STATE VECTOR SPLITTING: A NUMERICAL SCHEME FOR THE EULER EQUATIONS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS ANS ASTRONAUTICS

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Hakan Öksüzoğlu

September 1992

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Robert W. MacCormack
(Principal Adviser)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Dean R. Chapman

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Donald Baganoff

Approved for the University Committee on Graduate Studies:

Abstract

A new numerical scheme was developed for the solution of the Euler equations in multi-dimensions. The scheme removes the critical dependency on the choice of a "good" grid by making use of the multidimensional character of the Euler Equations. In addition, a new procedure for implementing solid wall boundary conditions was introduced which does not require body-fitted grids. This grid independent nature of the scheme allows one to solve complicated geometry problems on simple regular grids.

The scheme uses an upwind direction which is independent of the underlying coordinate system. This direction is determined only by the local state of the fluid. The scheme is explicit and easy to implement for both serial and parallel computation.

The current implementation of the algorithm for two space dimensions can handle unsteady flows around almost arbitrary geometries using a simple cartesian mesh. Several numerical experiments have been done to show the flexibility and the robustness of the algorithm.

Acknowledgements

I wish to thank my undergraduate advisor at Istanbul Technical University, Professor Ülgen Gülçat, who gave me the idea of graduate study at Stanford in the field of Computational Fluid Dynamics. I am indebted to my thesis advisor , Professor Robert W. MacCormack, for his support and encouragement. He and Professor Dean R. Chapman have provided a stimulating environment for academic research. I also wish to thank Professor Donald Baganoff for agreeing to read my thesis, and Professors Milton Van Dyke and Joseph E. Oliger for agreeing to be in my committee. Carl F. Gooch deserves special credit for critical discussions and reading my first drafts. Stephane Moreau helped to improve the presentation at the early stages of this work. Finally, I wish to thank my friends at Stanford without their support this work would not be possible: Kutlu Aricanli, Can Cinbiş, Birol Dindoruk, Joseph Friedman, Derek Gerlach, Tahir Gökçen, Levent Karaoğlan, İbrahim Kocabaş, Sadun Saran, Hakan Önel, Haldun Özaktaş, Ekmel Özbay, Hasan Yıldız, and many others I have not named here.

Dedicated to

my parents Aysel and Celal,

my grandparents Fevziye and Mehmet Karadeniz,

Gülizar and Mehmet Öksüzoğlu.

Contents

List of Figures

Chapter 1

Introduction

This thesis presents a new numerical method for the solution of compressible flow equations in multiple space dimensions. The variety of physical phenomena described by these equations is so rich that trying to find an "ultimate" scheme seems hopeless for the near future. The extensive literature devoted to that subject is an indication of its importance and the lack of completely satisfactory algorithms. This work concerns one important aspect of these equations: the multidimensional character of the convective terms and the implications of this multidimensionality in algorithm development.

In this chapter, first, the Euler equations describing the inviscid flow of an ideal gas will be examined. Then, the difficulties associated with multidimensional problems will be reviewed. This will provide the necessary background and motivation for the derivation of a new genuinely multidimensional upwind algorithm. For brevity, the Euler equations in two dimensions will be considered throughout the main text; generalization to three dimensional problems will be given in an appendix.

In the second chapter, the algorithm will be presented in detail, and its connection to other upwind schemes will be discussed. In the third chapter, a new implementation for solid wall boundary conditions will be presented. The last chapter will include

numerical examples for validation and showing the advantages of the scheme. Future directions for research will also be considered.

1.1 Equations of Gasdynamics

The Euler equations in two dimensions can be written in the following conservative form

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = 0 \tag{1}$$

where $\mathbf{U}$ is the state vector and $\mathbf{F}$, $\mathbf{G}$ are the flux vectors in the corresponding coordinate directions.

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{bmatrix} \tag{2}$$

where ρ is density, u is velocity in x direction, v is velocity in y direction, e is total energy per unit volume and p is pressure. By assuming that we are dealing with ideal gas, we close the equations with the following equation of state

$$p = (\gamma - 1)(e - \rho(u^2 + v^2)/2) \tag{3}$$

For the derivation of these equations representing mass, momentum, and energy conservation, see [Hir88].

This system can also be written in the following form

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y = 0 \tag{4}$$

where $\mathbf{A}$ and $\mathbf{B}$ are called flux jacobians ($\mathbf{A} = \partial\mathbf{F}/\partial\mathbf{U}, \mathbf{B} = \partial\mathbf{G}/\partial\mathbf{U}$). The system of equations is said to be hyperbolic if the matrix

$$\mathbf{P} = k_1\mathbf{A} + k_2\mathbf{B} \tag{5}$$

has real eigenvalues and linearly independent eigenvectors for any set of real numbers k_1, k_2. In order to see what this definition means, let us assume that flux jacobians are constant matrices, and look for a solution of the form

$$\mathbf{U} = \mathbf{R}\sigma(k_1 x + k_2 y - \lambda t) \tag{6}$$

where $\mathbf{R}$ is a vector in a state space and σ is a scalar function. Here, $\mathbf{R}$, a vector in state space, represents a wave propagating in the $\mathbf{k} = [k_1, k_2]$ direction, in physical space, with the wave speed λ. The magnitude of $\mathbf{k}$ is arbitrary, and can be taken as unity for convenience. Substituting this expression for $\mathbf{U}$ and taking derivatives gives the following eigenvalue problem

$$(k_1\mathbf{A} + k_2\mathbf{B})\mathbf{R} = \lambda\mathbf{R} \tag{7}$$

This shows that we have a solution if $\mathbf{R}$ is an eigenvector and λ is an eigenvalue of matrix $\mathbf{P}$. Since the equation is assumed to be linear, a linear combination of these eigenvectors for all possible real k_1, k_2 is also a solution. Thus, if we can find a unique decomposition of the initial conditions in terms of these eigenvectors, we can easily construct the general solution. This is called the method of decomposition into plane waves [CH62].

For the Euler equations in 2D, there are four eigenvalues(three distinct) for each direction. These are

$$\lambda_1 = q + c\ , \quad \lambda_2 = q - c\ , \quad \lambda_3 = \lambda_4 = q \tag{8}$$

where c is the speed of sound and q is the fluid velocity component in the direction of $\mathbf{k}$, $q = uk_1 + vk_2$ [SW81].

From this much information, it is not obvious what the domain of influence (or dependence) would look like. For a detailed presentation of the subject, see [Hir90] [CH62]. A plausibility argument for finding the domain of influence can be given as follows.

Let us consider a family of plane waves passing from the origin(Figure 1). There will be three distinct waves propagating in each direction: one moving with the velocity component in that direction and two acoustic waves. Acoustic waves will form an envelope; they are all tangent to a circle. This circular region will be a subset of the domain of influence for the plane waves passing from the given point irrespective of the wave direction. This suggests that this circle covers the domain of influence for the given point in space.

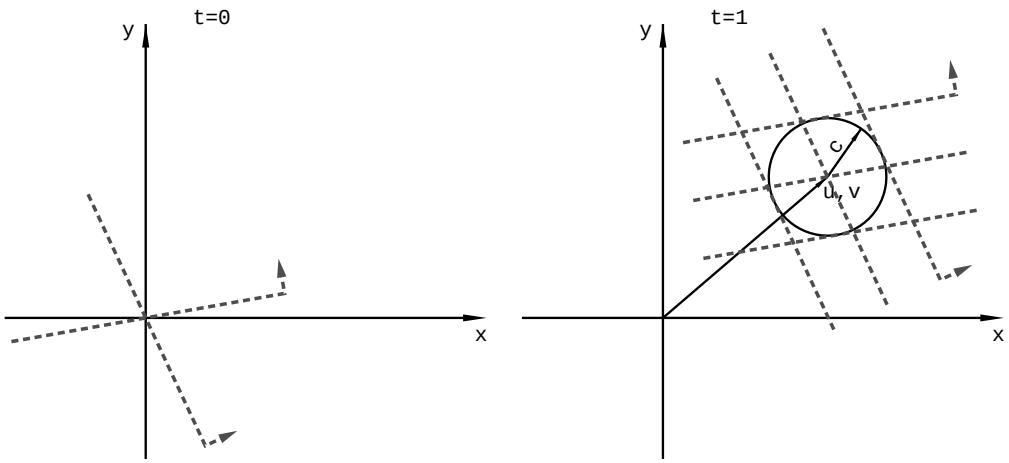

Figure 1: Plane Waves and the Domain of Influence

An important property of the domain of influence is its coordinate independent orientation. This is expected because the choice of the coordinate system is arbitrary and has nothing to do with the physics. The domain of influence for a given point is only determined by the fluid state at this point. It is this property that will be taken into account for the design of a new genuinely multidimensional upwind scheme.

1.2 Multidimensional Problems

Numerical computation of compressible flows around complicated geometries has been a challenging problem for the following reasons.

First, a solution may have discontinuities whose position is not known a priori. These so called "weak solutions" are difficult to handle; simply replacing derivatives with approximations based on Taylor series does not work. This is because these non-smooth solutions are not solutions in the classical sense; they satisfy an integral conservation law whose solutions need not be differentiable. For one dimensional problems, there are very sophisticated schemes for dealing with this difficulty. However, the situation is much more complicated in multidimensions due to the complex shapes and interactions of these discontinuities.

Second, it may be necessary to generate elaborate grids (body fitted or even shock fitted) in order to obtain reasonable solutions. Modern CFD codes are difficult to use because of the need of user intervention to choose a "good" grid. Although very sophisticated tools are available for generating grids, generating a grid or modifying the code to fit the specific geometry at hand is very time consuming. In addition, the quality of the solution can critically depend upon the grid. Many researchers believe that this grid dependency can be removed by taking advantage of the multidimensional nature of the Euler Equations [Roe86a].

One example of the problems related to grid dependency can be found in [PV92]. This so called singular line boundary problem is due to the interaction of inherently one dimensional schemes with grid singularities. The treatment of this problem requires either the generation of more sophisticated grids or a special care in those singular lines.

Once we have a genuinely multidimensional upwind scheme, we can use grids which are easy to generate without human intervention. Two promising paths in that

direction are the use of unstructured grids which can be generated by computer and the use of simple cartesian grids which need not fit boundaries or discontinuities. In this work, a multidimensional upwind scheme will be implemented by using simple cartesian grids.

The use of cartesian grids for complicated geometries has been difficult for two main reasons. First, boundary conditions are hard to enforce when wall boundaries do not coincide with grid lines. Second, most modern schemes are straightforward generalizations of inherently one dimensional schemes, and typically require the grid to fit discontinuities.

In generalizing to multi-dimensions, it is assumed that the problem can be treated as a combination of one dimensional problems in each coordinate direction . A crucial assumption is made without justification: waves propagate along coordinate lines (Finite Difference) or normal to cell interfaces (Finite Volume). This assumption is not critical, however, if the region of rapid variation, such as shock waves, is aligned with one of the coordinate lines. This straightforward generalization can lead to either more restrictive stability bounds in multi-dimensions or degraded accuracy. Currently, there is considerable interest in developing genuinely multidimensional upwind schemes. For a critical discussion of these issues, see the review paper by Roe [Roe86a] and a recent monograph by LeVeque [LeV90].

In his exploratory paper [Roe86b], Roe suggested looking at the local gradients to choose the upwind direction. The essential feature of his model is the grid independent character of the directions in which information can propagate. Similar approaches based on looking at the local gradients for multidimensional upwinding have been employed by many authors [BL91] [DG91] [Dav84].

In this thesis, a new multidimensional upwind scheme which uses only local data rather than its gradients for choosing the upwind direction is presented[1]. This simple

[1]Major results of this work have been given in [Öksüzoğlu92].

approach requires less computation and avoids anomalies like indeterminate upwind direction due to zero gradients. Also, it it not obvious which gradient to look after, and one more arbitrary decision has to be made if we use a gradient based upwinding.

Chapter 2

State Vector Splitting

The importance of using the information about the domain of dependence for the numerical solution of hyperbolic differential equations has been known for a long time. In 1928, Courant *et al.* introduced the so called CFL condition which stated that the numerical domain of dependence should include the physical domain of dependence for stability. The earliest upwind scheme making use of the information about domain of dependence was given in [CIR52]. However, this scheme was not conservative and was not successful for discontinuous solutions. In this chapter, we will develop a conservative upwind scheme which will utilize the domain of influence rather than dependence. More detailed information on upwind schemes can be found in review papers by Roe [Roe86a], Harten *et al.* [HLL83] and monograph by LeVeque [LeV90]. A very good introduction to the subject is given in the recent text by Hirsch [Hir90].

The development of upwind schemes has been in the context of hyperbolic differential equations in one space dimension. The generalization into multidimensional problems was done in a straightforward way without making use of the information about the multidimensional domain of influence. Problems related to that approach was reported recently by Roe [Roe86a]. Currently, there is a growing interest in understanding the multidimensional wave structure of the Euler equations and using

that information for developing genuinely multidimensional upwind schemes.

In this chapter, we begin with a review of Flux Vector Splitting as proposed by Steger and Warming [SW81]. A different interpretation of this scheme will be the starting point for the development of the new scheme. After introducing the scheme (in fact, a family of schemes) in its most general form, possible ways of implementing it will be discussed.

2.1 Flux Vector Splitting

Consider the following hyperbolic system in one space dimension

$$\mathbf{U}_t + \mathbf{F}_x = 0 \tag{9}$$

If the flux vector is a homogeneous function of the state vector, $\alpha\mathbf{F}(\mathbf{U}) \equiv \mathbf{F}(\alpha\mathbf{U})$, then the following identity holds.

$$\mathbf{F} \equiv \mathbf{A}\mathbf{U} \tag{10}$$

where $\mathbf{A} = \partial\mathbf{F}/\partial\mathbf{U}$. It is easy to check that this is true for the Euler equations

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix} \tag{11}$$

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (e+p)u \end{bmatrix} \tag{12}$$

.

If we can diagonalize the matrix $\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$, we can write the flux vector as the sum of two vectors

$$\mathbf{F} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}\mathbf{U} = \mathbf{S}\mathbf{\Lambda}^{+}\mathbf{S}^{-1}\mathbf{U} + \mathbf{S}\mathbf{\Lambda}^{-}\mathbf{S}^{-1}\mathbf{U} = \mathbf{F}^{+} + \mathbf{F}^{-} \tag{13}$$

where $\mathbf{\Lambda}^+, \mathbf{\Lambda}^-$ contain the positive and the negative eigenvalues of the flux jacobian, respectively. This splits the flux into vectors associated with the positive and negative wave speeds.

$$\mathbf{U}_t + (\mathbf{F}^+ + \mathbf{F}^-)_x = 0 \tag{14}$$

The idea of the Steger-Warming scheme is to apply the forward difference operator to the negative wave speeds and the backward difference operator to the positive wave speeds; that is, to upwind difference both kinds of waves.

$$\frac{\delta \mathbf{U}}{\Delta t} + \frac{D^- \mathbf{F}^+}{\Delta x} + \frac{D^+ \mathbf{F}^-}{\Delta x} = 0 \tag{15}$$

where

$$\delta \mathbf{U} = \mathbf{U}^{n+1} - \mathbf{U}^n, D^- \mathbf{F} = \mathbf{F}_i - \mathbf{F}_{i-1}, D^+ \mathbf{F} = \mathbf{F}_{i+1} - \mathbf{F}_i$$

2.2 The Method of Decomposition into Plane Waves

In the following sections, we will construct an upwind scheme from a different point of view and show that this is equivalent to the Flux Vector Splitting of Steger-Warming. However, this interpretation will be helpful in generalization to multidimensional problems.

Let us assume that the jacobian matrix is constant (or frozen for a given solution) and look for the solutions of the form $\mathbf{U} = \mathbf{R}\sigma(x - \lambda t)$. If we substitute this solution into the equation

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U}_o)\mathbf{U}_x = 0 \tag{16}$$

we obtain the following eigenvalue problem.

$$\mathbf{AR} = \lambda \mathbf{R} \tag{17}$$

Let $\mathbf{R}_i(\mathbf{U}_o)$ and $\lambda_i(\mathbf{U}_o)$ be eigenvectors and eigenvalues of matrix $\mathbf{A}$. Then, a general solution can be written as

$$\mathbf{U} = \sum_i \mathbf{R}_i \sigma_i (x - \lambda_i t) \tag{18}$$

The magnitudes of the waves will be obtained from the initial conditions.

If we want to apply this method to a nonlinear equation to find an approximate solution, first, we assume that the initial data can be represented by a piecewise constant function on a given grid. Then for each cell, we find the eigenvalues and eigenvectors for the local jacobian. Then, we use the above solution formula to obtain an approximation for short times. In order to continue this process, at the end of the short time interval, we have to average the solution so that we again have a piecewise constant function.

2.3 State Vector Splitting in One Space Dimension

In this section, we will use the method of decomposition into plane waves to construct a numerical scheme, and later, try to give another interpretation by using ideas from a molecular point of view. This new interpretation will be the basis for the genuinely multidimensional upwind scheme.

In one space dimension, the eigenvalues of the flux jacobian (wave speeds) for a given state are

$$\alpha_1 = u - c \tag{19}$$

$$\alpha_2 = u \tag{20}$$

$$\alpha_3 = u + c \tag{21}$$

and the state vector can be decomposed into eigenvectors of the flux jacobian

$$\mathbf{R}_1 = \frac{\rho}{2\gamma}\begin{bmatrix} 1 \\ u-c \\ h-uc \end{bmatrix} \tag{22}$$

$$\mathbf{R}_2 = \frac{\rho(\gamma-1)}{\gamma}\begin{bmatrix} 1 \\ u \\ u^2/2 \end{bmatrix} \tag{23}$$

$$\mathbf{R}_3 = \frac{\rho}{2\gamma}\begin{bmatrix} 1 \\ u+c \\ h+uc \end{bmatrix} \tag{24}$$

where

$$h = (e+p)/\rho \tag{25}$$

is the specific enthalpy, and

$$c = \sqrt{\gamma p/\rho} \tag{26}$$

is the speed of sound.

The eigenvectors has been scaled such that the following condition is satisfied

$$\mathbf{R}_1 + \mathbf{R}_2 + \mathbf{R}_3 = \mathbf{U} \tag{27}$$

representing the splitting of the state vector. It is interesting to note that the eigenvalues and the eigenvectors also satisfy

$$\alpha_1\mathbf{R}_1 + \alpha_2\mathbf{R}_2 + \alpha_3\mathbf{R}_3 = \mathbf{F} \tag{28}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (e+p)u \end{bmatrix}$$

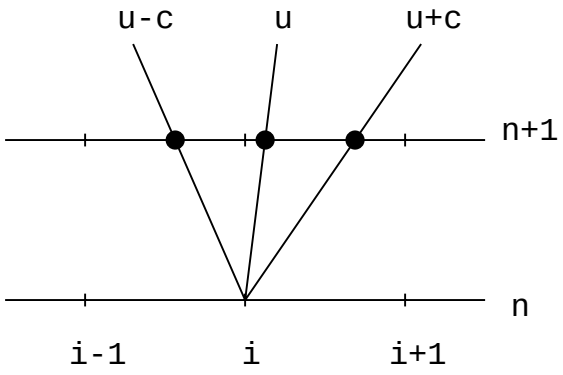

Figure 2: Wave Decomposition in One Dimension

The first stage of the numerical implementation is to approximate the initial data by a piecewise constant vector valued grid function. Then, this approximation is decomposed into three wave components and shifted according to corresponding wave speeds before combining them at the next time level (Figure 2). Of course, in order to continue this process, we have to average this approximate solution to obtain piecewise constant data on the same grid. If we use a simple averaging (Figure 3), the update formula can be expressed as

$$\begin{aligned} \mathbf{U}_i^{n+1} = &(\sum_{+}(1-c_k)\mathbf{R}_k)_i^n + (\sum_{+}(c_k)\mathbf{R}_k)_{i-1}^n \\ &+(\sum_{-}(1+c_k)\mathbf{R}_k)_i^n + (\sum_{-}(-c_k)\mathbf{R}_k)_{i+1}^n \end{aligned} \tag{29}$$

where $c_k = \alpha_k \Delta t/\Delta x$ and α_k is the kth eigenvalue of the flux jacobian. The symbols $\sum_+$, $\sum_-$ indicate summation over positive and negative eigenvalues. This type of averaging guarantees that the scheme is conservative, and positivity preserving, since we are not creating or destroying any one of the conserved variables; we are splitting them at one time level, moving them according to corresponding wave speeds, and recombining them at the next time level. This particular averaging for the given decomposition in 1D leads to a scheme which is equivalent to the original form of the Steger-Warming Flux Vector splitting [SW81]. This will be proved in the next section.

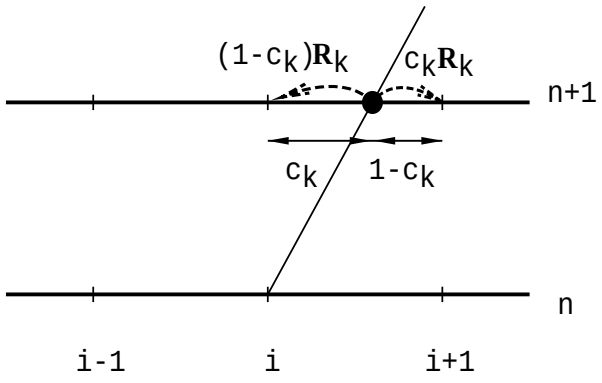

Figure 3: Averaging Stage of the algorithm

2.4 Equivalence of State and Flux Vector Splitting in One dimension

In the previous section, we have decomposed the state vector into the eigenvectors of the flux jacobian.

$$\mathbf{U} = \sum \mathbf{R}_k \tag{30}$$

By using the homogeneity of the flux vector, we can also represent the flux vector in terms of these eigenvectors.

$$\mathbf{F} = \mathbf{AU} = \sum \mathbf{AR}_k = \sum \alpha_k \mathbf{R}_k \tag{31}$$

In the derivation of the Flux Vector Splitting of Steger-Warming, positive and negative fluxes are defined according to signs of the eigenvectors of the jacobian matrix.

$$\mathbf{A} = \mathbf{S\Lambda S^{-1}} = \mathbf{S\Lambda^{+}S^{-1}} + \mathbf{S\Lambda^{-}S^{-1}} = \mathbf{A^{+}} + \mathbf{A^{-}} \tag{32}$$

where $\mathbf{\Lambda}^{+}$ has the positive eigenvalues of $\mathbf{A}$ and zeros replacing the negative ones. $\mathbf{\Lambda}^{-}$ is defined similarly with negative eigenvalues. Both $\mathbf{A}^{+}$ and $\mathbf{A}^{-}$ have the same set of eigenvectors as $\mathbf{A}$. This means that

$$\mathbf{F}^{+} = \mathbf{A}^{+}\mathbf{U} = \sum_{+} \alpha_k \mathbf{R}_k \tag{33}$$

$$\mathbf{F}^{-} = \mathbf{A}^{-}\mathbf{U} = \sum_{-} \alpha_k \mathbf{R}_k \tag{34}$$

Now, we can prove that equations 15 and 29 are equivalent. Equation 29 can be rearranged in the following form

$$\begin{aligned}\mathbf{U}_i^{n+1} = (\sum_{+} \mathbf{R}_k)_i^n + (\sum_{-} \mathbf{R}_k)_i^n - (\sum_{+} c_k\mathbf{R}_k)_i^n + (\sum_{+} c_k\mathbf{R}_k)_{i-1}^n \\ +(\sum_{-} c_k\mathbf{R}_k)_i^n - (\sum_{-} c_k\mathbf{R}_k)_{i+1}^n\end{aligned} \tag{35}$$

The first two terms add up to the state vector at the time level n at the grid point i, $\mathbf{U}_i^n$. If we substitute $c_k = \alpha_k \Delta t/\Delta x$ and use the definitions of positive and negative fluxes, we obtain

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}(\mathbf{F}^{+}{}_i^n - \mathbf{F}^{+}{}_{i-1}^n) - \frac{\Delta t}{\Delta x}(\mathbf{F}^{-}{}_{i+1}^n - \mathbf{F}^{-}{}_i^n) \tag{36}$$

Collecting all the terms on the left hand side and dividing by Δt gives equation 15.

The difference between state and flux vector splittings in one dimension is just the order of arithmetic evaluation. State vector splitting operates on the state vector and never explicitly computes the flux vector. For the given type of splitting and averaging the two schemes give the identical results if we use an exact arithmetic. This also means that the new scheme is first order accurate in space and time.

2.5 Particle Interpretation

We can give another interpretation to the scheme by using analogies from the molecular point of view. In this section, we will attempt to form a basis for upwinding by using ideas from the kinetic theory. This approach was originally introduced by Sanders and Prendergast [SP74] under the name of "the beam scheme". The relation of the flux vector splitting to this beam scheme was pointed out by Harten *et al* [HLL83]. Harten used the name "Boltzmann-type" schemes for this class of algorithms, because the derivation of the scheme is based on a collisionless Boltzmann equation. The fundamental assumption of the beam scheme is that the velocity distribution function is discrete[1] rather than continuous[2]. That means particles can have only certain speeds as a function of space and time. This is different from the "discrete velocity models" of Carleman or Broadwell where particles can have only fixed velocities assigned a priori. Godunov and Sultangazin [GS71] has a detailed

[1] a combination of delta functions

[2] Maxwellian or Gaussian

review of these and related models. See also Platkowski and Illner [PI88] for a recent review of the subject which has attracted the interest of mathematicians.

Sanders and Prendergast used a scalar velocity distribution function to arrive at the beam scheme. In contrast, Harten used a vector valued distribution function to derive the flux vector vector splitting from a Boltzmann equation. Our interpretation will be closer to the first approach, which has a physical motivation.

Let us assume that there are only three types of particles at a given point in space and time. Each particle group is identified by its velocity. In our case, one group moves with the fluid velocity u and the other two groups move with $u + c$ and $u - c$. These are the eigenvalues of the flux jacobian. The eigenvectors represent the fraction of mass, momentum and energy carried by the each group of particles. The group number 2 has mass $m = \rho(\gamma - 1)/\gamma$ per unit volume. Momentum is just mass times velocity, which gives the second term in vector $\mathbf{R_2}$. We assume that this group of particles has no internal degrees of freedom and the energy is purely kinetic, which gives the last term. The groups 1 and 3 have $m = \rho/2\gamma$ amount of mass. Their momentum is computed similarly. These two groups also have internal degrees of freedom in addition to the translational degree of freedom in the x direction. Translational degrees of freedom in y and z directions will be considered as internal modes for one dimensional splitting. Therefore, these two groups have kinetic energy plus internal energy. From statistical mechanics, we know that the energy in an equilibrium state is equally divided among each degree of freedom. This is called the principle of equipartition of energy [VK65]. According to this principle, each mode has an average energy of $\frac{1}{2}kT$ per molecule where k is the Boltzmann constant and T is the temperature of the gas. If gas has ξ degrees of freedom, the average molecular energy per unit volume is

$$\frac{\xi}{2}\rho RT = \frac{\xi}{2}p \tag{37}$$

where

$$\xi = \frac{2}{\gamma - 1} \tag{38}$$

or

$$\gamma = \frac{\xi + 2}{\xi} \tag{39}$$

Since we are accounting for the degree of freedom in x direction, we have $\xi - 1$ internal degrees of freedom. We divide the internal energy equally between group 1 and 3. Thus the energy carried by the group 3 as a combination of kinetic and internal energies

$$\frac{\rho}{2\gamma}\frac{(u+c)^2}{2} + \frac{1}{2}\frac{\xi - 1}{2}\rho RT$$

using $c^2 = \gamma RT$

$$\frac{\rho}{2\gamma}(\frac{u^2}{2} + uc + \frac{\xi}{2}c^2)$$

We know that

$$e = \frac{p}{\gamma - 1} + \frac{\rho u^2}{2} \tag{40}$$

$$h = \frac{e+p}{\rho} = \frac{u^2}{2} + \frac{c^2}{\gamma - 1} \tag{41}$$

We can write the energy of this group

$$\frac{\rho}{2\gamma}(h + uc)$$

Similarly, we can find the energy carried by the group 1.

To sum up,

$$\mathbf{R}_3 = \frac{\rho}{2\gamma}\begin{bmatrix} 1 \\ u+c \\ h+uc \end{bmatrix}$$

represents the mass, momentum and energy carried by the group of particles moving with the speed $u + c$. The last entry includes the kinetic energy of the group plus half of the internal energy per unit volume. Similar things can be said for $\mathbf{R}_1$ and $\mathbf{R}_2$,

except that $\mathbf{R}_2$ does not have any internal energy. The second half of the internal energy is carried by $\mathbf{R}_1$.

2.6 Positivity

One important property which a robust numerical scheme should posses is "positivity"; that is the thermodynamic quantities should be kept within their physical limits. For example, density or pressure should never be negative. This is important because trying to replace these negative quantities with positive values will violate the conservation property of the scheme and cause instabilities. In this section, we will show that our scheme for the given type of averaging preserves the positivity of density and pressure.

It is trivial to check that density will always be positive if we start with a positive density distribution and satisfy a CFL condition. Update formula 29 has always positive coefficients if

$$|c_k| < 1$$

If each particle group has a positive fraction of density which is the case for our splitting, than the resulting state vector at the next time level will have a positive density. The same argument holds for total energy, too. The only thing we need to assure is that each group has a positive fraction of the total energy.

$$h \pm uc > 0$$

where

$$h = \frac{c^2}{\gamma - 1} + \frac{u^2}{2}$$

After some manipulation

$$\frac{3-\gamma}{\gamma-1}c^2 + (u \pm c)^2 > 0 \tag{42}$$

From that expression, we can conclude that energy of each group is positive if

$$1 < \gamma < 3 \tag{43}$$

The kinetic theory predicts that

$$1 < \gamma < \frac{5}{3} \tag{44}$$

so each group has a positive fraction of the energy.

Showing the positivity of pressure or internal energy is not difficult, either. Let ρ_m denote the mass of the particle packet[3] m reaching the grid point i at time level $n+1$. These packets may originate from grid points $i-1, i, i+1$ at time level n, and may be associated with different wave speeds (index k). Let α_m designate the speed of the packet m. If we sum all the masses reaching the grid point, we obtain the density[4] of the fluid at that point at the next time level

$$\rho = \sum \rho_m \tag{45}$$

Momentum of the each packet is equal to its density times its velocity. Summation gives the fluid momentum

$$M = \sum \rho_m \alpha_m \tag{46}$$

The total energy of the fluid will consist of the sum of kinetic and internal energies of each packet.

$$e = \sum (\frac{1}{2}\rho_m \alpha_m^2 + \varepsilon_m) \tag{47}$$

where ε_m represents the internal energy carried by packet m. Kinetic energy of the fluid can be computed using

$$KE = \frac{M^2}{2\rho} \tag{48}$$

[3]Packet is the fraction of a group after the averaging stage. For example, $(1-c_k)\mathbf{R}_k$ represents a packet.

[4]Here we are actually talking about the mass which is an additive quantity. In fact, everything is multiplied by the cell volume which is a constant number for a uniform mesh.

We are trying to show that internal energy of the fluid is positive

$$e - KE \geq 0 \tag{49}$$

or

$$\sum(\frac{1}{2}\rho_m\alpha_m^2 + \varepsilon_m) \geq \frac{(\sum \rho_m\alpha_m)^2}{2\sum \rho_m}$$

Assuming that we have a positive internal energy distribution initially, and using the fact that density is always positive, if we can show the following inequality holds, we can also show that internal energy (or pressure) is positive.

$$\sum \rho_m \sum \rho_m\alpha_m^2 + 2\sum \varepsilon_m \sum \rho_m \geq (\sum \rho_m\alpha_m)^2 \tag{50}$$

We know that all ρ_m and $\rho_m\alpha_m^2$ are positive quantities, so they can be written as a square of a number

$$\sum \rho_m \sum \rho_m\alpha_m^2 = \sum(\sqrt{\rho_m})^2 \sum(\sqrt{\rho_m\alpha_m^2})^2$$

and we can use Cauchy-Schwarz inequality to show

$$\sum(\sqrt{\rho_m})^2 \sum(\sqrt{\rho_m\alpha_m^2})^2 \geq (\sum \sqrt{\rho_m}\sqrt{\rho_m\alpha_m^2})^2 = (\sum |\rho_m\alpha_m|)^2 \geq (\sum \rho_m\alpha_m)^2$$

Thus, we proved that

$$\sum \rho_m \sum \rho_m\alpha_m^2 \geq (\sum \rho_m\alpha_m)^2$$

Knowing that particles have positive internal energy to begin with ($\epsilon_m \geq 0$), this implies that the inequality 50 holds, i.e. pressure stays positive.

2.7 Two Dimensional Flux Vector Splitting

Steger and Warming gave also the two dimensional extension of flux vector splitting in their paper [SW81]. This straightforward extension uses one dimensional splitting

in each coordinate direction. Here, we will give a particle interpretation for this kind of splitting and examine under what conditions it preserves the positivity of density.

The update formula for 2D flux vector splitting can be given in the following compact form

$$\frac{\delta \mathbf{U}}{\Delta t} + \frac{D_x^- \mathbf{F}^+}{\Delta x} + \frac{D_x^+ \mathbf{F}^-}{\Delta x} + \frac{D_y^- \mathbf{G}^+}{\Delta y} + \frac{D_y^+ \mathbf{G}^-}{\Delta y} = 0 \tag{51}$$

where we used the forward and backward difference operators in x and y directions. Let $\mathbf{R}_k$ and $\mathbf{Q}_k$ be the eigenvectors of the flux jacobians in x and y directions respectively. They are normalized such that

$$\mathbf{U} = \sum \mathbf{R}_k = \sum \mathbf{Q}_k \tag{52}$$

Let α_k and β_k be the corresponding eigenvalues used in the following definitions

$$c_k = \frac{\alpha_k \Delta t}{\Delta x} \tag{53}$$

$$d_k = \frac{\beta_k \Delta t}{\Delta y} \tag{54}$$

Using the decomposition of the flux vectors in terms of these eigenvectors and eigenvalues, we can write the update formula 51 as follows

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^n - \frac{\Delta t}{\Delta x}\left((\sum_+ \alpha_k \mathbf{R}_k)_{i,j}^n - (\sum_+ \alpha_k \mathbf{R}_k)_{i-1,j}^n - (\sum_- \alpha_k \mathbf{R}_k)_{i,j}^n + (\sum_- \alpha_k \mathbf{R}_k)_{i+1,j}^n \right)$$
$$-\frac{\Delta t}{\Delta y}\left((\sum_+ \beta_k \mathbf{Q}_k)_{i,j}^n - (\sum_+ \beta_k \mathbf{Q}_k)_{i,j-1}^n - (\sum_- \beta_k \mathbf{Q}_k)_{i,j}^n + (\sum_- \beta_k \mathbf{Q}_k)_{i,j+1}^n \right)$$

adding and subtracting $2\mathbf{U}_{i,j}^n$ and substituting the definitions of c_k and d_k, we obtain

$$\begin{aligned} \mathbf{U}_{i,j}^{n+1} = -\mathbf{U}_{i,j}^n &+ (\sum_+ (1 - c_k)\mathbf{R}_k)_{i,j}^n + (\sum_+ c_k \mathbf{R}_k)_{i-1,j}^n + \\ &(\sum_- (1 + c_k)\mathbf{R}_k)_{i,j}^n + (\sum_- -c_k \mathbf{R}_k)_{i+1,j}^n + \\ &(\sum_+ (1 - d_k)\mathbf{Q}_k)_{i,j}^n + (\sum_+ d_k \mathbf{Q}_k)_{i,j-1}^n + \\ &(\sum_- (1 + d_k)\mathbf{Q}_k)_{i,j}^n + (\sum_- -d_k \mathbf{Q}_k)_{i,j+1}^n \end{aligned} \tag{55}$$

By looking at this equation[5], we can see the following interpretation. At each grid point, we have a given amount of mass, momentum and energy for time step n, stored as the state vector $\mathbf{U}_{i,j}^n$. First, we assume that particles move only in the x direction with the corresponding eigenvalues, and use the averaging as in the one dimensional case. Then, we do the same thing for the y direction. Since we have doubled the amount of mass, momentum and energy by sending the state vector in x and y directions, we subtract this additional state vector from each grid point giving the first term on the right hand side.

In order to preserve the positivity of density, more mass than is already there should come back after the averaging state, because we are subtracting the amount which was there. This can be guaranteed if we satisfy the following restriction on the time step.

$$\Delta t < \frac{1}{\frac{|u|+c}{\Delta x} + \frac{|v|+c}{\Delta y}}$$

or

$$1 > max(c_k) + max(d_k) \tag{56}$$

This is the recommended time step for 2D problems on empirical basis and is more restrictive than its one dimensional counterpart.

2.8 Multidimensional Upwinding

So far, implementation of the new scheme in one space dimension has been described. It was also shown that for the given splitting of the state vector and the given averaging, we obtain a scheme equivalent to flux splitting of Steger-Warming. However, it is clear that neither the splitting nor the averaging is unique for the algorithm. In this section, we will present it in its general form and generalize it to multiple space dimension.

[5] compare with the update formula 29 in one dimensional case

We have two different interpretations of the scheme in one space dimension. One is based on the molecular point of view, the other is based on the idea of wave propagation. As we have already mentioned many attempts were made to design a multidimensional upwind algorithm by using the idea of wave propagation. This approach has one intrinsic difficulty: in 1d we have only a finite number of waves to represent a solution, in multiple dimensions, on the other hand, we have infinitely many directions and consequently infinitely many waves. Most multidimensional upwind schemes to date tried to decide which directions are important by looking at the local gradients. This resulted in schemes which apply some sort of splitting in the direction of steepest gradient[6]. This decision is arbitrary, and brings an additional cost[7]. We will try to use the particle point of view to construct a multidimensional upwind scheme which does not require the evaluation of gradients and is easy to implement.

We will assume that we have a finite number of particle types identified by their velocity. We will let those particles move in a deterministic way between grid points according to their velocities. Of course, in order to simulate a fluid described by the Euler equations, our choice of particle velocities and fractions of particles having a certain velocity cannot be totally arbitrary. We will argue that if we impose the following conditions based on physical considerations, we will arrive at consistent finite difference schemes with additional conditions on averaging.

Let $\mathbf{R}_k$ represent the fraction of mass, momentum and energy carried by the group k in a given computational cell, and $[\alpha_k, \beta_k]$ be the velocity of the group. The first condition states that if we sum up mass, momentum and energy of each group, we should obtain the fluid mass, momentum and energy for the given cell.

$$\mathbf{U} = \sum_k \mathbf{R}_k \tag{57}$$

[6] Usually pressure

[7] evaluation of gradients

The second condition means if we multiply the mass, momentum and energy with the particle velocity we should get the flux associated with the given state.

$$[\mathbf{F}, \mathbf{G}] = \sum_k [\alpha_k, \beta_k] \mathbf{R}_k \tag{58}$$

We will also require that the momentum of the particle should be consistent with its velocity and mass. These conditions altogether will be called *consistency conditions*. Here, $\mathbf{R}_k$ and $[\alpha_k, \beta_k]$ are not necessarily the eigenvectors and the eigenvalues of the flux jacobians. These conditions are a heuristic generalization of the observations made in one dimensional case from a particle point of view.

In one space dimension, we have six equations, so we need at least two vectors[8] to satisfy these conditions. In 2D, we have twelve equations, so three vectors[9] will be sufficient. In other words, at least two types of particles in 1d, and three types of particles in 2d will be necessary for consistency.

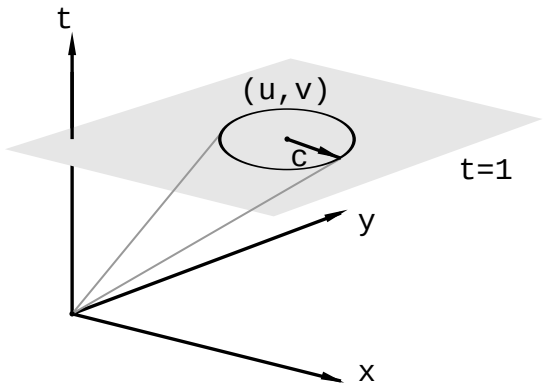

Figure 4: Domain of Influence in 2D

In the one dimensional splitting we have given, there are three particles. The

[8] with three components each
[9] with four components each

velocities of these particles coincide with the eigenvalues of the flux jacobian so that the numerical domain of influence covers the physical domain of influence. There is one known two particle splitting leading to a consistent difference scheme. However, the velocities are such that the numerical domain of influence does not cover the physical one, and we would expect stability problems. In an appendix, we will provide this two particle splitting in addition to the three particle version of Sanders and Prendergast [SP74]. In the next section we will give a five particle splitting for two space dimensions. The particle velocities will be chosen to represent the physical domain of influence to avoid the stability problems.

2.9 Two Space Dimensions

The straightforward generalization of Steger-Warming method to multi-dimensions leads to a scheme in which there are three waves in each coordinate direction (Figure 5a). Information can propagate only in those directions between time steps, and this brings a more restrictive stability bound with the increasing number of dimensions, as seen in section 2.7.

In the two dimensional case, it is possible to obtain a consistent difference equation by using only five vectors[10]. The new scheme decomposes the initial data into five particle[11] groups in 2D, satisfying the consistency conditions, such that information can propagate in the directions dictated by the local data, as opposed to arbitrarily chosen coordinate directions (Figure 5b). One particle group moves with the center of the domain of influence and has the fluid velocity. As in the one dimensional case, its energy is only kinetic. The remaining four particles move at the corners of a square which encloses the domain of influence. The orientation of this square can be

[10] not necessarily the eigenvectors of the flux jacobians

[11] It should be possible to find a three particle decomposition, but it would not satisfy the condition on the domain of influence

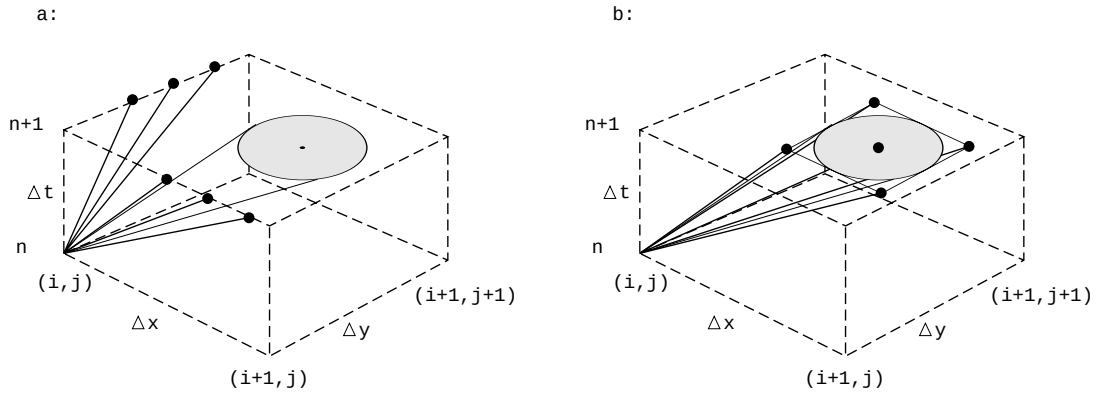

Figure 5: a) Particle Groups in Two Dimensional Steger-Warming Flux Vector Splitting; b) State Vector Splitting in 2D

arbitrary with respect to underlying coordinate system. We have chosen it such that we can take the maximum time step without violating the CFL condition. One may argue that this almost arbitrary decision makes the scheme grid dependent. However, this is the best we can do with a finite collection of particle groups. In other words, it is not possible to find a decomposition that is totally grid independent with a finite amount of computational effort. Velocities of these particle groups, and the fraction of mass, momentum, and energy they carry are given as follows.

$$\begin{array}{cc} k & [\alpha_k, \beta_k] \\ 1 & [u+c, v+c] \\ 2 & [u-c, v+c] \\ 3 & [u+c, v-c] \\ 4 & [u-c, v-c] \\ 5 & [u, v] \end{array} \tag{59}$$

$$
\begin{aligned}
\mathbf{R}_1 &= \frac{\rho}{4\gamma}\begin{bmatrix} 1 \\ u+c \\ v+c \\ h+(u+v)c \end{bmatrix} \qquad
\mathbf{R}_2 = \frac{\rho}{4\gamma}\begin{bmatrix} 1 \\ u-c \\ v+c \\ h-(u-v)c \end{bmatrix} \\
\mathbf{R}_3 &= \frac{\rho}{4\gamma}\begin{bmatrix} 1 \\ u+c \\ v-c \\ h+(u-v)c \end{bmatrix} \qquad
\mathbf{R}_4 = \frac{\rho}{4\gamma}\begin{bmatrix} 1 \\ u-c \\ v-c \\ h-(u+v)c \end{bmatrix} \\
\mathbf{R}_5 &= \frac{\rho(\gamma-1)}{\gamma}\begin{bmatrix} 1 \\ u \\ v \\ (u^2+v^2)/2 \end{bmatrix}
\end{aligned} \tag{60}
$$

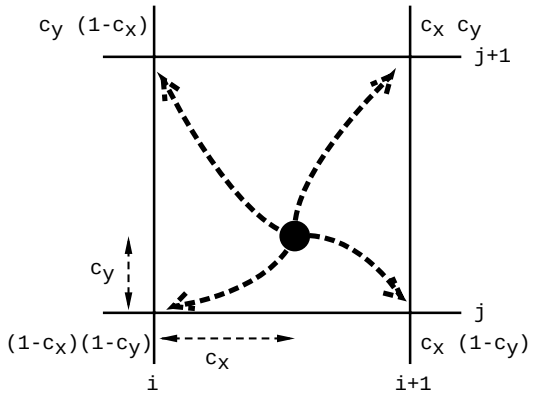

Figure 6: Averaging in 2D

This new model takes into account the multidimensional nature of the domain of influence. The waves/particles now follow the physical paths rather than an arbitrary coordinate system. If we let the particle groups propagate to the nearest neighbors, we obtain the following CFL restriction on cartesian grids

$$\Delta t < min(\frac{\Delta x}{|u| + c}, \frac{\Delta y}{|v| + c}) \tag{61}$$

This is less restrictive than the stability condition for 2D Steger-Warming, because this new scheme represents the physical domain of influence better.

The simplest averaging in 2D is the straightforward generalization of the 1D case. If a particle ends up in a cell identified by indexes i, $i+1$, j, $j+1$ and its non-dimensional coordinates with respect to the grid point i, j are (c_x, c_y) then $(1-c_x)(1-c_y)\mathbf{R}$ amount of mass, momentum and energy goes to the grid point i, j. The weight coefficients we use add up to 1. This is a condition on averaging for the conservation of physical quantities.

Chapter 3

Boundary Conditions

One of the major difficulties in numerical solution of partial differential equations for complex geometries is the implementation of numerical boundary conditions. This necessitates the use of boundary conforming grids, because it is easier to implement boundary conditions when the grid coincides with the boundary. In this chapter, we will introduce a new way of implementing the solid wall boundary conditions for the Euler equations in a grid independent way. This will allow us to solve complex geometry problems on cartesian grids.

In developing a new type of numerical boundary condition for solid walls, we will use the "particle" concept we have discussed in the previous chapter. Analogies from the kinetic theory will guide us in developing an intuitive, easy to implement and robust boundary procedure which is a natural extension of the internal scheme. We will also discuss the inflow and outflow boundary conditions from a particle point of view.

3.1 Boundary Conditions on Solid Walls: a Short Review

The meaningful boundary conditions on solid walls for the Euler equations is the flow tangency. No mass transfer means that the normal velocity at the wall should be zero[1]. This is the only condition we can impose, because there is only one characteristic entering the flow domain. This boundary condition can be stated as

$$\mathbf{q} \cdot \mathbf{n} = 0$$

where $\mathbf{q} = [u, v]$ is the velocity vector, $\mathbf{n}$ is the normal to the wall. It can be shown that this is equivalent to a condition on pressure

$$\frac{\delta p}{\delta n} = \frac{\rho v_t^2}{R}$$

where the derivative is taken normal to the wall, ρ, v_t, and R represent the density, the tangential velocity at the wall, and the radius of curvature of the wall, respectively. This equation can be discretized to determine the wall pressure for which it is easier to use body conforming grids. If the grid is not orthogonal, computing the normal derivative requires a special treatment. For more detail see [Hir90].

Another approach to solid wall boundary conditions uses fictitious cells outside the flow field to enforce the tangency condition. In this case, the fictitious cell has the same values as the nearest grid point inside the computational domain for everything except the normal velocity. Normal velocity changes sign to make it zero in between two grid points.

These procedures are difficult to apply if the boundary does not fit the grid lines. One recent paper addressing the solid wall boundary condition on cartesian grids is by Berger and LeVeque [BL90]. They show how to overcome the stability limit brought

[1] or equal to the normal velocity of the wall if it is moving

by the small cells resulting from the intersection of an arbitrary boundary with the cartesian grid. Their approach involves solving a one dimensional Riemann problem normal to the wall.

3.2 Solid Wall Boundary Conditions

In this section, we will present a new numerical boundary condition procedure for solid walls using the particle point of view. The interior scheme we introduced used this particle description to implement multidimensional upwinding. Here, we will describe how these particles behave when they encounter a solid boundary.

We know that for the Euler equations we have to preserve the tangential momentum on a solid wall since there is no friction. We also have to make sure that the normal velocity is zero[2]. We know one reflection rule that would satisfy both conditions: *specular reflection*. If a particle is reflected specularly, its tangential momentum is conserved and normal momentum changes sign. This sign change ensures that the normal velocity will be zero on the average.

The advantage of this idea is its grid independence; solid boundaries need not coincide with the grid lines. It is also conservative in the sense that no new mass or energy is produced. Non-conserved quantities like pressure do not enter the computation. A particle conserves its energy as expected, because there is no mechanism of transfer between a solid wall and the fluid for the Euler equations if the wall is stationary. However, there is a mechanism for momentum transfer in the normal direction; and, this change in the normal momentum will manifest itself as pressure on a solid boundary. So the force acting on a body can be computed by adding up of these momentum changes rather than integrating the pressure over the surface.

This reflection rule also obeys the same CFL restriction for the interior scheme:

[2] no mass flux

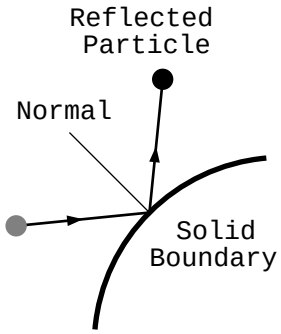

Figure 7: Specular Reflection from a Solid Boundary

if a particle is going to stay inside a certain region at the absence of a boundary because of the CFL restriction, it will also stay inside the same region even if there is boundary. So, we do not have the usual stability problem related to the small cells[3] near the boundary.

This idea of specular reflection sounds simple and intuitive but the actual implementation on a computer can be tricky. The obvious first approach of testing for each particle whether they encounter a solid wall or not during a time step can be very costly. In the next section, we will propose one method to overcome this difficulty.

3.3 Approximate Specular Reflection

One observation on specular reflection from a flat boundary will make the implementation very efficient: specular reflection is equivalent to delayed reflection for a flat wall. By delayed reflection, we mean a particle moves freely during a time step

[3]these small cells comes about because of the intersection of the boundary with the cartesian grid

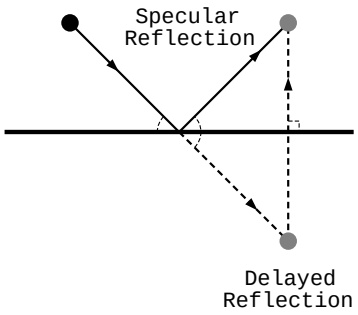

Figure 8: Delayed Reflection from a Flat Boundary

and is reflected in the normal direction to the wall if it ends up inside the boundary (Figure 8). The reflection distance is twice as much as the distance to the wall. If the boundary is not flat this approach is only an approximation to specular reflection (Figure 9). Now, we will try to show that the error introduced by this approximation is acceptable. Instead of giving a rigorous error analysis for an arbitrarily shaped wall, we will use an engineers approach (proof by example) and do it for a very special case[4].

Let us consider a cylinder of radius R (Figure 10). If a particle originates at point A and specularly reflected, it goes to point B at the end of a time step. If we let the particle move freely between time steps, it will end up at point D. In order to enforce the boundary condition, we will reflect it to the point C. The distance between point C and B, e, will be our error measure. We will find an expression for e in terms of radius R and the distance traveled by the particle after the specular reflection from the wall, h.

We will use the following vector notation: EB represents the vector whose initial

[4]praying that it is also true for more general cases

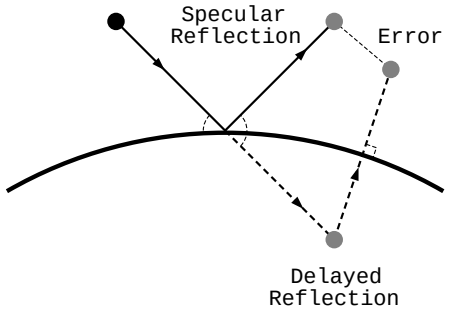

Figure 9: Delayed Reflection from a Curved Boundary

point is E and terminal point is B. The length of vector EB is

$$|EB| = h$$

Similarly

$$|OE| = R$$

$$|OD| = S$$

$$|BC| = e$$

If we use a coordinate system whose origin is located at the reflection point E on the wall and x axis is in the tangential, y axis is in the normal direction, we can write down the following expressions

$$EB = [h\sin\theta, h\cos\theta]$$

$$ED = [h\sin\theta, -h\cos\theta]$$

$$OE = [0, R]$$

$$OD = OE + ED = [h\sin\theta, R - h\cos\theta]$$

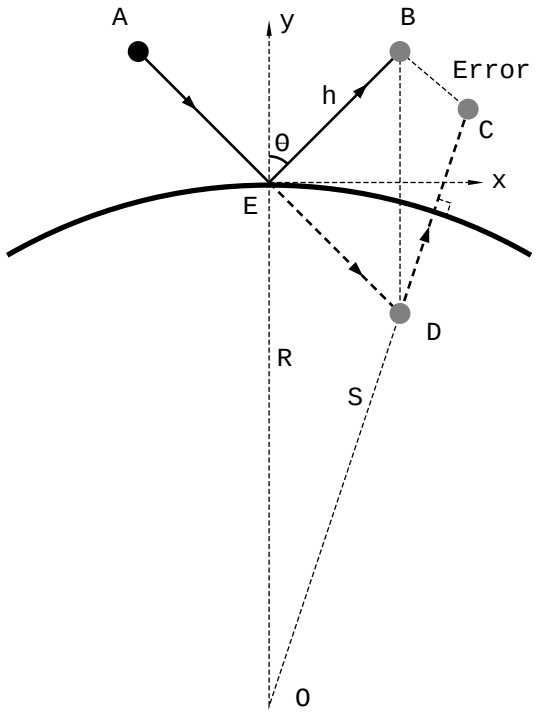

Figure 10: Error Analysis for Delayed Reflection

$$|OD| = S = \sqrt{R^2 - 2hR\cos\theta + h^2}$$

The length of OC is

$$|OC| = 2R - S$$

and it is in the direction of OD, thus

$$OC = \frac{2R - S}{S}[h\sin\theta, R - h\cos\theta]$$

Let $k = \frac{R}{S}$ then

$$\begin{aligned} EC = OC - OE &= (2k-1)[h\sin\theta, R - h\cos\theta] - [0, R] \\ &= [(2k-1)h\sin\theta, 2(k-1)R - (2k-1)h\cos\theta] \end{aligned}$$

$$BC = EC - EB = [2(k-1)h\sin\theta, 2(k-1)R - 2kh\cos\theta]$$

where

$$k^2 = \frac{R^2}{R^2 - 2hR\cos\theta + h^2} = \frac{1}{1 - 2\frac{h}{R}\cos\theta + (\frac{h}{R})^2}$$

for $\frac{h}{R} \ll 1$

$$k^2 = 1 + 2\frac{h}{R}\cos\theta + O((\frac{h}{R})^2)$$

$$k = 1 + \frac{h}{R}\cos\theta + O((\frac{h}{R})^2)$$

Thus, we obtain

$$BC = \frac{h^2}{R}[\sin 2\theta + O(\frac{h}{R}), -2\cos^2\theta + O(\frac{h}{R})]$$

and the error is

$$e = |BC| = O(\frac{h^2}{R}) \tag{62}$$

Because of the CFL restriction, we have chosen the time step such that a particle cannot move beyond the nearest grid points. That means $h = O(\Delta x)$. If $R \gg h$ then

$$e = O((\Delta x)^2)$$

This also indicates where the inaccuracy can become important: $R \ll h$, i.e. sharp corners. Although we have done this analysis for a cylinder, a similar result could be obtained for any wall whose radius of curvature is always much larger than the grid size.

3.4 Actual Implementation

In the previous section, we focused on how particles move around and did not pay attention to where there are stored. We need to use a discrete set of points to store the information about the particles in a computer. We already know how to store the particles at interior points by doing an averaging. We will combine this stage with the idea of delayed reflection of the previous section in the following manner:

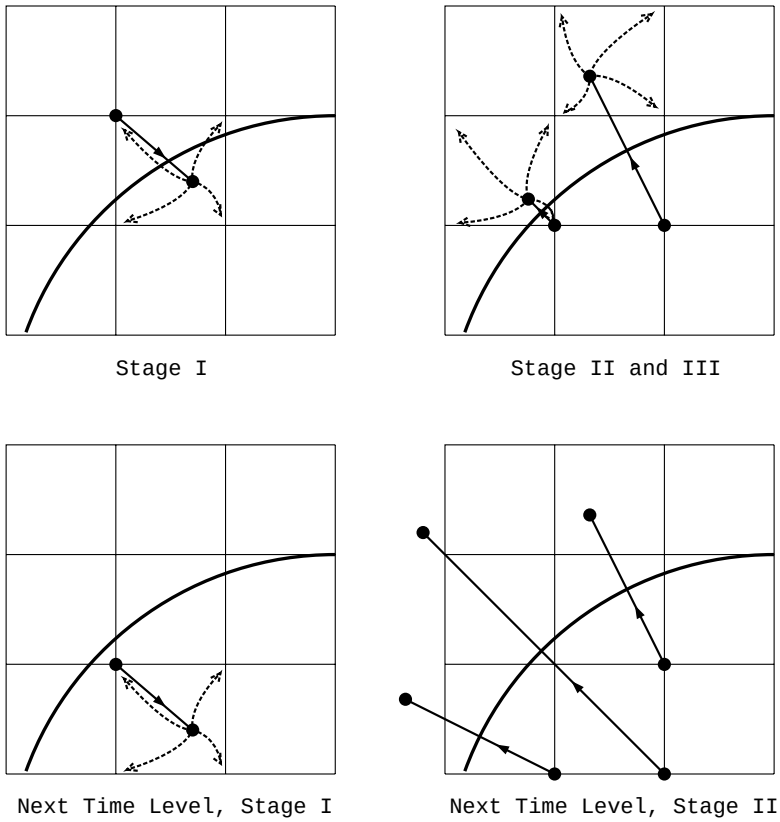

Figure 11: History of a particle group during two time steps when it is close to a solid boundary

1. Let the particles move freely and be distributed as if there is no solid boundary (interior scheme = convection + averaging)

2. Send the particles in the normal direction to the boundary, if they end up at a grid point inside the wall (delayed reflection)

3. Apply the same averaging to redistribute the reflected particles to the grid points (averaging)

In order to implement the second stage efficiently, reflection directions and distances are determined a priori and used as needed. Even after the delayed reflection and redistribution some particles will end up inside the solid wall. At the next time step,

they will penetrate further into the wall at the convection stage. So we need to consider not only the closest grid points to boundary but also the grid points behind them when determining the reflection directions and distances. At stage 2, we also have to reverse the momentum of the particle in the reflection direction.

3.5 Handling Arbitrary Geometries

By removing the grid dependency, it is possible to solve arbitrary geometries on cartesian grids. That means it is feasible to write a computer code that accepts the geometry information as data, as opposed to writing a new code or modifying the existing one for each specific geometry. In this section, we will propose a way of representing the given geometry in a computer independent of the grid resolution. The code will use this geometry representation to enforce the solid wall boundary conditions.

For a given solid body, we have to determine the reflection directions and distances at the beginning of the computation for efficiency. We can do this by writing a subroutine for each geometry. Instead, we can write subroutines for simple geometrical objects[5] and call these routines multiple times to construct more complicated objects. One such object is a convex polygon. Any geometry in two dimensions can be approximated to any desired accuracy by a set of convex polygons. By calling the subroutine for a convex polygon as many times as required to define a more complex geometry, we can handle arbitrarily shaped boundaries in 2D.

A polygon can be represented by a set of coordinate pairs. After reading the coordinates of the vertices, the subroutine checks every grid point to decide if it is inside or outside the polygon. After that, it determines whether it is a boundary point or not[6] and finds the reflection direction and distance for each boundary point.

[5] cylinder etc.

[6] grid points "close" to the boundary

There will be two types of boundary points: interior and exterior. Interior boundary points are used to store the information about the particles which goes through the solid boundary. By knowing the reflection direction and distance, we can send those particles back to the flow domain after reversing their momentum in the reflection direction. Exterior boundary points are used as regular storage points in addition to keeping the momentum reversal direction. This is necessary because the part of the computational cell which is represented by a boundary point may be inside the wall even though the point itself is outside the boundary.

In order to be able to use multiple overlapping polygons, the subroutine should not erase the information about the previous polygons. The trickiest part of the subroutine is the determination of the reflection direction for the boundary points close to polygon corners. Although it is not quite accurate, we will use some kind of average reflection direction for simplicity.

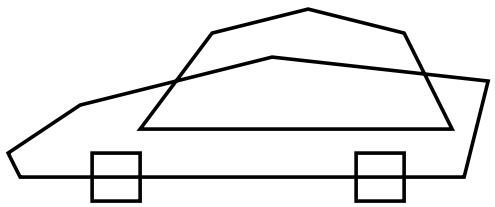

Figure 12: Multiple Convex Polygons Approximating a Non-convex Object

3.6 Inflow and Outflow Boundaries

In order to be able to solve some realistic problems, we also have to consider inflow and outflow boundaries. The theory of characteristics tells us what we can specify

at those boundaries and is also helpful in actual implementation. For a discussion of the numerical implementation of these type of boundary conditions see [Hir90].

For supersonic flow, no special treatment is necessary at the entrance and exit, because the upwind scheme itself takes care of it. For a supersonic entrance, we have to specify everything since all the characteristics enter the flow domain. This is consistent with the fact that all the particles move into the computational domain so that no influence from the domain can reach back to the boundary. For a supersonic exit, all the particles move out of the computational domain; therefore, we do not have to specify anything, consistent with the characteristics.

Subsonic boundary conditions from a particle point of view have not been considered in this thesis and left as a future research topic.

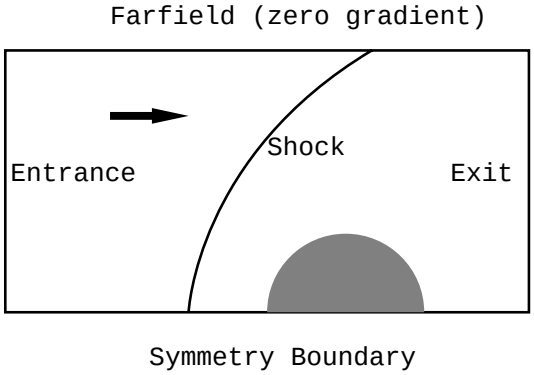

Figure 13: Far field and Symmetric Boundary Conditions

Another type of boundary condition that is going to be used in the computations is the zero gradient condition for some far field boundaries (see Figure 13). Although it is not very accurate where the oblique shock crosses the boundary, it gives satisfactory results. We also used boundary conditions exploiting symmetry in a flow. We could exploit periodicity in a similar way.

Chapter 4

Numerical Examples and Conclusion

A new numerical scheme has been introduced that is grid independent: information propagates in a grid independent way (true multidimensional upwinding) and the grid does not need to be aligned with solid boundaries. This made it possible to write a computer code that can handle arbitrary geometries on cartesian grids. This chapter will include a number of numerical examples to validate the code and the numerical scheme. All numerical solutions we present can be obtained with conventional techniques[1], however, it would be very time consuming for a person to adapt a conventional code to this variety of problems. Our less than 600 line Fortran code can handle all of these problems without a single modification, just by entering brief information on the geometry as input data.

First, we will present some steady-state solutions. These solutions were obtained by running the unsteady code for a sufficiently large number of time steps. As expected, this explicit code will not be very efficient for all steady-state problems. Of course we could improve the efficiency by using multiple grids; however, we have tried

[1] straightforward generalization of one dimensional upwind schemes for example

to keep the code as simple as possible in order not to lose the sight of our original purpose and left these further improvements for the future.

Later, we will give some time-accurate computation results. Despite the low order accuracy and the lack of grid adaptation, we can still get reasonable answers for some realistic problems by using a moderate number of grid points. Computations were done on a desktop workstation[2] and a typical runtime is on the order of a few hours for a 100×100 grid. Again increasing accuracy and taking advantage of the special structure of the solution by adapting the grid is left as possible improvements for the future. On another section, we will talk about what needs to be done to make this code a practical design tool. The final section will be a summary of what we have done and will include some concluding remarks.

4.1 Steady State Problems

4.1.1 Cylinder

The first example we have is the supersonic flow around a cylinder, the classical blunt body problem. There are many experimental and computational results for this geometry to verify our computations. The particular mach number we have chosen is 4.0. For this case, experimental data and the computational results using a body fitted grid can be found in [Zho91]. Our computations are in excellent agreement with the previous results although we have used a uniform cartesian mesh. The comparison is based on the shock position which is a good indicator of accuracy.

The computational domain has been chosen such that the entrance and the exit both have supersonic flow conditions. The far field boundary is far enough away so that the shock wave does not cross it.

[2] Sun Sparcsation 1+

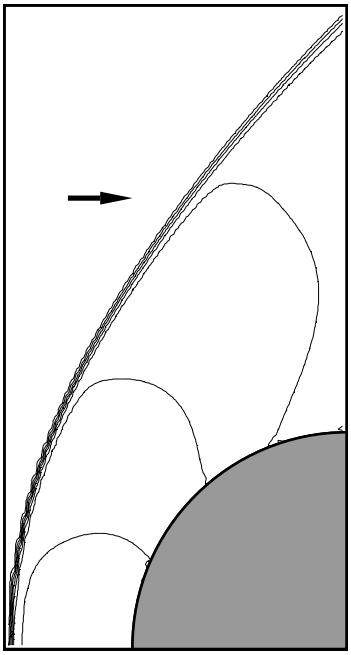

Figure 14: Mach Contours for Supersonic Flow around a Cylinder (64×120 cartesian grid, $M_\infty = 4.$).

4.1.2 Wedge

The second example is another simple geometry problem for which there is an experimental data. It is the supersonic flow around a wedge of 45^o half angle. For the freestream mach number chosen (2.5), the shock wave is detached. The experimental results can be found in [Dyk82] for comparison. The computed shock shape and position is in excellent agreement with the experimental picture.

The entrance, exit and far field boundary conditions are the same as the previous example.

4.1.3 Cylinder and Wedge

This example has been chosen to show that more complicated geometry problems

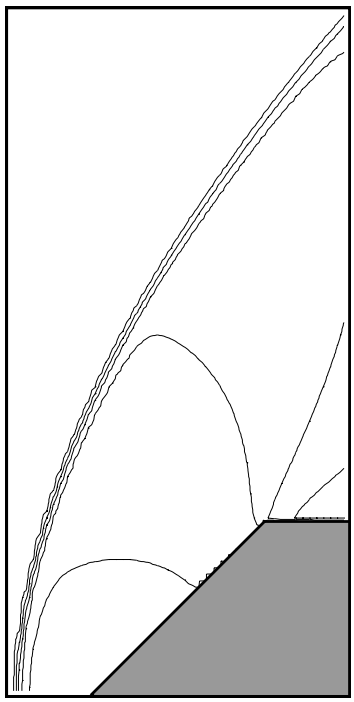

Figure 15: Mach Contours for the Wedge of 45^o half angle (50×100 cartesian grid, $M_\infty = 2.5$).

can be solved by the same code without additional difficulty. Although we do not have any experimental data for this arbitrary geometry, the results look reasonable.

4.1.4 Double Ellipse

The last steady state example is the supersonic flow around a double ellipse configuration. This realistic looking geometry has been relatively difficult problem to handle because of the nontrivial grid needed for conventional schemes. For our algorithm, it is no more difficult than the flow around a cylinder.

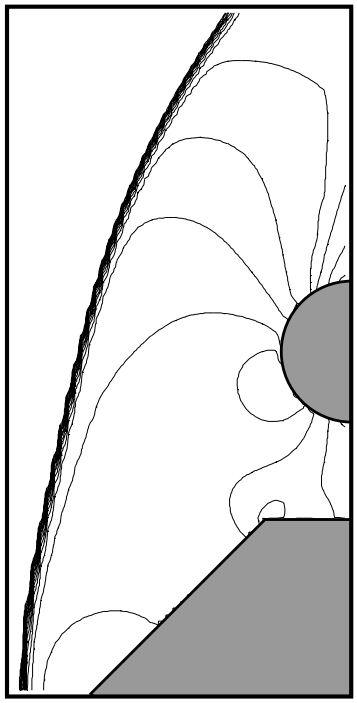

Figure 16: Mach Contours for the Wedge of 45^o half angle and a Cylinder (50×100 cartesian grid, $M_\infty = 4.$).

4.2 Unsteady Problems

4.2.1 Diffraction from a Cylinder

Again, we start with an elementary geometry problem to test our time-accurate code. In this case, a moving shock wave is diffracted from a cylinder. Experimental results can be found in [BG61].

The computed shock shape is in good agreement with the experimental picture; but, the contact discontinuity which is visible in experimental picture is smeared out in the computation due to diffusive nature of our first order accurate algorithm.

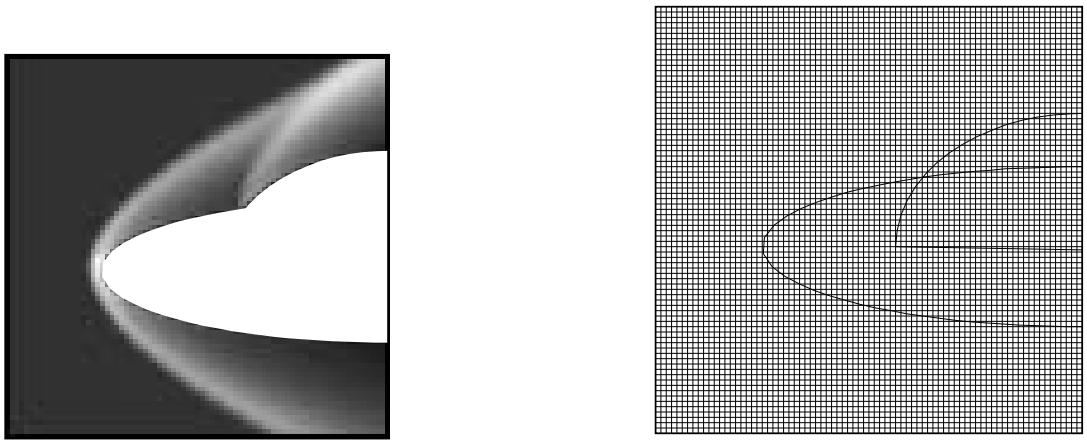

Figure 17: Density Plot for a Double Ellipse Configuration ($M_\infty = 7.$) and 80×80 cartesian grid used in the computation.

4.2.2 Shock Wave Moving in a Converging-Diverging Tunnel

The last example is an unsteady flow inside a tunnel (Figure 19). Initially, the fluid inside the tunnel is at rest. A shock wave moves from left to right as in the previous example. For this case the shock wave mach number[3] $M_o = 2$. We have density contours for two different times. In the first plot, we see mach reflections from both the circular arc and the wedge. Once again, the contact discontinuities are not visible. In the second plot, two reflected shock waves interact with each other while the main shock is moving down stream.

[3]Shock wave much number is defined as the ratio of the shock speed to the speed of sound of the stationary medium.

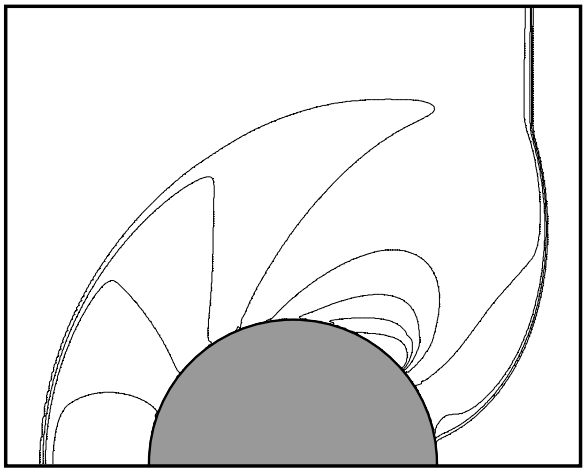

Figure 18: Mach Contours for a Shock Diffracted from a Cylinder (250×200 cartesian grid, $M_o = 2.82$).

4.3 Future Research

The code we have developed using the new upwind and boundary procedures can handle the compressible flow around almost any geometry in two dimensions. Nevertheless, it is still far from being a practical design tool. There are several major improvements that need to be done in order to turn this code into a computational wind tunnel.

Accuracy: First order accuracy has been known to be inadequate for realistic problems. The necessity of using a large number of grid points makes a code inefficient. The interaction of numerical viscosity with the real viscosity terms, if they are included, makes it difficult to resolve boundary and shear layers. That is why, in the last 20 years, the improvement of accuracy without causing unwanted oscillations near shock waves has been a popular area of research.

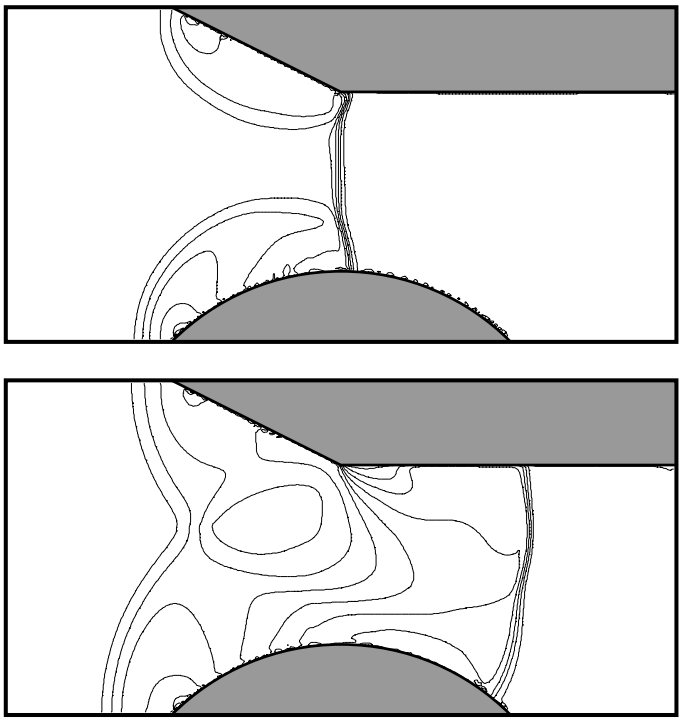

Figure 19: Density Contours for a Shock Moving Inside a Tunnel (200×100 cartesian grid, $M_o = 2.$).

There is an extensive literature on how to construct nonoscillatory higher order accurate schemes [Har91] [Har83] [Swe85] [CO85] [Jam85] [Lee85]. Under the guidance of these references, we can modify our new scheme to attain high accuracy. This can possibly be done either by modifying the averaging stage[4] or by using piecewise linear instead of piecewise constant approximations.

Grid Adaptation: Another way of improving the efficiency is to take advantage of the special form of the solution. Assuming that we generally do not know enough about the solution a priori to do this manually, we have to let the computer adapt the grid so that the grid points are concentrated where the

[4] it is also called projection stage

solution changes rapidly. This also has been a fruitful area of research in recent years. See the following reference and the references therein with regard to the refinement strategies for cartesian grids [Ber85].

Steady-state: If we are interested only in the steady state problems, it is not wise to use an unsteady code without any modification. Because the details of the flowfield before reaching the steady state is generally unimportant, and computing them is a waste of resources. The simplest way to overcome this is to use a coarse grid at the initial stages and to refine the grid as the solution approach the steady state. There is also a vast literature on application of the multi-grid method to steady-state problems. See for example [Hir88].

Navier-Stokes: Ultimately, we need to include the viscosity terms for obvious reasons. The prerequisites for that are achieving higher order of accuracy and implementing a grid adaptation strategy since the boundary and internal layers requires higher accuracy than smooth inviscid regions of the flow. One way of including diffusive terms is to use the Navier-Stokes flux and follow the path to find a consistent decomposition. Another way could be, after writing down the finite difference form of the upwind scheme, to add the Navier-Stokes terms with central differencing. We also have to change the boundary conditions for the viscous flow[5]. This can still be done within the particle description with a new reflection rule.

Real Gas: For high flow speeds, the assumption of ideal gas is not valid. Introduction of real gas effects into the particle picture also is needed to be explored.

Nearly Incompressible Flow: For low mach numbers, the upwind scheme we introduced becomes very inefficient because of the disparity between wave speeds.

[5] no slip etc.

Acoustic waves propagate much faster than convective waves and restrict the time step for stability reasons even though their resolution is not essential. An iterative or semi implicit version of the particle convection idea is needed. This can be also helpful in viscous regions where the accumulation of grid points put a severe restriction on time steps.

Moving Boundaries: Modifying the reflection rule to take into account of the motion of a solid wall should not be difficult. This would enable us to solve arbitrarily complex and moving boundary problems on fixed cartesian grids.

4.4 Concluding Remarks

In this thesis, a new class numerical schemes having the following properties for the solution of gasdynamics equations were introduced.

- *The convective part of the procedure is truly multidimensional:* Particles move in the directions dictated by the domain of influence rather than arbitrarily chosen coordinate directions. In addition, these directions are only a function of the local data as opposed to its gradients unlike some other multidimensional upwind schemes. This makes it less expensive because we do not need to compute gradients to choose upwind direction.

- *Thermodynamic quantities stay in their physical limits:* The scheme preserves the positivity of density and internal energy (or pressure) under a CFL restriction.

- *The scheme is conservative*

- *Solid wall boundary conditions are a straightforward extension of the internal scheme:* The same interpretation of particles is used for both internal and

boundary updates. This type of numerical boundary condition treatment ensures that no mass or energy can be transferred between the fluid and the wall.

- *For the given type of averaging, the scheme is first order accurate in space and time*

The disadvantages of the scheme are given in the previous section. Here, we will summarize the advantages of the scheme:

- *It is grid independent:* Complex geometries can be handled on cartesian grids. In fact, the convective part is independent of the coordinate system being used, therefore we can use any type grids just by changing the averaging.

- *It is intuitive and easy to implement:* All of the results shown in this thesis were computed by a Fortran program less then 17 Kbytes long.

- *It is suitable for parallel computation:* The method is explicit and treats internal and boundary cells uniformly. It also requires a relatively small amount of communication between grid points.

Appendix A

Three Space Dimensions

A.1 Consistency conditions in 3D

$$\mathbf{U} = \sum_k \mathbf{R}_k$$

$$[\mathbf{F}, \mathbf{G}, \mathbf{H}] = \sum_k [\alpha_k, \beta_k, \gamma_k]\mathbf{R}_k$$

A.2 An example of 3D splitting

k	$[\alpha_k, \beta_k, \gamma_k]$	k	$[\alpha_k, \beta_k, \gamma_k]$
1	$[u+c, v+c, w+c]$	5	$[u+c, v+c, w-c]$
2	$[u-c, v+c, w+c]$	6	$[u-c, v+c, w-c]$
3	$[u+c, v-c, w+c]$	7	$[u+c, v-c, w-c]$
4	$[u-c, v-c, w+c]$	8	$[u-c, v-c, w-c]$
		9	$[u, v, w]$

$$\mathbf{R}_1 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u+c \\ v+c \\ w+c \\ h+(u+v+w)c \end{bmatrix} \qquad \mathbf{R}_2 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u-c \\ v+c \\ w+c \\ h-(u-v-w)c \end{bmatrix}$$

$$\mathbf{R}_3 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u+c \\ v-c \\ w+c \\ h+(u-v+w)c \end{bmatrix} \qquad \mathbf{R}_4 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u-c \\ v-c \\ w+c \\ h-(u+v-w)c \end{bmatrix}$$

$$\mathbf{R}_5 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u+c \\ v+c \\ w-c \\ h+(u+v-w)c \end{bmatrix} \qquad \mathbf{R}_6 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u-c \\ v+c \\ w-c \\ h-(u-v+w)c \end{bmatrix}$$

$$\mathbf{R}_7 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u+c \\ v-c \\ w-c \\ h+(u-v-w)c \end{bmatrix} \qquad \mathbf{R}_8 = \frac{\rho}{8\gamma}\begin{bmatrix} 1 \\ u-c \\ v-c \\ w-c \\ h-(u+v+w)c \end{bmatrix}$$

$$\mathbf{R}_9 = \frac{\rho(\gamma-1)}{\gamma}\begin{bmatrix} 1 \\ u \\ v \\ w \\ (u^2+v^2+w^2)/2 \end{bmatrix}$$

Appendix B

Alternative Particle Models

B.1 A Consistent Two Particle Decomposition

k	α_k
1	$u + \frac{c}{\sqrt{\gamma}}$
2	$u - \frac{c}{\sqrt{\gamma}}$

$$\mathbf{R}_1 = \frac{\rho}{2}\begin{bmatrix} 1 \\ u + \frac{c}{\sqrt{\gamma}} \\ \frac{e}{\rho} + \frac{uc}{\sqrt{\gamma}} \end{bmatrix}$$

$$\mathbf{R}_2 = \frac{\rho}{2}\begin{bmatrix} 1 \\ u - \frac{c}{\sqrt{\gamma}} \\ \frac{e}{\rho} - \frac{uc}{\sqrt{\gamma}} \end{bmatrix}$$

B.2 The Beam Scheme of Sanders and Prendergast

k	α_k
1	$u + \sqrt{3}\sigma$
2	u
3	$u - \sqrt{3}\sigma$

$$\mathbf{R}_1 = \frac{\rho}{6}\begin{bmatrix} 1 \\ u+\sqrt{3}\sigma \\ \frac{(u+\sqrt{3}\sigma)^2}{2} + \sigma^2 \end{bmatrix}$$

$$\mathbf{R}_2 = \frac{2\rho}{3}\begin{bmatrix} 1 \\ u \\ \frac{u^2}{2} + \sigma^2 \end{bmatrix}$$

$$\mathbf{R}_3 = \frac{\rho}{6}\begin{bmatrix} 1 \\ u-\sqrt{3}\sigma \\ \frac{(u-\sqrt{3}\sigma)^2}{2} + \sigma^2 \end{bmatrix}$$

where

$$\sigma = \frac{c}{\sqrt{5/3}}.$$

This splitting is valid for monatomic gas.

Bibliography

[Ber85] M. J. Berger. Adaptive mesh refinement for hyperbolic equations. In *Lectures in Applied Mathematics: Large-Scale Computations in Fluid Mechanics*, volume 22, pages 31–40. American Mathematical Society, 1985. part 1.

[BG61] A. E. Bryson and R. W. F. Gross. Diffraction of strong shocks by cones, cylinders and spheres. *Journal of Fluid Mechanics*, 10:1–16, 1961.

[BL90] M. J. Berger and R. J. LeVeque. Stable boundary conditions for cartesian grid calculations. Technical Report NASA CR-182048, NASA Langley/ICASE, May 1990.

[BL91] M. J. Berger and R. J. LeVeque. A rotated difference scheme for cartesian grids in complex geometries. *AIAA 10th Computational Fluid Dynamics Conference, Hawaii, AIAA paper CP-91-1602*, June 1991.

[CH62] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 2. John Wiley & Sons, 1962.

[CIR52] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure Appl. Math.*, 5:243, 1952.

[CO85] S. R. Chakravarthy and S. Osher. Computing with high-resolution upwind schemes for hyperbolic equations. In *Lectures in Applied Mathematics: Large-Scale Computations in Fluid Mechanics*, volume 22, pages 57–86. American Mathematical Society, 1985. part 1.

[Dav84] S. F. Davis. A rotationally-biased upwind difference scheme for the Euler equations. *Journal of Computational Physics*, 56:65–92, 1984.

[DG91] A. Dadone and B. Grossman. A rotated upwind scheme for the Euler equations. *AIAA 29th Aerospace Sciences Meeting, Reno, Nevada, AIAA paper 91-0635*, January 1991.

[Dyk82] M. Van Dyke. *An Album of Fluid Motion*. The Parabolic Press, 1982.

[GS71] S. K. Godunov and U. M. Sultangazin. On discrete models of the kinetic Boltzmann equation. *Russian Mathematical Surveys*, 26:1–56, 1971.

[Har83] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49:357–393, 1983.

[Har91] A. Harten. Recent developments in shock-capturing schemes. Technical Report NASA CR-187502, NASA Langley/ICASE, January 1991.

[Hir88] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1. John Wiley & Sons, 1988.

[Hir90] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 2. John Wiley & Sons, 1990.

[HLL83] A. Harten, P. D. Lax, and B. Van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25:35–61, 1983.

[Jam85] A. Jameson. A nonoscillatory shock capturing scheme using flux limited dissipation. In *Lectures in Applied Mathematics: Large-Scale Computations in Fluid Mechanics*, volume 22, pages 345–370. American Mathematical Society, 1985. part 1.

[Lee85] B. Van Leer. Upwind-difference methods for aerodynamic problems governed by the euler equations. In *Lectures in Applied Mathematics: Large-Scale Computations in Fluid Mechanics*, volume 22, pages 327–336. American Mathematical Society, 1985. part 2.

[LeV90] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser Verlag, 1990.

[Öksüzoğlu92] H. Öksüzoğlu. State vector splitting for the Euler equations of gasdynamics. *AIAA 30th Aerospace Sciences Meeting, Reno, Nevada, AIAA paper 92-0326*, January 1992.

[PI88] T. Platkowski and R. Illner. Discrete velocity models of the Boltzmann equations: A survey of the mathematical aspects of the theory. *SIAM Review*, 30:213–255, 1988.

[PV92] G. Palmer and E. Venkatapathy. Effective treatments of the singular line boundary problem for three dimensional grids. *AIAA 30th Aerospace Sciences Meeting, Reno, Nevada, AIAA paper 92-0545*, January 1992.

[Roe86a] P. L. Roe. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics*, 18:337–365, 1986.

[Roe86b] P. L. Roe. Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics. *Journal of Computational Physics*, 63:458–476, 1986.

[SP74] R. H. Sanders and K. H. Prendergast. The possible relation of 3-kiloparsec arm to explosions in the galactic nucleus. *The Astrophysical Journal*, 188:498–500, 1974.

[SW81] J. L. Steger and R. F. Warming. Flux vector splitting of the inviscid gasdynamics equations with applications to finite difference methods. *Journal of Computational Physics*, 40:263–293, 1981.

[Swe85] P. Sweby. High resolution TVD schemes using flux limiters. In *Lectures in Applied Mathematics: Large-Scale Computations in Fluid Mechanics*, volume 22, pages 289–310. American Mathematical Society, 1985. part 2.

[VK65] W. G. Vincenti and G. H. Kruger. *Introduction to Physical Gas Dynamics*. John Wiley & Sons, 1965.

[Zho91] X. Zhong. Stabilization of the Burnett equations and application to high-altitude hypersonic flows. *AIAA 29th Aerospace Sciences Meeting, Reno, Nevada, AIAA paper 91-0770*, January 1991.