Numerical Schemes for Conservation Laws on Moving Mesh

A thesis submitted towards partial fulfilment of
BS-MS Dual Degree Programme

by

Shubham Pandey

under the guidance of

Dr. Praveen C

TIFR Centre for Applicable Mathematics

Indian Institute of Science Education and Research Pune

Certificate

This is to certify that this thesis entitled “Numerical Schemes for Conservation Laws on Moving Mesh” submitted towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research Pune represents original research carried out by “Shubham Pandey” at “TIFR Centre for Applicable Mathematics”, under the supervision of “Dr. Praveen C” during the academic year 2013-2014.

Student	Supervisor
Shubham Pandey	Praveen C

Acknowledgements

With a feeling of immense gratitude, I would like to thank my advisor Dr. Praveen C for his invaluable suggestions, encouragement and guidance. I am grateful to him for helping me by several enlightening discussion. I would like to express my sincere gratitude to Prof. Christian Klingenberg for his invaluable guidance and support during my two month visit to Germany. His motivation and supervision with constructive criticism enabled me to complete this project.

I would like to thank the rest of my thesis committee: Dr. M. S. Santhanam, for his encouragement, enthusiasm and confidence on me. I would like the EADS chair at TIFR-CAM for financial support during the visit. I would also like to acknowledge the academic and technical support of staff of “TIFR Centre for Applicable Mathematics, Bangalore" and “University of Wuerzburg, Germany".

Abstract

Fluid flows are governed by non-linear partial differential equations and their solutions exhibit localised features like vorticity and shocks. In spite of many advances, their accurate computation still remains challenging task. In this thesis, we review the theory of scalar conservation laws and their numerical solution techniques. In order to compute shocks accurately, we explore the use of moving grids that will automatically adapt the grid resolution to the solution features. We first study finite volume methods on non-uniform grids and then extend the scheme to moving grid case.

Contents

Chapter 1

Introduction

Conservation Laws occur in many contexts in Physics and these preserve certain measurable physical quantities of a physical system in time. The very basic laws of Conservation of Mass, Conservation of Energy, Conservation of Electric Charge, hold everywhere without any exceptions. The laws of Conservation of Momentum and Conservation of Angular Momentum also hold under certain conditions and play very important role in determination of dynamics of physical systems and thus it is very important to analyse and physically interpret solutions to the Conservation Laws.

Finite Volume approach has been used to closely analyse the solutions to both Linear and Nonlinear Conservation Laws. Advection equation which governs the flow in many physical systems like the flow of heat in a conductor, flow of ink dropped at the top of liquid, seawater currents in oceans etc. has been considered. Moreover, important phenomena like formation of Advection Fog is also governed by Advection equation. Burgers equation which is used to model phenomena like turbulent flow in a fluid, shock waves around nozzles and jets, vapour deposition on solid surface etc. has also been studied and its solutions have been analysed.

1.1 Overview

Uniform grid is the simplest one to start with for the numerical implementation of Conservation Laws. Uniform grid and non-uniform grid, including both smooth and random grids have been considered in numerical implementation. A certain types of boundary value problems have been considered. Proper error analysis has been done considering the difference of numerical solution with the cell averaged value of pointwise solution and different schemes/grids/parameters have been compared in terms of total error obtained. Various fluxes including first order Upwind Flux, Lax-Wendroff flux, Lax-Friedrichs Flux, Roe Flux, Godunov Flux have been compared.

As the grid is refined, error keeps on decreasing and convergence rate is a measure of rate decrease of error with decrease in grid width. Different schemes/grids /parameters have been compared based on the convergence rate obatined. Runge-Kutta(RK) schemes are used for time-descretization. First, second and third order Runge-Kutta schemes have been tried. Both Vertex-Centered and Cell-Centered approaches have been tested. First order and second order schemes spacial scheme have been tested and second order schemes are found to be far better than first order schemes. For the second order scheme, MINMOD limiter, which returns the argument with lowest absolute value, has been used with different-different values of parameter beta, in order to limit the left and right reconstructed states for calculating flux through faces.

Moving meshes are being widely used in numerical implementation of Partial Differential Equations. The focus can be on a subdomian of numerical domain and thus highly accurate solution can be obtained in the selected regions requiring more computational effort. There is a need to look for proper velocity for mesh movement which could render very accurate solution for a general initial condition. Proper error analysis has been done when different mesh velocity profiles independent of solution have been used for mesh movement.

If velocity based on the local solution is used for mesh movement, then it is found that finite volume cells near the discontinuity get compressed very soon in case of shock wave and thus long-time simulation is not possible. This suggests the need for a scheme which allows crossing of cells. AREPO moving mesh scheme was introduced by Volker Springel in 2009. In the present work, based on the idea used in his paper, AREPO moving mesh code has been implemeted and studied for Scalar Conservation Laws.

1.2 Objective

The objective of this research is to accurately simulate non-linear partial differential equations which govern fluid flows. The aim is also to find improved numerical solution techniques and to compute the shocks accurately.

1.3 Thesis Organization

Chapter 2 describes Conservation Laws including their solutions, important properties and characteristics.

The numerical study done on Static Meshes is described in Chapter 3. Finite Volume method used is described in detail. Different numerical flux schemes have been compared. First and higher order schemes spatial and temporal schemes have been used and compared based on Convergence studies.

Chapter 4 deals with moving meshes. Different profiles for mesh movement have been considered. Proper Converegence Analysis has been done and different profiles have been compared with each other. Important Numerical Flux schemes have also been compared. Mesh velocity independent of solution have been used and mesh velocities based on solution have also been used. Lagrangian velocity has been tried for mesh movement. Few ways of smoothing the mesh-velocity profile have also been tried in an attempt to avoid extensive compression of finite volume cells.

Chapter 5 describes the implemetaion of AREPO moving mesh method based on the idea originally introduced by Volker Springel. In this method, since cells used in Finite Volume method for storing solution are allowed to cross each other. Shock Wave can be seen to be moving in the mesh for long time even if lagrangian velocity is considered for mesh movement.

Chapter 2

Conservation Laws

Consider an equation of the following form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{f(U)}}{\partial x} = 0 \tag{2.1}$$

where $\mathbf{U}$ is a n-component vector, $\mathbf{F}$ is a second rank tensor with entry on i'th row and j'th column, F_{ij} being the flux of U_i in x_j direction. This equation is known as Conservation Law. Let us consider the 1-D case.

$$\frac{\partial u\left(x,t\right)}{\partial t} + \frac{\partial f\left(u\left(x,t\right)\right))}{\partial x} = 0 \tag{2.2}$$

This partial differential equation conserves total value of a quantity over space, $\int\limits_{-\infty}^{+\infty} u\left(t,x\right)dx$ in time and thus it is also termed as Conservation Law. Integral form of Conservation Law can also be obtained from differential form of Conservation Law (2.2) by integrating it over space and time.

On integrating (2.2) over arbitrary space-time interval $[x_1, x_2] \times [t_1, t_2]$,

$$\int\limits_{x_1}^{x_2}\int\limits_{t_1}^{t_2}\left(\frac{\partial u\left(x,t\right)}{\partial t}\right)dtdx = -\int\limits_{t_1}^{t_2}\int\limits_{x_1}^{x_2}\left(\frac{\partial f(u\left(x,t\right))}{\partial x}\right)dxdt$$

$$\int\limits_{x_1}^{x_2}\left(\int\limits_{t_1}^{t_2}\frac{\partial u\left(x,t\right)}{\partial t}dt\right)dx = -\int\limits_{t_1}^{t_2}\left(\int\limits_{x_1}^{x_2}\frac{\partial f(u\left(x,t\right))}{\partial x}dx\right)dt$$

$$\int\limits_{x_1}^{x_2}\left(u\left(x,t\right)\Big|_{t_1}^{t_2}\right)dx = -\int\limits_{t_1}^{t_2}\left(f(u\left(x,t\right))\Big|_{x_1}^{x_2}\right)dt$$

$$\int\limits_{x_1}^{x_2}\left[u(x,t_2) - u(x,t_1)\right]dx = -\int\limits_{t_1}^{t_2}\left[f(u(x_2,t)) - f(u(x_1,t))\right]dt \tag{2.3}$$

The above equation is the Integral form of the Conservation Law. Test function $\phi(x,t)$ is chosen which is assumed to have compact support in space and time which means the function vanishes outside a bounded domain in space and time. On left multiplying (2.2) by $\phi(x,t)$ and integrating over $[x_1,x_2]\times[t_1,t_2]$,

$$\int\limits_{t=0}^{t=+\infty}\int\limits_{x=+\infty}^{x=-\infty}\phi(x,t)\left(\frac{\partial u\,(x,t)}{\partial t}+\frac{\partial f(u)}{\partial x}\right)dxdt=0$$

On following the steps, we are finally left with

$$\Rightarrow\int\limits_{t=0}^{t=+\infty}\int\limits_{x=-\infty}^{x=+\infty}\left[f(u\,(t,x))\left(\frac{\partial\phi(x,t)}{\partial x}\right)+u\,(t,x)\left(\frac{\partial\phi(x,t)}{\partial t}\right)\right]dxdt=$$
$$\int\limits_{t=0}^{t=+\infty}\left((\phi(x,t)f(u\,(t,x)))\Big|_{x=-\infty}^{x=+\infty}\right)dt+\int\limits_{x=-\infty}^{x=+\infty}\left((\phi(x,t)u\,(t,x))\Big|_{t=0}^{t=+\infty}\right)dx$$

First term in RHS vanishes because $\phi(x,t)$ has compact support in x. Also since, $\phi(x,t)$ has compact support in t so the upper limit of second term in RHS is zero. So, the above equation reduces to

$$\begin{aligned}\int\limits_{t=0}^{t=+\infty}\int\limits_{x=-\infty}^{x=+\infty}\left[f(u\,(t,x))\left(\frac{\partial\phi(x,t)}{\partial x}\right)+u\,(t,x)\left(\frac{\partial\phi(x,t)}{\partial t}\right)\right]dxdt\\+\int\limits_{x=-\infty}^{x=+\infty}(\phi(0,x)u(0,x))\,dx=0\end{aligned}\tag{2.4}$$

This is another form of Conservation Law. This equation is satisfied by a particular kind of solutions called Weak Solutions which may not even be differentiable and they fail to satisfy differential form of Conservation Law (2.4). There can be infinitely many weak solutions to a given Conservation Law but just one of them is physically correct. The solution which is physically correct satisfies Entropy Condition, which is discussed later.

2.1 Basic Solutions

Conservation Laws can be written in differential form as a Partial Differential Equation(PDE). There are many kinds of solutions to PDE's. Each type of solution has its own advantage and importance.

• **Classical Solution**: Classical solution to any general PDE of order n is an expression which when substituted back satisfies the equation and is at least n times differentiable. The classical solution might not be infinitely differentiable.

- **Smooth Solution**: These are very nice and well-behaved solution to PDE. These solutions are infinitely differentiable.

- **Weak Solution**: Weak solution of PDE might not be differentiable even once but it behaves nicely under certain conditions. Weak solutions are sometimes very useful and easy to guess. For some higher order PDE's, it is very difficult to find Classical solutions but sometimes it is easy to guess weak solution. There can be infinitely many weak solutions to a given PDE.

2.2 Rankine-Hugoniot Condition

The jump conditions that are satisfied across any discontinuity are called Rankine-Hugoniot Jump condition. Integrating (2.2) over $x \in [x_s - \delta, x_s - \delta]$ where x_s is the position of the discontinuity yields,

$$\frac{d}{dt}\left(\int_{x_s-\delta}^{x_s+\delta} u\left(x,t\right) dx\right) + f(u)\Big|_{x_s-\delta}^{x_s+\delta} = 0 \tag{2.5}$$

Since integrand is discontinuous at (t, x_s), so integral in the first term can be split into two integrals with integrands continuous in the intervals $[x_s - \delta, x_s)$ and $(x_s, x_s + \delta]$,

$$\int_{x_s-\delta}^{x_s+\delta} u\left(x,t\right) dx = \int_{x_s-\delta}^{x_s} u\left(x,t\right) dx + \int_{x_s}^{x_s+\delta} u\left(x,t\right) dx$$

Then, (2.5) becomes

$$\frac{d}{dt}\left(\int_{x_s-\delta}^{x_s} u\left(x,t\right) dx + \int_{x_s}^{x_s+\delta} u\left(x,t\right) dx\right) + f(u)\Big|_{x_s-\delta}^{x_s+\delta} = 0$$

$$\Rightarrow \frac{d}{dt}\left(\int_{x_s-\delta}^{x_s} u\left(x,t\right) dx\right) + \frac{d}{dt}\left(\int_{x_s}^{x_s+\delta} u\left(x,t\right) dx\right) + f(u)\Big|_{x_s-\delta}^{x_s+\delta} = 0$$

By using Chain Rule of differentiation and Leibnitz Integral Rule, both terms on LHS can be evaluated and above expression reduces to

$$\Rightarrow \left[\int_{x_s-\delta}^{x_s} u_t(x,t)dx + u(x_s,t)\frac{d}{dt}x_s - u(x_s-\delta,t)\frac{d}{dt}(x_s-\delta)\right]$$
$$+ \left[\int_{x_s}^{x_s+\delta} u_t(x,t)dx + u(x_s+\delta,t)\frac{d}{dt}(x_s+\delta) - u(x_s,t)\frac{d}{dt}x_s\right] + f(u)\Big|_{x_s-\delta}^{x_s+\delta} = 0$$

Identical terms with opposite signs cancel out and the following expression is left behind.

$$\Rightarrow \left[\int\limits_{x_s-\delta}^{x_s} u_t(x,t)dx + u(x_s,t)\frac{d}{dt}x_s - u(x_s-\delta,t)\frac{d}{dt}(x_s-\delta)\right]$$
$$+\left[\int\limits_{x_s}^{x_s+\delta} u_t(x,t)dx + u(x_s+\delta,t)\frac{d}{dt}(x_s+\delta) - u(x_s,t)\frac{d}{dt}x_s\right] + f(u)\Big|_{x_s-\delta}^{x_s+\delta} = 0$$

$$\Rightarrow \int\limits_{x_s-\delta}^{x_s} u_t(x,t)dx - u(x_s-\delta,t)\frac{d}{dt}(x_s-\delta)+$$
$$\int\limits_{x_s}^{x_s+\delta} u_t(x,t)dx + u(x_s+\delta,t)\frac{d}{dt}(x_s+\delta) + f(u)\Big|_{x_s-\delta}^{x_s+\delta} = 0$$

$$\Rightarrow \int\limits_{x_s-\delta}^{x_s} u_t(x_s-\delta,t)dx + \int\limits_{x_s}^{x_s+\delta} u_t(x_s+\delta,t)dx$$
$$\left(u\left(x_s-\delta,t\right) - u\left(x_s+\delta,t\right)\right)\frac{dx_s}{dt} = -f(u)\Big|_{x_s-\delta}^{x_s+\delta}$$

Since $\lim_{\delta\to 0}\left(x_s+\delta\right)\to x_s$ and $\lim_{\delta\to 0}\left(x_s-\delta\right)\to x_s$, first two terms in LHS drop out. In the shorthand notations $\lim_{\delta\to 0}\left(u\left(x_s-\delta\right)\right) = u_l$ and $\lim_{\delta\to 0}\left(u(x_s+\delta)\right) = u_r$, above equation simplifies to

$$\left[u\left(x_l,t\right) - u\left(x_r,t\right)\right]\frac{dx_s}{dt} = -\left[f\left(u\left(x_r,t\right)\right) - f\left(u\left(x_l,t\right)\right)\right]$$
$$\Rightarrow \frac{dx_s}{dt} = \frac{f(u(x_r,t)) - f(u(x_l,t))}{u(x_r,t) - u(x_l,t)}$$

$\frac{dx_s}{dt}$ is basically the speed of propagation of discontinuity. It can be written in more convenient form by dropping t from the above equation.

$$\frac{dx_s}{dt} = \frac{f\left(u_r\right) - f\left(u_l\right)}{u_r - u_l}$$

2.3 Linear Conservation Law: Advection Equation

The flux function appearing in Conservation Law is linear in u, i.e.$f(u) = au, a \in \mathbb{R}$. Then, equation (2.2) reduces to

$$\frac{\partial u\left(x,t\right)}{\partial t} + a\frac{\partial u\left(x,t\right)}{\partial x} = 0 \tag{2.6}$$

Now, consider $\frac{du}{dt}$ along the straight lines given by $\frac{dx}{dt} = a, \; a \in \mathbb{R}$,

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \left(\frac{\partial u}{\partial x}\right)\left(\frac{dx}{dt}\right)$$

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \left(\frac{\partial u}{\partial x}\right) a$$

$$\frac{du}{dt} = -a\left(\frac{\partial u}{\partial x}\right) + a\left(\frac{\partial u}{\partial x}\right) = 0$$

The family of straight lines $x = at + k$ for arbitrary constant k have slope a and hence the solution is constant along these lines. The solution at any location at any time can be obtained by tracing the characteristic passing through this point back in time and then the solution at any position x at time t is same as the initial value at position x_0 at time t_0 where $x_0 = x - a\,(t - t_0)$. The initial profile just travels with velocity a towards right if $a > 0$ and towards left if $a < 0$.

2.4 Nonlinear Conservation Laws

The flux function $f(u)$ is not linear in u. Nonlinear Conservation Laws are special in the sense that the solutions to Conservation Laws are not always smooth. There are many types of discontinuities which can arise in this case.
Consider Conservation Law

$$\frac{\partial u\,(x,t)}{\partial t} = -\frac{\partial f(u\,(x,t))}{\partial x} \tag{2.7}$$

This could also be written in the form,

$$\frac{\partial u\,(x,t)}{\partial t} = -a\,(u\,(x,t))\,\frac{\partial u\,(x,t)}{\partial x}$$

where $a(u\,(x,t)) = f'(u)$.

This form gives some insight in analyzing the evolution of solution with time. Consider the curve

$$\mathbf{C}\colon \frac{dx(t)}{dt} = a(u\,(x,t))$$

where $a(u)$ is some function of u . It can be shown that the solution is constant along the curve the **C**.

Proof :

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \left(\frac{\partial u}{\partial x}\right)\left(\frac{dx}{dt}\right)$$

$$\Rightarrow \frac{du}{dt} = \frac{\partial u}{\partial t} + a(u)\left(\frac{\partial u}{\partial x}\right)$$

$$\Rightarrow \frac{du}{dt} = -a(u)\left(\frac{\partial u}{\partial x}\right) + a(u)\left(\frac{\partial u}{\partial x}\right) = 0$$

Lines of constant solution are called Characteristics which are given by

$$\frac{dx}{dt} = f'(u\,(t, x)) \tag{2.8}$$

where f' indicates the first order derivative with respect to u. The solution at time $t \geq t_0$ and position x is same as that at position x_0 at time t_0 such that points (x, t) and (x_0, t_0) lie on the same characteristic. The analysis of evolution of the solution with time can be carried out by means of characteristics. Similar to Advection Equation consideration, in this case too, tracing the characteristics back in time gives the solution at current time. A frequently occuring non-linear Conservation Law is **Burgers Equation** with the flux given by $f(u) = \frac{u^2}{2}$,

$$u_t + \left(u^2/2\right)_x \tag{2.9}$$

2.4.1 Discontinuities

Consider a nonlinear Conservation Law of the form (2.2). Consider the initial condition

$$u_0(x) \forall x \in [a, b] \tag{2.10}$$

Characteristics originating from some $x_0 \in [a, b]$ is given by

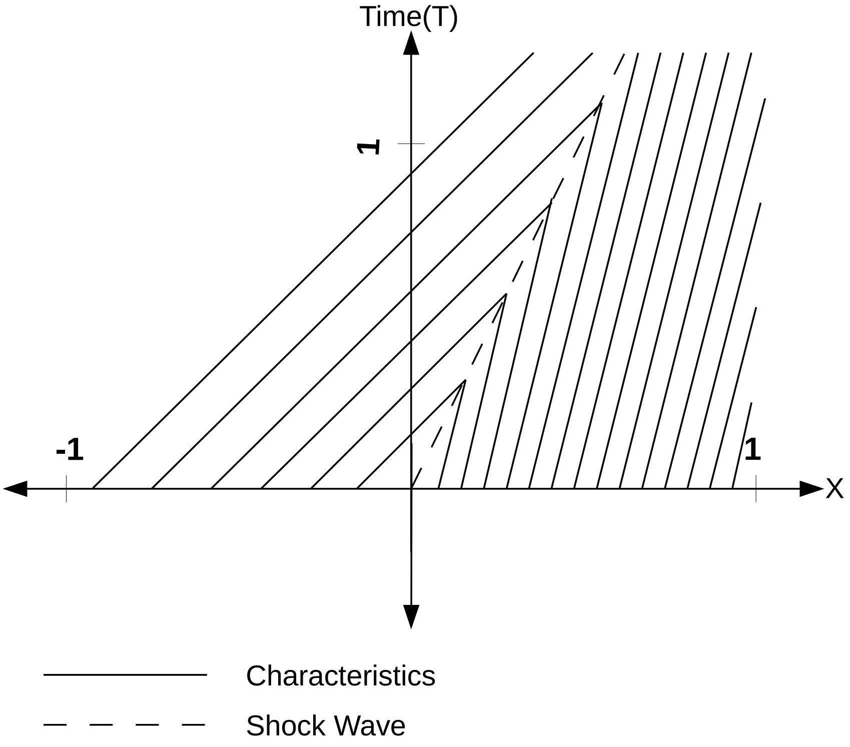
$$\frac{dx}{dt} = f'(u(x_0, 0)) \Rightarrow u(x_0 + f'(u(x_0, 0)), t) = u_0(x_0)$$

SHOCK WAVE:

Consider the case when lines $x = u(x_1, 0) + f'(u(x_1, 0))t$ and $x = u(x_2, 0) + f'(u(x_2, 0))t$ intersect, where $x_1, x_2 \in [a, b]$. In that case, solution does not exist at the point of intersection. The solution may be continuous on either side of the point of intersection but is discontinuous at the point of intersection itself. This type of discontinuity is called shock. The discontinuity propagates with shock speed give by Rankine-Hugoniot Condition. For simple illustration, consider solving Burgers Equation (2.4.1) for an initial condition given by Riemann Problem:

$$u_0(x) = \begin{cases} u_l, & \text{if } x \leq 0 \\ u_r, & \text{if } x > 0 \end{cases} \tag{2.11}$$

with $u_l > 0$ and $u_r < 0$. The shock speed given by $S = \frac{dx_s}{dt} = \frac{f(u_r) - f(u_l)}{u_r - u_l}$ which is then $S = \frac{{u_r}^2 - {u_l}^2}{u_r - u_l} = \frac{u_r + u_l}{2}$ for Burgers and thus depending on the sign of S, the wave travels either towards left or right.



Characteristics from left and right intersect thus giving rise to shock discontinuity. Function is discontinuous at the point of shock and may be continuous on either side of the shock. The characteristics originiating from the region with x<0 have correspond to speed 1 while the characteristics originating from right correspond to speed being 0.2. Hence the shock speed is 0.6 and so the shock travels towards right.

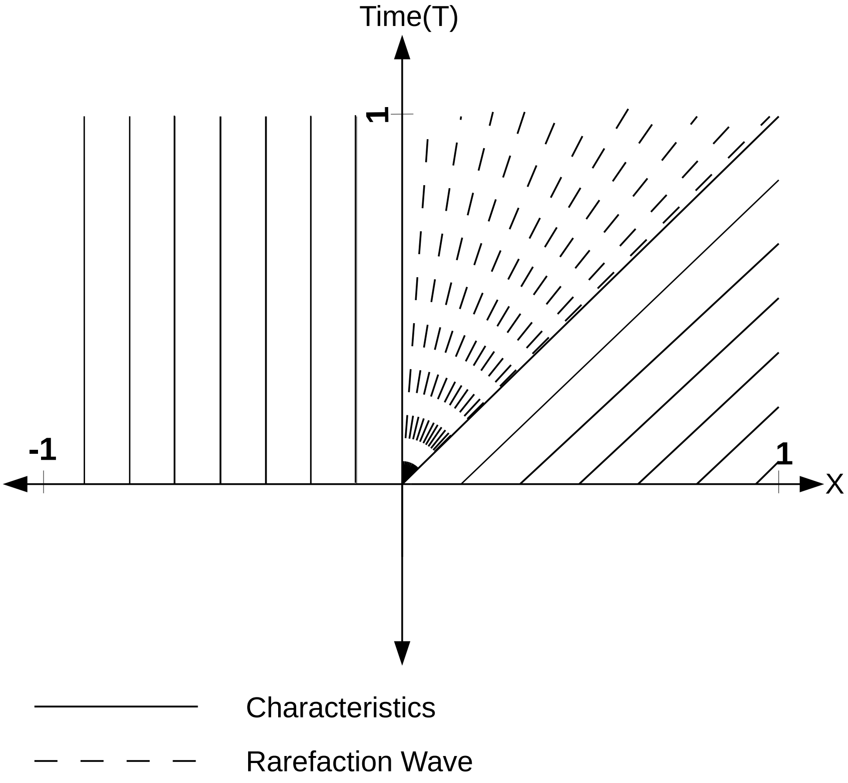
Figure 2.1: *A typical shock wave generated when* $f'(u_l) > f'(u_r)$.

RAREFACTION WAVE:

This type of discontinuity arises when the characteristics do not cross but they do not span the entire space. In this case there can be many weak solutions but only one of them is physically correct solution, which is vanishing viscosity solution. For example, for the Burgers Equation , the solution satisfy viscous Burgers Equation , in the limit as $\nu \to 0$ mentioned below.

$$u_t + uu_x = \nu u_{xx}$$

Entropy condition is used to find physically correct solution to the Burgers Equation which is described later in this section.



Characteristics from left region are confined to the left region since they correspond to zero velocity and the characteristics from the right region correspond to velocity 1. In the portion which is not touched upon by characteristics, there exist infinitely many weak solutions but the solution drawn here is the solution which indeed satisfies the entropy condition.

Figure 2.2: *Rarefaction front obtained when* $f'(u_l) = 0$ *and* $f'(u_r) = 1$.

In this case, we look for solution which satisfy the integral form of Conservation Law 2.3 and also the condition (2.4). This can be illustrated by a simple example. Again consider solving Burgers Equation for an initial condition given by Riemann problem in interval $[-1, 1]$

$$u_0(x) = \begin{cases} 1, & \text{if } x \leq 0 \\ 0, & \text{if } x > 0 \end{cases}$$

Here there are regions which are not spanned by characteristics and there can be many infinitely many weak solutions. Two of them are shown in $fig.$ 2.3 . However, Entropy Condition selects the physically correct solution. Here, Solution A is the physically correct solution.

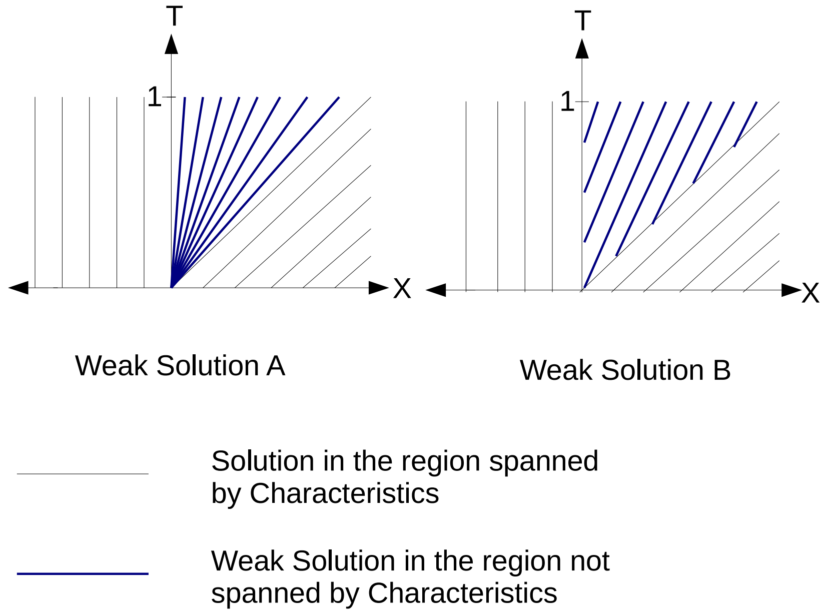

T
1
X
T
1
X
Weak Solution A
Weak Solution B
Solution in the region spanned by Characteristics
Weak Solution in the region not spanned by Characteristics

Figure 2.3: *Two weak solutions in case of rarefaction wave*

Entropy Condition: It puts a restriction on physically correct solutions. It states that a physically correct solution $u(x), x \in [a, b]$ must satisfy

$$f'(u_l) > S > f'(u_r)$$

where u_l is the upstream value of function, u_r is the downstream value of function and $S = \frac{f(u_r)-f(u_r)}{u_r-u_l}$, is the speed of propagating discontinuity.

2.4.2 Solutions to Riemann Problems for Burgers Equation

Burgers Equation which has already been introduced is a nonlinear Conservation Law with the flux given by

$$f(u) = \frac{u^2}{2}$$

The velocity with which Characteristics move is given by

$$f'(u) = \frac{\mathrm{d}}{\mathrm{d}u}\left(\frac{u^2}{2}\right) = u \tag{2.12}$$

Solutions to (2.4.1) for an arbitrary initial condition $u_0(x), x \in [a, b]$, where $[a, b]$ is the domain of interest, can be obtained by tracing the characteristics back in time. Characteristics are the curves of constant solution and thus the solution at any position and time $u(x, t)$ is same as $u_0(x_0)$ if points (x, t) and (x_0, t_0) lie on the same characteristic. For instance, consider the following Initial Value

Problem(IVP)

$$u(x,0) = \begin{cases} u_r, & \text{if } x \leq c \\ u_l, & \text{if } x > c \end{cases}$$

where $c \in \mathbb{R}$. The solution at any position and time $u(x,t)$ can be obtained by finding x_0 which satisfies

$$x_0 + u_0(x_0)t = x$$

For the IVP considered, we need to look for x_0 satisfying following conditions

$$\begin{aligned} C1: \ & x_0 + u_r t = x \ \text{ for } x_0 > 0 \\ C2: \ & x_0 + u_l t = x \ \text{ for } x_0 < 0 \end{aligned} \tag{2.13}$$

In general, (2.13) can have single, multiple, infinite or no solutions.

Here, various initial conditions are considered in domains $[0,1]$ and $[-1,1]$. In the equations that follow, the expression on the left is the initial condition and the expression on the right is the solution at any later time t.

- $u(x,0) = \begin{cases} 1, & \text{if } x \leq 0 \\ 0, & \text{if } x > 0 \end{cases}$ $\quad u(x,t) = \begin{cases} 1, & \text{if } x \leq \frac{1}{2}t \\ 0, & \text{if } x > \frac{1}{2}t \end{cases}$

In the above case, the shock wave travels with speed $\frac{1}{2}$

- $u(x,0) = \begin{cases} -1, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$ $\quad u(x,t) = \begin{cases} -1, & \text{if } x \leq -t \\ x/t, & \text{if } -t < x \leq t \\ 1, & \text{if } x > t \end{cases}$

- $u(x,0) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$ $\quad u(x,t) = \begin{cases} 0, & \text{if } x \leq 0 \\ x/t, & \text{if } 0 < x < t \\ 1, & \text{if } x > t \end{cases}$

- $u(x,0) = \begin{cases} -1, & \text{if } x \leq 0 \\ 0, & \text{if } x \geq 0 \end{cases}$ $\quad u(x,t) = \begin{cases} -1, & \text{if } x \leq -t \\ x/t, & \text{if } -t < x < 0 \\ 0, & \text{if } x > 0 \end{cases}$

□ Consider the following initial condition,

$$u(x,0) = \begin{cases} -1, & \text{if } x <= 0 \\ 1 - 2x, & \text{if } 0 < x < \frac{1}{2} \\ 0, & \text{if } x > \frac{1}{2} \end{cases}$$

It has the following solution

$$u\left(x, t < \frac{1}{2}\right) = \begin{cases} \frac{x-\frac{1}{2}}{t-\frac{1}{2}}, & \text{if } t < x < \frac{1}{2} \\ 0, & \text{if } x > \frac{1}{2} \end{cases} \quad u\left(x, \frac{1}{2} < t < 1\right) = \begin{cases} -1, & \text{if } x \leq 0 \\ -1, & \text{if } 0 < x < \frac{1}{2} + \frac{t}{2} \\ 0, & \text{if } x > \frac{1}{2} + \frac{t}{2} \end{cases}$$

□ Consider the following initial condition,

$$u(x,0) = \begin{cases} -1, & \text{if } x \leq 0 \\ 1 - x, & \text{if } 0 < x < \frac{1}{2} \\ \frac{1}{2}, & \text{if } x > \frac{1}{2} \end{cases}$$

It has the following solution

$$u\,(t < 1) = \begin{cases} -1, & \text{if } x \leq t \\ \frac{1}{2} + \frac{x-\frac{1}{2}-\frac{1}{2}t}{t-1}, & \text{if } t < x < \frac{1}{2} + \frac{1}{2}t \\ 0, & \text{if } x > \frac{1}{2} + \frac{1}{2}t \end{cases} \quad u\,(t > 1) = \begin{cases} -1, & \text{if } x \leq 1 + \frac{t}{2} \\ 0, & \text{if } x > 1 + \frac{t}{2} \end{cases}$$

In some of the above cases, shock wave gets generated when the characteristics originating from different points intersect and then there is no unique solution at points of intersection. For a given initial smooth data, the time after which shock gets generated can be calculated as follows. In order to calculate the time after which shock forms in case of smooth initial profile $u_0(x), x \in D$(domain), suppose the characteristics starting from two locations $x_1, x_2 \in [a, b]$ cross each other at (x, t). Then the following two equations must hold simultaneously.

$$\begin{aligned} x_1 + u_0(x_1)t &= x \\ x_2 + u_0(x_2)t &= x \end{aligned} \tag{2.14}$$

$$\begin{aligned} \Rightarrow x_1 + u_0(x_1)t &= x_2 + u_0(x_2)t \\ \Rightarrow t &= -\frac{x_2 - x_1}{u_0(x_2) - u_0(x_1)} \\ \Rightarrow t &= -\frac{1}{\frac{u_0(x_2)-u_0(x_1)}{x_2-x_1}} \end{aligned} \tag{2.15}$$

The smallest time which corresponds to the shock formation is obtained by minimizing (2.15) over all pairs of x_1 and x_2. This is equivalent to maximizing the denominator of (2.15) over all pairs of x_1 and x_2. So the problem reduces to finding the following quantity

$$Q = \min_{x_1, x_2 \in D} \frac{u_0(x_2) - u_0(x_1)}{x_2 - x_1}$$

Since

$$\frac{u_0(x_2) - u_0(x_1)}{x_2 - x_1} \geq \min_{x_3 \in D} u_0'(x_3) \ \forall \ x_1, x_2 \in D$$

So

$$Q = \min_{x_1, x_2 \in D} \frac{u_0(x_2) - u_0(x_1)}{x_2 - x_1} = \min_{x_3 \in D} u_0'(x_3)$$

Thus, (2.15) yields

$$T_{\min} = -\frac{1}{\min_{x \in D} u_0'(x)} \tag{2.16}$$

Thus, for the smooth initial profile, necessary condition for shock generation is that the space derivative of initial solution is negative somewhere, i.e. $u_0'(x) < 0$ for some x in domain.

Chapter 3

Finite Volume Method on Static Mesh

3.1 General Overview

Various time discretisation and space discretisation techniques are used to analyse the solution of a Conservation Law numerically. The domain is divided into finitely many cells and the solution obtained in any cell actually represents the average value in that cell. Consider solving (2.2) in the 1-**D** domain $D := [a, b]$ for the some initial condition $u_0(x)$ considered. The domain $[a, b]$ under consideration is divided into cells in the following way:

$$D = \underset{\mathbf{i \in K}}{\cup} \left(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} \right) \text{ or } D = \underset{\mathbf{i \in K}}{\cup} C_i$$

where $K = \{1, 2, ..., N-2, N\}$ is set of nodes , N being the total number of cells considered and $x_{\frac{1}{2}} = a$, $x_{N+\frac{1}{2}} = b$ and $C_i = \left(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} \right), i \in K$ are the cells. The leftmost face is taken to be at the left boundary of the the domain and the rightmost face to be at the right boundary. Cells are indexed by integers $i \in K$, and the faces are indexed by half-integers $\left(i + \frac{1}{2}\right)' s \; \forall i \in \{0\} \cup K$. The common face of any two cells, cell i and cell $i+1$ is indexed by $\left(i + \frac{1}{2}\right)$. Solution on each cell depends on the fluxes coming in and going out. Each cell has two faces and each face has two adjoining cells. The flux across any face is positive for one adjoining cell and negative for the other one. So for any cell considered at random, contribution from one face is added and contribution from the other face is subtracted.

Consider the differential form of Conservation Law (2.2)

$$\frac{\partial u\,(x,t)}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

$$\Rightarrow \frac{\partial u\,(x,t)}{\partial t} = -\frac{\partial f(u)}{\partial x} \tag{3.1}$$

Consider finding solution to Conservation Law with given initial condition in an arbitrary domain Ω

$$\Omega = \bigcup_i \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$$

$$u\left(x,0\right) = u_0\left(x\right)$$

On integrating (2.2) over space-time interval $\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times \left[t_n, t_{n+1}\right]$,

$$\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int\limits_{t_n}^{t_{n+1}} \left(\frac{\partial u\left(x,t\right)}{\partial t}\right) dtdx = - \int\limits_{t_n}^{t_{n+1}} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\frac{\partial f(u\left(x,t\right))}{\partial x}\right) dxdt$$

$$\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\int\limits_{t_n}^{t_{n+1}} \frac{\partial u\left(x,t\right)}{\partial t} dt \right) dx = - \int\limits_{t_n}^{t_{n+1}} \left(\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial f(u\left(x,t\right))}{\partial x} dx \right) dt$$

$$\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(u\left(x,t\right) \Bigg|_{t_n}^{t_{n+1}} \right) dx = - \int\limits_{t_n}^{t_{n+1}} \left(f(u\left(x,t\right)) \Bigg|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \right) dt$$

$$\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[u\left(x,t_{n+1}\right) - u\left(x,t_n\right)\right] dx = - \int\limits_{t_n}^{t_{n+1}} \left[f\left(u\left(x_{i+\frac{1}{2}},t\right)\right) - f\left(u\left(x_{i-\frac{1}{2}},t\right)\right)\right] dt \tag{3.2}$$

Suppose the average value of solution in cell i is denoted by z_i

$$z_i^n = \frac{\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u\left(x,t_n\right) dx}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}$$

$$\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u\left(x,t_n\right) dx = z_i^n \left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right)$$

Further, following approximation is used

$$\int\limits_{t_n}^{t_{n+1}} f\left(u\left(x_{i+\frac{1}{2}},t\right)\right) dt \approx F_{i+\frac{1}{2}}\left(t_{n+1} - t_n\right)$$

where $F_{i+\frac{1}{2}} = g\left(u^L_{i+\frac{1}{2}}, u^R_{i+\frac{1}{2}}\right)$ is numerical flux at the face, an approximation to physical flux, with $u^L_{i+\frac{1}{2}}$ and $u^R_{i+\frac{1}{2}}$ being left and right reconstructed values at the face.

Then (3.2) becomes

$$\left(z_i^{n+1}-z_i^n\right)\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)=-\left[F_{i+\frac{1}{2}}\left(t_{n+1}-t_n\right)-F_{i-\frac{1}{2}}\left(t_{n+1}-t_n\right)\right]$$

$$\left(z_i^{n+1}-z_i^n\right)\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)=-\left(F_{i+\frac{1}{2}}-F_{i-\frac{1}{2}}\right)\left(t_{n+1}-t_n\right)$$

$$\frac{z_i^{n+1}-z_i^n}{t_{n+1}-t_n}+\frac{F_{i+\frac{1}{2}}-F_{i-\frac{1}{2}}}{x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}}=0$$

Here

$$z_i^n=u\left(x_i,t_n\right)\quad\forall x\in\left[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}\right],t\in\left[t_n,t_{n+1}\right]$$

$$z_i^{n+1}=z_i^n-\frac{t_{n+1}-t_n}{x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}}\left(F_{i+\frac{1}{2}}-F_{i-\frac{1}{2}}\right)\tag{3.3}$$

z_i^n is an approximation to the solution in i'th cell at any timestep n.

3.2 Space discretisation

3.2.1 First Order Scheme

In the first order schemes, for evaluating numerical flux through the common face of cell j and cell $j+1$, left state is considered to be cell average in j'th cell, u_j^n and the right state is considered to be cell average in $(j+1)'$th cell, u_{j+1}^n, without any reconstruction i.e. $F_{j+\frac{1}{2}}$=$g(u_l,u_r)$=$g(u_j^n,u_{j+1}^n)$. This is very less accurate scheme and so mostly more accurate higher order schemes are used in numerical experiments.

3.2.2 Second Order Scheme

Linear reconstruction is done at the face to construct left and right states. These reconstructed values can be further limited by use of a proper limiter. One way of reconstructing left and right states at the face at $x_{j+\frac{1}{2}}$ is by using *minmod* limiter,
Left state :

$$u_{j+\frac{1}{2}}^L=u_j+\left(x_{j+\frac{1}{2}}-x_j\right)\phi\left(\beta\left(\frac{u_j-u_{j-1}}{x_j-x_{j-1}}\right),\frac{u_{j+1}-u_{j-1}}{x_{j+1}-x_{j-1}},\beta\left(\frac{u_{j+1}-u_j}{x_{j+1}-x_j}\right)\right)\tag{3.4}$$

Right state :

$$u_{j+\frac{1}{2}}^R=u_{j+1}-\left(x_{j+1}-x_{j+\frac{1}{2}}\right)\phi\left(\beta\left(\frac{u_{j+1}-u_j}{x_{j+1}-x_j}\right),\frac{u_{j+2}-u_j}{x_{j+2}-x_j},\beta\left(\frac{u_{j+1}-u_j}{x_{j+1}-x_j}\right)\right)\tag{3.5}$$

where β is a parameter of choice and ϕ is the *minmod* function which returns the argument with the lowest absolute value.

$$minmod(x_1,x_2,x_3) = \begin{cases} sign(x_1)\;\; \min\left(\min|x_1|\left(|x_2|,|x_3|\right)\right), & \text{if } x_1x_2 \geq 0,\; x_2x_3 \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

By using the recontructed states obtained from (3.5) and (3.4), flux across the face can be calculated and this can be used to finally calculate u_j^{n+1} from u_j^n using (3.3)

3.2.3 More Accurate Scheme: Second Order accurate Central Slope

The idea is to look for a first order derivative which is second order accurate. Starting with the linear combination of left and right slopes s_l and s_r,

$$c.s_r + d.s_l = c\left(\frac{u_{j+1}-u_j}{x_{j+1}-x_j}\right) + d\left(\frac{u_j - u_{j-1}}{x_j - x_{j-1}}\right) \tag{3.6}$$

where c and d are arbitrary coefficients.

Taylor expanding u_{j+1} upto second order in $\left(x_{j+1}-x_j\right)$ gives

$$u_{j+1} = u_j + \left(x_{j+1}-x_j\right)\frac{\partial u_j}{\partial x} + \frac{\left(x_{j+1}-x_j\right)^2}{2}\frac{\partial^2 u_j}{\partial x^2} \tag{3.7}$$

Taylor expanding u_{j-1} upto second order in $\left(x_j - x_{j-1}\right)$ gives

$$u_{j-1} = u_j - \left(x_j - x_{j-1}\right)\frac{\partial u_j}{\partial x} + \frac{\left(x_j - x_{j-1}\right)^2}{2}\frac{\partial^2 u_j}{\partial x^2} \tag{3.8}$$

By using (3.7) and (3.8), (3.6) becomes

$$\begin{aligned} c.s_r + d.s_l = c &\left(\frac{\left[u_j + \left(x_{j+1}-x_j\right)\frac{\partial u_j}{\partial x} + \frac{\left(x_{j+1}-x_j\right)^2}{2}\frac{\partial^2 u_j}{\partial x^2}\right] - u_j}{x_{j+1}-x_j}\right) \\ +d &\left(\frac{\left[u_j - \left(x_j - x_{j-1}\right)\frac{\partial u_j}{\partial x} + \frac{\left(x_j - x_{j-1}\right)^2}{2}\frac{\partial^2 u_j}{\partial x^2}\right] - u_j}{x_{j-1}-x_j}\right) \end{aligned} \tag{3.9}$$

$$\Rightarrow c.s_r + d.s_l = (c+d)\frac{\partial u_j}{\partial x} + \frac{1}{2}\left[c\left(x_{j+1}-x_j\right) - d\left(x_j - x_{j-1}\right)\right]\frac{\partial^2 u_j}{\partial x^2} \tag{3.10}$$

Allowing the coefficient of $\frac{\partial^2 u_j}{\partial x^2}$ go to zero gives a relation between c and d

$$c\left(x_{j+1} - x_j\right) - d\left(x_j - x_{j-1}\right) = 0$$

$$\Rightarrow d = c\left(\frac{x_{j+1} - x_j}{x_j - x_{j-1}}\right)$$

Since the coefficients c and d must add up to 1, we get

$$c = \frac{x_j - x_{j-1}}{x_{j+1} - x_{j-1}}\ ,\ d = \frac{x_{j+1} - x_j}{x_{j+1} - x_{j-1}}$$

Plugging d in terms of c back in (3.10)

$$c.s_r + c\left(\frac{x_{j+1} - x_j}{x_j - x_{j-1}}\right).s_l = \left(c + c\frac{x_{j+1} - x_j}{x_j - x_{j-1}}\right)\frac{\partial u_j}{\partial x}$$

$$\frac{\partial u_j}{\partial x} = \frac{s_r + s_l\left(\frac{x_{j+1}-x_j}{x_j-x_{j-1}}\right)}{1 + \frac{x_{j+1}-x_j}{x_j-x_{j-1}}}$$

$$\frac{\partial u_j}{\partial x} = \frac{s_r\left(x_j - x_{j-1}\right) + s_l\left(x_{j+1} - x_j\right)}{\left(x_j - x_{j-1}\right) + \left(x_{j+1} - x_j\right)}$$

$$\frac{\partial u_j}{\partial x} = \frac{\left(\frac{u_{j+1}-u_j}{x_{j+1}-x_j}\right)\left(x_j - x_{j-1}\right) + \left(\frac{u_j-u_{j-1}}{x_j-x_{j-1}}\right)\left(x_{j+1} - x_j\right)}{x_{j+1} - x_{j-1}}$$

$$\frac{\partial u_j}{\partial x} = \frac{\left(\frac{x_j-x_{j-1}}{x_{j+1}-x_j}\right)\left(u_{j+1} - u_j\right) + \left(\frac{x_{j+1}-x_j}{x_j-x_{j-1}}\right)\left(u_j - u_{j-1}\right)}{x_{j+1} - x_{j-1}}$$

$$\frac{\partial u_j}{\partial x} = \frac{\frac{1}{R_j}\left(u_{j+1} - u_j\right) + R_j\left(u_j - u_{j-1}\right)}{x_{j+1} - x_{j-1}} \qquad \left(R_j = \frac{x_{j+1}-x_j}{x_j-x_{j-1}}\right)$$

The resulting first order derivative is second order accurate.

3.3 Basic Finite Volume approches

- **Cell-Centered Approach**: In this approach, the solution is stored at the centre of the cell and the stored value is the average value of the numerical solution in the finite volume cell. The node lies at the geometric centre of the cell. In simple 1D case, mathematically,

$$x_i = \frac{x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}}}{2} \quad \forall \quad i \in \{1, 2, ..., N-1, N\}$$

- **Vertex-Centered Approach**: In this approach, the solution is stored at the vertices. The vertex is equidistant from all the nearby nodes. In simple 1D case, mathematically,

$$x_{i+\frac{1}{2}} = \frac{x_i + x_{i+1}}{2} \quad \forall \quad i \in \{1, 2, ..., N-1\}$$

and leftmost and righmost faces being at the leftmost and rightmost boundary respectively.

3.4 Grids Considered

3.4.1 Uniform Grid

Nodes are equally spaced and their locations are given by

$$x_i = x_{min} + ih \quad \forall \quad i \in \{1, 2, ..., N-1, N\}$$

where $h = \frac{x_{max} - x_{min}}{N}$ is the grid spacing. Faces are also equally spaced and their locations are given by

$$x_{i+\frac{1}{2}} = x_{min} + \left(i - \frac{1}{2}\right) h \quad \forall \quad i \in \{1, 2, ..., N, N+1\}$$

3.4.2 Smooth Grid

Two types of smooth grids have been considered. In one case, grid points are clustered in the middle and the spacing is large at the ends. In the other type of smooth grid, grid points are clustered at the ends and the spacing is large at the centre.

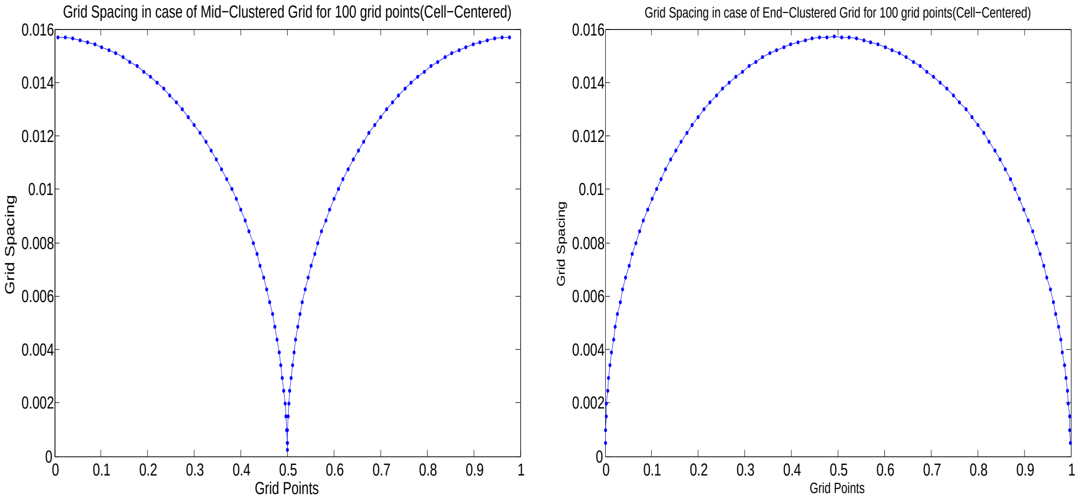


Figure 3.1: Smooth Meshes used in Numerical Implementation(Cell-Centered scheme is considered): Cells are clustered in the middle in one case, and at the ends in the other case

3.4.3 Random Grid

Random Grid: Generation when implementing Vertex-Centered Scheme

Initially uniformly spaced nodes are generated and then they are perturbed a bit from their position. The perturbation is random which could be on both left and right, and the degree of perturbation is controlled by a parameter α , more the α, more the randomness and vice-verse. Finally cell boundaries are taken to be at the midpoint of adjacent nodes.

Suppose that total number of cells considered is N and total number of faces is $N+1$. The following steps are followed in the generation of the random mesh.

- Initially, for i'th node,

$$x_i = x_{min} + \left(i + \frac{1}{2}\right)\left(\frac{x_{max} - x_{min}}{N}\right), i \in \{1, 2, ..., N-1, N\}$$

where x_i denotes location of i'th node, x_{min} is the left boundary and x_{max} is the right boundary.

- Slight perturbation is added

$$x_i = x_i + \alpha(1 - 2r)h$$

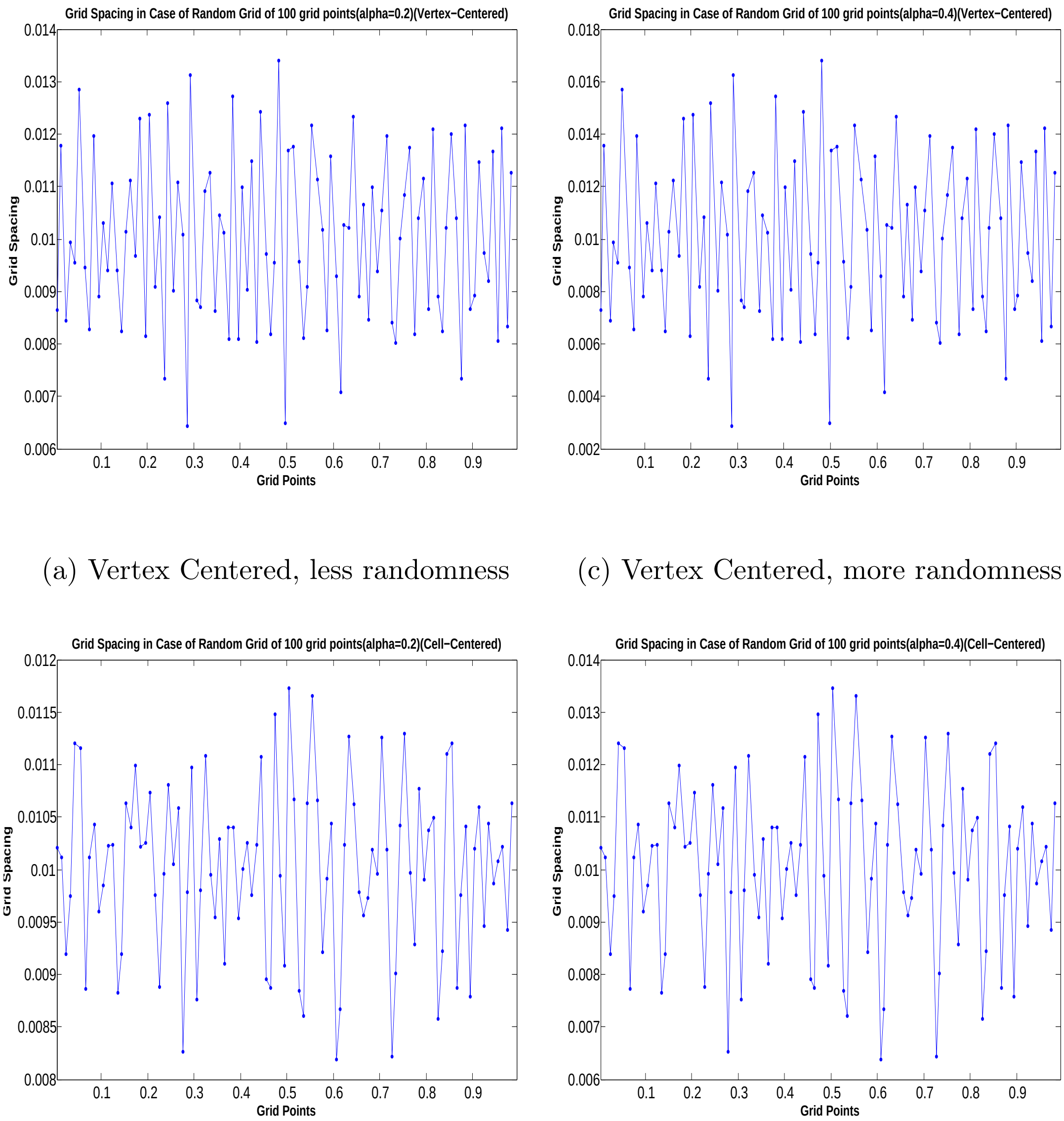
where $h = \frac{x_{max} - x_{min}}{N}$ is the node spacing in uniform grid, $r \in [0, 1]$ is a uniformly distributed random number and α is a parameter which can be varied as long as adjacent nodes do not cross each other.

- Faces are then taken to be at the midpoint of nodes of adjoining cells.

$$x_{i+\frac{1}{2}} = \frac{x_i + x_{i+1}}{2}, i \in \{1, 2, ..., N-1, N, N\}$$

The following plots show the spacing between consecutive grid points.

The following plots show the spacing between consecutive grid points



(a) Vertex Centered, less randomness

(c) Vertex Centered, more randomness

(b) Cell Centered, less randomness

(d) Cell Centered, more randomness

Figure 3.2: Random Grids used in Numerical Experiments

Random Grid: Generation when implementing Cell-Centered Scheme

Initially uniformly spaced faces are generated and then the ones at the the interior are perturbed a bit from their position. The perturbation could be on both right and left and the degree of perturbation is controlled by a parameter α , more the α, more the randomness and vice-verse. Finally nodes are taken to be at the midpoint of adjacent nodes.

Suppose that total number of cells considered is N and total number of faces is $N + 1$. The following steps are followed in the generation of the random mesh.

- Initially, for i'th face,

$$x_{i-\frac{1}{2}} = x_{min} + (i-1)\left(\frac{x_{max} - x_{min}}{N}\right), \quad i \in \{1, 2, ..., N-1, N, N+1\}$$

where $x_{i-\frac{1}{2}}$ denotes location of i'th face, x_{min} is the left boundary and x_{max} is the right boundary.

- Slight perturbation is added

$$x_{i-\frac{1}{2}} = x_{i-\frac{1}{2}} + \alpha(1-2r)h, \quad i \in \{2, ..., N-1, N, N\}$$

where $h = \frac{x_{max}-x_{min}}{N}$ is the node spacing in uniform grid, $r \in [0,1]$ is a uniformly distributed random number and α is a parameter which can be varied as long as adjacent nodes do not cross each other.

- Nodes are taken to be at the midpoint of left and right faces of the cell

$$x_i = \frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2}, \quad i \in \{1, 2, ..., N-1, N, N\}$$

3.5 CFL Condition

Basic requirement to get convergence to the true solution is that numerical domain must be contained in physical domain. CourantâĂŞFriedrichsâĂŞLewy(CFL) condition requires

$$\left|\max_i f'(u_i^n)\left(\frac{dt}{\Delta x_i}\right)\right| \leq N_{CFL}$$

where $f'(u_i^n) = \partial f(u)/\partial u$ at u_i^n. N_{CFL} is called CFL number. It has been proved that CFL condition has to hold for the numerical solutions to converge to the actual solution. This is a necessary but not a sufficient condition. If CFL condition is violated then the numerical solution cannot converge to the actual solution. CFL number must be less than or equal to one for proper convergence for First Order scheme.

3.6 Conservative Form

Any Numerical Scheme for Conservation Laws is said to be in Conservative Form if it can be written in the following form

$$\begin{aligned} u_j^{n+1} = u_j^n + \frac{t_{n+1} - t_n}{x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}} \Big[& F\left(u_{j-a_1+1}, u_{j-a_1+2}, u_{j-a_1+3}, ..., u_{j+a_2}, u_{j+a_2+1}\right) \\ & -F\left(u_{j-a_1}, u_{j-a_1+1}, u_{j-a_1+2}, ..., u_{j+a_2-1}, u_{j+a_2}\right)\Big] \end{aligned} \tag{3.11}$$

where $a_1, a_2 \in \mathbf{Z}^+$. The function F used in the above equation is called the Numerical Flux function.

Properties of Numerical Flux

• **Smoothness**: Since in conservative form (3.11), total number of arguments is $a_1 + a_2 + 1$, which can be denoted by $\{z_1, z_2, ..., z_{a_1+a_2+1}\}$, $z_i \in \mathbb{R}\ \forall i \in \{1, 2, ..., a_1 + a_2 + 1\}$, then smoothness requires

$$F\left(z_1, z_2, ..., z_{a_1+a_2+1}\right) \to f(z') \quad \text{as} \quad z_i \to z'\ \forall i \in \{z_1, z_2, ..., z_{a_1+a_2+1}\}$$

where f is the physical flux and $z' \in \mathbb{R}$.

• **Consistency**: Suppose all the arguments of numerical flux are the same then the it must be the same as physical flux with the same argument.

$$F\left(z', z', ..., z'\right) = f(z')$$

where f is the physical flux.

3.7 Total Variation Diminishing Schemes

3.7.1 TVD Property

Suppose u_j^n denotes the numerical solution at (x_i, t_n), then *Total Variation* at any time t_n is defined as follows: $TV(n) = \sum_j \left(|u_j - u_{j-1}|\right)$
Any Scheme is said to be TVD preserving if Total Variation is monotonically decreasing function of time.

i.e. if $TV(u^{n+1}) \leq TV(u^n)\ \forall\ n \in \mathbb{Z}^+$

3.7.2 Harten's Lemma

Consider a scheme which can be written in the following form

$$u_i^{n+1} = u_i^n - C_{i-\frac{1}{2}}^n \Delta u_{i-\frac{1}{2}} + D_{i+\frac{1}{2}}^n \Delta u_{i+\frac{1}{2}}^n \tag{3.12}$$

with coefficients $C_{i-\frac{1}{2}}^n$ and $D_{i+\frac{1}{2}}^n$ and where u_i^n denotes the solution at (t_n, x_i), $\Delta u_{i-\frac{1}{2}}^n = u_i - u_{i-1}$ and $\Delta u_{i+\frac{1}{2}}^n = u_{i+1}^n - u_i^n$.

Lemma : If the coefficients $C_{i-\frac{1}{2}}^n$ and $D_{i+\frac{1}{2}}^n$ satisfy the following conditions $\forall\ n$,
• $C_{i+\frac{1}{2}}^n > 0$
• $D_{i+\frac{1}{2}}^n > 0$
• $C_{i+\frac{1}{2}}^n + D_{i+\frac{1}{2}}^n \leq 0$
then the scheme is TVD
Proof :
Replacing i by $i + 1$ in (3.12) yields

$$u_{i+1}^{n+1} = u_{i+1}^n - C_{i+\frac{1}{2}}^n \Delta u_{i+\frac{1}{2}}^n + D_{i+\frac{3}{2}}^n \Delta u_{i+\frac{3}{2}}^n \tag{3.13}$$

(3.13) - (3.12) gives

$$u_{i+1}^{n+1} - u_i^{n+1} = u_{i+1}^n - u_i^n - C_{i+\frac{1}{2}}^n \Delta u_{i+\frac{1}{2}}^n + C_{i-\frac{1}{2}}^n \Delta u_{i-\frac{1}{2}}^n + D_{i+\frac{3}{2}}^n \Delta u_{i+\frac{3}{2}}^n - D_{i+\frac{1}{2}}^n \Delta u_{i+\frac{1}{2}}^n$$

$$\Rightarrow u_{i+1}^{n+1} - u_i^{n+1} = u_{i+1}^n - u_i^n - C_{i+\frac{1}{2}}^n \left(u_{i+1}^n - u_i^n\right) + C_{i-\frac{1}{2}}^n \left(u_i^n - u_{i-1}^n\right)$$
$$+ D_{i+\frac{3}{2}}^n \left(u_{i+2}^n - u_{i+1}^n\right) - D_{i+\frac{1}{2}}^n \left(u_{i+1}^n - u_i^n\right)$$

$$\Rightarrow u_{i+1}^{n+1} - u_i^{n+1} = C_{i-\frac{1}{2}}^n \left(u_i^n - u_{i-1}^n\right) + D_{i+\frac{3}{2}}^n \left(u_{i+2}^n - u_{i-1}^n\right)$$
$$+ \left(1 - C_{i+\frac{1}{2}}^n - D_{i+\frac{1}{2}}^n\right) \left(u_{i+1}^n - u_i^n\right)$$

Taking the modulus on both sides and by using the fact that the three terms in RHS have positive coeffiecients, we get

$$\left|u_{i+1}^{n+1} - u_i^{n+1}\right| \leq C_{i-\frac{1}{2}}^n \left|u_i^n - u_{i-1}^n\right| + D_{i+\frac{3}{2}}^n \left|u_{i+2}^n - u_{i-1}^n\right|$$
$$+ \left(1 - C_{i+\frac{1}{2}}^n - D_{i+\frac{1}{2}}^n\right) \left|u_{i+1}^n - u_i^n\right|$$

Further summing over index i on both sides gives

$$\Sigma_{i\in K} \left|u_{i+1}^{n+1} - u_i^{n+1}\right| \leq \Sigma_{i\in K} \{C_{i-\frac{1}{2}} \left|u_i^n - u_{i-1}^n\right|\} + \Sigma_{i\in K} \{D_{i+\frac{3}{2}} \left|u_{i+2}^n - u_{i-1}^n\right|\}$$
$$+ \Sigma_{i\in K} \{\left(1 - C_{i+\frac{1}{2}} - D_{i+\frac{1}{2}}\right) \left|u_{i+1}^n - u_i^n\right|\}$$

$$\Rightarrow \Sigma_{i\in K} \left|u_{i+1}^{n+1} - u_i^{n+1}\right| \leq -\Sigma_{i\in K} \{C_{i+\frac{1}{2}}^n \left|u_{i+1}^n - u_i^n\right| - C_{i-\frac{1}{2}}^n \left|u_i^n - u_{i-1}^n\right|\}$$
$$+ \Sigma_{i\in K} \{D_{i+\frac{3}{2}}^n \left|u_{i+2}^n - u_{i-1}^n\right| - D_{i+\frac{1}{2}}^n \left|u_{i+1}^n - u_i^n\right|\} + \Sigma_{i\in K} \{\left|u_{i+1}^n - u_i^n\right|\}$$

It can be seen that the first and second summation terms in right of inequality are zero as i loops over all possible values. The above equation thus reduces to

$$\Sigma_{i\in K} \left|u_{i+1}^{n+1} - u_i^{n+1}\right| \leq \Sigma_{i\in K} \left|u_{i+1}^n - u_i^n\right|$$

$$\Rightarrow TV(n+1) \leq TV(n)$$

3.7.3 MUSCL Scheme on Uniform grid

MUSCL stands for "Monotone Upstream-centered Schemes for Conservation Laws". This is a Finite Volume scheme which involves polynomial reconstruction of left and right values at each interface which used to calculate flux across the interface. Polynomial reconstruction can be done from the already known cell averages in previous iteration. This scheme is second order accurate. The function can be Taylor expanded about the node at x_j in j'th cell.

$$v(x) = v(x_j) + \left(x - x_j\right) \frac{\partial v}{\partial x}\Big|_{x_j} + \frac{1}{2}\left(x - x_j\right)^2 \frac{\partial^2 v}{\partial x^2}\Big|_{x_j} + O(\Delta x^3) \tag{3.14}$$

Integrating the above equation over $\left(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}\right)$ yields

$$\int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(x)dx = v(x_j) \int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} dx + \frac{\partial v}{\partial x}\Big|_{x_j} \int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \left(x - x_j\right) dx + \frac{1}{2}\frac{\partial^2 v}{\partial x^2}\Big|_{x_j} \int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \left(x - x_j\right)^2 dx + O(\Delta x^3)$$

$$v_j \Delta x = v(x_j)\Delta x + \frac{1}{2}\frac{\partial v}{\partial x}\Big|_{x_j} \left(x - x_j\right)^2 \Big|_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} + \frac{1}{6}\frac{\partial^2 v}{\partial x^2}\Big|_{x_j} \left(x - x_j\right)^3 \Big|_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} + O(\Delta x^3)$$

Second term in the RHS drops out because the grid considered is uniform grid and the node is at the centre of the grid.

$$v_j \Delta x = v(x_j)\Delta x + \frac{1}{6}\frac{\partial^2 v}{\partial x^2}\Big|_{x_j} \left(2\left(\frac{\Delta x}{2}\right)^3\right) + O(\Delta x^3)$$

$$v_j \Delta x = v(x_j)\Delta x + \frac{1}{24}\frac{\partial^2 v}{\partial x^2}\Big|_{x_j} (\Delta x)^3 + O(\Delta x^3)$$

$$v(x_j) \approx v_j - \frac{1}{24}\frac{\partial^2 v}{\partial x^2}\Big|_{x_j} (\Delta x)^2$$

Using this value of $v(x_j)$ in (3.14),

$$v(x) = v_j - \frac{1}{24}\frac{\partial^2 v}{\partial x^2}\Big|_{x_j} (\Delta x)^2 + \left(x - x_j\right) \frac{\partial v}{\partial x}\Big|_{x_j} + \frac{1}{2}\left(x - x_j\right)^2 \frac{\partial^2 v}{\partial x^2}\Big|_{x_j} + O(\Delta x^3)$$

$$v(x) = v_j + \left(x - x_j\right) \frac{\partial v}{\partial x}\Big|_{x_j} + \frac{1}{2}\left(\left(x - x_j\right)^2 - \frac{(\Delta x)^2}{12}\right) \frac{\partial^2 v}{\partial x^2}\Big|_{x_j} + O(\Delta x^3)$$

Using central approximation of first order partial derivative,

$$v(x) = v_j + \left(x - x_j\right) \left(\frac{v_{j+1} - v_{j-1}}{2\Delta x}\right) + \frac{1}{2}\left(\left(x - x_j\right)^2 - \frac{(\Delta x)^2}{12}\right) \left(\frac{v_{j+1} + v_{j-1} - 2v_j}{\Delta x^2}\right) + O\left(\Delta x^3\right)$$

On introducing parameter κ in the above equation,

$$v(x) = v_j + \left(x - x_j\right) \left(\frac{v_{j+1} - v_{j-1}}{2\Delta x}\right) + \frac{3\kappa}{2}\left(\left(x - x_j\right)^2 - \frac{(\Delta x)^2}{12}\right) \left(\frac{v_{j+1} + v_{j-1} - 2v_j}{\Delta x^2}\right) + O(\Delta x^3) \tag{3.15}$$

Putting $x = x_{j-\frac{1}{2}}$ in (3.15) results in

$$v^R_{j-\frac{1}{2}} = v_j - \frac{1}{4}\left[(1+\kappa)\delta v_{j-\frac{1}{2}} + (1-\kappa)\delta v_{j+\frac{1}{2}}\right]$$

where $\delta v_{j+\frac{1}{2}} = v_{j+1} - v_j$ and $\delta v_{j-\frac{1}{2}} = v_j - v_{j-1}$.

Putting $x = x_{j-\frac{1}{2}}$ in (3.15) results in

$$v^L_{j+\frac{1}{2}} = v_j + \frac{1}{4}\left[(1-\kappa)\delta v_{j-\frac{1}{2}} + (1+\kappa)\delta v_{j+\frac{1}{2}}\right]$$

3.7.4 TVD Construction

TVD schemes are very useful as they follow very important TVD property. Flux Limiter function can be used in order to put a limit on left and right reconstructed states $v^L_{j+\frac{1}{2}}$ and $v^R_{j+\frac{1}{2}}$ at interface of j'th and $(j+1)'$th cell. Starting with the interfacial states reconstructed from solution in j'th cell,

$$v^R_{j-\frac{1}{2}} = v_j - \frac{1}{4}\left[(1+\kappa)\delta v_{j-\frac{1}{2}} + (1-\kappa)\delta v_{j+\frac{1}{2}}\right]$$

$$v^L_{j+\frac{1}{2}} = v_j + \frac{1}{4}\left[(1-\kappa)\delta v_{j-\frac{1}{2}} + (1+\kappa)\delta v_{j+\frac{1}{2}}\right]$$

The expressions for the left and right states can further be written as

$$v^R_{j-\frac{1}{2}} = v_j - \frac{1}{4}\left[(1+\kappa)\frac{\delta v_{j-\frac{1}{2}}}{\delta v_{j+\frac{1}{2}}}\delta v_{j+\frac{1}{2}} + (1-\kappa)\frac{\delta v_{j+\frac{1}{2}}}{\delta v_{j-\frac{1}{2}}}\delta v_{j-\frac{1}{2}}\right]$$

$$v^L_{j+\frac{1}{2}} = v_j - \frac{1}{4}\left[(1-\kappa)\frac{\delta v_{j-\frac{1}{2}}}{\delta v_{j+\frac{1}{2}}}\delta v_{j+\frac{1}{2}} + (1+\kappa)\frac{\delta v_{j+\frac{1}{2}}}{\delta v_{j-\frac{1}{2}}}\delta v_{j-\frac{1}{2}}\right]$$

Parameter $R_j = \frac{\delta v_{j+\frac{1}{2}}}{\delta v_{j-\frac{1}{2}}}$ is introduced to simplify the notations. Limiter function $\psi(R_j)$ is also introduced at this step. The above two expressions can be written as

$$v^R_{j-\frac{1}{2}} = v_j - \frac{1}{4}\left[(1+\kappa)\,\frac{1}{R_j}\delta v_{j+\frac{1}{2}} + (1-\kappa)\,R_j\delta v_{j-\frac{1}{2}}\right]$$

$$v^L_{j+\frac{1}{2}} = v_j - \frac{1}{4}\left[(1-\kappa)\,\frac{1}{R_j}\delta v_{j+\frac{1}{2}} + (1+\kappa)R_j\delta v_{j-\frac{1}{2}}\right]$$

We further introduce the limiter function $\psi(R_j)$ and replace the occurances R_j in the above expression by $\psi(R_j)$.

$$v^R_{j-\frac{1}{2}} = v_j - \frac{1}{4}\left[(1+\kappa)\,\psi\left(\frac{1}{R_j}\right)\delta v_{j+\frac{1}{2}} + (1-\kappa)\,\psi\left(R_j\right)\delta v_{j-\frac{1}{2}}\right]$$

$$v^L_{j+\frac{1}{2}} = v_j - \frac{1}{4}\left[(1-\kappa)\,\psi\left(\frac{1}{R_j}\right)\delta v_{j+\frac{1}{2}} + (1+\kappa)\,\psi\left(R_j\right)\delta v_{j-\frac{1}{2}}\right]$$

v^{n+1}_j can be derived from v^n_j in terms flux at the left and right faces of the j'th cell.

$$v^{n+1}_j = v^n_j - \lambda\left[g(v^L_{j+\frac{1}{2}}, v^R_{j+\frac{1}{2}}) - g(v^L_{j-\frac{1}{2}}, v^R_{j-\frac{1}{2}})\right]$$

Adding and subtracting an additional term in the above equation yields

$$v^{n+1}_j = v^n_j - \lambda\left[g\left(v^L_{j+\frac{1}{2}}, v^R_{j+\frac{1}{2}}\right) - g\left(v^L_{j-\frac{1}{2}}, v^R_{j+\frac{1}{2}}\right) + g\left(v^L_{j-\frac{1}{2}}, v^R_{j+\frac{1}{2}}\right) - g\left(v^L_{j-\frac{1}{2}}, v^R_{j-\frac{1}{2}}\right)\right] \tag{3.16}$$

By making use of Mean Value Theorem, it is guaranteed to find a point $u_1 \in [v^L_{j-\frac{1}{2}}, v^L_{j+\frac{1}{2}}]$ such that

$$\frac{\partial g}{\partial u}\left(u_1, v^R_{j+\frac{1}{2}}\right) = \frac{g\left(v^L_{j+\frac{1}{2}}, v^R_{j+\frac{1}{2}}\right) - g\left(v^L_{j-\frac{1}{2}}, v^R_{j+\frac{1}{2}}\right)}{v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}}}$$

Similarily, there exist a point $u_2 \in [v^R_{j-\frac{1}{2}}, v^R_{j+\frac{1}{2}}]$ such that

$$\frac{\partial g}{\partial u}\left(v^L_{j-\frac{1}{2}}, u_2\right) = \frac{g\left(v^L_{j-\frac{1}{2}}, v^R_{j+\frac{1}{2}}\right) - g\left(v^L_{j-\frac{1}{2}}, v^R_{j-\frac{1}{2}}\right)}{v^R_{j+\frac{1}{2}} - v^R_{j-\frac{1}{2}}}$$

The difference equation (3.16) can be written as,

$$v_j^{n+1} = v_j^n - \lambda \frac{\partial g}{\partial u}\left(u_1, v^R_{j+\frac{1}{2}}\right)\left(v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}}\right) - \lambda \frac{\partial g}{\partial u}\left(v^L_{j-\frac{1}{2}}, u_2\right)\left(v^R_{j+\frac{1}{2}} - v^R_{j-\frac{1}{2}}\right) \tag{3.17}$$

Consider $v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}}$,

$$\begin{aligned} v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}} = \Bigg(& v_j - \frac{1}{4}\left[(1-\kappa)\,\psi\left(\frac{1}{R_j}\right)\delta v_{j+\frac{1}{2}} + (1+\kappa)\,\psi\left(R_j\right)\delta v_{j-\frac{1}{2}}\right]\Bigg) \\ & - \Bigg(v_{j-1} - \frac{1}{4}\Bigg((1-\kappa)\,\psi\left(\frac{1}{R_{j-1}}\right)\delta v_{j-\frac{1}{2}} + (1+\kappa)\,\psi\left(R_{j-1}\right)\delta v_{j-\frac{3}{2}}\Bigg]\Bigg) \end{aligned}$$

$$\begin{aligned} \Rightarrow v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}} = & \left(v_j - \frac{1}{4}\left[(1-\kappa)\,\psi\left(\frac{1}{R_j}\right)\frac{\delta v_{j+\frac{1}{2}}}{\delta v_{j-\frac{1}{2}}}\delta v_{j-\frac{1}{2}} + (1+\kappa)\,\psi\left(R_j\right)\delta v_{j-\frac{1}{2}}\right]\right) \\ & - \left(v_{j-1} - \frac{1}{4}\left[(1-\kappa))\,\psi\left(\frac{1}{R_{j-1}}\right)\delta v_{j-\frac{1}{2}} + (1+\kappa)\,\psi\left(R_{j-1}\right)\frac{\delta v_{j-\frac{3}{2}}}{\delta v_{j-\frac{1}{2}}}\delta v_{j-\frac{1}{2}}\right]\right) \end{aligned}$$

$$\begin{aligned} \Rightarrow v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}} = & \left(v_j - \frac{1}{4}\left[(1-\kappa)\,\psi\left(\frac{1}{R_j}\right)R_j\delta v_{j-\frac{1}{2}} + (1+\kappa)\,\psi\left(R_j\right)\delta v_{j-\frac{1}{2}}\right]\right) \\ & - \left(v_{j-1} - \frac{1}{4}\left[(1-\kappa)\,\psi\left(\frac{1}{R_{j-1}}\right)\delta v_{j-\frac{1}{2}} + (1+\kappa)\,\psi\left(R_{j-1}\right)\frac{1}{R_{j-1}}\delta v_{j-\frac{1}{2}}\right]\right) \end{aligned}$$

$$\begin{aligned} \Rightarrow v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}} = & \left(v_j - v_{j-1}\right) - \frac{1}{4}\left[(1-\kappa)\,\psi\left(\frac{1}{R_j}\right)R_j\delta v_{j-\frac{1}{2}} + (1+\kappa)\,\psi\left(R_j\right)\delta v_{j-\frac{1}{2}}\right] \\ & + \frac{1}{4}\left[(1-\kappa)\,\psi\left(\frac{1}{R_{j-1}}\right)\delta v_{j-\frac{1}{2}} + (1+\kappa)\,\psi\left(R_{j-1}\right)\frac{1}{R_{j-1}}\delta v_{j-\frac{1}{2}}\right] \end{aligned}$$

$$\begin{aligned} \Rightarrow v^L_{j+\frac{1}{2}} - v^L_{j-\frac{1}{2}} = \Bigg(1 - \frac{(1-\kappa)}{4} & \left[\psi\left(\frac{1}{R_j}\right)R_j - \psi\left(\frac{1}{R_{j-1}}\right)\right] \\ & - \frac{(1+\kappa)}{4}\left[(\psi(R_j) - \psi(R_{j-1})\frac{1}{R_{j-1}}\right]\Bigg)\delta v_{j-\frac{1}{2}} \end{aligned}$$

In the similar way, it can be shown that

$$\begin{aligned} v^R_{j+\frac{1}{2}} - v^R_{j-\frac{1}{2}} = \Bigg(1 - \frac{(1-\kappa)}{4} & \left[\psi\left(R_{j+1}\right) - \frac{\psi\left(R_j\right)}{R_j}\right] \\ & - \frac{(1+\kappa)}{4}\left[(\psi\left(\frac{1}{R_{j+1}}\right)R_{j+1} - \psi\left(\frac{1}{R_j}\right)\right]\Bigg)\delta v_{j-\frac{1}{2}} \end{aligned}$$

Using (3.7.4), (3.7.4) and then plugging it back in (3.17), it can be shown that the scheme could be recast in Harten's incremental form with coefficients satisfying the required criteria under the following conditions.

1. $$0 \le \psi(R) \le \frac{3-\kappa}{1-\kappa} - (1+\alpha)\frac{1+\kappa}{1-\kappa}$$

2. $$0 \le \frac{\psi(R)}{R} \le 2+\alpha$$

where

$$\alpha \in \Big[-2, 2\frac{1-\kappa}{1+\kappa}\Big]$$

and then the scheme can be written in the form (3.13) with the coefficients $C^n_{i-\frac{1}{2}}$ and $D^n_{i+\frac{1}{2}}$ satisfying the condition demanded by Harten's Lemma and thus the scheme is TVD.

3.8 Roe Solver

Roe Solver is based on linear approximation of Riemann problem. The problem of finding solution to

$$U_t + F'(U)U_x$$

where U is n dimensional vector, F is flux function and $F'(U)$ is $n \times n$ Jacobian matrix, is approximated by another problem of solving

$$U_t + AU_x$$

which is set of n linear equations where A is $n \times n$ Jacobian matrix. Here

$$A = A\left(U_l, U_r\right)$$

is a function of left and right states. A satisfies the following conditions.

$$\begin{cases} F(U) - F(V) = A(U,V)(U-V) \\ A(U,V) \to F'(U) \text{ as } V \to U \\ A(U,V) \text{ has only real eigenvalues} \\ A(U,V) \text{ has complete set of eigenvectors} \end{cases}$$

The flux function is given by

$$F^{Roe}\left(U,V\right) = \frac{F\left(U\right)+F\left(V\right)}{2} - \frac{1}{2}\left|A\left(U,V\right)\right|\left(U-V\right) \qquad (3.18)$$

Consider the one dimensional case, Roe flux function is given by

$$F_{i+\frac{1}{2}} = \frac{1}{2}\left(F\left(U_i\right)+F\left(U_{i+1}\right)\right) + \frac{1}{2}\left|\lambda\right|\left(U_{i+1}-U_i\right)$$

where $\lambda = A(U_i, U_{i+1})$. If further, the eigenvalue λ happens to be zero, then flux reduces to

$$F_{i+\frac{1}{2}} = \frac{1}{2}\left(F(U_i) + F(U_{i+1})\right)$$

This does not let the discontinuity insufficiently smear out. Hence there is a need to prevent the second term on right hand side in (3.18) from going to zero.

$$\lambda = \begin{cases} \lambda, & \text{if } \lambda \geq \epsilon \\ \frac{1}{2}\left(\epsilon + \frac{\lambda^2}{\epsilon}\right), & \text{if } \lambda \leq \epsilon \end{cases}$$

for small real $\epsilon > 0$. This correction is also called Entropy Fix since it fixes the entropy-violating solution which is obtained when eigenvalue $\lambda = 0$.

3.9 Important Numerical Flux schemes

Various flux schemes are known with respective advantages and disadvantages. Low resolution schemes like Lax-Wendroff, Lax-Friedrichs do not yield accurate solution mainly in case of nonlinear Partial Differential Equation. Some of the useful schemes are described below with their flux functions:

First Order Upwind scheme

Considering the case of $a > 0$, we can use backward difference, central difference or forward difference to approximate $\frac{\partial u}{\partial x}$ but the one which works out is backward difference. Since $a > 0$, the information travels towards right and backward difference works because it requires function value at the left node in order to calculate the function value at right node, which is in accordance with actual flow of information. In this case, Upwind Scheme is the following

$$\frac{u_i^{n+1} - u_i^n}{dt} + a\left(\frac{u_i^n - u_{i-1}^n}{dx}\right) = 0$$

Considering the case of $a < 0$, we can use backward difference, central difference or forward difference to approximate $\frac{\partial u}{\partial x}$ but the one which works out is forward difference. Since $a < 0$, the information travels towards left and forward difference works because it requires function value at the right node in order to calculate the function value at left node, which is in accordance with actual flow of information. In this case, Upwind Scheme is the following

$$\frac{u_i^{n+1} - u_i^n}{dt} + a\left(\frac{u_{i+1}^n - u_i^n}{dx}\right) = 0$$

Here both can be combined to approximate (2.6) by

$$\begin{cases} \frac{u_i^{n+1}-u_i^n}{dt} + a\left(\frac{u_i^n-u_{i-1}^n}{dx}\right) = 0, & \text{if } a > 0 \\ \frac{u_i^{n+1}-u_i^n}{dt} + a\left(\frac{u_{i+1}^n-u_i^n}{dx}\right) = 0, & \text{if } a < 0 \end{cases} \tag{3.19}$$

Note: Here Forward Euler scheme is used to approximate partial time derivative of function.

Further introducing the quantities a_+ and a_- such that, $a_+ = \max\,(a,0)$ and $a_- = \min\,(a,0)$, then (3.19) can be collectively written as

$$\frac{u_i^{n+1} - u_i^n}{dt} + a_+\left(\frac{u_i^n - u_{i-1}^n}{dx}\right) + a_-\left(\frac{u_{i+1}^n - u_i^n}{dx}\right) = 0$$

$$u_i^{n+1} = u_i^n - \frac{dt}{dx}\left[a_+\left(u_i^n - u_{i-1}^n\right) + a_-\left(u_{i+1}^n - u_i^n\right)\right] = 0 \tag{3.20}$$

Numerical Flux function is given by

$$F(u,v) = \begin{cases} u, & \text{if } a > 0 \\ v, & \text{if } a > 0 \end{cases}$$

The above flux function is for the Advection equation considered for which flux is linear in velocity, $f(u) = au$. The general flux function is the following which reduces to the above expression for Advection equation.

$$F(u,v) = \begin{cases} u, & \text{if } f \text{ is non-decreasing function between } u \text{ and } v \\ v, & \text{if } f \text{ is non-increasing function between } u \text{ and } v \end{cases}$$

Lax-Wendroff scheme

For any function $u\,(x,t)$, second order Taylor approximation gives

$$u(x,t+dt) = u\,(x,t) + \frac{\partial u}{\partial t}dt + \frac{1}{2}\frac{\partial^2 u}{\partial t^2}dt^2 \tag{3.21}$$

Further, for Advection equation, $\frac{\partial u}{\partial t} = a\frac{\partial u}{\partial x}$ and

$$\frac{\partial^2 u}{\partial^2 t} = \frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right) = \frac{\partial}{\partial t}\left(a\frac{\partial u}{\partial x}\right) = a\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right)$$

$$= a\frac{\partial^2 u}{\partial t \partial x} = a\frac{\partial^2 u}{\partial x \partial t} = a\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\right)$$

$$= a\frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right) = a^2\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) = a^2\frac{\partial^2 u}{\partial^2 x}$$

Then (3.21) becomes,

$$u(x, t + dt) = u\,(x, t) + a\frac{\partial u}{\partial x}dt + \frac{1}{2}a^2\frac{\partial^2 u}{\partial^2 x}dt^2$$

Suppose u_i^n is the numerical solution in i'th cell at n'th time step, then the above equation is approximated by

$$u_j^{n+1} - u_j^n = a\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}\Delta t + \frac{1}{2}a^2\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}\Delta t$$

$$\Rightarrow u_j^{n+1} - u_j^n = \frac{1}{2}\left(a\frac{\Delta t}{\Delta x}\right)(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}\left(a^2\frac{\Delta t^2}{\Delta x^2}\right)\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)\Delta t$$

$$\Rightarrow u_j^{n+1} = u_j^n + \frac{1}{2}\left(a\frac{\Delta t}{\Delta x}\right)(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}\left(a^2\frac{\Delta t^2}{\Delta x^2}\right)\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)\Delta t$$

The quantity $a\frac{\Delta t}{\Delta x}$ denoted by ν is called CFL number. The above equation reduces to

$$u_j^{n+1} = u_j^n + \frac{1}{2}\nu(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}\nu^2\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)\Delta t$$

Flux function is given by

$$F_{j+\frac{1}{2}} = \frac{f_j + f_{j+1}}{2} - \frac{\nu_{j+\frac{1}{2}}}{2}\left(f_{j+1} - f_j\right)$$

where

$$v_{j+\frac{1}{2}} = \frac{a_{j+\frac{1}{2}}\left(f_{j+1} - f_j\right) - a_{j-\frac{1}{2}}\left(f_j - f_{j-1}\right)}{h^2}$$

and $a_{j+\frac{1}{2}}$ is given by

$$a_{j+\frac{1}{2}} = \begin{cases} \frac{1}{2}\left[a\left(u_j\right) + a\left(u_{j+1}\right)\right], & \text{if } a(u_j) \neq a(u_{j+1}) \\ a(u_j), & \text{if } a(u_j) = a(u_{j+1}) \end{cases}$$

Flux function is given by

$$F_{j+\frac{1}{2}} = \frac{f_j + f_{j+1}}{2} - \frac{\nu_{j+\frac{1}{2}}}{2}\left(f_{j+1} - f_j\right)$$

where

$$v_{j+\frac{1}{2}} = \frac{a_{j+\frac{1}{2}}\left(f_{j+1} - f_j\right) - a_{j-\frac{1}{2}}\left(f_j - f_{j-1}\right)}{h^2}$$

and $a_{j+\frac{1}{2}}$ is given by

$$a_{j+\frac{1}{2}} = \begin{cases} \frac{1}{2}\left[a\left(u_j\right) + a\left(u_{j+1}\right)\right], & \text{if } a(u_j) \neq a(u_{j+1}) \\ a(u_j), & \text{if } a(u_j) = a(u_{j+1}) \end{cases}$$

Godunov Scheme

Consider the following problem with piecewise constant initial data.

$$\begin{cases} w_t + (f(w))_x = 0 \ , \quad t \in (t_n, t_{n+1}] \\ w(x, t_n) = u(x, t_n) \end{cases} \tag{3.22}$$

where $u(x, t_n)$ is also denoted by u_i^n if $x \in \left(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right)$. The problem could be decomposed into many local Riemann problems centered at interfaces $x_{i+\frac{1}{2}}$.

$$\begin{cases} w_t + (f(w))_x = 0 \ , \quad t \in (t_n, t_{n+1}] \\ w(x, t_n) = \begin{cases} u_i^n, & \text{if } x < x_{i+\frac{1}{2}} \\ u_{i+1}^n, & \text{if } x > x_{i+\frac{1}{2}} \end{cases} \end{cases} \tag{3.23}$$

If we further take the CFL number $N_{CFL} \leq \frac{1}{2}$, then the waves from $x_{i+\frac{1}{2}}$ cannot reach the line $x = x_i$ and $x = x_{i+1}$ in the time used for evolution. Hence, problem (3.22) can be interpreted as superposition of local Riemann problems if $N_{CFL} \leq \frac{1}{2}$.

The solution of (3.22) in $(x_i, x_{i+1}) \times (t_n, t_{n+1}]$ is given by

$$w(x,t) = w_R\left(\left(x - x_{i+\frac{1}{2}}\right) / \left(t - t_n\right), u_i^n, u_{i+1}^n\right)$$

$$\Rightarrow w(x, t_{n+1}) = w_R\left(\left(x - x_{i+\frac{1}{2}}\right) / \Delta t, u_i^n, u_{i+1}^n\right)$$

where $\Delta t = t_{n+1} - t_n$. Average value of $w(x, t_{n+1})$ in $\left(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right)$, denoted by u_i^{n+1}, in terms of grid width $h_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ is given by

$$u_i^{n+1} = \frac{1}{h_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} w(x, t_{n+1})$$

Integrating (3.22) over space-time interval $\left(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right) \times (t_n, t_{n+1}]$,

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{t_n}^{t_{n+1}} \left(\frac{\partial w(x,t)}{\partial t}\right) dt dx = - \int_{t_n}^{t_{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\frac{\partial f(w(x,t))}{\partial x}\right) dx dt$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\int_{t_n}^{t_{n+1}} \frac{\partial w(x,t)}{\partial t} dt\right) dx = - \int_{t_n}^{t_{n+1}} \left(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial f(w(x,t))}{\partial x} dx\right) dt$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(w(x,t) \Big|_{t_n}^{t_{n+1}}\right) dx = - \int_{t_n}^{t_{n+1}} \left(f(w(x,t)) \Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\right) dt$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[w\left(x,t_{n+1}\right)-w\left(x,t_n\right)\right]dx = -\int_{t_n}^{t_{n+1}} \left[f\left(w\left(x_{i+\frac{1}{2}},t\right)\right)-f\left(w\left(x_{i-\frac{1}{2}},t\right)\right)\right]dt$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} w\left(x,t_{n+1}\right)dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} w\left(x,t_n\right)dx = -\int_{t_n}^{t_{n+1}} \left[f\left(w\left(x_{i+\frac{1}{2}},t\right)\right)-f\left(w\left(x_{i-\frac{1}{2}},t\right)\right)\right]dt$$

$$h_i u_i^{n+1} = h_i u_i^n - \int_{t_n}^{t_{n+1}} \left[f\left(w\left(x_{i+\frac{1}{2}},t\right)\right)-f\left(w\left(x_{i-\frac{1}{2}},t\right)\right)\right]dt$$

$$h_i u_i^{n+1} = h_i u_i^n - \int_{t_n}^{t_{n+1}} \left[f\left(w_R\left(0,u_i^n,u_{i+1}^n\right)\right)-f\left(w_R\left(0,u_{i-1}^n,u_i^n\right)\right)\right]dt$$

$$h_i u_i^{n+1} = h_i u_i^n - \Delta t\left[f\left(w_R\left(0,u_i^n,u_{i+1}^n\right)\right)-f\left(w_R\left(0,u_{i-1}^n,u_i^n\right)\right)\right]$$

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{h_i}\left[f\left(w_R\left(0,u_i^n,u_{i+1}^n\right)\right)-f\left(w_R\left(0,u_{i-1}^n,u_i^n\right)\right)\right]$$

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{h_i}\left[F_{i+\frac{1}{2}}-F_{i-\frac{1}{2}}\right]$$

where $F_{i+\frac{1}{2}} = F(u^L_{i+\frac{1}{2}}, u^R_{i+\frac{1}{2}})$ where the numerical flux function given by

$$F(u_i^n,u_{i+1}^n) = f\left(u_R\left(0,u_i^n,u_{i+1}^n\right)\right)$$

A simple formula which is used for calculation purposes is

$$F(u,v) = \begin{cases} \min_{w\in[u,v]} f(w), & \text{if } u \le v \\ \max_{w\in[u,v]} f(w), & \text{if } u > v \end{cases}$$

In terms of left and right reconstructed states $u_{i-\frac{1}{2}}$ and $u_{i+\frac{1}{2}}$, Godunov flux is calculated as

$$F_{i+\frac{1}{2}} = F\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}\right) = \max\left\{f\left(\max\left(0,u^-_{i+\frac{1}{2}}\right),0\right), f\left(\min\left(0,u^+_{i+\frac{1}{2}}\right),0\right)\right\}$$

Roe Scheme

The problem of finding solution to (3.22) could be decomposed into many local Riemann problems centered at interfaces $x_{i+\frac{1}{2}}$. The following linear problem is considered in this case.

$$\begin{cases} w_t + a(u_i^n, u_{i+1}^n)u_x = 0 \ , \quad t \in (t_n, t_{n+1}] \\ w(x, t_n) = \begin{cases} u_i^n, & \text{if } x < x_{i+\frac{1}{2}} \\ u_{i+1}^n, & \text{if } x > x_{i+\frac{1}{2}} \end{cases} \end{cases} \tag{3.24}$$

where

$$a(u,v) = \begin{cases} \frac{f(u)-f(v)}{u-v}, & \text{if } u \neq v \\ f'(u), & \text{if } u = v \end{cases}$$

Solution of (3.24) is given by

$$w(x,t) = \begin{cases} u_i^n, & \text{if } x - x_{i+\frac{1}{2}} < a(u_i^n, u_{i+1}^n)t \\ u_{i+1}^n, & \text{if } x - x_{i+\frac{1}{2}} > a(u_i^n, u_{i+1}^n)t \end{cases}$$

Scheme can be written in the form

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{h_i}\left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right)$$

where $F_{i+\frac{1}{2}} = F\left(x^L_{i+\frac{1}{2}}, x^R_{i+\frac{1}{2}}\right)$ with flux function F given by

$$F\left(u,v\right) = \frac{f(u)+f(v)}{2} - \frac{1}{2}\left|a(u,v)\right|(u-v)$$

Local Lax-Frederichs Scheme/ Rusanov Scheme

$$F(u,v) = \frac{1}{2}\left(f(u)+f(v)\right) - \frac{1}{2}\lambda\left(v-u\right)$$

where

$$\lambda = max(\left|f'(u)\right|, \left|f'(v)\right|)$$

Lax-Frederichs Scheme

$$F(u,v) = \frac{1}{2}\left(f(u)+f(v)\right) - \frac{1}{2}\frac{\Delta x}{\Delta t}\left(v-u\right)$$

where Δx is grid width and Δt is time-step

Engquist Osher Scheme

$$F(u,v) = \begin{cases} f(u), & \text{if } a \geq 0 \\ f(v), & \text{if } a < 0 \end{cases}$$

$$a = \begin{cases} \frac{f(u)-f(v)}{u-v}, & \text{if } u \neq v \\ f'(u), & \text{if } u = v \end{cases}$$

3.10 Periodic Boundary Problems

If we consider $N + 1$ faces and N cells, and suppose for the initial domain $I_0 = [x_{min}, x_{max}]$ considered , x_0^L denotes the position of leftmost face and x_N^R denotes the position of rightmost face. The interval can be extended in the following manner ,

$$I_i = [x_{min} + i\left(x_{max} - x_{min}\right), x_{max} + i\left(x_{max} - x_{min}\right)]$$

for any $i \in \mathbb{Z}$. Then

$$\left(x_0^L\right)_{x \in I_{i+1}} = \left(x_N^R\right)_{x \in I_i}$$

and

$$\left(x_N^R\right)_{x \in I_{i-1}} = \left(x_0^L\right)_{x \in I_i}$$

for integers $i \in \mathbb{Z}$.

Updating solution at any node i requires fluxes from the left and right faces i.e. $F_{i+\frac{1}{2}}$ and $F_{i-\frac{1}{2}}$.

$$u_i^{n+1} = u_i^{n+1}\left(u_i^n; x_{i+\frac{1}{2}}, x_{i-\frac{1}{2}}; F_{i+\frac{1}{2}}, F_{i-\frac{1}{2}}\right) \tag{3.25}$$

For the second order schemes, the flux through right face i.e. $F_{i+\frac{1}{2}}$ depends on function values at nodes $i-1, i, i+1, i+2$ and their locations.

$$F_{i+\frac{1}{2}} = F_{i+\frac{1}{2}}\left(x_{i-1}, x_i, x_{i+1}, x_{i+2}; u_{i-1}, u_i, u_{i+1}, u_{i+2}\right) \tag{3.26}$$

where x_i is the location of i'th node and u_i is the solution at i'th node at some time step.

Similarily

$$F_{i-\frac{1}{2}} = F_{i-\frac{1}{2}}\left(x_{i-2}, x_{i-1}, x_i, x_{i+1}; u_{i-2}, u_{i-1}, u_i, u_{i+1}\right) \tag{3.27}$$

Thus, from (3.25), (3.25) and (3.25)

$$u_i^{n+1} = u_i^{n+1}\left(x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{i+\frac{1}{2}}, x_{i-\frac{1}{2}}; u_{i-2}, u_{i-1}, u_i, u_{i+1}, u_{i+2}\right) \tag{3.28}$$

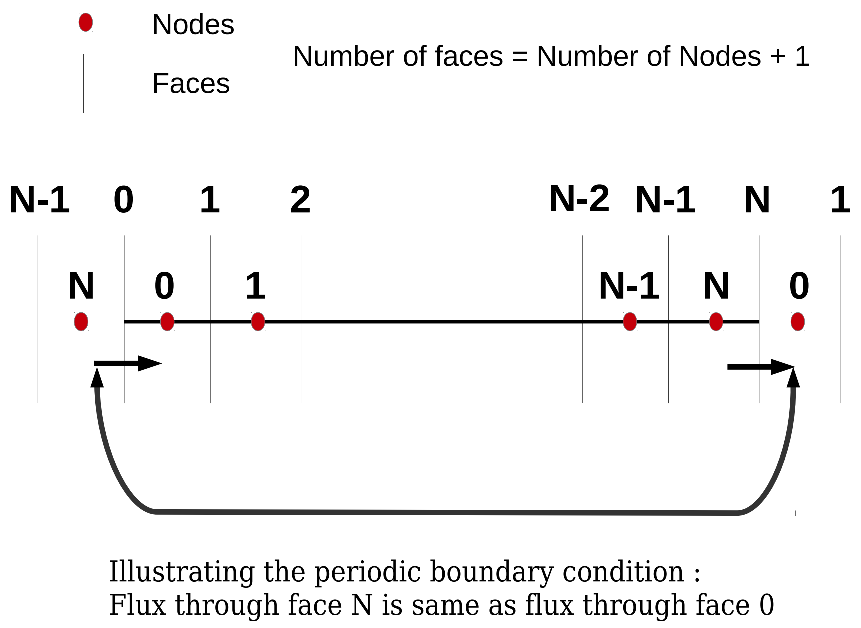

Nodes
Faces
Number of faces = Number of Nodes + 1
N-1
0
1
2
N-2
N-1
N
1
N
0
1
N-1
N
0
Illustrating the periodic boundary condition :
Flux through face N is same as flux through face 0

Figure 3.3: *This figure shows the treatment done at the boundary while taking into account the periodic boundary conditions.*

It has already been proved in Section 3.2.3 that the following first order derivative is second order accurate

$$\frac{\partial u_j}{\partial x} = \frac{\frac{1}{R_j}\left(u_{j+1} - u_j\right) + R_j\left(u_j - u_{j-1}\right)}{x_{j+1} - x_{j-1}}$$

For j'th node located at x_j, consider (j-1)'th, j'th and (j+1)'th cells. We define here left difference and right difference in any cell.

Right difference : $d_j^r = x_{j+\frac{1}{2}} - x_j$

Left difference : $d_j^l = x_j - x_{j-\frac{1}{2}}$

Left and right states at the $\left(i + \frac{1}{2}\right)'$th face can be constructed using first order derivative,

Left state :

$$u_{j+\frac{1}{2}}^L = u_j + d_j^r\left(\frac{\partial u_j}{\partial x}\right)$$

$$u_{j+\frac{1}{2}}^L = u_j + \left(x_{j+\frac{1}{2}} - x_j\right)\left(\frac{\frac{1}{R_j}\left(u_{j+1} - u_j\right) + R_j\left(u_j - u_{j-1}\right)}{x_{j+1} - x_{j-1}}\right)$$

Right state :

$$u_{j+\frac{1}{2}}^R = u_{j+1} - d_{j+1}^l\left(\frac{\partial u_{j+1}}{\partial x}\right)$$

$$u_{j+\frac{1}{2}}^R = u_{j+1} - \left(x_{j+1} - x_{j+\frac{1}{2}}\right)\left(\frac{\frac{1}{R_{j+1}}\left(u_{j+2} - u_{j+1}\right) + R_{j+1}\left(u_{j+1} - u_j\right)}{x_{j+2} - x_j}\right)$$

To take periodicity into account, the initial domain can be extended in a manner such that the last face of i'th interval coincides with the first face of $(i+1)$'th interval and the final face of $(i+1)$'th interval coincides with the last face of i'th interval. In our convention, N nodes are indexed by integers from 0 to N-1. In our convention, x_i denotes the position of nodes for $i \in \{0, 1, ..., N-2, N-1\}$ and it denotes position of faces for $i \in \{-\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, ..., N-\frac{3}{2}, N-\frac{1}{2}\}$. We need to know x_i and u_i for $i = -1$ and $i = -2$ while evaluating solution at the left end. We further need to know x_i and u_i for $i = N$ and $i = (N+1)$ while updating solution at the right boundary. We can impose the periodicity while calculating the above two quantities in the following way. We can set $i = \mod\left(i_{\in\{-2,-1,0,1,....,N-1,N,N+1\}}, N\right)$, which results in $i \in \{0, 1, ..., N-1\}$, for calculating u_i's. Further the extension is done in the way described below.

Since we know x_i for $i \in \{-2, -1, 0, 1, ..., N-1\}$,we can use the following to deal with boundary indices.

$$x_i = \begin{cases} x_{N-1}, & \text{if } i = -1 \\ x_{N-2}, & \text{if } i = -2 \\ x_1, & \text{if } i = N+1 \end{cases} \tag{3.29}$$

$$x_{i+\frac{1}{2}} = \begin{cases} x_{N-\frac{3}{2}}, & \text{if } i = -2 \\ x_{N-\frac{1}{2}}, & \text{if } i = -1 \\ x_{\frac{1}{2}}, & \text{if } i = N \\ x_{\frac{3}{2}}, & \text{if } i = N+1 \end{cases} \tag{3.30}$$

In this convention,

$$u^R_{-\frac{1}{2}} = u_0 - \left(x_N - x_{N-\frac{1}{2}}\right)\left(\frac{\frac{1}{R_0}(u_1 - u_0) + R_0\left(u_0 - u_{n_{face}-2}\right)}{(x_1 - x_0) + (x_N - x_{N-1})}\right)$$

$$u^L_{-\frac{1}{2}} = u_{N-1} + \left(x_{N-\frac{1}{2}} - x_{N-1}\right)\left(\frac{\frac{1}{R_{N-1}}(u_0 - u_{N-1}) + R_{N-1}(u_{N-1} - u_{N-3})}{x_{N-1} - x_{N-3}}\right)$$

Flux at the leftmost face,

$$F_{-\frac{1}{2}} = F_{-\frac{1}{2}}\left(u^L_{-\frac{1}{2}}, u^R_{-\frac{1}{2}}\right)$$

$$u^R_{-\frac{3}{2}} = u_{N-1} - \left(x_{N-1} - x_{N-\frac{3}{2}}\right)\left(\frac{\frac{1}{R_{N-1}}(u_0 - u_{N-1}) + R_{N-1}(u_{N-1} - u_{N-2})}{x_{N-1} - x_{N-3}}\right)$$

$$u^L_{-\frac{3}{2}} = u_{N-2} + \left(x_{N-\frac{3}{2}} - x_{N-2}\right)\left(\frac{\frac{1}{R_{N-2}}\left(u_{N-1} - u_{N-2}\right) + R_{N-2}\left(u_{N-2} - u_{N-3}\right)}{x_{N-2} - x_{N-4}}\right)$$

and then

$$F_{-\frac{3}{2}} = F_{-\frac{3}{2}}\left(u^L_{-\frac{3}{2}}, u^R_{-\frac{3}{2}}\right)$$

$$u^L_{\frac{1}{2}} = u_0 + \left(x_1 - x_0\right)\left(\frac{\frac{1}{R_0}\left(u_1 - u_0\right) + R_0\left(u_0 - u_{n_{face}-2}\right)}{\left(x_1 - x_0\right) + \left(x_{N-1} - x_{N-2}\right)}\right)$$

$$u^R_{\frac{1}{2}} = u_1 - \left(x_1 - x_{\frac{1}{2}}\right)\left(\frac{\frac{1}{R_1}\left(u_2 - u_1\right) + R_1\left(u_1 - u_0\right)}{x_2 - x_0}\right)$$

and then

$$F_{\frac{1}{2}} = F_{\frac{1}{2}}\left(u^L_{\frac{1}{2}}, u^R_{\frac{1}{2}}\right)$$

By using these recontructed states, we can first calculate the flux across the face and this can be made use of to finally calculate u_j^{n+1} from u_j^n.

3.11 Convergence Analysis

The error considered is

$$e_{p,h,\alpha} = \left(\sum_j |v_j - u_j|^p \Delta x_j\right)^{\frac{1}{p}}$$

where

$$v_j = \frac{1}{\Delta x_j}\int_{C_j} u\left(x,t\right)dx$$

where C_j denotes j'th cell. u_j is the numerical solution at j'th cell

3.12 Numerical Results

3.12.1 Comparison of Various Flux Schemes

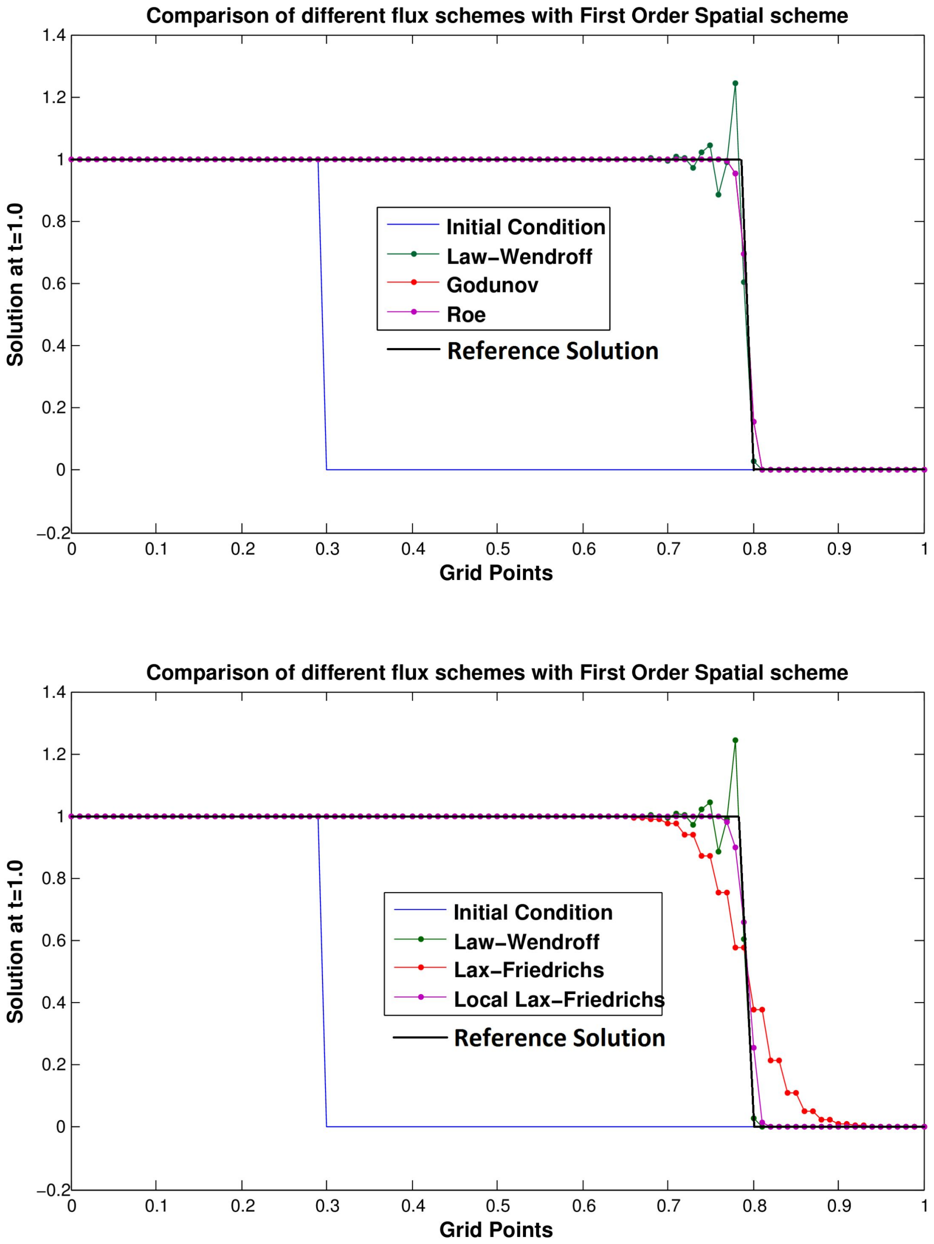


Figure 3.4: Riemann Problem considered, Shock wave is generated which moves to right

Lax-Wendroff scheme produces lot of oscillations while Godunov and Roe schemes yield solutions which apparently coincide with each other.

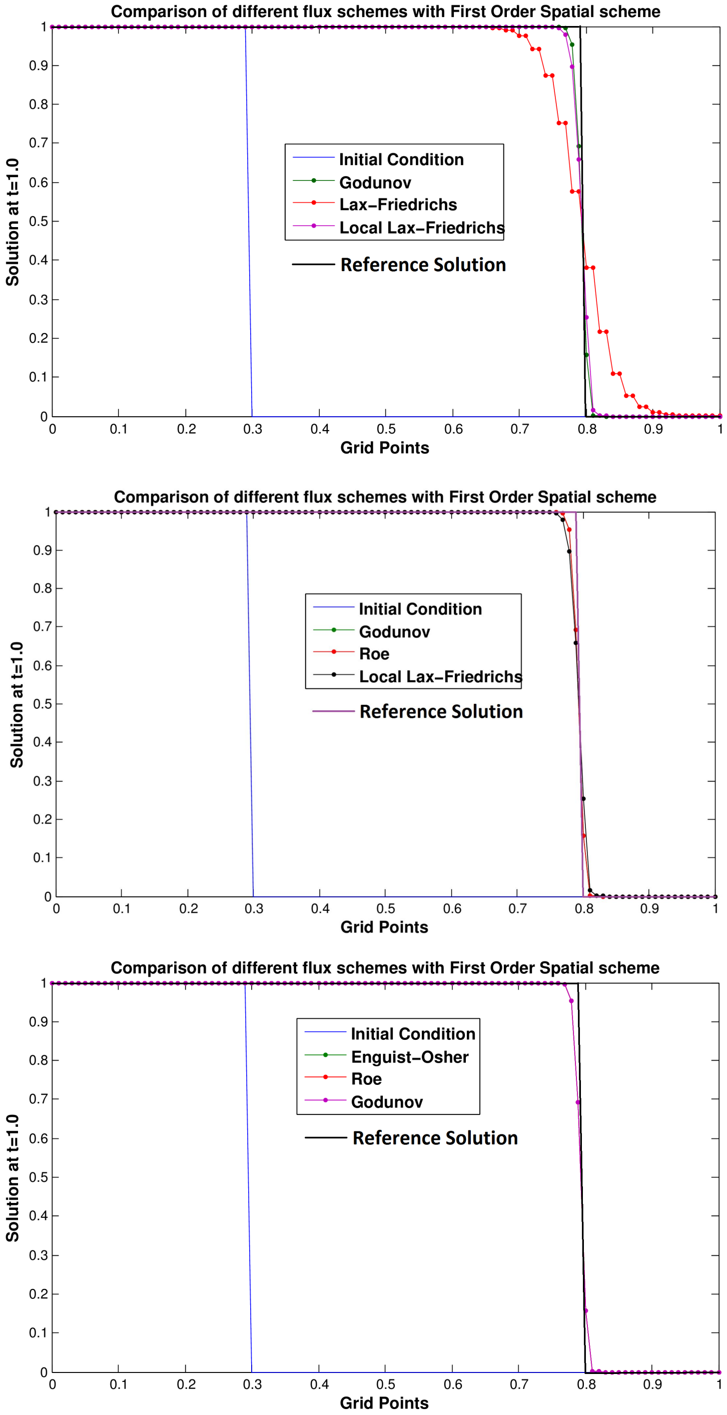
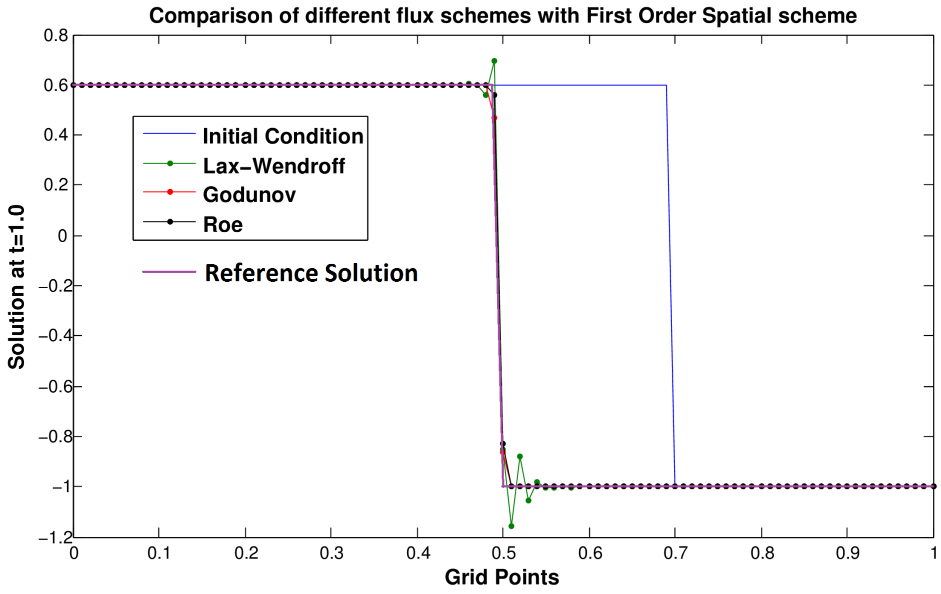


Figure 3.5: Riemann Problem considered, Shock wave is generated which moves to right

Lax-Friedrichs scheme yields solution which deviates a lot from the reference solution. Local Lax-Friedrichs does better than Lax-Friedrichs. Roe and Godunov schemes do slightly better than Local lax-Friedrichs.



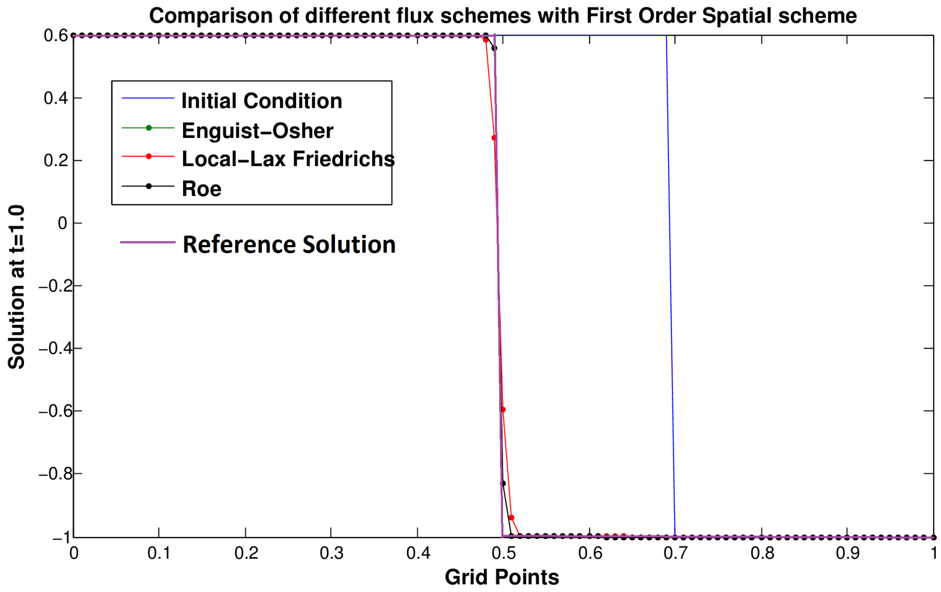


Figure 3.6: Riemann Problem considered, Shock wave is generated which moves to left

Solutions corresponding to Roe and Enguist-Osher nearly coincide. Lax-Wendroff produces oscillations near the shock.

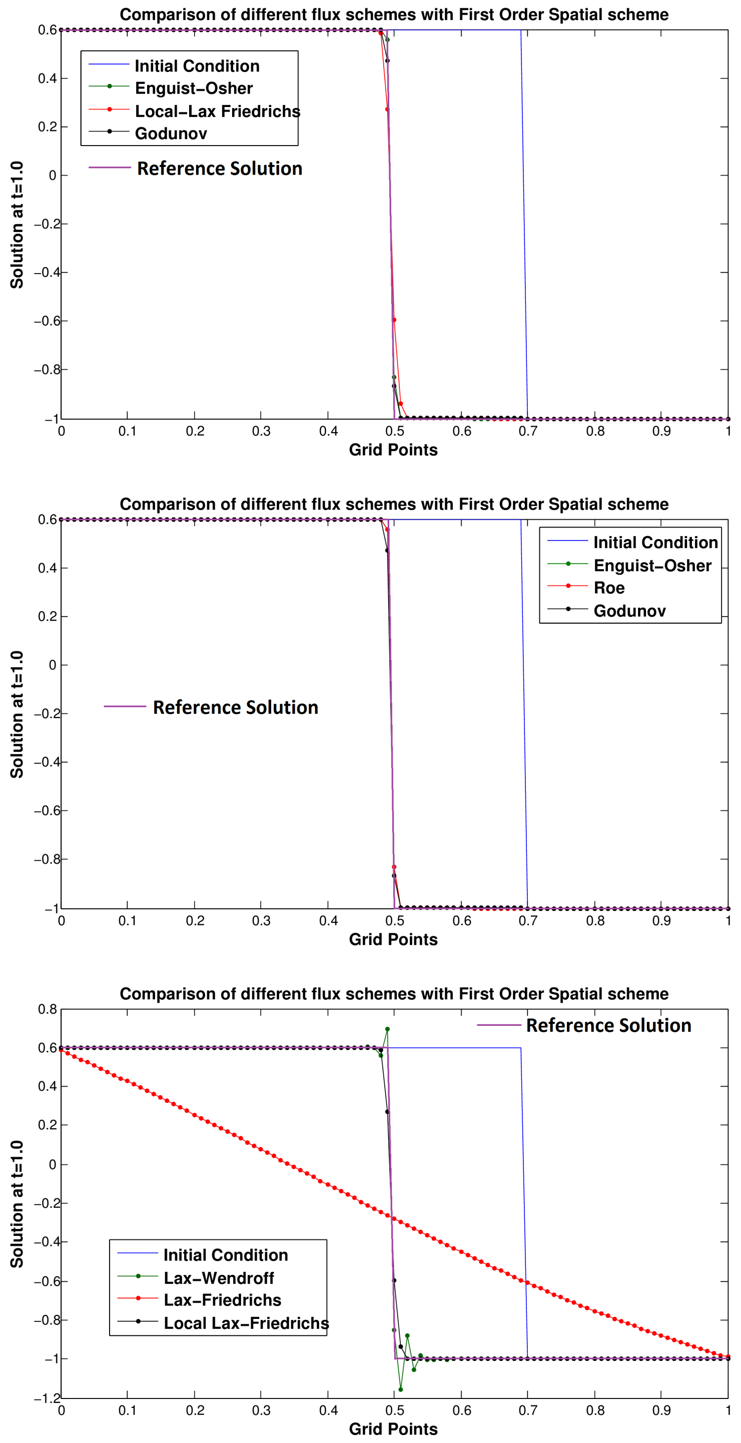
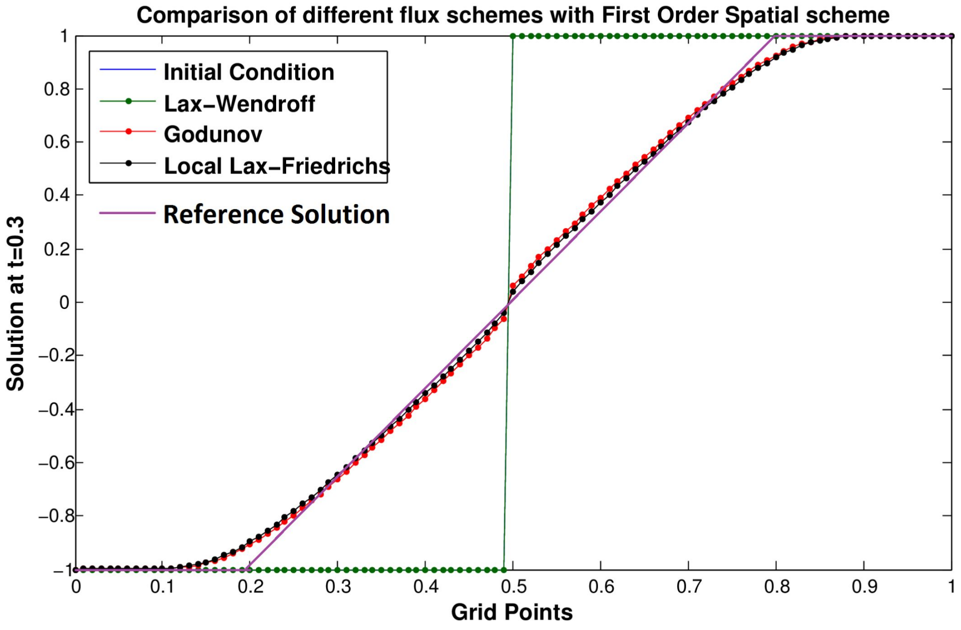


Figure 3.7: Riemann Problem considered, Shock wave is generated which moves to left

Solution corresponding to Lax-Friedrichs is very different from the reference solution. Enguist-Osher, Godunov, Roe and Local-Lax Friedrichs all result in solution close to reference solution but the former three do slightly better than Local Lax-Friedrichs.



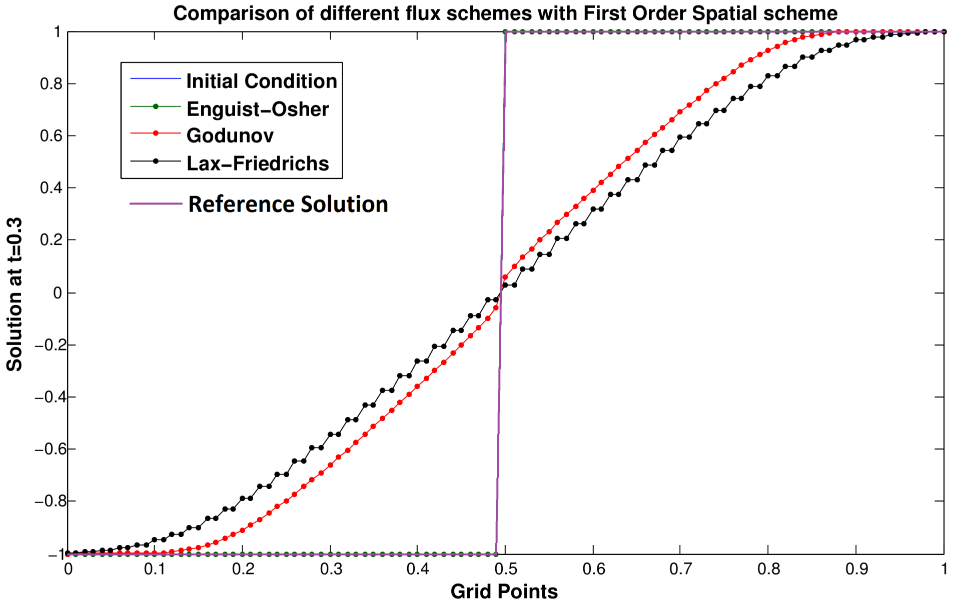


Figure 3.8: Riemann Problem considered, Rarefaction wave is generated

Godunov and Local-Lax Friedrichs yield solution close to reference solution and Godunov does better than Local-Lax Friedrichs. Lax-Wendroff and Enguist-Osher schemes do not produce the correct solution.

3.12.2 Comparison of First and Second Order Schemes based on solutions to Burgers Equation

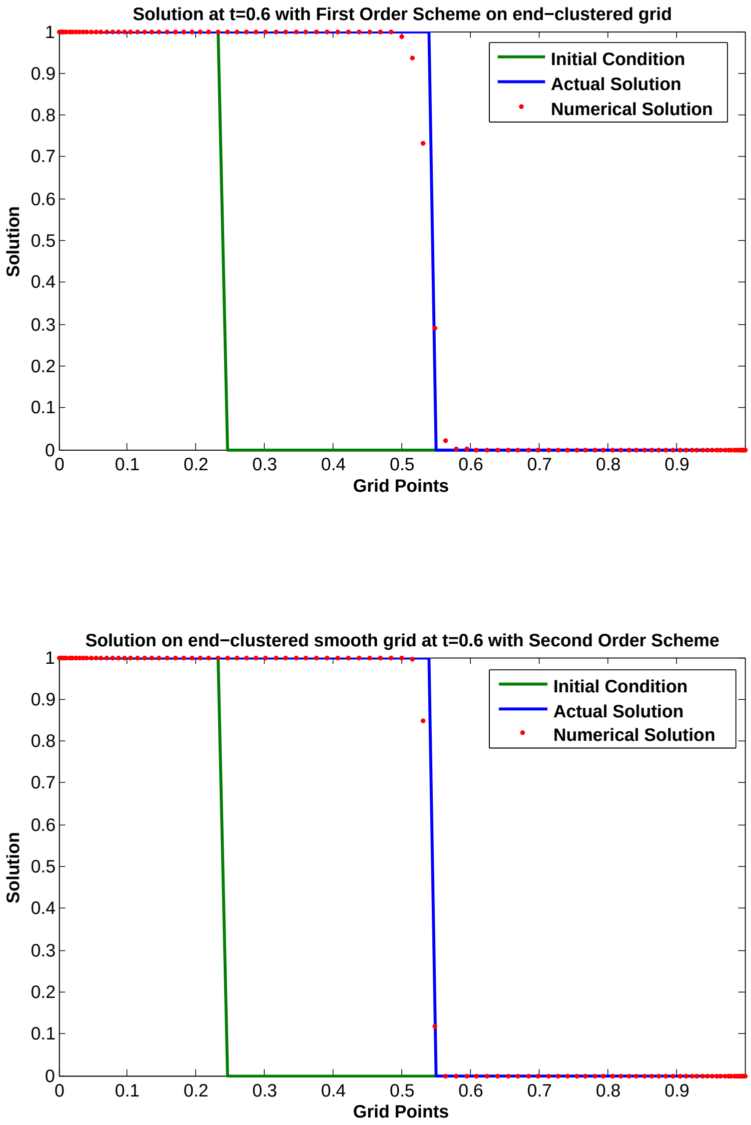


Figure 3.9: Riemann Problem considered, Shock wave is generated

The shock moves towards right with velocity $\frac{1}{2}$. The number of red dots in the shock region indicate the accuracy of the scheme. Second order scheme produces fewer red dots as compared to first order scheme in both grids.

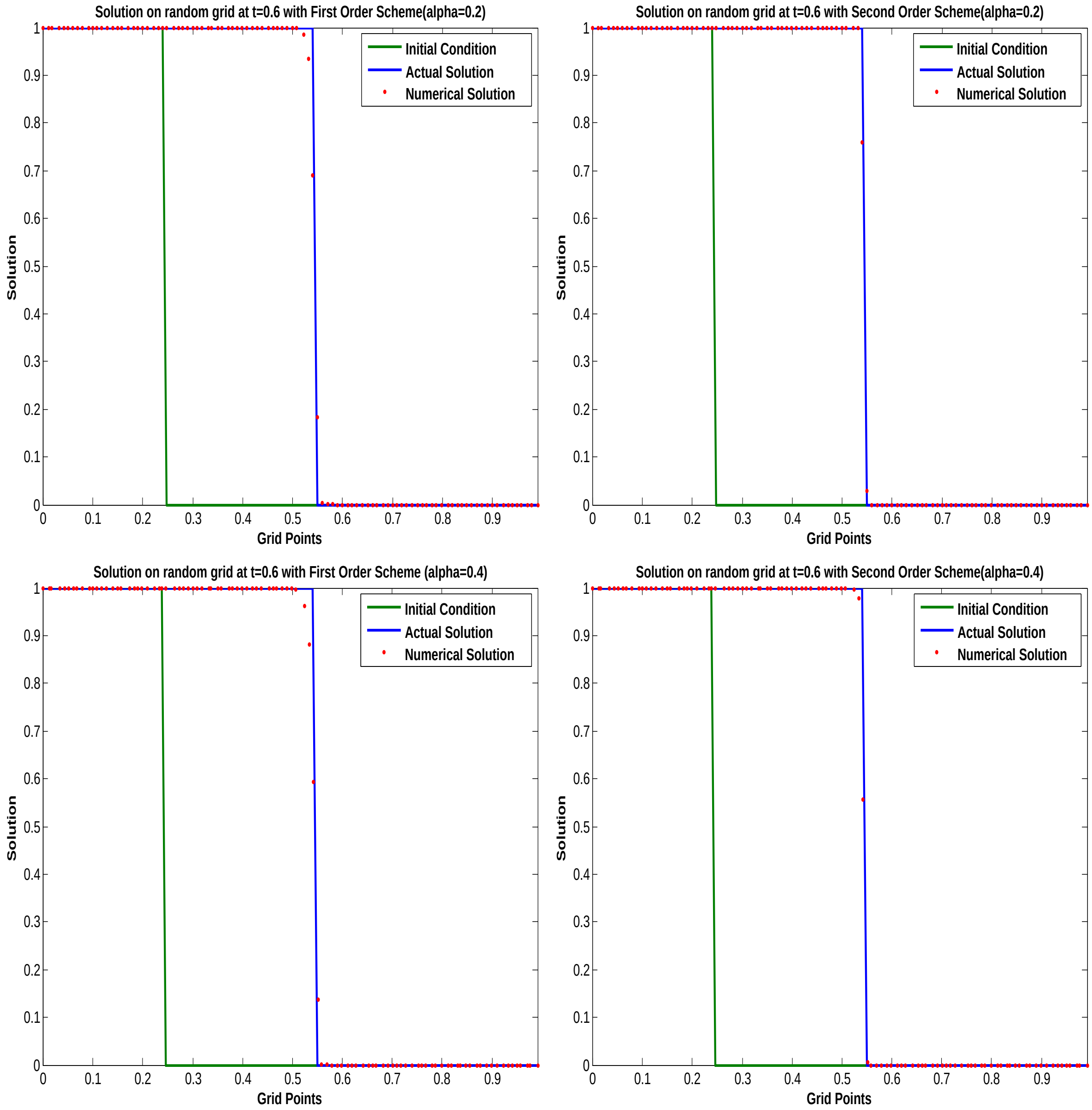


Figure 3.10: Riemann Problem with u_l=1 and u_r=-1, on different grids

The results are on the random grid. Two different random grids used have different randomness with degree of randomness being 0.2 and 0.4 for the both of them. The shock moves towards right with velocity $\frac{1}{2}$. The number of red dots in the shock region indicate the accuracy of the scheme. Second order scheme produces fewer red dots as compared to first order scheme.

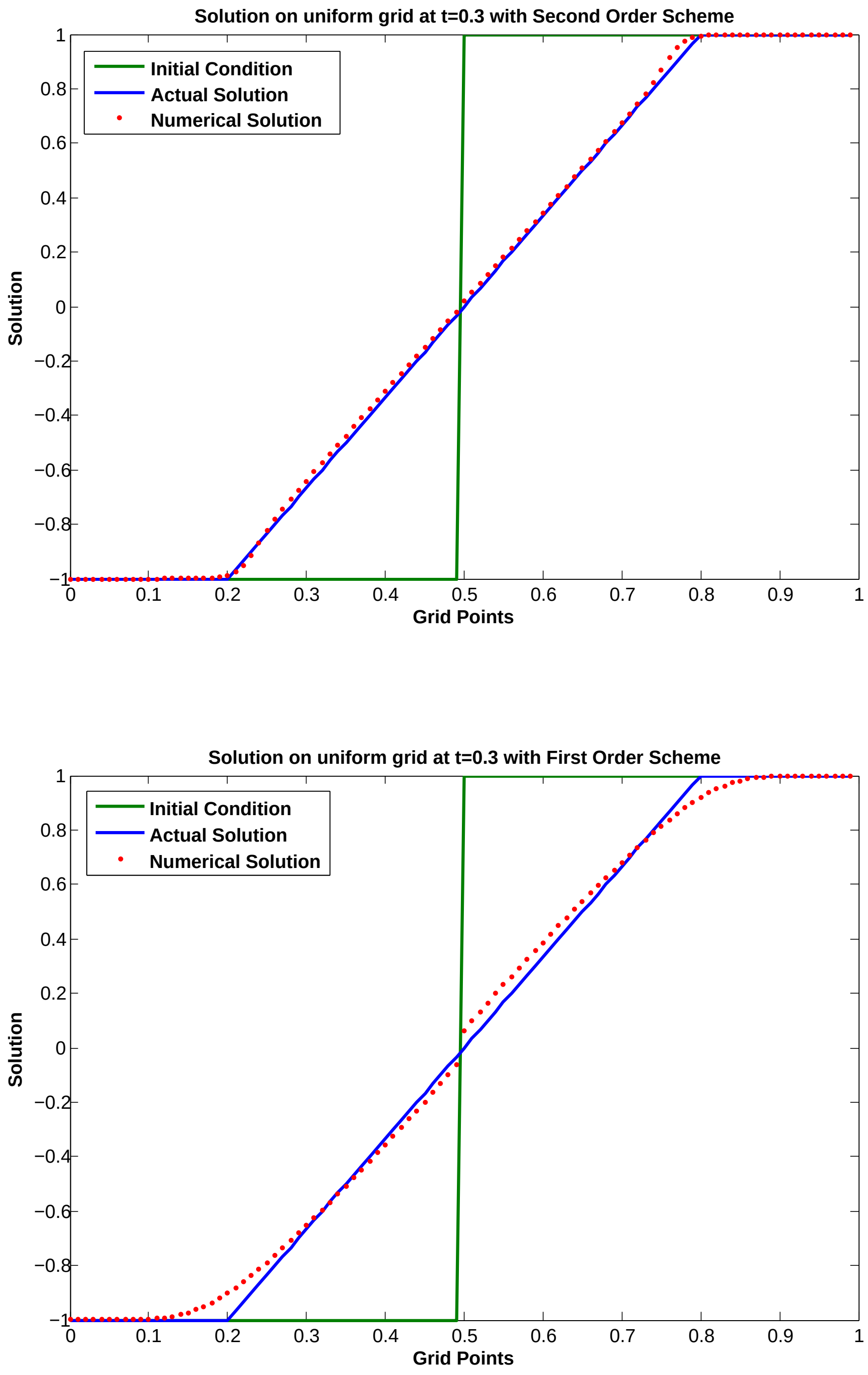


Figure 3.11: Riemann problem considered results in rarefaction waves

The solution with second order scheme is closer to the reference solution as compared to the first order scheme.

3.12.3 Comparison of Cell-centered and Vertex-centered approach for different values of parameter beta considering solutions to Burgers Equation

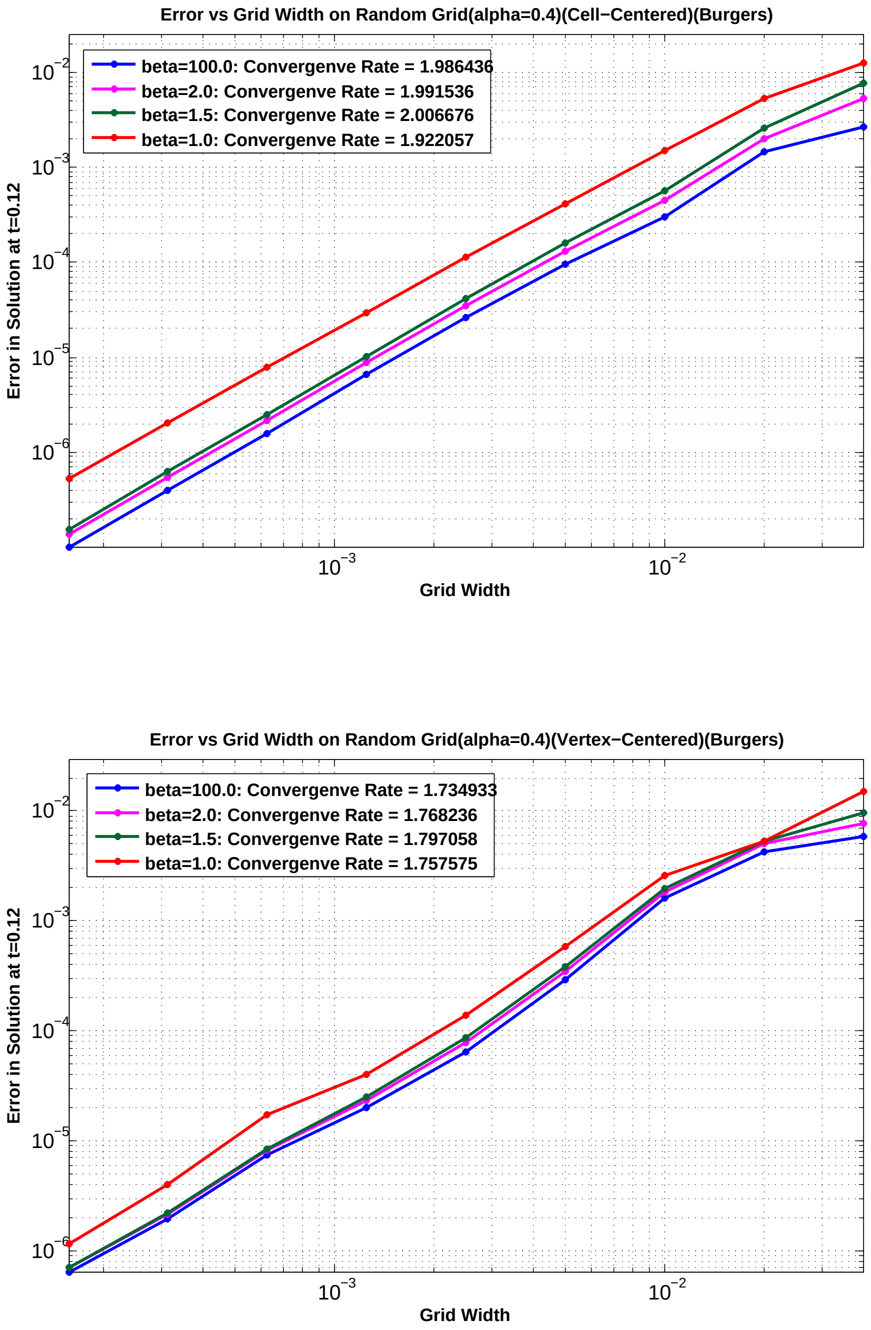
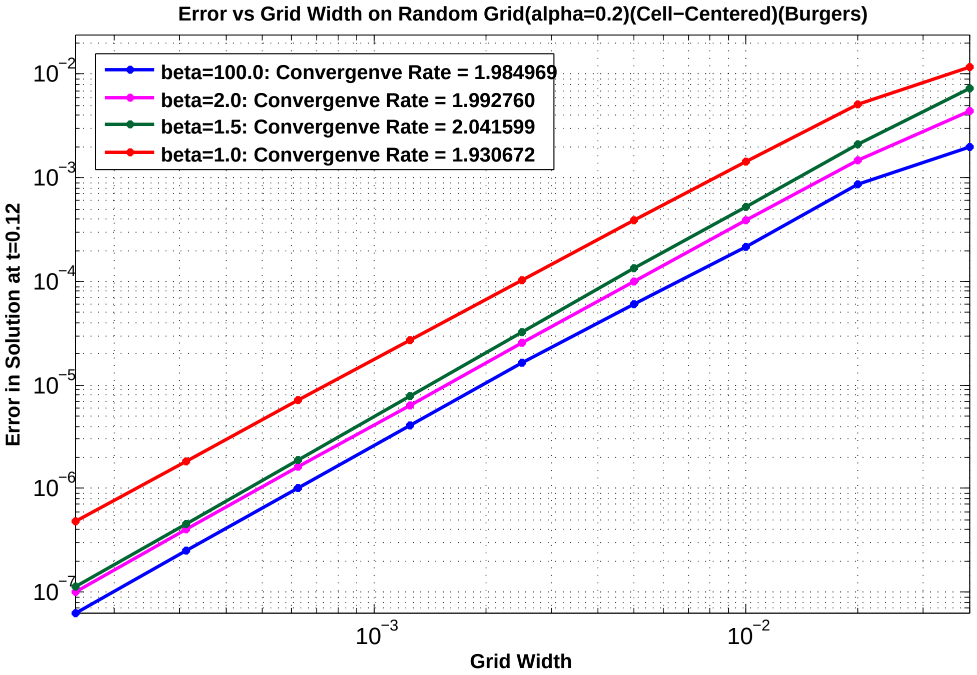


Figure 3.12: Convergence Analysis on Random Grid: Convergence is slow for beta = 1



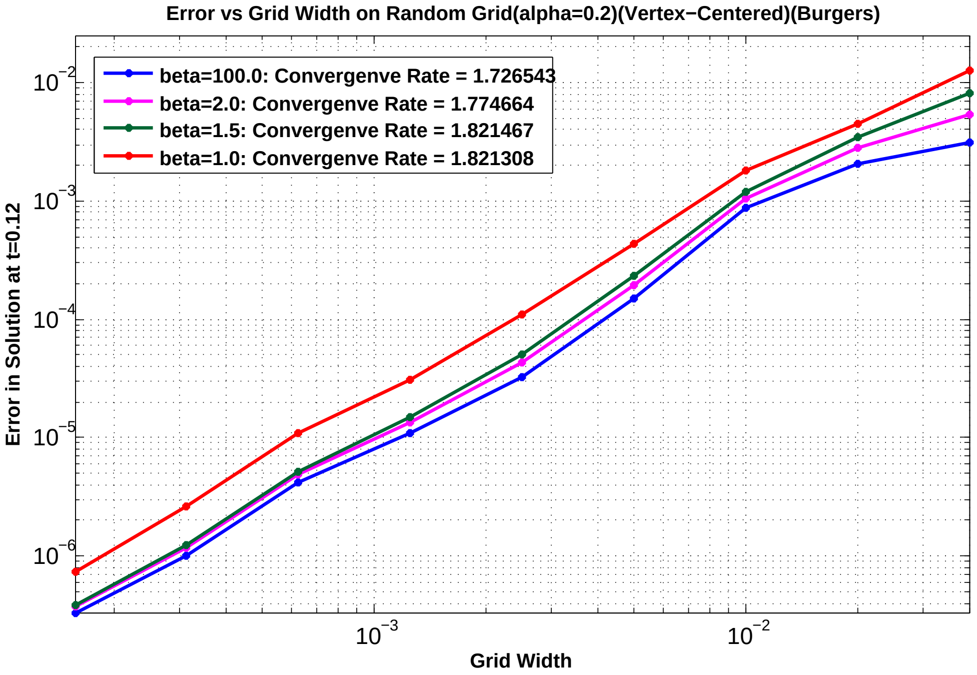
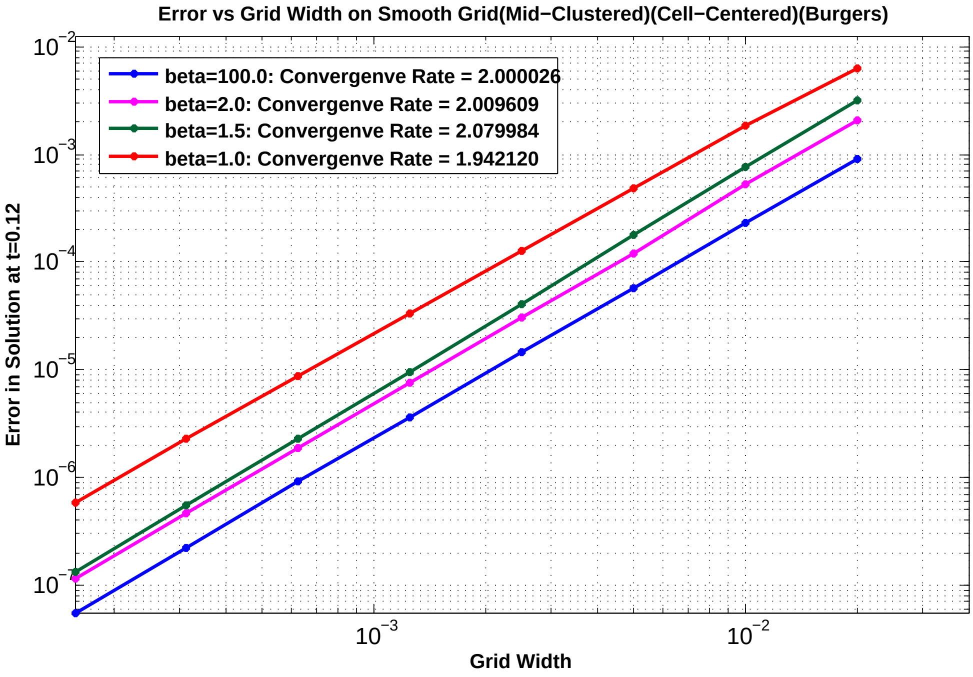


Figure 3.13: Convergence Analysis on Random Grid: The rate of convergence is very close to 2 with Cell-centered scheme



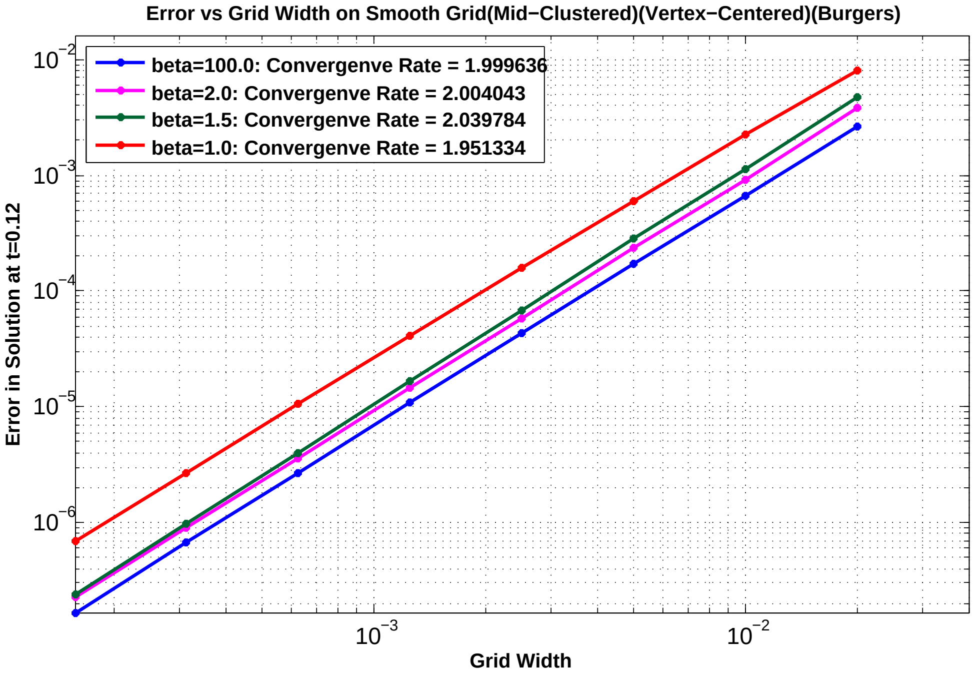
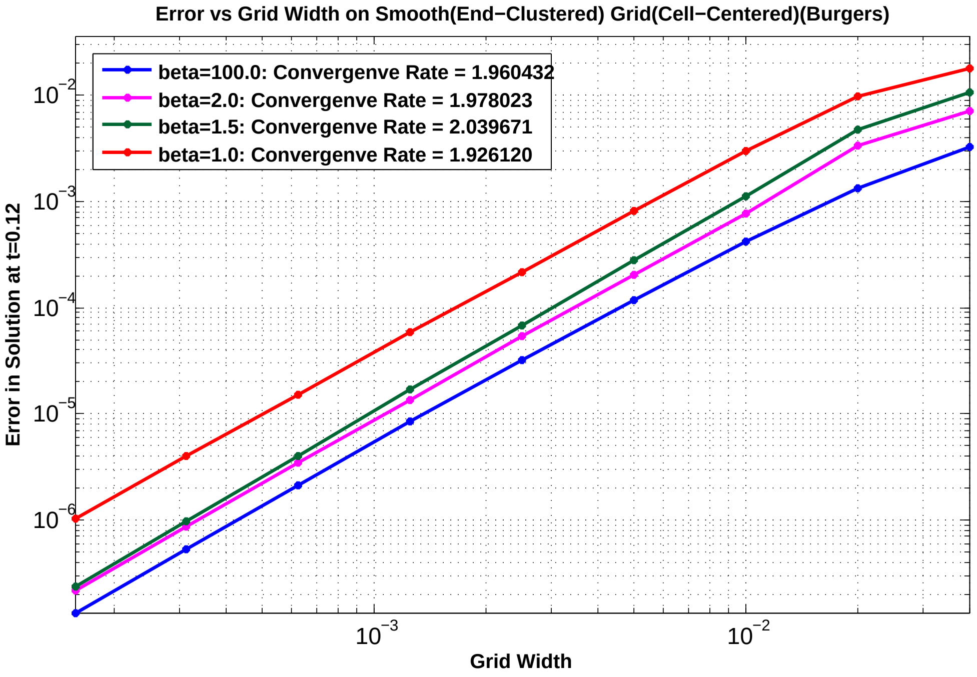


Figure 3.14: Convergence Analysis on Smooth Grid in which cells are clustered at the middle: beta = 1.0 results in slow convergence



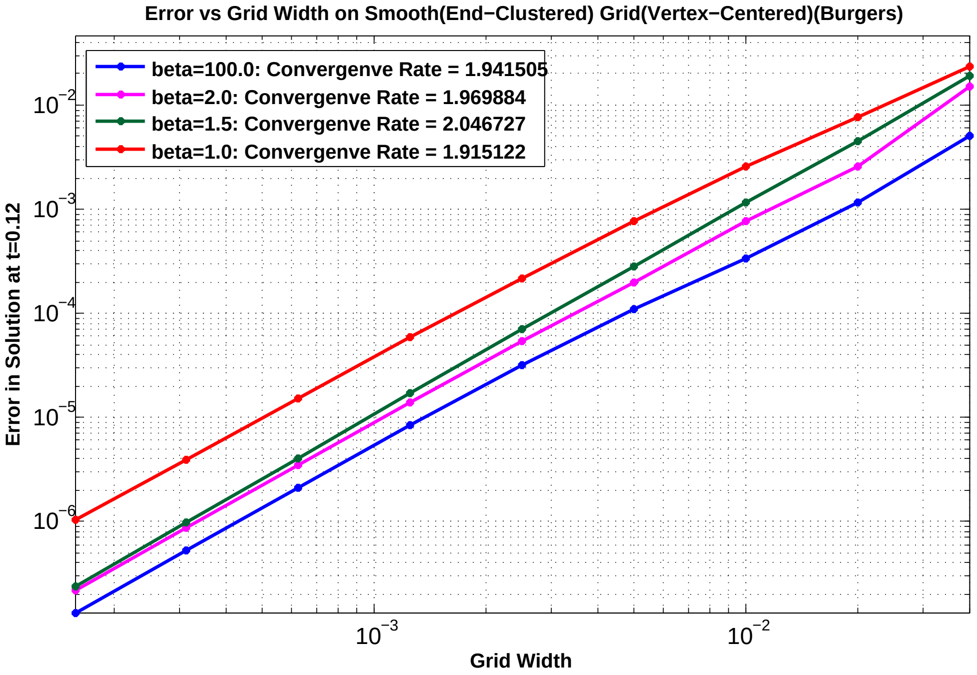
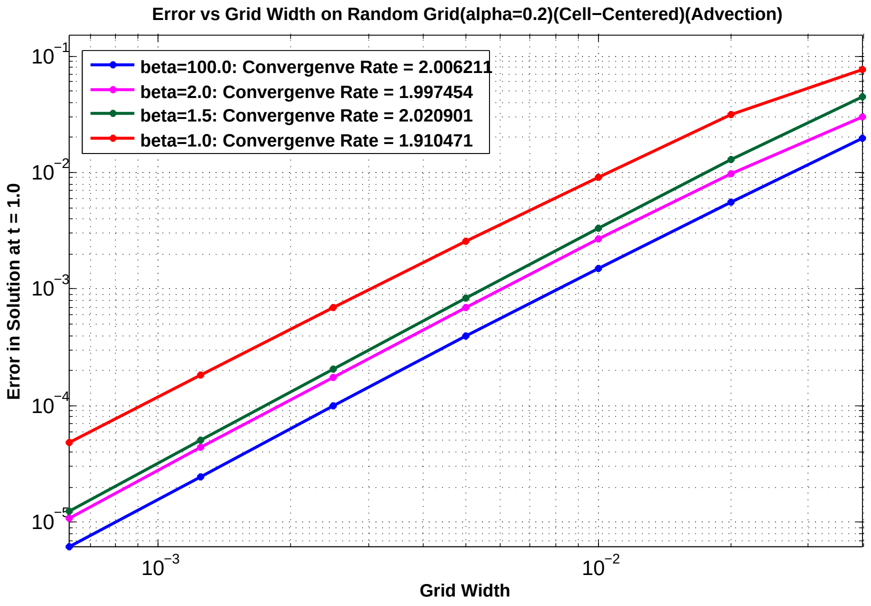


Figure 3.15: Convergence Analysis on Smooth Grid in which cells are clustered at the ends: beta = 1.0 results in slow convergence

3.12.4 Comparison of Cell-centered and Vertex-centered approach for different values of parameter beta considering solution to 1D Advection Equation



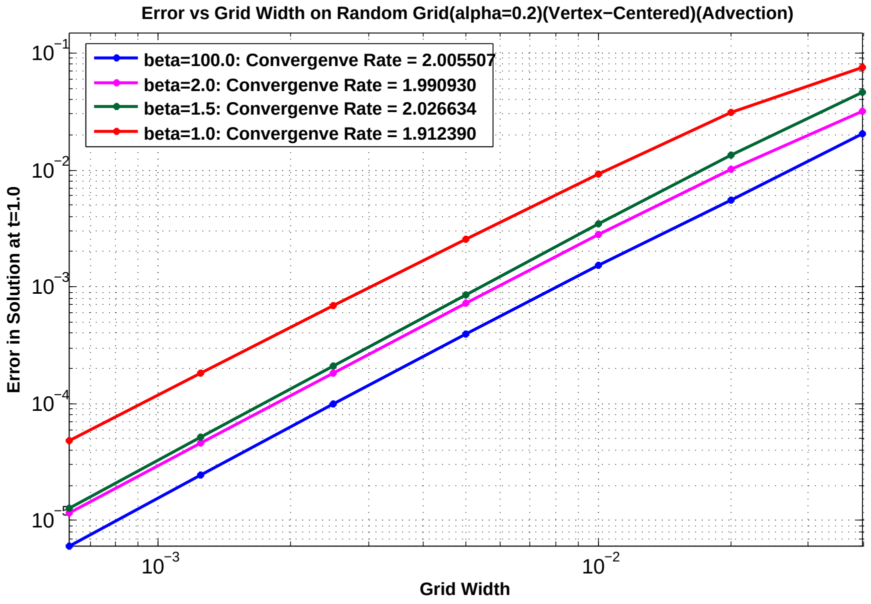
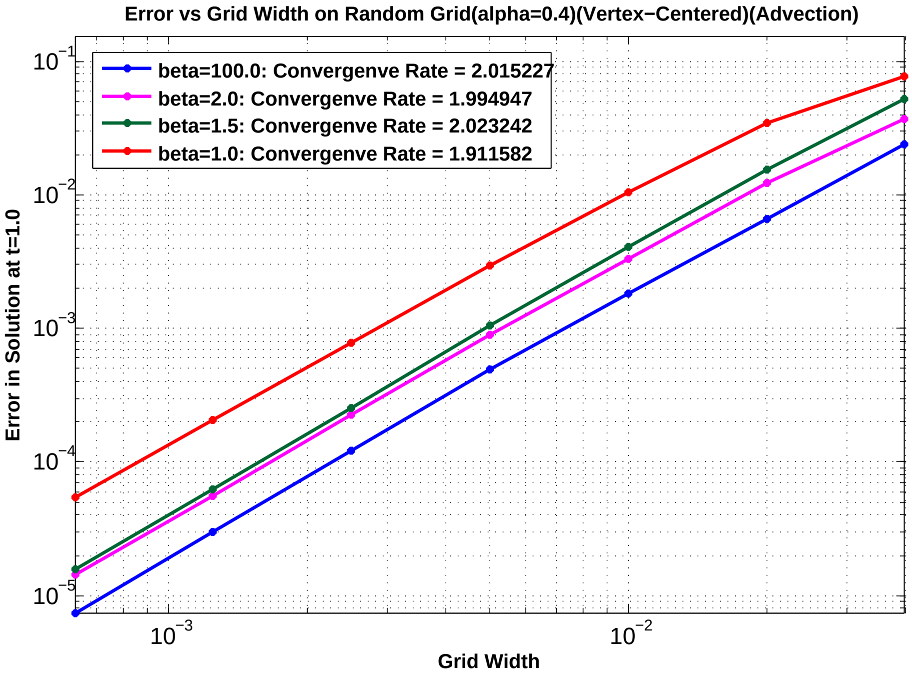


Figure 3.16: Convergence Analysis on Random grid with Second Order scheme: beta = 1.0 results in slow convergence



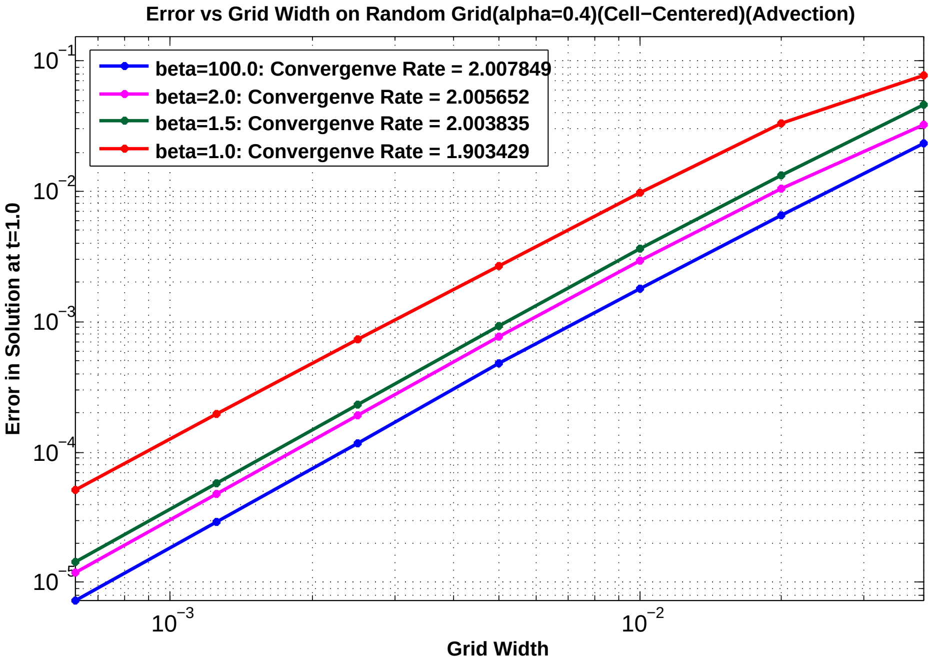


Figure 3.17: Convergence Analysis on Random grid with Second Order scheme: beta = 1.0 results in slow convergence

3.13 Conclusions

• Careful consideration of plots which compare different flux schemes suggest that Lax-Friedrichs is very dissipative scheme. Lax-Wendroff also results in lot of oscillations near the shock. Godunov, Roe and Engquist-Osher are very good flux schemes in general but both Roe and Inguist-Osher sometimes require incorporation of Entropy Fix. Local Lax-Friederichs also does a good job but Godunov seems to be the best.

• Considering any of the figures $fig.$ 4.2 , $fig.$ 3.11 , it can be concluded that second order scheme is better than the first order scheme since there are lesser number of points in the shock region when second order scheme is considered when compared to first order scheme.

• Considering $fig.$ 3.12 , $fig.$ 3.13 ,$fig.$ 3.14, $fig.$ 3.15 , $fig.$ 3.16 ,$fig.$ 3.17, it can be concluded that the cell-centered scheme does is better than the vertex-centered scheme because in most of the cases for different grids considered, error is found to be less in case when cell-centered scheme is implemented.

• Again considering $fig.$ 3.12 , $fig.$ 3.13 ,$fig.$ 3.14, $fig.$ 3.15 , $fig.$ 3.16 ,$fig.$ 3.17, it can be consistently seen that convergence rate is greater for beta =1.5 when compared to beta being 1, 2 or 100. This suggests that the intermediate value of beta should be chosen in the *minmod* limiter in order to get high rates of convergence.

.

Chapter 4

Finite Volume Method on Moving Mesh

4.1 Overview

Moving mesh, as the name suggests, refers to completely dynamic mesh where finite volume cells may change their location in time. Static meshes can be replaced by Moving meshes in order to get highly accurate solution in the regions which require more computational effort. For example, if there is a shock moving through the mesh, then there are lot of points clustered around the shock, clustering might be denser at the position of the shock than other positions on the mesh. In that case, it is desirable to have more cells in the shock region to get more accurate numerical solution. Not many finite volume cells could be considered in the regions which does not require more computational effort.

Starting with a mesh lying in the domain $[a_0, b_0]$, numerical domain at time t, $D(t) := [a(t), b(t)]$, where $a(t)$ and $b(t)$ denote the left and right mesh boundaries after time t, for an initial value problem is divided into cells in the following way:

$$D(t) = \underset{\mathbf{i\in K}}{\cup} \left[x_{i-\frac{1}{2}}\left(t\right), x_{i+\frac{1}{2}}\left(t\right)\right] \text{ or } D(t) = \underset{\mathbf{i\in K}}{\cup} C_i\left(t\right)$$

where $K = \{1, 2, ..., N-1, N\}$ is set of nodes , N being the total number of cells considered and $x_{\frac{1}{2}}\left(t\right) = a\left(t\right)$, $x_{N+\frac{1}{2}}\left(t\right) = b\left(t\right)$ and $C_i\left(t\right) = \{[x_{i-\frac{1}{2}}\left(t\right), x_{i+\frac{1}{2}}\left(t\right)], i \in K\}$ are the cells. Here the leftmost face is taken to be at the left boundary of the the numerical domain and the rightmost face to be at the right boundary. Here cells are indexed by integers $i \in K$, and the faces are indexed by half-integers $\left(i+\frac{1}{2}\right) \forall i \in \{0\} \cup K$. The common face of cell i and cell $i+1$ is indexed by $\left(i+\frac{1}{2}\right)$. Mesh at any time t is represented by set of points $a(t) = x_1(t) \leq x_2(t) \leq ...... \leq x_{N-1}(t) \leq x_N(t) = b(t)$. Solution at each cell depends on the fluxes coming in and going out. In this simple 1-**D** case, each cell has two faces and each face has two adjoining cells. The flux across any face is positive for one adjoining cell and negative for the other one. So, for an arbitrary cell, contribution from one face is added and contribution from the other face is subtracted.

Consider an arbitrary finite volume cell,

$$C_i\left(t\right) = \left[x_{i-\frac{1}{2}}\left(t\right), x_{i+\frac{1}{2}}\left(t\right)\right]$$

Considering the following quantity, the total mass inside the i'th cell.

$$Q(i) = \int\limits_{x_{i-\frac{1}{2}}(t)}^{x_{i+\frac{1}{2}}(t)} u\left(x,t\right)dx \tag{4.1}$$

In terms of grid width's, $h_i(t) = x_{i+\frac{1}{2}}\left(t\right) - x_{i-\frac{1}{2}}\left(t\right)$, cell average value of the solution in i'th cell, denoted by $u_i\left(t\right)$ is

$$u_i\left(t\right) = \frac{1}{h_i\left(t\right)} \int\limits_{x_{i-\frac{1}{2}}(t)}^{x_{i+\frac{1}{2}}(t)} u\left(x,t\right)dx$$

$$\Rightarrow \int\limits_{x_{i-\frac{1}{2}}(t)}^{x_{i+\frac{1}{2}}(t)} u\left(x,t\right)dx = u_i\left(t\right)h_i\left(t\right) \tag{4.2}$$

Consider rate of change of $Q(i)$ with respect to time given defined in (4.1)

$$\frac{\mathrm{d}}{\mathrm{d}t}Q(i) = \frac{\mathrm{d}}{\mathrm{d}t}\left[\int\limits_{x_{i-\frac{1}{2}}(t)}^{x_{i+\frac{1}{2}}(t)} u\left(x,t\right)dx\right]$$

By using Chain Rule of differentiation and Leibnitz Integral Rule, above expression reduces to

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}Q(i) &= \int\limits_{x_{i-\frac{1}{2}}(t)}^{x_{i+\frac{1}{2}}(t)} \frac{\partial u\left(x,t\right)}{\partial t}dx + u\left(x_{i+\frac{1}{2}}\left(t\right), t\right)\frac{\mathrm{d}}{\mathrm{d}t}x_{i+\frac{1}{2}}\left(t\right) \\ &\qquad u\left(x_{i-\frac{1}{2}}\left(t\right), t\right)\frac{\mathrm{d}}{\mathrm{d}t}x_{i-\frac{1}{2}}\left(t\right)\end{aligned}$$

$$= -\int\limits_{x_{i-\frac{1}{2}}(t)}^{x_{i+\frac{1}{2}}(t)} \frac{\partial f(x,t)}{\partial x}dx + u\left(x_{i+\frac{1}{2}}\left(t\right), t\right)\omega_{i+\frac{1}{2}}\left(t\right) - u\left(x_{i-\frac{1}{2}}\left(t\right), t\right)\omega_{i-\frac{1}{2}}\left(t\right) \tag{4.3}$$

where $w_{i+\frac{1}{2}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}x_{i+\frac{1}{2}}(t)$ is the velocity with which interface of i'th and $(i+1)'$th cell moves. In this notation, (4.3) further simplifies to

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}Q(i) = &-\left[f\left(u\left(x_{i+\frac{1}{2}}(t),t\right)\right) - f\left(u\left(x_{i-\frac{1}{2}}(t),t\right)\right)\right] \\ &+\left[u\left(x_{i+\frac{1}{2}}(t),t\right)\omega_{i+\frac{1}{2}}(t) - u\left(x_{i-\frac{1}{2}}(t),t\right)\omega_{i-\frac{1}{2}}(t)\right]\end{aligned}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}Q(i) = &-\left[f\left(u\left(x_{i+\frac{1}{2}}(t),t\right)\right) - u\left(x_{i+\frac{1}{2}}(t),t\right)\omega_{i+\frac{1}{2}}(t)\}\right] \\ &-\left[f\left(u\left(x_{i-\frac{1}{2}}(t),t\right)\right) - u\left(x_{i-\frac{1}{2}}(t),t\right)\omega_{i-\frac{1}{2}}(t)\}\right]\end{aligned} \tag{4.4}$$

Further introducing a velocity dependent flux

$$g(u;w) = f(u) - uw$$

and then approximating flux g by numerical flux $\hat{g}$, (4.4) reduces to

$$\frac{\mathrm{d}}{\mathrm{d}t}Q(i) = -\left[F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right] \tag{4.5}$$

$$\text{where } F_{i+\frac{1}{2}} = \hat{g}\left(u^L_{i+\frac{1}{2}}, u^R_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right) \tag{4.6}$$

is the numerical flux at common face of and i'th and $(i+1)'$th and i'th cell
From (4.2) and (4.5), it is apparent that Moving Mesh finite volume scheme is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[u_i(t)\,h_i(t)\right] = -\left[F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right] \tag{4.6}$$

Since discrete time steps are considered for numerical implementation, interface velocity is taken to be constant in any time interval

$$w_{i+\frac{1}{2}}(t) = w_{i+\frac{1}{2}}(t_n) = w^n_{i+\frac{1}{2}}(\text{shorthand notation}) \ \forall\ t \in [t_n, t_{n+1}]$$

Further in Cell-centered approach, cell boundaries are moved and then node is considered to be at the middle of the two boundaries.

$$x^{n+1}_{i+\frac{1}{2}} = x^n_{i+\frac{1}{2}} + w^n_{i+\frac{1}{2}}\Delta t$$

where $\Delta t = t^{n+1} - t^n$.

$$x^{n+1}_i = \frac{x^{n+1}_{i+\frac{1}{2}} + x^{n+1}_{i-\frac{1}{2}}}{2}$$

4.2 Reconstruction

The left and right states for calculating the flux across a face are reconstructed from the already known cell average values $u_i's$,

$$u_i(t) = \frac{1}{h_i(t)}\int_{x_{i-\frac{1}{2}}(t)}^{x_{i+\frac{1}{2}}(t)} u(x,t)$$

Left State: $u^L_{i+\frac{1}{2}}(t) = u_i(t) + \left[x_{i+\frac{1}{2}}(t) - x_i(t)\right] minmod\left(s^L_i(t), s^M_i(t), s^R_i(t)\right)$

where minmod is a function which returns the argument with lowest absolute value,

$s^L_i(t) = \dfrac{u_{i+1}(t) - u_i(t)}{x_{i+1}(t) - x_i(t)}$ is the left difference, $s^R_i(t) = \dfrac{u_i(t) - u_{i-1}(t)}{x_i(t) - x_{i-1}(t)}$ is the right difference

$s^M_i(t)$, the central difference could be $\dfrac{u_{i+1}(t) - u_{i-1}(t)}{x_{i+1}(t) - x_{i-1}(t)}$ which is not second order accurate.

We could also choose $s^M_i(t) = \dfrac{\frac{1}{R_j(t)}\left[u_{j+1}(t) - u_j(t)\right] + R_j(t)\left[u_j(t) - u_{j-1}(t)\right]}{x_{j+1}(t) - x_{j-1}(t)}$

which is second order accurate difference, where $R_j(t) = \dfrac{x_{j+1}(t) - x_j(t)}{x_j(t) - x_{j-1}(t)}$

Right State: $u^R_{i+\frac{1}{2}}(t) = u_{i+1}(t) - \left[x_{i+1}(t) - x_{i+\frac{1}{2}}(t)\right] - minmod\left(s^L_{i+1}(t), s^M_{i+1}(t), s^R_{i+1}(t)\right)$

and then finally

$$F_{i+\frac{1}{2}}(t) = F_{i+\frac{1}{2}}\left(u^L_{i+\frac{1}{2}}(t), u^R_{i+\frac{1}{2}}(t); w_{i+\frac{1}{2}}(t)\right)$$

4.3 Flux Scheme

Roe Scheme: The following flux function is used in Roe scheme.

$$F^{Roe}_{i+\frac{1}{2}}\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right) = \frac{g\left(u^-_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right) + g\left(u^+_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right)}{2} - \frac{1}{2}|a_{i+\frac{1}{2}}|\left(u^+_{i+\frac{1}{2}} - u^-_{i+\frac{1}{2}}\right)$$

where

$$a_{i+\frac{1}{2}} = \frac{f(u^+_{i+\frac{1}{2}}) - f(u^-_{i+\frac{1}{2}})}{u^+_{i+\frac{1}{2}} - u^-_{i+\frac{1}{2}}} - w_{j+\frac{1}{2}}$$

and $g(u; w) = f(u) - uw$.

This scheme could be modified by incorporating entropy fix which has been explained in section 3.8. $a_{j+\frac{1}{2}}$ is not allowed to go zero and is lower bounded by a quantity. This could be done by using $a'_{j+\frac{1}{2}}$ in place of $a_{j+\frac{1}{2}}$ where

$$a'_{j+\frac{1}{2}} = \begin{cases} a_{j+\frac{1}{2}}, & \text{if } a_{j+\frac{1}{2}} > \epsilon \\ \frac{\epsilon + \frac{\left(a_{j+\frac{1}{2}}\right)^2}{\epsilon}}{2} & \text{if } a_{j+\frac{1}{2}} < \epsilon \end{cases}$$

for some small $\epsilon > 0$. This leaves $a'_{j+\frac{1}{2}} > \frac{\epsilon}{2}$. And thus, finally,

$$F_{i+\frac{1}{2}}\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right) = \frac{g\left(u^-_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right) + g\left(u^+_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right)}{2} \\ -\frac{1}{2}|a'_{i+\frac{1}{2}}|\left(u^+_{i+\frac{1}{2}} - u^-_{i+\frac{1}{2}}\right)$$

Godunov Scheme:
The following flux function is used in Godunov scheme.

$$F_{i+\frac{1}{2}}\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\right) = \max\Big\{g\left(\max\left(w_{i+\frac{1}{2}}, u^-_{i+\frac{1}{2}}\right), w_{i+\frac{1}{2}}\right), \\ g\left(\min\left(w_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}\right), w_{i+\frac{1}{2}}\right)\Big\}$$

where $g(u; w) = f(u) - uw$

4.4 Time Integration

3-stage Runge Kutta scheme has been used which is described as follows.

□ Initialisation

$$x^{(0)}_{i+\frac{1}{2}} = x^n_{i+\frac{1}{2}}, \quad x^{(0)}_i = x^n_i, \quad u^{(0)}_i = u^n_i$$

□ Step 1

$$x^{(1)}_{i+\frac{1}{2}} = x^{(0)}_{i+\frac{1}{2}} + w_{i+\frac{1}{2}}\Delta t_n$$
$$h^{(1)}_i = x^{(1)}_{i+\frac{1}{2}} - x^{(1)}_{i-\frac{1}{2}}$$
$$u^{(1)}_i = \frac{1}{h^{(1)}_i}\left[h^{(0)}_i u^{(0)}_i + dt\mathcal{R}\left(x^{(0)}, u^{(0)}\right)\right]$$

□ Step 2

$$x^{(2)}_{i+\frac{1}{2}} = \frac{3}{4}x^{(0)}_{i+\frac{1}{2}} + \frac{1}{4}\left[x^{(1)}_{i+\frac{1}{2}} + w_{i+\frac{1}{2}}\Delta t_n\right]$$
$$x^{(2)}_i = \frac{x^{(2)}_{i+\frac{1}{2}} + x^{(2)}_{i-\frac{1}{2}}}{2}, \quad h^{(2)}_i = x^{(2)}_{i+\frac{1}{2}} - x^{(2)}_{i-\frac{1}{2}}$$
$$u^{(2)}_i = \frac{1}{h^{(2)}_i}\left[\frac{3}{4}h^{(0)}_i u^{(0)}_i + \frac{1}{4}\left(h^{(1)}_i u^{(1)}_i + dt\mathcal{R}\left(x^{(1)}, u^{(1)}\right)\right)\right]$$

□ Step 3

$$x^{(3)}_{i+\frac{1}{2}} = \frac{1}{3}x^{(0)}_{i+\frac{1}{2}} + \frac{2}{3}\left[x^{(2)}_{i+\frac{1}{2}} + w_{i+\frac{1}{2}}\Delta t_n\right]$$
$$x^{(3)}_i = \frac{x^{(3)}_{i+\frac{1}{2}} + x^{(3)}_{i-\frac{1}{2}}}{2}, \quad h^{(3)}_i = x^{(3)}_{i+\frac{1}{2}} - x^{(3)}_{i-\frac{1}{2}}$$
$$u^{(3)}_i = \frac{1}{h^{(3)}_i}\left[\frac{1}{3}h^{(0)}_i u^{(0)}_i + \frac{2}{3}\left(h^{(2)}_i u^{(2)}_i + dt\mathcal{R}\left(x^{(2)}, u^{(2)}\right)\right)\right]$$

where $\mathcal{R}\left(x^{(k)}, u^{(k)}\right) = g\left(\left(u^-_{i+\frac{1}{2}}\right)^{(k)}, \left(u^+_{i+\frac{1}{2}}\right)^{(k)}; w_{i+\frac{1}{2}}\right) - g\left(\left(u^-_{i-\frac{1}{2}}\right)^{(k)}, \left(u^+_{i-\frac{1}{2}}\right)^{(k)}; w_{i-\frac{1}{2}}\right)$ for $k = 0, 1, 2$ and g is the numerical flux.

4.5 Thomas algorithm

It is an special form of Gaussian Elimination used for solving the system of linear equations $Ax = q$ for a tridigonal matrix A. A sequence of row operations are performed which finally lead to much simplification and evaluation of x given A and d.

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & b_3 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & a_N & b_N \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ . \\ . \\ q_{N-1} \\ q_N \end{pmatrix}$$

This algorithm is a three-step process consisting of initial decomposition of matrix A into lower triangular and upper triangular matrices followed by solving th resulting linear systems by forward and backward substitution. The flow of the algorithms goes as follows

$$Ax = q \Rightarrow LUx = q \Rightarrow Lp = q \text{ and } Ux = p$$

and finally the last two linear systems are solved by Forward substitution and Backward substitution respectively. The detailed explanation is given below

□ *Step 1: LU-Decomposition*

Matrix A is decomposed into a lower triangular matrix L and an upper triangular matrix U, $A = LU$.

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & b_3 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & a_N & b_N \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & l_3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & l_{N-1} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & l_N & 0 \end{pmatrix}$$

$$\begin{pmatrix} d_1 & u_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & u_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3 & u_3 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{N-1} & u_{N-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & d_N \end{pmatrix}$$

with the elements of L and U given by

- $d_1 = a_1, \quad u_1 = c_1$
- $l_i = b_i/d_{i-1}, \quad d_i = d_i + a_i - l_i u_{i-1}, \quad u_i = c_i \ \forall i \in \{2, 3, ...N-2, N-1\}$
- $l_N = b_N/d_{N-1}, \quad d_N = a_N - l_N u_{N-1}$

□ *Step 2: Forward Substitution*

Here system $Lp = q$ is solved by forward substitution, which means p_{i+1} is found out by already known $p_1, p_2, ..., p_i$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & l_3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & l_{N-1} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & l_N & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ . \\ . \\ p_{N-1} \\ p_N \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ . \\ . \\ q_{N-1} \\ q_N \end{pmatrix}$$

The coefficients ${p_i}'s$ are given by

- $p_1 = q_1$
- $p_i = q_i - l_i p_{i-1} \ \forall i \in \{2, 3, ..., N-1, N\}$

□ *Step 3: Backward Substitution*

Here system $Ux = p$ is solved by backward substitution, which means p_{i-1} is found out by already known $p_i, p_{i+1}, ..., p_{N-1}, p_N$

$$\begin{pmatrix} d_1 & u_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & u_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3 & u_3 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{N-1} & u_{N-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & d_N \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ . \\ . \\ p_{N-1} \\ p_N \end{pmatrix}$$

The coefficients ${p_i}'s$ are given by

- $x_N = p_N/d_N$
- $x_i = (p_i - u_i x_{i+1})/d_i \ \forall i \in \{N-1, N-2, ..., 2, 1\}$

To sum up, in slightly different notations, this is how various steps are implemented:

□ ${c'}_1 = c_1/b_1$

□ ${c'}_i = c_i/\left(b_i - {c'}_{i-1} a_i\right) \quad \forall i \in \{2, 3, ..., N-1, N\}$

□ $d'_1 = d_1/b_1$
□ $d'_i = (d_i - d'_{i-1}a_i) \,/\, (b_i - c'_{i-1}a_i) \quad \forall i \in \{2, 3, ..., N-1, N\}$

□ $x_N = d'_N$
□ $x_i = d'_i - c'_i x_{i+1} \quad \forall i \in \{N-1, N-2, ..., 2, 1\}$
x is the final solution.

4.6 CFL Condition

Here parameters $\alpha < 1$ and $\beta > 1$ are chosen so that $h_{new}(i) \geq (1-\alpha)\, h_{old}(i)$, $\forall i \in \mathbb{Z}$ where $h_{old}(i) = x^n_{i+\frac{1}{2}} - x^n_{i-\frac{1}{2}}$ is the old spacing and $h_{new}(i) = x^{n+1}_{i+\frac{1}{2}} - x^{n+1}_{i-\frac{1}{2}}$ is the new spacing. In the implementation $\alpha = 0.9$ and $\beta = 1.1$ have been used. This is implemented in the following way.

- Choose $\Delta t^1_n = \min\limits_i \left(\dfrac{h^n_i}{f'\left(u^n_i - w^n_i\right)} \right)$

- Choose $\Delta t^2_n = \min\limits_i \left(\dfrac{(1-\alpha)\, h^n_i}{-\Delta w^{n,-}_i}, \dfrac{(\beta - 1)\, h^n_i}{\Delta w^{n,+}_i} \right)$

where $\Delta w^{n,-}_i = \min \left(w^n_{i+\frac{1}{2}} - w^n_{i-\frac{1}{2}}, 0 \right)$ and $\Delta w^{n,+}_i = \max \left(w^n_{i+\frac{1}{2}} - w^n_{i-\frac{1}{2}}, 0 \right)$

- Finally, take Δt_n to be the minimum of the two.

$$\Delta t_n = \min\left(\Delta t^1_n, \Delta t^2_n\right)$$

4.7 Different Mesh Velocities considered and corresponding results with Cell-Centered Moving Mesh method

□ One way is to choose constant velocity for the mesh, i.e. $w^n_{i+\frac{1}{2}} = c, \forall n \in \mathbb{Z}^+$, for all faces, where c is a constant. In this case, the solution can evolve in the mesh indefinitely with excellent scheme like Godunov Scheme.

□ Velocity equivalent to local shock speeds
Following velocity is considered

$$\overline{w}^n_{i+\frac{1}{2}} = \frac{f(u^n_{i+1}) - f(u^n_i)}{u^n_{i+1} - u^n_i}$$

Nodes near the shock come very close to each other which makes time step becomes very small and thus the numerical implementation cannot be performed for long time with this choice of velocities.

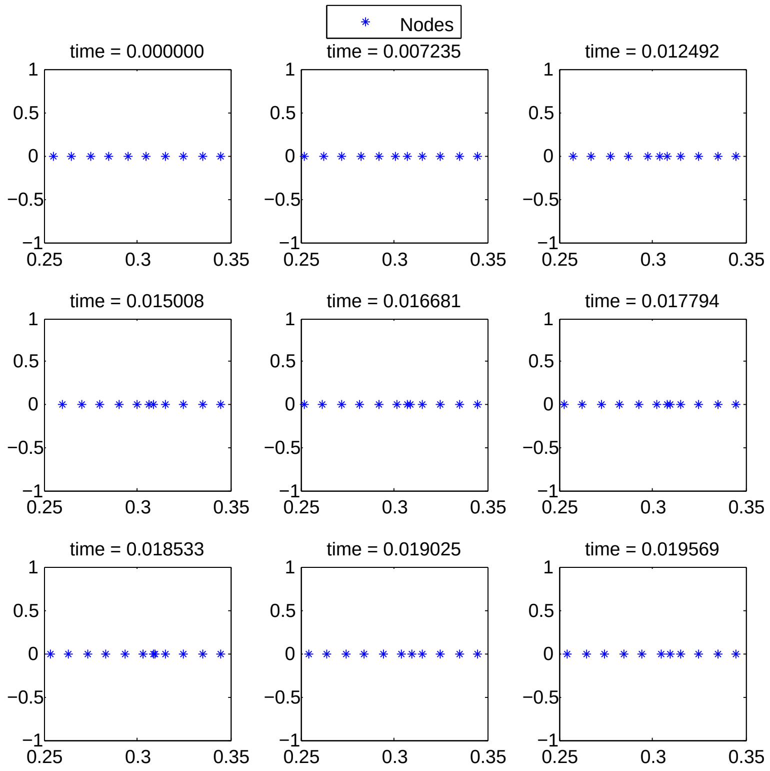


Figure 4.1: For the Riemann problem with u_l=1 and u_r=0, the two nodes coming closer can be clearly seen which results in very small time step

□ Velocity smoothed out using local shock speeds

- Assigning velocity to cell faces

$$\overline{w}^n_{i+\frac{1}{2}} = \frac{f(u^n_{i+1}) - f(u^n_i)}{u^n_{i+1} - u^n_i}$$

- Smoothing it

Consider the quantity,

$$J = \frac{1}{2}\Sigma_{j\in\mathbb{Z}}\left(w_{j+\frac{1}{2}} - \overline{w}_{j+\frac{1}{2}}\right)^2 + \frac{\alpha}{2}\Sigma_{j\in\mathbb{Z}}\left(\frac{w_{j+\frac{1}{2}} - w_{j-\frac{1}{2}}}{x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}}\right)^2$$

To get smooth velocity profile from unsmoothed velocities, J could be minimised over $w_{i+\frac{1}{2}}$. This could be done by setting $\frac{\partial J}{\partial w_{j+\frac{1}{2}}}$ equal to zero.

$$\frac{\partial J}{\partial w_{j+\frac{1}{2}}} = 0 \quad \Rightarrow \quad \left(w_{j+\frac{1}{2}} - \overline{w}_{j+\frac{1}{2}}\right) + \alpha\left(\frac{w_{j+\frac{1}{2}} - w_{j-\frac{1}{2}}}{\left(x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}\right)^2} + \frac{w_{j+\frac{1}{2}} - w_{j+\frac{3}{2}}}{\left(x_{j+\frac{1}{2}} - x_{j+\frac{3}{2}}\right)^2}\right) = 0$$

$$\Rightarrow \quad \left(w_{j+\frac{1}{2}} - \overline{w}_{j+\frac{1}{2}}\right) + \alpha\left(\frac{w_{j+\frac{1}{2}} - w_{j-\frac{1}{2}}}{{h_j}^2} + \frac{w_{j+\frac{1}{2}} - w_{j+\frac{3}{2}}}{{h_{j+1}}^2}\right) = 0$$

$$\Rightarrow a_{j+\frac{1}{2}}w_{j-\frac{1}{2}} + b_{j+\frac{1}{2}}w_{j+\frac{1}{2}} + c_{j+\frac{1}{2}}w_{j+\frac{3}{2}} = \overline{w}_{j+\frac{1}{2}}$$

with coefficients given by

$$a_{j+\frac{1}{2}} = -\frac{\alpha}{h_j^2}, \quad b_{j+\frac{1}{2}} = 1 - \frac{\alpha}{h_j^2} - \frac{\alpha}{h_{j+1}^2}, \quad c_{j+\frac{1}{2}} = -\frac{\alpha}{h_{j+1}^2}$$

The smooth velocities $w_{j+\frac{1}{2}}'s$ can be found out by solving the system $Aw = \overline{w}$ with A being a tridigonal matrix using Thomas Algorithm.

The smooth velocities $w_{j+\frac{1}{2}}'s$ obtained above are slightly problematic since the velocity profile starts becoming nearly vertical in the shock region. In order to avoid this, the velocities $w_{j+\frac{1}{2}}'s$ could be chosen such that points ahead of the shock in the direction of shock are assigned slightly larger velocity as compared to the points before the shock. For instance for the Reimann problem with $u_l = 1, u_r = 0$, choice $w'_{j+\frac{1}{2}} = w_{j+\frac{1}{2}} + 0.3 + 0.2n_i/N$ works out where $n_i = i$ is the node index corresponding to the i'th node, $w_{j+\frac{1}{2}}$ being the smoothed velocities obtained from the above rule. In general, choice $w'_{j+\frac{1}{2}} = w_{j+\frac{1}{2}} + a_1 + a_2 * s * n_i/N$, s being the shock speed works out for certain values of a_1 and a_2 for the Reimann problem considered where shock gets generated.

When smoothed velocities obtained by minimising the quantity J are used then it can be seen that smaller values of α result in velocities close to local shock speeds. Larger values of α, result in almost a straight line profile of velocities .

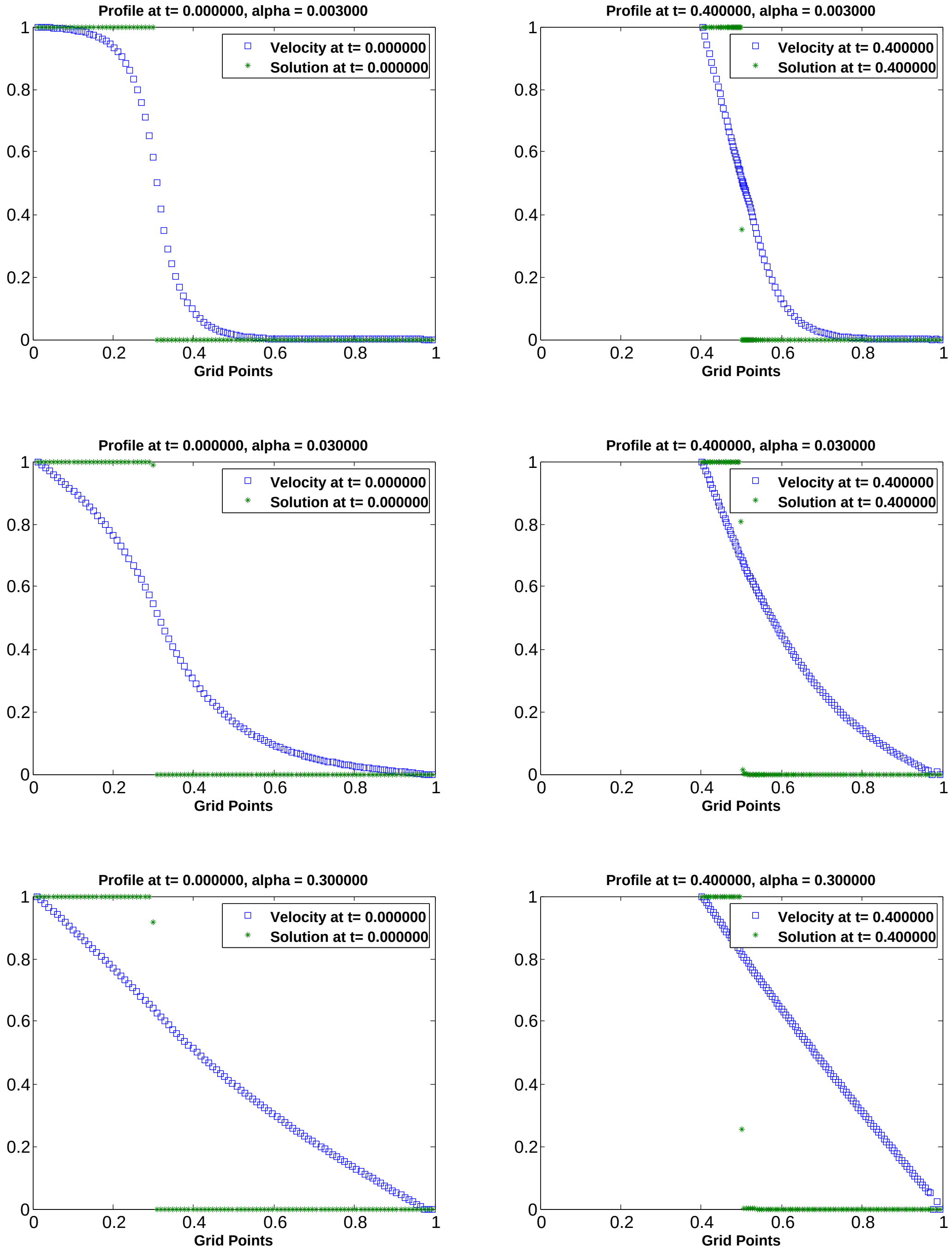


Figure 4.2: Riemann Problem with u_l=1 and u_r=0, on moving grids

□ Another choice of velocities is as follows: • Assign velocity to cell faces

$$\overline{w}^n_{j+\frac{1}{2}} = \frac{u^n_j + u^n_{j+1}}{2} \qquad \text{for Burgers}$$

- Consider the quantity,

$$S_{j+\frac{1}{2}} = \frac{1}{3}\left(u_j + u_{j+1} + \frac{u_{j-1}+u_{j+2}}{2}\right) \text{ or } \overline{w}_{j+\frac{1}{2}}$$

- If S_j is positive, then choose

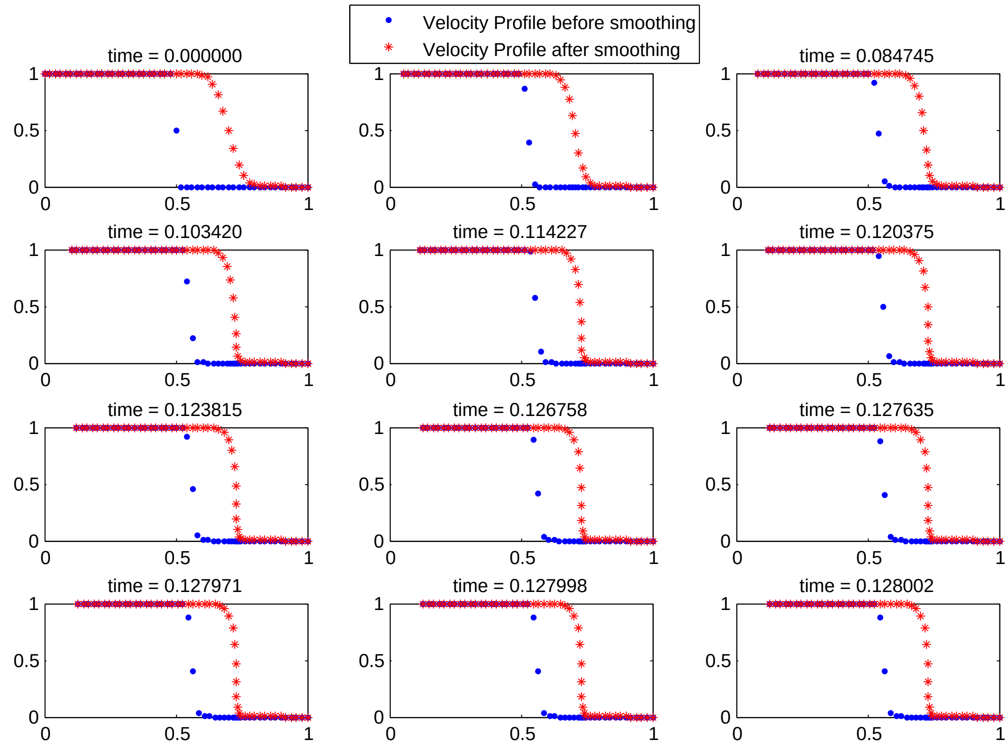
$$w_{j+\frac{1}{2}} = \frac{1}{2}\left(\overline{w}_{j-\frac{1}{2}} + \overline{w}_{j+\frac{1}{2}}\right)$$

- If S_j is negative, then choose

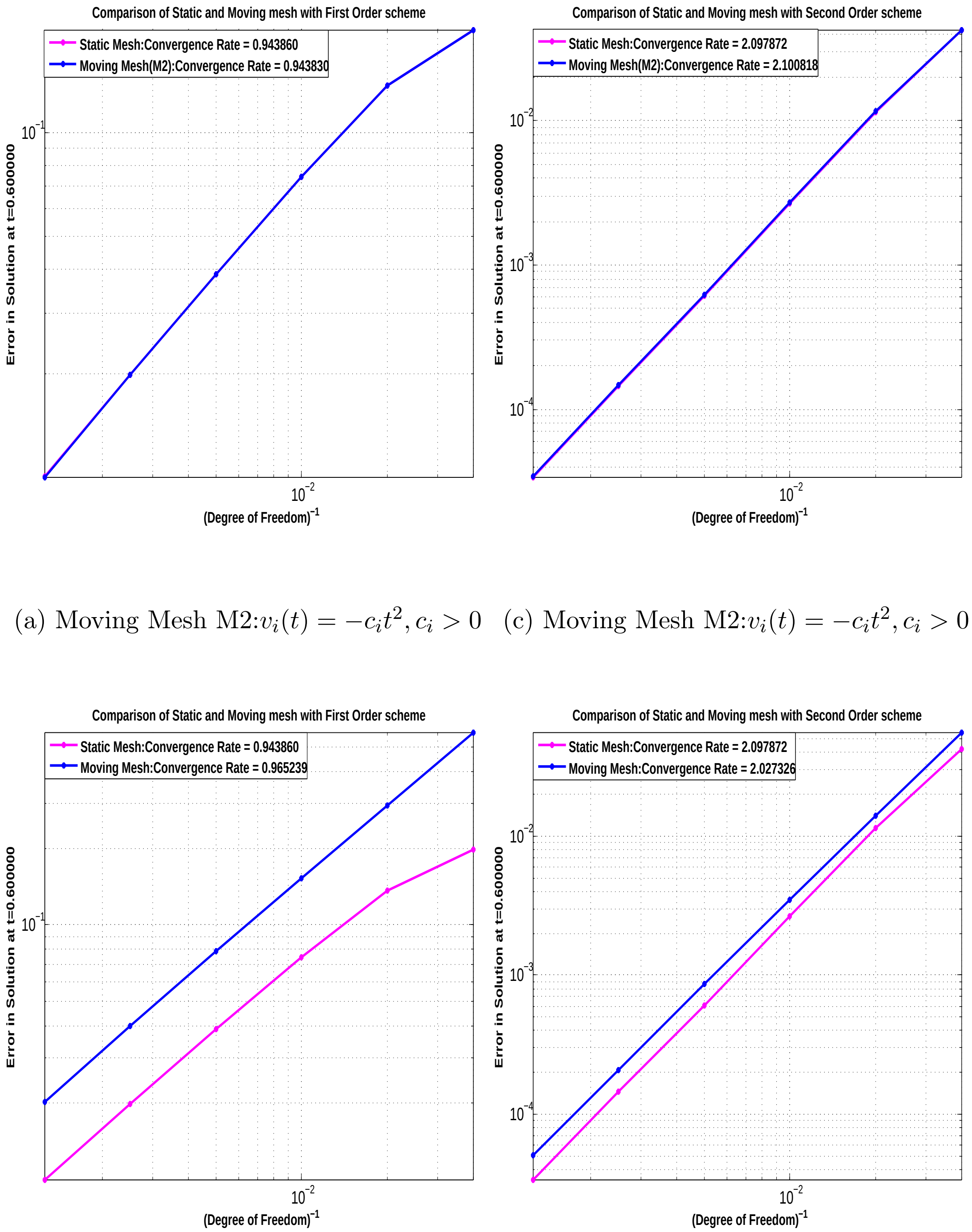
$$w_{j+\frac{1}{2}} = \frac{1}{2}\left(\overline{w}_{j+\frac{1}{2}} + \overline{w}_{j+\frac{3}{2}}\right)$$

The last two steps could be repeated multiple times in order to get very smooth velocity profile with few grid points right after the shock moving with significant non-zero velocity.

For $p = 20$, the velocity profile looks as follows and it be seen that it starts becoming vertical as time increases. After a certain time time-step becomes very small as some nodes come very close to each other and hence the solution cannot be obtained for more time.



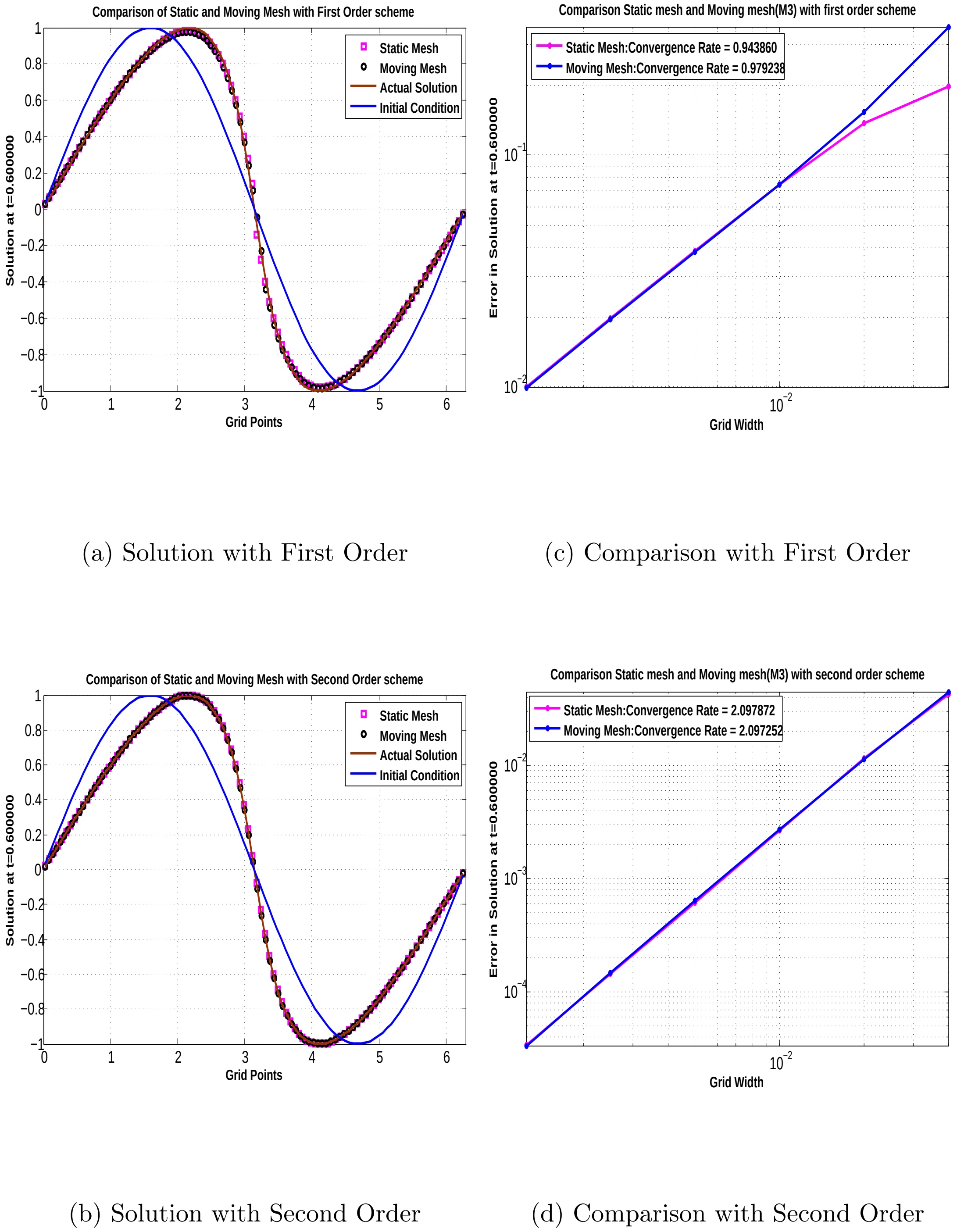
4.8 Convergence Analysis on different moving meshes



(a) Moving Mesh M2:$v_i(t) = -c_i t^2, c_i > 0$ (c) Moving Mesh M2:$v_i(t) = -c_i t^2, c_i > 0$

(b) Mesh moving with constant velocity 1.2(d) Mesh moving with constant velocity 1.2

Figure 4.3: Comparison of Moving meshes with Static Mesh



(a) Solution with First Order

(c) Comparison with First Order

(b) Solution with Second Order

(d) Comparison with Second Order

Figure 4.4: Solution on Moving Mesh M3:$[v(t) = -c_i \cos(t), c_i \geq 0]$ and its comparison with Static Mesh

It can be seen that second order scheme does better than the first order scheme since the solution with second order scheme is very close the reference solution while the solution with the first order scheme is not very close to the reference solution everywhere.

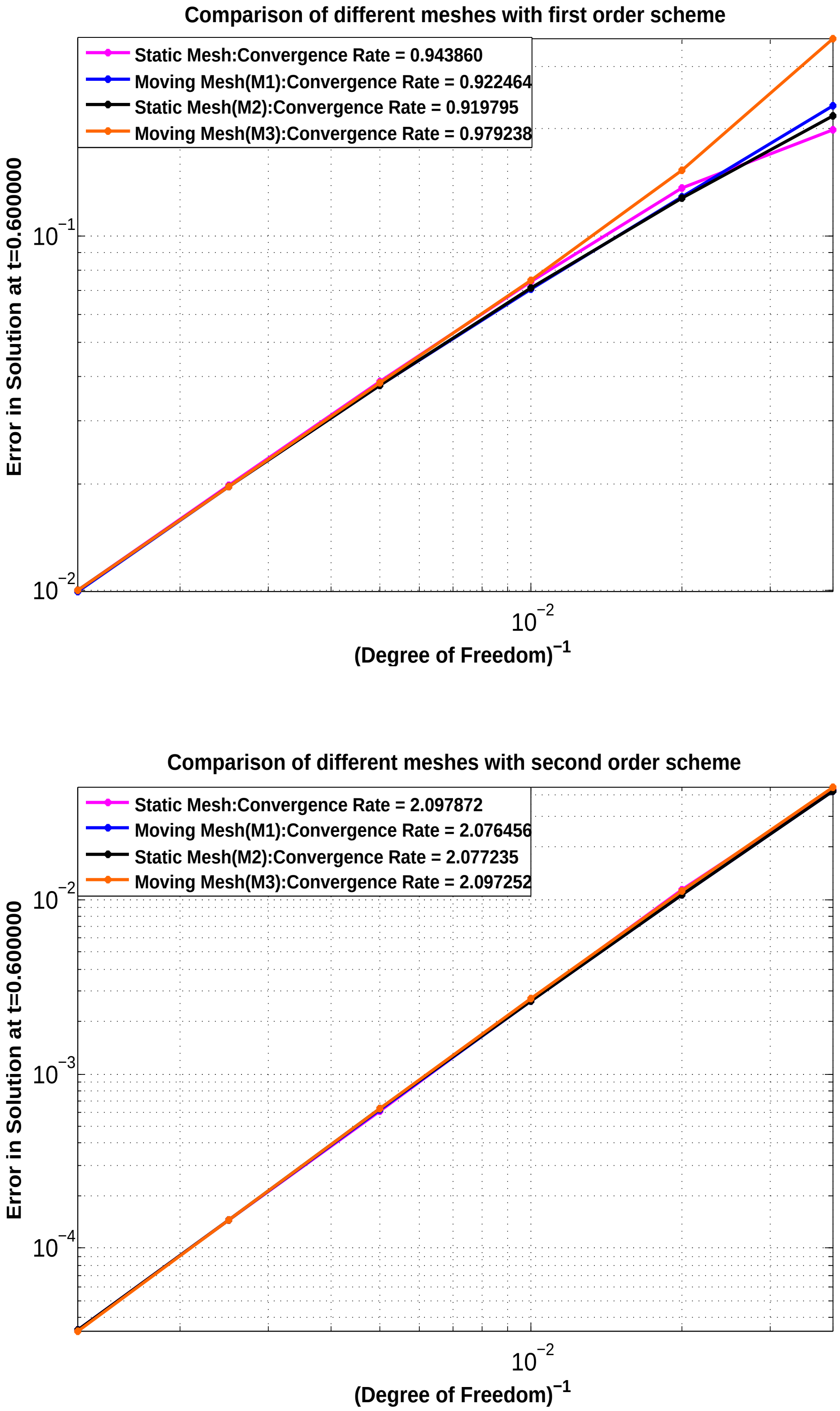


Figure 4.5: Comparison of different meshes, M1: $v_i(t) = -c_i t$, M2: $v_i(t) = -c_i t^2$, M3: $v(t) = -c_i \cos(t)$, $c_i \geq 0, c_i = -kx_{0i}\left(x_{0i} - 2\pi\right), k > 0$

Convergence plot for different meshes seems to coincide with each other when degree of freedom is large for both first and second order schemes.

4.9 Conclusion

Looking at fig. 4.5, it can be seen that the different meshes considered result in nearly the same convergence rate eventually upon mesh refinement. However, since selected velocity profiles have been tried, so it is not possible to make a general comment on the role of velocity profile. If lagrangian velocity is considered for mesh movement, then the early compression of cells could be delayed by properly smoothing out the mesh velocity profile.

Chapter 5

AREPO Moving Mesh Scheme

5.1 Overview

This scheme is finite volume analogue of Adaptive Mesh Refinement finite element method. The accuracy in the solution can be increased in the regions of interest, such as region near the shock. For $1D$ mesh generation, Voronoi approach is adopted which takes very simple form in one space dimension. It involves taking common face of any two adjacent mesh-generating points, also called nodes, at their geometric center.

$$x_{i+\frac{1}{2}}(t) = \frac{x_i(t) + x_{i+1}}{2}(t)$$

and then the numerical domain is subdivided into cells

$$D = \underset{\mathbf{i}\in\mathbf{K}}{\cup}\left[x_{i-\frac{1}{2}}\left(t\right), x_{i+\frac{1}{2}}\left(t\right)\right] \text{ or } D = \underset{\mathbf{i}\in\mathbf{K}}{\cup} C_i\left(t\right)$$

where $K = \{1, 2, ..., N-1, N\}$ is set of nodes , N being the total number of cells considered and $x_{\frac{1}{2}}\left(t\right) = a\left(t\right)$, $x_{N+\frac{1}{2}}\left(t\right) = b\left(t\right)$ and $C_i\left(t\right) = \{[x_{i-\frac{1}{2}}\left(t\right), x_{i+\frac{1}{2}}\left(t\right)], i \in K\}$ are the cells

Starting with one dimensional Conservation Law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

Note: In the discussion below $x = x(t), u = u(x,t)$ arguments have been omitted for notational simplicity

Integrating it over an arbitrary cell indexed by $i, C_i = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$

$$\int_{C_i} \frac{\partial u}{\partial t} dx = -\int_{C_i} \frac{\partial f(u)}{\partial x} dx \tag{5.0}$$

Using the rule

$$\frac{d}{dt}\left(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u dx\right) = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial u}{\partial t} dx + u\left(x_{i+\frac{1}{2}}\right)\frac{dx_{i+\frac{1}{2}}}{dt} - u\left(x_{i-\frac{1}{2}}\right)\frac{dx_{i-\frac{1}{2}}}{dt}$$

$$\Rightarrow \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial u}{\partial t} dx = \frac{d}{dt}\left(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u dx\right) + u\left(x_{i-\frac{1}{2}}\right)\frac{dx_{i-\frac{1}{2}}}{dt} - u\left(x_{i+\frac{1}{2}}\right)\frac{dx_{i+\frac{1}{2}}}{dt}$$

Using the above expression, (5.1) becomes

$$\frac{d}{dt}\left(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u dx\right) = -\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial f(u)}{\partial x} dx + u\left(x_{i+\frac{1}{2}}\right)\frac{dx_{i+\frac{1}{2}}}{dt} - u\left(x_{i-\frac{1}{2}}\right)\frac{dx_{i-\frac{1}{2}}}{dt}$$

Using the notations $Q_i = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u dx$, above expression reduces to

$$Q_i = -\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial f(u)}{\partial x} dx + u\left(x_{i+\frac{1}{2}}\right)\frac{dx_{i+\frac{1}{2}}}{dt} - u\left(x_{i-\frac{1}{2}}\right)\frac{dx_{i-\frac{1}{2}}}{dt}$$

$$\frac{dQ_i}{dt} = \left[f_{i-\frac{1}{2}} - f_{i+\frac{1}{2}} + u\left(x_{i+\frac{1}{2}}\right)\omega_{i+\frac{1}{2}} - u\left(x_{i-\frac{1}{2}}\right)\omega_{i-\frac{1}{2}}\right]$$

$$\frac{dQ_i}{dt} = -\left[\{f_{i+\frac{1}{2}} - u\left(x_{i+\frac{1}{2}}\right)\omega_{i+\frac{1}{2}}\} - \{f_{i-\frac{1}{2}} - u\left(x_{i-\frac{1}{2}}\right)\omega_{i-\frac{1}{2}}\}\right]$$

$$Q_i^{(n+1)} = Q_i^{(n)} - \Delta t\left[\hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}}\right], \ \hat{F} \text{ being the approximation of } \textit{f(u) - uw}$$

$$v_i^{n+1}h_i^{n+1} = v_i^n h_i^n - \Delta t\left[\hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}}\right]$$

where v_i^n is the numerical solution at $t = t_n$

5.2 Reconstruction

There is a need to limit the reconstructed value of a quantity at cell-interfaces from the current values in surrounding cells in order to get higher order accuracy in numerical solution. However, it is very important to apply this reconstruction only in the selected regions. In the regions of discontinuity, for example, near a shock wave, reconstruction of a can yield values which may be an overshoot or undershoot and may thus give rise to oscillations in the solution in the neighbouring finite volume cells. The gradient that has been used for reconstruction of various quantities in different cells is obtained as follows.

Starting with the following gradient

$$(\nabla\phi)_i' = \frac{1}{h_i}\sum_{j=i-1,i+1}\left(\frac{\phi_i + \phi_j}{2}\frac{\mathbf{r_{i,j}}}{r_{i,j}}\right)$$

In the convention, $\hat{r}_{i,i+1} = 1$ and $\hat{r}_{i,i-1} = -1$, above quantity takes the following form

$$(\nabla\phi)_i' = \frac{\phi_{i+1} - \phi_{i-1}}{2h_i}$$

which is simply the first order central approximation based on values of ϕ on left and right. Further, a factor limiting the gradient is added

$$(\nabla\phi)_i = \alpha_i(\nabla\phi)'_i$$

where $\alpha_i = \min\big(1, \psi_{ij}\big)$, with

$$\psi_{ij} = \begin{cases} \left(\phi_i^{max} - \phi_i\right)/\Delta\phi_{ij} & \text{if } \Delta\phi_{ij} > 0 \\ \left(\phi_i^{min} - \phi_i\right)/\Delta\phi_{ij} & \text{if } \Delta\phi_{ij} < 0 \\ 1, & \text{if } \Delta\phi_{ij} = 0 \end{cases}$$

$$\Delta\phi_{ij} = \begin{cases} (\nabla\phi)_i\left(x_{i+\frac{1}{2}} - x_i\right) & \text{for } j = i + \frac{1}{2} \\ (\nabla\phi)_i\left(x_{i-\frac{1}{2}} - x_i\right) & \text{for } j = i - \frac{1}{2} \\ 1, & \text{for } j = i \end{cases}$$

$$\phi_i^{max} = \max\left(\phi_j\right) \text{ and } \phi_i^{min} = \min\left(\phi_j\right), j \in \{i-1, i, i+1\}$$

$(\nabla\phi)_i$ is the final gradient that has been used. The factor α_i which has been multiplied acts like a limiter. It does not let the gradient become too large in the regions where ϕ fluctuates a lot.

The consequent steps below describe the reconstruction done at the face where *Muscle-Hancock* scheme has been used to calculate left and right reconstructed state at the face

□ $u(x, t_n) = u_i^n + (\nabla u)_i(x - x_i^n)$
□ $u_{i+\frac{1}{2}}^{n,-} = u_i^n + (\nabla u)_i\frac{h_i^n}{2}$
□ $u_{i-\frac{1}{2}}^{n,+} = u_i^n - (\nabla u)_i\frac{h_i^n}{2}$

Muscle-Hancock scheme is based on the reconstructed value of numerical flux at the faces at half time steps. The following finite volume scheme is used

$$u_i^{n+1}h_i^{n+1} = u_i^n h_i^n - (t_{n+1} - t_n)\left[F_{i+\frac{1}{2}}^{n+\frac{1}{2}} - F_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right]$$

where

$$F_{i+\frac{1}{2}}^{n+\frac{1}{2}} = F\left(u_{i+\frac{1}{2}}^{n+\frac{1}{2},L}, u_{i+\frac{1}{2}}^{n+\frac{1}{2},R}\right)$$

where

$$u_{i+\frac{1}{2}}^{n+\frac{1}{2},L} = u\left(x_{i+\frac{1}{2}}^L\left(t_n + \Delta t_n/2\right), t_n + \Delta t_n/2\right)$$

Following the steps, we finally get

□ $u_{i+\frac{1}{2}}^{n+\frac{1}{2},-} = u_i^n + (\nabla u)_i\frac{h_i^n}{2} - \frac{\Delta t}{2h_i^n}\left(F\left(u_{i+\frac{1}{2}}^{n,-}\right) - F\left(u_{i-\frac{1}{2}}^{n,+}\right)\right) - w_{i+\frac{1}{2}}^n\frac{\Delta t}{2}$

5.3 Mesh Movement

The following velocities have been tried out for mesh movement starting from local shock speeds.

- Velocity of node is taken to be the lagrangian velocity itself $\omega_i = v_i$

- Velocity of node is taken to be the slightly different from lagrangian velocity in order to move the node towards the centre of the cell $\omega_i = (1-\alpha)v_i + \alpha\left(x_{i+\frac{1}{2}} - x_i\right)/\Delta t$ where Δt is the time-step and α is the degree by which node is moved towards center of the cell.

Finally for the calculation purposes, velocity of face is taken to be the the average of left and right values $\omega_{i+\frac{1}{2}} = \frac{\omega_i+\omega_{i+1}}{2}$

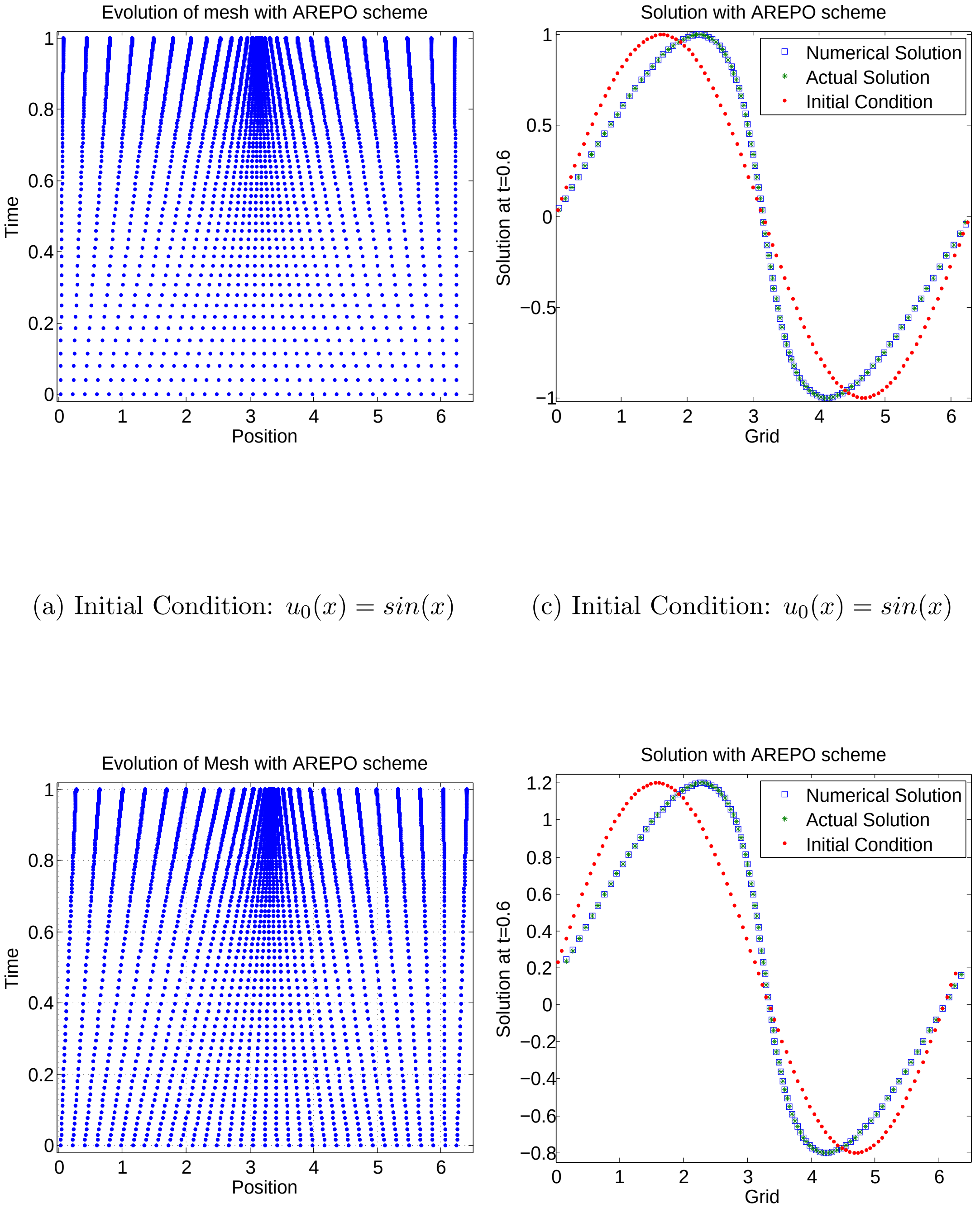
5.4 Time Integration

Local time step is calculated as follows

$$\Delta t_i = N_{CFL}\frac{h_i}{|v_i - w_i|} \text{ for cell } i$$

Finally the global time-step is chosen for the mesh evolution which is the minimum value of Δt_i over all nodes

$$\Delta t = \min_i t_i$$



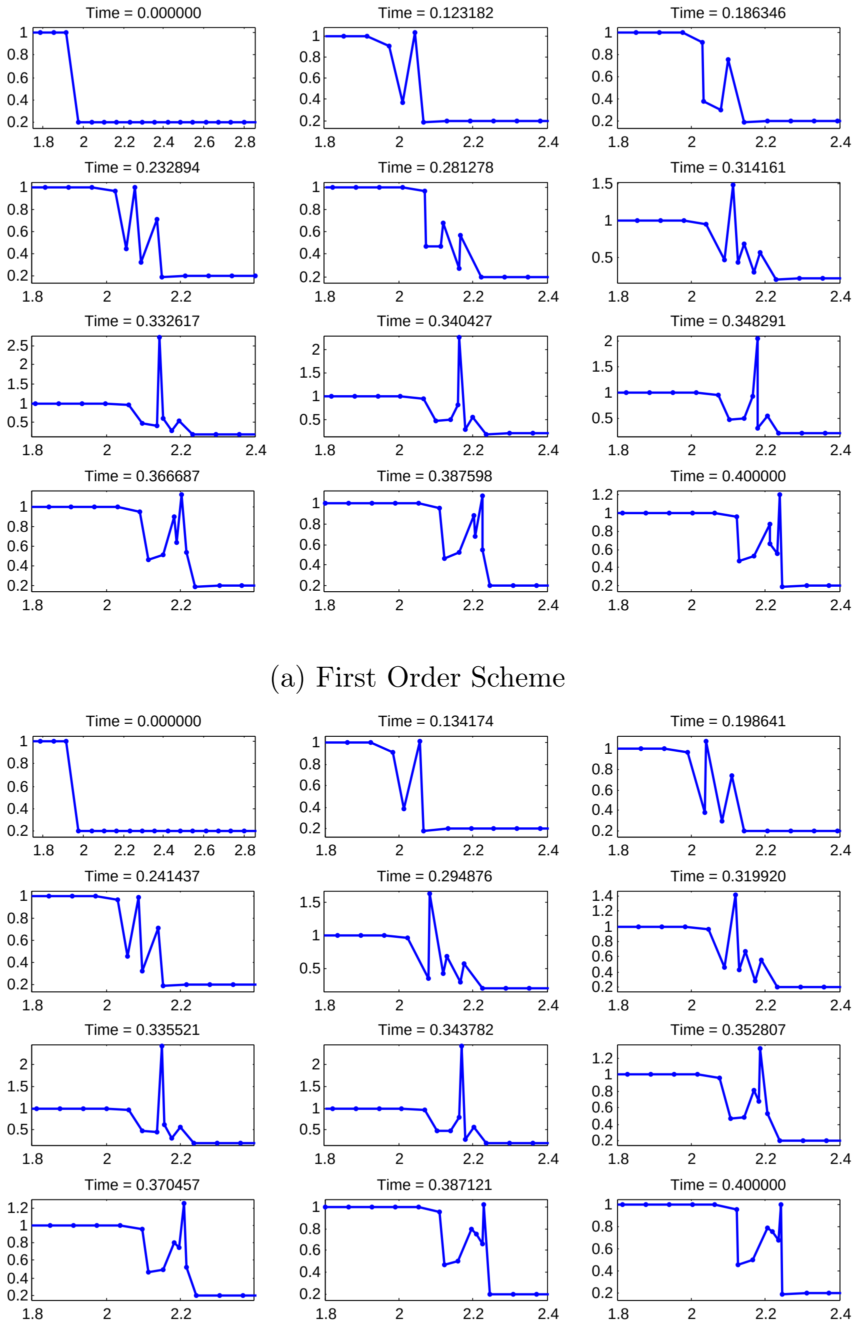
(a) Initial Condition: $u_0(x) = sin(x)$

(c) Initial Condition: $u_0(x) = sin(x)$

(b) Initial Condition: $u_0(x) = sin(x) + 0.2$

(d) Initial Condition: $u_0(x) = sin(x) + 0.2$

Figure 5.1: Evolution of Moving Mesh and the Numerical Solution on Moving Mesh with AREPO scheme when mesh moves with local shock speed

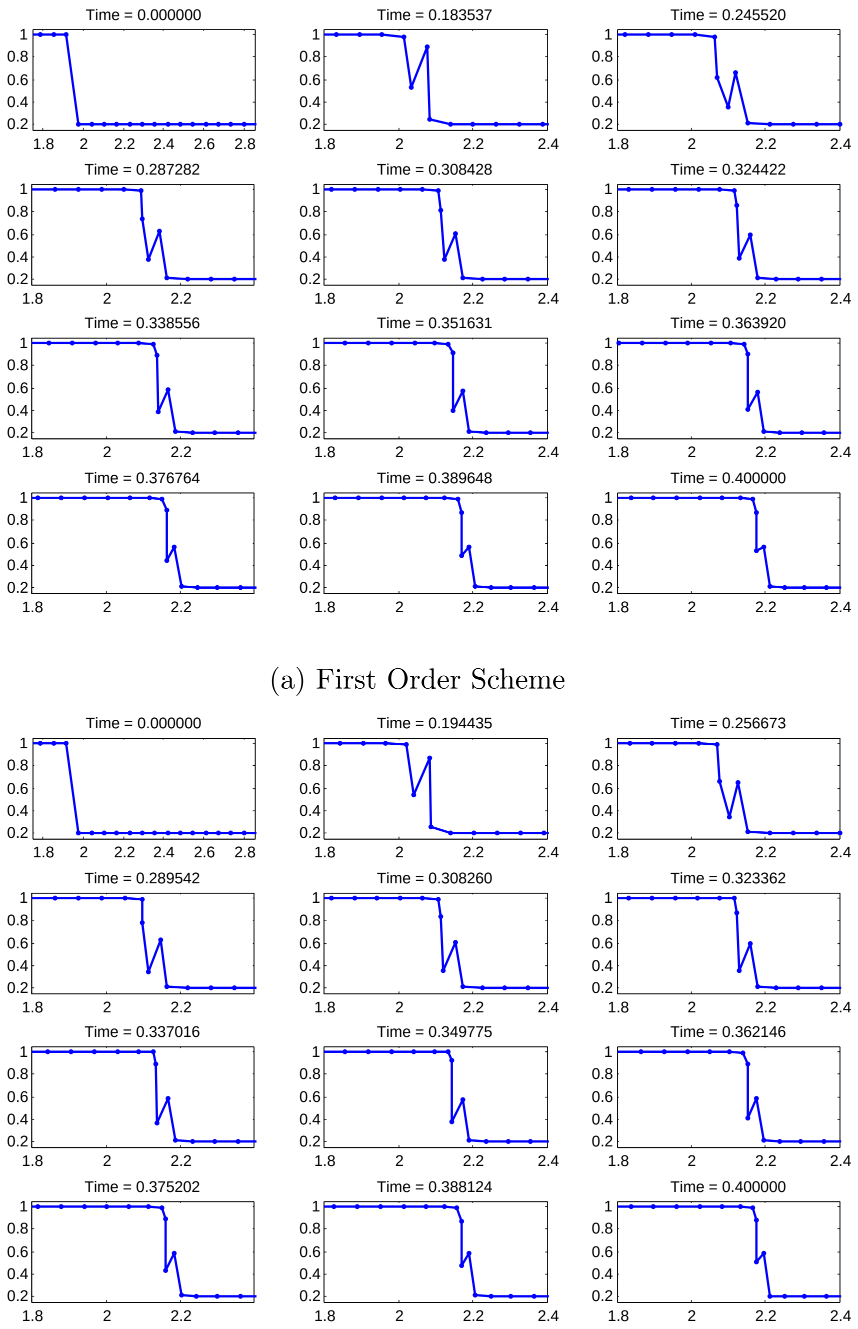


(a) First Order Scheme

(b) Second Order Scheme

Figure 5.2: Shock propagation with AREPO scheme(α=0.0): Solution in Y-axis and Grid on X-axis

Since the degree of movement of node towards the cell centre is zero, points near the shock cross each other. Lot of oscillation can be seen in the solution near the shock region.

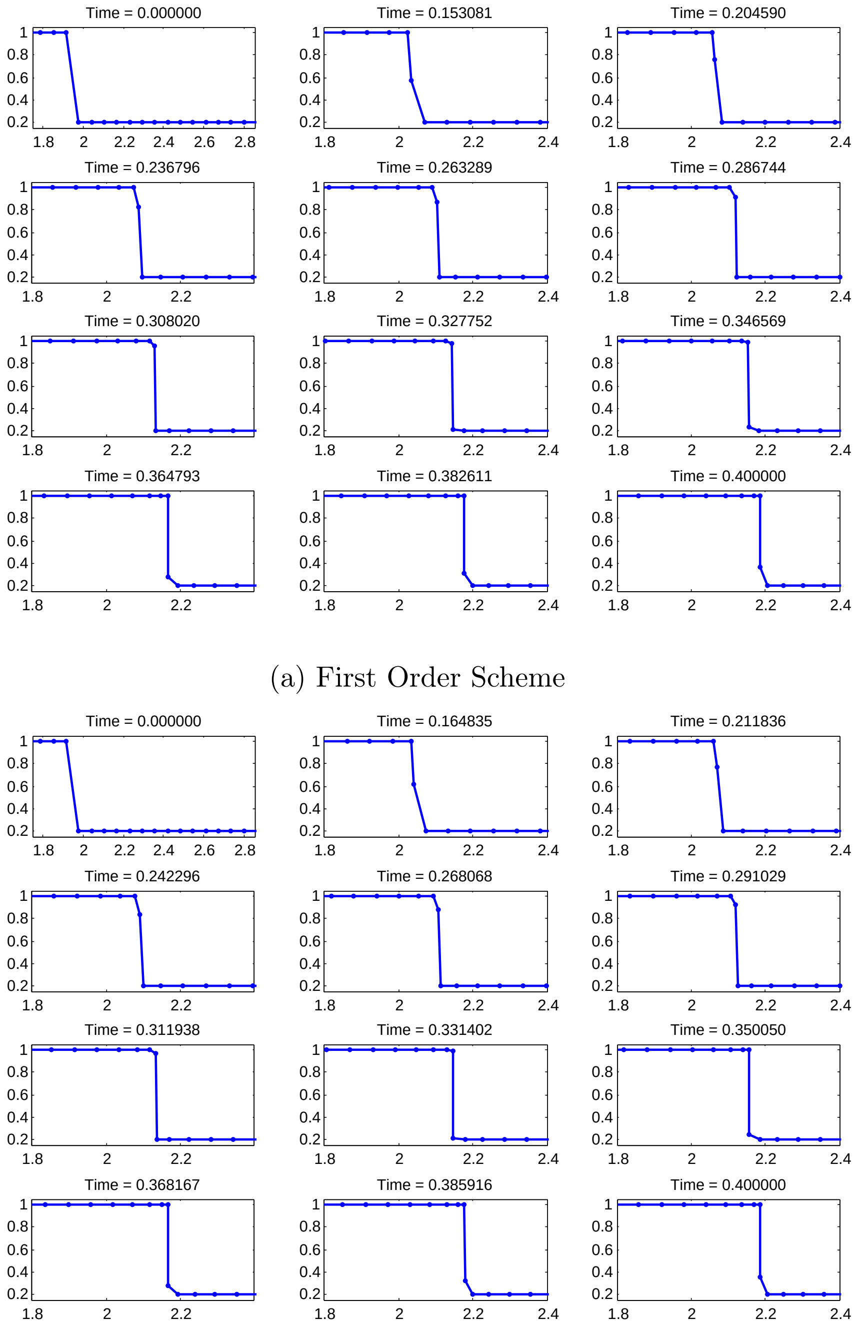


(a) First Order Scheme

(b) Second Order Scheme

Figure 5.3: Shock propagation with AREPO scheme(α=0.1): Solution in Y-axis and Grid on X-axis

Since the node is moved slightly towards the cell centre, oscillations are controlled to some extent. Points near the shock cross each other near the shock region.

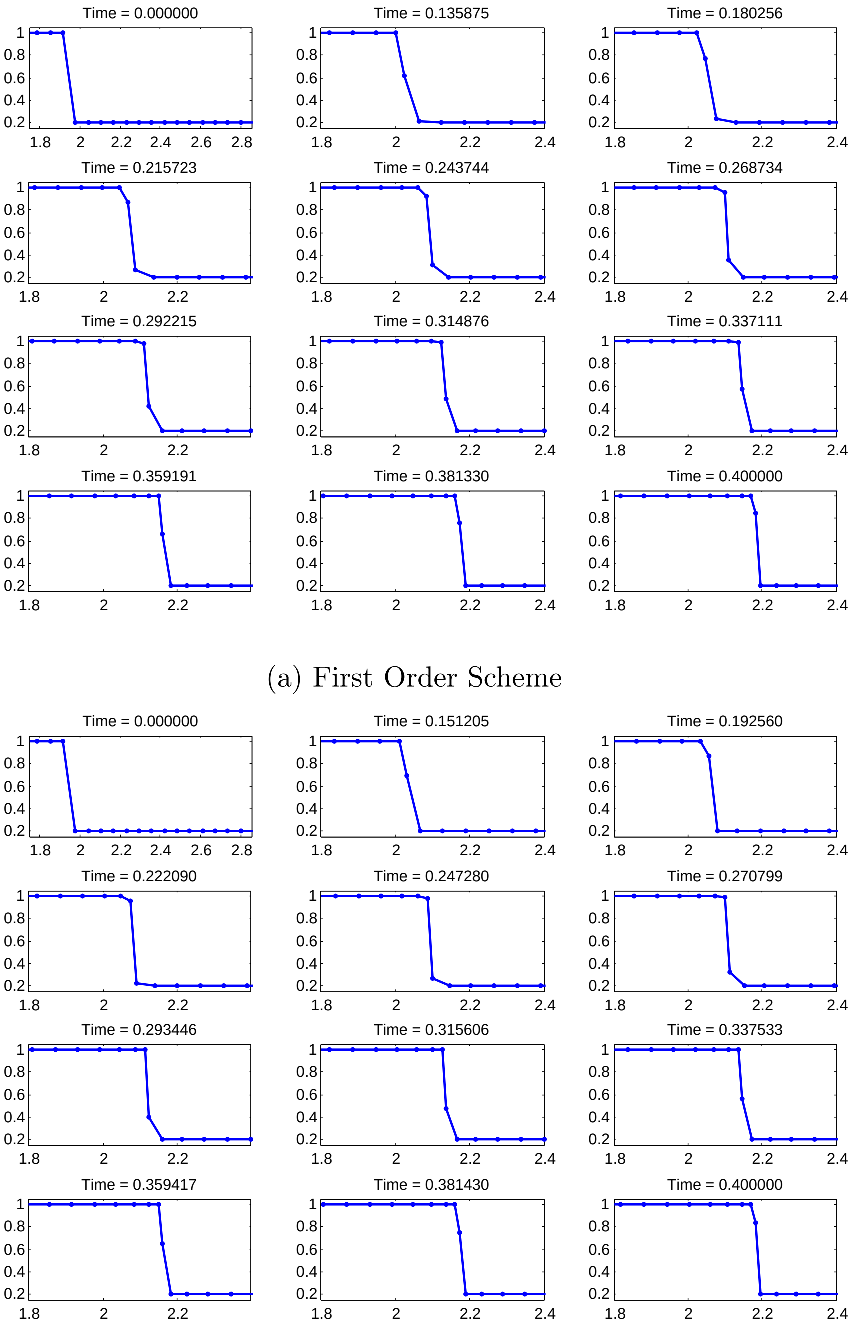


(a) First Order Scheme

(b) Second Order Scheme

Figure 5.4: Shock propagation with AREPO scheme(α=0.2): Solution in Y-axis and Grid on X-axis

Degree of movement of nodes towards the respective cell-centers is high and thus there is no crossing of nodes. Oscillations in the solution are not seen anymore.

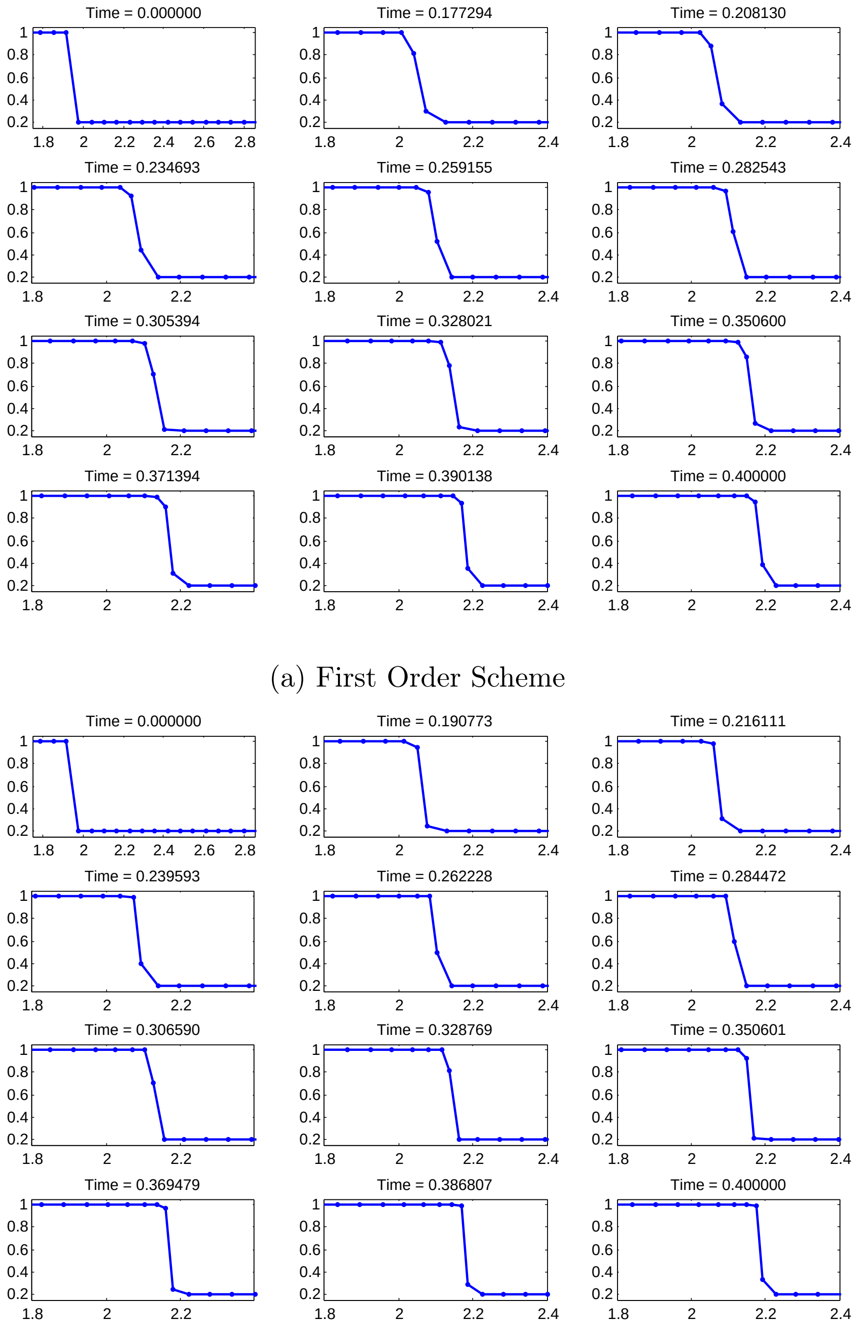


(a) First Order Scheme

(b) Second Order Scheme

Figure 5.5: Shock propagation with AREPO scheme(α=0.3): Solution in Y-axis and Grid on X-axis

Degree of movement of nodes towards the respective cell-centers is high and hence nodes do not cross one another. There are no oscillations in the solution but the solution seems to be slightly diffused near the shock.

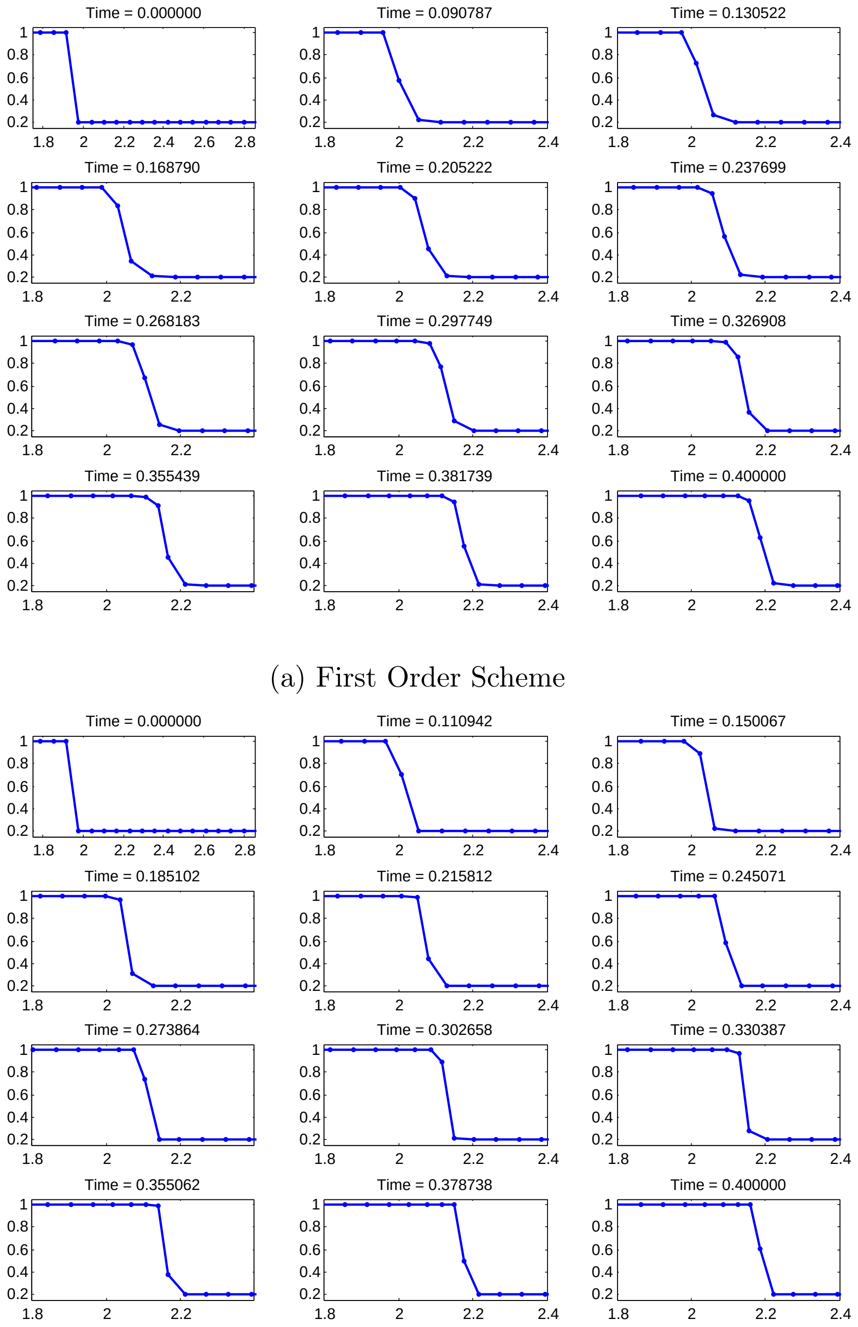


(a) First Order Scheme

(b) Second Order Scheme

Figure 5.6: Shock propagation with AREPO scheme(α=0.4): Solution in Y-axis and Grid on X-axis

Nodes do not cross one another as the degree of movement of nodes towards the respective cell-centers is very high. There are no oscillations in the solution but there is pronounced diffusion in the solution near the shock.



(a) First Order Scheme

(b) Second Order Scheme

Figure 5.7: Shock propagation with AREPO scheme(α=0.5): Solution in Y-axis and Grid on X-axis

Nodes are taken to be at the respective cell-centers and thus crossing of nodes is totally avoided. There are no oscillations in the solution but the degree of diffusion in the solution is very high.

5.5 Conclusion

Looking at $fig.$ 5.2,$fig.$ 5.3,$fig.$ 5.4,$fig.$ 5.5,$fig.$ 5.6,$fig.$ 5.7 it can be seen that AREPO moving mesh scheme effectively handles the shock waves when the local shock speed is used for mesh movement and the nodes are properly moved towards the cell-centres. When α, the degree of movement of node towards the cell-centre, is zero, then the nodes cross each other and there are lot of oscillations in the solution near the shock. Increase in α results in reduction in number of nodes crossing one another. Increase in α also decreases the oscillations in the solution. However, it increases the degree of diffusion in the solution near the shock. So in case of $\alpha = 0.5$, which corresponds to node being at the cell-centres, although there are no nodes crossing and there are no oscillation, but the solution is very diffused near the shock. So, some intermediate value of α seems to do the best job overall. Out of the values of α tried out, α=0.2 seems to do better than all others since the there are no oscillations in this case and the solution is is not very diffused near the shock.

Appendices

Appendix A

Runge-Kutta Scheme

Runge-Kutta schemes are used for updating the solution at the next iteration given the current value. First order Runge-Kutta scheme is also known as Forward Euler scheme. Runge-Kutta Scheme of higher orders are more accurate.
Consider solving the equation,

$$\frac{du}{dt} = L(u)$$

First, second and third order schemes for updating the solution are described below.

RK1 Scheme: This is first order accurate scheme.

$$\begin{aligned} u^{(0)} &= u^n \\ u^{(1)} &= u^n + \delta t L(u^{(0)}) \\ u^{n+1} &= u^{(1)} \end{aligned}$$

RK2 Scheme: This is second order accurate scheme

$$\begin{aligned} u^{(0)} &= u^n \\ u^{(1)} &= u^n + \delta t L(u^{(0)}) \\ u^{(2)} &= \frac{1}{2}u^n + \frac{1}{2}\left(u^{(1)} + \delta t L(u^1)\right) \\ u^{n+1} &= u^{(2)} \end{aligned}$$

RK3 Scheme: This is third order accurate scheme

$$\begin{aligned} u^{(0)} &= u^n \\ u^{(1)} &= u^n + \delta t L(u^{(0)}) \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}\left(u^{(1)} + \delta t L(u^{(1)})\right) \\ u^{(3)} &= \frac{1}{3}u^n + \frac{2}{3}\left(u^{(2)} + \delta t L(u^{(2)})\right) \\ u^{n+1} &= u^{(3)} \end{aligned}$$

Bibliography

[1] Volker Springel *Galiliean-invariant cosmological hydrodynamical simulations on a moving mesh* Mon. Not. R. Astron. Soc. (31 October, 2009)

[2] Marsha Berger, Michael J. Aftosmis, Scott M. Murman *Analysis of Slope Limiters on Irregular Grids.* American Institute of Aeronautics and Austronautics (May, 2005): NAS-05-007

[3] Xianyi Zeng *A General Approach to enhence slope limiters on non-uniform grids.* American Mathematical Society (6 Jan, 2013): 1301.0967

[4] J.A. Mackenzie *Moving Mesh Finite Volume Methods for One-Dimensional Evolutionary Partial Differential Equations*

[5] Eleuterio F. Toro *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* Springer

[6] Randall J. LeVeque *Numerical Methods for Conservation Laws* Birkhauser

[7] John M. Stockie, John A. Mackenzie, Robert D. Russell *A Moving Mesh method for one dimensional hyperblic Conservation Laws* American Mathematical Society. (1999)