Numerical Experiments on the Unified Coordinates System for Two Dimensional Steady Flow

By

LEUNG Shing Yu, BSc. Mathematics

A Thesis Presented to
The Hong Kong University of Science & Technology
in Partial Fulfillment
of the Requirements for
the Degree of Master of Philosophy
in Mathematics

Hong Kong, June 2001

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Numerical Experiments on the Unified Coordinates System for Two Dimensional Steady Flow

By

LEUNG Shing Yu, BSc. Mathematics

Approved:

Prof. Grafton W. H. Hui, Supervisor

Prof. S.Y. Cheng, Head of Department

Department of Mathematics

June, 2001

Acknowledgment

I would like to express my appreciation to my supervisor Prof. W. H. Hui for his guidance and encouragement. And, I would like to thank him and Dr. S. Kudriakov for their helpful comments and suggestions during the many discussions which have taken place over the past two years.

Contents

Authorization Page ii

Signature Page iii

Acknowledgment Page iv

Table of Contents v

Abstract viii

1 Introduction 1

2 Euler Equations Written in the Unified Coordinates 4

2.1 The Unified Coordinates 4

2.2 Euler Equations in the Unified Coordinates 6

3 Numerical Method 9

3.1 Numerical Procedures 9

3.2 The ξ-Split Riemann Problem 12

4 Modifications on the Numerical Method **21**

4.1 Determination of h 21

4.1.1 Angle Preserving 22

4.1.2 Orthogonality Preserving 23

4.1.3 Area Preserving 25

4.1.4 Weighted Preserving 26

4.2 The h-Equation Solver and the Improvements in its Efficiency 28

4.2.1 Method of Characteristics 28

4.2.2 Unsteady Approach 29

4.2.3 Direct Solver 31

4.3 Riemann Problem Solver and the Improvements in its Efficiency . . . 32

4.3.1 Collisional BGK Method 33

4.3.2 NT Scheme 34

4.3.3 Hybrid Riemann Solver 36

4.4 Composite Operator Approach 39

5 The Use of the Entropy Equation **41**

6 Test Examples **43**

6.1 Two Dimensional Steady Riemann Problem 43

6.1.1 The Problem 43

6.1.2 Exact Riemann Solver 45

6.1.3 Collisional BGK Scheme 50

6.1.4 Hybrid Riemann Solver 50
6.1.5 NT scheme 53
6.1.6 On the Efficiency of the Composite Schemes 55
6.1.7 On the Constrain in h and Its Solver 57
6.2 Two Dimensional Steady Double Riemann Problem 59
6.3 Prandtl-Meyer Expansion Flow 63
6.4 Wedge Flow 70
6.5 Double Blocking Test 74
6.5.1 $M = 1.7$ 74
6.5.2 $M = 2$ 75

7 Conclusions **82**

Appendix **84**

A Kinetic Energy Loss Due to Slipline Averaging in Eulerian Computations **84**

Biliography **87**

Numerical Experiments on the Unified Coordinates System for Two Dimensional Steady Flow

By

LEUNG Shing Yu, BSc. Mathematics

A Thesis Presented to

The Hong Kong University of Science & Technology

Hong Kong, June 2001

Abstract

A Unified Coordinate approach in describing the fluid motion for computational fluid dynamics has been introduced by Hui et. al. (W.H. Hui, P.Y. Li and Z.W. Li, A Unified Coordinate System for Solving the Two-Dimensional Euler Equations, *J. Comp. Phys.* **153** (1999), 596-637). This coordinate system moves with velocity $h\mathbf{q}$ where h is arbitrary and $\mathbf{q}$ is the velocity of the fluid particle. Hence, it includes the Eulerian Coordinates and the Lagrangian Coordinates as two particular cases where $h \equiv 0$ and $h \equiv 1$ respectively. h is suggested to be chosen so that the grid angle is preserved. One of the purposes of this thesis is to explore the possibilities on choosing this free function h. Improvements on the accuracy and the efficiency of the h-equation solver will be given. We will also discuss the way to improve the efficiency of the computations using the Unified Coordinates, including the replacement of the Exact Riemann Solver and the introduction of the Composite Operator approach.

Chapter 1

Introduction

For over two hundred years, two different coordinate systems for describing fluid motion have existed: the Eulerian one describes fluid motion at fixed locations, whereas the Lagrangian one does so following fluid particles. Accordingly, the Eulerian description considers velocities and other properties of fluid particles to be functions of time and of fixed space coordinates. By contrast, the Lagrangian description considers the positions of fluid particles and their other properties to be functions of time and of their permanent identifications, such as their initial positions or any set of material functions of fluid particles.

Computationally, in using the Eulerian Coordinates, the computational cells are fixed in space. Therefore, due to the averaging in the cell, sliplines are usually smeared badly and shocks are also smeared, but somewhat better than sliplines. This fortunate latter situation is just a consequence of the convergence of the characteristics in this genuinely nonlinear wave.

Computational cells in the Lagrangian Coordinates, on the other hand, are literally fluid particles. However, the very fact that computational cells exactly follow fluid particles can result in severe grid deformation, causing inaccuracy and even break down of the computation.

As proposed by Hui, Li and Li [2], we may use a Unified Coordinates approach to do the computations. Instead of really following the fluid particles as in the Lagrangian Coordinates, we may follow pseudo-particles which move with velocity $h\mathbf{q}, \mathbf{q}$ being velocity of fluid particles and h being an arbitrary function which can be determined by suitable choices. This turns out to be a unified description, ranging from the Eulerian when $h \equiv 0$ to the Lagrangian when $h \equiv 1$, and the freedom in choosing h makes it possible to avoid the disadvantages of excessive numerical diffusion across slip lines in the Eulerian description and of severe grid deformation in the Lagrangian description. Extensions from the two dimensional flows [2] to one dimensional Euler equations [3] and three dimensional Euler equations [4] have been carried out successfully. The method has also been generalized and applied to the Shallow water equations in [5].

As suggested in the above articles, h can be chosen so that the grid angle is preserved. The most important consequence is that if the initial grid is orthogonal, the grid will remain orthogonal and hence will be highly regular everywhere in the flow field. One of the purposes of this thesis is to explore other possibilities on choosing the free function h so that it may be easier to be computed, or may give a better resolution of the solutions on the flow variables. Another thing motivates this research is the efficiency of the h-equation solver. From past experience, large proportion of computational time is used to solve the h-equation. In this thesis, we will explain why this happened and try to introduce another way to obtain an accurate solution. Two other ways will also be explained and implemented to improve the efficiency of the computations in using the Unified Coordinates: The first way is to replace the Exact Riemann solver by other type of non-iterative schemes. The other way is to consider a Composite Operator approach in using both the Unified Coordinates and the Eulerian Coordinates.

This thesis is organized as follow. The formulation of the Euler equations written in the Unified Coordinates will first be stated in Chapter 2. Original nu-

merical procedures and the Riemann solutions, which are used in constructing the numerical fluxes at the cell interfaces, are given in Chapter 3. Chapter 4 proposes several improvements in the efficiency and the robustness of the numerical schemes. Newly developed theory [6] applied to the calculations in the Unified Coordinates will be explained in Chapter 5. Lastly, Chapter 6 includes several test cases showing how the suggested methods improve the efficiency of the Unified Coordinates system approach on solving the two-dimensional unsteady Euler equations when the flow is steady.

Chapter 2

Euler Equations Written in the Unified Coordinates

In this chapter, we will first state the Unified Coordinates and then apply them to the two dimensional unsteady Euler equations. Some properties of the transform and the equations will also be given.

2.1 The Unified Coordinates

Starting from Cartesian Coordinates (x, y) and time t in Eulerian description, we make a transformation to coordinates (λ, ξ, η) by

$$dt = d\lambda \tag{2.1}$$

$$dx = hud\lambda + Ad\xi + Ld\eta \tag{2.2}$$

$$dy = hvd\lambda + Bd\xi + Md\eta \tag{2.3}$$

where u and v are the x and y components of fluid velocity $\mathbf{q}$, respectively. Let

$$\frac{D_h}{Dt} \equiv \frac{\partial}{\partial t} + hu\frac{\partial}{\partial x} + hv\frac{\partial}{\partial y} \tag{2.4}$$

denote the material derivative following the pseudo-particle, whose velocity is $h\mathbf{q}$. Then it is easy to show that

$$\frac{D_h\xi}{Dt} = 0 \text{ and } \quad \frac{D_h\eta}{Dt} = 0\,; \tag{2.5}$$

that is, the coordinates (ξ, η) are material functions of the pseudo-particles, and hence are their permanent identifications. Accordingly, computation cells move and deform with pseudo-particles, rather than with fluid particles as in the Lagrangian Coordinates.

The transformation (2.1-2.3) has the following properties:

1. Unlike transformations used in grid generation, which are flow-independent, the unique feature of transformation (2.1-2.3) is it depends on the fluid velocity.

2. In (2.2-2.3), h is an arbitrary function of coordinates (λ, ξ, η). On the other hand, (A, L, B, M) are determined by the compatibility conditions. For example, for dx to be a total differential,

$$\frac{\partial A}{\partial \lambda} = \frac{\partial (hu)}{\partial \xi} \text{ and } \frac{\partial L}{\partial \lambda} = \frac{\partial (hu)}{\partial \eta}\,. \tag{2.6}$$

3. In the special case when $h \equiv 0$, (A, L, B, M) are independent of λ. Then the coordinates (ξ, η) are independent of time λ and are hence fixed in space. This coordinates system is thus Eulerian. Transform (2.1-2.3) is then flow independent and is just like any other transformation from Cartesian Coordinates (x, y) to Curvilinear Coordinates (ξ, η) used in grid generation. In particular, if $A \equiv M \equiv 1$ and $B \equiv L \equiv 0$, (ξ, η) are identical with Cartesian Coordinates (x, y).

4. In the special case when $h \equiv 1$, on the other hand, the pseudo-particles coincide with fluid particles and (ξ, η) are the material functions of fluid particles, and hence are the Lagrangian Coordinates. The conventional choice of the Lagrangian Coordinates, i.e. $(\xi, \eta) = (x, y)|_{t=0}$, is just a special choice of material

functions, corresponding to choosing $A \equiv M \equiv 1$ and $B \equiv L \equiv 0$. It does not offer any particular advantage in numerical computation. In particular, the computational domain in (ξ, η) space can always be easily made regular, rectangular for example, even if it is irregular in the physical space. This cannot be done with the conventional choice of the Lagrangian Coordinates.

5. In the general case, h is arbitrary. It thus provides a new degree of freedom which may be used to advantage: to avoid excessive numerical diffusion in the Eulerian Coordinates, or to avoid severe grid deformation in the Lagrangian Coordinates. The ways of choosing h will be given in Chapter 4.1 where the effect of this parameter on the resolution of the flow discontinuities and the regularity of the grid will be presented in details.

2.2 Euler Equations in the Unified Coordinates

The Euler equations in Cartesian Coordinates for inviscid flow of an ideal gas obeying the γ-law are

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u\left(e + \frac{p}{\rho}\right) \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v\left(e + \frac{p}{\rho}\right) \end{pmatrix} = 0 \qquad (2.7)$$

where ρ, p, and e are the density, pressure, and specific total energy of the gas respectively, with

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

Using the relation (2.1-2.3) in differential form, we can obtain the corresponding expressions for the partial derivatives as

$$\begin{aligned} \frac{\partial}{\partial t} &= \frac{\partial}{\partial \lambda} \\ (AM - BL)\frac{\partial}{\partial x} &= -hI\frac{\partial}{\partial \lambda} + M\frac{\partial}{\partial \xi} - L\frac{\partial}{\partial \eta} \\ (AM - BL)\frac{\partial}{\partial y} &= -hJ\frac{\partial}{\partial \lambda} - B\frac{\partial}{\partial \xi} + A\frac{\partial}{\partial \eta}\,. \end{aligned}$$

Therefore, the Euler equations for two-dimensional unsteady flow become

$$\frac{\partial \mathbf{E}}{\partial \lambda} + \frac{\partial \mathbf{F}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} = 0\,, \tag{2.8}$$

where

$$\mathbf{E} = \begin{pmatrix} \rho\triangle \\ \rho\triangle u \\ \rho\triangle v \\ \rho\triangle e \\ A \\ B \\ L \\ M \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \rho(1-h)I \\ \rho(1-h)Iu + pM \\ \rho(1-h)Iv - pL \\ \rho(1-h)Ie + pI \\ -hu \\ -hv \\ 0 \\ 0 \end{pmatrix}, \mathbf{G} = \begin{pmatrix} \rho(1-h)J \\ \rho(1-h)Ju - pB \\ \rho(1-h)Jv + pA \\ \rho(1-h)Je + pJ \\ 0 \\ 0 \\ -hu \\ -hv \end{pmatrix}, \tag{2.9}$$

with

$$\Delta = det\begin{pmatrix} A & L \\ B & M \end{pmatrix},\ I = uM - vL \text{ and } J = Av - Bu\,. \tag{2.10}$$

We note that:

1. the system of equations (2.9) is in conservation form, consequently it can be solved numerically using any well-established shock-capturing schemes. We will discuss more on this later in Chapter 4.3.

2. the last 4 equations of (2.9) arise from the compatibility requirements of transformation (2.1-2.3). They are called geometric conservation laws, in contrast to the physical conservation laws in the first 4 equations.

As remarked earlier, the Unified Coordinates system is Lagrangian when $h \equiv 1$. In this case, system (2.9) is the equations of motion in the Lagrangian Coordinates which are now written in conservation form. In this regard, it should be pointed out that it is difficult to write the conventional Lagrangian equations in conservation form except, of course, in the special case of 1-D unsteady flow.

Chapter 3

Numerical Method

Following [2], the numerical procedures on how to solve the system (2.8) is given in this chapter. Using Godunov scheme, we have to use the solution from the Riemann problem which will also be included in this chapter.

3.1 Numerical Procedures

The numerical procedure of the Godunov/MUSCL scheme can now be summarized as follows:

1. Initialization. Assume the initial conditions of a flow problem are given at $t = 0(\lambda = 0)$ in the $x-y$ plane. Then an appropriate $\xi-\eta$ Coordinate grid is laid on the $x-y$ plane (for instance, we take ξ and η equal to the arc length of their corresponding Coordinate line on $x-y$ plane), with $\xi = \xi_0, \xi_1, \xi_2, \cdots, \xi_m$, $\eta = \eta_0, \eta_1, \eta_2, \cdots, \eta_n$, and the curve $\xi = \xi_0$ (or $\eta = \eta_0$) coinciding with the solid surface if there is one. Hence $\mathbf{K}^0_{i,j} = (A^0, B^0, L^0, M^0)^T_{i,j}$ as well as the flow variable $\mathbf{Q}^0_{i,j} = (\rho^0, p^0, u^0, v^0)^T_{i,j}$ are obtained by averaging the given flow over the computational cell (i, j). They are used together with $h^0_{i,j} = 0$ as initial

conditions. Consequently, $\mathbf{E}^0_{i,j}, i = 1, 2, \cdots, m, \quad j = 1, 2, \cdots, n$, are available. For example, if we choose ξ, η to be the respective arc lengths of x- and y-coordinate lines then, form (2.1-2.3), $\mathbf{K}^0_{i,j} = (1, 0, 0, 1)^T$ and $\mathbf{E}^0_{i,j}$ follow from its expressions in (2.9).

2. The operation $\mathcal{L}^{\xi}_{\Delta\lambda}$ for marching from λ^k to $\lambda^{k+1} = \lambda^k + \Delta\lambda, k = 0, 1, 2, \cdots$. For every pair of adjacent cells (i, j) and $(i + 1, j)$,

   (a) Do a MUSCL type data reconstruction in a component by component manner. For example, in the ξ direction, let f be any of the above physical variables ρ, p, u, v, then, instead of assuming a uniform state in the cells (i, j) and $(i + 1, j)$, we assume linearly distributed states and use linear extrapolation to determine cell interface flow variables: $f_r = f_{i+1,j} - 0.5(f_{i+2,j} - f_{i+1,j})\phi(r^+)$ with $r^+ = (f_{i+1,j} - f_{i,j})/(f_{i+2,j} - f_{i+1,j})$ and $f_\ell = f_{i,j} + 0.5(f_{i,j} - f_{i-1,j})\phi(r^-)$ with $r^- = (f_{i+1,j} - f_{i,j})/(f_{i,j} - f_{i-1,j})$, where $\phi(r) = \max(0, \min(1, r))$ is the minmod flux limiter and subscripts r and ℓ of f correspond to right and left states, respectively.

   (b) Define the normal direction of the cell interface $\xi_{i+\frac{1}{2},j}$ between the two adjacent cells (i, j) and $(i + 1, j)$ as

   $$\mathbf{n} = \frac{(\nabla\xi)_{i,j} + (\nabla\xi)_{i+1,j}}{|(\nabla\xi)_{i,j} + (\nabla\xi)_{i+1,j}|} \tag{3.1}$$

   i.e., the average of $(\nabla\xi)_{i,j}$ and $(\nabla\xi)_{i+1,j}$. Project the velocity vector $\mathbf{q} = (u, v)$ into the normal and the tangential components (ω and τ).

   (c) Solve the Riemann problem to get the interfacial flow variables $(\rho, p, \omega, \tau)^T$ and hence $(\rho, p, u, v)^T$ at $\xi = \xi_{i+\frac{1}{2},j}$. These are constants and will be denoted by $(\cdot)_{i+\frac{1}{2},j}$. The idea will be further explained later in Chapter 3.2 and Chapter 4.3.

(d) Update $\mathbf{K}_{i,j}^k$ to $\mathbf{K}_{i,j}^{k+1}$ as follows

$$\begin{pmatrix} A_{i,j}^{k+1} \\ B_{i,j}^{k+1} \end{pmatrix} = \begin{pmatrix} A_{i,j}^{k} \\ B_{i,j}^{k} \end{pmatrix} + \frac{\Delta\lambda^k}{\Delta\xi_i} h_{i,j}^k \begin{pmatrix} u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j} \\ v_{i+\frac{1}{2},j} - v_{i-\frac{1}{2},j} \end{pmatrix} \tag{3.2}$$

$$\begin{pmatrix} L_{i,j}^{k+1} \\ M_{i,j}^{k+1} \end{pmatrix} = \begin{pmatrix} L_{i,j}^{k} \\ M_{i,j}^{k} \end{pmatrix} \tag{3.3}$$

(e) Calculate the first four components of the cell interface flux. For instance, the 2nd component of the interface flux $\mathbf{F}_{i+\frac{1}{2},j}^{k+\frac{1}{2}}$ is evaluated as

$$\rho_{i+\frac{1}{2},j}(1 - h_{i,j}^k)(u_{i+\frac{1}{2},j} M_{i,j}^{k+1} - v_{i+\frac{1}{2},j} L_{i,j}^{k+1}) + p_{i+\frac{1}{2},j} M_{i,j}^{k+1} \tag{3.4}$$

3. The operation $\mathcal{L}_{\Delta\lambda}^{\eta}$ for marching from λ^k to $\lambda^{k+1} = \lambda^k + \Delta\lambda, k = 0, 1, 2, \cdots$. For every pair of adjacent cells (i, j) and $(i, j+1)$, repeat a similar procedures as step 2(a-e).

4. Update

(a) Update the conserved variables $\mathbf{E}_p$ in the physical conservation laws (2.9) using

$$\mathbf{E}_{p_{i,j}}^{k+1} = \mathbf{E}_{p_{i,j}}^{k} - \frac{\Delta\lambda^k}{\Delta\xi_i}(\mathbf{F}_{i+\frac{1}{2},j}^{k+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2},j}^{k+\frac{1}{2}}) - \frac{\Delta\lambda^k}{\Delta\eta_i}(\mathbf{G}_{i,j+\frac{1}{2}}^{k+\frac{1}{2}} - \mathbf{G}_{i,j-\frac{1}{2}}^{k+\frac{1}{2}}) \tag{3.5}$$

(b) Decode $\mathbf{E}_{p_{i,j}}^{k+1}$ to get $\mathbf{Q}_{i,j}^{k+1}$, using $\Delta = A_{i,j}^{k+1} M_{i,j}^{k+1} - B_{i,j}^{k+1} L_{i,j}^{k+1}$

(c) (Note: this step is, of course, to be by-passed if $h \equiv$ const is assumed in the computation) Update $h_{i,j}^k$ to $h_{i,j}^{k+1}$. We will explain more on this later in Chapter 4.1-4.2.

(d) Calculate the grid in the $x - y$ plane at λ^{k+1}:

$$\begin{cases} x_{i,j}^{k+1} = x_{i,j}^k + \triangle\lambda h_{i,j}^{k+1} u_{i,j}^{k+1} \\ y_{i,j}^{k+1} = y_{i,j}^k + \triangle\lambda h_{i,j}^{k+1} v_{i,j}^{k+1} \end{cases} \tag{3.6}$$

By a grid we mean the lines joining the cell centers, not the cell interface lines. We remark that the grid in the physical plane is not used in the

subsequent computation (only the values of $\mathbf{K}$ are used) as the whole computation is carried out in the transformed plane (the $\xi - \eta$ plane). So, this step 4(d) is optional. However, the grid information is useful in computing steady flow as asymptotic state of unsteady flow for large λ. In this case to check if a steady state is reached, which means the flow at every fixed location in the $x - y$ plane does not change with increasing time, we should compare the flow variables $\mathbf{Q}$ at the same fixed point (x, y) in the physical plane and not at the same points (ξ, η) in the transformed plane; the latter are simply the pseudo-particles whose positions in the $x - y$ plane in general move with λ and never reach an asymptotic state.

After this, we repeat steps 2-4 to advance the solution further to λ^{k+2}, and so on.

3.2 The ξ-Split Riemann Problem

Based on the solution strategies explained in the last section, the key step is the solution to the 1-D Riemann problem over the time step $\Omega^k(\lambda) : \lambda^k < \lambda \leq \lambda^{k+1}$ resulting from dimensional splitting.

In this section, we will explain how to derive the 1-D Riemann solution in the $\lambda - \xi$ plane, in particular the flow variable $\mathbf{Q}$ at the interface $\xi = 0$ for $\lambda \in \Omega^k(\lambda)$. The 1-D Riemann problem in the $\lambda - \eta$ plane can be obtained similarly.

From (2.9), at time step λ^k (to be taken as 0 for simplicity) the 1-D physical conservation law equations in the $\lambda - \xi$ plane resulting from dimensional splitting are

$$\frac{\partial \mathbf{E}_p}{\partial \lambda} + \frac{\partial \mathbf{F}_p}{\partial \xi} = 0, \quad \lambda \in \Omega(\lambda) : 0 < \lambda \leq \triangle\lambda \tag{3.7a}$$

where

$$\mathbf{E}_p = \begin{pmatrix} \rho\triangle \\ \rho\triangle u \\ \rho\triangle v \\ \rho\triangle e \\ A \\ B \end{pmatrix}, \quad \mathbf{F}_p = \begin{pmatrix} \rho(1-h)I \\ \rho(1-h)Iu + pM \\ \rho(1-h)Iv - pL \\ \rho(1-h)Ie + pI \\ -hu \\ -hv \end{pmatrix} \tag{3.7b}$$

with

$$\Delta = AM - BL\,, \quad I = uM - vL\,, \quad e = \frac{1}{2}(u^2+v^2) + \frac{1}{\gamma-1}\frac{p}{\rho}\,. \tag{3.8}$$

The last two equations in (2.9) are dropped here due to the fact that L and M will be kept constant in this ξ-splitting. Similarly, equations for A and B are dropped in η-splitting Riemann Problem.

In applying the Godunov scheme to advance the solution from $\lambda = 0$ to $\lambda = \Delta\lambda$, the initial data for the adjacent cells (i,j) and $(i+1,j)$ are the following Riemann (constant) data (for simplicity we take the cell interface between these two cells to be located at $\xi = 0$)

$$\mathbf{E_p}\Big|^{\lambda=0} = \begin{cases} \mathbf{E}_\ell (= \mathbf{E}_{i,j}^{\lambda=0}), & \xi < 0 \\ \mathbf{E}_r (= \mathbf{E}_{i+1,j}^{\lambda=0}), & \xi > 0\,. \end{cases} \tag{3.9}$$

To put the Riemann problem in the $\lambda - \xi$ plane more explicitly in one-dimensional form, we note that the normal direction of the plane ξ = constant is

$$\mathbf{n} = \frac{\nabla\xi}{|\nabla\xi|} = \frac{(M,-L)}{S} \tag{3.10}$$

and project the flow velocity $\mathbf{q}$ into the normal direction $\mathbf{n}$ and the tangential direction $\mathbf{t}$ to get

$$\begin{cases} \omega = \mathbf{q}\cdot\mathbf{n} = (uM - vL)/S \\ \tau = \mathbf{q}\cdot\mathbf{t} = (uL + vM)/S\,. \end{cases} \tag{3.11}$$

We also replace (L, M) by S and ψ as follows

$$\begin{cases} S = \sqrt{L^2 + M^2} \\ \tan\psi = M/L\,. \end{cases} \tag{3.12}$$

The ξ-split Riemann problem then becomes

$$\begin{cases} \dfrac{\partial \mathbf{E}'}{\partial \lambda} + \dfrac{\partial \mathbf{F}'}{\partial \xi} = 0\,,\ \lambda \in \Omega(\lambda)\,,\ -\infty < \xi < \infty \\ \mathbf{E}'(0,\xi) = \begin{cases} \mathbf{E}'_l\,,\ \xi < 0 \\ \mathbf{E}'_r\,,\ \xi > 0 \end{cases} \end{cases} \tag{3.13}$$

where

$$\mathbf{E}' = \begin{pmatrix} \rho\Delta \\ \rho\Delta\omega \\ \rho\Delta\tau \\ \rho\Delta e \\ A \\ B \end{pmatrix}, \quad \mathbf{F}' = S \begin{pmatrix} \rho(1-h)\omega \\ \rho(1-h)\omega^2 + p \\ \rho(1-h)\omega\tau \\ \rho(1-h)\omega e + \omega p \\ -hu/S \\ -hv/S \end{pmatrix}. \tag{3.14}$$

Our purpose is to find the flux $\mathbf{F}'$ on $\xi = 0$ to be used in the Godunov scheme to update the conserved quantities $\mathbf{E}'$. h in (3.14) is taken to be equal to $h_i^0 = h_l$ for $\xi < 0$ and $h_{i+1}^0 = h_r$ for $\xi > 0$. That is to say, they are assumed constant for $0 \le \lambda < \Delta\lambda$. This is actually consistent with the h-equation proposed in Chapter 4.1 where λ appears as a parameter only.

Now we first find all possible solutions to the above system for $\xi > 0$ and $\xi < 0$ separately, and then use them to construct solution to the Riemann problem for $-\infty < \xi < \infty$.

Case (1) : $\xi > 0$

The Riemann Problem is

$$\begin{cases} \dfrac{\partial \mathbf{E}'}{\partial \lambda} + \dfrac{\partial \mathbf{F}'}{\partial \xi} = 0\,,\ \lambda \in \Omega(\lambda)\,,\ 0 < \xi < \infty \\ \mathbf{E}'(0,\xi) = \mathbf{E}'_r \end{cases} \tag{3.15}$$

(a) The Eigenfields

For smooth solution, we first find the eigenvalues and the corresponding right eigenvectors of the system which can be written as

$$\mathbf{G}\,\frac{\partial \mathbf{U}}{\partial \lambda} + \mathbf{H}\,\frac{\partial \mathbf{U}}{\partial \xi} = 0\,, \tag{3.16}$$

where $U = (\rho, p, \omega, \tau, A, B)^T$ and $\mathbf{G} = \dfrac{d\mathbf{E}'}{d\mathbf{U}}$, $\mathbf{H} = \dfrac{d\mathbf{F}'}{d\mathbf{U}}$. In order to obtain the eigenvalues σ we find the determinant of the matrix $(\sigma\mathbf{G} - \mathbf{H})$, which is given by

$$\begin{pmatrix} m & 0 & Y & 0 & \sigma\rho M & -\sigma\rho L \\ m\omega & -S & \rho m + Y\omega & 0 & \sigma\rho M\omega & -\sigma\rho L\omega \\ m\tau & 0 & Y\tau & \rho m & \sigma\rho M\tau & -\sigma\rho L\tau \\ \dfrac{m|\mathbf{q}|^2}{2} & \dfrac{m}{\gamma-1} - S\omega & \rho\omega m + Ye + \dfrac{Yp}{(1-h)\rho} & \rho m\tau & \sigma\rho Me & -\sigma\rho Le \\ 0 & 0 & \dfrac{hM}{S} & \dfrac{hL}{S} & \sigma & 0 \\ 0 & 0 & -\dfrac{hL}{S} & \dfrac{hM}{S} & 0 & \sigma \end{pmatrix}$$

where $m = \sigma\Delta - S(1-h)\omega$ and $Y = -S(1-h)\rho$.

Direct computation gives

$$|\sigma\mathbf{G} - \mathbf{H}| = \frac{m^2\rho^2\sigma^2(m^2\rho - \gamma p S^2)}{\gamma - 1}\,. \tag{3.17}$$

Therefore, the eigenvalues of equation (3.16) are

$$\begin{aligned} \sigma_{1,2} &= 0 && \text{(multiplicity of 2)} \\ \sigma_{3,4} &= \frac{(1-h)S\omega}{\Delta} && \text{(multiplicity of 2)} \\ \sigma_{\pm} &= \frac{S}{\Delta}[(1-h)\omega \pm a] \end{aligned} \tag{3.18}$$

Their corresponding right eigenvectors are

$$\begin{aligned} \mathbf{r}_1 &= (0,0,0,0,1,0)^T \\ \mathbf{r}_2 &= (0,0,0,0,0,1)^T \end{aligned} \tag{3.19}$$

for $\sigma_{1,2}$,

$$\begin{aligned} \mathbf{r}_3 &= (1,0,0,0,0,0)^T \\ \mathbf{r}_4 &= (0,0,0,1,\frac{-hL}{S\sigma_{3,4}},\frac{-hM}{S\sigma_{3,4}})^T \end{aligned} \tag{3.20}$$

for $\sigma_{3,4}$ and

$$\mathbf{r}_\pm = \left(\frac{1}{a^2},1,\pm\frac{1}{a\rho},0,\frac{\mp hM}{S\sigma_\pm\rho a},\frac{\pm hL}{S\sigma_\pm\rho a}\right)^T \tag{3.21}$$

for $\sigma_\pm$. It can be proven that

$$\nabla\sigma_{1,2}\cdot\mathbf{r}_{1,2}=0$$

$$\nabla\sigma_{3,4}\cdot\mathbf{r}_{3,4}=0$$

which implies that the eigenfield $\sigma_{1,2,3,4}$ are linearly degenerated. On the other hand,

$$\nabla\sigma_\pm\cdot\mathbf{r}_\pm=\pm\frac{S(1+\gamma)}{2\Delta\rho a}\neq 0\,.$$

Therefore, the eigenfield $\sigma_\pm$ are genuinely non-linear.

(b) Smooth Solutions

The smooth solution for the eigenfields $\sigma_\pm$ are determined from

$$\begin{cases} \dfrac{d\rho}{dp}=\dfrac{1}{a^2} \\ \dfrac{d\omega}{dp}=\pm\dfrac{1}{a\rho} \\ \dfrac{d\tau}{dp}=0 \\ \dfrac{dA}{dp}=\dfrac{\mp hM}{S\sigma_\pm\rho a} \\ \dfrac{dB}{dp}=\dfrac{\pm hL}{S\sigma_\pm\rho a}\,. \end{cases} \tag{3.22}$$

The solution for ρ,ω, and τ relates the flow state $\mathbf{Q}=(\rho,p,\omega,\tau)^T$ in the expansion fan to the initial state $\mathbf{Q}_0=(\rho_0,p_0,\omega_0,\tau_0)^T$ upstream of the fan. This solution can be easily found and is most conveniently given in

terms of the pressure ratio $\alpha = p/p_0$ as follows

$$\begin{cases} \rho = \rho_0 \alpha^{\frac{1}{\gamma}} \\ \omega = \omega_0 \pm \frac{2a_0}{\gamma-1}\left(\alpha^{\frac{\gamma-1}{2\gamma}} - 1\right) \\ \tau = \tau_0 \end{cases} \tag{3.23}$$

where $a_0 = \sqrt{\gamma p_0/\rho_0}$. Note that τ does not change across an expansion fan and that equations (3.23) are identical to those of the purely 1-D unsteady flow; in particular, they are independent of $\mathbf{K}_r$ and h_r.

Let (λ, ξ) be a general point inside the expansion fan. The slope of the characteristic is given by

$$\frac{d\xi}{d\lambda} = \frac{\xi}{\lambda} = \sigma_\pm = \frac{S}{\Delta}\left[(1-h)\omega \pm a\right]. \tag{3.24}$$

The solution for flow inside the fan is

$$\begin{cases} p = & p_0 \left\{ \frac{2(1-h)}{\gamma-2h+1} \pm \frac{\gamma-1}{(\gamma-2h+1)a_0}\left((1-h)\omega_0 - \frac{\Delta}{S}\frac{\xi}{\lambda}\right)\right\}^{\frac{2\gamma}{\gamma-1}} \\ \rho = & \rho_0 \alpha^{\frac{1}{\gamma}} \\ \omega = & \omega_0 \pm \frac{2a_0}{\gamma-1}\left(\alpha^{\frac{\gamma-1}{2\gamma}} - 1\right) \\ \tau = & \tau_0 \,. \end{cases} \tag{3.25}$$

If we put $h = 0$ in (3.25), we recover the solution as obtained in the Eulerian Coordinates, as it should. The variation of A and B across an expansion fan can also be obtained from (3.22), but they are not needed in calculating the flux $\mathbf{F}'$ and are thus not given here.

For discontinuous solutions, we start from the Rankine-Hugoniot conditions for (3.15),

$$\begin{cases} c[\rho\Delta] = [S(1-h)\rho\omega] \\ c[\rho\Delta\omega] = [S(1-h)\rho\omega^2 + Sp] \\ c[\rho\Delta\tau] = [S(1-h)\rho\omega\tau] \\ c[\rho\Delta e] = [S(1-h)\rho\omega e + S\omega p] \\ c[A] = -[hu] \\ c[B] = -[hv] \end{cases} \tag{3.26}$$

where $[\cdot]$ denotes the jump across the discontinuity whose speed is denoted by $c = d\xi/d\lambda$.

(c) Shock Waves

We denote the pre-shock (upstream) flow state by $\mathbf{Q}_0 = (\rho_0, p_0, \omega_0, \tau_0)^T$ and the post-shock (downstream) flow state by $\mathbf{Q} = (\rho, p, \omega, \tau)^T$, respectively. Then the shock jump relations can be expressed in terms of $\alpha = p/p_0$ as follows:

$$\begin{cases} \rho & = \rho_0 \frac{\alpha(\gamma+1)+\gamma-1}{\alpha(\gamma-1)+\gamma+1} \\ \omega & = \omega_0 \pm \frac{a_0(\alpha-1)}{\sqrt{\frac{1}{2}\gamma(\alpha(\gamma+1)+\gamma-1)}} \\ \tau & = \tau_0 \,. \end{cases} \tag{3.27}$$

Again, we see that τ does not jump across a shock and that equations (3.27) are identical to those of the purely 1-D unsteady flow; in particular they are independent of $\mathbf{K}_r$ and h_r. We note that the jumps of A and B across a shock may also be obtained from (3.26), but they are not used in calculating the flux $\mathbf{F}'$ and are thus not given here.

(d) Slip Lines

In this case, we get

$$\begin{cases} p & = p_0 \\ \omega & = \omega_0 \end{cases} \tag{3.28}$$

but the density ρ, tangential velocity τ, A and B may jump arbitrarily. Once again, we note that (3.28) are identical to the purely 1-D flow; and in particular they are independent of $\mathbf{K}_r$ and h_r.

Case (2): $\xi < 0$

The solution for $\xi < 0$ can be obtained similarly.

Now after obtaining all possible solutions for $\xi > 0$ and $\xi < 0$ separately, the question is how to construct the solution to the Riemann problem for $\lambda > 0, -\infty <$

$\xi < \infty$. We find that at $\xi = 0$ the coefficients in $\mathbf{E}'$ and $\mathbf{F}'$ jumps discontinuously. This is the difficulty one would face with in the Eulerian system using curvilinear coordinates rather than Cartesian Coordinates.

The Riemann solution in the neighborhood of $\xi = 0$ is given by the $\sigma_{1,2}$ field whose speed is $c = 0$. The flow states on the two sides of cell interface $\xi = 0$ are related by equations (3.26) with $c = 0$. These are six equations relating four jumps of p, ρ, ω and τ and therefore, in general have no solution, except when $h_l = h_r$, $L_l = L_r$ and $M_l = M_r$; in the latter case the flow is continuous across $\xi = 0$.

To avoid the difficulty of non-existence of solution to the Rankine-Hugoniot relations (3.26), we replace both h_l and h_r by their average, i.e. $h_l = h_r = \bar{h}$, and similarly replace L_l and L_r, M_l and M_r by $\bar{L}$ and $\bar{M}$ respectively. Consequently, the Rankine-Hugoniot relations are satisfied and and the flow is continuous across $\xi/\lambda = 0$. We note from previous discussions that these replacements do not alter the relations of the flow variables (p, ρ, ω, τ) across the elementary waves as they do not depend on $(\mathbf{K}, h)$. It should be pointed out that the replacements of the geometric variables (L, M) by their averages is a fictitious one - they are invoked only to ensure the existence of the solution to (3.26) - but these average values are never used in the computation. On the other hand, the replacement of h_l and h_r by $\bar{h}$ is a necessary one: it is used in equation (3.25) when the line $\xi/\lambda = 0$ is inside the expansion fan.

The Riemann solution for $-\infty < \xi < \infty$ can now be constructed in the usual way as if the slipline corresponding to $c = \sigma_{1,2} = 0$ did not exist.

In summary, we note that for the flow variables $\mathbf{Q} = (\rho, p, \omega, \tau)^T$ their relations across a shock (3.27), across a slip line (3.28) and across an expansion wave (3.23) hold separately in their regions $\xi < 0$ or $\xi > 0$ and are all independent of the values of the geometric variables $\mathbf{K} = (A, B, L, M)^T$ and h, provided the expansion wave lies entirely in the region $\xi < 0$ or entirely in $\xi > 0$. On the other hand, shock

speed, slip line speed and the structure of the flow inside the expansion fan (3.25), e.g. fan width and location, are dependent on the values of $\mathbf{K}$ and h. Such dependence would be needed to construct the complete Riemann solution for $\lambda \in \Omega(\lambda)$ and for all ξ values. But, in using the Godunov scheme to advance the solution from $\lambda = 0$ to $\lambda = \triangle\lambda$, we need only the flow variables $\mathbf{Q}$ at the cell interface $\xi = 0$ (to compute the flux $\mathbf{F}^{k+\frac{1}{2}}_{i+\frac{1}{2},j}$) which are entirely independent of the values of $\mathbf{K}$ and h and are continuous across the interface, provided the expansion wave lies entirely in either $\xi < 0$ or $\xi > 0$.

Chapter 4

Modifications on the Numerical Method

In this chapter, we will concentrate on different ways to improve the efficiency of the numerical procedures by (Chapter 4.1) giving several ways on determining h and (Chapter 4.2) the ways to solve the constrain equation, (Chapter 4.3) replacing the exact Riemann solver by other type of schemes and (Chapter 4.4) considering the situations when we use the Unified Coordinates instead of the Eulerian Coordinates.

4.1 Determination of h

As mentioned earlier, the chief advantage of the Unified Coordinates is the new degree of freedom in choosing h. Many choices are possible and the simplest one is to choose a constant value for it. Numerical experiments for constant h will be presented in Chapter 6 to show its effects on grid deformation and on resolution of flow discontinuities. In general, it is necessary to restrict h to within the range $0 \leq h \leq 1$. For $h > 1$, the eigenvalue $\sigma_{3,4}$ in equation (3.18) has an opposite sign to that for $h < 1$, indicating signals propagate in the wrong direction. Our

computations for $h > 1$ break down immediately. On the other hand, for $h < 0$, which means the pseudo-particles are moving in the opposite direction to the fluid particles, computation can be carried out initially but after some finite time it breaks down also. In this section, we will discuss some other possible choices of h.

4.1.1 Angle Preserving

One of a good choice for h is to preserve the grid angles in the solution process which marches in λ, i.e.

$$\frac{\partial}{\partial \lambda}\left[\frac{\nabla \xi}{|\nabla \xi|}.\frac{\nabla \eta}{|\nabla \eta|}\right] = 0\,. \tag{4.1}$$

Since

$$\nabla \xi = (M, -L)/\triangle \text{ and } \nabla \eta = (-B, A)/\triangle\,, \tag{4.2}$$

condition (4.1) becomes

$$\frac{\partial}{\partial \lambda}\left[\frac{AL + BM}{\sqrt{A^2 + B^2}\sqrt{L^2 + M^2}}\right] = 0\,. \tag{4.3}$$

By making use of the last four equations of (2.9), it is easy to show that (4.3) is equivalent to

$$S^2 J\frac{\partial h}{\partial \xi} + T^2 I\frac{\partial h}{\partial \eta} = \left[S^2\left(B\frac{\partial u}{\partial \xi} - A\frac{\partial v}{\partial \xi}\right) - T^2\left(M\frac{\partial u}{\partial \eta} - L\frac{\partial v}{\partial \eta}\right)\right]h \tag{4.4}$$

where

$$S^2 = L^2 + M^2\,,\ T^2 = A^2 + B^2\,,\ I = uM - vL \text{ and } J = Av - Bu\,. \tag{4.5}$$

A consequence of determining h from (4.4) is that if the grid is orthogonal at $\lambda = 0$ it will remain so for subsequent λ. Orthogonal grid is known to possess many desirable properties over non-orthogonal grids, e.g. attaining higher accuracy than non-orthogonal grids. Computationally, (4.4) is to be solved at every time step after the flow variables $\mathbf{Q} = (\rho, p, u, v)^T$ and the geometric variable $\mathbf{K} = (A, B, L, M)^T$ are found. Therefore, it is a first order linear partial differential equations for $h(\xi, \eta; \lambda)$ with λ appearing as a parameter.

To find solution h in the range

$$0 \leq h \leq 1\,, \tag{4.6}$$

we note that (4.4) is linear and homogeneous, therefore it possesses two properties: (a) positive solution $h > 0$ always exists, and (b) if h is a solution to (4.4) so is h/C, C being any constant. Making use of property (a), we let $g = \ln(hq)$ to get

$$\begin{aligned} & S^2(A\sin\theta - B\cos\theta)\frac{\partial g}{\partial \xi} + T^2(M\cos\theta - L\sin\theta)\frac{\partial g}{\partial \eta} \\ = \;& S^2\left(B\frac{\partial \cos\theta}{\partial \xi} - A\frac{\partial \sin\theta}{\partial \xi}\right) - T^2\left(M\frac{\partial \cos\theta}{\partial \eta} - L\frac{\partial \sin\theta}{\partial \eta}\right) \end{aligned} \tag{4.7}$$

where $q = \sqrt{u^2 + v^2}$ and θ is the flow angle with $u = q\cos\theta$ and $v = q\sin\theta$. Or, we write it as

$$a_{\angle}\frac{\partial g}{\partial \xi} + b_{\angle}\frac{\partial g}{\partial \eta} + c_{\angle} = 0\,. \tag{4.8}$$

Now, if g_1 is any solution to (4.8) then $h = e^{g_1}/qC$ is a solution to (4.4) satisfying condition (4.1), provided we choose C equal to the maximum of e^{g_1}/q over the whole flow field being computed. The reason to work with $\ln(hq)$ instead of $\ln h$ is that from the experience with steady flow [1], hq is continuous across slip lines, hence working with hq can minimize the numerical errors.

4.1.2 Orthogonality Preserving

When orthogonal grid is what we want to get finally, condition (4.1) is not the only way what we can choose. If the initial grid is orthogonal, the quantity $\nabla\xi \cdot \nabla\eta$ is zero already, no matter what $|\nabla\xi|$ and $|\nabla\eta|$ are. Therefore, we may use

$$\frac{\partial}{\partial \lambda}[\nabla\xi \cdot \nabla\eta] = 0\,, \tag{4.9}$$

giving

$$\frac{\partial}{\partial \lambda}\left[\frac{AL + BM}{(AM - BL)^2}\right] = 0\,. \tag{4.10}$$

By replacing the λ-derivative by ξ- and η- derivatives, we have

$$\begin{aligned}
[\Delta(uL+vM)-2(AL+BM)I]\frac{\partial h}{\partial \xi}+ \\
[\Delta(uA+vB)-2(AL+BM)J]\frac{\partial h}{\partial \eta}+ \\
\left[\Delta\left(L\frac{\partial u}{\partial \xi}+M\frac{\partial v}{\partial \xi}+A\frac{\partial u}{\partial \eta}+B\frac{\partial v}{\partial \eta}\right)+\right. \\
\left.2(AL+BM)\left(B\frac{\partial u}{\partial \eta}+L\frac{\partial v}{\partial \xi}-A\frac{\partial u}{\partial \eta}-M\frac{\partial v}{\partial \xi}\right)\right]h &= 0\,. \qquad (4.11)
\end{aligned}$$

Transforming with $g=\ln(hq)$, the above condition becomes

$$\begin{aligned}
[\Delta(L\cos\theta+M\sin\theta)-2(AL+BM)I]\frac{\partial g}{\partial \xi}+ \\
[\Delta(A\cos\theta+B\sin\theta)-2(AL+BM)J]\frac{\partial g}{\partial \eta}+ \\
\left[\Delta\left(L\frac{\partial\cos\theta}{\partial \xi}+M\frac{\partial\sin\theta}{\partial \xi}+A\frac{\partial\cos\theta}{\partial \eta}+B\frac{\partial\sin\theta}{\partial \eta}\right)+\right. \\
\left.2(AL+BM)\left(B\frac{\partial\cos\theta}{\partial \eta}+L\frac{\partial\sin\theta}{\partial \xi}-A\frac{\partial\cos\theta}{\partial \eta}-M\frac{\partial\sin\theta}{\partial \xi}\right)\right] &= 0\,, \qquad (4.12)
\end{aligned}$$

or, denoted by

$$a_{O_1}\frac{\partial g}{\partial \xi}+b_{O_1}\frac{\partial g}{\partial \eta}+c_{O_1}=0\,. \qquad (4.13)$$

On the other hand, we note that the denominator in equation (4.10), $(AM-BL)$ represents the Jacobian of our transformation (2.1-2.3) or the area of the grid, which will never be zero. Therefore, we may also set the constrain to be

$$\frac{\partial}{\partial \lambda}[\Delta^2(\nabla\xi\cdot\nabla\eta)]=0 \qquad (4.14)$$

in order to have the orthogonal grid. Simplifying the above condition, we have

$$\frac{\partial}{\partial \lambda}(AL+BM)=0 \qquad (4.15)$$

which gives

$$(uL+vM)\frac{\partial h}{\partial \xi}+(uA+vB)\frac{\partial h}{\partial \eta}+\left[L\frac{\partial u}{\partial \xi}+M\frac{\partial v}{\partial \xi}+A\frac{\partial u}{\partial \eta}+B\frac{\partial v}{\partial \eta}\right]h=0\,. \qquad (4.16)$$

Having the new variable, we get

$$\begin{aligned}(L\cos\theta + M\sin\theta)\frac{\partial g}{\partial \xi} + (A\cos\theta + B\sin\theta)\frac{\partial g}{\partial \eta} + \\ \left[L\frac{\partial \cos\theta}{\partial \xi} + M\frac{\partial \sin\theta}{\partial \xi} + A\frac{\partial \cos\theta}{\partial \eta} + B\frac{\partial \sin\theta}{\partial \eta}\right] &= 0\,, \end{aligned} \tag{4.17}$$

or, denoted by

$$a_{O_2}\frac{\partial g}{\partial \xi} + b_{O_2}\frac{\partial g}{\partial \eta} + c_{O_2} = 0\,. \tag{4.18}$$

We note that the coefficients of this equation is the simplest one with comparing with (4.7) and (4.12). In the case where the flow is steady and the initial grid is orthogonal, all three conditions will generate the same final grid where the grid angles are all orthogonal everywhere. Therefore, the flow quantities given by $\angle$-grid, O_1-grid and O_2-grid should be basically the same.

Although orthogonal grid may attain a higher accuracy solution, in some cases, it may be very difficult for the grid to satisfy the above condition, like the double-blocking test in Chapter 6.4 where the problem is singular in geometry. Or, in some other cases, the grid will be highly compressed. When the numerical scheme is not accurate enough, the grid may squeeze together and which may blow up the computation (like figure 6.27(b)). Therefore, we now propose another idea on choosing h.

4.1.3 Area Preserving

The cell-area, or the Jacobian of our transformation is denoted by Δ. By choosing h where the cell-area is preserved in the solution processes which marches in λ, we have

$$\frac{\partial \Delta}{\partial \lambda} = \frac{\partial}{\partial \lambda}[AM - BL] = 0\,. \tag{4.19}$$

Therefore, using the last four equations of (2.9), it can be shown that condition (4.19) is transformed to

$$I\frac{\partial h}{\partial \xi} + J\frac{\partial h}{\partial \eta} = \left(B\frac{\partial u}{\partial \eta} + L\frac{\partial v}{\partial \xi} - A\frac{\partial v}{\partial \eta} - M\frac{\partial u}{\partial \xi} \right) h \,. \tag{4.20}$$

Defining the same variable as in previous sections, i.e. $g = \ln(hq)$, we have

$$\begin{aligned} &(M\cos\theta - L\sin\theta)\frac{\partial g}{\partial \xi} + (A\sin\theta - B\cos\theta)\frac{\partial g}{\partial \eta} \\ &+ \left(M\frac{\partial \cos\theta}{\partial \xi} - L\frac{\partial \sin\theta}{\partial \xi} + A\frac{\partial \sin\theta}{\partial \eta} - B\frac{\partial \cos\theta}{\partial \eta} \right) = 0 \,, \end{aligned} \tag{4.21}$$

where $q = \sqrt{u^2 + v^2}$ and θ is the flow angle. Or, simply written in the form of

$$a_A\frac{\partial g}{\partial \xi} + b_A\frac{\partial g}{\partial \eta} + c_A = 0 \,. \tag{4.22}$$

In the stage of grid evolution, although the grid angle may change a lot, the area of the grid will still be the same. This allows the continuation of the computation in some cases. However, since the grid is not orthogonal, and therefore not optimal [1], the error in some computations may be large in some cases.

Another advantage is that, the idea can be applied to three dimensional flow calculations easily. Except the skewness-preserving condition proposed in [4], we may also use the condition

$$\frac{\partial \Delta}{\partial \lambda} = \frac{\partial}{\partial \lambda}\left[AMR + LQC + PBN - PMC - AQN - LBR \right] = 0 \,.$$

4.1.4 Weighted Preserving

As we have discussed above, orthogonality preserving grid may attain a higher accuracy in general, while area preserving grid may attain a higher robustness. Therefore, it is natural hoping to have both advantages by combining the solution of h from previous section. Now, we define $(\omega_O, \omega_A, \omega_L)$ to be real numbers so that

$$\omega_O + \omega_A + \omega_L = 1 \,, \tag{4.23}$$

denoting the contributions from orthogonality-preserving scheme, area-preserving scheme and from using the Lagrangian Coordinates ($h \equiv 1$) respectively. Having the equations (4.4), (4.11), (4.16) and/or (4.20), we consider

$$a^* \frac{\partial h^*}{\partial \xi} + b^* \frac{\partial h^*}{\partial \eta} + c^* h^* = 0\,, \tag{4.24}$$

with $a^* = \omega_O a_O + \omega_A a_A$, $b^* = \omega_O b_O + \omega_A b_A$ and $c^* = \omega_O c_O + \omega_A c_A$. This equation thus includes orthogonality-preserving scheme when $\omega_A = \omega_L = 0$, area-preserving scheme when $\omega_O = \omega_L = 0$ and the Lagrangian Coordinates when $\omega_O = \omega_A = 0$.

By introducing a new variable $g = \ln(h^* q)$, we have

$$a \frac{\partial g}{\partial \xi} + b \frac{\partial g}{\partial \eta} + c = 0\,. \tag{4.25}$$

After $g_{i,j}$ is obtained, we transform it back to the original variable h^* by using $h^* = e^{g_{i,j}}/q_{i,j}$. By choosing $h \in [0,1]$ and considering the contribution from the Lagrangian Coordinates, ω_L, we finally get

$$h_{i,j} = (\omega_O + \omega_A) \frac{h^*_{i,j}}{\max_{i,j} h^*_{i,j}} + \omega_L\,. \tag{4.26}$$

Although this gives us 2 more degree of freedoms (3 parameters ω_O, ω_A and ω_L with one constrain $\omega_O + \omega_A + \omega_L = 1$), we will only concentrate ourselves on the following combinations:

$(\omega_O, \omega_A, \omega_L)$	Interpretation
$(1,0,0)$	orthogonal grid (Chapter 4.1.1 and 4.1.2)
$(0,1,0)$	area preserving grid (Chapter 4.1.3)
$(0,0,1)$	Lagrangian grid
$(1/2,1/2,0)$	AO grid
$(0,1/2,1/2)$	AL grid
$(1/2,0,1/2)$	OL grid
$(1/3,1/3,1/3)$	AOL grid

It should be noticed that the idea of the weighted preserving scheme is to combine the advantages from orthogonality preserving and area preserving conditions. The contribution from the Lagrangian Coordinates is not necessary. From the numerical experiments, the grid generated by the Lagrangian Coordinates is highly distorted. Therefore, the last three weighted preserving schemes AL, OL and AOL in the chart should not be good combinations in general. The distortion property in the grid from the Lagrangian Coordinates passes to these weighted preserving schemes, which may be problematic in some cases.

4.2 The h-Equation Solver and the Improvements in its Efficiency

Given the weights $(\omega_O, \omega_A, \omega_L)$, we can obtain the h-equation written in the form

$$a\frac{\partial g}{\partial \xi} + b\frac{\partial g}{\partial \eta} + c = 0\,, \tag{4.27}$$

where $a = \omega_O a_O + \omega_A a_A$, $b = \omega_O b_O + \omega_A b_A$ and $c = \omega_O c_O + \omega_A c_A$.

We now discuss 3 approaches to solve this equation.

4.2.1 Method of Characteristics

Because the equation for g is a linear and homogeneous partial differential equation, it can be solved by method of characteristics. However, the main difficulty is that we do not know in advance the slopes of the characteristics lines- they are different in each time step due to their dependence on the flow. Although in simple geometry and simple flow situation, we can apply this method to solve for g, and therefore h, when the flow situation become complicated, we have to find other way to determine the value of the function h.

4.2.2 Unsteady Approach

A better method is to obtain the steady state solution of the unsteady 2-dimensional hyperbolic equation by adding one more pseudo-time dependent term to the equation, giving

$$\frac{\partial g}{\partial T} + a\frac{\partial g}{\partial \xi} + b\frac{\partial g}{\partial \eta} + c = 0 \tag{4.28}$$

for some pseudo-time T. We then discretize the equation using finite difference according to the characteristics and solve the equation using iteration until

$$\max_{i,j} ||g_{i,j}^{(n+1)} - g_{i,j}^{(n)}|| < \epsilon \tag{4.29}$$

for some torlerence $\epsilon > 0$. For example, we can obtain the following explicit iterative scheme

$$\begin{aligned} \frac{g_{i,j}^{(n+1)} - g_{i,j}^{(n)}}{\Delta T} &+ \frac{a_+}{\Delta\xi}(g_{i,j}^{(n)} - g_{i-1,j}^{(n)}) + \frac{a_-}{\Delta\xi}(g_{i+1,j}^{(n)} - g_{i,j}^{(n)}) \\ &+ \frac{b_+}{\Delta\eta}(g_{i,j}^{(n)} - g_{i,j-1}^{(n)}) + \frac{b_-}{\Delta\eta}(g_{i,j+1}^{(n)} - g_{i,j}^{(n)}) + c_{i,j} = 0\,, \end{aligned} \tag{4.30}$$

with

$$a_+ = \frac{a_{i,j} + |a_{i,j}|}{2}\,, a_- = \frac{a_{i,j} - |a_{i,j}|}{2}\,, b_+ = \frac{b_{i,j} + |b_{i,j}|}{2} \text{ and } b_- = \frac{b_{i,j} - |b_{i,j}|}{2}\,. \tag{4.31}$$

Or, the following point-implicit iterative scheme

$$\begin{aligned} \frac{g_{i,j}^{(n+1)} - g_{i,j}^{(n)}}{\Delta T} &+ \frac{1}{2\Delta\xi}[a_{i,j}(g_{i+1,j}^{(n)} - g_{i-1,j}^{(n)}) - |a_{i,j}|(g_{i+1,j}^{(n)} - g_{i-1,j}^{(n)}) + 2|a_{i,j}|g_{i,j}^{(n+1)}] \\ &+ \frac{1}{2\Delta\eta}[b_{i,j}(g_{i,j+1}^{(n)} - g_{i,j-1}^{(n)}) - |b_{i,j}|(g_{i,j+1}^{(n)} - g_{i,j-1}^{(n)}) + 2|b_{i,j}|g_{i,j}^{(n+1)}] \\ &+ c_{i,j} = 0\,. \end{aligned} \tag{4.32}$$

The initial value $g^{(0)}$ at $\lambda = \lambda^{(k+1)}$ is chosen to be the solution obtained at the previous time step, $\lambda = \lambda^{(k)}$. This may reduce the number of iteration steps, (n), needed for convergence.

Boundary Condition

The boundary values for this discretized equation can always be fixed to be a constant. The reason why we can fix the value on the boundary can be explained by considering the following easier 1-D version of the problem

$$\frac{\partial g}{\partial T} + a\frac{\partial g}{\partial \xi} + c = 0$$

or written in the discretized form

$$\frac{g_i^{(n+1)} - g_i^{(n)}}{\Delta T} + \frac{a_+}{\Delta \xi}(g_i^{(n)} - g_{i-1}^{(n)}) + \frac{a_-}{\Delta \xi}(g_{i+1}^{(n)} - g_i^{(n)}) + c_i = 0\,.$$

Assume the required solutions are g_i for $i = 1, 2, \cdots, m$. Now, we add two more cells g_0 and g_{m+1} act as the boundary cells and fixed the values to be a constant.

When a_1 is positive, a_+ would be equal to a_1 and a_- would reduce to be zero. We have the equation for $i = 1$:

$$\frac{g_1^{(n+1)} - g_1^{(n)}}{\Delta T} + \frac{a_1}{\Delta \xi}(g_1^{(n)} - g_0^{(n)}) + c_1 = 0\,,$$

where $g_1^{(n)}$ and $g_0^{(n)} = g_0$ are given and $g_1^{(n+1)}$ is the only unknown in the equation.

When a_1 is negative on the other hand, a_+ would be equal to zero and a_- would be set to a_1. We have the equation for $i = 1$:

$$\frac{g_1^{(n+1)} - g_1^{(n)}}{\Delta T} + \frac{a_1}{\Delta \xi}(g_2^{(n)} - g_1^{(n)}) + c_1 = 0\,.$$

Even if we fixed the value of g_0, it would not affect the value of the interior cell.

Similar argument can be applied to the cell $i = m$ and can be generalized to the above g-equation. Therefore, for simplicity in computation, we simply fix the values on the boundary to be a constant no matter the characteristic are in-coming from or out-going to the boundary.

Disadvantage

However, the problem of the unsteady approach is that we need to choose ΔT small enough so that the CFL condition is satisfied. When the flow is complicated, it takes much time for the method to converge. The second motivation of the search for another h-equation solver is that what we need is the steady state solution only. Once we get it, we do not care how the solution of $g_{i,j}$ evolves in T towards the steady state. Here, we proposed another solver.

4.2.3 Direct Solver

Instead of adding the pseudo-time depending term, we can directly solve the equation given by the same distretization as in (4.30)

$$\frac{a_+}{\Delta\xi}(g_{i,j} - g_{i-1,j}) + \frac{a_-}{\Delta\xi}(g_{i+1,j} - g_{i,j}) + \frac{b_+}{\Delta\eta}(g_{i,j} - g_{i,j-1}) + \frac{b_-}{\Delta\eta}(g_{i,j+1} - g_{i,j}) + c_{i,j} = 0\,,$$

or, re-written it in the form of

$$\left[\left\{\begin{array}{ccc} 0 & \dfrac{b_-}{\Delta\eta} & 0 \\ \dfrac{-a_+}{\Delta\xi} & \left(\dfrac{a_+}{\Delta\xi} - \dfrac{a_-}{\Delta\xi} + \dfrac{b_+}{\Delta\eta} - \dfrac{b_-}{\Delta\eta}\right) & \dfrac{a_-}{\Delta\xi} \\ 0 & \dfrac{-b_+}{\Delta\eta} & 0 \end{array}\right\}\right] g_{i,j} + c_{i,j} = 0\,, \tag{4.33}$$

or simply

$$\mathbf{M}\mathbf{g} = \mathbf{c}\,,$$

where $\mathbf{M}$ is a sparse matrix, $\mathbf{g}$ is an unknown vector and $\mathbf{c}$ is a vector determined by $c_{i,j}$ and the boundary values. Therefore, it becomes a problem of solving a system of linear equations, which can be solved by using Jacobi iteration, Gauss-Seidel iteration, SOR, GMRES or any other iterative methods.

The convergence of the iteration using Jacobi iteration is proven here. In order to prevent division-by-zero, the diagonal of the matrix $\mathbf{M}$ is replaced by adding

a very small number $\epsilon > 0$

$$\mathbf{M}_{i,i} = \frac{a_+}{\Delta\xi} - \frac{a_-}{\Delta\xi} + \frac{b_+}{\Delta\eta} - \frac{b_-}{\Delta\eta} + \epsilon = \frac{|a|}{\Delta\xi} + \frac{|b|}{\Delta\eta} + \epsilon = |\mathbf{M}_{i,i}|\,,$$

while

$$\sum_{j=1, i\neq j}^{n} |\mathbf{M}_{i,j}| = \left|\frac{a_+}{\Delta\xi}\right| + \left|\frac{a_-}{\Delta\xi}\right| + \left|\frac{b_+}{\Delta\eta}\right| + \left|\frac{b_-}{\Delta\eta}\right| = \frac{|a|}{\Delta\xi} + \frac{|b|}{\Delta\eta} < |\mathbf{M}_{i,i}|\,.$$

Hence, the matrix $\mathbf{M}$ is diagonally dominant, and this guarantees the convergence of using Jacobi iterations.

4.3 Riemann Problem Solver and the Improvements in its Efficiency

As described in section 3.2, the flow variables (ρ, p, u, v) at the interface are determined by doing iterations. This type of Exact Riemann solver is long regarded to be expansive. Especially when the flow and the geometry are simple, exact solver may be replaced by other type of schemes which are easy to use, and more importantly, take less time to solve the problem. From Chapter 3.2, we note that the Riemann solution in the Unified Coordinates is the same as the one in the Eulerian Coordinates, except that the expansion fan solution in the former one considered the movement of the grid. Therefore, replacement of this Riemann solver becomes possible. By substituting Step 2(c) in the algorithm given in Chapter 3 by the following schemes, we try to see whether the efficiency of the Unified Code can be improved, while still attaining a more accurate solution than that from the same scheme in the Eulerian Coordinates.

4.3.1 Collisional BGK Method

Other than the Exact Riemann Solver, one of the possibilities of splitting the computation of the flux $\mathbf{F}^{k+\frac{1}{2}}_{i+\frac{1}{2},j}$ is as follow:

1. Determination of $\mathbf{Q}^*$ at $\xi = 0$ using $\mathbf{Q}_l$ and $\mathbf{Q}_r$ as if the grids are fixed;
2. Determination of $\mathbf{Q}$ at $\xi = 0$ using $\mathbf{Q}^*$ by considering the contributions from the movement of the grid.

Step 1 here can be done by using the Collisional BGK scheme [14]. Instead of describing the flow motion based on macroscopic quantities, such as mass, momentum and energy, we now turn to another type of description which comes from microscopic considerations, i.e. the gas kinetic theory. Therefore, the flow variables at the interface are given by

$$\begin{pmatrix} \rho \\ \rho U \\ \rho e \end{pmatrix}_{j+\frac{1}{2}} = \rho_i \begin{pmatrix} < u^0 >_{>0} \\ < u^1 >_{>0} \\ < u^2 >_{>0} \end{pmatrix} + \rho_{i+1} \begin{pmatrix} < u^0 >_{<0} \\ < u^1 >_{<0} \\ < u^2 >_{<0} \end{pmatrix} \tag{4.34}$$

with

$$< u^0 >_{>0} = \frac{1}{2}\mathrm{erfc}(-\sqrt{\lambda_l}u_l) \tag{4.35}$$

$$< u^0 >_{<0} = \frac{1}{2}\mathrm{erfc}(\sqrt{\lambda_r}u_r) \tag{4.36}$$

$$< u^1 >_{>0} = u_l < u^0 >_{>0} + \frac{1}{2}\frac{\exp(-\lambda_l u_l^2)}{\sqrt{\pi\lambda_l}} \tag{4.37}$$

$$< u^1 >_{<0} = u_r < u^0 >_{<0} - \frac{1}{2}\frac{\exp(-\lambda_r u_r^2)}{\sqrt{\pi\lambda_r}} \tag{4.38}$$

$$< u^2 >_{>0} = u_l < u^1 >_{>0} + \frac{< u^0 >_{>0}}{2\lambda_l} \tag{4.39}$$

$$< u^2 >_{<0} = u_r < u^1 >_{<0} + \frac{< u^0 >_{<0}}{2\lambda_r}\,, \tag{4.40}$$

and $\lambda = \rho/(2p)$.

Now, we have to encounter the motion of the interface due to the motion of the grid. In the moving frame of reference, we define $(\rho,\, \rho U,\, \rho e)^*_{j+\frac{1}{2}}$ to be the density, momentum and energy respectively and $w_{j+1/2}$ to be the velocity of the grid movement. Because the particles must have a velocity larger than $u + w_{j+1/2}$ in order the pass through the interface moving from the cell x_j to x_{j+1} and the particles in the cell x_{j+1} must have a velocity greater than the same velocity as above, i.e. $u + w_{j+1/2}$, in order to stay in the cell. We have the following relations

$$\begin{pmatrix} \rho \\ \rho U \\ \rho e \end{pmatrix}_{j+1/2} = \iint_{u^*>0} \begin{pmatrix} 1 \\ u \\ \frac{1}{2}(u^2+\xi^2) \end{pmatrix} g^*_j du^* d\xi \tag{4.41}$$

$$+ \iint_{u^*<0} \begin{pmatrix} 1 \\ u \\ \frac{1}{2}(u^2+\xi^2) \end{pmatrix} g^*_{j+1} du^* d\xi\,, \tag{4.42}$$

with $u^* = u - w_{j+1/2}$. Now, the above relation can be simplified to

$$\begin{pmatrix} \rho \\ \rho U \\ \rho e \end{pmatrix}_{j+\frac{1}{2}} = \begin{pmatrix} \rho \\ \rho u + w_{j+1/2}\rho \\ \rho e + w_{j+1/2}\rho u + \frac{1}{2}w^2_{j+1/2}\rho \end{pmatrix}^*_{j+1/2}\,. \tag{4.43}$$

In summary, we first obtain $(\rho, \rho u, \rho e)^*_{j+1/2}$ by using (4.34) with $u_l = u_j - w_{j+1/2}$, $u_r = u_{j+1} - w_{j+1/2}$ and

$$w_{j+1/2} = \frac{h_j u_j + h_{j+1} u_{j+1}}{2}\,, \tag{4.44}$$

and then transform to $(\rho, \rho u, \rho e)_{j+1/2}$ by using (4.43).

4.3.2 NT Scheme

The Lax-Friedrich scheme (LxF) [8] is a prototype of a central difference approximation, which offers a great simplicity over the upwind Godunov scheme, i.e. no

Riemann Problems are solved. Based on the staggered form of the LxF scheme,

$$v_{j+1/2}(t+\Delta t) = \frac{1}{2}(v_j + v_{j+1}) - \alpha[f(v_{j+1}) - f(v_j)], \tag{4.45}$$

Nessyahu and Tadmor [9] proposed a family of schemes which takes an easily implemented predictor-corrector form, welly known as NT schemes. As a natural extension of this one-dimensional second-order central scheme, Jiang and Tadmor [7] constructed a non-oscillatory high-resolution scheme for two-dimensional hyperbolic conservation laws. In this section, we will try to implement this scheme for our hyperbolic system (2.9), given by

$$\frac{\partial \mathbf{E}}{\partial \lambda} + \frac{\partial \mathbf{F}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} = 0\,. \tag{4.46}$$

To approximate this equation, we begin with a piecewise constant solution of the form $\sum \mathbf{E}^n_{pq}\psi_{pq}(\xi,\eta)$. Here $\mathbf{E}^n_{pq}$ is the approximate cell average at $\lambda = \lambda^n$, associated with the cell $C_{pq} = I_p \times J_q$ centered around $(\xi_p = p\Delta\xi, \eta_q = q\Delta\eta)$, i.e.

$$C_{pq} := \left\{ (x,y) : |x-\xi_p| \le \frac{\Delta\xi}{2}, |y-\eta_q| \le \frac{\Delta\eta}{2} \right\}.$$

As a first step, we use Taylor expansion to express

$$\mathbf{E}(\xi_j, \eta_k, \lambda^{n+1/2}) = \mathbf{E}^n_{jk} + \frac{\Delta\lambda}{2}\mathbf{E}(\xi_j, \eta_k, \lambda^n) + O(\Delta\lambda^2)\,, \tag{4.47}$$

or, we write

$$\mathbf{E}(\xi_j, \eta_k, \lambda^{n+1/2}) = \mathbf{E}(\xi_j, \eta_k, \lambda^n) - \frac{\alpha}{2}\mathbf{F}(\mathbf{E})'_{jk} - \frac{\mu}{2}\mathbf{G}(\mathbf{E})^{\backprime}_{jk}\,, \tag{4.48}$$

with

$$\alpha = \frac{\Delta\lambda}{\Delta\xi} \text{ and } \mu = \frac{\Delta\lambda}{\Delta\eta}\,,$$

where $(\cdot)'$ and $(\cdot)^{\backprime}$ are discrete slopes in the $\xi-$ and $\eta-$ directions respectively, which are reconstructed from the given cell averages. For example, we use the following limiter,

$$\begin{aligned}
(\cdot)'_{j,k} &= MM\left\{\theta[(\cdot)^n_{j+1,k} - (\cdot)^n_{j,k}], \frac{1}{2}[(\cdot)^n_{j+1,k} - (\cdot)^n_{j-1,k}], \theta[(\cdot)^n_{j,k} - (\cdot)^n_{j-1,k}]\right\}, \\
(\cdot)^{\backprime}_{j,k} &= MM\left\{\theta[(\cdot)^n_{j,k+1} - (\cdot)^n_{j,k}], \frac{1}{2}[(\cdot)^n_{j,k+1} - (\cdot)^n_{j,k-1}], \theta[(\cdot)^n_{j,k} - (\cdot)^n_{j,k-1}]\right\},
\end{aligned} \tag{4.49}$$

where

$$MM\{v1, v2, \cdots\} = \begin{cases} \min_p\{v_p\} & \text{if} \quad v_p > 0 \,\forall p \\ \max_p\{v_p\} & \text{if} \quad v_p < 0 \,\forall p \\ 0 & \text{otherwise} \end{cases}$$

Here, the choice $\theta = 1$ coincides with the classical MinMod limiter. The larger the value of θ, the more numerical viscosity is reduced [7]. Therefore, having the mid-values, we conclude with new staggered averages at $\lambda = \lambda^{n+1}$, given by

$$\begin{aligned} \mathbf{E}^{n+1}_{j+1/2,k+1/2} &= \frac{1}{2}\left(\mathbf{E}_{j,k} + \mathbf{E}_{j+1,k} + \mathbf{E}_{j,k+1} + \mathbf{E}_{j+1,k+1}\right) \\ &+\frac{1}{16}\left(\mathbf{E}'_{j,k} - \mathbf{E}'_{j+1,k}\right) - \frac{\alpha}{2}\left[\mathbf{F}(\mathbf{E}^{n+1/2}_{j+1,k}) - \mathbf{F}(\mathbf{E}^{n+1/2}_{j,k})\right] \\ &+\frac{1}{16}\left(\mathbf{E}'_{j,k+1} - \mathbf{E}'_{j+1,k+1}\right) - \frac{\alpha}{2}\left[\mathbf{F}(\mathbf{E}^{n+1/2}_{j+1,k+1}) - \mathbf{F}(\mathbf{E}^{n+1/2}_{j,k+1})\right] \\ &+\frac{1}{16}\left(\mathbf{E}^{`}_{j,k} - \mathbf{E}^{`}_{j,k+1}\right) - \frac{\mu}{2}\left[\mathbf{G}(\mathbf{E}^{n+1/2}_{j,k+1}) - \mathbf{G}(\mathbf{E}^{n+1/2}_{j,k})\right] \\ &+\frac{1}{16}\left(\mathbf{E}^{`}_{j+1,k} - \mathbf{E}^{`}_{j+1,k+1}\right) - \frac{\mu}{2}\left[\mathbf{G}(\mathbf{E}^{n+1/2}_{j+1,k+1}) - \mathbf{G}(\mathbf{E}^{n+1/2}_{j+1,k})\right] . \end{aligned} \tag{4.50}$$

In summary, this scheme composes of a simple two-step predictor-corrector scheme (4.47) and (4.50). Starting with the cell averages, $\mathbf{E}^n_{j,k}$, we first use the first order predictor (4.47) for the evaluation of the midpoint values, $\mathbf{E}^{n+1/2}_{j,k}$, which is followed by the second-order corrector (4.50) for the computation of the new cell average, $\mathbf{E}^{n+1}_{j,k}$. This results in a second-order accurate non-oscillatory scheme. As in the one-dimensional case, [9], no Riemann solvers are involved. This may speed up the computation time needed, but the slipline may be smeared badly.

4.3.3 Hybrid Riemann Solver

Based on the property of the Riemann solution from Chapter 3.2, we can split the computation of the flux $\mathbf{F}^{k+\frac{1}{2}}_{i+\frac{1}{2},j}$ into 2 steps:

1. Determination of p^* using $\mathbf{Q}_l$ and $\mathbf{Q}_r$;

2. Determination of $\mathbf{Q}$ at $\xi = 0$ using $\mathbf{K}$ and h, if necessary.

Therefore, step 1 here can be done by using any type of Riemann solvers, like the Exact Riemann solver proposed in Chapter 3 or some other linearized Riemann solvers. Proposed by Toro [10, 11, 12], we may use linearized Riemann Solvers to obtain the $*$-state solution. According to the idea, we use all together the Primitive Variable Riemann Solver (PVRS), the Two-Rarefraction Riemann Solver (TRRS) and the Two-Shock Riemann Solver (TSRS) in determining the $*$-state pressure.

The PVRS is a very simple linearized solution to the Riemann problem in the primitive variables p, ρ and u, i.e.

$$\mathbf{W}_\lambda + \mathbf{A}(\mathbf{W})\mathbf{W}_\xi = 0\,. \tag{4.51}$$

The difficulty in solving this equation is due to the fact that the matrix $\mathbf{A}(\mathbf{W})$ depends on the solution vector $\mathbf{W}$. Therefore, we approximate the solution by using the average $\mathbf{W}$ of $\mathbf{W}_L$ and $\mathbf{W}_R$ and obtain

$$\mathbf{W}_\lambda + \bar{\mathbf{A}}\mathbf{W}_\xi = 0\,, \tag{4.52}$$

where $\bar{\mathbf{A}} = \mathbf{A}(\bar{\mathbf{W}})$ becomes a matrix with constant coefficients. This equation can be solved exactly and the pressure in the *-state is given by

$$p_* = \frac{1}{2}(p_l + p_r) + \frac{1}{8}(u_L - u_R)(p_L + p_R)(a_L + a_R)\,, \tag{4.53}$$

where $a_{[\cdot]} = \sqrt{\gamma p_{[\cdot]}/\rho_{[\cdot]}}$.

The TRRS and the TSRS are based, on the other hand, on the approximations based on the Exact Riemann solver. If we assume the solution of the Riemann problem contains two rarefaction waves, the pressure in the *-state can be calculated to be

$$p_* = \left[\frac{a_L + a_R - \dfrac{\gamma - 1}{2}(u_R - u_L)}{\dfrac{a_L}{p_L^z} + \dfrac{a_R}{p_R^z}}\right]^{\frac{1}{z}}\,, \tag{4.54}$$

where $z = (\gamma - 1)/(2\gamma)$.

If both non-linear waves from the Riemann problem are assumed to be shock wave, the pressure in the *-state can be approximated by

$$p_* = \frac{\left[\frac{A_L}{p_{PVRS} + B_L}\right]^{1/2} p_L + \left[\frac{A_R}{p_{PVRS} + B_R}\right]^{1/2} p_R - (u_R - u_L)}{\left[\frac{A_L}{p_{PVRS} + B_L}\right]^{1/2} + \left[\frac{A_R}{p_{PVRS} + B_R}\right]^{1/2}}, \tag{4.55}$$

with

$$A_{[\cdot]} = \frac{2}{(\gamma + 1)\rho_{[\cdot]}} \text{ and } B_{[\cdot]} = \left(\frac{\gamma - 1}{\gamma + 1}\right) p_{[\cdot]}.$$

The determination of the *-state solution of pressure, and therefore the velocity u_*, relay on the following conditions: The PVRS is used when the following two conditions are met:

$$Q = \frac{p_{\max}}{p_{\min}} < Q_{\max} \text{ and } p_{\min} < p_{PVRS} < p_{\max}, \tag{4.56}$$

where $p_{\max} = \max(p_L, p_R)$ and $p_{\min} = \min(p_L, p_R)$. As proposed by Toro, $Q_{\max} = 2$ seems to be a good choice and is recommended. The first condition here ensures that the pressures in the two sides are not widely different. Otherwise, we will use the TRRS or TSRS instead. The other condition is a condition on the difference in the velocities u_L and u_R and excludes the two-rarefaction and the two-shock cases, where the solution from the PVRS is unreliable. To use the TRRS or the TSRS, we check whether $p_{PVRS} > p_{\min}$ or not. If so, we expect strong shocks will probably happen and therefore, we choose the TSRS. Otherwise, we use the TRRS to approximate the *-state pressure.

This adaptive Riemann solver has an advantage that all the components involve non-iterative scheme. We do not need to do any iteration procedure like what we did in the Exact Riemann solver. However, this hybrid scheme depends on the user's choice of the constant $Q_{\max}$ appearing in equation (4.56). However, this switching parameter may seriously affect the solution in the cases where the flow is complicated.

4.4 Composite Operator Approach

Comparing with the Eulerian computation, i.e. $h \equiv 0$, the extra time we need to spend on using the Unified Coordinates is the computations of solving the h-equation (4.27). Several more iterations are needed to obtain the value of h, in order to have the velocities of the moving of grid points so that one of the conditions is satisfied on section 4.1. In this section, we try to think of a strategy so that we can reduce the total time needed to obtain the steady state solution in the Unified Coordinates, thus, increasing the efficiency of the Unified Code.

From our computational experiences in steady flow, we found that the grid will first move in the wrong direction due to the fact that the initial flow condition is different from the steady flow solution. But the movement of the grids corrects the flow gradually. As a result, instead of using the Unified Code starting the computation from the very first beginning, we can use *a better initial flow condition* to start our computation in the Unified Coordinates.

Defining E_{λ_0} to be the operator of using the Eulerian Coordinates from $\lambda = 0$ to $\lambda = \lambda_0$, U_{λ_0} to be the operator of using the Unified Coordinates for λ_0. This operator U will be written as O if the function h is determined using orthogonality preserving condition, written as A if we choose area preserving h and written as (AO) if we use the weighted preserving h as discussed in Chapter 4.1.4. As a result, we use the composite operations $U_{\lambda_1} E_{\lambda_0}$ to do our computations. This means we first use the Eulerian Coordinates to do our computation for λ_0, then follow by the Unified Coordinates for λ_1 more time.

Before the switching of the coordinates, the flow is approximately steady and this can be interpreted as an initial condition which is closer to the exact solution of the flow. The importance of this idea is this: All the steady state solutions, regardless on the computation schemes, from the Eulerian Coordinates can be regarded as an initial condition of the flow, and can be improved by our

Unified Coordinates system.

Most of our computational experiments in steady flow given here are carrying out first by E_{λ_0} for some time λ_0 so that the solution converges in the Eulerian Coordinates, and then by U_{λ_1} in order to obtain a better resolution of the solutions.

Another way to reduce the computation time needed by using the same idea is to try the combination $E_{\lambda_1}U_{\lambda_0}$. We first use the Unified Coordinates for λ_0, followed by λ_1 of the Eulerian Coordinates. We noted that the grid will not change much after some time of computation in the Unified Coordinates. Therefore, we try to see whether the computational time on solving the h-equation can be saved or not when the structure of the grid is formed, we switch the computation back to the Eulerian Coordinates, i.e. $h \equiv 0$, which avoids the iterations required for obtaining the values of h at different grid points. By doing this, the Unified Coordinates can be regarded as a grid-generator, depending on both the flow and the geometry of the problem, instead of the geometry alone, as with most grid generations.

In summary, in this section, we have introduced two ideas on how to use, and when to use the Unified Coordinates for steady flow computations. The computations involve two composite steps, $U_{\lambda_1}E_{\lambda_0}$ (Eulerian followed by Unified) or $E_{\lambda_1}U_{\lambda_0}$ (Unified followed by Eulerian). The idea of $U_{\lambda_1}E_{\lambda_0}$ is to give a better initial condition for Unified Coordinates, while that of $E_{\lambda_1}U_{\lambda_0}$ is to stop solving the h-equation, hence saving time, when the structure of the grids becomes steady.

Chapter 5

The Use of the Entropy Equation

Sudden compression of gas flow due to shock reflection from a solid wall or sudden expansion due to an abrupt withdrawal of a piston from the gas are often associated with a phenomenon known generically as wall heating. As explained by Hui and Kudriakov in [6] recently, these problems are singular in that the contact line (or the wall) changes its direction abruptedly at some point at the boundary or in the interior of the flow field, resulting in a geometric singular point there. As a result, the Riemann solution is multi-valued at that point and varies rapidly around that point. This causes difficulties in any shock-capturing scheme. As proposed by [6], in order to eliminate the contact overheating, we must avoid the entropy error generated due to cell-averaging, especially near the singular point.

When the flow is smooth, we can replace the energy conservation equation, the fourth equation in (2.7), by the entropy conservation equation in the Eulerian Coordinates

$$\frac{\partial(\rho S)}{\partial t} + \frac{\partial(\rho u S)}{\partial x} + \frac{\partial(\rho v S)}{\partial y} = 0\,, \tag{5.1}$$

or in the Unified Coordinates

$$\frac{\partial(\rho \Delta S)}{\partial \lambda} + \frac{\partial[\rho(1-h)IS]}{\partial \xi} + \frac{\partial[\rho(1-h)JS]}{\partial \eta} = 0\,, \tag{5.2}$$

with $S = p/\rho^{\gamma}$. Physically, this means the entropy is constant following a fluid

particle, i.e. S =constant along a pathline. When the flow is steady, the pathline coincides with the streamline. Therefore, in the Unified Coordinates, we have

$$\frac{\partial S}{\partial \lambda} = 0 \tag{5.3}$$

when the flow is smooth. This implies S is constant in any grid points with time.

In the Godunov scheme, the update of flow variables from one time level to the next consists of two steps: (i) solving Riemann problems for every pair of adjacent cells and (ii) cell-averaging of the conserved variables. The Riemann problem is solved exactly and there is no spurious entropy generation in this step no matter we use the energy equation or the entropy equation when the flow is smooth. However, spurious entropy is generated due to cell-averaging of the conserved variables.

Therefore, to implement the replacement of the energy equation, with the above assumptions that the flow is smooth and steady, we can keep all the numerical procedures as discussed in section 3.1, except with a small modification in Step 4(b). When we decode $E_{p_{i,j}}^{k+1}$ to get $Q_{i,j}^{k+1}$, the energy equation

$$\frac{\partial(\rho \Delta e)}{\partial \lambda} + \frac{\partial[\rho(1-h)Ie + pI]}{\partial \xi} + \frac{\partial[\rho(1-h)Je + pJ]}{\partial \eta} = 0$$

will be ignored and be replaced by the entropy condition. Therefore, the pressure at the $(k+1)$-step is determined by the density at that grid point using

$$p^{(k+1)} = p^{(k)} \left(\frac{\rho^{(k+1)}}{\rho^{(k)}} \right)^{\gamma} . \tag{5.4}$$

This means the solution is theoretically fitted there so that the entropy S is conserved exactly, including the singular point region which is problematic if we use the energy equation.

Chapter 6

Test Examples

In this chapter, several test examples will be studied in details. The Unified Coordinates will be applied to the flow problems to see whether the solution obtained is better than that from the Eulerian Coordinates. The methods proposed in Chapter 4 and 5 will also be used to improve the efficiency and the robustness.

6.1 Two Dimensional Steady Riemann Problem

6.1.1 The Problem

In this section, we will consider a two dimensional steady Riemann problem generated by two uniform parallel flows as

$$(p, \rho, M, \theta) = \begin{cases} (0.25, 0.5, 7, 0) & , \quad y > 0 \\ (1, 1, 2.4, 0) & , \quad y < 0 \end{cases}$$

where M is the Mach number and θ is the flow angle defined as $\tan^{-1}(v/u)$. The flow contains a shock wave, a slipline and an expansion wave, as shown in figure 6.1.

This problem is difficult in a way that the stationary slipline is sensitive to the dissipative property of the numerical methods and the orientation of the grids. As

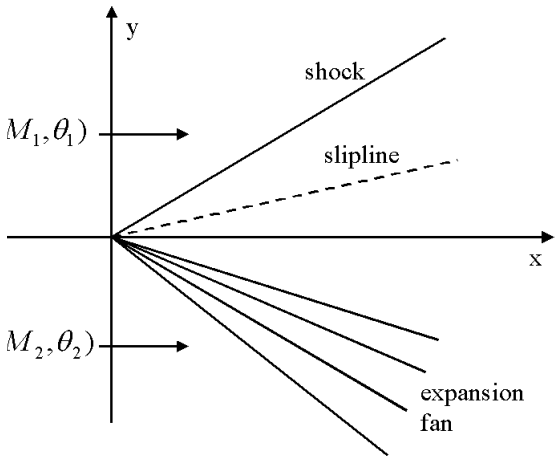

Figure 6.1: Two dimensional steady Riemann Problem.

suggested in [14, 15], the density fluctuation around the slipline may be due to the kinetic energy loss in the cell-averaging stage. This loss in kinetic energy has been transferred into thermal energy and this increases the pressure which pushes the gas away around the slip line. Therefore, a density sink is artificially formed. This phenomenon can be removed only if we can find a way to improve the computational grid in such a way that the slipline coincides with the cell interface. By doing this, we can reduce the kinetic energy loss due to cell averaging. A similar analysis is given here in Appendix A.

In the computations here in this section, the physical domain will change with time according to the pseudo-particle's velocity $h\mathbf{q}$. Therefore, if we follow the computational cells, they will move out of the initial physical domain, if $h \not\equiv 0$, and it would be difficult to have a steady state of flow in the original physical domain. To avoid this, we have used a technique called the "motionless viewing window" as in the classical Lagrangian method. Accordingly, the column of cells which have moved out of the original physical domain to the opposite direction of the incoming flow is deleted, while a new column of cells is added at the input flow boundary.

6.1.2 Exact Riemann Solver

In this subsection, we will use the Exact Riemann Solver to do the computations first. Figure 6.2 shows the density distributions obtained by different choices of coordinates. The x-axis of each graph represents the ratio (y/x) in the two dimensional space. Because the solution is self-similar under the transform (y/x), the solution of this problem can be obtained exactly, which is drawn by the solid line. The computed solution is plotted by circles. Figure 6.2(a) represents the solution obtained by the Eulerian Coordinates, i.e. using fixed and Cartesian grid. As what have explained, a density sink is formed near the slipline due to the averaging in those cells containing the strong contact wave. Figure 6.2(b) is obtained by using orthogonality preserving grid in the Unified Coordinates. As we can see, the sink is now removed and the slipline is resolved within two to three cells. Although the expansion fan is smeared a little bit, the solution is now improved quite a lot. The main reason why the extra-dissipation near the slipline can be removed is due to the fact that the slipline coincides with a gridline, figure 6.3(a). So, the problem of averaging is minimized.

As proposed in Section 4.1, other type of h-equation constrains are also possible. Figure 6.2(d) shows the solution when we use one of the weighted-preserving h, the AO grid solver. The grid obtained is intermediate between the orthogonality preserving and area preserving. We see that from figure 6.3(b), the gridline also coincides with the slipline, and therefore, the density sink can also be removed. With comparing the solutions using the O-grid and Lagrangian grid (figure 6.2(e)), the slipline in AO-grid can be resolved very sharply, like in the Lagrangian Coordinates. But, the grid in the AO case shows a much better regularity than that in Lagrangian case (figure 6.3(c)).

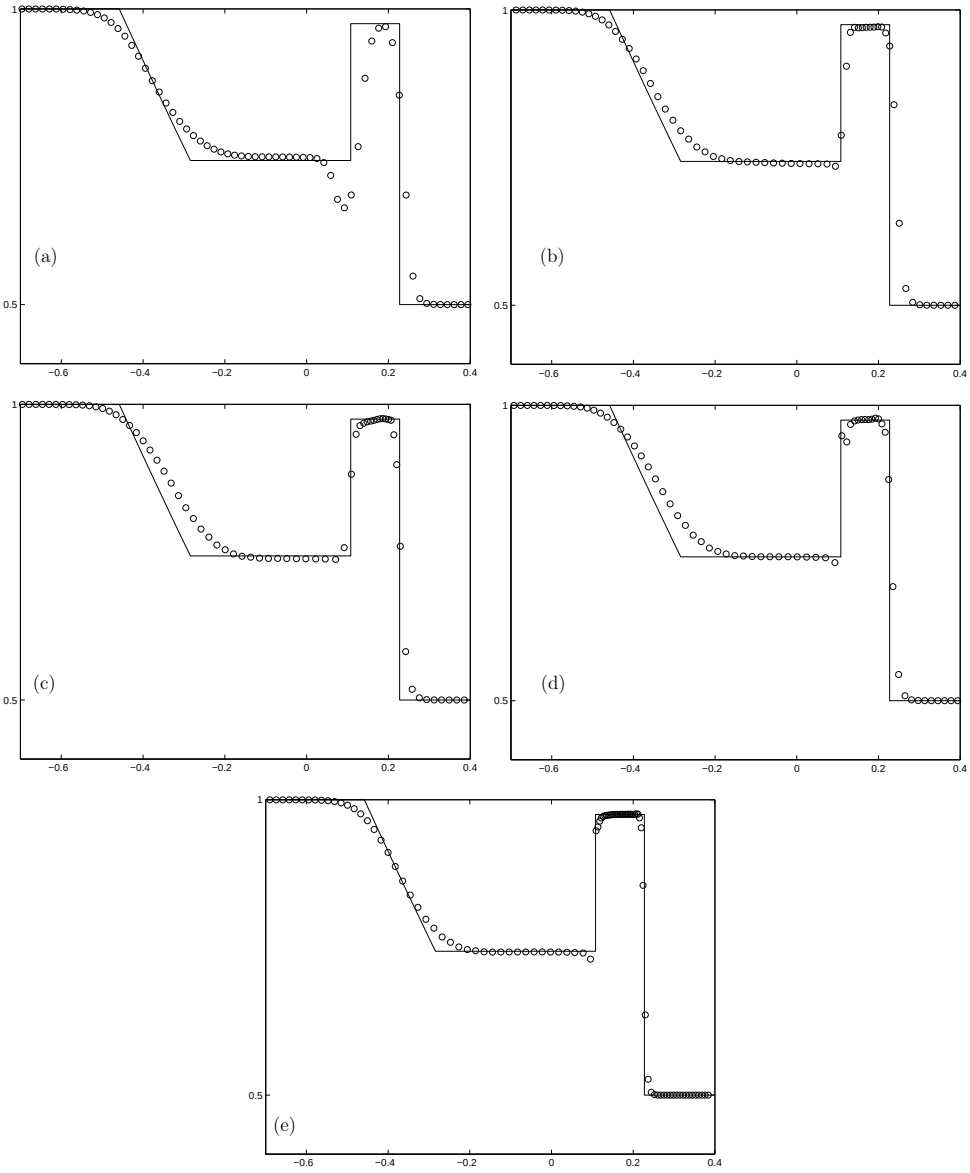

Figure 6.2: Density distribution of the two dimensional steady Riemann Problem using (a) the Eulerian Coordinates ($h \equiv 0$), the Unified Coordinates with (b) orthogonality preserving grid (O-grid), (c) area preserving grid (A-grid), (d) AO-grid and (e) the Lagrangian Coordinates ($h \equiv 1$).

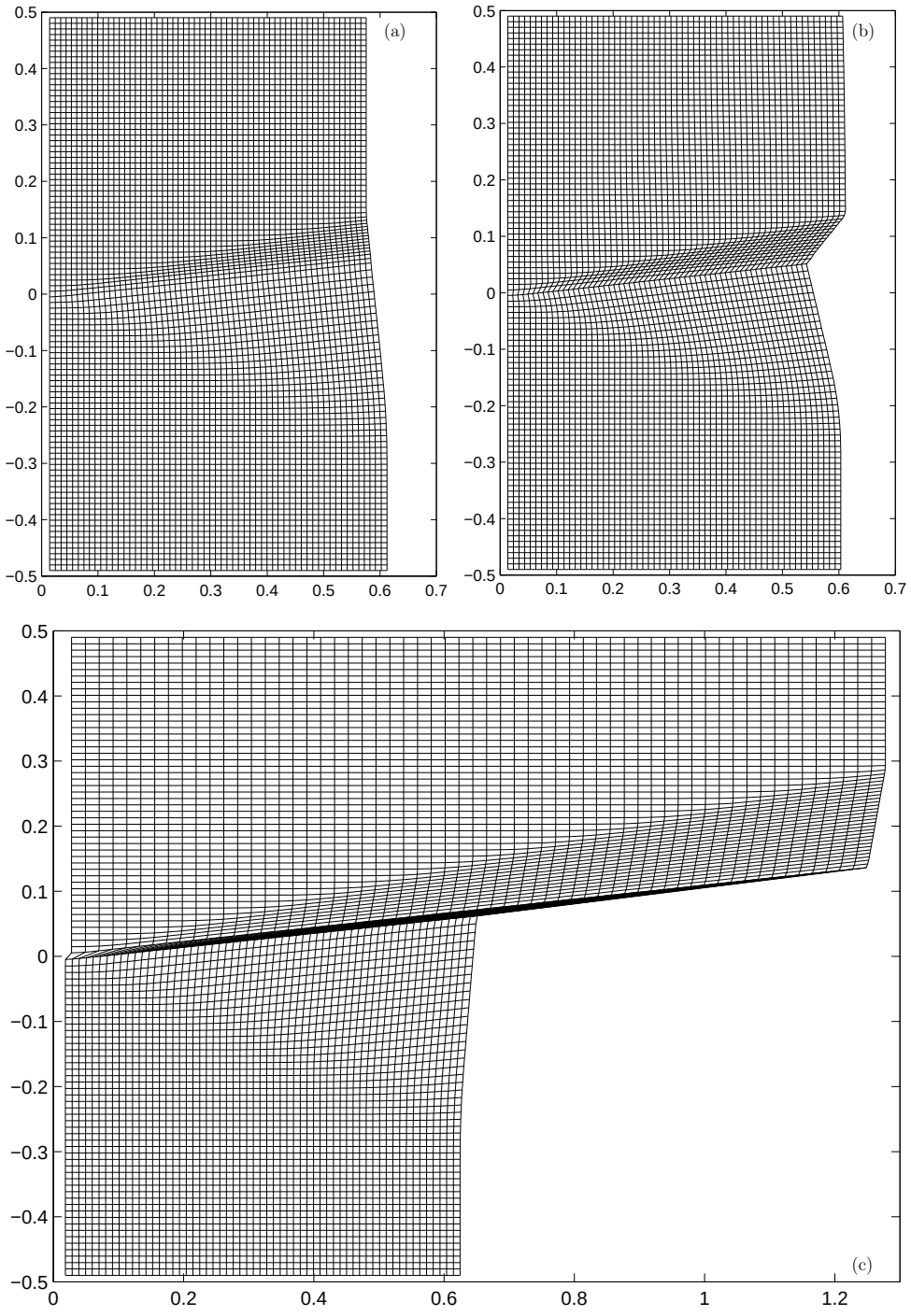

Figure 6.3: Flow-generated grid of the two dimensional steady Riemann Problem using the Unified Coordinates with (a) orthogonality preserving grid (O-grid), (b) AO-grid and (c) the Lagrangian Coordinates ($h \equiv 1$).

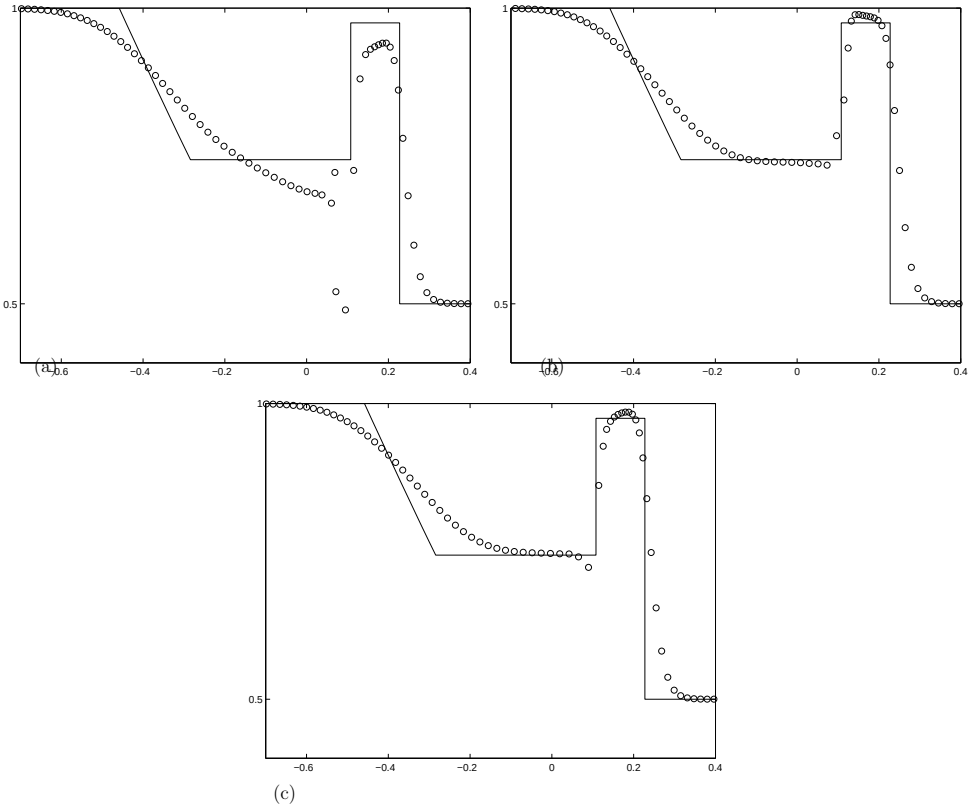

Figure 6.4: Density distribution of the two dimensional steady Riemann Problem using first order Collisional BGK Scheme in (a) the Eulerian Coordinates ($h \equiv 0$) and the Unified Coordinates with (b) orthogonality preserving grid (O-grid) and (c) AO-grid.

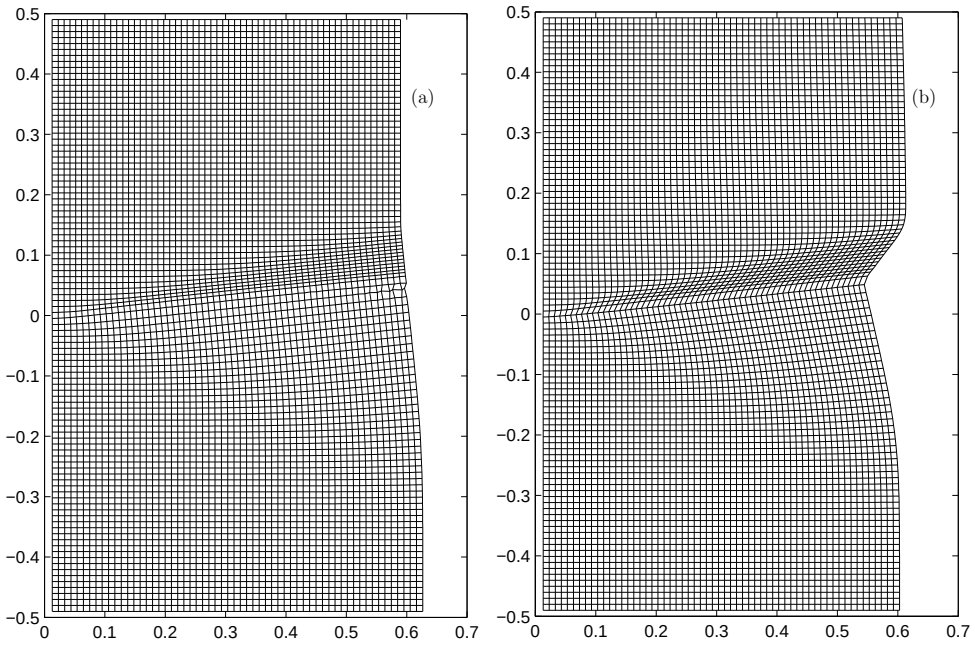

Figure 6.5: Flow-generated grid of the two dimensional steady Riemann Problem using first-order Collisional BGK Scheme in the Unified Coordinates with (a) orthogonality preserving grid (O-grid) and (b) AO-grid.

6.1.3 Collisional BGK Scheme

One way to improve the efficiency of the Riemann Problem solver, as discussed in Section 4.3, is to replace the Exact Riemann Solver by other type of scheme, when the flow situation is relatively simple. Introduced in Section 4.3.1, we have tried to use the collisional BGK method for this test case. Figure 6.4(a) shows the solution using first order collisional BGK scheme. From the graph, we see that the expansion fan is much more smeared than the one we obtained using the Exact Riemann Solver (figure 6.2(a)). The density sink is more serious here. The error near the slipline is increased from around 10% (Exact Riemann Solver) to now around 30%. When the orthogonality preserving grid, or the AO grid, is used, not only the density sink can be removed, the constant state immediate after the shock and the constant state immediate after the slipline are obtained qualitatively correctly, figure 6.4(c). Of course, the resolution is not as good as in the previous cases (using the Exact Riemann Solver) because we have used the MUSCL update, which is regarded as second order, in the former cases but not here in the collisional BGK scheme.

6.1.4 Hybrid Riemann Solver

As suggested in Chapter 4.3.3, we now use the proposed hybrid Riemann solver to see how this simple scheme can be improved by using the Unified Coordinates. The solution from the Eulerian Coordinates is given in figure 6.6(a). With comparing to the one obtained using the Exact Riemann Solver, both solutions are quantitatively the same. With considering the reduction in the computation time needed using this cheap solver, the solution is surprisingly good.

When the Unified Coordinates is used here, the grid generated and the flow solution are basically the same as what we obtained using the Exact Riemann Solver.

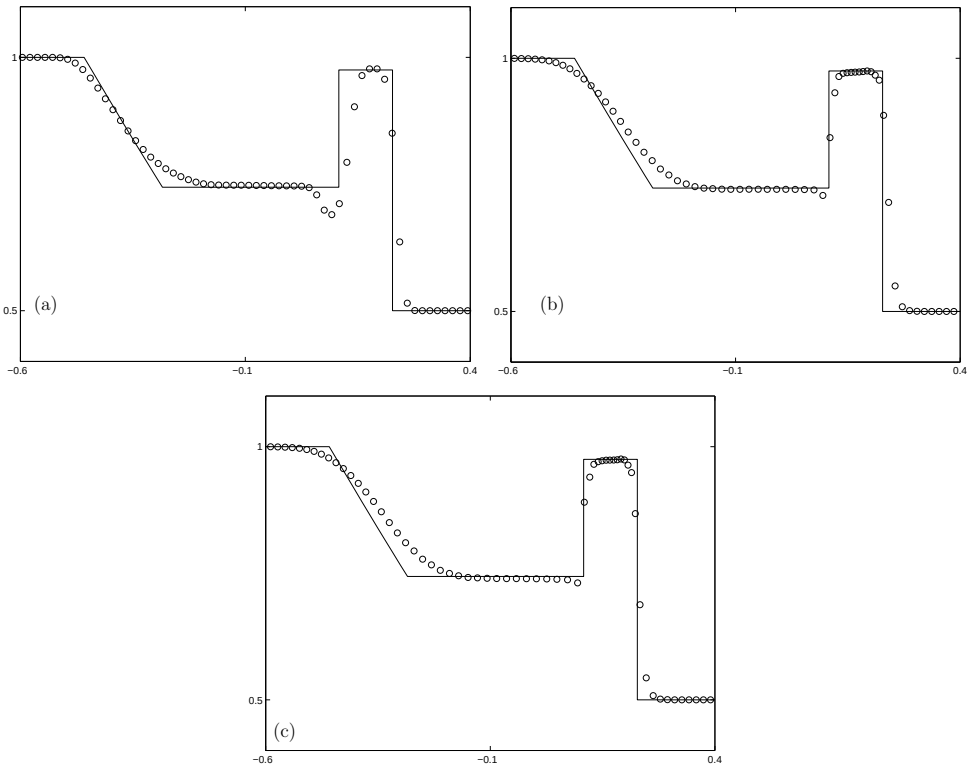

Figure 6.6: Density distribution of the two dimensional steady Riemann Problem using Hybrid Riemann Solver proposed by Toro [12] in (a) the Eulerian Coordinates ($h \equiv 0$) and the Unified Coordinates with (b) orthogonality preserving grid (O-grid) and (c) AO-grid.

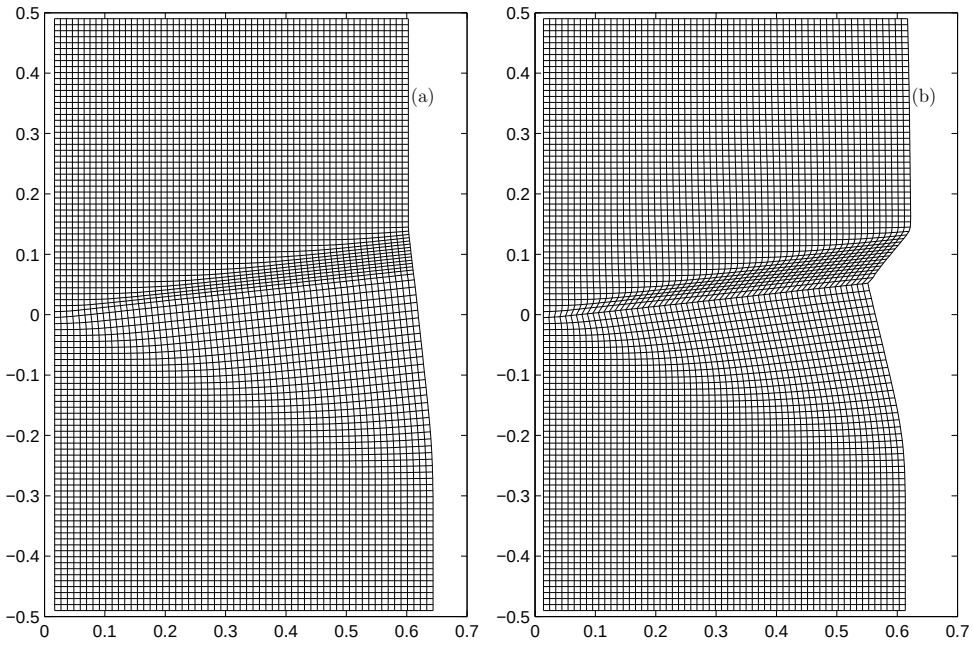

Figure 6.7: Flow-generated grid of the two dimensional steady Riemann Problem using Hybrid Riemann Solver proposed by Toro [12] in the Unified Coordinates with (a) O-grid and (b) AO-grid.

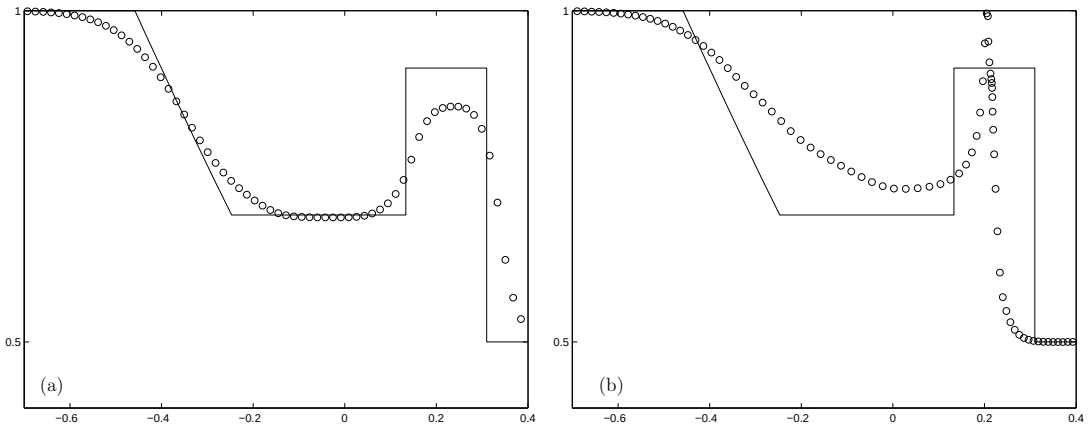

Figure 6.8: Density distribution of the two dimensional steady Riemann Problem with $M_{y>0} = 5$ using NT scheme proposed by [7] when $\theta = 1$ in (a) the Eulerian Coordinates and (b) the Lagrangian Coordinates.

6.1.5 NT scheme

To implement the NT scheme proposed in Chapter 4.3.2, we first try a simpler two-dimensional steady Riemann problem

$$(p, \rho, M, \theta) = \begin{cases} (0.25, 0.5, 5, 0) & , \quad y > 0 \\ (1, 1, 2.4, 0) & , \quad y < 0 \end{cases}$$

by replacing the upper flow M by 5, instead of 7. By doing this, the kinetic energy loss due to averaging is reduced. Numerical experiments using the Eulerian Coordinates show that the solution is still too dissipative, one of the major property of the central schemes. Because of this inaccuracy, we cannot continue our computations when the Unified Coordinates are used. In some cases, the solution simply blows up, due to the seriously wrong determination of the fluxes. In some other cases, the h-solver cannot reach a converged solution, this also introduces errors in the computations, which makes our solution highly unreliable, like in the case where the Lagrangian Coordinates ($h \equiv 1$) is used as shown in figure 6.8(b) and figure 6.9(b). Figure 6.9 shows the results using the Eulerian Coordinates and the Lagrangian

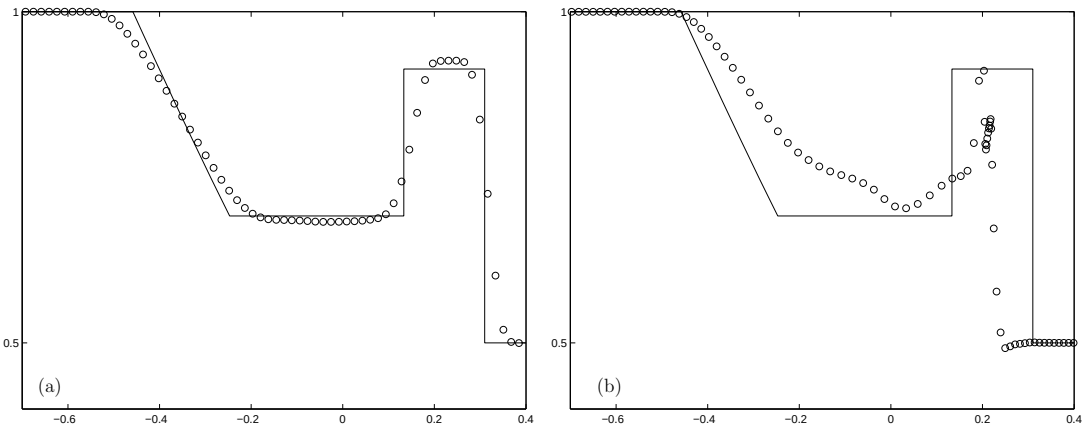

Figure 6.9: Density distribution of the two dimensional steady Riemann Problem with $M_{y>0} = 5$ using NT scheme proposed by [7] when $\theta = 2$ in (a) the Eulerian Coordinates and (b) the Lagrangian Coordinates.

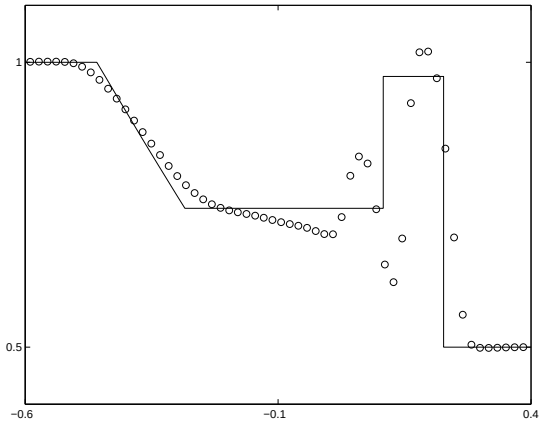

Figure 6.10: Density distribution of the two dimensional steady Riemann Problem with $M_{y>0} = 7$ using NT scheme proposed by [7] when $\theta = 2$ in the Eulerian Coordinates.

Schemes	6.1.2			6.1.3			6.1.4		
	t	λ_0	λ_1	t	λ_0	λ_1	t	λ_0	λ_1
E_{λ_0}	8	0.50	-	7	1.00	-	5	0.67	-
O_{λ_0}	17	0.66	-	36	2.18	-	14	0.45	-
$O_{\lambda_1}E_{\lambda_0}$	14	0.50	0.39	29	1.00	1.32	12	0.50	0.20
$(AO)_{\lambda_0}$	20	1.10	-	40	2.55	-	17	1.10	-
$(AO)_{\lambda_1}E_{\lambda_0}$	15	0.50	0.31	31	1.00	1.46	13	0.50	0.46

Table 6.1: Time consumption (to nearest minute) on solving the two-dimensional steady Riemann Problem using the Exact Riemann Solver, the Collisional BGK scheme and the Hybrid Riemann Solver with different Composite Operators.

Coordinates when θ in equation (4.49) is chosen to be 1. With comparing to figure 6.9, where $\theta = 2$, we see that the dissipation is much reduced in the Eulerian Coordinates. However, the computations in the Lagrangian Coordinates are much worse, in the sense that the grid is much distorted. Although the computation in the Eulerian Coordinates is acceptable when $M_{y>0} = 5$, the solution become highly oscillatory near the slipline when $M_{y>0} = 7$, as shown in figure 6.10, and which cannot be improved using the Unified Coordinates approach.

Therefore, as a conclusion for the replacement by using NT schemes, we found that such highly dissipative scheme is not suitable in solving the Riemann problem in the Unified Coordinate systems. Although we can save a little bit of computational time in using this non-iterative scheme, we cannot improve the solution much. Both the shock and the slipline in the solution are still smeared.

6.1.6 On the Efficiency of the Composite Schemes

As proposed in Chapter 4.4, we now try to compare the total time needed to obtain the steady state solutions from different Riemann solvers. Table 6.1 lists all the

total time spent on the two-dimensional steady Riemann problem from the Exact Riemann solver (6.1.2), the collisional BGK solver (6.1.3) and the hybrid Riemann solver (6.1.4). The column "t" denotes the real time spent, rounded to nearest minute. It should be remembered that, the numbers here are just an rough estimate due to the fact that its difficult to really tell when the solution is absolutely converged because the grids are always moving with velocity $h\mathbf{q}$. We cannot simply compare the flow quantities at the same point in the $\xi - \eta$ space.

The values in the columns λ_0 and λ_1 denote the value in the λ-space needed when the solution "converges". When the Composite Operator approach, OE or $(AO)E$, is used, λ_0 ("time" spent on the Eulerian Coordinates) will generally equal to the same value as used in E alone. For example, we use λ_0 for 0.5 in the case of Exact Riemann solver. And this value 0.5 is exactly the "time" needed for the solution becomes steady when we use the Eulerian Coordinates alone. By doing this, we can treat the extra time needed, by comparing the operator E_{λ_0} and $U_{\lambda_1}E_{\lambda_0}$, is the time needed to improve the solution.

When the Exact Riemann solver is used, we note that approximately 8 minutes are used to obtain the steady state solution when we use the Eulerian Coordinates. 6 more minutes are needed to improve the solution by using orthogonality preserving scheme, while 7 more minutes are needed when AO-preserving grid is chosen. The extra time we spent, of course, due to the fact that we have to do more iterations in obtaining the solution from the h-equation. But more importantly, when the grids move, it takes extra time for the grid motion to be steady.

With comparing the Composite Operator approach, $U_{\lambda_1}E_{\lambda_0}$, with the scheme where we start the computations using the Unified Coordinates alone, U_{λ_0}, we see that the former takes less time to have the steady state solution. The reason, as explained in Chapter 4.4, is the *better choice of initial flow conditions*, as some time steps are spent on the Eulerian Coordinates which provides an approximately correct direction for the grid to move.

For the other type of Composite Operator approach $E_{\lambda_1}U_{\lambda_0}$, we found that we cannot save much time actually. The reason is that once the structure of the grid becomes steady, so does the flow solutions. Therefore, this type of approach cannot help much on improving the efficiency of the computation using the Unified Coordinates.

We have also suggested in Chapter 4.3 that we may try to replace the Exact Riemann solver, which is long regarded as time-consuming and expansive, by other Riemann solvers. In the Eulerian Coordinates, it's true that less time is used. When the Hybrid Riemann solver is chosen, only half the time is needed to obtain the steady state solution. On the other hand, it seems that it takes more time by using the collisional BGK scheme instead of the Exact Riemann solver. However, it should be noticed that we are using first order scheme only. It is not fair to make such conclusion. Of course, solution from first order scheme is worse than a second order scheme in general, and this makes the grids more difficult to be steady. Therefore, it takes more time for this less accurate scheme to obtained the steady solution.

6.1.7 On the Constrain in h and Its Solver

When the Exact Riemann Solver is used, the solution of h by solving corresponding h-equation is given in the figure 6.11. Figure 6.11(a) corresponds to the solution of h obtained by using area-preserving scheme (A-grid), while the other graph shows the distribution of h by using the orthogonality preserving scheme. Note that the discontinuity in both of the solutions are captured sharply. The maximum of h in the domain is unity, as it should due to the scaling.

Now, we are going to check whether the preserving quantity in different preserving schemes is really preserved or not.

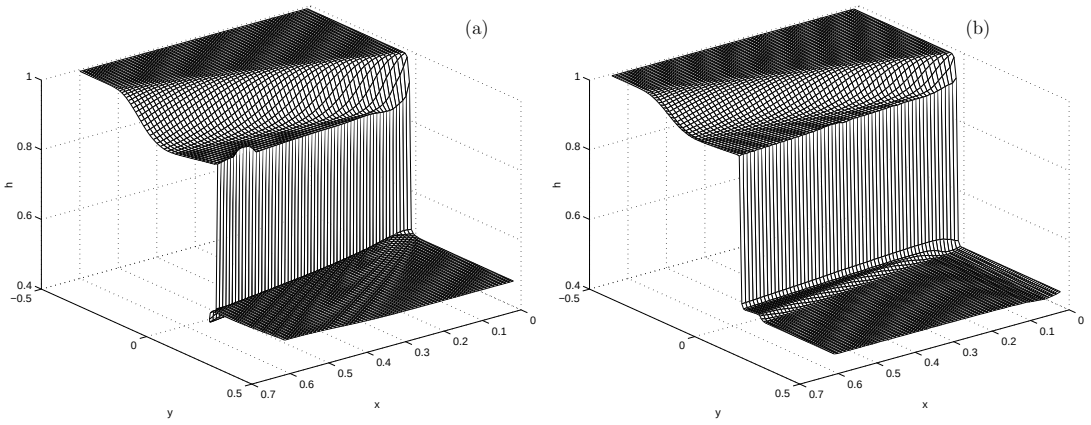

Figure 6.11: Solution of h obtained by using (a) AO-grid and (b) orthogonality preserving grid (O-grid).

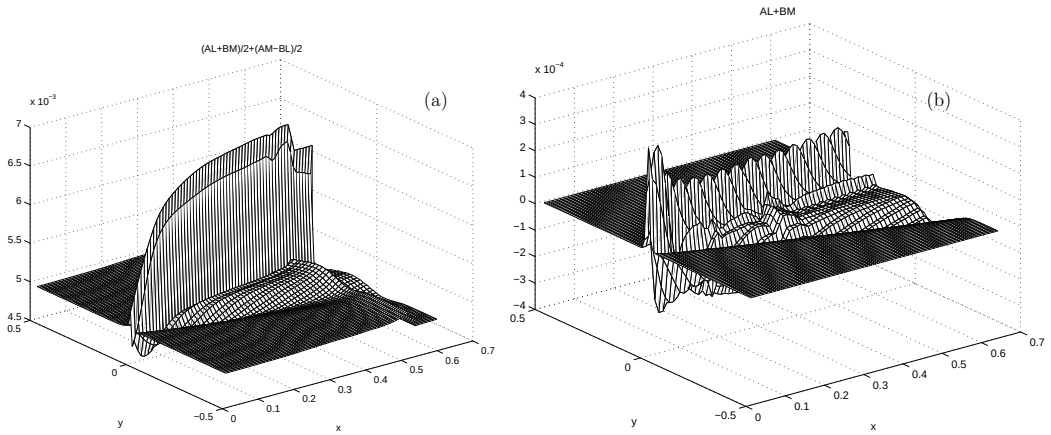

Figure 6.12: The conserved quantities when different preserving schemes are used. (a) $[(A+B)(M+L)/2-BL]$ for AO-grid and (b) $(AL+BM)$ for the orthogonality preserving (O-grid).

When AO-grid is used, we are actually preserving the quantity

$$\frac{1}{2}(AL + BM) + \frac{1}{2}(AM - BL)$$

which is a linear combination of the preserving quantity of A-grid and that of O-grid. After rearrangement, the preserving quantity for this case is

$$\frac{(A+B)(M+L)}{2} - BL\,, \tag{6.1}$$

which is shown in figure 6.12(a). We note that the magnitude of the fluctuation is in order of 10^{-3} which is still acceptable.

The situation is simpler in the case of orthogonality preserving. The quantity we want to preserve is $(AL + BM)$. Accepting the small perturbation of order of 10^{-4}, the quantity is actually keeping as zero, indicating the grids are orthogonal to each other.

In summary, different choices of the preserving quantities are preserved pretty well.

To consider the efficiency of the h-equation solver, we found that the average number of iterations required for the convergence for the solver varies according to the type of preserving quantities we want. For example, it takes 2 iterations only for solving h for area-preserving scheme. However, the number of iterations increases to around 10 for the orthogonality preserving scheme. And, approximately 4 iterations are needed for the AO-scheme. As what we have discussed in Chapter 4.1, orthogonality preserving is a tougher condition than area preserving.

6.2 Two Dimensional Steady Double Riemann Problem

Different from the steady Riemann Problem described in Section 6.1, we will now consider a two dimensional steady double Riemann Problem, generated by three

uniform parallel flows as

$$(p, \rho, M, \theta) = \begin{cases} (1, 1, 2.4, 0) & , \quad |y| > 0.15 \\ (0.25, 0.5, 5, 0) & , \quad |y| < 0.15 \end{cases} .$$

The flow not only contains shock waves, sliplines and expansion waves, but also shows interactions among these waves. As explained in the previous section, due to the dissipative property of the numerical method, the sliplines will be smeared badly in the Eulerian case. And, because the flow solution after the interactions in this test case highly depends on the accuracy of these waves, it is very important to calculate the waves correctly before the two shocks meet each other.

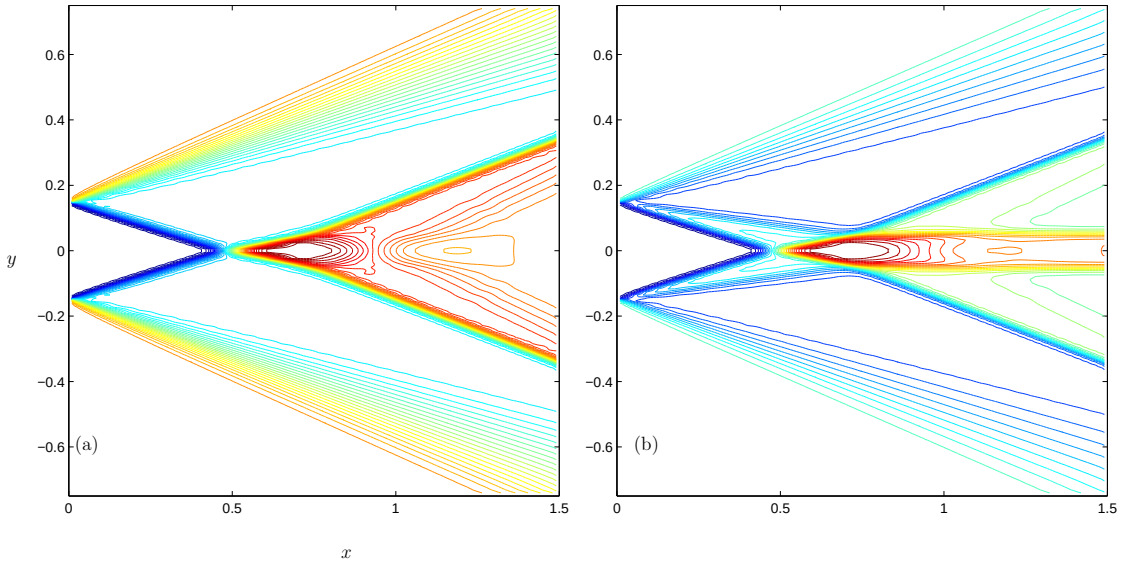

Figure 6.13: (a) Pressure contours and (b) density contours of the two dimensional steady double Riemann problem using the Eulerian Coordinates ($h \equiv 0$).

Because the solution is not a self-similar one in this case, we cannot simply plot with the transformation (y/x). Now, we plot the solution of the density distribution at the position $x = 1.3$, where the shock waves have met and are reflected due to the collision with the slipline, as shown in figure 6.14. The horizontal axis of the plots in figure 6.14 corresponds to the line $x = 1.3$ parallel to the y-axis in figure 6.13.

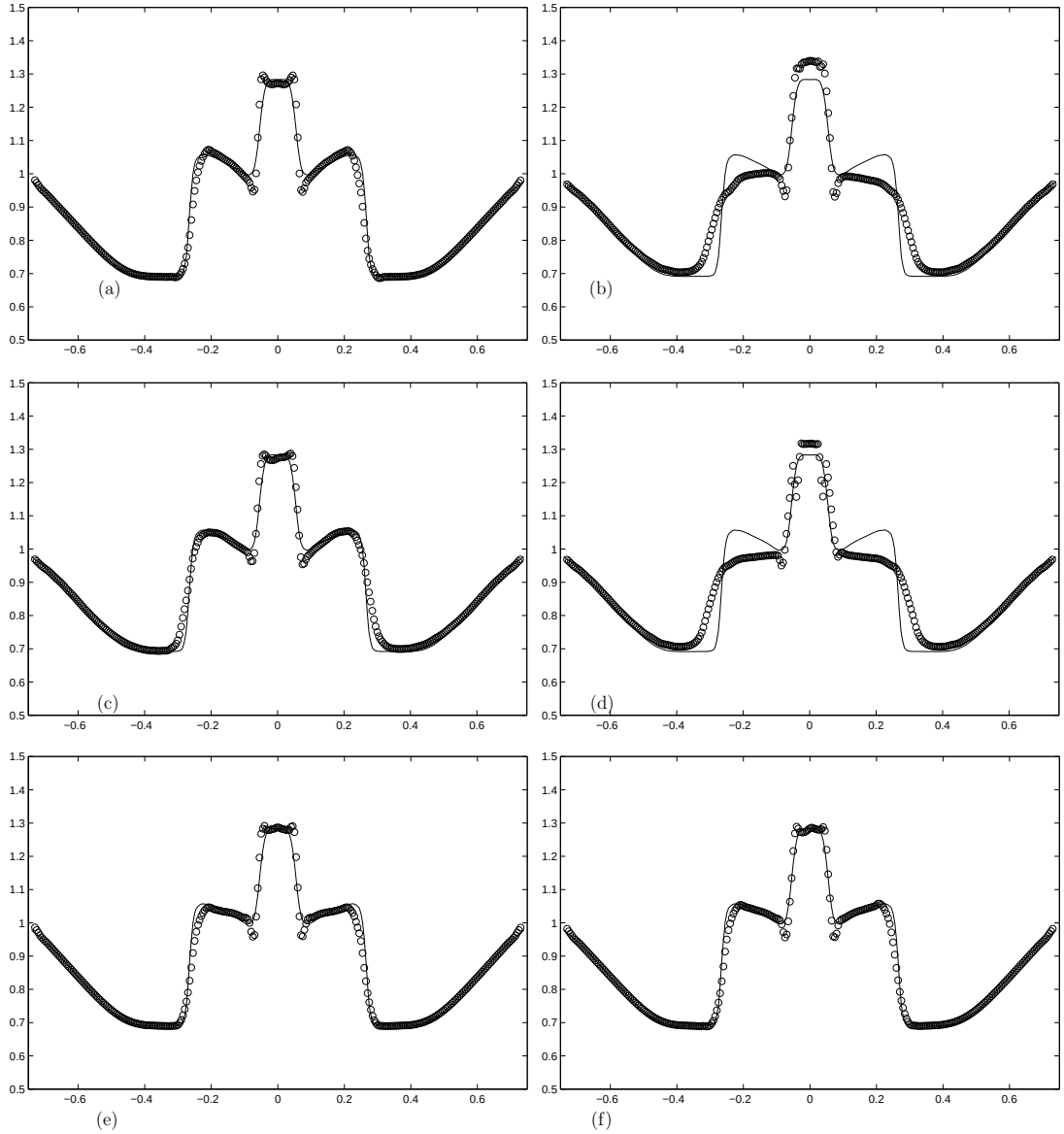

Figure 6.14: Density distributions of the two dimensional steady double Riemann problem using the Unified Coordinates using (a) orthogonality preserving (O-grid), (b) area preserving (A-grid), (c) AO-grid, (d) AL-grid, (e) OL-grid and (f) AOL-grid at the position $x = 1.3$. The solid line denotes the solution from the Eulerian Coordinates and the circles represents the solution from the Unified Coordinates.

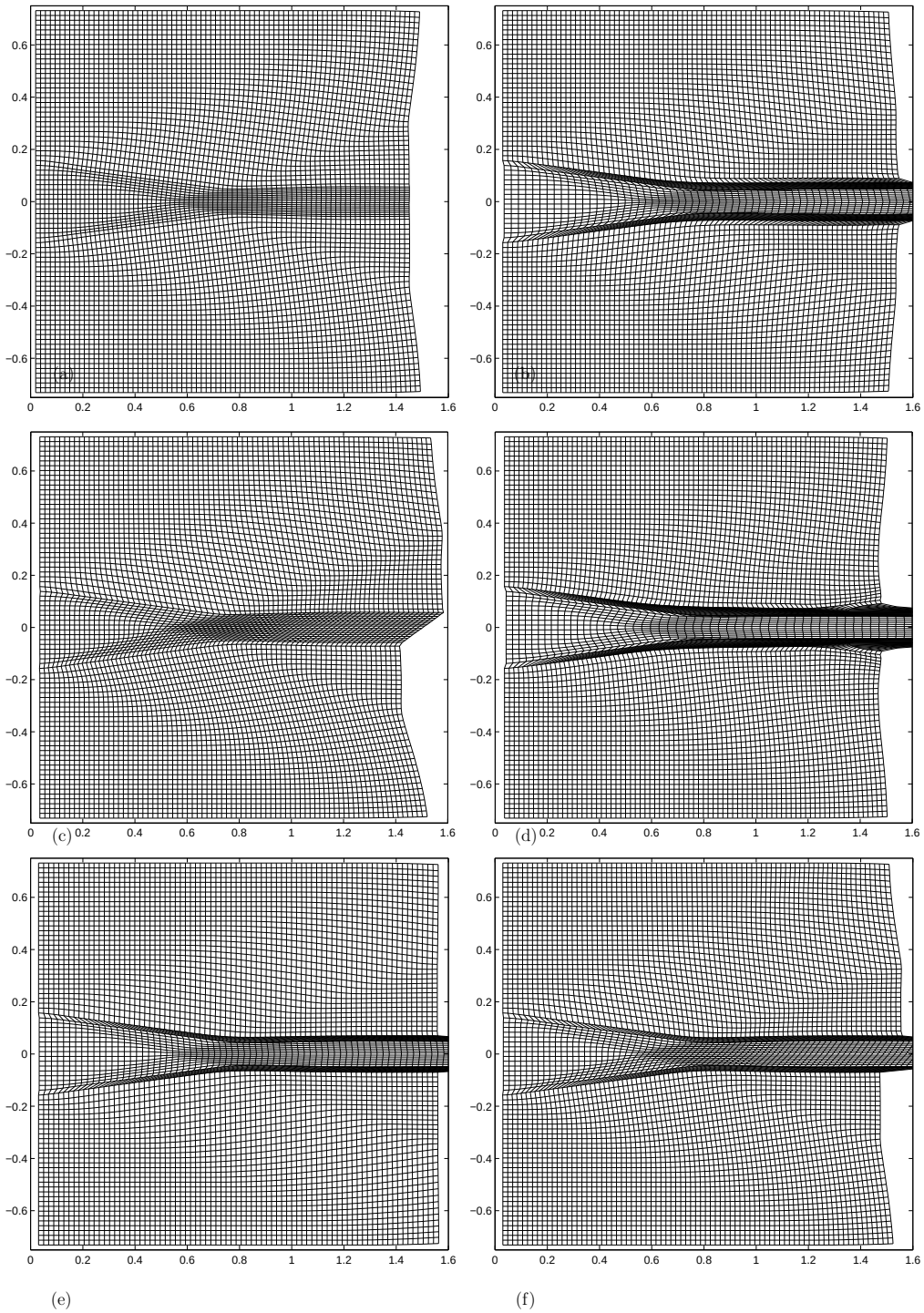

Figure 6.15: Flow-generated grid of the two dimensional steady double Riemann problem using the Unified Coordinates using (a) orthogonality preserving (O-grid), (b) area preserving (A-grid), (c) AO-grid, (d) AL-grid, (e) OL-grid and (f) AOL-grid.

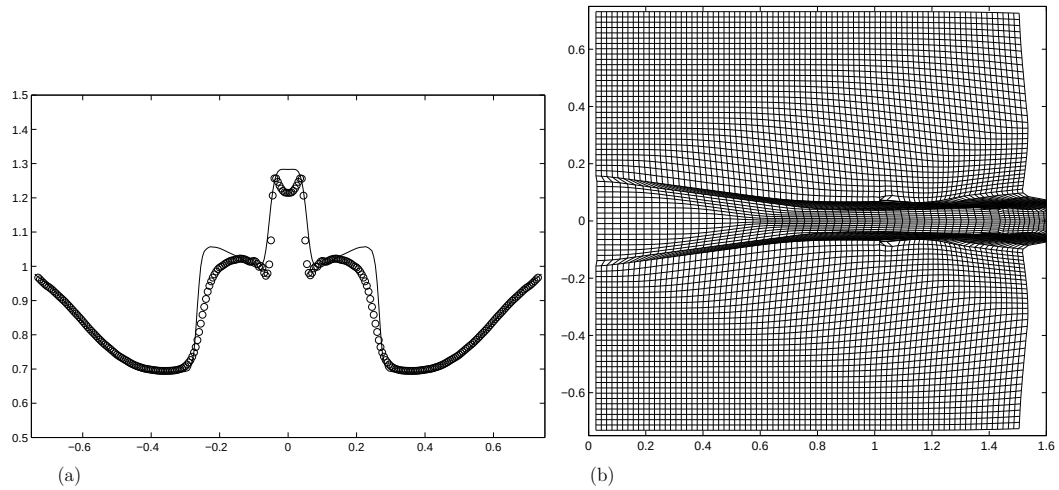

Figure 6.16: (a) Density distribution and (b) the flow-generated grid of the two dimensional steady using the Lagrangian Coordinates ($h \equiv 1$).

The solid line in those graphs in figure 6.14 represents the solution by using the Eulerian Coordinates with four times the number of grids than that from using the Unified Coordinates.

Most solutions from the composite schemes give reasonably good results in the sense that the sliplines are captured pretty sharply. However, the computations from using A-grid, AL-grid and the Lagrangian Coordinates blow up, due to the irregularity of the grids. Around the sliplines, the grids from the above three coordinate systems are highly distorted, which makes the computations difficult to continue. In summary, the solutions from the O-grid show the best resolution of the slipline among other possible choices of h-equation solver.

6.3 Prandtl-Meyer Expansion Flow

Here, a supersonic stream is turned by an angle α (Figure 6.17), the instream condition (p, ρ, M) is chosen to be (1,1,2).

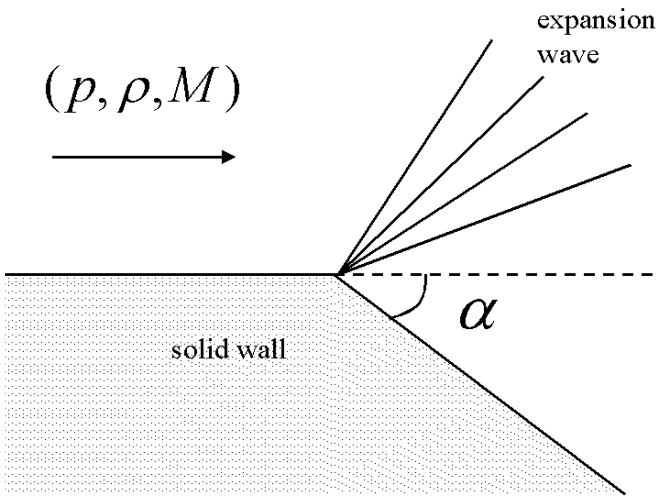

Figure 6.17: Prandtl-Meyer Expansion Flow.

As discussed in Chapter 5, the problem is singular at the corner where the flow velocity is not unique. Therefore, special treatments are used here. In order to have a regular problem, we have removed the sharp corner and replaced it by a curve such that the slope changes smoothly. This modification is very mild. Only one to two cells near the corner are affected in general. However, although tiny, it shows great improvement in the easiness of computation. By doing this, the flow velocity on the surface of the wall will not be multi-valued any more. The second treatment we have implemented is to replace the energy equation by the entropy equation, as discussed in Chapter 5. Because the solution contains an expansion fan only, the solution is smooth in this case. More importantly, this is a constant entropy flow or the so-called homentropic flow. Therefore we can simply obtain the pressure using equation (5.4),

$$p^{(k+1)} = p^{(k)} \left(\frac{\rho^{(k+1)}}{\rho^{(k)}} \right)^{\gamma} .$$

By doing this, the entropy of the flow is obtained exactly everywhere in the computational domain.

Figure 6.18 shows the solution obtained by using the Eulerian Coordinates,

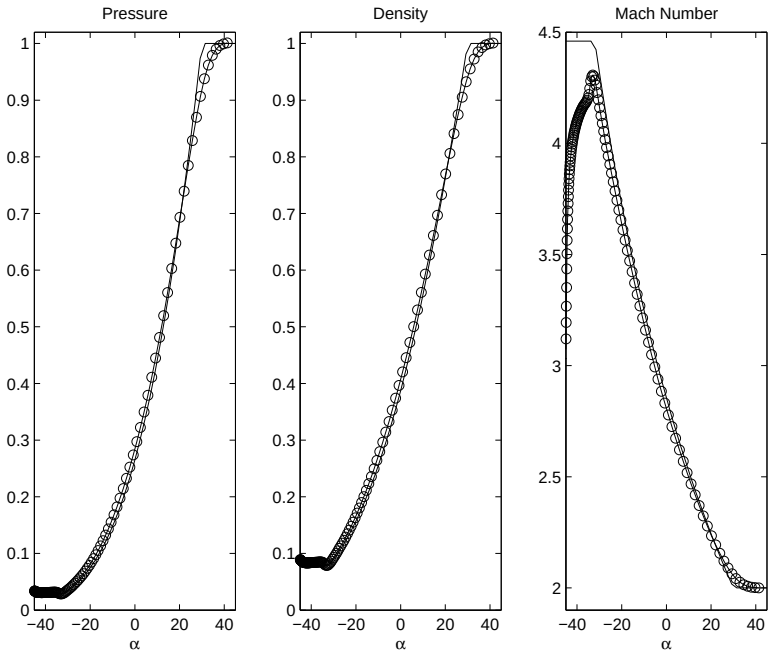

Figure 6.18: Prandtl-Meyer Expansion Flow using the Eulerian Coordinates ($h \equiv 0$) when $\alpha = 45°$. The solid line represents the exact solution, while the circles denotes the computed solution.

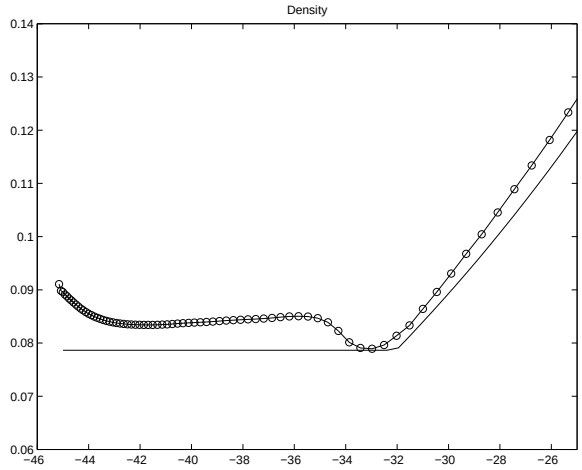

Figure 6.19: Density distribution of the Prandtl-Meyer Expansion Flow using the Eulerian Coordinates near the wall, showing about 15% of error.

i.e. $h \equiv 0$. The solid line denotes the exact solution, while the circles represents the computed solution. When $\alpha = 45°$, we see that the density and the pressure seem to be calculated correctly, but the Mach number is wrong near the solid wall. This is one form of the wall-overheating problem discussed in Chapter 5. The spurious generation of the entropy increases the density near the wall by approximately 15% (figure 6.19) and this reduces the Mach number by about 30%.

The solution by the Lagrangian Coordinates is much improved. Instead of having a sudden drop in the Mach number near the wall, we see that the Mach number in the constant state near the wall is quite accurate, although it shows a little bit of over-shooting on the other hand.

When the orthogonality preserving grid is used, the solution obtained (figure 6.21) is quite similar to that of the Lagrangian Coordinates (figure 6.20). But, one of the very differences between these two methods is the regularity of their grids.

Figure 6.22 shows the flow-generated grid for the Prandtl-Meyer expansion using the Lagrangian Coordinates ($h \equiv 1$) and the Unified Coordinates when the orthogonality preserving grid is used. We see that the grid in the latter case is highly regular and, therefore, avoids the crossing of any cell and minimizes the error due to the irregularity in the grids.

If the turning angle α is now increased from 45° to 55° (figure 6.23-6.24), the computation becomes very difficult. One reason is that the pressure near the wall very closes to zero. Unless special care is taken in the Newton's iteration from the Exact Riemann Solver, this may cause negative pressure easily. However, because we have "fitted" the pressure using the entropy equation, the computation now shows no difficulty. The solution we obtained are very accurate, except a little bit of over-shooting in the Mach number near the wall.

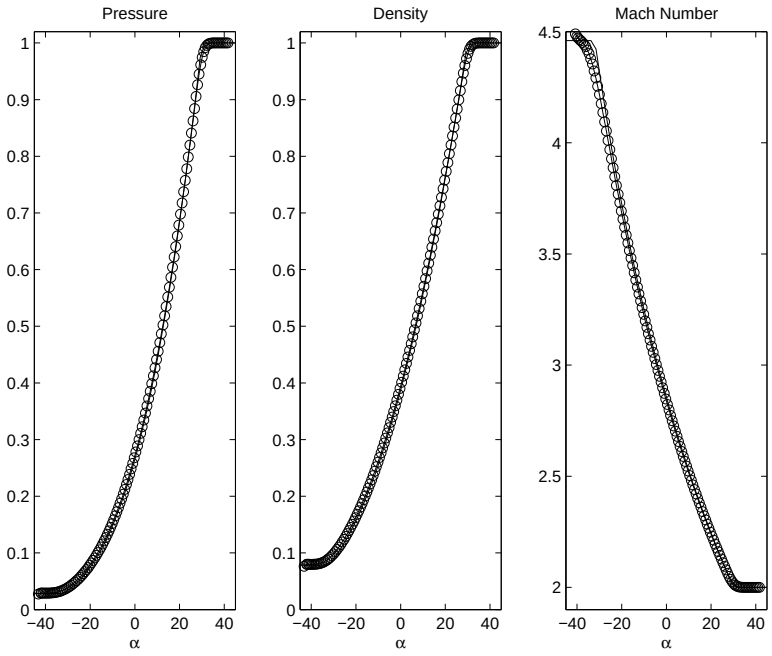

Figure 6.20: Prandtl-Meyer Expansion Flow using the Lagrangian Coordinates ($h \equiv 1$) when $\alpha = 45°$.

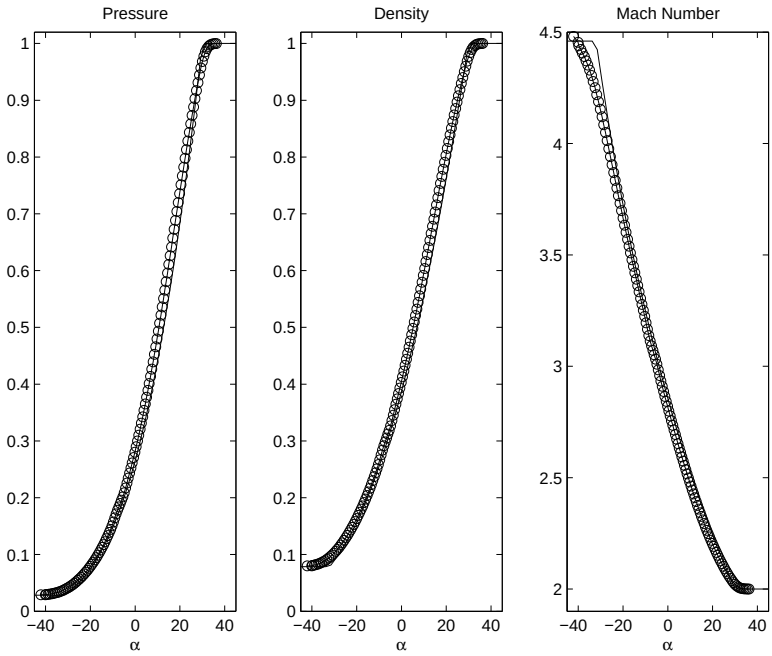

Figure 6.21: Prandtl-Meyer Expansion Flow using the Unified Coordinates (Orthogonality Preserving) when $\alpha = 45°$.

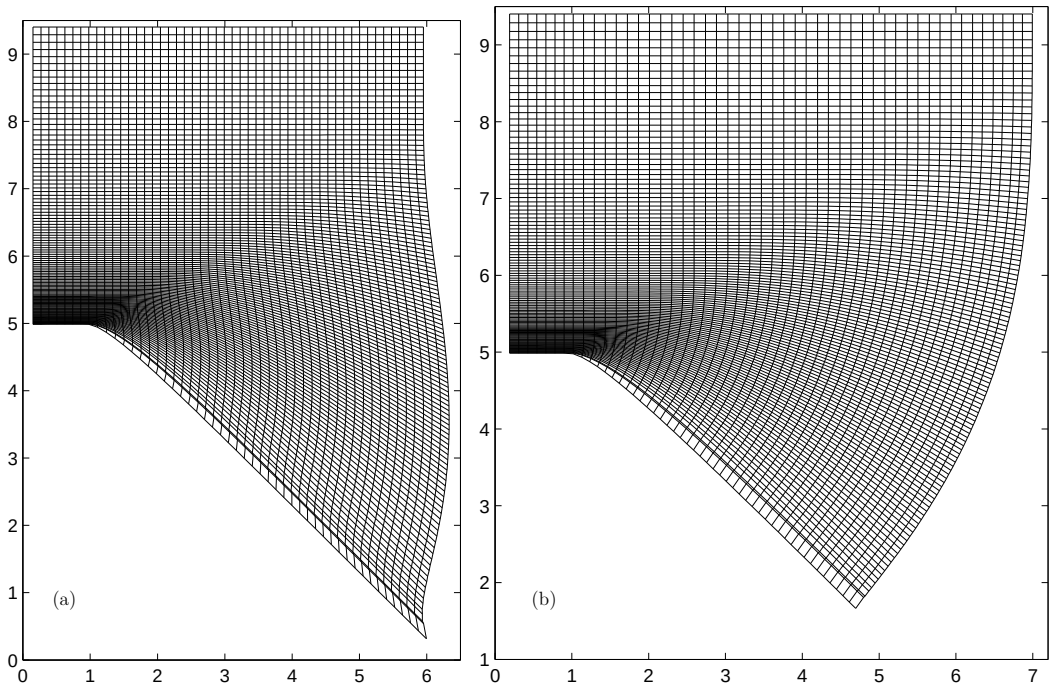

Figure 6.22: Flow-generated grid for Prandtl-Meyer Expansion Flow using (a) Lagrangian Coordinates ($h \equiv 1$) and (b) Unified Coordinates with O-grid when $\alpha = 45°$.

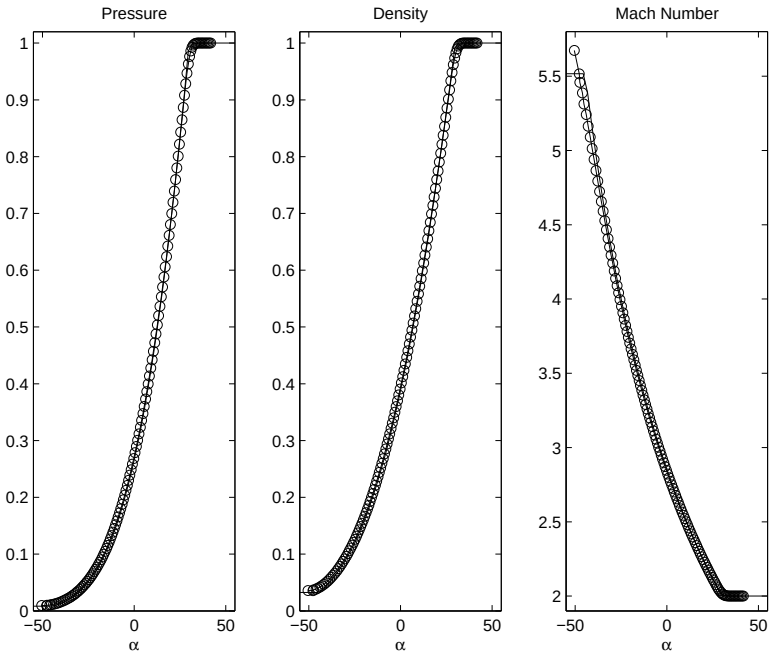

Figure 6.23: Prandtl-Meyer Expansion Flow using the Lagrangian Coordinates ($h \equiv 1$) when $\alpha = 55°$.

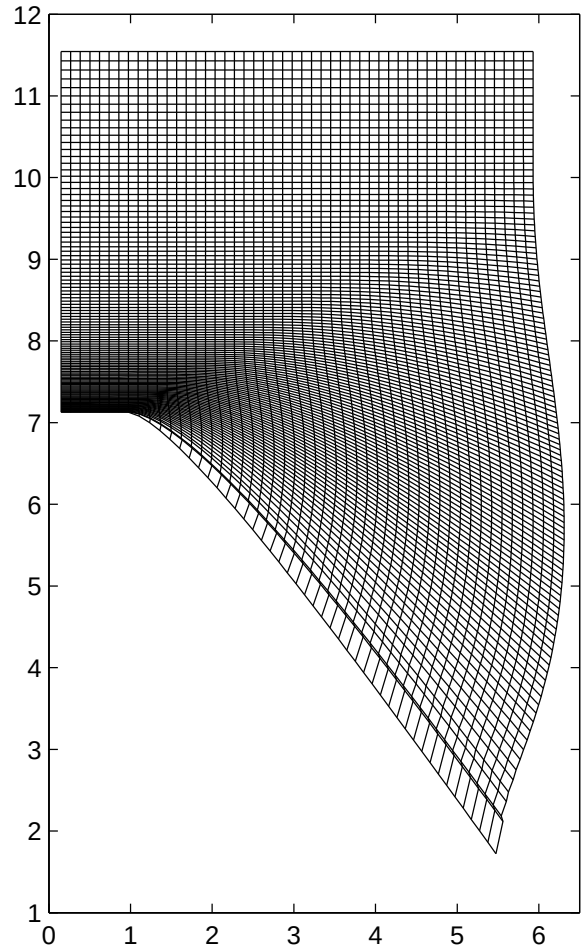

Figure 6.24: The flow generated grid for Prandtl-Meyer Expansion Flow using the Lagrangian Coordinates ($h \equiv 1$) when $\alpha = 55°$.

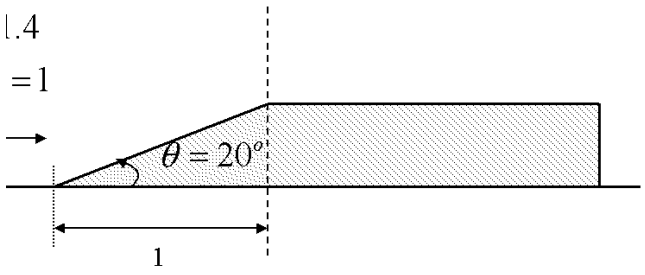

Figure 6.25: Wedge Flow test.

6.4 Wedge Flow

The reason why we have to consider the unsteady flow equation, rather than the steady one, is that when the flow is not supersonic everywhere, the problem become an elliptic one. We now try to consider a test case where there exists a region the flow is subsonic. Consider a wedge with an inclination angle of 20°, as shown in Figure 6.25.

The inflow Mach number is chosen to be 1.4, where the density and the pressure are set to be 1. Because 20° is much larger than the critical angle $\theta_{\max}$ for which the shock is attached (the expression of $\theta_{\max}$ can be found in most Fluid Mechanics book, e.g. [13]), a normal shock is formed which extends around the body as a curved oblique shock.

Pressure contours of the flow is given in Figure 6.26. It should be noticed that there are fluctuations along the bowed shock when the Eulerian Coordinates are used. This may be due to the averaging of the shock in a cell, or the fact that the velocity of the shock is small when the flow becomes steady, which generates spurious oscillation know as the problem of "Slowly Moving Shock". When the Unified Coordinates is used instead, we found that this problem is eliminated. The oscillations are smoothened and the expansion fan can be clearly seen. Although the constant state region after the expansion fan shows a little bit of oscillation when

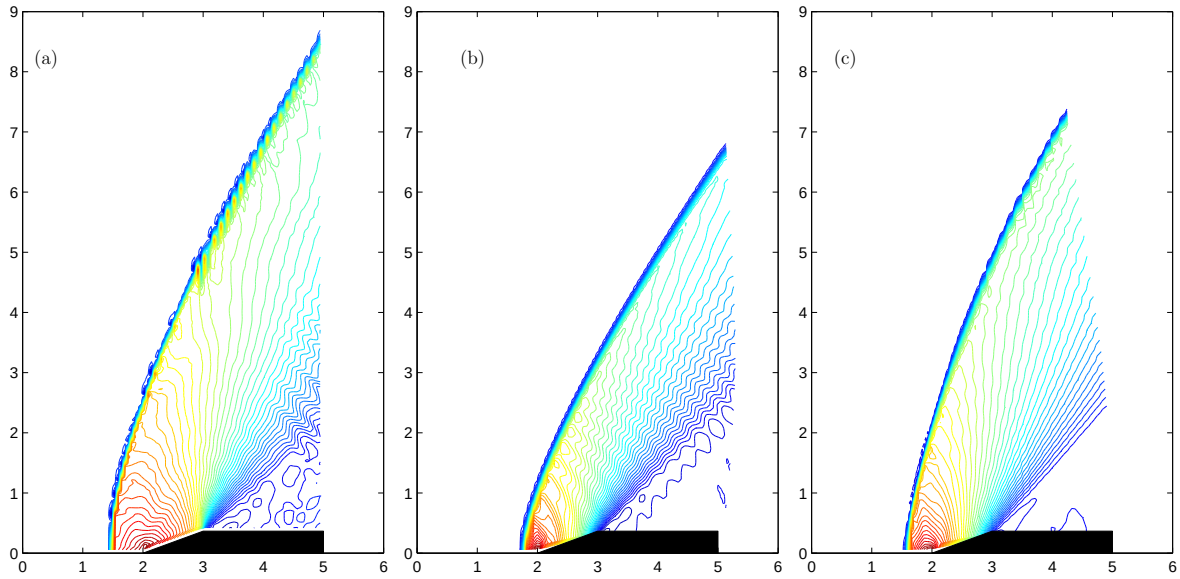

Figure 6.26: Pressure contours of the Wedge flow test case using (a) the Eulerian Coordinates and the Unified Coordinates with (b) area-preserving grid (A-grid) and (c) orthogonality-preserving grid (O-grid).

we are using the area-preserving grid, it is still acceptable.

The flow-generated grids from the two different h-equations are given in Figure 6.27. Note that from the orthogonality-preserving scheme, there is a region in the domain where the computational grids are highly compressed and squeezed. This might cause difficulty in computations if the flow situation is more complicated, although it shows no problem in this test case. With comparing to the grid generated by the area-preserving scheme, the grids in the latter case have similar size, as it should.

The quantity $(AL + BM)$ for orthogonality-preserving grid and $(AM - BL)$ for area-preserving grid are also given here in Figure 6.28.

For the area-preserving grid, $\max(AM - BL)$ is found to be 0.3809 while $\min(AM - BL)$ is 0.1513. Although the fluctuation is large, the oscillation appears

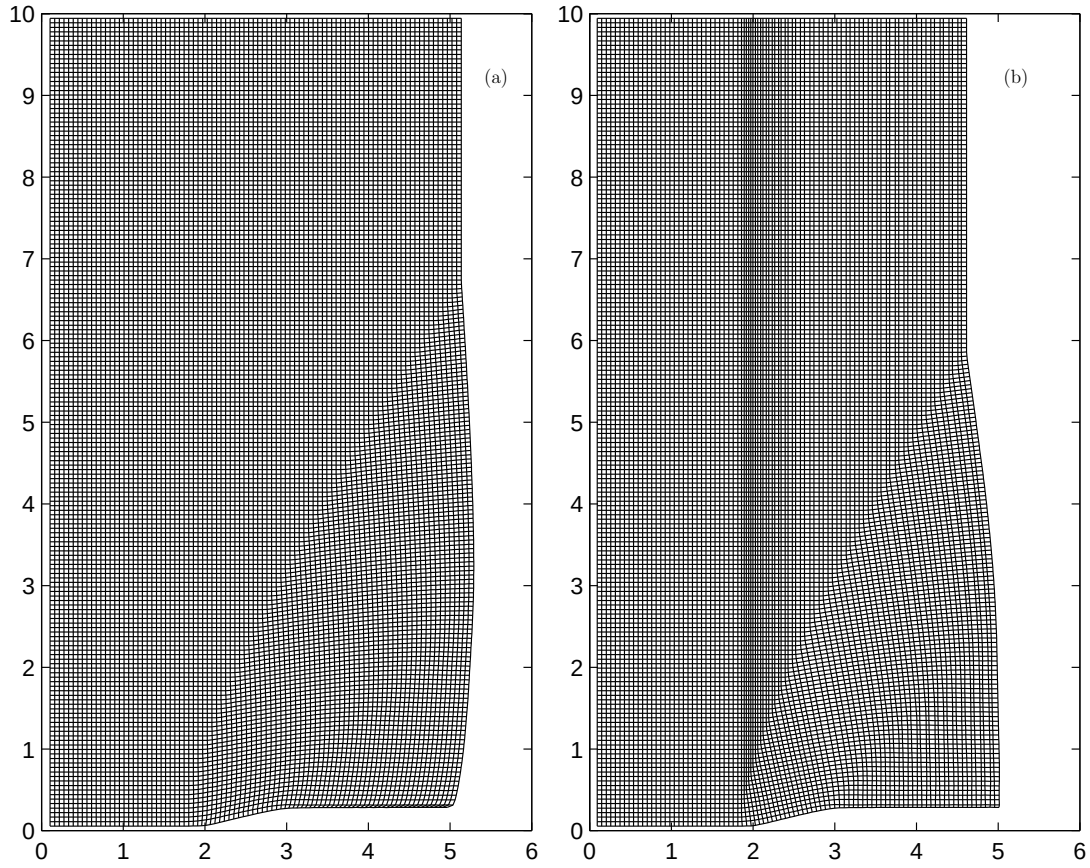

Figure 6.27: Flow-generated grid of the wedge flow test case using the Unified Coordinates with (a) area-preserving grid and the (b) orthogonality-preserving grid.

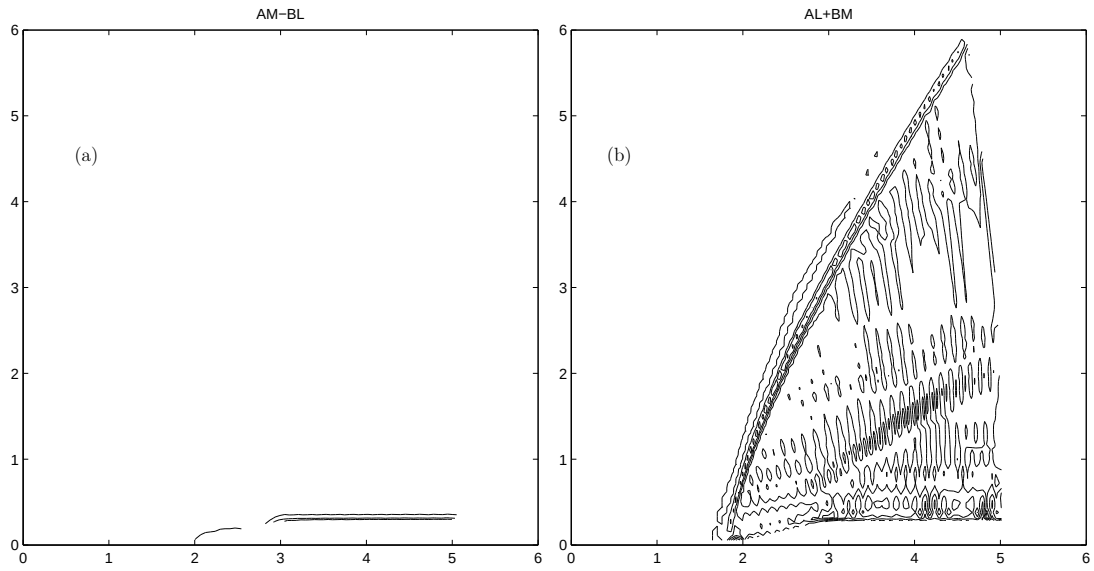

Figure 6.28: The preserving quantity in the h-solver of the Wedge flow test case using the Unified Coordinates with (a) area-preserving grid and (b) orthogonality-preserving grid.

near the boundary only and the quantity is kept approximately constant throughout the rest part of the domain.

The oscillation in the orthogonality-preserving grid is smaller. $\max(AL + BM)$ is 0.0174 and $\min(AL + BM)$ is found to be -0.0209. In the case where the grid is orthogonal, the quantity $(AL+BM)$ is zero; when comparing to the numerical values, the errors are still acceptable.

With the acceptance of the small error from the computations, both quantities are kept constant approximately, as what we intended to preserve originally.

6.5 Double Blocking Test

In order to study how well the h-equations behaves, we now try a geometrically complicated test problem.

As shown in figure 6.29, there are two triangular blocks in a channel. These two triangular blocks will generate shocks and expansion waves. When the shocks meet each other near the center of the channel, two strong contact waves, with different strengths in general, are produced.

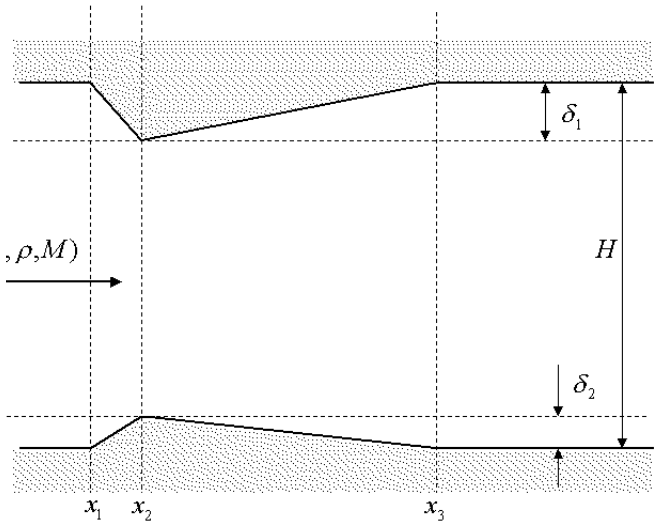

Figure 6.29: Double blocking test case showing notations. p and ρ are chosen to be 1, $x_1 = 0.5$, $x_2 = 1.0$, $x_3 = 2.5$, $\delta_1 = 0.134$, $\delta_2 = 0.067$ and $H = 1.0$.

6.5.1 $M = 1.7$

When the inflow Mach number is taken to be 1.7, we found that there exists a region where the Mach number reduces to be less than 1.0. Therefore, although the inflow is supersonic, the flow become subsonic after the two generated shocks meet each other. This makes the problem more complicated. The motion of the grid not

only need to overcome the singularity problem due to the sharp corner, but also the change in the fluid flow condition, i.e. from supersonic to subsonic.

Except the Eulerian grid, i.e. $h \equiv 0$, all the other combinations mentioned in Section 4.3.1 are implemented. We found that, due to the challenges discussed above, solutions will break down for some types of grid. For example, figure 6.30 shows the orthogonality preserving grid. We see that although the grids try to remain orthogonal when it evolves, due to the fact that the volume of some cells become smaller and smaller, the solution blows up after some time of computation. Even though the cell volume remains the same when evolves as shown in figure 6.30, the grids are highly distorted in the computational domain, the computations also stop.

The weighted preserving schemes show a better robustness property here in this test case. As shown in figure 6.30-6.31, AO-grid and AOL-grid show that the grid is good to resolve the two strong sliplines. Note the difference in AOL-grid, due to the contributions from the Lagrangian, the grid is a little bit stretched near the contact waves. However, with the constrains from the area preserving and orthogonality preserving, these distortions show no bad behavior in the computations, but surprisingly is a good sign of an existence of sliplines.

6.5.2 $M = 2$

Now, if the inflow Mach number is increased from 1.7 to 2.0, we found that the third normal shock disappears, indicating that the flow become supersonic everywhere. Although the problem of singularity in geometry still remains, the problem is actually easier than the one we have just described.

Figure 6.32-6.33 shows the flow-generated grid using different types of weighted preserving schemes. Most of them show satisfactory results. Especially the grid immediate after the two shocks, due to the nature of the Unified Coordinates, the

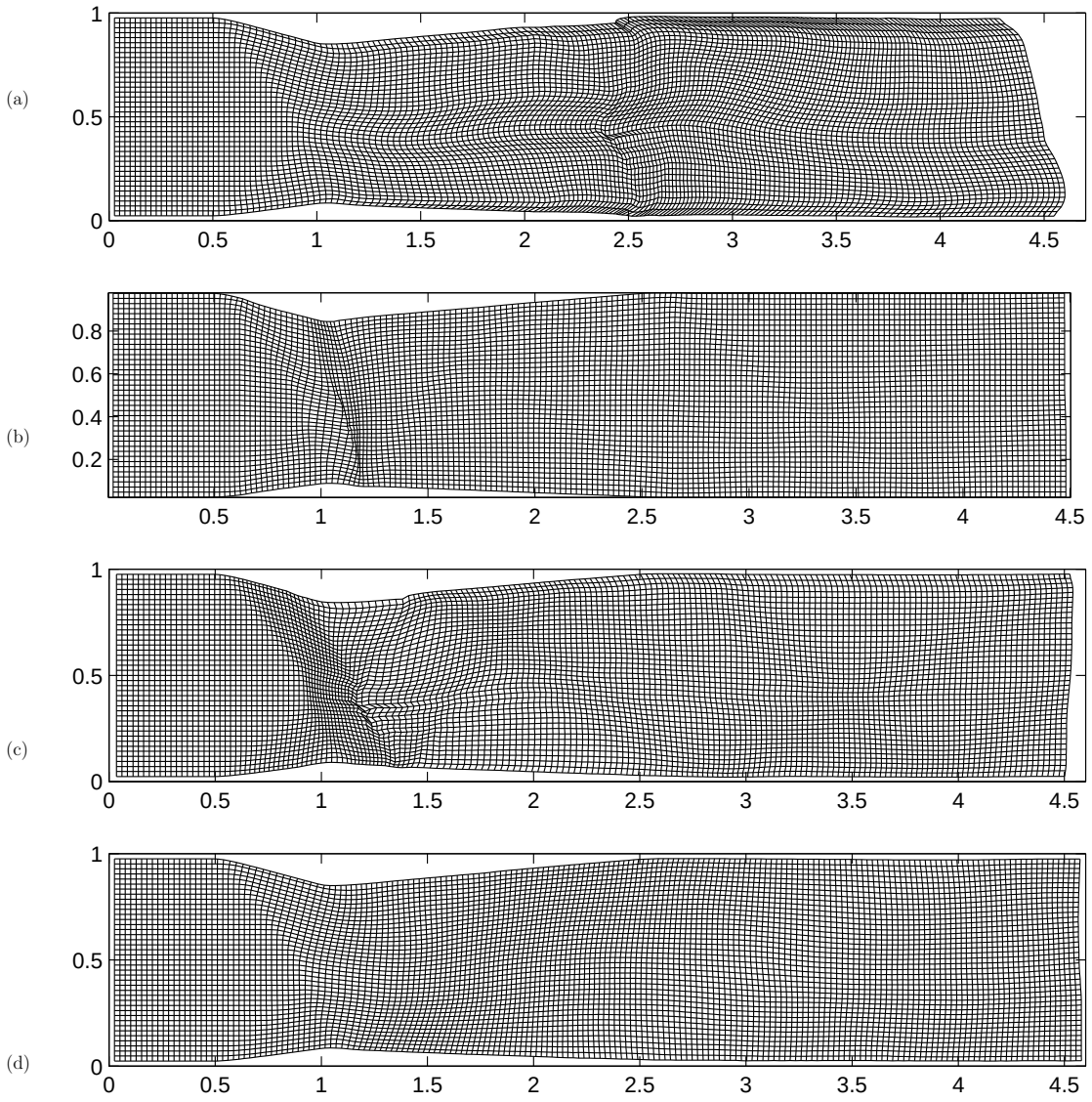

Figure 6.30: Flow-generated grid for the double blocking test ($M = 1.7$) using the Unified Coordinates with (a) A-Grid, (b) O-Grid, (c) L-Grid and (d) AO-Grid.

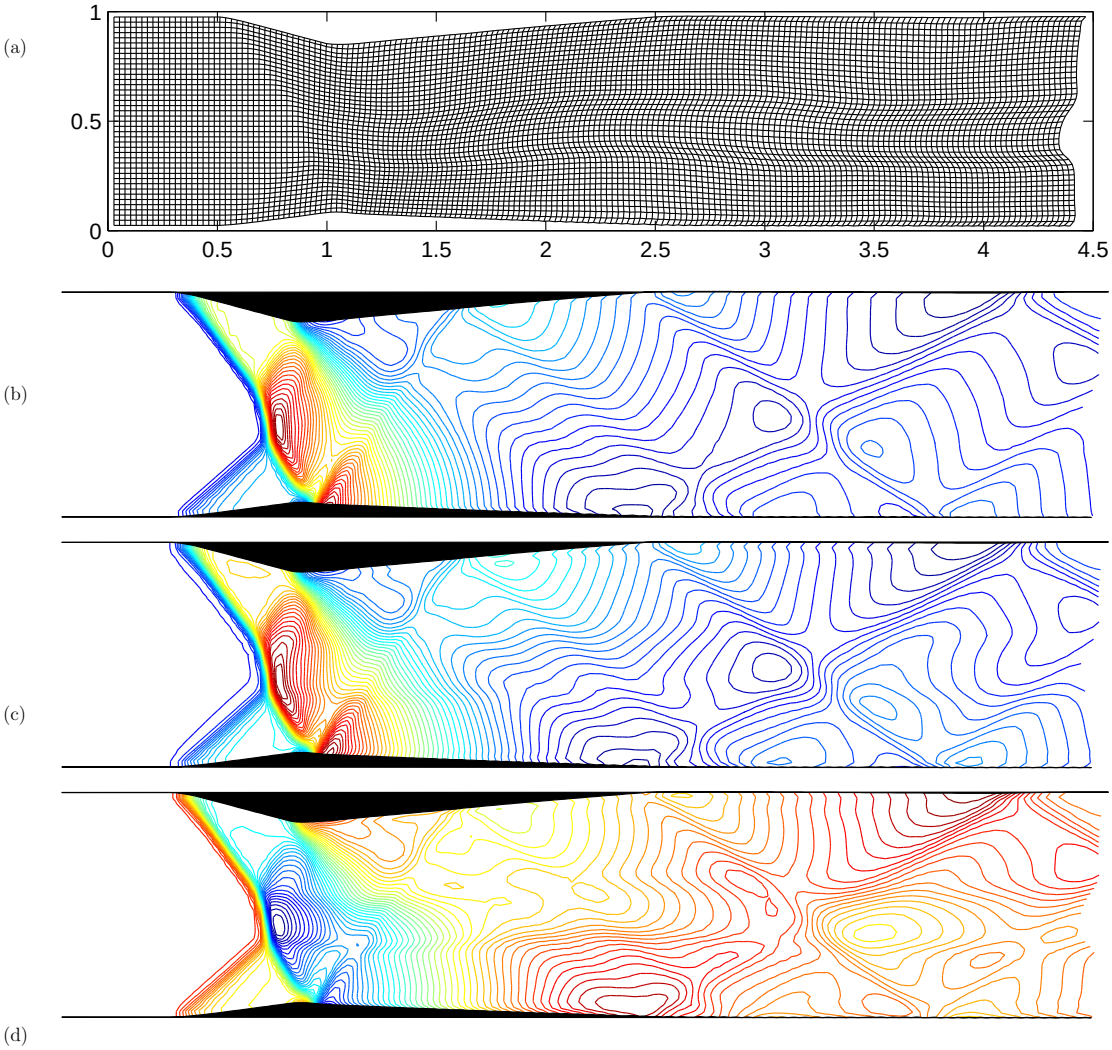

Figure 6.31: (a) Flow-generated grid, (b) pressure contours, (c) density contours and (d) the Mach number contours for the double blocking test ($M = 1.7$) using the Unified Coordinates with AOL-grid.

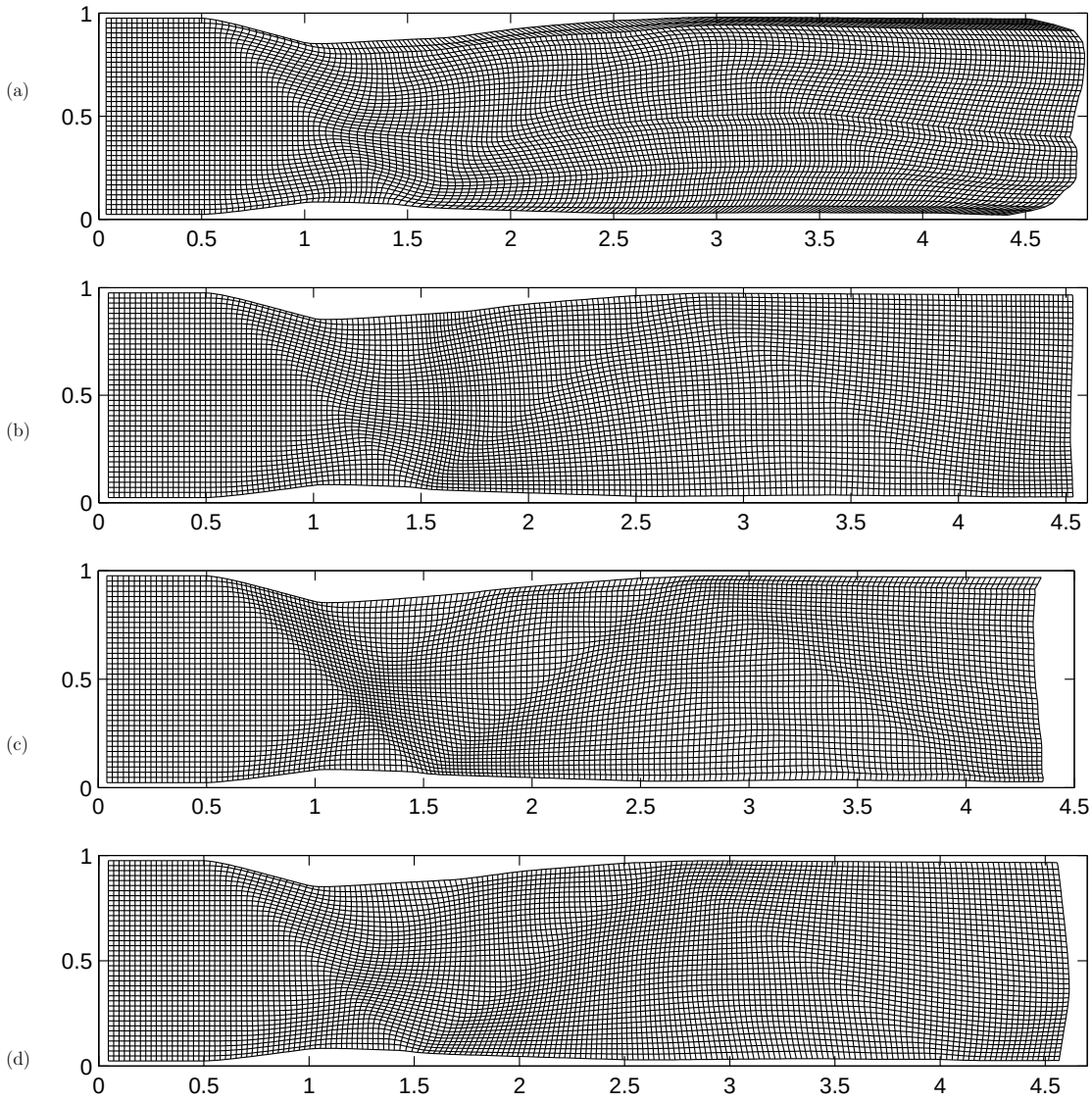

Figure 6.32: Flow-generated grid for the double blocking test ($M = 2$) using the Unified Coordinates with (a) A-grid, (b) O-grid, (c) L-grid and (d) AO-grid.

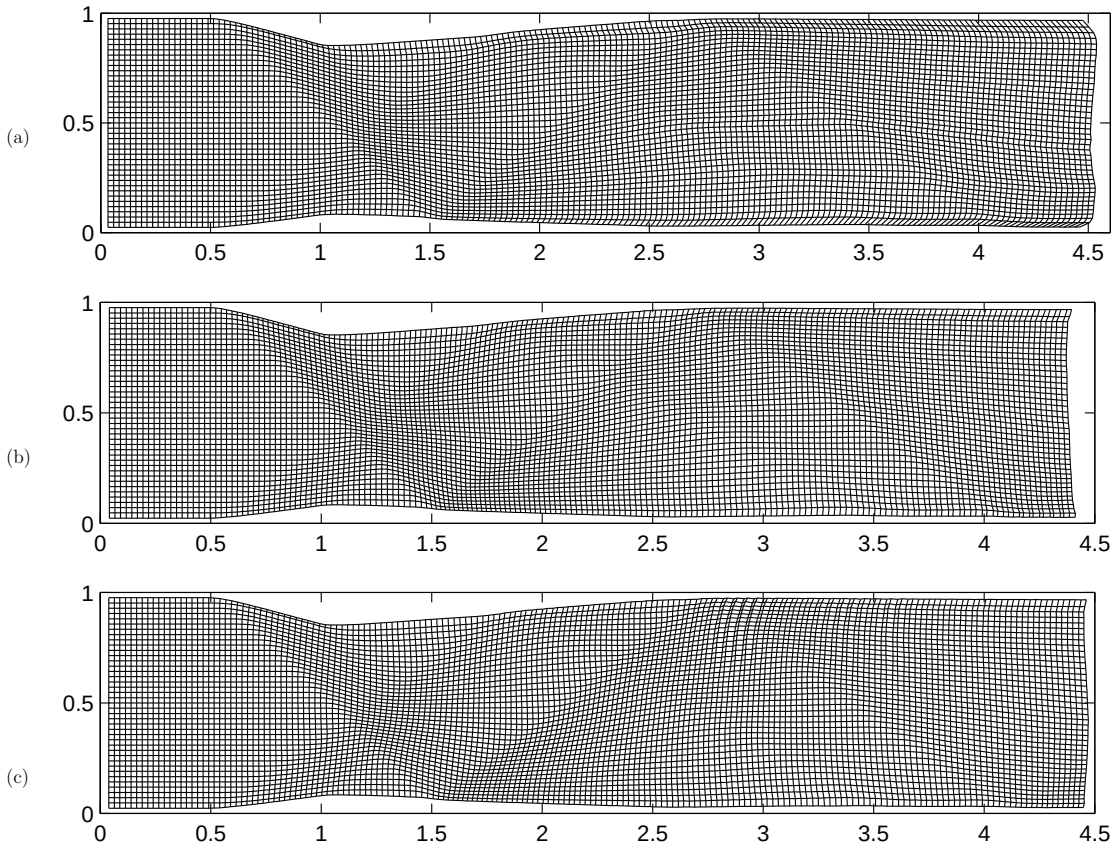

Figure 6.33: Flow-generated grid for the double blocking test ($M = 2$) using the Unified Coordinates with (a) AL-grid, (b) OL-grid and (c) AOL-grid.

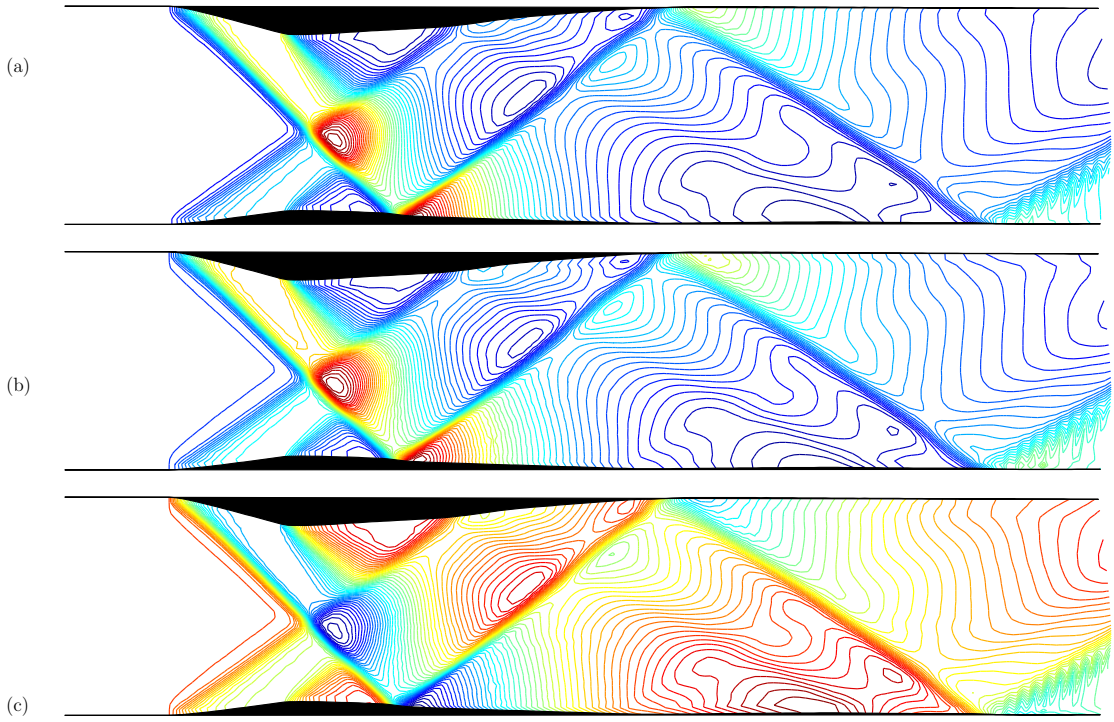

Figure 6.34: (a) Pressure contours, (b) density contours and (c) the Mach number contours of the double blocking test ($M = 2$) using the Unified Coordinates with *AO*-grid.

pseudo-particles tend to crowd together when compressed which result in automatic refinement of the grid in the compression region.

To consider the slipline resolution, instead of having two sliplines as in the previous case where three shocks meet each other, we now have two shocks only, resulting in only one slipline. This contact wave is a weak one unfortunately and therefore can hardly be noticed in the plot of the density distribution. Actually, we should be able to observe the existence of a slipline in a contour plot of entropy, S, on the other hand. However, due to the singularity nature at the sharp corner, spurious entropy will be generated which prevents us to make any conclusion of the existence of a slipline in the flow field.

Chapter 7

Conclusions

In this thesis, we have studied several conditions on determining the grid-controlling function h in the Unified Coordinates approach. Orthogonality-preserving scheme, Area-preserving scheme and the Weighted preserving schemes are introduced and studied. This suggests that other choices on h are possible. But in some cases we met in Chapter 6, the grid from these Weighted-preserving schemes may still squeeze together, like the computation using Lagrangian Coordinates, which blows up the solution.

Orthogonality preserving scheme gives regular grid which helps attending the best accuracy. However, singularity problem in geometry, as in the test case 6.5.1, will sometimes make the computations difficult to continue in the cases where the grid can move, i.e. $h \not\equiv 0$. In these complicated cases it does not work, AO grid can be used instead.

A direct solver for the h-equation is introduced. With comparing to the original Unsteady approach, the new method is easy to implement. Because we do not need to restrict our iteration time step size to satisfy the CFL condition and because iterative schemes for solving a matrix problem are well-developed, the new solver developed in Chapter 4.2.3 can obtain the solution in a shorter period of time.

We have also tried to replace the Exact Riemann solver by some other non-iterative Riemann solvers. However, the time needed for solving the flow problem is reduced a little bit only in general, while the solution obtained could be worsen quite a lot.

A Composite Operator (UE) approach is also introduced in Chapter 4.4. Efficiency of the Unified Coordinates can be improved. The reason is that a better choice of initial flow condition is used to determine the direction of the grid moves.

Appendix A

Kinetic Energy Loss Due to Slipline Averaging in Eulerian Computations

Now, we try to analysis the KE loss due to slipline averaging where a rectangular computational cell in Eulerian coordinates contains a slipline, splitting the cell into two parts, a triangular part and a trapezoidal part or two trapezoidal parts. With the following flow conditions: $(p, \rho, u, v)_1 = (p, \rho, M_1 \cos\theta, M_1 \sin\theta)$ and $(p, \rho, u, v)_2 = (p, \rho, M_2 \cos\theta, M_2 \sin\theta)$, we will consider the problem into four different cases as shown in figure A.1, i.e.

Case (a): $y_0 \geq 0$ and $y_0 + x\tan\theta \leq y$;

Case (b): $y_0 \geq 0$ and $y_0 + x\tan\theta \geq y$;

Case (c): $y_0 \leq 0$ and $y_0 + x\tan\theta \leq y$;

Case (d): $y_0 \leq 0$ and $y_0 + x\tan\theta \geq y$,

where x and y are the dimensions of the cell, $\theta \in [0, \pi/2]$ is the angle between the horizontal axis and the slipline and y_0 is the vertical-axis intercept.

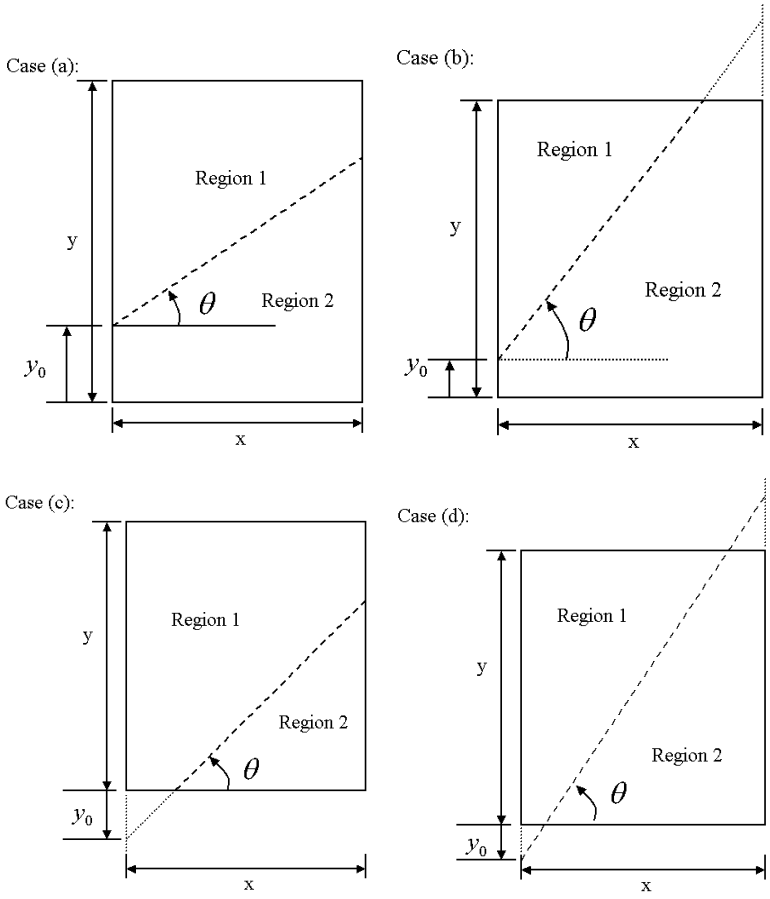

Figure A.1: Four different cases of slipline splitting.

Now, we further define A to be the cell size, i.e. $A = xy$, and A_2 to be the area of the lower part of the cell, divided by the contact wave. With all of the above cases and notations used, A_2 is calculated to be

Case (a): $A_2 = \dfrac{x}{2}(2y_0 + x\tan\theta)$;

Case (b): $A_2 = A - \dfrac{(y - y_0)^2}{2\tan\theta}$;

Case (c): $A_2 = \dfrac{(x\tan\theta + y_0)^2}{2\tan\theta}$;

Case (d): $A_2 = A + \dfrac{y(2y_0 - y)}{2\tan\theta}$.

Therefore, with the relations

$$\mathrm{KE}_{\mathrm{exact}} = \frac{1}{2}A\rho_1 M_1^2 - \frac{1}{2}A_2(\rho_1 M_1^2 - \rho_2 M_2^2)$$

and

$$\mathrm{KE}_{\mathrm{averaged}} = \frac{\left[\rho_1 u_1 A + A_2(\rho_1 u_1 + \rho_2 u_2)\right]^2 + \left[\rho_1 v_1 A + A_2(\rho_1 v_1 + \rho_2 v_2)\right]^2}{2\left[\rho_1 A + (\rho_2 - \rho_1)A_2\right]},$$

we have

$$\begin{aligned} \mathrm{KE}_{\mathrm{loss}} &= \mathrm{KE}_{\mathrm{exact}} - \mathrm{KE}_{\mathrm{averaged}} \\ &= \frac{(A - A_2)A_2(M_1 - M_2)^2\rho_1\rho_2}{2[A\rho_1 + A_2(\rho_2 - \rho_1)]}. \end{aligned} \tag{A.1}$$

By fixing y_0 but varying θ or fixing θ but varying y_0, we can study the effect of the $\mathrm{KE}_{\mathrm{loss}}$ due to the change in A_2. Solving

$$\frac{d(\mathrm{KE}_{\mathrm{loss}})}{dA_2} = 0\,,$$

we get

$$A^*_{2_\pm} = \frac{A\sqrt{\rho_1}}{\sqrt{\rho_1} \pm \sqrt{\rho_2}}\,. \tag{A.2}$$

With the conditions

$$\left.\frac{d^2(\mathrm{KE}_{\mathrm{loss}})}{dA_2^2}\right|_{A^*_{2_\pm}} = \mp\frac{(M_1 - M_2)^2\sqrt{\rho_1\rho_2}}{A}\,,$$

we conclude that the loss in KE is maximized when $A_2 = A^*_{2_+}$.

Bibliography

[1] W. H. Hui and D. L. Chu, Optimal Grid for the Steady Euler Equations, *Comput. Fluid Dyn. J.* **4**, 403 (1996).

[2] W. H. Hui, P. Y. Li and Z. W. Li, A Unified Coordinate System for Solving the Two-Dimensional Euler Equations, *J. Comp. Phys.* **153**, 596-637 (1999).

[3] W. H. Hui and S. Kudriakov, The Role of Coordinates in the Computation of Discontinuities in One-Dimensional Flow, *Comput. Fluid Dyn. J.* **8**, no.4, 495-510 (2000). (See also **http://www.math.ust.hk/∼whhui/**)

[4] W. H. Hui and S. Kudriakov, A Unified Coordinate System for Solving the Three-Dimensional Euler Equations, to be published.

[5] W. H. Hui and S. Kudriakov, Computation of the Shallow Water Equations using the Unified Coordinates, to be published.

[6] W. H. Hui and S. Kudriakov, On Wall Overheating and Other Computational Difficulties of Schock-Captering Methods, to be published.

[7] G. S. Jiang and E. Tadmor, Nonoscillatory Central Schemes for Multidimensional Hyperbolic Conservation Laws, *SIAM J. Sci. Comput.* **19**, no. 6, 1892-1917 (1998).

[8] P. D. Lax, Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation, *Communications on Pure and Applied Mathematics* **VII**, 159-193 (1954).

[9] H. Nessyahu and E. Tadmor, Non-oscillatory Central Differencing for Hyperbolic Conservation Laws, *J. Comp. Phys.* **87**, 408-463 (1990).

[10] E. F. Toro, A Linearized Riemann Solver for the Time-dependent Euler Equations of Gas Dynamics, *Proc. R. Soc. Lond.* **A434**, 683-693 (1991).

[11] E. F. Toro, Direct Riemann Solvers for the Time-dependent Euler Equations, *Shock Waves* **5**, 75-80 (1995).

[12] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, *Springer* (1997).

[13] F. M. White, Fluid Mechanics, Chapter 9, Fourth edition, *McGraw-Hill* (1999).

[14] K. Xu, Gas-Kinetic Schemes for Unsteady Compressible Flow Simulations, *von Karman Institute for Fluid Dynamics Lecture Series 1998-03*, 1998. (See also **http://www.math.ust.hk/∼makxu/**)

[15] K. Xu and Z. Li, Dissipative Mechanism in Godunov-type Schemes, to be published in *Int. J. Numer. Methods in Fluids.* (See also **http://www.math.ust.hk/∼makxu/**)