To the Graduate Council:

I am submitting herewith a dissertation written by Robert Tramel entitled "A Nonlinear Filter for Inviscid Flow Equations." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science with a major in Mathematics.

K C Reddy

K. C. Reddy, Major Professor

We have read this dissertation and recommend its acceptance:

Accepted for the Council:

Cwminkel

Vice Provost
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Robert W. Granel

Date 7-19-90

A NONLINEAR FILTER FOR INVISCID FLUID FLOW EQUATIONS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Robert Wallace Tramel

August 1990

ACKNOWLEDGEMENTS

The support of part of this research by the Air Force Systems Command and the Air Force Office of Scientific Research is greatly appreciated. I would like to thank Dr. K. C. Reddy and Dr. Jack Benek for advice and thoughtful discussions throughout the course of this research. Dr. William Thomson is due many thanks for his advice on many matters relating to the computer systems used in the course of this research. Linda Williams' help in the preperation of this manuscript is also gratefully acknowledged.

ABSTRACT

A nonlinear filter based on artificial dissipation is tested and shown to produce a nonoscillatory shock capturing scheme for hyperbolic systems of conservation laws. In order to control spurious oscillations, the filter is applied after every time step to the solution from a locally implicit scheme for the Euler equation. The method is tested on both unsteady and steady state flow problems in order to demonstrate the efficiency and the efficacy of the technique. On steady state flow problems, an odd/even or red/black ordering of the grid points is considered. This ordering is shown to be robust and also well-suited for implementation on vector computors.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In recent years the methods of computational fluid dynamics (CFD) have been used sucessfully for solving a wide variety of problems of interest to both scientists and engineers. However, the scope and magnitude of the problems that are under consideration for solution by CFD grow constantly. Thus, there is a very important need for more efficient methods for solving large scale problems.These algorithms must be not only efficient but they must also be both robust (i.e. perform well under a wide variety of fluid flow conditions without the need for retuning) and well-suited to the current computer architectures such as vector processors if they are to be implemented effectively.

Attempts to meet these goals have been focused on developing shock capturing schemes for hyperbolic systems of conservation laws. Hyperbolic systems of partial differential equations have warrented this attention because they exhibit many of the features one finds in real fluid flows as well as possesing a well-defined mathematical theory. See Lax (1972).

A major complication of any attempt to numerically solve hyperbolic systems is the fact that the solution may contain discontinuities even if the initial data for the problem is smooth. Early efforts by many researchers to numerically solve hyperbolic systems revealed that second order accurate centered difference schemes produce spurious oscillations in the vicinity of discontinuities. Further research revealed that the unwanted and nonphysical oscillations could be supressed if the methods are supplemented with a well-chosen blend of second and fourth order artificial dissipation terms. However, the cost of the computation of these terms is more than the cost of the corresponding conservation quantities and fluxes. If the equations are solved by an iterative technique, then the costly dissipation terms must be computed many times per time step.

On the other hand, Engquist, Lötstedt, and Sjögreen (1989) have shown recently that the oscillations associated with central difference schemes can be checked by applying a nonlinear filter to the difference scheme at each time step. This filter searches for unwanted maxima and minima in the flow field. These unwanted extrema are then eliminated by removing conserved densities from the cells in which the unwanted extrema are located and adding them to the neighboring cells in such a manner that conserved densities are neither created nor destroyed by this process.

The main objective of this thesis is to develop a nonlinear filter that is conservative like those proposed by Engquist, Lötstedt, and Sjögreen but based on adaptive artificial dissipation techniques as developed by Baldwin and McCormack (1975), Jameson (1985A), (1985B) and modified by Pulliam (1985) and others. This filter should control oscillations in all parts of the flow field on both steady and unsteady test problems. The filter is tested on a scheme for the solution of hyperbolic systems of conservation laws developed by Reddy and Jacocks (1987) known as the Locally Implicit Method(***LIM***). The filter reduces the computational cost of the artificial dissipation terms substantially when it is used with an iterative technique such as the ***LIM***.

It has been found that Reddy and Jacocks' scheme works well with a modified Gauss–Seidel iteration, but numerical evidence and Fourier stability analysis of model scalar equations indicate that when a modified Jacobi iteration is applied, the method tends to amplify medium wavelength disturbances. Unfortunately, a Gauss–Seidel-type iteration is a recursive process, and a recursive process is not suitable for efficient implementation on vector computers. Similar problems occur when solving elliptic partial differential equations such as diffusion equations in that the convergence rate of the Jacobi method is much slower than the convergence rate of the Gauss–Seidel Method. This problem was resolved when Erikson (1972), Hayes (1974), and Lambiotte (1974) demonstrated that a odd/even or red/black ordering of the grid points would allow Gauss–Seidel iteration to be carried out in a Jacobi-like manner. See Ortega and Voigt (1982) for a review as well as a general discussion of the solution of partial differential equations on vector computers. The second objective of this thesis is to develop and test an odd/even ordering of the grid points for hyperbolic equations. This ordering will remove the instability inherent with the modified Jacobi method while maintaining the desirable feature of compatibility with vector computers.

In order to demonstrate the efficacy of the methods described above, they are tested on three standard test problems. The first test case considered is the shock tube problem recommended by Sod (1978) as a standard test of finite difference schemes for time dependent hyperbolic systems of conservation laws. The problem is solved by the modified Gauss–Seidel method with both an artificial dissipation filter as well as Jameson-type artificial dissipation and the results are compared. The second test case is the steady state nozzle flow problem of Shubin, Glaz, and Stephen. The modified Odd/Even Jacobi Method with artificial dissipation filter is used to solve the problem and the result is compared to the solution computed by the Beam and Warming (1976) implicit scheme. For the third and final test case, we apply a combination of the modified Gauss–Seidel and Odd/Even Jacobi methods plus artificial dissipation filter to the two-dimensional shock reflection problem considered by Harten, Yee, and Warming (1983). The results are compared to the solutions computed by a Total Variation Diminishing(TVD) scheme as well as the Beam and Warming implicit method.

CHAPTER 2

LOCALLY IMPLICIT METHOD WITH A FILTER

2.1 1–D Euler Equation

The Euler equation for inviscid flows is an example of a hyperbolic system of conservation laws. We focus on the Euler equation because it is general enough to illustrate all of the features of the ***LIM*** as well as the fact that all of the test cases considered in this paper involve the solution of the Euler equation. The Euler equation in one dimension can be written in the form

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = 0, \tag{1}$$

where $\boldsymbol{Q}$ is the vector $(\rho, \rho u, e)$ of conserved variables and $\boldsymbol{F}$ is the flux vector $(\rho u, \rho u^2 + P, u(e+P))$. Here ρ is the density, u the velocity, e the energy per unit volume, and P the pressure. The pressure is related to the other variables for a perfect gas by the relation $P = (\gamma - 1)\left(e - \frac{1}{2}\rho u^2\right)$.

The Euler implicit scheme for Eq. (1) with central difference spatial approximations is written at each grid point j as follows

$$\frac{1}{\Delta t}\left(\boldsymbol{Q}_j^{n+1} - \boldsymbol{Q}_j^n\right) + \frac{1}{2\Delta x}\left(\boldsymbol{F}_{j+1}^{n+1} - \boldsymbol{F}_{j-1}^{n+1}\right) = 0. \tag{2}$$

$\boldsymbol{Q}_j^{n+1}$ is the value of the vector $\boldsymbol{Q}$ at the jth lattice point at time step $n+1$, while $\boldsymbol{F}_j^{n+1}$ is the value of the vector $\boldsymbol{F}$ evaluated using $\boldsymbol{Q}_j^{n+1}$. Δt is the lattice spacing in the time direction, and Δx is the lattice spacing in the spatial direction. From time step n to time step $n+1$, the system of Eqs. (2) is solved by the Locally Implicit Method developed by Reddy and Jacocks (1987).

2.2 Modified Gauss–Seidel Iteration

Define

$$\Delta \boldsymbol{Q}_j = \boldsymbol{Q}_j^{n+1} - \boldsymbol{Q}_j^n .$$

Within the truncation error of the scheme, the $\boldsymbol{F}_{j+1}^{n+1}$ term may be approximated to be

$$\boldsymbol{F}_{j+1}^{n+1} \simeq \boldsymbol{F}_{j+1}^{n} + A_{j+1}\Delta \boldsymbol{Q}_{j+1}$$

where $A_{j+1} = \left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}\right)_{j+1}$ is the flux Jacobian evaluated using $\boldsymbol{Q}_{j+1}^n$.

With this notation and approximation, Eq. (2) becomes

$$\frac{\Delta \boldsymbol{Q}_j}{\Delta t} + \frac{1}{2\Delta x}\left(A_{j+1}\Delta \boldsymbol{Q}_{j+1} - A_{j-1}\Delta \boldsymbol{Q}_{j-1}\right) = \boldsymbol{Res}_j^n \tag{3}$$

where

$$\boldsymbol{Res}_j^n = -\frac{1}{2\Delta x}\left(\boldsymbol{F}_{j+1}^n - \boldsymbol{F}_{j-1}^n\right).$$

Next, define

$$\Delta \boldsymbol{Q}_j^{(m)} = \boldsymbol{Q}_j^{(m)} - \boldsymbol{Q}_j^n$$

to be the mth iterative approximation to $\Delta \boldsymbol{Q}_j$ where $\boldsymbol{Q}_j^{(m)}$ is the mth iterative approximation to $\boldsymbol{Q}_j^{n+1}$. Note that at the start of each iteration, $\Delta \boldsymbol{Q}_j^{(0)}$ is set equal to $\boldsymbol{0}$. Also, define

$$d\boldsymbol{Q}_j^{(m)} = \Delta \boldsymbol{Q}_j^{(m+1)} - \Delta \boldsymbol{Q}_j^{(m)} = \boldsymbol{Q}_j^{(m+1)} - \boldsymbol{Q}_j^{(m)}$$

to be the correction to $\Delta \boldsymbol{Q}_j^{(m)}$ computed during the mth sweep.

Equation (3) now becomes

$$\frac{d\boldsymbol{Q}_j^{(m)}}{\Delta t} + \frac{1}{2\Delta x}\left(A_{j+1} d\boldsymbol{Q}_{j+1}^{(m)} - A_{j-1} d\boldsymbol{Q}_{j-1}^{(m)}\right) = \boldsymbol{Rhs}_j^{(m)} \tag{4}$$

where the right hand side of Equation (4) , $\boldsymbol{Rhs}_j^{(m)}$, is defined to be

$$\boldsymbol{Rhs}_j^{(m)} = \boldsymbol{Res}_j^n - \frac{1}{2\Delta x}\left(A_{j+1}\Delta\boldsymbol{Q}_{j+1}^{(m)} - A_{j-1}\Delta\boldsymbol{Q}_{j-1}^{(m)}\right) - \frac{\Delta\boldsymbol{Q}_j^{(m)}}{\Delta t}.$$

If one were to solve Eq. (4) by implementing a Gauss–Seidel iteration in a sequence of left to right grid sweeps then $d\boldsymbol{Q}_{j-1}^{(m)}$ would already be computed at grid point j and $d\boldsymbol{Q}_{j+1}^{(m)}$ would be set equal to $\mathbf{0}$. This iteration, however, is neutrally stable for small time steps, but it becomes unstable in practice for large time steps. Instead, we implement a modified Gauss–Seidel scheme in which we approximate the $d\boldsymbol{Q}_{j+1}^{(m)}$ term by $d\boldsymbol{Q}_{j+1}^{(m)} \simeq d\boldsymbol{Q}_j^{(m)}$. In this case Eq. (4) becomes

$$\left(\frac{I}{\Delta t} + \frac{A_{j+1}}{2\Delta x}\right) d\boldsymbol{Q}_j^{(m)} = \boldsymbol{Rhs}_j^{(m)} \tag{5}$$

where $\boldsymbol{Rhs}_j^{(m)}$ is now defined to be

$$\boldsymbol{Rhs}_j^{(m)} = \boldsymbol{Res}_j^n - \frac{1}{2\Delta x}\left(A_{j+1}\Delta\boldsymbol{Q}_{j+1}^{(m)} - A_{j-1}\Delta\boldsymbol{Q}_{j-1}^{(m+1)}\right) - \frac{\Delta\boldsymbol{Q}_j^{(m)}}{\Delta t}$$

and I is the unit matrix. In addition, we make the approximation

$$\boldsymbol{F}_{j+1}^n + A_{j+1}\Delta\boldsymbol{Q}_{j+1}^{(m)} \simeq \boldsymbol{F}_{j+1}^{(m)}$$

where $\boldsymbol{F}_{j+1}^{(m)}$ is the vector $\boldsymbol{F}$ evaluated using $\boldsymbol{Q}_{j+1}^{(m)}$, etc. and, on the left-hand side of Eq. (5), the flux Jacobian matrix A_{j+1} is approximated by the spectral radius of the matrix A_j, (i.e. $A_{j+1} \simeq \lambda_j I$ where $\lambda_j = \left(\left|u_j^n\right| + a_j^n\right)$). Here $\left|u_j^n\right|$ is the absolute value of the velocity and a_j^n is the local speed of sound at grid point j.

With these approximations Eq. (5) is transformed into

$$b_j d\boldsymbol{Q}_j^{(m)} = \boldsymbol{Rhs}_j^{(m)} \tag{6}$$

or

$$Q_j^{(m+1)} = Q_j^{(m)} + \frac{Rhs_j^{(m)}}{b_j}$$

where

$$b_j = \frac{1}{\Delta t} + \frac{\lambda_j}{2\Delta x}$$

and

$$Rhs_j^{(m)} = -\left[\frac{1}{\Delta t}\left(Q_j^{(m)} - Q_j^n\right) + \frac{1}{2\Delta x}\left(F_{j+1}^{(m)} - F_{j-1}^{(m+1)}\right)\right]. \tag{7}$$

It has been found that if one implements the modified Gauss–Seidel method in a series of left to right grid sweeps then the method is unstable for problems with strong convection. This can be overcome by implementing the method in a process known as symmetric sweeping. A symmetric sweep is defined to be a left to right sweep followed by a right to left sweep.

Following the reasoning behind the derivation of Eqs. (6) and (7) one can show that for a right to left sweep, Eq. (6) remains unchanged while Eq. (7) becomes

$$Rhs_j^{(m)} = -\left[\frac{1}{\Delta t}\left(Q_j^{(m)} - Q_j^n\right) + \frac{1}{2\Delta x}\left(F_{j+1}^{(m+1)} - F_{j-1}^{(m)}\right)\right]. \tag{8}$$

For steady state problems, one usually performs one to two symmetric sweeps per time step before setting the value of $Q_j^{n+1} = Q_j^{(M)}$ where M is the number of sweeps carried out. For unsteady flow problems, where time accuracy is necesary, several symmetric sweeps are carried out in order to insure that the L_2 norm of $dQ^{(m)}$ becomes sufficiently small before Q_j^{n+1} is updated. Computational experence shows that three to four symmetric sweeps per time step are sufficient to acomplish this task.

2.3 Stability Analysis of 1–D Scalar Equation

In order to gain some insight into the nature of the modified Gauss–Seidel scheme, we analyze the method for a scalar hyperbolic equation in one dimension. A von Neumann or Fourier stability analysis is performed in order to estimate the local amplification of the numerical solution from one time step to the next. The model equation is written in the form

$$\frac{\partial u}{\partial t} + d\frac{\partial u}{\partial x} = 0 \tag{9}$$

where u is the conserved scalar density and d is the wave speed. Following the lines of the derivation in section 2.2, define

$$\Delta u_j^{(m)} = u_j^{(m)} - u_j^n$$

and

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

We make the same approximations that are used in the previous section to arrive at the following equation

$$bdu_j^{(m)} = rhs_j^{(m)} \tag{10}$$

where

$$b = 1 + \frac{|c|}{2}$$

and

$$c = \left(\frac{d\Delta t}{\Delta x}\right).$$

In the definition of b, the absolute value of c is used in order to accomodate advection in both directions. The right-hand side of Eq. (10) ,$rhs_j^{(m)}$, is defined to be

$$rhs_j^{(m)} = res_j^n - \left(\Delta u_j^{(m)} + \frac{c}{2}\left(\Delta u_{j+1}^{(m)} - \Delta u_{j-1}^{(m+1)}\right)\right)$$

for a left to right sweep and

$$rhs_j^{(m)} = res_j^n - \left(\Delta u_j^{(m)} + \frac{c}{2}\left(\Delta u_{j+1}^{(m+1)} - \Delta u_{j-1}^{(m)}\right)\right)$$

for a right to left sweep where

$$res_j^n = -\frac{c}{2}\left(u_{j+1}^n - u_{j-1}^n\right).$$

The Fourier stability analysis examines the amplification of a single Fourier mode from one time step to the next. In order to do this, we seek solutions to Eq. (10) of the form

$$\begin{aligned} u_j^n &= V^n\ e^{i\alpha j\Delta x} = V^n\ e^{ij\zeta}, \quad \zeta = \alpha\Delta x, \quad 0 \le \zeta \le \pi \\ \Delta u_j^{(m)} &= \Delta V^{(m)}\ e^{ij\zeta} \\ du_j^{(m)} &= \left(\Delta V^{(m+1)} - \Delta V^{(m)}\right)\ e^{ij\zeta} \end{aligned}$$

where $i = \sqrt{-1}$ and $\Delta u_j^{(0)}$ is set equal to 0 at every grid point. Substituting these terms into Eq. (10) with $m = 0$ and canceling the common factor $e^{ij\zeta}$ we obtain for a left to right sweep

$$\Delta V^{(1)} = \frac{r}{h_1} V^n$$

where

$$r = -ic\ \sin(\zeta) \quad \text{and} \quad h_1 = b - \left(\frac{c}{2}\right)\ e^{-i\zeta}.$$

Similarly for the $m = 1$ right to left sweep one obtains

$$\Delta V^{(2)} = V^n \left(\frac{r}{h_3}\frac{h_1 + h_2}{h_1}\right)$$

where r and h_1 are as defined before and

$$h_2 = b - 1 + \left(\frac{c}{2}\right)\ e^{-i\zeta} \quad \text{and} \quad h_3 = b + \left(\frac{c}{2}\right)\ e^{i\zeta}.$$

If the iteration is halted at two sweeps (i.e., one symmetric sweep) then we set

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

or

$$V^{n+1} = V^n \left(1 + r\left(\frac{h_1 + h_2}{h_1 \cdot h_3}\right)\right). \tag{11}$$

A necessary and sufficient condition for localized linear stability is that

$$G(\zeta) = \left|\frac{V^{n+1}}{V^n}\right| \leq 1.$$

Some straight forward but tedious algebra shows that for V^{n+1} defined by Eq. (11) the stability condition reduces to

$$1 \leq \frac{2 + |c|}{1 + |c|}$$

which is always true. Thus, the modified symmetric Gauss–Seidel method is shown to be (locally) unconditionally stable for the model equation. This conclusion remains true for an arbitrary number of symmetric sweeps. The method is also unconditionally stable when diffusion is added to the model equation.

2.4 Modified Odd/Even Jacobi Iteration

An examination of Eqs. (6), (7), and (8) shows that for left to right grid sweeps, the value of $\boldsymbol{Q}_j^{(m+1)}$ depends on the values of $\boldsymbol{Q}^{(m+1)}$ at all the grid points to the left of j while for right to left grid sweeps, the value of $\boldsymbol{Q}_j^{(m+1)}$ depends on the values of $\boldsymbol{Q}^{(m+1)}$ at all grid points to the right of j. Thus, it is clearly seen that that the modified Gauss–Seidel method is recursive, and as was mentioned in the introduction, this prevents an efficient implementation of the method on computors with vector architectures. In order to develop a method which is non-recursive, one might be tempted to modify Eqs. (7) and (8) such that they are replaced by the single equation

$$\boldsymbol{Rhs}_j^{(m)} = -\left[\frac{1}{\Delta t}\left(\boldsymbol{Q}_j^{(m)} - \boldsymbol{Q}_j^n\right) + \frac{1}{2\Delta x}\left(\boldsymbol{F}_{j+1}^{(m)} - \boldsymbol{F}_{j-1}^{(m)}\right)\right] \tag{12}$$

with Eq. (6) and the definition of b_j remaining unchanged. However, if one carries out a Fourier stability analysis of the model scalar equation for this modified Jacobi-like method, then one can show that the amplification $\Delta V^{(m+1)}$ can be defined recursively by the relation

$$b\Delta V^{(m+1)} = V^n(r) + \Delta V^{(m)}(b + r - 1) \tag{13}$$

where b and r are as previously defined in section 2.3 and $\Delta V^{(0)} = 0$. Computation of the magnitude of the amplification factor for this scheme indicates that for $|c| \geq 1$, this method tends to amplify long to medium wavelength disturbances. Numerical computations also reveal that as more sweeps are performed, this amplification becomes more and more severe. Thus, the Jacobi-like iteration is unstable for large time steps. In order to remedy this situation, we propose to implement a modified Odd/Even Jacobi iteration. In this method one first computes the the values of $\boldsymbol{Q}_j^{(m+1)}$ for even integers j by using Eqs. (6) and (12). One next computes the $\boldsymbol{Q}_j^{(m+1)}$ terms for odd integers j with $\boldsymbol{Rhs}_j^{(m)}$ defined to be

$$\boldsymbol{Rhs}_j^{(m)} = -\left[\frac{1}{\Delta t}\left(\boldsymbol{Q}_j^{(m)} - \boldsymbol{Q}_j^n\right) + \frac{1}{2\Delta x}\left(\boldsymbol{F}_{j+1}^{(m+1)} - \boldsymbol{F}_{j-1}^{(m+1)}\right)\right]. \tag{14}$$

This two step process is defined to be one sweep of the grid and one implements two to three sweeps of the grid per time step before setting $\boldsymbol{Q}_j^{n+1} = \boldsymbol{Q}_j^{(M)}$.

In order to analyze the amplification properties of the modified Odd/Even Jacobi method, we examine how the odd and even Fourier modes are amplified separately. If one carries out a Fourier stability analysis for the model scalar equation, then one can show that the amplitudes for the even and odd Fourier modes are given recursively by the expressions

$$b\Delta V_e^{(m+1)} = V^n(r) + \Delta V_e^{(m)}(b-1) + \Delta V_o^{(m)}(r) \tag{15}$$

and

$$b\Delta V_o^{(m+1)} = V^n(r) + \Delta V_o^{(m)}(b-1) + \Delta V_e^{(m+1)}(r) \tag{16}$$

where the subscripts e and o refer to the even and odd modes respectively. b and r are as previously defined and

$$\Delta V_o^{(0)} = \Delta V_e^{(0)} = 0.$$

The amplification factors for the even and odd modes from one time step to the next can be computed numerically by using Eqs. (15) and (16). Such computations indicate that the modifed Odd/Even Jacobi method is stable for any value of c if one implements a sufficient number of sweeps. Numerical computations also reveal that the odd and even amplification factors are asymptoticaly convergent to the modified Gauss–Seidel amplification factor.

More details on the locally implicit scheme can be found in Reddy and Jacocks (1987) , Nayani (1988) and Towne(1989).

2.5 Artificial Dissipation Filter

The locally implicit schemes described above are not sufficient for performing fluid flow calculations. The central difference schemes are known to produce spurious oscillations and suffer from pressure undershoots (i.e., negative pressures) in rarefaction regions and oscillations in the vicinity of moderate to strong shocks.

The traditional method of controlling these problems is to add artificial dissipation terms to the Euler equation and then solve the augmented equations (Jameson (1985A), Pulliam (1985)). In the present approach, the artificial dissipation is applied as a filter. The filter is consistent with the definition of a filter as given by Engquist, et al. (1989) but is implemented quite differently.

Let $\boldsymbol{Q}_j^{(M)}$ be the numerically computed approximation to $\boldsymbol{Q}_j^{n+1}$ after M sweeps. The value of $\boldsymbol{Q}_j^{n+1}$ is computed to be

$$\boldsymbol{Q}_j^{n+1} = \boldsymbol{Q}_j^{(M)} + \mathcal{S}(\boldsymbol{Q}^{(M)})_j, \tag{17}$$

where $\mathcal{S}(\boldsymbol{Q}^{(M)})_j$ is a nonlinear filter defined by

$$\mathcal{S}(\boldsymbol{Q}^{(M)})_j = \left(\psi_j^{(2)}\boldsymbol{S1}_j - \psi_{j-1}^{(2)}\boldsymbol{S1}_{j-1}\right) - \left(\psi_j^{(4)}\boldsymbol{S3}_j - \psi_{j-1}^{(4)}\boldsymbol{S3}_{j-1}\right). \tag{18}$$

Here

$$\boldsymbol{S1}_j = \boldsymbol{Q}_{j+1}^{(M)} - \boldsymbol{Q}_j^{(M)} \quad ; \quad \boldsymbol{S3}_j = \boldsymbol{S1}_{j+1} - 2\boldsymbol{S1}_j + \boldsymbol{S1}_{j-1}. \tag{19}$$

The dissipation coefficients $\psi^{(2)}$ and $\psi^{(4)}$ are chosen such that the second order dissipation term is significant only near shocks while the fourth order term is used to damp oscillations in smooth flow regions and is turned off near shocks. This is accomplished by the use of a pressure sensor. Following an idea originally proposed by Baldwin and McCormack (1975) and later developed by Jameson, one defines

$$\nu_j = \frac{|P_{j+1} - 2P_j + P_{j-1}|}{|P_{j+1} + 2P_j + P_{j-1}|} \quad ; \quad \bar{\nu}_j = \max\,(\nu_{j+1}, \nu_j). \tag{20}$$

The second and fourth order dissipation coefficients $\psi_j^{(2)}$ and $\psi_j^{(4)}$ are then defined to be

$$\psi_j^{(2)} = \kappa^{(2)}\psi_j\bar{\nu}_j \quad ; \quad \psi_j^{(4)} = \kappa^{(4)}\psi_j \max\,(0.0, 1.0 - \kappa\bar{\nu}_j) \tag{21}$$

where $\kappa^{(2)}$, $\kappa^{(4)}$, and κ are adjustable parameters and ψ_j is defined to be

$$\psi_j = \frac{1}{2}\left(\lambda_{j+1} + \lambda_j\right), \tag{22}$$

where

$$\lambda_j = \left(|u_j| + a_j\right) = \text{ spectral radius of } \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}. \tag{23}$$

The $\boldsymbol{S3}$ terms can also be modeled after Jameson's (1985B) flux limited artificial dissipation. In this approach the $\boldsymbol{S3}_j$ term becomes

$$\boldsymbol{S3}_j = B\left(\boldsymbol{S1}_{j+1}, \boldsymbol{S1}_j\right) - 2\boldsymbol{S1}_j + B\left(\boldsymbol{S1}_j, \boldsymbol{S1}_{j-1}\right), \tag{24}$$

where $B(p,q)$ is Roe's MINMOD function defined for scalars by the equation

$$\begin{aligned} B(p,q) &= \left(S(p) + S(q)\right)\ \min\left(|p|, |q|\right); \\ S(p) &= \begin{cases} \frac{1}{2} & ; p \geq 0 \\ -\frac{1}{2} & ; p < 0 \end{cases} \quad . \end{aligned} \tag{25}$$

The function is applied to vectors component by component.

At the boundaries of the computational domain, the dissipative flux terms $\boldsymbol{S1}$ and $\boldsymbol{S3}$ are set equal to $\boldsymbol{0}$ in order to insure that density, momentum, and energy are neither created nor destroyed at the boundaries by the filter.

CHAPTER 3

ONE DIMENSIONAL TEST PROBLEMS

3.1 Shock Tube Problem

As a test of the technique, we apply the modified Gauss–Seidel scheme with the flux limited artificial dissipation filter to the shock tube problem recommended by Sod (1978) as a standard test problem for finite difference schemes for the solution of hyperbolic systems of conservation laws.

The problem is as follows. The gas is assumed to be held at rest with a partition located at the origin ($x = 0$) separating the gas into two seperate states. The left hand side ($x < 0$) of the gas is held at a higher density and pressure than the right hand side ($x > 0$). The initial conditions of the gas are

$$\boldsymbol{Q}_L = (1.0, 0.0, 2.5) \;\; ; \;\; \boldsymbol{Q}_R = (0.125, 0.0, 0.25). \tag{26}$$

At time $t = 0.0$, the partition separating the gas is assumed to be removed instantaneously. As a result a shock wave propagates into the low pressure ($x > 0$) side of the gas and an expansion wave propagates into the high pressure ($x < 0$) side of the gas.

In all the cases reported in this section, the values of $\kappa^{(2)}$, $\kappa^{(4)}$, and κ are held fixed at the values 0.2, 0.03, and 12.0, and a Courant–Friedrichs–Léwy (CFL) number of 0.9 is used in all cases where the CFL number is defined to be

$$\mathrm{CFL} = \frac{\max(\lambda_j)\,\Delta t}{\Delta x}.$$

The method is advanced one time step to the next by applying three symmetric sweeps before the variables are updated.

The numerically computed velocity, and energy per unit mass after 40 time steps with $\Delta x = 0.1$ are shown in Fig. 1. The exact solution of the Euler equation is plotted as the solid line and the solution computed by the modified Gauss–Seidel method plus the artificial dissipation filter is shown as circles centered at the computed points. The scheme maintains its spatial order of accuracy in smooth regions, but it smears the shocks somewhat and the contact discontinuity even more so.

In Fig. 2, the velocity and energy are shown for the same test problem, but Jameson-type artificial dissipation is included as part of the equation rather than applied as a filter. Comparison of the graphs shows that in this case the filter is superior to the traditional method in suppressing overshoots and spurious oscillations. Also, on a SUN–4 computer, the filter requires less than a third of the CPU time to execute than the scheme with bulit–in dissipation.

Engquist, et al. (1989) have shown that for scalar conservation laws a consistent finite difference scheme with a filter will converge to the correct weak solution in the limit $\Delta t, \Delta x \to 0$ provided that the difference solution converges in this limit. As a test of these ideas for systems of equations, the density and velocity after 400 time steps with a fine mesh ($\Delta x = 0.01$) are presented in Fig. 3 (only every tenth point is plotted for clarity). In this case the shocks are not smeared at all by the artificial dissipation and the contact discontinuity is only slightly smeared. In this limit very sharp nonoscillatory solutions are obtained.

3.2 Nozzle Flow

We consider the nozzle flow problem of Shubin, Glaz, and Stephen (1981) as our test case for steady flow problems. Shubin, et. al. considered a nozzle with a

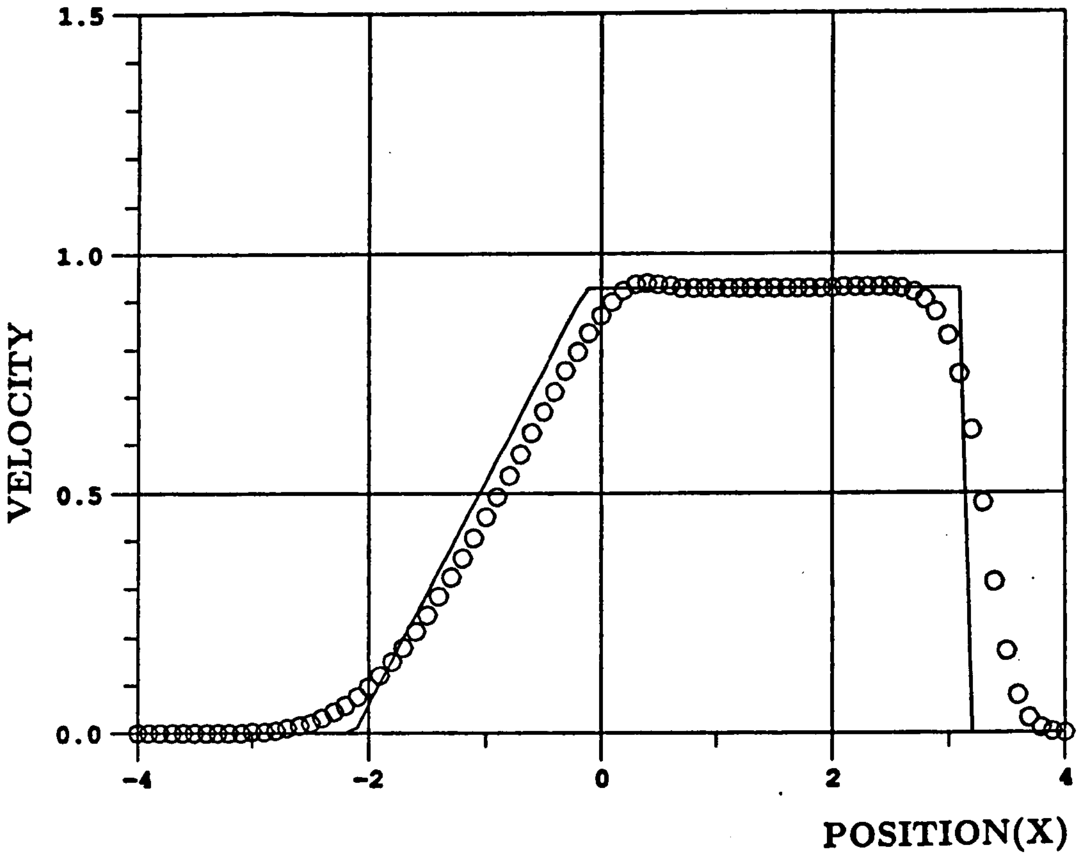

Figure 1: Modified Gauss–Seidel Method with Artificial Dissipation Filter for Sod's Shock Tube (Coarse Mesh, $\Delta x = 0.1$)

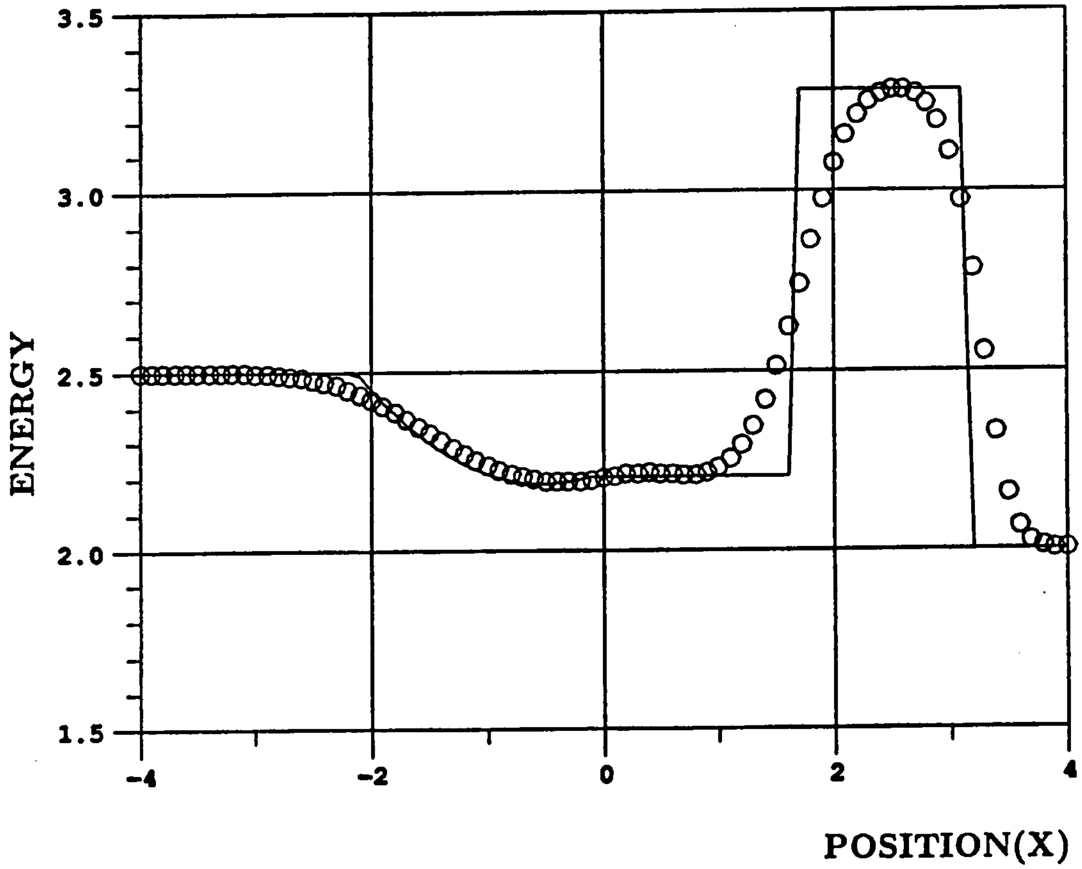

Figure 1: Continued

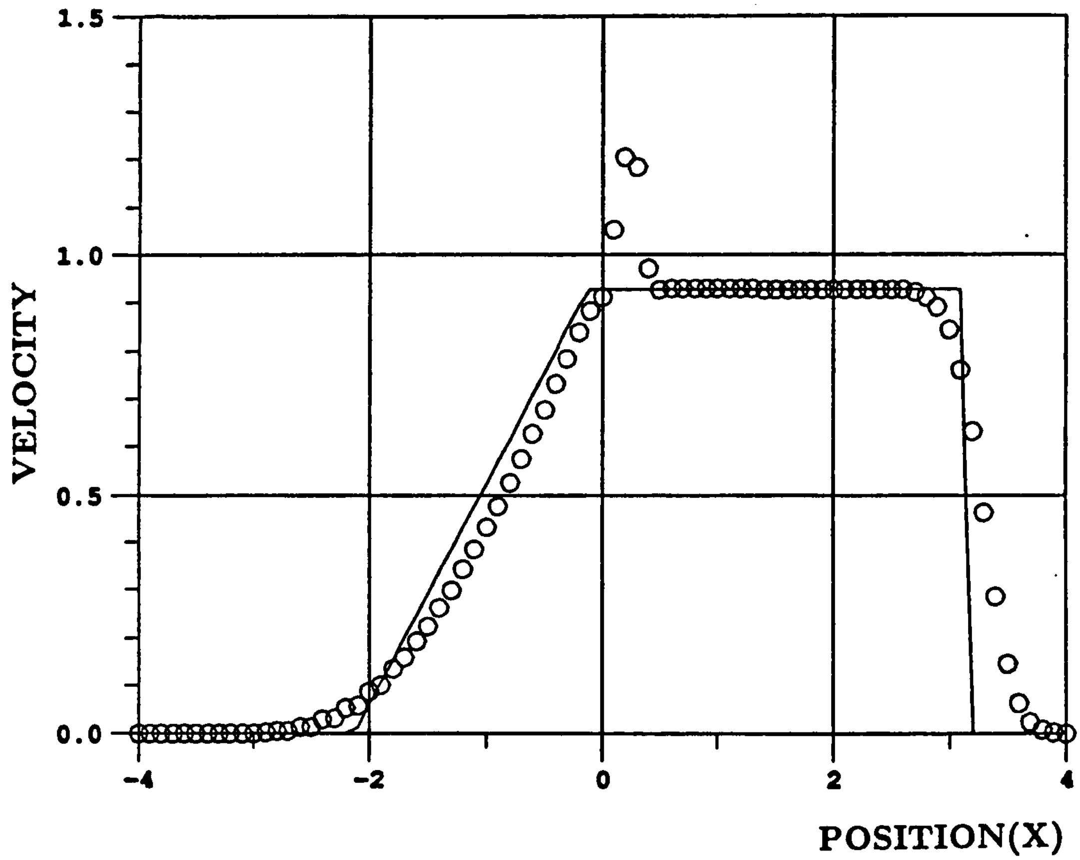

Figure 2: Modified Gauss–Seidel Method with Jameson–Type Artificial Dissipation for Sod's Shock Tube (Coarse Mesh, $\Delta x = 0.1$)

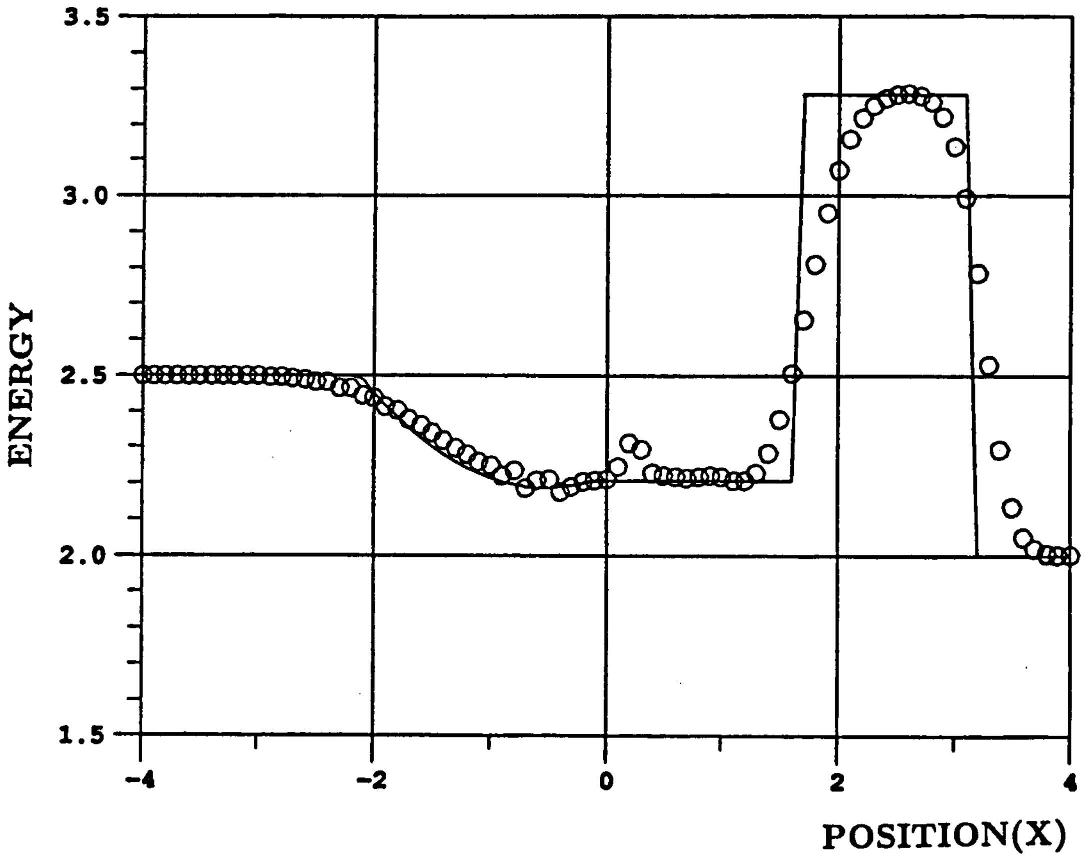

Figure 2: Continued

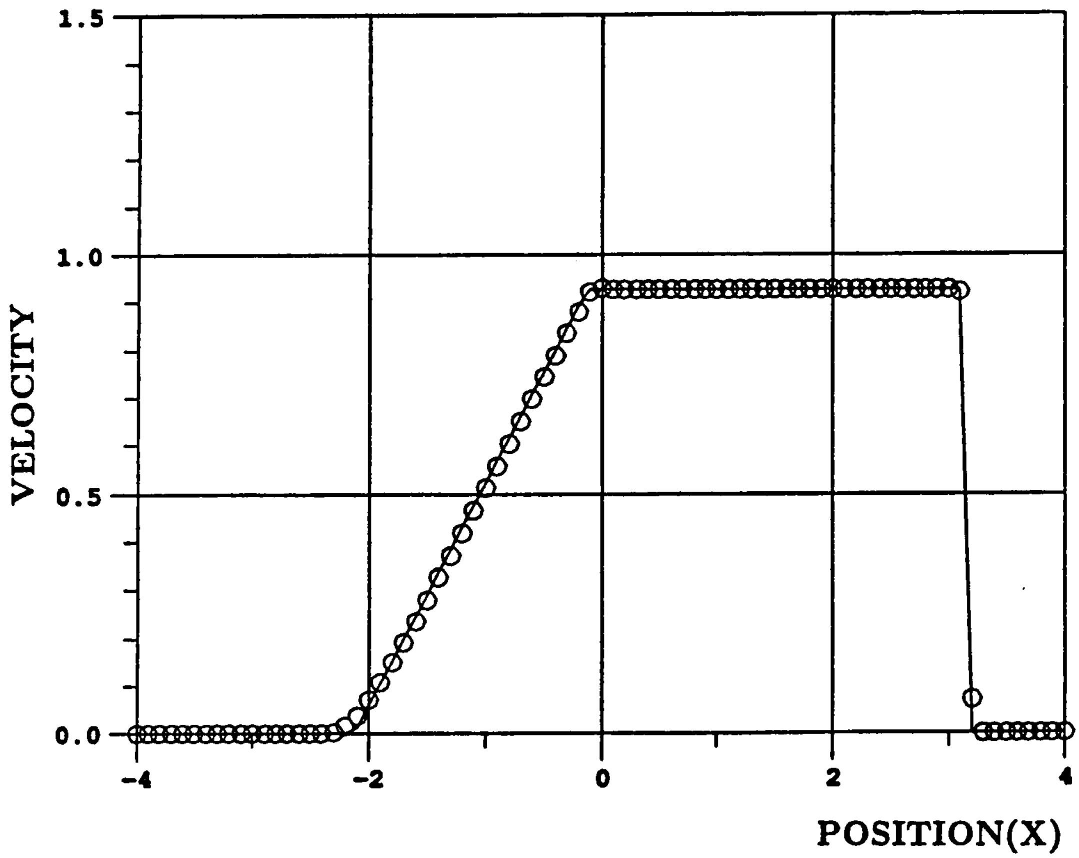

Figure 3: Modified Gauss–Seidel Method with Artificial Dissipation Filter for Sod's Shock Tube (Fine Mesh, $\Delta x = 0.01$)

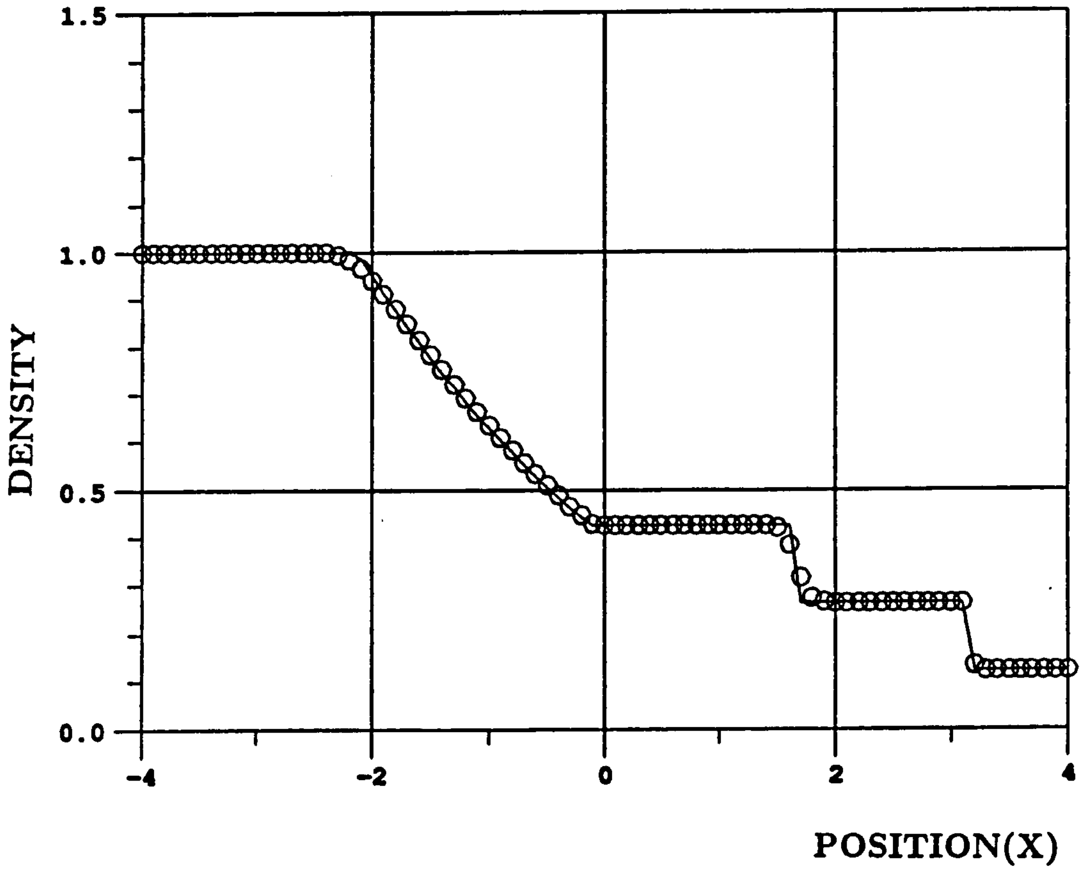

Figure 3: Continued

supersonic inflow and a subsonic outflow with an intervening shock. The governing equation is

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = \boldsymbol{H}(\boldsymbol{Q}), \tag{27}$$

where $\boldsymbol{Q}$ is the vector $A(x)*(\rho, \rho u, e)$, $\boldsymbol{F}$ is the vector $A(x)*(\rho u, \rho u^2 + P, u(e+P))$, and $\boldsymbol{H}$ is the vector $(0, P\frac{dA}{dx}, 0)$. $A(x)$ is the area variation of the nozzle given by relation $A(x) = 1.398 + 0.347 * TANH(0.8 * x - 4)$, $x \in [0.0, 10.0]$.

For these equations the locally implicit scheme still has the form of Eq. (3). b_j is as defined before, but for the modified Odd/Even Jacobi method $\boldsymbol{Rhs}_j^{(m)}$ is defined to be

$$\boldsymbol{Rhs}_j^{(m)} = -\left[\frac{1}{\Delta t}\left(\boldsymbol{Q}_j^{(m)} - \boldsymbol{Q}_j^n\right) + \frac{1}{2\Delta x}\left(\boldsymbol{F}_{j+1}^{(m)} - \boldsymbol{F}_{j-1}^{(m)}\right) - \boldsymbol{H}\left(\boldsymbol{Q}_j^{(m)}\right)\right] \tag{28}$$

for even integers j and

$$\boldsymbol{Rhs}_j^{(m)} = -\left[\frac{1}{\Delta t}\left(\boldsymbol{Q}_j^{(m)} - \boldsymbol{Q}_j^n\right) + \frac{1}{2\Delta x}\left(\boldsymbol{F}_{j+1}^{(m+1)} - \boldsymbol{F}_{j-1}^{(m+1)}\right) - \boldsymbol{H}\left(\boldsymbol{Q}_j^{(m)}\right)\right] \tag{29}$$

for odd integers j.

For initial conditions the variables are set equal to the free stream conditions, $\boldsymbol{Q}_j^0 = A(j) * (1.0, 1.26, 3.29)$. The boundaries are treated as follows. The $j = 1$ boundary ($x = 0.0$) is a supersonic inflow, and thus the independent variables are held fixed at the free stream conditions. The $j = J$ boundary ($x = 10.0$) where J is the total number of grid points is a subsonic outflow. This is specified by fixing the exit pressure at Pexit $= 1.40844$. The density and momentum are extrapolated from the interior by the formula

$$\rho_J = 2\rho_{J-1} - \rho_{J-2} \quad , etc. \tag{30}$$

The energy is computed from the exit pressure and the extrapolated density and momentum.

The modified Odd/Even Jacobi method with flux limited artificial dissipation filter is used to advance the solution as many time steps as necessary to reach a converged solution. Two grid sweeps are performed each time step before the variables are updated.

In order to demonstrate the spatial accuracy of the method, we computed the solution with 21 grid points. The computed density normalized by the total density after 75 time steps with a CFL number of 10.0 is shown in Fig. 4. The solution is compared with a 251 grid point solution which is displayed as a solid line. In all test cases, the flux limited dissipation filter is used with $\kappa = 12.0$, $\kappa^{(2)} = 0.2$, and $\kappa^{(4)} = 0.03$. In Fig. 5 the density computed for the same test problem by a Beam and Warming (1976) block tri–diagonal algorithm with Jameson-type artificial dissipation is shown. Comparison of Figs. 4 and 5 indicate that the iterative method produces results that agree well with those produced by the Beam and Warming method without the need for time consuming Jacobian calculations and block matrix reductions. Table 1 displays the CPU time in seconds required by the modified Gauss–Seidel, modified Odd/Even Jacobi and Beam and Warming implicit methods to execute 1000 time steps on an Alliant/FX8 mini–supercomputer equiped with vector processors. Table 1 shows that the modified Odd/Even Jacobi method executes substantially faster on a vectorized computer than either the modified Gauss–Seidel or the Beam and Warming methods.

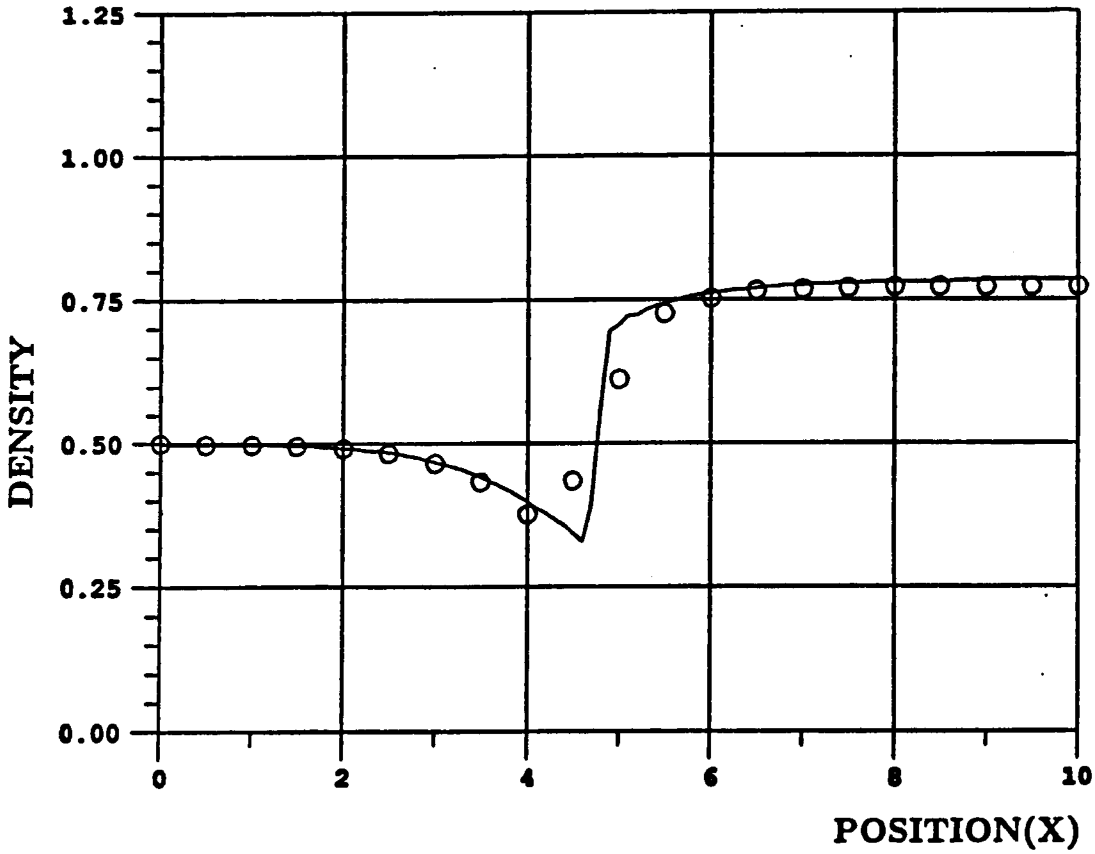

Figure 4: Modified Odd/Even Jacobi Method with Artificial Dissipation Filter for Shubin's Nozzle ($\Delta x = 0.5$)

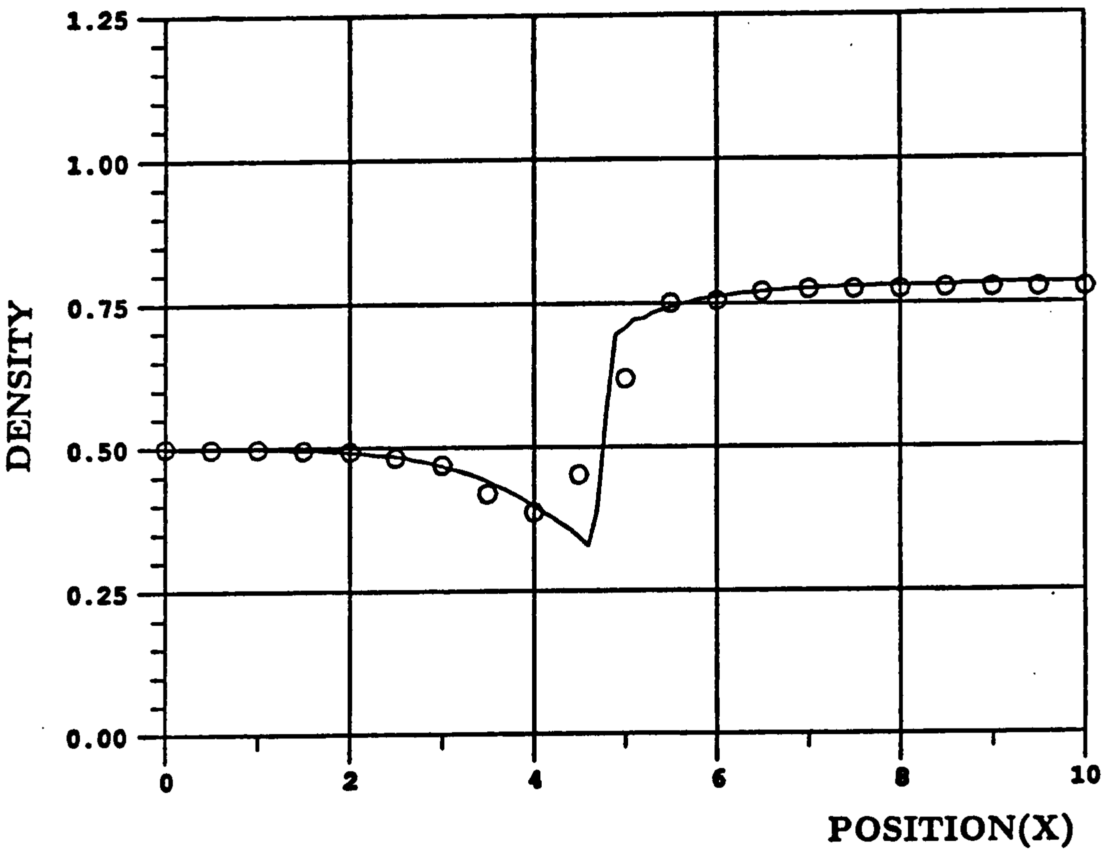

Figure 5: Beam and Warming Implicit scheme with Jameson–Type Artificial Dissipation for Shubin's Nozzle ($\Delta x = 0.5$)

Table 1. Execution Times in Seconds of Various Methods on Alliant/FX8

method	CPU time
modified Gauss–Seidel	28.4
modified Odd/Even Jacobi	13.5
Beam and Warming	33.6

Fig. 6 displays a convergence history for the modified Gauss–Seidel, modified Odd/Even Jacobi, and Beam and Warming methods. The base 10 log of the normalized L_2 norm of $(Q^{n+1} - Q^n)$ is plotted vs. the iteration count n. Local time stepping is used in all cases. With local time stepping the definition of b_j is altered to be

$$b_j = \frac{1}{\Delta t_j} + \frac{\lambda_j}{2\Delta x}$$

where Δt_j is defined to be

$$\Delta t_j = \frac{\mathrm{CFL}\Delta x}{\lambda_j}.$$

Both the iterative methods ran with a CFL number of 10.0. The Beam and Warming code ran with a CFL number of 4.0. This value was experimentally determined to produce the best convergence rate for the Beam and Warming method. The results show that all methods have similar convergence rates. Thus, Fig. 6 also shows that the modified Odd/Even Jacobi method converges almost as well as the modified Gauss–Seidel method while maintaining its ability to be implemented efficiently on vector computers.

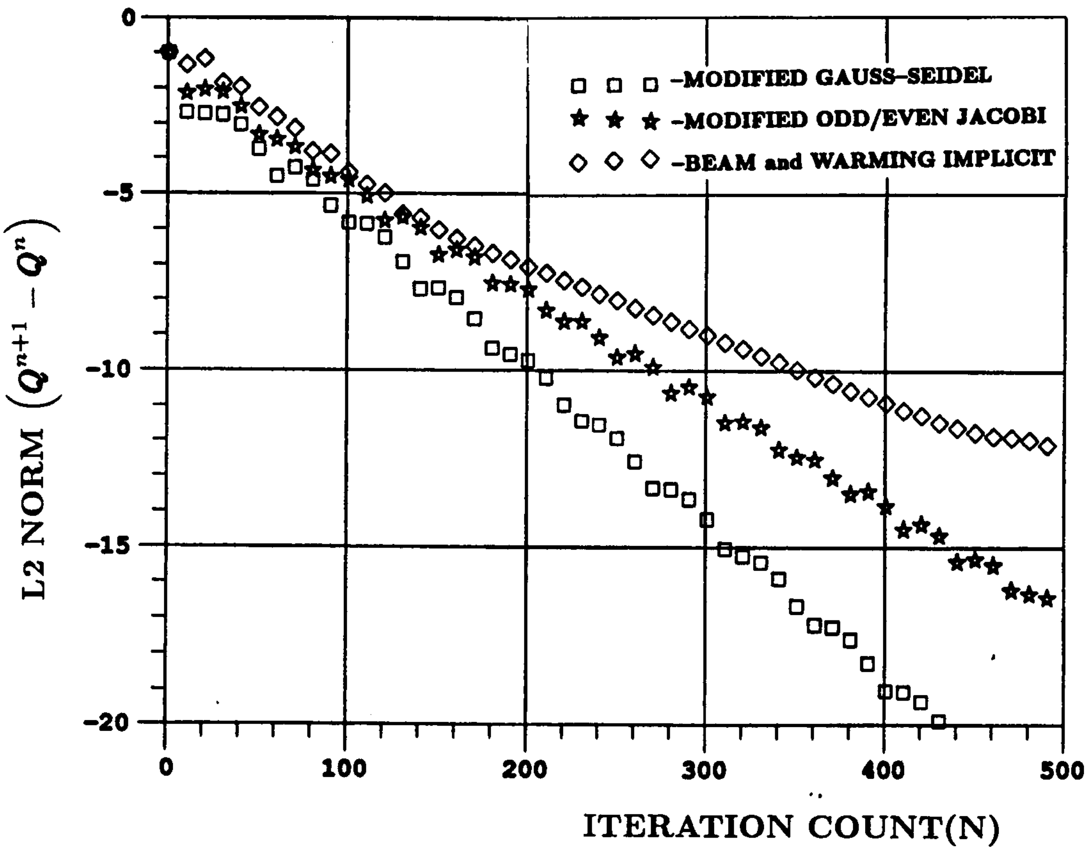

Figure 6: Convergence History of Various Methods for Shubin's Nozzle ($\Delta x = 0.5$)

CHAPTER 4

TWO DIMENSIONAL SHOCK REFLECTION PROBLEM

As a final test of both the artificial dissipation filter and the modified Odd/Even Jacobi method we apply both techniques to the inviscid shock reflection problem considered by Harten, Yee, and Warming (1983).

This problem is governed by the two dimensional Euler equation. This equation can be written in the form

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = \boldsymbol{0}, \tag{31}$$

where $\boldsymbol{Q}$ is the vector $(\rho, \rho u, \rho v, e)$ of conserved variables and $\boldsymbol{F}$ and $\boldsymbol{G}$ are respectively the flux vectors $(\rho u, \rho u^2 + P, \rho uv, u(e+P))$ and $(\rho v, \rho uv, \rho v^2 + P, v(e+P))$. Here ρ is the density, u the velocity in the x direction, v the velocity in the y direction, e the energy per unit volume, and P the pressure. The pressure is related to the other variables for a perfect gas by the relation $P = (\gamma - 1)\left(e - \frac{1}{2}(\rho u^2 + \rho v^2)\right)$.

In all of the test cases reported below, the computational domain will be the rectangle $0.0 \leq x \leq 4.1$ and $0.0 \leq y \leq 1.0$. All the numerical computations reported in this chapter are run on an uniform 61x21 grid. The test case considered is that of an uniform inflow at mach 2.9 encountering a shock aligned at 29^o to the free stream. The indexing of the computational domain is shown in Fig. 7, while the exact solution for the pressure is shown in Fig. 8.

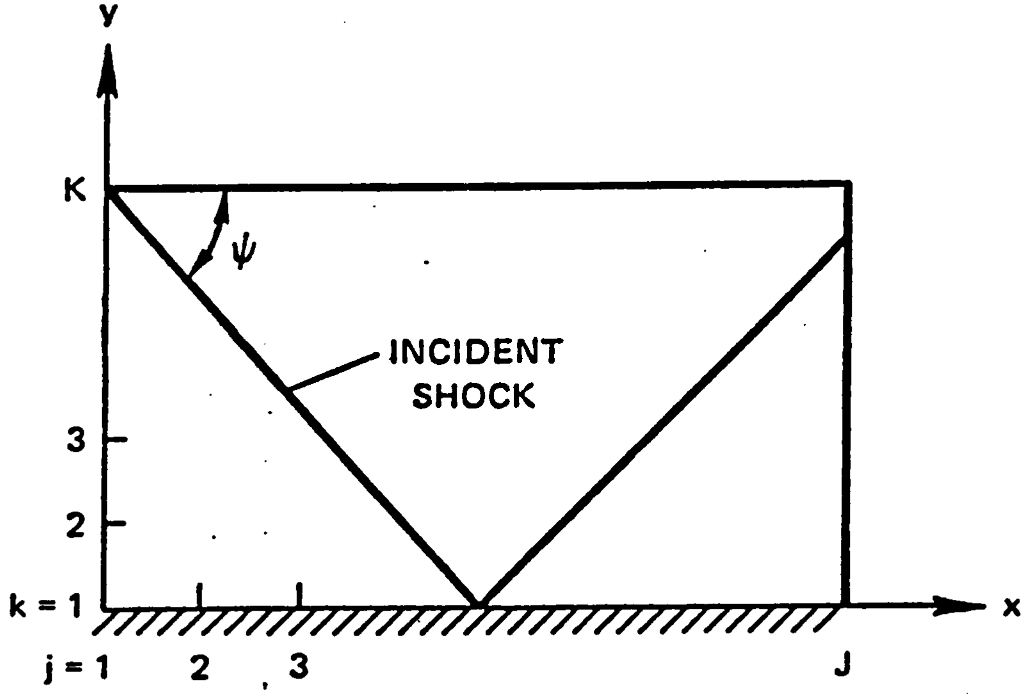

Figure 7: Indexing of Computational Domain for 2–D Shock Reflection Problem

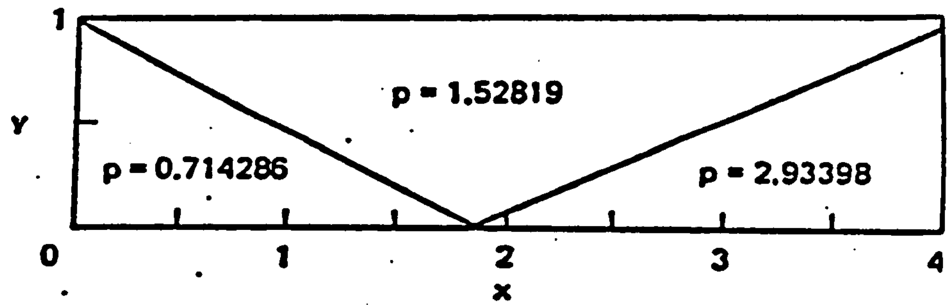

Figure 8: Eact Pressure Solution for Shock Reflection Problem

With the indexing of the computational domain as shown in Fig. 7, the Euler implicit scheme for Eqn. (31) with central difference spatial approximations is written at each grid point j,k as follows

$$\frac{1}{\Delta t}\left(Q^{n+1}_{j,k}-Q^{n}_{j,k}\right)+\frac{1}{2\Delta x}\left(F^{n+1}_{j+1,k}-F^{n+1}_{j-1,k}\right)+\frac{1}{2\Delta y}\left(G^{n+1}_{j,k+1}-G^{n+1}_{j,k-1}\right)=0. \tag{32}$$

$Q^{n+1}_{j,k}$ is the value of the vector Q at the j,kth lattice point at time step $n+1$, while $F^{n+1}_{j,k}$ is the value of the vector F evaluated using $Q^{n+1}_{j,k}$, etc. . Δt is the lattice spacing in the time direction, Δx the lattice spacing in the x direction, and Δy the lattice spacing in the y direction.

4.1 2-D Locally Implicit Scheme

The system of Eqs. (32) is solved from one time step to the next by a combined modified Gauss–Seidel and Odd/Even Jacobi method. In this technique a modified Odd/Even Jacobi iteration is applied in the x direction with a modified Gauss–Seidel iteration applied in the y direction. The iteration is started by setting $Q^{(0)}_{j,k}$ equal to $Q^{n}_{j,k}$ at every grid point j,k where again $Q^{(m)}_{j,k}$ represents the approximation to $Q^{n+1}_{j,k}$ after m sweeps. The iteration starts at the $j=1$, $k=1$ corner of the computational domain. First, all the values of $Q^{(1)}_{j,1}$ when j is an even integer are computed. Next, the values of $Q^{(1)}_{j,1}$ are computed when j is an odd integer. The sweep moves to the $k=2$ level when all the $Q^{(1)}_{j,1}$ terms have been computed. At this level another odd/even sweep is carried out making use of the previously computed $Q^{(1)}_{j,1}$ terms when computing the $Q^{(1)}_{j,2}$ terms. This process is then repeated until the $k=K$ level has been updated. At this stage the process is reversed with the sweep starting at the $j=J$, $k=K$ corner of the computational domain and ending at the $j=1$, $k=1$ level. This constitutes one symmetric sweep.

At each point j,k during an iterative sweep the value of $\boldsymbol{Q}_{j,k}^{(m)}$ is updated as follows

$$\boldsymbol{Q}_{j,k}^{(m+1)} = \boldsymbol{Q}_{j,k}^{(m)} + \boldsymbol{Rhs}_{j,k}^{(m)}/b_{j,k}, \tag{33}$$

where $b_{j,k}$ is given by the formula

$$b_{j,k} = \frac{1}{\Delta t} + \left(\frac{1}{2\Delta x} + \omega_{imp}\right)\left(\left|u_{j,k}^{n}\right| + a_{j,k}^{n}\right) + \left(\frac{1}{2\Delta y} + \omega_{imp}\right)\left(\left|v_{j,k}^{n}\right| + a_{j,k}^{n}\right)$$

where ω_{imp} is the so called implicit smoothing factor and is included because it has been found to increase the stability of the scheme and $a_{j,k}^{n}$ is the local speed of sound computed using $\boldsymbol{Q}_{j,k}^{n}$. The right hand side of Eq. (33), $\boldsymbol{Rhs}_{j,k}^{(m)}$, in this case takes the form

$$\begin{aligned} \boldsymbol{Rhs}_{j,k}^{(m)} = -\Bigg[& \frac{1}{\Delta t}\left(\boldsymbol{Q}_{j,k}^{(m)} - \boldsymbol{Q}_{j,k}^{n}\right) + \frac{1}{2\Delta x}\left(\boldsymbol{F}_{j+1,k}^{(*)} - \boldsymbol{F}_{j-1,k}^{(*)}\right) \\ & + \frac{1}{2\Delta y}\left(\boldsymbol{G}_{j,k+1}^{(*)} - \boldsymbol{G}_{j,k-1}^{(*)}\right)\Bigg], \end{aligned} \tag{34}$$

where $\boldsymbol{G}^{(*)}$ indicates that the vector $\boldsymbol{G}$ is evaluated using $\boldsymbol{Q}^{(m+1)}$ if it has been computed at that grid point and $\boldsymbol{Q}^{(m)}$ otherwise, etc. For time accurate calculations three to four symmetric sweeps are needed per time step, while on steady state problems only one to two symmetric sweeps are implemented before updating $\boldsymbol{Q}_{j,k}^{n+1}$.

4.2 <u>2–D Artificial Dissipation Filter</u>

In order to control spurious oscillations in the flow field, a straight forward generalization of the one dimensional filter presented in chapter 2 can be implemented. Again let $\boldsymbol{Q}_{j,k}^{(M)}$ denote the computed approximation to $\boldsymbol{Q}_{j,k}^{n+1}$ after M sweeps. The value of $\boldsymbol{Q}_{j,k}^{n+1}$ is computed to be

$$\boldsymbol{Q}_{j,k}^{n+1} = \boldsymbol{Q}_{j,k}^{(M)} + \mathcal{S}(\boldsymbol{Q}^{(M)})_j + \mathcal{S}(\boldsymbol{Q}^{(M)})_k, \tag{35}$$

where $\mathcal{S}(\boldsymbol{Q}^{(M)})_j$ is a one dimensional nonlinear filter in the x direction and $\mathcal{S}(\boldsymbol{Q}^{(M)})_k$ is a one dimensional filter in the y direction.

This simple generalization, however, does not work. The method converges a few orders of magnitude and then it does not converge any furthur but instead falls into a limit cycle. This is overcome by computing the pre–filter term

$$\mathcal{S}^n_{j,k} = \mathcal{S}(\boldsymbol{Q}^n)_j + \mathcal{S}(\boldsymbol{Q}^n)_k$$

at each grid point j , k. These values are then used to change Eq. (33) into

$$\boldsymbol{Q}^{(m+1)}_{j,k} = \boldsymbol{Q}^{(m)}_{j,k} + \boldsymbol{Rhs}^{(m)}_{j,k}/b_{j,k} + \mathcal{S}^n_{j,k}. \qquad (36)$$

The $\mathcal{S}^n_{j,k}$ terms are held fixed during the iterative process, and are updated only at the start of the next of the computation of the next time step. The values of $\kappa^{(2)}$, $\kappa^{(4)}$ and κ are held fixed at the values of 0.2 , 0.009 , and 12.0 for both the j and k filters in the test case reported.

4.3 Initial Conditions

For initial conditions, all the variables are set equal to the free stream conditions at every grid point except on the $k = K$ boundary where analytical boundary conditions are applied.

4.4 Boundary Conditions

Analytical boundary conditions are applied at the $j = 1$ and $k = K$ boundaries. The variables at the $j = 1$ boundary are held fixed at the free stream conditions while those at the $k = K$ boundary are held fixed at the values necessary to to produce the appropiate shock strengh and location. Zeroth–order boundary conditions are applied at the $j = J$ boundary (i.e. $\boldsymbol{Q}_{J,k} = \boldsymbol{Q}_{J-1,k}, k = 1, 2, ..., K$).

On the solid wall ($k = 1$) boundary, the velocity normal to the wall $v_{j,1}$ is set equal to zero for $j = 1, 2, ..., J$ while zeroth–order extrapolation is applied to the density and x velocity with a first order extrapolation applied to the pressure (i.e. $P_{j,1} = 2P_{j,2} - P_{j,3}$). The energy is then computed from the extrapolated pressure, density, and velocity.

4.5 Numerical Results

The pressure contours computed on a uniform 61x21 grid are shown in Fig. 9. This solution is run 100 steps with a CFL number of 10.0 and $\omega_{imp} = 0.1$. For comparison, the solution computed by Harten, et. al. (1983) with a total variation diminishing (TVD) scheme is shown in Fig. 10. The contours shown in Figs. 9 and 10 are computed using 40 uniform pressure increment levels of 0.1 between 0.0 and 4.0. Comparison of the figures shows that the solution computed by the present method compares favourably with the TVD scheme. The pressure coefficients $\left(c_P = \frac{2}{\gamma m_\infty^2}\left(\frac{P}{P_\infty} - 1\right)\right)$ computed by the ***LIM*** scheme and the TVD scheme are shown in Figs. 11 and 12 respectively. Again the ***LIM*** with a filter compares favorably with the TVD solution with the TVD scheme producing slightly narrower shocks and the ***LIM*** producing cleaner shocks. The filter is much simpler than the TVD scheme; thus, the ***LIM*** provides an efficient method for computing supersonic flows with embeded shocks with an accuracy comparable to the more elaborate TVD scheme.

As a final comparison the pressure coefficient computed by a Beam and Warming block tri–diagonal algorithm (***ARC2D***) is shown in Fig. 13. Fig. 14 shows a covergence history of the ***LIM*** and Beam and Warming scheme. The Beam and Warming scheme ran with a CFL number of 4.0 and the ***LIM*** ran with a CFL

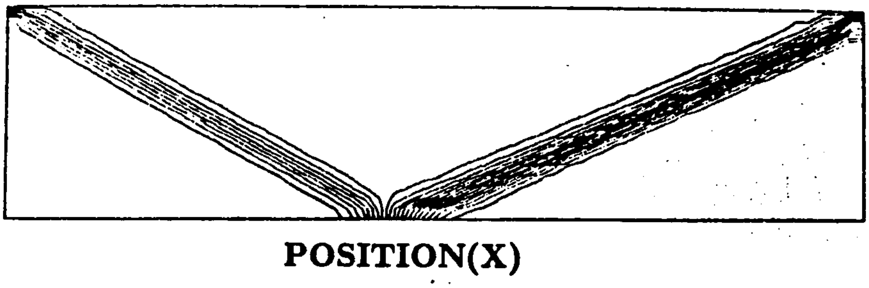

Figure 9: Pressure Contours Computed with Artificial Dissipation Filter

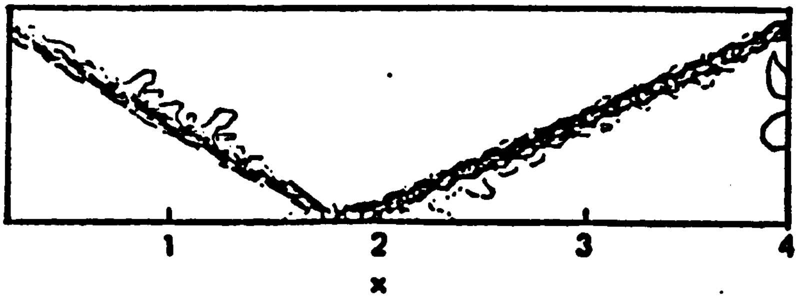

Figure 10: Pressure Contours Computed with Implicit TVD Scheme

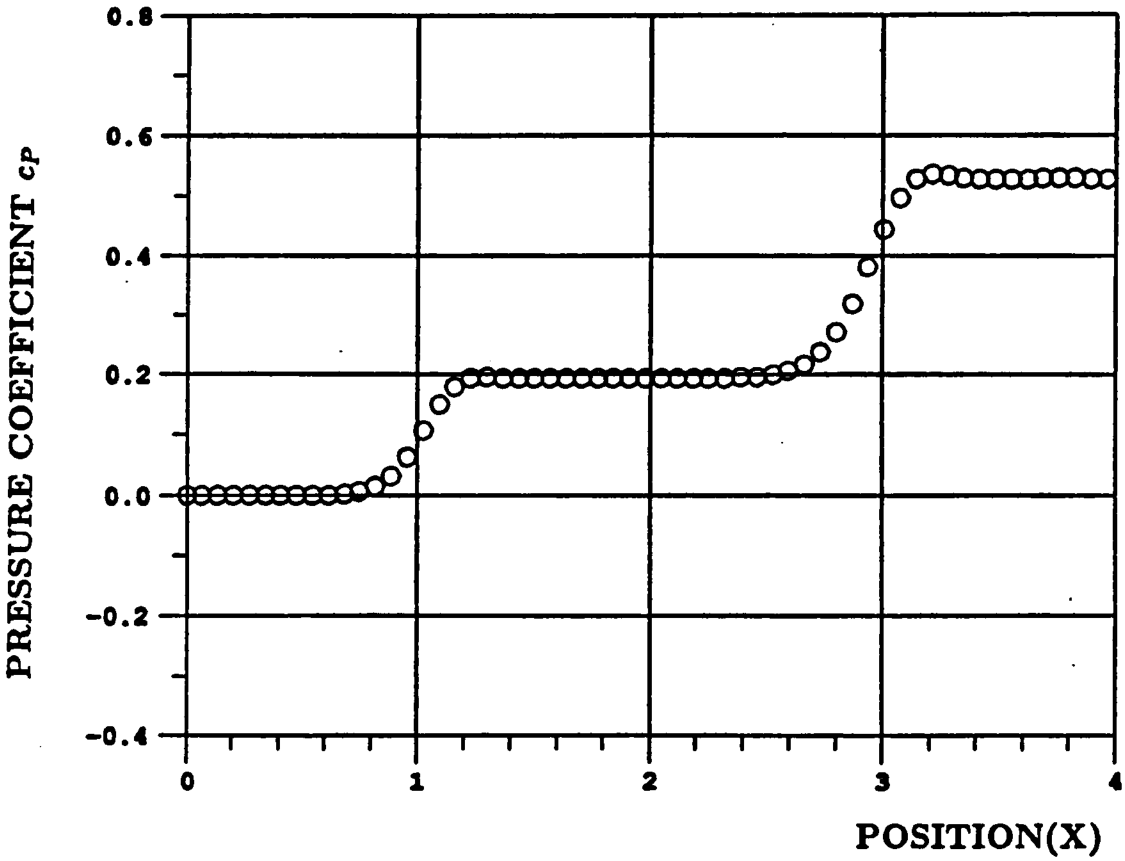

Figure 11: Pressure Coefficient at $y = 0.5$ Computed with Artificial Dissipation Filter

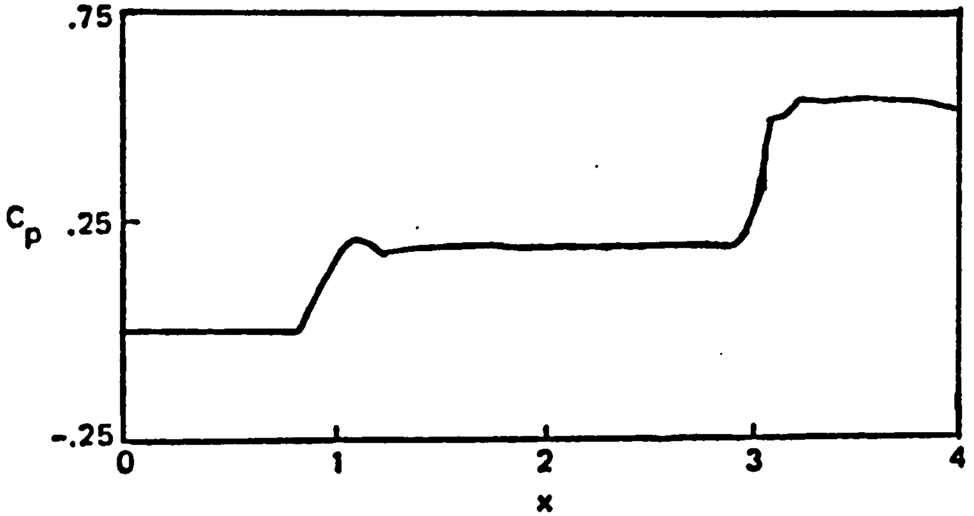

Figure 12: Pressure Coefficient at $y = 0.5$ Computed with an Implicit TVD Scheme

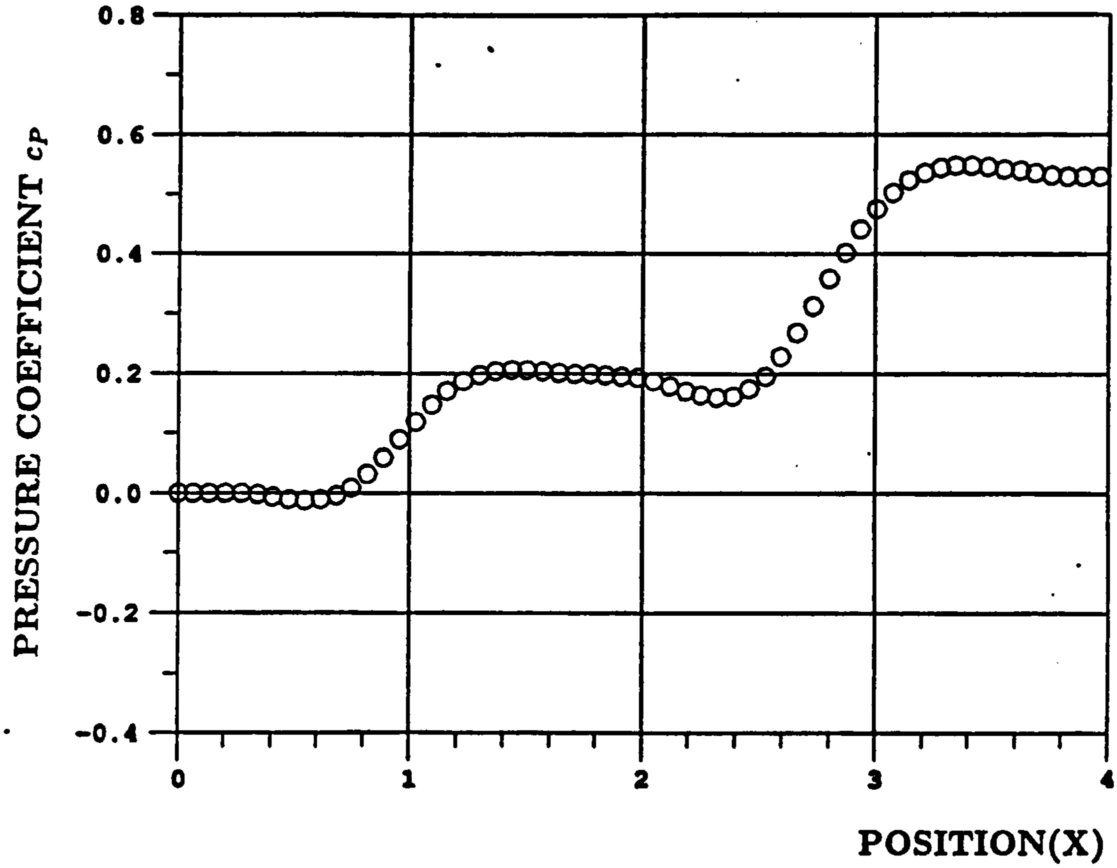

Figure 13: Pressure Coefficient at $y = 0.5$ Computed with a Beam and Warming Implicit Scheme

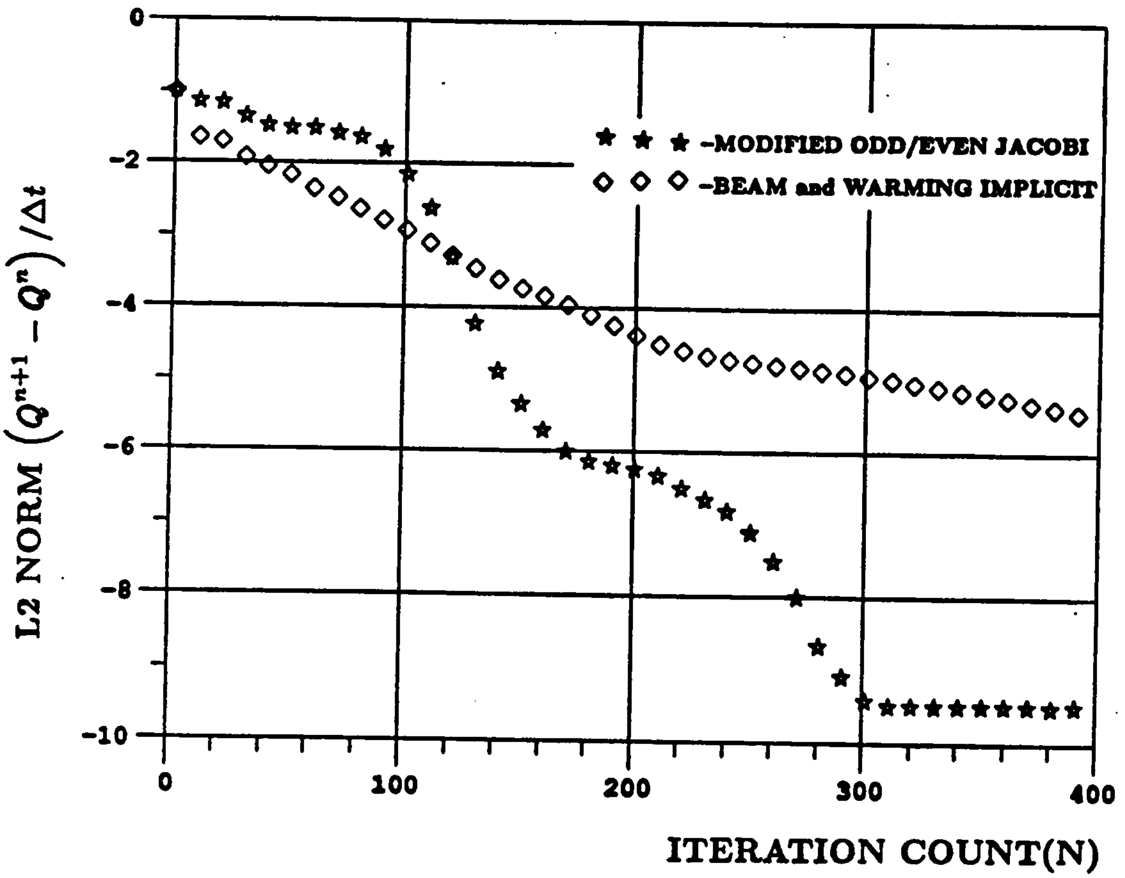

Figure 14: Convergence History of Various Methods for 2-D Shock Reflection Problem

number of 10.0. In both cases the CFL numbers was fixed at the values which produced the best convergenge rate. Table 2 contains the CPU time in seconds that the methods needed to execute an an Alliant/FX8 computer. The solution computed by the ***LIM*** is cleaner than the Beam and Warming scheme and as Table 2 shows again executes substantially faster on a vector computer.

Table 2. Execution Times in Seconds for Various Methods on Alliant/FX8

method	CPU time
LIM+filter	277
Beam and Warming	399.2

CHAPTER 5

CONCLUSIONS

A nonlinear filter based on flux limited artificial dissipation has been tested on a variety of both steady and unsteady as well as one and two dimensional test problems for the Euler Equation. The filter has been shown to be a cost effective way to control spurious oscillations in the flow field. The filter substantially reduces the computaional cost of the artificial dissiapation terms as well as increasing the execution times of the algorithm when it is used in conjuction with the ***LIM***.

In addition, an odd/even ordering of the grid points has been considered on steady state flow problems. This ordering has been shown to remove the numerical instability associated with the standard modified Jacobi method. The modified Odd/Even Jacobi method has been shown to convergenge almost as well as the modified Gauss–Seidel method while maintaining its ability to be implemented efficiently on vector computors.

In conclusion it can be said that the modified Gauss–Seidel and modified Odd/Even Jacobi methods with an artificial dissipation filter are effective nonoscillatory shock capturing schemes for the Euler Equation.

BIBLIOGRAPHY

BIBLIOGRAPHY

Baldwin, B.S., MacCormack, R.W., (1975), A Numerical Method for Solving the Navier–Stokes Equations with Application to Shock–Boundary Layer Interactions. AIAA PAPER 75–1.

Beam, R.M., Warming, R.F., (1976), An Implicit Finite–Difference Algorithm for Hyperbolic Systems in Conservation Law Form. *J. Comp. Phys.*, **22**, pp. 87–110.

Engquist, B., Lötstedt, P., Sjögreen, B., (1989), Nonlinear Filters for Efficient Shock Computation. *Math. Comp.*, 52, pp. 509–537.

Erikson, J., (1972), Iterative and Direct Methods for Solving Poisson's Equation and their Adaptibility to ILLIAC IV. Center for Advanced Computation Document 60, Univ. Illinois, Urbana–Champaign.

Harten, A., Yee, H., Warming, R.E., (1983), Implicit Total Variation Diminishing Schemes for Steady–State Calculations. AIAA PAPER 83–1902.

Hayes, L., (1974), Comparitve Analysis of Iterative Techniques for Solving Laplace's Equation on the Unit Square on a Parallel Processor. M.S. Thesis, Dept., Mathematics, Univ. Texas, Austin.

Jameson, A., (1985A), Numerical Solution of the Euler Equations for Compressible Inviscid Fluids. *Numerical Methods for the Euler Equations of Fluid Dynamics, Ed. Angrand, et al.*, SIAM, 1985, pp. 199–245.

Jameson, A., (1985B), A Nonoscillatory Shock Capturing Scheme Using Flux Limited Dissipation, in: B.E. Enguist, S. Osher, and R.C.J. Somerville, Eds., *Large Scale Computations in Fluid Mechanics, Lectures in Appl. Math.*, **22**, part 1 (American Mathematical Society, Providence, RI, 1985), pp. 345–370.

Lambiotte, J., (1975), The Solution of Linear Systems of Equations on a Vector Computor. P.hd. Dissertation , Univ. Virginia, Charlottesville.

Lax, P.D., (1972), *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves.* Society of Industrial and Applied Mathematics, Philadelphia, 1972.

Nayani, S.N., (1988), A Locally Implicit Scheme for the Navier–Stokes Equations. P.hd. Dissertation, Univ. Tennessee, Knoxville.

Ortega, J.M., Voigt, R.G., (1985), Solution of Partial Differential Equations on Vector and Parallel Computers. *SIAM Rev.* **27**, pp. 149–240.

Pulliam, T.H., (1985), Artificial Dissipation Models for the Euler Equations. AIAA PAPER 85–0438 presented at the AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, 1985.

Reddy, K.C., J.L. Jacocks., (1987), A Locally Implicit Scheme for the Euler Equations. *Proceedings of the AIAA 8th Computational Fluid Dynamics Conference, Honolulu, 1987.* pp. 470–477.

Sod, G.A., (1987) A Survey of Several Finite Difference Methods for Systems of Hyperbolic Conversation Laws. *J. Comp. Phys.*, **27**, pp. 1–31.

Shubin, G.R., Stephens, A.B., Glaz, H.M., (1981), Steady Shock Tracking and Newtons Method Applied to a One–Dimensional Duct Flow. *J. Comp. Phys.*, **39**, pp. 364-374.

Towne, M.C., (1989), A Matrix–Free Locally Implicit Scheme for Navier–Stokes Equations. M.S. Thesis, Univ. Tennessee, Knoxville.

VITA

Robert Wallace Tramel was born on July 26, 1963 in McMinnville, Tennessee. He attended elementary school there and received his high school diploma from Warren County Senior High School in 1981. In the fall of 1981, he started undergraduate studies at Tennessee Technological University in Cookeville, Tennessee and was awarded a Bachelor of Science degree with honors in Physics in August 1986. In January 1988 he was admitted as a student in the Mathematics Department of The University of Tennessee Space Institute at Tullahoma, Tennessee and received his Master of Science degree in Mathematics in August 1990. When not thinking about mathematics or physics he likes to read poetry and listen to jazz music.