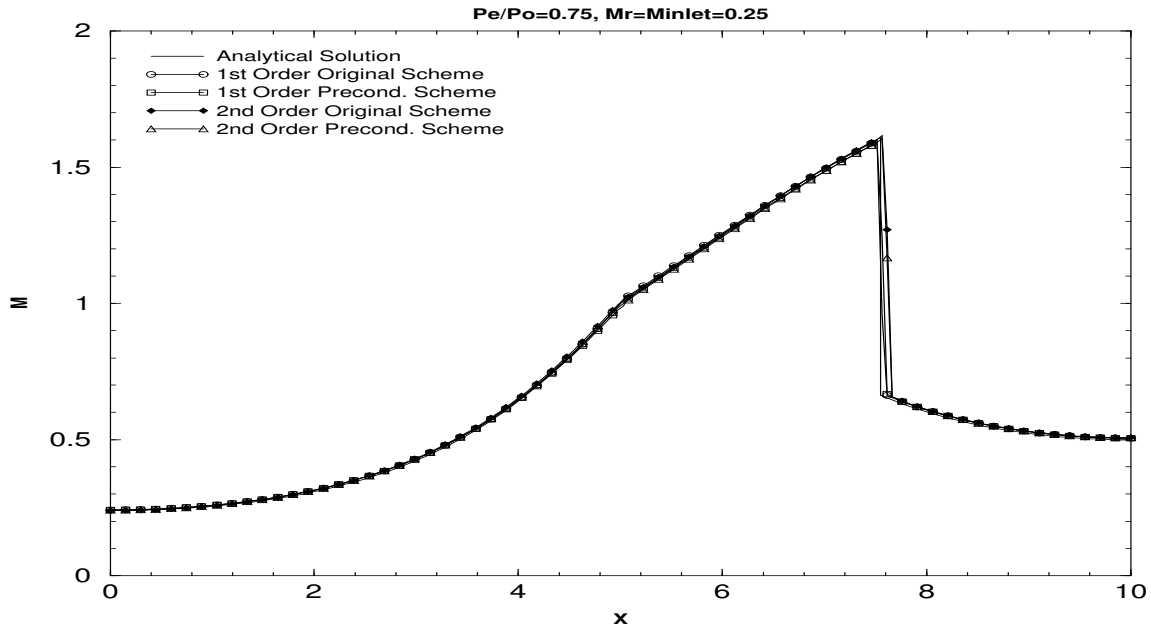

Figure 7.41: Average Mach number distribution along convergent-divergent nozzle for choked flow

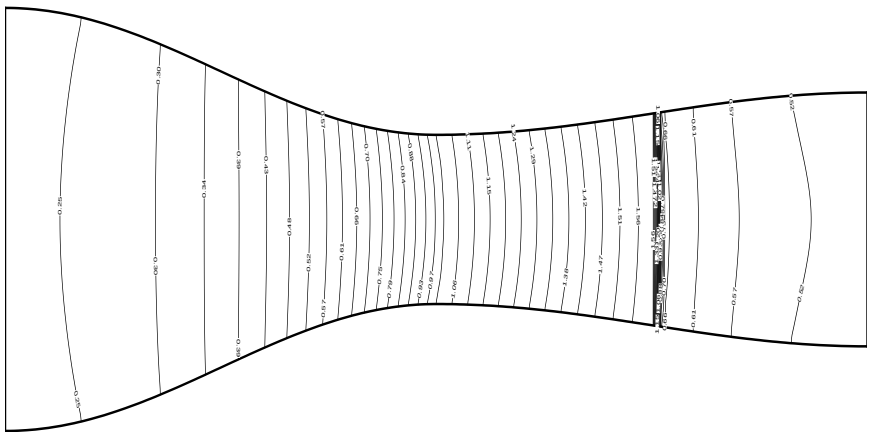

Figure 7.42: Mach contours for $\frac{p_e}{p_o} = 0.75$

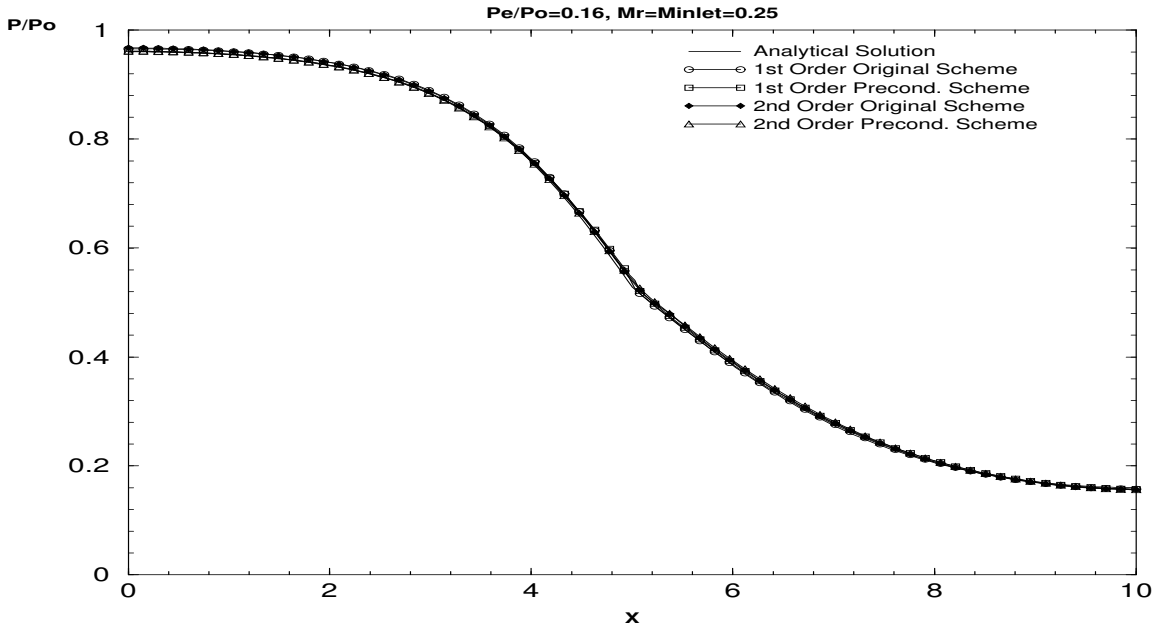

Figure 7.43: Pressure ratios, $\frac{p}{p_o}$, distribution along convergent-divergent nozzle for supersonic nozzle

as initial condition. The centerline pressure distribution is in very good agreement with the analytical solution, as seen in Fig.(7.43) for all schemes. Again, the original scheme residual reduction is superior to the one for the preconditioned algorithm, as seen in Fig.(7.44). Various larger A_{inlet}/A_{throat} were also run in order to further investigate this behavior (at larger are ratios, the flow speed decreases in the convergent portion). The original scheme is still outperforming the preconditioned one. Mach contours are given in Fig.(7.45).

7.4 Unsteady Case: Shock Tube Problem

The shock tube problem is computed as a preliminary unsteady test case on a 201×6 grid, shown in Fig. (7.46), with the following initial conditions:

$$q_L(\rho, u, v, p) = \{1, 0, 0, \frac{1}{\gamma}\}^T \tag{7.1a}$$

$$q_R(\rho, u, v, p) = \{\frac{1}{10}, 0, 0, \frac{1}{10\gamma}\}^T \tag{7.1b}$$

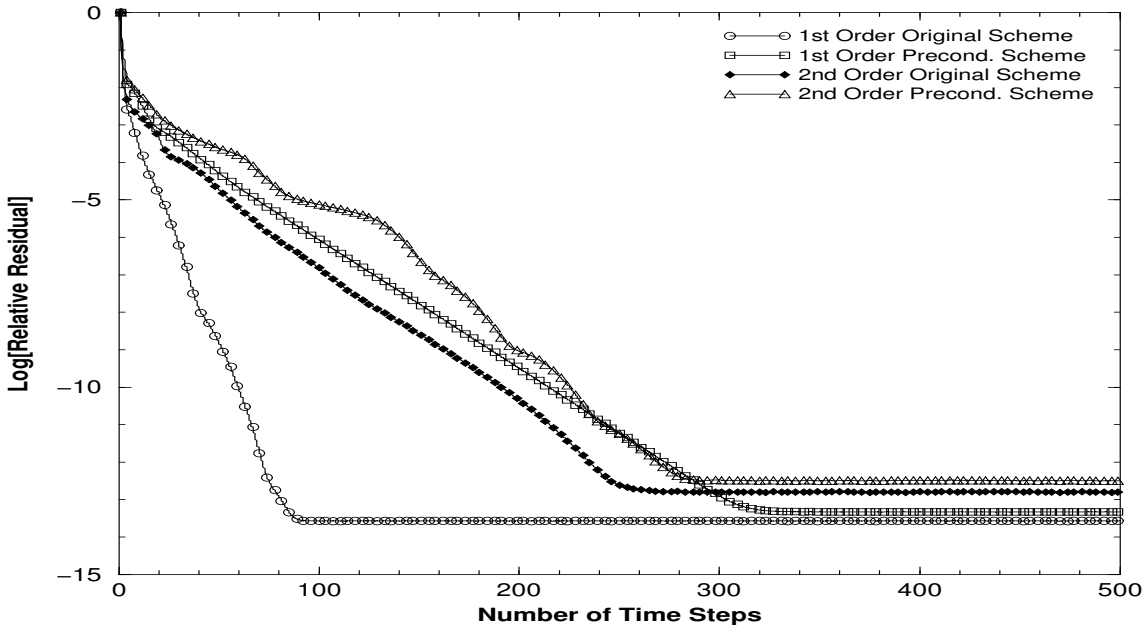

Figure 7.44: Pressure ratios, $\frac{p}{p_o}$, distribution along convergent-divergent nozzle for supersonic nozzle

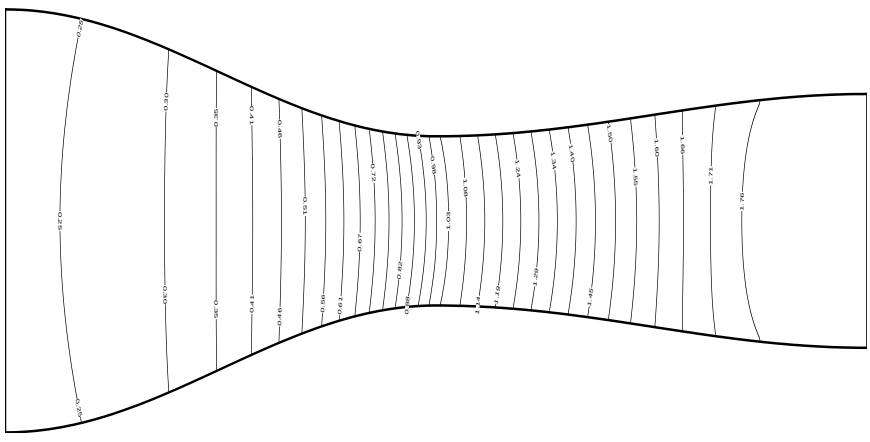

Figure 7.45: Mach contours for $\frac{p_e}{p_o} = 0.16$

Initial discontinuity is located in the middle of the shock tube. The purpose of this case is to test the unsteady algorithm for varying reference Mach numbers. As mentioned before, introducing a preconditioning matrix has no effect on steady-state cases, but it destroys time accuracy. In order to recover time-accurate problems, the implicit time integration with Newton iterations and symmetric Gauss-Seidel (LU/SGS) sweeps is employed. Third-order fluxes are used in all computations.

The shock tube problem has two discontinuities which are the shock and the contact surface. The discontinuities are proceeding to the right as time advances. All computations are run to a time of 0.17 with time steps dt=0.00017 (1000 steps), and compared to available analytical results. Any two-dimensional Euler code should be applicable to a one-dimensional shock tube problem when proper boundary conditions are used. The boundary conditions located at the left and right boundaries of the shock tube are the initial conditions.

The results are compared to the analytical solution in Fig. (7.47) for various reference Mach number. All numerical results are in good agreement with analytical solution. In case of $M_r = 1.0$, the preconditioning does not destroy time accuracy, hence in this case the residual reached to a fifteen order of magnitude reduction with 15 Newton iterations at every time step. Even though this problem does not require preconditioning, the preconditioning is introduced to verify the recovery of time accuracy. Reference Mach numbers such as 0.5 and 0.3 are chosen for the preconditioning matrix. In order to reach acceptable order of magnitude residual reduction, one may have to use more Newton iterations. The reference Mach number 0.5 case is required to run 20 Newton iterations to attain four orders of magnitude in residual reduction, and the case $M_r = 0.3$ does require 50 Newton iterations to reach the same residual reduction. Note that Newton iteration convergence at every time steps is not important in steady-state cases, but it is very important in any unsteady case.

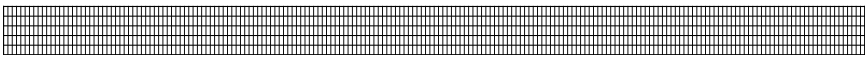

Figure 7.46: Grid for Shock Tube Problem (201×6)

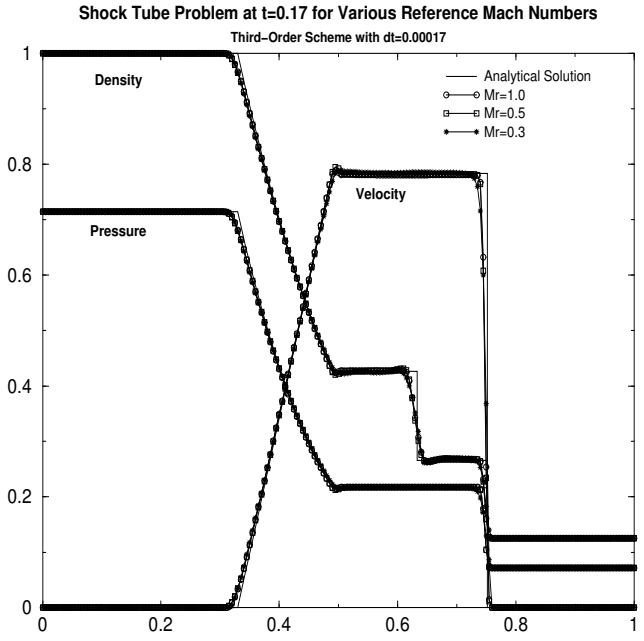

Figure 7.47: Mach contours for $\frac{p_e}{p_o} = 0.75$

CHAPTER VIII

SUMMARY AND CONCLUDING REMARKS

Numerical solutions of the two-dimensional Euler equations were investigated using a finite volume formulation in conjunction with Roe flux difference splitting and explicit and implicit time integration algorithms. A discretized-Newton linearization with nested (LU/SGS) subiterations was used for solving the conservative implicit scheme. A global preconditioning matrix was introduced in the hyperbolic system, in order to improve convergence and preserve accuracy at low speeds. CVBCs were developed for the preconditioned scheme at farfield boundaries. Tangency, along with constant enthalpy and pressure gradient, were imposed as wall boundary conditions.

A few traditional test cases were performed, with the aim to measure the ability of the new global preconditioner to accommodate low-speed flows. All test cases exclude implicit integration and boundary conditions effects, in order to observe only the preconditioner influence. Point disturbance tests show that the new preconditioner makes the convergence rate independent of Mach number as the Mach number approaches zero; however, its convergence rate falls slightly behind other local preconditioners. Furthermore, flow angle and stagnation point tests prove that the new global preconditioner is extremely robust for highly deflected flow angles and captures perfectly the stagnation point.

An external flow application involving a NACA 0012 airfoil was investigated next. This test case shows that new global preconditioning retains accuracy as well as a high convergence rate for a realistic case. For subsonic speeds, the new scheme successfully fixed the inaccuracy found with compressible algorithms at very low Mach numbers. A transonic case was also investigated, in order to assess the preconditioning converging ability, although the original scheme works fine in the transonic case (transonic airfoil results did not show any significant gains in convergence rate).

An converging-diverging nozzle was chosen to test the new scheme for internal flows. The convergence rate for the new global preconditioning is roughly independent of the Mach number, and the convergence only accelerates at very low speeds. Otherwise, it is apparent that the original scheme has better convergence properties than the new preconditioned scheme in the nozzle case. The author investigated larger A_{inlet}/A_{throat} ratios, thus increasing the region of very subsonic flow; however, the convergence rate was essentially unchanged. Quality and importance of boundary conditions were demonstrated in the case of a shock standing in the diffuser section.

For unsteady cases, the preconditioned scheme destroys time accuracy. Newton formulation is designed to overcome this problem, at the price of increasing the number of Newton iteration per time steps.

The new global preconditioning was proven to be numerically very reliable and robust, and it compares favorably with local preconditioners. Future work should focus on investigating its convergence acceleration ability in more detail. Moreover, the current implementation should be extended to three space dimensions, viscous cases using the Navier-Stokes equations, and reactive flow problems (e.g. combustion).

REFERENCES

[1] E. Turkel, V. N. Vatsa, and R. Radespiel, "Preconditioning Methods for Low-Speed Flows," tech. rep., NASA Langley Research Center, NASA Contract No. NAS1-19480, NASA Langley Research Center, Hampton, VA 23681-0001, April 1995.

[2] M. R. Zaccanti, *Analysis and Design of Preconditioning Methods for The Euler Equations*. PhD thesis, Mississippi State University, December 1999.

[3] D. L. Darmafol and B. V. van Leer, "Local Preconditioning of The Euler Equations: A Characteristic Interpretation," in , Lecture Series 1999-03, VKI, VKI, 1999.

[4] W. R. Briley, H. McDonald, , and S. J. Shamroth, "A Low Mach Number Euler Formulation and Application to Time-Iterative LBI Schemes," *AIAA*, vol. 21, pp. 1467–1469, October 1983.

[5] D. Lee, *Local Preconditioning of The Euler and Navier-Stokes Equations*. PhD thesis, University of Michigan, 1996.

[6] D. L. Darmafol and P. J. Schmid, "The Importance of Eigenvectors for Local Preconditioners of the Euler Equations," *AIAA Paper 95-1655*, 1995.

[7] W.R.Briley, L. K. Taylor, , and D. L. Whitfield, "High-Resolution Viscous Flow Simulations at Arbitrary Mach Number," *Journal of Computational Physics*, vol. 184, pp. 79–105, January 2003.

[8] A. Fiterman, E. Turkel, and B. van Leer, "Preconditioning and The Limit to The Incompressible Flow Equations," Tech. Rep. ICASE REPORT No.93-42, NASA, 1995.

[9] J. M. Janus, "The Development of a Three-dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications," Master's thesis, Mississippi State University, August 1984.

[10] F. M. White, *Viscous Fluid Flow*. McGRAW-HILL, 1974.

[11] C. B. Laney, *Computational Gasdynamics*. Cambridge University Press, 1998.

[12] R. J. L. Veque, *Numerical Methods for Conservation Laws*. Springer Verlag, 1992.

[13] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer Verlag, 1999.

[14] P. L. ROE, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, vol. 43, pp. 357–372, 1981.

[15] D. L. Whitfield, J. M. Janus, and L. B. Simpson, "Implicit Finite Volume High Resolution Wave-Split Scheme for Solving the Unsteady Three Dimensional Euler and Navier-Stokes Equations on Stationary or Dynamic Grids," tech. rep., NASA Lewis Research Center, February 1988.

[16] Z. U. A. Warsi, *Fluid Dynamics Theoretical and Computational Approaches*. CRC Press, 1999.

[17] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*. Taylar&Francis, 1997.

[18] C. Hirsch, *Numerical Computation of Internal and External Flows*. John Wiley&Sons, 1988.

[19] A. Chorin, "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, vol. 2, pp. 12–26, August 1967.

[20] E. Turkel, "A Precondtioned Methods for Solving the Incompressible and Low-Speed Compressible Flows," *Journal of Computational Physics*, pp. 277–298, 1987.

[21] J. F. III and M. B. Giles, "Accelerated Convergence of Euler solutions Using Time Inclining," *AIAA*, vol. 28, pp. 1457–1463, August 1990.

[22] D. Choi and L. Merkle, "Application of Time-Iterative Schemes to Incompressible Flow," *AIAA*, vol. 23, pp. 1518–1524, December 1984.

[23] A. Fiterman, E. Turkel, and B. van Leer, "Pressure Updating Methods for The Steady-State Fluid Equations," Tech. Rep. ICASE REPORT No.93-42, NASA, 1995.

[24] J. M. Weiss and W. A. Smith, "Preconditioning Applied to Variable and Constant Density Flows," *AIAA*, vol. 33, pp. 2050–2057, November 1994.

[25] M. R. Zaccanti and P. Cinnella, "Effective Preconditioning Methods for The Euler Equations," *AIAA*, pp. 2000–2253, June 2000.

[26] W.R.Briley and H.McDonald, "An Overview and Generalization of Implicit Navier-Stokes Algorithms and Approximate Factorization," *Computer and Fluids*, vol. 21, pp. 1467–1469, March 2001.

[27] D. L. Whitfield and L. K. Taylor, "Numerical Solution of The Two-Dimensional Time-Dependent Incompressible Euler Equations," Tech. Rep. MSSU-EIRS-ERC-93-14, MSU, 1994.

[28] J. C. Newman, III, W. Anderson, and D. L. Whitfield, "Multidisciplinary Sensitivity Derivatives Using Complex Variables," Tech. Rep. MSSU-COE-ERC-98-08, MSU, 1998.

[29] T. W. Swafford, *Computational Fluid Dynamics, An Introduction to the Governing Equations and Formulations for Numerical Solution*. Mississippi State University, 1994.

[30] A. Dadone and B. Grossman, "Surface boundary conditions for the numerical solution of the euler equations," *AIAA Journal*, vol. 32, pp. 285–293, February 1994.

[31] AGARD, "Test Cases for Inviscid Flowfield Methods : Report / of Fluid Dynamics Panel Working Group 07.," Tech. Rep. AGARD AR-211, NATO, May 1985.

[32] M. Liou, "Generalized Procedure Constructing an Upwind-Based TVD Scheme," *AIAA*, pp. 87–0355, January 1987.

[33] J. W. Slater, "NPARC Alliance CFD Verification and Validation Archieve," *NASA*, p. www.grc.nasa.gov/WWW/wind/valid, February 2001.

APPENDIX A

CURVILINEAR TRANSFORMATION

In order to derive a transformed version of the governing equation (2.11), one starts with the following relations between Cartesian x, y, z and general curvilinear coordinates ξ, η, ζ (also included in possibility of a change in the time coordinate, from t to τ):

$$\begin{aligned} \tau &= \tau(t), & t &= t(\tau), \\ \xi &= \xi(x,y,z,t), & x &= x(\xi,\eta,\zeta,\tau), \\ \eta &= \eta(x,y,z,t), & y &= y(\xi,\eta,\zeta,\tau), \\ \zeta &= \zeta(x,y,z,t), & z &= z(\xi,\eta,\zeta,\tau). \end{aligned} \tag{A.1}$$

Partial derivatives with respect to curvilinear and Cartesian coordinates can be written as follows, using the rule of differentiation:

$$\frac{\partial()}{\partial\tau} = \frac{\partial()}{\partial t}\frac{\partial t}{\partial\tau} + \frac{\partial()}{\partial x}\frac{\partial x}{\partial\tau} + \frac{\partial()}{\partial y}\frac{\partial y}{\partial\tau} + \frac{\partial()}{\partial z}\frac{\partial z}{\partial\tau}, \tag{A.2a}$$

$$\frac{\partial()}{\partial\xi} = \frac{\partial()}{\partial t}\frac{\partial t}{\partial\xi} + \frac{\partial()}{\partial x}\frac{\partial x}{\partial\xi} + \frac{\partial()}{\partial y}\frac{\partial y}{\partial\xi} + \frac{\partial()}{\partial z}\frac{\partial z}{\partial\xi}, \tag{A.2b}$$

$$\frac{\partial()}{\partial\eta} = \frac{\partial()}{\partial t}\frac{\partial t}{\partial\eta} + \frac{\partial()}{\partial x}\frac{\partial x}{\partial\eta} + \frac{\partial()}{\partial y}\frac{\partial y}{\partial\eta} + \frac{\partial()}{\partial z}\frac{\partial z}{\partial\eta}, \tag{A.2c}$$

$$\frac{\partial()}{\partial\zeta} = \frac{\partial()}{\partial t}\frac{\partial t}{\partial\zeta} + \frac{\partial()}{\partial x}\frac{\partial x}{\partial\zeta} + \frac{\partial()}{\partial y}\frac{\partial y}{\partial\zeta} + \frac{\partial()}{\partial z}\frac{\partial z}{\partial\zeta}, \tag{A.2d}$$

and

$$\frac{\partial()}{\partial t} = \frac{\partial()}{\partial\tau}\frac{\partial\tau}{\partial t} + \frac{\partial()}{\partial\xi}\frac{\partial\xi}{\partial t} + \frac{\partial()}{\partial\eta}\frac{\partial\eta}{\partial t} + \frac{\partial()}{\partial\zeta}\frac{\partial\zeta}{\partial t}, \tag{A.3a}$$

$$\frac{\partial()}{\partial x} = \frac{\partial()}{\partial\tau}\frac{\partial\tau}{\partial x} + \frac{\partial()}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial()}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial()}{\partial\zeta}\frac{\partial\zeta}{\partial x}, \tag{A.3b}$$

$$\frac{\partial()}{\partial y} = \frac{\partial()}{\partial\tau}\frac{\partial\tau}{\partial y} + \frac{\partial()}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial()}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial()}{\partial\zeta}\frac{\partial\zeta}{\partial y}, \tag{A.3c}$$

$$\frac{\partial()}{\partial z} = \frac{\partial()}{\partial\tau}\frac{\partial\tau}{\partial z} + \frac{\partial()}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial()}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial()}{\partial\zeta}\frac{\partial\zeta}{\partial z}. \tag{A.3d}$$

Writing Eqs. (A.2) and (A.3) in matrix form, one obtains:

$$\begin{bmatrix} \frac{\partial()}{\partial\tau} \\ \frac{\partial()}{\partial\xi} \\ \frac{\partial()}{\partial\eta} \\ \frac{\partial()}{\partial\zeta} \end{bmatrix} = \begin{bmatrix} t_\tau & x_\tau & y_\tau & z_\tau \\ t_\xi & x_\xi & y_\xi & z_\xi \\ t_\eta & x_\eta & y_\eta & z_\eta \\ t_\zeta & x_\zeta & y_\zeta & z_\zeta \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix}, \tag{A.4}$$

and

$$\begin{bmatrix} \frac{\partial()}{\partial t} \\ \frac{\partial()}{\partial x} \\ \frac{\partial()}{\partial y} \\ \frac{\partial()}{\partial z} \end{bmatrix} = \begin{bmatrix} \tau_t & \xi_t & \eta_t & \zeta_t \\ \tau_x & \xi_x & \eta_x & \zeta_x \\ \tau_y & \xi_y & \eta_y & \zeta_y \\ \tau_z & \xi_z & \eta_z & \zeta_z \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial\tau} \\ \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\zeta} \end{bmatrix}. \tag{A.5}$$

The matrix relationship (A.4) can be inverted to find Cartesian derivatives in terms of curvilinear ones, as follows:

$$\begin{bmatrix} \frac{\partial()}{\partial t} \\ \\ \\ \\ \frac{\partial()}{\partial x} \\ \frac{\partial()}{\partial y} \\ \frac{\partial()}{\partial z} \end{bmatrix} = \frac{1}{t_\tau J} \begin{bmatrix} & \{-x_\tau(y_\eta z_\zeta - z_\eta y_\zeta) & \{-x_\tau(z_\xi y_\zeta - y_\xi z_\zeta) & \{-x_\tau(y_\xi z_\eta - z_\xi y_\eta) \\ J & -y_\tau(z_\eta x_\zeta - x_\eta z_\zeta) & -y_\tau(x_\xi z_\zeta - z_\xi x_\zeta) & -y_\tau(z_\xi x_\eta - x_\xi z_\eta) \\ & -z_\tau(x_\eta y_\zeta - y_\eta x_\zeta)\} & -z_\tau(y_\xi x_\zeta - x_\xi y_\zeta)\} & -z_\tau(x_\xi y_\eta - y_\xi x_\eta)\} \\ 0 & t_\tau(y_\eta z_\zeta - z_\eta y_\zeta) & t_\tau(z_\xi y_\zeta - y_\xi z_\zeta) & t_\zeta(y_\xi z_\eta - z_\xi y_\eta) \\ 0 & t_\tau(z_\eta x_\zeta - x_\eta z_\zeta) & t_\tau(x_\xi z_\zeta - z_\xi x_\zeta) & t_\tau(z_\xi x_\eta - x_\xi z_\eta) \\ 0 & t_\tau(x_\eta y_\zeta - y_\eta x_\zeta) & t_\tau(y_\xi x_\zeta - x_\xi y_\zeta) & t_\tau(x_\xi y_\eta - y_\xi x_\eta) \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial\tau} \\ \\ \\ \\ \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\zeta} \end{bmatrix}, \tag{A.6}$$

where $J = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} = [x_\xi(y_\eta z_\zeta - z_\eta y_\zeta) + y_\xi(z_\eta x_\zeta - x_\eta z_\zeta) + z_\xi(x_\eta y_\zeta - y_\eta x_\zeta)]$ is the Jacobian of the inverse coordinate transformation.

If a comparison is made between the matrix elements of Eqs. (A.5) and (A.6), metrics relations are obtained, as follows:

$$\begin{aligned}
\tau_t &= \frac{1}{t_\tau} \\
\xi_t &= \frac{\tau_t}{J}\{-x_\tau(y_\eta z_\zeta - z_\eta y_\zeta) - y_\tau(z_\eta x_\zeta - x_\eta z_\zeta) - z_\tau(x_\eta y_\zeta - y_\eta x_\zeta)\}, \\
\eta_t &= \frac{\tau_t}{J}\{-x_\tau(z_\xi y_\zeta - y_\xi z_\zeta) - y_\tau(x_\xi z_\zeta - z_\xi x_\zeta) - z_\tau(y_\xi x_\zeta - x_\xi y_\zeta)\}, \\
\zeta_t &= \frac{\tau_t}{J}\{-x_\tau(y_\xi z_\eta - z_\xi y_\eta) - y_\tau(z_\xi x_\eta - x_\xi z_\eta) - z_\tau(x_\xi y_\eta - y_\xi x_\eta)\}, \\
\xi_x &= \frac{1}{J}(y_\eta z_\zeta - z_\eta y_\zeta) \;,\; \eta_x = \frac{1}{J}(z_\xi y_\zeta - y_\xi z_\zeta) \;,\; \zeta_x = \frac{1}{J}(y_\xi z_\eta - z_\xi y_\eta), \\
\xi_y &= \frac{1}{J}(z_\eta x_\zeta - x_\eta z_\zeta) \;,\; \eta_y = \frac{1}{J}(x_\xi z_\zeta - z_\xi x_\zeta) \;,\; \zeta_y = \frac{1}{J}(z_\xi x_\eta - x_\xi z_\eta), \\
\xi_z &= \frac{1}{J}(x_\eta y_\zeta - y_\eta x_\zeta) \;,\; \eta_z = \frac{1}{J}(y_\xi x_\zeta - x_\xi y_\zeta) \;,\; \zeta_z = \frac{1}{J}(x_\xi y_\eta - y_\xi x_\eta).
\end{aligned} \tag{A.7}$$

Note that time derivatives can be further simplified by using the space derivatives. The space-time derivatives follows:

$$\begin{aligned}
\xi_t &= \tau_t(-x_\tau\xi_x - y_\tau\xi_y - z_\tau\xi_z), \\
\eta_t &= \tau_t(-x_\tau\eta_x - y_\tau\eta_y - z_\tau\eta_z), \\
\zeta_t &= \tau_t(-x_\tau\zeta_x - y_\tau\zeta_y - z_\tau\zeta_z).
\end{aligned} \tag{A.8}$$

At this point, consider a generic governing equation can be represented strong conservation form, as follows:

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = 0. \tag{A.9}$$

The equation above can be written in curvilinear coordinates by using equations (A.3), as follows:

$$\begin{aligned}
&\left(\tau_t\frac{\partial q}{\partial \tau} + \xi_t\frac{\partial q}{\partial \eta} + \eta_t\frac{\partial q}{\partial \eta} + \zeta_t\frac{\partial q}{\partial \zeta}\right) + \left(\xi_x\frac{\partial f}{\partial \xi} + \eta_x\frac{\partial f}{\partial \eta} + \zeta_x\frac{\partial f}{\partial \zeta}\right) + \\
&\left(\xi_y\frac{\partial g}{\partial \xi} + \eta_y\frac{\partial g}{\partial \eta} + \zeta_y\frac{\partial g}{\partial \zeta}\right) + \left(\xi_z\frac{\partial h}{\partial \xi} + \eta_z\frac{\partial h}{\partial \eta} + \zeta_z\frac{\partial h}{\partial \zeta}\right) = 0.
\end{aligned} \tag{A.10}$$

Note that the above equation is no longer in a strong conservation law form, because coefficients do multiply derivatives.

Multiplying equation (A.10) by J and adding several zero terms(shown in brackets), one obtains:

$$
\begin{aligned}
&J\tau_t\frac{\partial q}{\partial \tau}+J\xi_t\frac{\partial q}{\partial \xi}+J\eta_t\frac{\partial q}{\partial \eta}+J\zeta_t\frac{\partial q}{\partial \zeta}+\left[q\frac{\partial J}{\partial \tau}-q\frac{\partial J}{\partial \tau}\right]+\left[q\frac{\partial J\xi_t}{\partial \xi}-q\frac{\partial J\xi_t}{\partial \xi}\right]+\\
&\left[q\frac{\partial J\eta_t}{\partial \eta}-q\frac{\partial J\eta_t}{\partial \eta}\right]+\left[q\frac{\partial J\zeta_t}{\partial \zeta}-q\frac{\partial J\zeta_t}{\partial \zeta}\right]+J\xi_x\frac{\partial f}{\partial \xi}+J\eta_x\frac{\partial f}{\partial \eta}+J\zeta_x\frac{\partial f}{\partial \zeta}+\\
&\left[f\frac{\partial J\xi_x}{\partial \xi}-f\frac{\partial J\xi_x}{\partial \xi}\right]+\left[f\frac{\partial J\eta_x}{\partial \eta}-f\frac{\partial J\eta_x}{\partial \eta}\right]+\left[f\frac{\partial J\zeta_x}{\partial \zeta}-f\frac{\partial J\zeta_x}{\partial \zeta}\right]+\\
&J\xi_y\frac{\partial g}{\partial \xi}+J\eta_y\frac{\partial g}{\partial \eta}+J\zeta_y\frac{\partial g}{\partial \zeta}+\left[g\frac{\partial J\xi_y}{\partial \xi}-g\frac{\partial J\xi_y}{\partial \xi}\right]+\left[g\frac{\partial J\eta_y}{\partial \eta}-g\frac{\partial J\eta_y}{\partial \eta}\right]+\\
&\left[g\frac{\partial J\zeta_y}{\partial \zeta}-g\frac{\partial J\zeta_y}{\partial \zeta}\right]+J\xi_z\frac{\partial h}{\partial \xi}+J\eta_z\frac{\partial h}{\partial \eta}+J\zeta_z\frac{\partial h}{\partial \zeta}+\left[h\frac{\partial J\xi_z}{\partial \xi}-h\frac{\partial J\xi_z}{\partial \xi}\right]+\\
&\left[h\frac{\partial J\eta_z}{\partial \eta}-h\frac{\partial J\eta_z}{\partial \eta}\right]+\left[h\frac{\partial J\zeta_z}{\partial \zeta}-h\frac{\partial J\zeta_z}{\partial \zeta}\right] \;=\; 0. \qquad \text{(A.11)}
\end{aligned}
$$

The above equation can be rearranged in the following form:

$$
\begin{aligned}
&\left(J\tau_t\frac{\partial q}{\partial \tau}+q\frac{\partial J}{\partial \tau}\right)+\left(J\xi_t\frac{\partial q}{\partial \xi}+q\frac{\partial J\xi_t}{\partial \xi}\right)+\left(J\eta_t\frac{\partial q}{\partial \eta}+q\frac{\partial J\eta_t}{\partial \eta}\right)+\\
&\left(J\zeta_t\frac{\partial q}{\partial \zeta}+q\frac{\partial J\zeta_t}{\partial \zeta}\right)+q\left[\frac{\partial J}{\partial \tau}+\frac{\partial J\xi_t}{\partial \xi}+\frac{\partial J\eta_t}{\partial \eta}+\frac{\partial J\zeta_t}{\partial \zeta}\right]+\left(J\xi_x\frac{\partial f}{\partial \xi}+f\frac{\partial J\xi_x}{\partial \xi}\right)+\\
&\left(J\eta_x\frac{\partial f}{\partial \eta}+f\frac{\partial J\eta_x}{\partial \eta}\right)+\left(J\zeta_x\frac{\partial f}{\partial \zeta}+f\frac{\partial J\zeta_x}{\partial \zeta}\right)-f\;\left[\frac{\partial J\xi_x}{\partial \xi}+\frac{\partial J\eta_x}{\partial \eta}+\frac{\partial J\zeta_x}{\partial \zeta}\right]+\\
&\left(J\xi_y\frac{\partial g}{\partial \xi}+g\frac{\partial J\xi_y}{\partial \xi}\right)+\left(J\eta_y\frac{\partial g}{\partial \eta}+g\frac{\partial J\eta_y}{\partial \eta}\right)+\left(J\zeta_y\frac{\partial g}{\partial \zeta}+g\frac{\partial J\zeta_y}{\partial \zeta}\right)-\\
&g\left[\frac{\partial J\xi_y}{\partial \xi}+\frac{\partial J\eta_y}{\partial \eta}+\frac{\partial J\zeta_y}{\partial \zeta}\right]+\left(J\xi_z\frac{\partial h}{\partial \xi}+h\frac{\partial J\xi_z}{\partial \xi}\right)+\left(J\eta_z\frac{\partial h}{\partial \eta}+h\frac{\partial J\eta_z}{\partial \eta}\right)+\\
&\left(J\zeta_z\frac{\partial h}{\partial \zeta}+h\frac{\partial J\zeta_z}{\partial \zeta}\right)-h\left[\frac{\partial J\xi_z}{\partial \xi}+\frac{\partial J\eta_z}{\partial \eta}\;+\;\frac{\partial J\zeta_z}{\partial \zeta}\right]\;=\;0. \qquad \text{(A.12)}
\end{aligned}
$$

The terms shown in brackets here are identically zero (this can be verified using the metrics relations in Eq. (A.7)). Furthermore, applying the product rule to the terms in

parenthesis and collecting them yields:

$$\frac{\partial\left(J\tau_t q\right)}{\partial\tau}+\frac{\partial\left(J\xi_t q+J\xi_x f+J\xi_y g+J\xi_z h\right)}{\partial\xi}+\frac{\partial\left(J\eta_t q+J\eta_x f+J\eta_y g+J\eta_z h\right)}{\partial\eta}+\frac{\partial\left(J\zeta_t q+J\zeta_x f+J\zeta_y g+J\zeta_z h\right)}{\partial\zeta}=0. \quad \text{(A.13)}$$

In order to write the above equation in a compact form, one can define:

$$\begin{aligned} Q &= Jq\tau_t, \\ F &= J(\xi_t q \ + \ \xi_x f \ + \ \xi_y g \ + \xi_z h), \\ G &= J(\eta_t q \ + \ \eta_x f \ + \ \eta_y g \ + \eta_z h), \\ H &= J(\zeta_t q \ + \ \zeta_x f \ + \ \zeta_y g \ + \zeta_z h). \end{aligned} \quad \text{(A.14)}$$

The transformed equation can then be written in the following vector form, where the strong conservation form has been restored:

$$\frac{\partial Q}{\partial\tau} \ + \ \frac{\partial F}{\partial\xi} \ + \ \frac{\partial G}{\partial\eta} \ + \ \frac{\partial H}{\partial\zeta} \ = \ 0. \quad \text{(A.15)}$$

The Euler equations can be transformed using the procedure outlined above, because they can be written in a strong conservation form consistent with Eq. (A.9). In order to find the transformed vector quantities for the Euler equations, the dependent variable and fluxes are rewritten in Cartesian coordinates, as follows:

$$q=\left\{\begin{array}{c}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{array}\right\}, \qquad f=\left\{\begin{array}{c}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uh_t\end{array}\right\}, \qquad g=\left\{\begin{array}{c}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ \rho vh_t\end{array}\right\}, \qquad h=\left\{\begin{array}{c}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ \rho wh_t\end{array}\right\}. \quad \text{(A.16)}$$

The use of the above vector quantities q, f, g, h results in the transformed corresponding vectors Q, F, G, H, as follows:

$$Q = \tau_t J \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{array} \right\}, \qquad K = J \left\{ \begin{array}{c} \rho\theta_k \\ \rho u\theta_k \ + \ k_x p \\ \rho v\theta_k \ + \ k_y p \\ \rho w\theta_k \ + \ k_z p \\ \rho h_t\theta_k \ - \ k_t E_c p \end{array} \right\}, \tag{A.17}$$

where

$$\begin{aligned} h_t &= e_t \ + \ E_c \frac{p}{\rho}, \\ \theta_k &= k_x u \ + \ k_y v \ + \ k_z w \ + \ k_t, \\ K &= F \ \text{ and } \ \theta_\xi \ = \ U \quad \text{for} \quad k \ = \ \xi, \\ K &= G \ \text{ and } \ \theta_\eta \ = \ V \quad \text{for} \quad k \ = \ \eta, \\ K &= H \ \text{ and } \ \theta_\zeta \ = \ W \quad \text{for} \quad k \ = \ \zeta. \end{aligned} \tag{A.18}$$

Note that $\tau_t = 1$ in most applications, because time is left unchanged in the transformed equations (A.15), i.e $\tau = t$.

APPENDIX B

FLUX JACOBIANS

The vector of primitive variables and the generic flux vector for the Euler equations can be written as follows:

$$q \;=\; J \left\{ \begin{array}{c} \rho \\ u \\ v \\ w \\ P \end{array} \right\} \;=\; \left\{ \begin{array}{c} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{array} \right\}, \qquad K \;=\; J \left\{ \begin{array}{c} \rho\theta_k \\ \rho u\theta_k \;+\; k_x p \\ \rho v\theta_k \;+\; k_y p \\ \rho w\theta_k \;+\; k_z p \\ \rho h_t\theta_k \;-\; k_t E_c p \end{array} \right\}. \tag{B.1}$$

The transformed flux vector K can be written in terms of the dependent variables q to construct the flux Jacobian matrix $\partial K/\partial q$. The contravariant velocity θ_k can be written in terms of the dependent variables, as follows:

$$\theta_k \;=\; k_x q_2 \;+\; k_y q_3 \;+\; k_z q_4 \;+\; k_t. \tag{B.2}$$

In a similar way, total specific energy and enthalpy definitions from Section 2.3 yield:

$$\begin{aligned} e_t &= \frac{q_5}{q_1}\,\frac{E_c}{\gamma-1} \;+\; E_c\left(\frac{q_2^2 \;+\; q_3^2 \;+\; q_4^2}{2}\right), \\ h_t &= \frac{q_5}{q_1}\,\frac{E_c}{\gamma-1} \;+\; E_c\left(\frac{q_2^2 \;+\; q_3^2 \;+\; q_4^2}{2} \;+\; \frac{q_5}{q_1}\right). \end{aligned} \tag{B.3}$$

At this point, the matrix can be found easily. If we define the flux Jacobian matrix $S \;=\; \partial K/\partial Q$ (using conserved variables), then the two matrices are related, as follows:

$$\frac{\partial K}{\partial q} \;=\; \frac{\partial K}{\partial Q}\frac{\partial Q}{\partial q} \;=\; S\;M, \tag{B.4}$$

where M is transformation matrix given in section (2.5)

The system matrix s is related the Jacobian matrix above as follows:

$$s \;=\; M^{-1}\frac{\partial K}{\partial q} = M^{-1}SM. \tag{B.5}$$

After some algebra, one obtains:

$$s = \begin{bmatrix} \theta_k & k_x\ \rho & k_y\ \rho & k_z\ \rho & 0 \\ 0 & \theta_k & 0 & 0 & \frac{k_x}{\rho} \\ 0 & 0 & \theta_k & 0 & \frac{k_y}{\rho} \\ 0 & 0 & 0 & \theta_k & \frac{k_z}{\rho} \\ 0 & k_x\ \gamma\ p & k_y\ \gamma\ p & k_z\ \gamma p & \theta_k \end{bmatrix}. \tag{B.6}$$

By using the relation $p = \frac{\rho\ c^2}{\gamma}$, the matrix s gets the following form.

$$s = \begin{bmatrix} \theta_k & k_x\ \rho & k_y\ \rho & k_z\ \rho & 0 \\ 0 & \theta_k & 0 & 0 & \frac{k_x}{\rho} \\ 0 & 0 & \theta_k & 0 & \frac{k_y}{\rho} \\ 0 & 0 & 0 & \theta_k & \frac{k_z}{\rho} \\ 0 & k_x\ \rho\ c^2 & k_y\ \rho\ c^2 & k_z\ \rho\ c^2 & \theta_k \end{bmatrix}. \tag{B.7}$$

Multiplying the matrix s by the preconditioning matrix Γ_q gives the preconditioned system matrix s_Γ, which is necessary for the preconditioned formulation:

$$s_\Gamma = \Gamma_q s = \begin{bmatrix} \theta_k & k_x\ \rho & k_y\ \rho & k_z\ \rho & 0 \\ 0 & \theta_k & 0 & 0 & \frac{k_x}{\rho} \\ 0 & 0 & \theta_k & 0 & \frac{k_y}{\rho} \\ 0 & 0 & 0 & \theta_k & \frac{k_z}{\rho} \\ 0 & \beta\ k_x\ \rho\ c^2 & \beta\ k_y\ \rho\ c^2 & \beta\ k_z\ \rho\ c^2 & \beta\ \theta_k \end{bmatrix}, \tag{B.8}$$

where

$$\begin{aligned} s_\Gamma &= a_\Gamma \quad for \quad k = \xi, \\ s_\Gamma &= b_\Gamma \quad for \quad k = \eta, \\ s_\Gamma &= c_\Gamma \quad for \quad k = \zeta. \end{aligned} \tag{B.9}$$

APPENDIX C

EIGENVALUES AND EIGENVECTORS

Let $\mathbf{A}$ be a $n \times n$ matrix. A number λ is said to be an eigenvalue of $\mathbf{A}$ if the there exists a nonzero solution vector K of the linear system:

$$\mathbf{A}\ K\ =\ \lambda\ K. \tag{C.1}$$

The solution vector K is said to be a right eigenvector corresponding to the eigenvalue λ. Equation (C.1) can be rearranged as

$$(\ \mathbf{A}\ -\ \lambda\ I\ )\ K\ =\ 0, \tag{C.2}$$

where I is the identity matrix.

In here, the trivial solution is $\{K\} = 0$. Nontrivial solutions exist if and only if the determinant of the coefficient matrix is equal to zero:

$$det\ (\mathbf{A}\ -\ \lambda\ I) =\ 0. \tag{C.3}$$

Eq. (C.3) leads to an n^{th} degree polynomial in λ . This polynomial is called the characteristic equation of $\mathbf{A}$, and solutions of the polynomial are the eigenvalues of $\mathbf{A}$. In order to find the eigenvectors corresponding to an eigenvalue λ, equation (C.2) is to be solved. Note that eigenvectors are not unique: for example, if K_1 is an eigenvector, then αK_1 is also an eigenvector, where α is an arbitrary value.

The eigenvalues and eigenvectors of the system matrix $s_\Gamma = \Gamma_q s$ are going to be found next. For convenience, the matrix s_Γ is rewritten here:

$$s_\Gamma = \Gamma_q s\ =\ \begin{bmatrix} \theta_k & k_x\ \rho & k_y\ \rho & k_z\ \rho & 0 \\ 0 & \theta_k & 0 & 0 & \frac{k_x}{\rho} \\ 0 & 0 & \theta_k & 0 & \frac{k_y}{\rho} \\ 0 & 0 & 0 & \theta_k & \frac{k_z}{\rho} \\ 0 & \beta\ k_x\ \rho\ c^2 & \beta\ k_y\ \rho\ c^2 & \beta\ k_z\ \rho\ c^2 & \beta\ \theta_k \end{bmatrix}. \tag{C.4}$$

After some algebra, the eigenvalues of the matrix s_Γ are obtained as follows

$$\lambda_1 \;=\; \theta_k, \quad \lambda_2 \;=\; \theta_k, \quad \lambda_3 \;=\; \theta_k, \quad \theta_k\;\beta^+ \;+\; \sigma, \quad \theta_k\;\beta^+ \;-\; \sigma, \tag{C.5}$$

where $\sigma \;=\; \Big[(\theta_k\;\beta^-)^2 \;+\; \beta\;c^2\;(k_x^2 \;+\; k_y^2 \;+\; k_z^2)\Big]^{1/2}$ with $\beta^\pm \;=\; \frac{1\;\pm\;\beta}{2}$.

As mentioned before, the eigenvectors are not unique. A good starting is to seek eigenvectors which contain as many zero elements as possible, in order to reduce arithmetic operations when evaluating fluxes. One of the simplest possible set of eigenvectors (a matrix showing the right eigenvectors as columns) was found by using a symbolic mathematic manipulator (*Mathematica*), and reads

$$R_q \;=\; \begin{bmatrix} 0 & 0 & 1 & \frac{\rho}{\hat\sigma-\hat\theta_k\beta^-} & \frac{\rho}{\hat\sigma+\hat\theta_k\beta^-} \\ -\hat k_z & -\hat k_y & 0 & \hat k_x & -\hat k_x \\ 0 & \hat k_x & 0 & \hat k_y & -\hat k_y \\ \hat k_x & 0 & 0 & \hat k_x & -\hat k_x \\ 0 & 0 & 0 & \rho(\hat\sigma-\hat\theta_k\beta^-) & \rho(\hat\sigma+\hat\theta_k\beta^-) \end{bmatrix}, \tag{C.6}$$

where $\hat k_x \;=\; \frac{k_x}{|\nabla k|}, \quad \hat k_y \;=\; \frac{k_y}{|\nabla k|}, \quad \hat k_z \;=\; \frac{k_z}{|\nabla k|}, \quad \hat k_t \;=\; \frac{k_t}{|\nabla k|}$ with

$$\begin{aligned} |\nabla k| \;&=\; \Big(k_x^2 \;+k_y^2 \;+\; k_z^2\Big)^{\frac12}, \\ \hat\theta_k \;&=\; u\;\hat k_x \;+\; v\;\hat k_y \;+\; w\;\hat k_z \;+\; \hat k_t, \\ \hat\sigma \;&=\; \Big[(\hat\theta_k\;\beta^-)^2 \;+\; \beta\;c^2\Big]^{\frac12}. \end{aligned} \tag{C.7}$$

In the above, $\hat k_*$ are the *normalized* metric coefficients.

The choice of eigenvectors shown before has some severe drawbacks: specifically, the normalized metric coefficients can be easily zero because they correspond to the component of unit vectors normal to cell faces. It is possible that any 2 out of 3 of these normalized metrics coefficients be zero. However, a linear combination of the first three eigenvectors

may be used to overcome the 'worst case' scenario:

$$R_q^{(1,2,3)} = \begin{Bmatrix} R_{q_{(1)}}^{(1,2,3)} \\ R_{q_{(2)}}^{(1,2,3)} \\ R_{q_{(3)}}^{(1,2,3)} \\ R_{q_{(4)}}^{(1,2,3)} \\ R_{q_{(5)}}^{(1,2,3)} \end{Bmatrix} = c_1 \begin{Bmatrix} 0 \\ -\hat{k}_z \\ 0 \\ \hat{k}_x \\ 0 \end{Bmatrix} + c_2 \begin{Bmatrix} 0 \\ -\hat{k}_y \\ \hat{k}_x \\ 0 \\ 0 \end{Bmatrix} + c_3 \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}. \tag{C.8}$$

The constants $c_1,\ c_2,\ and\ c_3$ are arbitrary values. A choice of these constants gives rise to three eigenvectors, as follows

$$\begin{aligned} {R_q}^1 &= \{\hat{k}_x\ ,\ 0\ ,\ \hat{k}_z\ ,\ -\hat{k}_y\ ,\ 0\}^T \quad \text{for} \quad c_1 = -\frac{\hat{k}_y}{\hat{k}_x}\ ,\ c_2 = \frac{\hat{k}_z}{\hat{k}_x}\ ,\ c_3 = \hat{k}_x, \\ {R_q}^2 &= \{\hat{k}_y\ ,\ -\hat{k}_z\ ,\ 0\ ,\ \hat{k}_x\ ,\ 0\}^T \quad \text{for} \quad c_1 = 1\ ,\ c_2 = 0\ ,\ c_3 = \hat{k}_y, \\ {R_q}^3 &= \{\hat{k}_z\ ,\ \hat{k}_y\ ,\ -\hat{k}_x\ ,\ 0\ ,\ 0\}^T \quad \text{for} \quad c_1 = 0\ ,\ c_2 = -1\ ,\ c_3 = \hat{k}_z. \end{aligned} \tag{C.9}$$

At this point, using the following identity:

$$(\hat{\sigma}\ -\ \hat{\theta_k}\ \beta^-)\ (\hat{\sigma}\ +\ \hat{\theta_k}\ \beta^-)\ =\ \beta\ c^2, \tag{C.10}$$

one can derive a well-behaved version of the eigenvector matrix, as follows:

$$R_q = \begin{bmatrix} \hat{k}_x & \hat{k}_y & \hat{k}_z & \frac{\rho}{\beta c^2}(\hat{\sigma} + \hat{\theta}_k\beta^-) & \frac{\rho}{\beta c^2}(\hat{\sigma} - \hat{\theta}_k\beta^-) \\ 0 & -\hat{k}_z & \hat{k}_y & \hat{k}_x & -\hat{k}_x \\ \hat{k}_z & 0 & -\hat{k}_x & \hat{k}_y & -\hat{k}_y \\ -\hat{k}_y & \hat{k}_x & 0 & \hat{k}_x & -\hat{k}_x \\ 0 & 0 & 0 & \rho(\hat{\sigma} - \hat{\theta}_k\beta^-) & \rho(\hat{\sigma} + \hat{\theta}_k\beta^-) \end{bmatrix}. \tag{C.11}$$

This nonsingular set of eigenvectors was proposed by Briley, Taylor and Whitfield [7]. It is seen that the determinant of the eigenvector matrix is $|R_q|\ =\ 2\ \rho\ \hat{\sigma}$, which is always nonzero.

The inverse of eigenvector matrix reads:

$$R_q^{-1} = \begin{bmatrix} \hat{k}_x & -\frac{2\rho\hat{k}_x^2\hat{\theta}_k\beta^-}{c^2\beta} & \hat{k}_z - \frac{2\rho\hat{k}_x\hat{k}_y\hat{\theta}_k\beta^-}{c^2\beta} & -\hat{k}_y - \frac{2\rho\hat{k}_x\hat{k}_z\hat{\theta}_k\beta^-}{c^2\beta} & -\frac{\hat{k}_x}{c^2\beta} \\ \hat{k}_y & -\hat{k}_z - \frac{2\rho\hat{k}_x\hat{k}_y\hat{\theta}_k\beta^-}{c^2\beta} & -\frac{2\rho\hat{k}_y^2\hat{\theta}_k\beta^-}{c^2\beta} & \hat{k}_x - \frac{2\rho\hat{k}_y\hat{k}_z\hat{\theta}_k\beta^-}{c^2\beta} & -\frac{\hat{k}_y}{c^2\beta} \\ \hat{k}_z & \hat{k}_y - \frac{2\rho\hat{k}_x\hat{k}_z\hat{\theta}_k\beta^-}{c^2\beta} & -\hat{k}_x - \frac{2\rho\hat{k}_y\hat{k}_z\hat{\theta}_k\beta^-}{c^2\beta} & -\frac{2\rho\hat{k}_z^2\hat{\theta}_k\beta^-}{c^2\beta} & -\frac{\hat{k}_z}{c^2\beta} \\ 0 & \frac{\hat{k}_x(\hat{\sigma}+\hat{\theta}_k\beta^-)}{2\hat{\sigma}} & \frac{\hat{k}_y(\hat{\sigma}+\hat{\theta}_k\beta^-)}{2\hat{\sigma}} & \frac{\hat{k}_z(\hat{\sigma}+\hat{\theta}_k\beta^-)}{2\hat{\sigma}} & \frac{1}{2\rho\hat{\sigma}} \\ 0 & -\frac{\hat{k}_x(\hat{\sigma}-\hat{\theta}_k\beta^-)}{2\hat{\sigma}} & -\frac{\hat{k}_y(\hat{\sigma}-\hat{\theta}_k\beta^-)}{2\hat{\sigma}} & -\frac{\hat{k}_z(\hat{\sigma}-\hat{\theta}_k\beta^-)}{2\hat{\sigma}} & -\frac{1}{2\rho\hat{\sigma}} \end{bmatrix}. \quad \text{(C.12)}$$

C.1 Eigenvalues and Eigenvectors for Two Dimensional System

A non-singular set of eigenvectors is proposed for two dimensional system as

$$R_q = \begin{bmatrix} 0 & 1 & \frac{\rho}{\beta\ c^2}\ (\hat{\sigma}\ +\ \hat{\theta}_k\ \beta^-) & \frac{\rho}{\beta\ c^2}\ (\hat{\sigma}\ -\ \hat{\theta}_k\ \beta^-) \\ -\hat{k}_y & 0 & \hat{k}_x & -\hat{k}_x \\ \hat{k}_x & 0 & \hat{k}_y & -\hat{k}_y \\ 0 & 0 & \rho\ (\hat{\sigma}\ -\ \hat{\theta}_k\ \beta^-) & \rho\ (\hat{\sigma}\ +\ \hat{\theta}_k\ \beta^-) \end{bmatrix}. \quad \text{(C.13)}$$

The inverse of this eigenvector matrix is as follows

$$R_q^{-1} = \begin{bmatrix} 0 & -\hat{k}_y & \hat{k}_x & 0 \\ 1 & -\frac{2\ \rho\ \hat{k}_x\ \hat{\theta}_k\ \beta^-}{c^2\ \beta} & -\frac{2\ \rho\ \hat{k}_y\ \hat{\theta}_k\ \beta^-}{c^2\ \beta} & -\frac{1}{c^2\ \beta} \\ 0 & \frac{\hat{k}_x\ (\hat{\sigma}\ +\ \hat{\theta}_k\ \beta^-)}{2\ \hat{\sigma}} & \frac{\hat{k}_y\ (\hat{\sigma}\ +\ \hat{\theta}_k\ \beta^-)}{2\ \hat{\sigma}} & \frac{1}{2\ \rho\ \hat{\sigma}} \\ 0 & -\frac{\hat{k}_x\ (\hat{\sigma}\ -\ \hat{\theta}_k\ \beta^-)}{2\ \hat{\sigma}} & -\frac{\hat{k}_y\ (\hat{\sigma}\ -\ \hat{\theta}_k\ \beta^-)}{2\ \hat{\sigma}} & \frac{1}{2\ \rho\ \hat{\sigma}} \end{bmatrix}. \quad \text{(C.14)}$$

Note that only one normalized metric coefficient can be zero in two dimensions. This eigenvector matrix for two-dimensional systems is also well-behaved because its determinant is $|R_q| = 2\rho\hat{\sigma}$, which is always nonzero.