REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPARISON OF HIGH RESOLUTION METHODS FOR BURGERS EQUATION

VELİ ÇOLAK

MSc. THESIS
DEPARTMENT OF MATHEMATICS

ADVISER
ASSOC. PROF. DR. SAMET YÜCEL KADIOĞLU

İSTANBUL, 2014

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPARISON OF HIGH RESOLUTION METHODS FOR BURGERS EQUATION

A thesis submitted by Veli ÇOLAK in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 24.06.2014 in Department of Mathematics.

Thesis Adviser
Assoc. Prof. Dr. Samet Yücel KADIOĞLU
Yıldız Technical University

Approved By the Examining Committee
Assoc. Prof. Dr. Samet Yücel KADIOĞLU
Yıldız Technical University _____

Assoc. Prof. Dr. Nuran GÜZEL, Member
Yıldız Technical University _____

Assist. Prof. Dr. Coşkun GÜLER, Member
Yıldız Technical University _____

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grant No: 2210.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Assoc. Prof. Dr. Samet Y. KADIOĞLU for his patience and support during the preparation of my thesis. I also would like to thank my all friends that encourage me to become academician.

Finally I would like to my all family, especially my wife Kübra, for their endless supports, love and for believing me to achieve.

June, 2014

Veli ÇOLAK

TABLE OF CONTENTS

page

LIST OF SYMBOLS

$v(x,t)$	Vector-valued function
u_i^n	Discrete value of $v(x,t)$ at $(i\Delta x, n\Delta t)$
$O(\Delta t)$	Big *O*-notation, order of Δt
τ^n	Truncation error
$\tilde{v}(w,t)$	Fourier transform of $v(x,t)$
$F_{i-1/2}$	Flux function at the point $x_{i-1/2}$
$v^{Rim}(u_{i-1}^n, u_i^n)$	Solution of Riemann problem at the point $x_{i-1/2}$
C_i	Cell or interval $(x_{i-1/2}, x_{i+1/2})$
σ_i	Slope defined on C_i
$\Delta u_{i-1/2}^n$	Difference between u_i^n and u_{i-1}^n $(u_i^n - u_{i-1}^n)$
$\delta_{i-1/2}^n$	Limited version of $\Delta u_{i-1/2}^n$
$A^+ \Delta u_{i-1/2}^n$	Net effect of right going waves from u_{i-1}^n

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABSTRACT

COMPARISON OF HIGH RESOLUTION METHODS FOR BURGERS EQUATION

Veli ÇOLAK

Department of Matematics

MSc. Thesis

Adviser: Assoc. Prof. Dr. Samet Yücel KADIOĞLU

Solving hyperbolic partial differential equations is extremely important for many engineering applications. These equations can be solved by analytical methods or numerical methods. Finding analytical solutions is difficult mostly impossible due to highly nonlinear nature of these equation types. On the other hand, solving hyperbolic equations numerically is relatively easy and therefore often prefered technique. Among the numerical techniques, high resolution finite volume methods have been effectively and robustly used for decades. Their high accuracy and stability features are most desirable.

In this thesis, we provide literature review for certain type of high resolution methods and introduce head on comparison study of these high resolution methods. To compare the methods, we first solve scalar linear one-way wave equation. This gives us the opportunity to perform some theoretical analysis. Finally, we apply the methods to the Burgers equation. This equation is nonlinear and it can be mimic the nonlinear behavior s of the systems of the nonlinear hyperbolic equations. For instance, the Burgers equationcan accommodate shock compression or rarefaction-depression waves. Therefore, by solving the Burgers equation with different methods, we gain a lot of insights about the stability, accuracy and thus the suitibility of the certain high resolution methods.

Key words: High resolution method, finite volume method, hyperbolic partial differential equations, Burgers equation

ÖZET

YÜKSEK ÇÖZÜNÜRLÜK YÖNTEMLERİNİN BURGERS DENKLEMİ ÜZERİNDE KARŞILAŞTIRILMASI

Veli ÇOLAK

Matematik Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Samet Yücel KADIOĞLU

Hiperbolik kısmi türevli diferansiyel denklemlerin çözümü bir çok mühendislik uygulamaları için çok büyük öneme sahiptir. Bu denklemler analitik ya da nümerik yöntemler kullanılarak çözülebilir. Analitik yöntemler ile çözmek denklemlerin nonlineer doğasından dolayı çoğu zaman zordur, hatta bazı denklemlerin analitik çözümü olmadığından dolayı imkansızdır. Analitik yöntemlere kıyasla hiperbolik denklemleri nümerik yöntemler ile çözmek daha kolaydır ve çoğunlukla tercih edilen yöntemdir. Numerik yöntemler arasonda yüksek çözünürlük sonlu hacimler yöntemi etkili ve sağlam bir şekilde onyıllardır kullanılmaktadır. Bu denklemlerin çok etkili bir şekilde kullanılmasının sebebi, yüksek doğruluk ve kararlılığa sahip olmalarıdır.

Bu tezde, belirlediğimiz yüksek çözünürlük metotlar ile ilgili geniş bir literatür taraması vereceğiz ve daha sonra bu metotları karşılaştıracağız. Karşılaştırma yapmak için önce skaler, lineer uzaysal olarak tek değişkenli dalga denklemini kullanacağız. Bu denklem bize metotların teorik analizlerini yapma fırsatı verecek. Daha sonra yöntemleri Burgers denklemine uygulayacağız. Bu denklemin özelliği nonlineer olması ve nonlinear hiperbolik denklem sistemlerinin özelliklerini taşımasıdır. Örneğin, Burgers denklemi şok yoğunlaşma dalgasını veya seyreltme dalgasını bünyesinde taşır. Bundan dolayı, Burgers denklemini farklı metotlarla çözerek, bu metotların kararlılığı, doğruluğu ve dolayısıyla uygunluğu hakkında fikir sahibi olacağız.

Anahtar Kelimeler: Yüksek çözünürlük metotları, sonlu hacim metodu, hiperbolik kısmi türevli diferansiyel denklemler, Burgers denklemi

CHAPTER 1

INTRODUCTION

1.1 Literature Review

Hyperbolic equations are one of the most significant equations class in partial differential equations. The study of gas dynamics, optics, geophysics, acoustics and many other fields involves solving hyperbolic partial differential equations, such as Euler equations. Solving these kinds of equation is challenging because their solution often contains shock or contact discontinuities. Accurately solving shock or contact phenomena can be extremely important in many engineering applications. Solutions mostly performed as numerical techniques, due to the highly nonlinear nature of equations set. To test the validity and reliability of these techniques, one usually considers simpler models such as scalar linear and nonlinear wave equations (one-way wave equation and Burgers equation [1]). Considering simple linear wave equation models can provide some theoretical insights such as stability and convergence analysis. On the other hand, the Burgers equation possesses the fundamental characteristics of the nonlinear hyperbolic systems in the sense that it can accommodate shock discontinuities. Most often, if a numerical method fails to solve the Burgers equation accurately and stably, then it also fails for other hyperbolic partial differential equation models.

Hyperbolic partial differential equations often can be interpreted as the physical conservation laws equations which model conservation of mass, energy and momentum [2]. Writing hyperbolic equations in conservation laws format is significant from the numerical methods perspective, because as it will be given with details that a numerical technique derived from conservation laws can be more stable, more accurate as well as better physics capturing.

Most of the numerical methods are not suitable for solving hyperbolic conservation laws. For example, one of the most popular numerical methods, finite difference method, can fail dramatically for these kinds of equations, since they rely on the differencing spatial derivatives and this should be avoided when solving shocks, etc. Thus, one has to consider specific numerical methods such as discontinuous Galerkin or finite volume methods when dealing with discontinuous phenomena. Finite volume methods are widely used and proven robust; therefore they will be our preferred method in this thesis.

The main idea behind the finite volume method is that it divides the spatial domain into grid cells and tries to approximate the average value of function (representing the conserved quantity) over each of these grid cells. Then for each time step, it updates the average amount of quantity according to the calculated fluxes that enter and leave from the cell edges/faces. Godunov is one of the pioneers who accurately and stably calculated discontinuous solutions by introducing the fundamental principles of finite volume method. In 1959, Godunov developed a new approach to this problem [3]. He gave an algorithm that consists of 3 steps. First step is to reconstruct a piecewise polynomial function according to the cell average in place of the initial data for each cell, second step is to evolve the hyperbolic equation to find the state of the function for next time step, and the last step is to average the function for each interval to find the new cell value. This approach has become one of the fundamental approaches for the construction of finite volume methods [1], [4].

Although Godunov's approach is fundamental, it is first-order accurate and it introduces numerical diffusion. After Godunov's method, in 1960, Peter Lax and Burton Wendroff developed a second-order method that based on the Taylor series expansion and central difference approximation [5]. A similar method to Lax-Wendroff was built by Fromm in 1968 [6] and by Warming and Beam in 1975 [7]. Although these methods have second-order accuracy for smooth region, they produce spurious oscillations around discontinuities.

Researches have been continued to tackle with the oscillation problems of second-order methods and various new shock-capturing methods have been developed. These methods generally called as high resolution methods [8]. Bram van Leer introduced the slope-limiter notion and improved the available numerical methods in a series of papers [9], [10], [11], [12], [13]. He reconsidered the oscillatory methods by introducing slope

notion that ultimately helped to eliminate oscillations. In particular, he introduced van Leer slope-limiter method [10] and MC slope-limiter method [12]. These are excellent methods compared to the classical second-order accurate methods in a way that they kill the numerical oscillations of the methods around discontinuity. In 1985, Roe contributed a new remarkable scheme called as superbee slope-limiter method [14]. Later, Sweby has found a new approach named as flux limiter [15]. Like slope limiter concept, flux limiter concept also enabled high resolution methods to be rewritten in a new and understandable way. This situation evoked a lot of new methods [1].

While new methods were discovered, there have also been studies that compared them. In [15], Sweby compared methods including superbee and van Leer, but he just investigated that whether the methods was satisfying the TVD condition or not. Another comparative study was done by Farthing and Miller [16]. They worked some high resolution methods in terms of order of accuracy and time efficiency. They used several test cases, but all of them consisted of the linear equations. Yang and Przekwas did a very nice study that classed with a lot of advanced shock-capturing methods [8]. They used the Burgers equation as a test problem with two different initial conditions. There are also many other comparative study for varied methods in term of different aspects of methods [17], [18], [19], [20].

1.2 Objective of the Thesis

In this thesis, we provide a comparative study of high resolution methods for the Burgers equation. In particular, we compare

- Upwind method,
- Lax-Wendroff method,
- Beam-Warming method,
- Fromm's method
- Minmod slope-limiter method,
- Superbee slope-limiter method,
- Van leer slope-limiter method,
- MC slope-limiter method

and show the advantages and drawbacks of these methods. We apply these numerical schemes to linear wave equation first to demonstrate the basic features. However,

applying the methods to the Burgers equation, we also show the true strength and weakness of the methods.

1.3 Hypothesis

We will give the basic necessary theoretical information in chapter 2, e.g. stability, consistency and convergence. In chapter 3, we will introduce finite volume method and other necessary concepts for high resolution methods. For the next chapter, we will describe the numerical methods clearly and apply these methods to the one-way wave equation. We will compare the methods in this chapter for scalar linear hyperbolic partial differential equations. We will try to explore the basic feature of the methods. In chapter 5, we apply the methods, given in chapter 4, to the Burgers equation and give the results in graphical forms and compare the methods to understand the behavior of the methods for nonlinear hyperbolic conservation laws equations. In the last chapter, we will summarize the results. We will use the FORTRAN as a programming language and MATLAB for figures.

CHAPTER 2

SOME THEORETICAL BASIS

Partial differential equations are large area of study. The tool to solve these equations includes components in the areas of mathematics, computers and physical applications. These three aspects of problem cannot be separated each other. When solving a problem, one cannot consider application aspect without the others. Sometimes, the mathematical aspect of a hyperbolic partial differential equation can be developed without taking application and computing into account, but experiences demonstrate that this way of studying does not generally yield useful consequences [21]. Therefore we first give some theoretical knowledge, after that we convert analytical hyperbolic problems to numerical problems via high resolution methods, and lastly we give results of application of methods to the problems.

In this chapter, we will see the definition of convergence, consistency and stability. For user of numerical methods, it is essential to understand what type of convergence their methods have and what are the assumptions to get this convergence. There is a relationship along the convergence, consistency and stability. The Lax Theorem says roughly that if the scheme is stable and consistent than it is convergent [21]. This theorem is very useful because it is easier to prove consistency and stability than to prove convergence.

2.1 Convergence

In one space dimension, a homogeneous first-order constant-coefficient linear hyperbolic partial differential equation in x and t has the form

$$v_t(x,t) + cv_x(x,t) = 0, \tag{2.1}$$

and the initial condition is

$$v(x,0) = f(x). \tag{2.2}$$

Here v represents the unknown function and it depends on time as well as one spatial variable. This function may be velocity, pressure, density etc. In the notation, subscript x and t denote the partial derivatives with respect to space and time respectively.

We will try to solve this problem numerically. For this purpose, we should reduce problem to a discrete problem. To accomplish this, we can use the following approximation,

$$v_t(n\Delta t, i\Delta x) \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}. \tag{2.3}$$

Here we assume the expression u_i^n denote the approximate solution to v at the point $x = i\Delta x$ and $t = n\Delta t$, n corresponds to time step and i to the spatial mesh point. Furthermore, Δt and Δx are grid steps with respect to time and space, respectively.

Using (2.3) in (2.1) we get

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_i^n}{\Delta x} = 0. \tag{2.4}$$

And then the initial condition in (2.2) becomes

$$u_i^0 = f(k\Delta x),\ i \text{ is from } -\infty \text{ to } \infty. \tag{2.5}$$

The idea behind any finite difference scheme is to approximate the solution of differential equation. Now, let investigate how good scheme (2.4)-(2.5) is for approximating the solution to problem (2.1)-(2.2). To achieve this, firstly, let us look how well difference equation (2.4) approximates partial differential equation (2.1). To examine this, we will use Taylor series expansion.

$$\begin{aligned} v_i^{n+1} &= v(\mathrm{i}\,\Delta x, (\mathrm{n}+1)\Delta t) = \mathrm{v}(\mathrm{i}\,\Delta x, \mathrm{n}\,\Delta t) + \frac{\partial v}{\partial t}(\mathrm{i}\,\Delta x, \mathrm{n}\,\Delta t)\frac{\Delta t}{1!} \\ &+ \frac{\partial^2 v}{\partial t^2}(\mathrm{i}\,\Delta x, \mathrm{n}\,\Delta t)\frac{\Delta t^2}{2!} + \ldots \end{aligned}, \tag{2.6}$$

So

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = \frac{\partial v}{\partial t}(i\Delta x, n\Delta t) + \frac{\Delta t}{2}\frac{\partial^2 v}{\partial t^2}(i\Delta x, n\Delta t) + \ldots. \tag{2.7}$$

We can also write this expression as

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = \frac{\partial v}{\partial t}(i\Delta x, n\Delta t) + O(\Delta t), \tag{2.8}$$

where the above equation assumes that the higher order derivatives of v at $(i\Delta x, n\Delta t)$ are bounded. Here, the notation $O(\Delta t)$ means that $f(\mathrm{x}) = \mathrm{O}(\phi(\mathrm{x}))$ for $x \in D$ if there exists a constant A such that $\left|f(x)\right| \le A\left|\phi(x)\right|$ for all $x \in D$ (D is the domain of function f). We say that $f(x)$ is of order $\phi(x)$. We can conclude from the above expression that when we replace v_t in the partial differential equation by $\frac{u_i^{n+1} - u_i^n}{\Delta t}$ we ignore some terms of order Δt. Note that sometimes ignored term (denoted by $O(\Delta t)$) can be very large. For instance, when solving problems that have sharp changes with respect to time, ignored term will be huge. However, in general, for sufficiently small Δt, $\frac{u_i^{n+1} - u_i^n}{\Delta t}$ is a nice approximation to v_t, and we can provide sufficiently small Δt.

We can also arrive the following results using the approach in (2.8).

$$\frac{v_{i+1}^n - v_i^n}{\Delta x} = \frac{\partial v}{\partial x}(\mathrm{i}\,\Delta x, n\Delta t) + O(\Delta x), \tag{2.9}$$

$$\frac{v_i^n - v_{i-1}^n}{\Delta x} = \frac{\partial v}{\partial x}(\mathrm{i}\,\Delta x, n\Delta t) + O(\Delta x), \tag{2.10}$$

and

$$\frac{v_{i+1}^n - v_{i-1}^n}{2\Delta x} = \frac{\partial v}{\partial x}(\mathrm{i}\,\Delta x, n\Delta t) + O(\Delta x^2). \tag{2.11}$$

Returning to the equation (2.1), we can write

$$v_t(\mathrm{i}\,\Delta x, n\Delta t) + \mathrm{cv}_x(\mathrm{i}\,\Delta x, n\Delta t) = \frac{v_i^{n+1} - v_i^n}{\Delta t} + c\frac{v_{i+1}^n - v_i^n}{\Delta x} + O(\Delta t) + O(\Delta x). \tag{2.12}$$

Therefore, we see that difference equation (2.4) approximates partial differential equation (2.1) to the first order in both Δt and Δx.

Equation (2.12) shows us how good the difference equation approximates the partial differential equation. However this does not mean that the solution of difference equation will approximate the solution of partial differential equation. Thus, still there is

an issue that we must consider. At this stage we can argue that the solution of difference scheme will generally approximate the solution of partial differential equation at the same order that the approximation of difference scheme to the differential equation.

We need exactly that the solution of the difference equation can be made to approach the solution of the partial differential equation to any desired accuracy. Therefore we require convergence of solution of the finite difference equation to the solution of the partial differential equation. Now, let consider a partial differential equation, say, $Lv = F$. Here F and v are vector-valued functions that define on the whole real line in terms of first variable (spatial variable) and initial condition $v(x,0) = f(x)$. Let u_i^n be the approximate solution to v. u_i^n is defined on a grid with grid steps Δx and Δt, satisfies the initial condition $u_i^0 = f(i\Delta x)$, where i is from $-\infty$ to ∞. Let v denote the analytic solution to initial-value problem. Then the definition of pointwise convergence is the following.

Definition 2.1 A difference scheme $L_i^n u_i^n = G_i^n$ approximating the partial differential equation $Lv = F$ is a pointwise convergent scheme if for any x and t, as $(i\Delta x,(n+1)\Delta t)$ converges to (x,t), u_i^n converges to $v(x,t)$ as Δx and Δt converges to *0* [21].

To clarify the definition, we solve an example.

Example 2.1 Show that the solution of the different scheme

$$u_i^{n+1} = (1-cfl)u_i^n + cflu_{i-1}^n, \tag{2.13}$$

$$u_i^0 = f(i\Delta x), \tag{2.14}$$

where $cfl = \frac{c\Delta t}{\Delta x}$, $0 < cfl \le 1$, converges pointwise to the solution of the initial-value problem

$$v_t + cv_x = 0, \quad x \in \mathbb{R}, \quad t > 0, \tag{2.15}$$

$$v(x,0) = f(x), \quad x \in \mathbb{R}. \tag{2.16}$$

Solution: Since the problem is initial-value problem on all of $\mathbb{R}$, we have to be aware of the fact that the i index on u_i^n span the whole real line.

Let $v = v(x,t)$ denote the exact solution of initial value problem (2.15)-(2.16) and let z_i^n denote the difference between the analytic solution and numerical solution at the point $(i\Delta x, n\Delta t)$. That is,

$$z_i^n = u_i^n - v(i\Delta x, n\Delta t). \tag{2.17}$$

From (2.8), we know that

$$v_t = \frac{v_i^{n+1} - v_i^n}{\Delta t} - O(\Delta t). \tag{2.18}$$

Using again the Taylor series expansion, we can get

$$\begin{aligned} v_{i-1}^n = v((\mathrm{i}-1)\Delta x, \mathrm{n}\,\Delta t) = \mathrm{v}(\mathrm{i}\,\Delta x, n\Delta t) - \frac{\partial v}{\partial x}(\mathrm{i}\,\Delta x, n\Delta t)\frac{\Delta x}{1!} \\ + \frac{\partial^2 v}{\partial x^2}(i\Delta x, n\Delta t)\frac{\Delta x^2}{2!} + \ldots \end{aligned}. \tag{2.19}$$

Therefore,

$$v_x(i\Delta x, n\Delta t) = \frac{v_i^n - v_{i-1}^n}{\Delta x} + O(\Delta x). \tag{2.20}$$

From (2.15), (2.19) and (2.20) we conclude that

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} - O(\Delta t) + \mathrm{c}\frac{v_i^n - v_{i-1}^n}{\Delta x} + O(\Delta x) = 0, \tag{2.21}$$

$$v_i^{n+1} = (1 - cfl)v_i^n + cfl\,v_{i-1}^n + O(\Delta t^2) - O(\Delta x \Delta t). \tag{2.22}$$

Then by subtracting equation (2.22) from equation (2.13) we see that z_i^n satisfies

$$z_i^{n+1} = (1 - cfl)z_i^n + z_{i-1}^n + O(\Delta t^2) - O(\Delta x \Delta t). \tag{2.23}$$

Because of $0 < cfl \le 1$, the coefficients of variable in equation (2.23) are non-negative and

$$\left|z_i^{n+1}\right| \le (1 - cfl)\left|z_i^n\right| + cfl\left|z_{i-1}^n\right| + K(\Delta t^2 - \Delta t \Delta x) \le \mathrm{Z}^n + K(\Delta t^2 - \Delta t \Delta x), \tag{2.24}$$

where K is a constant related with the "big *O*" term and assumed to be bounded, and $Z^n = \sup_i\left\{\left|z_i^n\right|\right\}$. Taking the supremum of $\left|z_i^{n+1}\right|$ over *i*, we arrive

$$Z^{n+1} \le Z^n + K(\Delta t^2 - \Delta t \Delta x). \tag{2.25}$$

Applying (2.25) repeatedly

$$Z^{n+1} \le Z^n + K(\Delta t^2 - \Delta t\Delta x) \le Z^{n-1} + 2K(\Delta t^2 - \Delta t\Delta x) \le ... \\ \le Z^0 + (n+1)K(\Delta t^2 - \Delta t\Delta x) \quad . \tag{2.26}$$

Because of $Z^0 = 0, \quad Z^{n+1} \le (n+1)K(\Delta t^2 - \Delta t\Delta x)$

Thus, $\left|u_i^{n+1} - v(i\Delta x,(n+1)\Delta t\right| \le Z^{n+1}$ and (n+1)$\Delta t \to t$,

$$\left|u_i^{n+1} - v(i\Delta x,(n+1)\Delta t\right| \le (n+1)\Delta tK(\Delta t - \Delta x) \to 0 \text{ as } \Delta t, \Delta x \to 0 \tag{2.27}$$

which means that for any *x* and *t*, as $(i\Delta x,(n+1)\Delta t)$ approaches to (x,t), u_k^n converges to $v(x,t)$.

Be aware of the fact that the assumption $(n+1)\Delta t \to t$ is needed otherwise the term $(n+1)\Delta t$ can goes to the infinity. Furthermore, the assumption $0 < cfl \le 1$ in the question is necessary. We will see the details later that without this assumption, it may not converge. This assumption enables us to bound the time step size Δt. In fact, for this example $\Delta t \le \Delta x / c$. One more note for this example is about the remainder of Taylor series expansion. We assume that *K* is bounded. To achieve this the derivative of the solution function $v(x,t)$ in the remainder term in expansion should be uniformly bounded on $\mathbb{R} \times [0,t]$.

In general, the pointwise convergence is not generally as useful as a more uniform sort of convergence and is more complicated to prove. Because of this, we will give another definition of convergence which is defined in terms of norm of the difference between the solution to the difference equation and solution of partial differential equation. For the prerequisite, let denote the sup-norm on the space of all bounded sequences, l_∞ , by

$$\left\|\{\alpha_i\}\right\|_\infty = \sup_{-\infty<i<\infty} \left|\alpha_i\right|. \tag{2.28}$$

Let us define $\mathbf{u}^n = (\cdots, u_{-1}^n, u_0^n, u_1^n, \cdots)^T$ and $\mathbf{v}^n = (\cdots, v_{-1}^n, v_0^n, v_1^n, \cdots)^T$. Here $\mathbf{u}^n$ is the vector of difference equation solution values u_i^n, and $\mathbf{v}^n$ is the vector of solution to the partial differential equation $v(i\Delta x, n\Delta t)$. By the way, we have proved in previous example that

for t such that $(n+1)\Delta t$ converges t, $\mathbf{u}^{n+1}$ converges $v(\cdot,t)$ where we mean by convergence that the sup-norm of $\mathbf{u}^{n+1}-\mathbf{v}^{n+1}$ approaches zero as $\Delta t, \Delta x \to 0$.

Definition 2.2 A difference scheme $L_i^n u_i^n = G_i^n$ approximating the partial differential equation $Lv = F$ is a convergent scheme at time t if, as $(n+1)\Delta t \to t$,

$$\left\|\mathbf{u}^{n+1}-\mathbf{v}^{n+1}\right\| \to 0 \tag{2.29}$$

as $\Delta x, \Delta t \to 0$ [21].

To demonstrate this definition, let solve an example.

Example 2.2 Show that the solution of difference equation

$$u_i^{n+1} = \frac{1}{2}(u_{i+1}^n + u_{i-1}^n) - \frac{cfl}{2}(u_{i+1}^n - u_{i-1}^n), \tag{2.30}$$

$$u_i^0 = f(i\Delta x). \tag{2.31}$$

(The Lax-Friedrichs scheme) converges in the sup-norm to the solution of the partial differential equation

$$v_t + cv_x = 0, \tag{2.32}$$

$$v(x,0) = f(x) \tag{2.33}$$

for $|cfl| \le 1$ where $cfl = \dfrac{c\Delta t}{\Delta x}$.

Solution: Before showing the solution, we should note that it will be used the sup-norm to solve the problem. Let denote the analytic solution to the partial differential equation by v and define $z_i^n = u_i^n - v_i^n$ and $Z^n = \sup_i \left\{\left|z_i^n\right|\right\}$. We use the definition with the sup-norm, so

$$\left\|\mathbf{u}^{n+1}-\mathbf{v}^{n+1}\right\|_\infty = \sup_{-\infty<i<\infty}\left|u_i^{n+1}-v_i^{n+1}\right| = \sup_{-\infty<i<\infty}\left|z_i^{n+1}\right| = Z^{n+1}. \tag{2.34}$$

From the equation (2.32) and (2.11) we arrive

$$\frac{v_i^{n+1} - \frac{1}{2}\left(v_{i+1}^n + v_{i-1}^n\right)}{\Delta t} + c\frac{v_{i+1}^n - v_{i-1}^n}{2\Delta x} - O\left(\Delta x^2\right) - O\left(\Delta t\right) = 0. \tag{2.35}$$

Note that we replace v_i^n with average value $\frac{v_{i+1}^n + v_{i-1}^n}{2}$. Then

$$v_i^{n+1} = \frac{1}{2}\left(v_{i+1}^n + v_{i-1}^n\right) - \frac{c\Delta t}{2\Delta x}\left(v_{i+1}^n - v_{i-1}^n\right) + O\left(\Delta t \Delta x^2\right) + O\left(\Delta t^2\right) \tag{2.36}$$

Therefore,

$$z_i^{n+1} = \frac{1}{2}\left(z_{i+1}^n + z_{i-1}^n\right) - \frac{1}{2} cfl \left(z_{i+1}^n - z_{i-1}^n\right) + O\left(\Delta t \Delta x^2\right) + O\left(\Delta t^2\right). \tag{2.37}$$

Rearranging the equation (2.37),

$$z_i^{n+1} = \frac{1}{2} z_{i+1}^n \left(1 - cfl\right) + \frac{1}{2} z_{i-1}^n \left(1 + cfl\right) + O\left(\Delta t \Delta x^2\right) + O\left(\Delta t^2\right). \tag{2.38}$$

Since the value of *cfl* is between -1 and 1, and the coefficients of right hand side are non-negative, we get

$$\left|z_i^{n+1}\right| \le \frac{1}{2}\left|z_{i+1}^n\right|\left(1 - cfl\right) + \frac{1}{2}\left|z_{i-1}^n\right|\left(1 + cfl\right) + K\left(\Delta t \Delta x^2 + \Delta t^2\right). \tag{2.39}$$

Taking the supremum over *i* on the both side of equation yields

$$Z^{n+1} \le Z^n + K\left(\Delta t \Delta x^2 + \Delta t^2\right). \tag{2.40}$$

Applying (2.40) repeatedly yields

$$\begin{aligned} &Z^{n+1} \le Z^n + K\left(\Delta t \Delta x^2 + \Delta t^2\right) \le Z^{n-1} + 2K\left(\Delta t \Delta x^2 + \Delta t^2\right) \le \cdots \\ &\le Z^0 + \left(n+1\right) K\left(\Delta t \Delta x^2 + \Delta t^2\right). \end{aligned} \tag{2.41}$$

Because of $Z^0 = 0$, we conclude using (2.34) that

$$Z^{n+1} = \left\|\mathbf{u}^{n+1} - \mathbf{v}^{n+1}\right\|_\infty \le \left(n+1\right)\Delta t K\left(\Delta x^2 + \Delta t\right) \to 0 \tag{2.42}$$

as $\left(n+1\right)\Delta t \to t$ and $\Delta t, \Delta x \to 0$

This proves that solution to the difference equation converges in the sup-norm to the solution of partial differential equation using the definition 2.2.

2.2 Norms

In the definition 2.2, we did not specify the norm because for all norms the definition is valid. For solving the example 2.2, we used the sup-norm but for different situations, it

may be appropriate to use different norms. In fact, it may happen that a method is convergent in one norm but not in another [22]. Therefore we should know also some of the other norms. Here, we will mention about the 1-norm, 2-norm, energy norm, and sup-norm.

We already define the sup-norm in (2.28). Note that sup-norm may appropriate for continuous solution but for discontinuous solution, it is not a good idea to use sup-norm to approximate the solution. While the grid is refined, the pointwise error around a discontinuity does not go to zero uniformly which is an unwanted case. Despite of this situation, the numerical results may be superbly satisfactory. That is why; sup-norm should not be used for the conservation laws.

1-norm, in general, is the appropriate norm for the conservation laws. For a general function $v(x)$, it is defined as follows

$$\|v\|_1 = \int_{-\infty}^{\infty} |v(x)| dx . \tag{2.43}$$

This definition is for the continuous case and $v(x)$ is the continuous function. This norm is natural since it requires just integrating the function, and form of the conservation laws generally allows us to say something about these integrals. For the discrete case, we use the following definition,

$$\|v^n\| = \sum_{i=-\infty}^{\infty} |v_i^n| . \tag{2.44}$$

Continuous case and discrete case of 2-norm is as follows

$$\|v\|_2 = \left[\int_{-\infty}^{\infty} |v(x)|^2 dx \right]^{1/2} \tag{2.45}$$

$$\|v^n\|_2 = \sqrt{\sum_{i=-\infty}^{\infty} |v_i^n|^2} . \tag{2.46}$$

2-norm is a suitable norm for linear equation because for linear equations, Fourier analysis can be used and Parseval's relation states that the Fourier transform of $v(x)$ has the same 2-norm with $v(x)$. This enables to simplify the stability analysis of linear methods seriously. Note that here *v* is a vector in l_2 space which is defined as

$$l_2 = \left\{ \mathbf{v} = \left(\cdots, v_{-1}, v_0, v_1, \cdots \right)^T : \sum_{i=-\infty}^{\infty} \left| v_i \right|^2 < \infty \right\}. \tag{2.47}$$

Definitions of energy norm for continuous function $v(x)$ and discrete grid function v_k^n are

$$\left\| v \right\|_2 = \left[\Delta x \int_{-\infty}^{\infty} \left| v(x) \right|^2 dx \right]^{1/2} \tag{2.48}$$

and

$$\left\| v^n \right\|_2 = \sqrt{\sum_{i=-\infty}^{\infty} \left| v_i^n \right|^2 \Delta x}\,. \tag{2.49}$$

Energy norm defined above can be considered as more suitable than the 2-norm since it retains all of the favorable properties of the 2-norm. Beside this, energy norm enable to measure the difference between discretization of functions as Δx goes to zero.

2.3 Consistency

Even though our final aim is to prove convergence, this is difficult to achieve directly. Instead, we begin by examining the local truncation error so that examining the consistency and then use the stability of the method to prove the convergence.

Definition 2.3 The finite difference scheme $L_i^n u_i^n = G_i^n$ is pointwise consistent with the partial differential equation $Lv = F$ at point (x,t) if for any smooth function $\phi = \phi(x,t)$,

$$\left(L\phi - F \right)\Big|_i^n - \left[L_i^n \phi\left(i\Delta x, \mathrm{n}\,\Delta t \right) - G_i^n \right] \to 0 \tag{2.50}$$

as $\Delta x, \Delta t \to 0$ and $\left(i\Delta x, (n+1)\Delta t \right) \to (x,t)$ [21].

It should be noticed that in equation (2.12), we were actually prove the pointwise consistency of the difference scheme (2.4) to the differential equation (2.1). In equation (2.12), we chose ϕ to be the solution, v, to the differential equation. This choice, in general, enables the expression in the definition 2.3 to reduce the following form

$$L_i^n v_i^n - G_i^n \to 0 \text{ as } \Delta x, \Delta t \to 0\,. \tag{2.51}$$

If we write the different scheme as

$$\mathbf{u}^{n+1} = Q\mathbf{u}^n + \Delta t\mathbf{G}^n \tag{2.52}$$

where

$$\mathbf{u}^n = \left(\cdots, u_{-1}^n, u_0^n, u_1^n, \cdots\right)^T,\ \mathbf{G} = \left(\cdots, \mathrm{G}_{-1}^n, \mathrm{G}_0^n, \mathrm{G}_1^n, \cdots\right)^T$$

and Q is an operator acting on the suitable space, then we can give a stronger definition for consistency as follows.

Definition 2.4 The difference scheme (2.52) is consistent with the partial differential equation in a norm $\|\cdot\|$ if the solution of the partial differential equation, v, satisfies

$$\mathbf{v}^{n+1} = Q\mathbf{v}^n + \Delta t\mathbf{G}^n + \Delta t\boldsymbol{\tau}^n, \tag{2.53}$$

and

$$\left\|\tau^n\right\| \to 0 \tag{2.54}$$

as $\Delta x, \Delta t \to 0$, where $\mathbf{v}^n$ denote the vector whose ith component is $v\left(i\Delta x, n\Delta t\right)$ [21].

Note that, when we writing the difference scheme as (2.52) we assume that the scheme have only *nth* and *(n+1)st* time level and the partial differential equation is first order according to *t*. Furthermore the norm consistency defined in definition 2.4 says that all of the components of vector $\boldsymbol{\tau}^n$ must converge to zero while the pointwise consistency defined in definition 2.3 require that τ_i^n must converge to zero only for some *i*. Another note is that the truncation error stems from both the error due to the approximation of F and the approximation of L by L_i^n. In sometimes the error that stems from approximation of F can lower the order of the scheme.

Another definition of consistency which includes the term order of accuracy is the following.

Definition 2.5 The different scheme (2.52) is said to be accurate of order $\left(p, q\right)$ to the given partial differential equation if

$$\left\|\boldsymbol{\tau}^n\right\| = O\left(\Delta x^p\right) + O\left(\Delta t^q\right) \text{ [21].} \tag{2.55}$$

Here, we call $\|\tau^n\|$ or τ^n as the truncation error. We use this term before but we did not mention about it clearly. To understand the truncation error well, first let look at the local truncation error.

The local truncation error tells us suitability of difference equation to partial differential equation locally. That is, how well the difference equation models the partial differential equation locally. It is defined by replacing the numerical solution u_k^n in the difference equation by the analytic solution $v(x,t)$. Clearly, the analytic solution of partial differential equation is an approximate solution for difference equation, and how well it is suitable for the difference equation implies that how well the numerical solution of the difference equation is appropriate for the partial differential equation.

Note that the order condition given in definition 2.5 contains a constant, say *K*, with respect to Δx and Δt. The constant *K*, in general, depends on *t*. It is not depend on *x* because τ^n is the truncation error for all $i \in \mathbb{R}$. To know this issue provides ease.

Remember that beginning of this chapter we applied the Taylor series expansion to the solution of partial differential equation and then we got a remainder term in addition to difference scheme. After that, we said the difference scheme approximates the partial differential equation to the first order in both Δt and Δx. This is also a demonstration of consistency in roughly. Now we will show the consistency of the same partial differential equation with using the definition 2.4-2.5.

Example 2.3 Discuss the consistency of the 2-level difference scheme

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+c\frac{u_{i+1}^n-u_i^n}{\Delta x}=0 \tag{2.56}$$

with partial differential equation

$$v_t+cv_x=0,\ -\infty<x<\infty,\ t>0. \tag{2.57}$$

Solution: Let denote the solution of partial differential equation by *v*, and put it into the difference equation (2.56). We see after the cancelation that

$$\frac{v_i^{n+1}-v_i^n}{\Delta t}+c\frac{v_{i+1}^n-v_i^n}{\Delta x}=O(\Delta t)+O(\Delta x). \tag{2.58}$$

Now we will look inside the $O(\Delta t)+O(\Delta x)$. In the beginning of the section we just write the "big *O*" notation for equation (2.12) and quit.

We derived the equation (2.8) by replacing the remainder of the Taylor series expansion with the "big *O*" so we see that $O(\Delta t)$ contains a second derivative of $v(x,t)$ with respect to *t* evaluated for $x = i\Delta x$ and some *t* in a neighborhood of $n\Delta t$, and $O(\Delta x)$ contains a second derivative of $v(x,t)$ with respect to *x* evaluated for $t = n\Delta t$ and some *x* in an interval around $i\Delta x$. That is,

$$\frac{v_i^{n+1}-v_i^n}{\Delta t}+c\frac{v_{i+1}^n-v_i^n}{\Delta x}=v_{xx}\left(x,n\Delta t\right)\frac{\Delta x}{2}+v_{tt}\left(i\Delta x,t\right)\frac{\Delta t}{2}. \tag{2.59}$$

If we assume that the second derivative of $v(x,t)$ with respect to *x* and *t* exist and are bounded for some interval around the point (x,t), then the right hand side of the equation (2.59) goes to zero when Δx and Δt go to zero. Therefore, difference scheme (2.56) is pointwise consistent with the partial differential equation in (2.57).

In order to demonstrate that the difference equation (2.56) is accurate of order $(1,1)$, we will first design the equation according to the form of equation (2.52) and to do that we multiply the whole equation by Δt and solve for u_i^{n+1} and derive

$$u_i^{n+1}=u_i^n-\frac{c\Delta t}{\Delta x}\left(u_{i+1}^n-u_i^n\right). \tag{2.60}$$

To apply the definition 2.5, let assume again *v* be the solution of partial differential equation (2.57) and then

$$\Delta t\tau_i^n=v_i^{n+1}-\left\{v_i^n-cfl\left(v_{i+1}^n-v_i^n\right)\right\} \tag{2.61}$$

$$=v_i^n+v_t\left(i\Delta x,n\Delta t\right)\Delta t+v_{tt}\left(i\Delta x,\mathrm{t}_0\right)\frac{\Delta t^2}{2}-v_i^n$$

$$+cfl\left(v_i^n+v_x\left(i\Delta x,n\Delta t\right)\Delta x+v_{xx}\left(x_0,n\Delta t\right)\frac{\Delta x^2}{2}-v_i^n\right) \tag{2.62}$$

After cancelation, we get

$$\Delta t\tau_i^n = v_t\left(i\Delta x, n\Delta t\right)\Delta t + v_{tt}\left(i\Delta x, \mathrm{t}_0\right)\frac{\Delta t^2}{2} + \frac{c\Delta t}{\Delta x}\left(v_x\left(i\Delta x, n\Delta t\right)\Delta x + v_{xx}\left(x_0, n\Delta t\right)\frac{\Delta x^2}{2}\right) \quad (2.63)$$

and then

$$\tau_i^n = v_t\left(i\Delta x, n\Delta t\right) + v_{tt}\left(i\Delta x, \mathrm{t}_0\right)\frac{\Delta t}{2} + cv_x\left(i\Delta x, n\Delta t\right) + v_{xx}\left(x_0, n\Delta t\right)\frac{\Delta x}{2} \quad (2.64)$$

where t_0 and x_0 are the points in the neighborhood of $i\Delta x$ and $\mathrm{n}\Delta t$ respectively and enable the Taylor series expansion to hold. From using equation (2.57), we get

$$\tau_i^n = v_{tt}\left(i\Delta x, \mathrm{t}_0\right)\frac{\Delta t}{2} + v_{xx}\left(x_0, n\Delta t\right)\frac{\Delta x}{2}. \quad (2.65)$$

For the last step we should choose the norm. If we assume that v_{tt} and v_{xx} are bonded on $\mathbb{R}\times[0,t_1]$ for some $t_1 > t$, then we can use the sup-norm and conclude that the scheme is accurate of order $(1,1)$. If we assume that v_{tt} and v_{xx} satisfy

$$\sum_{-\infty}^{\infty}\left[\left(v_{tt}\right)_i^n\right]^2 < A < \infty$$

and

$$\sum_{-\infty}^{\infty}\left[\left(v_{xx}\right)_i^n\right]^2 < B < \infty$$

Then we see that the difference equation is accurate order $(1,1)$ again with respect to the 2-norm.

Note that we do assumption on the partial derivative v_{tt} and v_{xx} that they are bounded on $\mathbb{R}\times[0,t_1]$.

2.4 Stability

Stability is a necessary condition that must be satisfied by any finite difference method if we want the solution of the method to converge to the solution of partial differential equation. However, it is not a sufficient condition. We will see the relation between them in the next section. The consistency is also a necessary condition but it is easy to show that the method is consistent. Moreover most of the methods in literature are

consistent, but we cannot say the same for the stability. It is hard to demonstrate that a scheme is stable.

Stable difference scheme has the property that the small error at the beginning of the time evaluation cannot grow unboundedly. That is, the error can grow but has limit which is the exponential growth. We will also see this in the definition. We will define stability for difference scheme of the form

$$\mathrm{u}^{n+1} = Q\mathrm{u}^{n},\ n \geq 0, \tag{2.66}$$

which is a two level different scheme. This type of scheme will be generally used for solving initial-value problems, especially homogeneous and linear partial differential equations.

Definition 2.6 The different scheme (2.66) is said to be stable with respect to norm $\|\cdot\|$ if there exist positive constants Δx_0 and Δt_0, and non-negative constants K and β so that

$$\left\|\mathrm{u}^{n+1}\right\| \leq Ke^{\beta t}\left\|\mathrm{u}^{0}\right\| \tag{2.67}$$

for $0 \leq t$ where $t = (n+1)\Delta t,\ 0<\Delta x < \Delta x_0$ and $0 < \Delta t < \Delta t_0$ [21].

We should be aware of the fact that definition 2.6 is given in terms of unspecified norm since this norm may change rely on the condition, and we should remember that the definition of consistency and convergence are also given in that form and their norm also not specified. Another issue that should be noticed is that the solution of difference scheme can rise with time, and is not affected by the increase of time step.

The definition of stability in (2.67) is for homogeneous equation. A question may come in mind that what we will do to prove the convergence of non-homogeneous partial differential equation. The answer is that stability of homogeneous equation with the consistency is enough to demonstrate that the non-homogeneous difference scheme is convergent because all of the effects of the non-homogeneous term will be killed by the truncation error, $\boldsymbol{\tau}^{n}$.

A remark that should be taken into account is that there are other definitions of stability. One common definition in the literature is

$$\left\|\mathrm{u}^{n+1}\right\| \leq K\left\|\mathrm{u}^{0}\right\|. \tag{2.68}$$

The inequality (2.67) in the definition 2.6 is taken over from the inequality (2.68). This definition of stability does not allow the exponential increase.

Clearly, the definition 2.6 is a very stronger one. That is, inequality (2.68) implies the inequality (2.67). The definition with the condition (2.68) does not allow the growth, it is bounded. Therefore it is hard to hold. Sometimes it may be necessary to use the latter definition but most of the case it is enough to satisfy the conditions in the first definition.

It is difficult to demonstrate stability of a scheme directly. Fortunately there are useful technics to show the stability, and we will mention about them. However, to understand the definition of the stability, we also solve problems with using the definition. For first example, we prove the scheme which is used in example 2.1 to prove the convergence, so that we can compare the similarity of the steps used to prove convergence and used to prove stability. For second example, we will show the stability of the Lax-Friedrichs scheme to compare the conditions on Δt and Δx with the first example.

Example 2.4 Show that the difference method

$$u_i^{n+1} = (1 - cfl)u_i^n + cflu_{i-1}^n \tag{2.69}$$

where $cfl = \dfrac{c\Delta t}{\Delta x}$ is stable with respect to sup-norm.

Solution: we will follow the same strategy as we did in example 2.1. Of course we do not need to use the difference between the analytic and numeric solution. From the equation (2.69), we can say that

$$\left|u_i^{n+1}\right| \le (1 - cfl)\left|u_i^n\right| + cfl\left|u_{i-1}^n\right|. \tag{2.70}$$

using the triangular inequality. If we assume that $0 < cfl \le 1$, then taking the supremum over both sides of inequality (2.70) with respect to *i*, we get

$$\left\|\mathrm{u}^{n+1}\right\|_\infty = \left\|\mathrm{u}^n\right\|_\infty. \tag{2.71}$$

Therefore inequality (2.67) in the definition of stability is satisfied with $K = 1$ and $\beta = 0$.

We want to take attention to the assumption $0 < cfl \le 1$. This assumption is necessary to hold the stability. In this case, we say that the scheme is conditionally stable where the

condition is $0 < cfl \le 1$. If there is no condition on the relationship between Δt and Δx, we say for this case that the method is unconditionally stable or just say stable.

Example 2.5 Prove that the difference scheme

$$u_i^{n+1} = \frac{1}{2}\left(u_{i+1}^n - u_{i-1}^n\right) - \frac{cfl}{2}\left(u_{i+1}^n - u_{i-1}^n\right) \tag{2.72}$$

is stable with respect to the sup-norm provided that $-1 \le cfl \le 1$.

Solution: To use the triangle inequality, let first rearrange the equation (2.72).

$$u_i^{n+1} = \frac{1}{2}\left(1 - cfl\right)u_{i+1}^n + \frac{1}{2}\left(1 + cfl\right)u_{i-1}^n. \tag{2.73}$$

We can see that all of the coefficient of right hand side terms of equation are non-negative where assuming $-1 \le cfl \le 1$. So,

$$\left|u_i^{n+1}\right| \le \frac{1}{2}\left(1 - cfl\right)\left|u_{i+1}^n\right| + \frac{1}{2}\left(1 + cfl\right)\left|u_{i-1}^n\right|. \tag{2.74}$$

Taking the supremum of both sides with respect to *i*, we see that

$$\left\|u_k^{n+1}\right\|_\infty \le \left\|u_k^n\right\|_\infty. \tag{2.75}$$

For $K = 1$ and $\beta = 0$, the condition on the definition is satisfied. Therefore the scheme is stable for $-1 \le cfl \le 1$.

Note that for the example 2.4, we have the condition $0 < cfl \le 1$. If we use the method (2.69) for an hyperbolic partial differential equation (2.1), then the constant *c* must be positive and Δt should be less or equal to $\Delta x/c$. Since we can arrange Δt and Δx, we can satisfy the second restriction. However, what if the constant *c* is negative? The answer is that the difference method fails to converge to the given partial differential equation. Hence, we cannot apply the scheme to the partial differential equation to get the numerical solution. For the example 2.5, the condition $-1 \le cfl \le 1$ says that one can apply the difference method in the equation (2.72) to the hyperbolic partial differential equation in (2.1) whatever constant *c* is positive or negative. We must still keep in mind that we should choose Δt so that the condition $-1 \le cfl \le 1$ is satisfied.

We have used the term *cfl* constantly, but we have not mentioned about it. As we explained above, the restriction on the Δt, Δx and c that is necessary to show stability is called the *CFL condition* and named after Courant, Friedrichs, and Lewy [1].

2.5 Method to Prove Stability

To show the convergence, we will use stability but we have seen that to prove the stability from the definition is as difficult as to prove the convergence directly. Fortunately, there are mathematical concepts to demonstrate the stability and they are easy to apply the problem. One of them is the Fourier transform.

Definition 2.7 The Fourier transform of $v(x,t)$ is denoted by $\tilde{v}(w,t)$, $w \in \mathbb{R}$, and defined by the integral

$$\tilde{v}(w,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-iwx} v(x,t) dx \text{ [21].} \tag{2.76}$$

A sufficient condition for $v(x,t)$ to have a Fourier transform is that $v(x,t)$ is absolutely integrable on $(-\infty, \infty)$ [23].

Consider, for instance, the problem

$$v_t + cv_x = 0, \; x \in \mathbb{R}, \; t > 0 \tag{2.77}$$

$$v(x,0) = f(x), \; x \in \mathbb{R}. \tag{2.78}$$

Now, taking the Fourier transform of $v_t(x,t)$,

$$\tilde{v}_t(w,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-iwx} v_t(x,t) dx = -\frac{c}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-iwx} v_x(x,t) dx$$

$$= -\frac{c}{\sqrt{2\pi}} \left[v(x,t) e^{-iwx} \right]_{-\infty}^{\infty} - \frac{ciw}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-iwx} v(x,t) dx \tag{2.79}$$

$$= -ciw\tilde{v}(w,t). \tag{2.80}$$

We assume that $v(x,t)$ is sufficiently good at $\pm\infty$ so that integral in equation (2.79) exists and the evaluated term in the same equation is zero.

Therefore, we conclude that the partial differential equation is translated to the ordinary differential equation in the space of transformed functions by Fourier transform. After transformation, we solve the ordinary differential equation and turn back to our usual space. To turn back, we will use the Fourier inversion formula. That is,

$$v(x,t)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{iwx}\tilde{v}(w,\mathrm{t})\,dw. \tag{2.81}$$

The very important aspect of Fourier transform for proving stability is the Parseval's Identity which is

$$\left\|v(x,t)\right\|_2=\left\|\tilde{v}(w,t)\right\|_2. \tag{2.82}$$

Using this identity, we can use the definition of stability in transformed space and can easily demonstrate the stability of any given difference scheme. We will formally give the definitions and after that we state a proposition.

To define the discrete Fourier transform of $\mathbf{u}$, first let define vector $\mathbf{u}$ in l_2 as

$\mathbf{u}=(\cdots,u_{-1},u_0,u_1,\cdots)^T$,then

Definition 2.8 The discrete Fourier transform of $\mathbf{u}\in l_2$ is the function $\tilde{u}\in L_2[-\pi,\pi]$ defined by

$$\tilde{u}(s)=\frac{1}{\sqrt{2\pi}}\sum_{-\infty}^{\infty}e^{-ims}u_m, \tag{2.83}$$

for $s\in[-\pi,\pi]$ [21].

Similar to the continuous Fourier transform, there is also an inversion form for the discrete Fourier transform.

Definition 2.9 If $\mathbf{u}\in l_2$ and $\tilde{u}$ is the discrete Fourier transform of $\mathbf{u}$, then

$$u_m=\frac{1}{\sqrt{2\pi}}\int_{-\pi}^{\pi}e^{ims}\tilde{u}(s)\,ds \text{ [21].} \tag{2.84}$$

Thanks to the Parseval's Identity, when we use the Fourier transform to get $\tilde{u}$ to prove stability, we do not need to translate back to l_2 space. To clear this, let us look at the following proposition.

Proposition 2.1 If $\mathbf{u} \in l_2$ and $\tilde{u}$ is the discrete Fourier transform of $\mathbf{u}$, then

$$\|\tilde{u}\|_2 = \|\mathbf{u}\|_2 \tag{2.85}$$

where the first norm is the L_2 norm on $[-\pi, \pi]$ and the second norm is the l_2 norm [21]. Note that L_2 is the space of complex valued and Lebesgue square integrable functions defined as

$$L_2[-\pi,\pi] = \left\{ v : [-\pi,\pi] \to \mathbb{C} : \int_{-\pi}^{\pi} |v(x)|^2 \, dx < \infty \right\} \tag{2.86}$$

with the norm

$$\|v\|_2 = \sqrt{\int_{-\pi}^{\pi} |v(x)|^2 \, dx} \, . \tag{2.87}$$

We skip the proof of the proposition 2.1. Reader can easily find the proof from any book that mentions Fourier transform, like [23].

Remember that, in the definition of stability, we require the following condition

$$\left\|\mathrm{u}^{n+1}\right\|_2 \le K e^{\beta(\mathrm{n}+1)\Delta t} \left\|\mathrm{u}^0\right\|_2 . \tag{2.88}$$

From the Parseval's identity, we know that

$\left\|u^{n+1}\right\|_2 = \left\|\tilde{u}^{n+1}\right\|_2$ and $\left\|u^0\right\|_2 = \left\|\tilde{u}^0\right\|_2$,

Thus we can conclude that

$$\left\|\tilde{u}^{n+1}\right\|_2 \le K e^{\beta(\mathrm{n}+1)\Delta t} \left\|\tilde{u}^0\right\|_2 . \tag{2.89}$$

Therefore, if we are able to find a K and β that satisfy (2.89), then we can also satisfy the inequality (2.88). It means that the scheme is stable.

(Note that we can choose any norm for the inequality (2.89), therefore without loss of generality we choose the l_2 norm).

Example 2.6 Discuss the stability of the following difference method

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_{i+1}^n - u_i^n}{\Delta x} = 0 , \quad -\infty < i < \infty . \tag{2.90}$$

Solution: Remember that, we analyzed the consistency of this scheme in example 2.3 and we found that it has been consitent. Now, rearranging terms, we get

$$u_i^{n+1} = u_i^n - cfl\left(u_{i+1}^n - u_i^n\right) = \left(1+cfl\right)u_i^n - cflu_{i+1}^n \tag{2.91}$$

where $cfl = \dfrac{c\Delta t}{\Delta x}$.

Taking the discrete Fourier transform of both side of the equation (2.91), we have

$$\tilde{u}^{n+1}(s) = \frac{1}{\sqrt{2\pi}}\sum_{k=-\infty}^{\infty} e^{-iks}u_k^{n+1}$$

$$= \frac{1}{\sqrt{2\pi}}\sum_{k=-\infty}^{\infty} e^{-iks}\left[\left(1+cfl\right)u_k^n - cflu_{k+1}^n\right]$$

$$= \left(1+cfl\right)\frac{1}{\sqrt{2\pi}}\sum_{k=-\infty}^{\infty} e^{-iks}u_k^n - cfl\frac{1}{\sqrt{2\pi}}\sum_{k=-\infty}^{\infty} e^{-iks}u_{k+1}^n$$

$$= \left(1+cfl\right)\tilde{u}^n(s) - cfl\frac{1}{\sqrt{2\pi}}\sum_{k=-\infty}^{\infty} e^{-iks}u_{k+1}^n \tag{2.92}$$

Now, by making change of variable $m = k+1$ we have

$$\frac{1}{\sqrt{2\pi}}\sum_{k=-\infty}^{\infty} e^{-iks}u_{k+1}^n = \frac{1}{\sqrt{2\pi}}\sum_{m=-\infty}^{\infty} e^{-i(\mathrm{m}-1)s}u_m^n = e^{is}\frac{1}{\sqrt{2\pi}}\sum_{m=-\infty}^{\infty} e^{-i(\mathrm{m}-1)s}u_m^n = e^{is}\tilde{u}^n(s). \tag{2.93}$$

Therefore, using (2.93) in (2.92) we get

$$\tilde{u}^{n+1}(s) = \left(1+cfl\right)\tilde{u}^n(s) - e^{is}cfl\tilde{u}^n(s) = \tilde{u}^n(s)\left[1+cfl-e^{is}cfl\right]$$

$$= \tilde{u}^n(s)\left[1+cfl-cfl\left(\cos(s)+i\sin(s)\right)\right]. \tag{2.94}$$

Here, we denote the coefficient of $\tilde{u}^n(s)$ in (2.94) as $\rho(s)$ and call it as the symbol of difference scheme (2.90). That is,

$$\rho(s) = 1+cfl-cfl\left(\cos(s)+i\sin(s)\right). \tag{2.95}$$

Notice that, by applying the discrete Fourier transform, we elude the spatial derivative. Applying the consequence of (2.94) $n+1$ times, we have

$$\tilde{u}^{n+1}(s) = \left[1+cfl-cfl\left(\cos(s)+i\sin(s)\right)\right]^{n+1}\tilde{u}^0(s). \tag{2.96}$$

By bounding *cfl*, if we can achieve

$$\left|1+cfl-cfl\left(\cos(s)+i\sin(s)\right)\right|\le 1, \tag{2.97}$$

then we can conclude that (2.89) holds so that the method is stable.

Now, it turns out to find the restriction on *cfl*. Let us take the square of (2.95).

$$\begin{aligned}\left|\rho(s)\right|^2 &= \left(1+cfl-cfl\cos(s)\right)^2+\left(cfl\sin(s)\right)^2\\ &= \left(1+cfl\right)^2-2\left(1+cfl\right)cfl\cos(s)+cfl^2\cos^2(s)+cfl^2\sin^2(s)\\ &= \left(1+cfl\right)^2+cfl^2-2cfl\left(1+cfl\right)\cos(s):=\mathrm{g}(s).\end{aligned} \tag{2.98}$$

Note that, we define the $g(s)$ in (2.98) and it is defined on $[-\pi,\pi]$. We must determine the maximum and minimum value of $g(s)$ to bound its magnitude value with 1. To do this, we take the derivative of $g(s)$ with respect to *s* and set it to zero to find the point that have potential to be maximum or minimum.

$$g'(s)=2cfl\left(1+cfl\right)\sin(s). \tag{2.99}$$

This may maximum or minimum at the points that do the (2.99) zero and the endpoints. These are the points $-\pi, 0$ and π.

$$\text{For } s=0,\ \left|\rho(s)\right|^2=1+2cfl+cfl^2+cfl^2-2cfl-2cfl^2=1, \tag{2.100}$$

$$\text{For } s=\pm\pi,\ \left|\rho(s)\right|^2=\left(1+cfl\right)^2+2cfl\left(1+cfl\right)+cfl^2=\left(1+2cfl\right)^2. \tag{2.101}$$

In order to bound $\left|\rho(s)\right|$ with 1, thus, we must have

$$\left(1+2cfl\right)^2\le 1\Leftrightarrow -1\le 1+2cfl\le 1\Leftrightarrow -2\le 2cfl\le 0\Leftrightarrow -1\le cfl\le 0. \tag{2.102}$$

Therefore, we conclude that the scheme in (2.90) is conditionally stable and the condition is $-1\le\dfrac{c\Delta t}{\Delta x}\le 0$. Notice that c have to be negative to satisfy the condition. This means that the method does not work for $c>0$.

In conclusion, to prove the stability by applying the Fourier transform is easy from proving directly from the definition.

2.6 The Lax-Richtmyer Equivalence Theorem

Finally, we come up the Lax Equivalence Theorem that relates the convergence with consistency and stability. This is the fundamental theorem in the theory field of the finite difference methods.

Theorem 2.1 A consistent finite difference scheme for a partial differential equation for which the initial value problem is well-posed is convergent if and only if it is stable [24].

This theorem is called Lax-Richtmyer Equivalence Theorem or just Lax Equivalence Theorem.

Generally we want to reach convergence from stability rather than to reach stability from convergence. Furthermore, we may ask for order of convergence. For these reasons, the following theorem is more useful for us.

Theorem 2.2 If a two-level difference scheme

$$\mathbf{u}^{n+1} = Q\mathbf{u}^n + \Delta t\mathbf{G}^n \tag{2.103}$$

is accurate of order (p,q) to a linear initial-value problem which is well-posed in the norm $\|\cdot\|$ and it is stable with respect to the norm $\|\cdot\|$, then it is convergent with respect to the same norm and same order [21].

We require in both definitions that the initial-value problem must be well-posed. An initial-value problem can be considered as well-posed if it depends on its initial condition while time evolves. We can define it as follow.

Definition 2.10 The initial-value problem for a first order equation is well-posed if for all *t*, there is a constant *K* such that the inequality

$$\|u(t,\cdot)\| \le K\|u(0,\cdot)\| \tag{2.104}$$

holds for all initial data $u(0,\cdot)$ [21].

We will not concern the well-posedness of initial value problem so much when solving the problems. One reason for this is that most of the problems satisfy the well-posed condition.

Up to now, we analyze some methods in terms of stability and consistency. For instance, we see that the scheme in (2.56) is accurate of order $(1,1)$ with respect to 2-norm in example 2.3 and also we see that the same method is stable if $-1 < cfl < 0$ with respect to 2-norm. Then we can conclude from the Theorem 2.2 that the scheme given in (2.56) is convergent of order $(1,1)$ with respect to 2-norm.

CHAPTER 3

FINITE VOLUME METHOD

In previous chapter we mentioned theoretical background for all types of partial differential equations and for all numerical methods that used to solve these partial differential equations. Since our main concern is about high resolution schemes, we will interest in related partial differential equation, namely hyperbolic equations. Specifically, we will deal with conservation laws which are a significant part of homogeneous hyperbolic equations.

3.1 Conservation Laws

The basic example of a conservation law is

$$v_t(x,t) + f\left(v(x,t)\right)_x = 0. \tag{3.1}$$

Here $f(v)$ is the flux function. The quasilinear form of equation (3.1) is

$$v_t + f'(v)v_x = 0. \tag{3.2}$$

This equation is hyperbolic if $f'(v)$ is real. Notice that the equation (2.1) is a conservation law with the flux function $f(v) = cv$. This flux function is linear but most of the physical problems cause to nonlinear conservation law and so nonlinear flux function. That is, $f(v)$ is a nonlinear function of *v*.

Conservation laws generally emerge from physical principles. To verify this, let consider a problem. Suppose that a liquid is flowing with velocity *c* through the one-dimensional pipe. Notice that the velocity can only change with time *t* and *x*. Assume also that there is some substance in this fluid and its quantity is so less that does not

influence the fluid dynamics. Our problem is to model the density of this substance in terms of *x* and *t*. Let $v(x,t)$ denote the density of this material.

The unit of density is mass per unit volume, usually. We are studying on one-dimensional pipe so that the only change in space is in the x direction. Therefore, it is logical to use mass per length as a unit. That is grams per meter. Now, to determine the total mass of this material between the point x_1 and x_2 for some time t, we can use the following expression.

$$\int_{x_1}^{x_2} v(x,t)dx \tag{3.3}$$

Note that if the material is conservative that is neither created nor destroyed, then the total mass in the section of pipe between x_1 and x_2 can only change due to the fluxes, i.e., flow of the substance through the edges of the given section. Now let $F_1(t)$ and $F_2(t)$ be the rates at which the material flows past the fixed points x_1 and x_2, respectively. Let unit of this rate be grams per second. We will consider that if $F_i(t)$ is positive then the material flows to the right and if $F_i(t)$ is negative then the material flows to the left. Because of the fact that the total mass in the section can vary only due to the fluxes at the endpoints with time evolves, we can derive the following.

$$\frac{d}{dt}\int_{x_1}^{x_2} v(x,t)\,dx = F_1(t) - F_2(t). \tag{3.4}$$

Equation (3.4) is the fundamental integral form of a conservation law and most of the methods that we will use are in this form. We can interpret this equation as that the rate of chance of total mass can only stem from the fluxes through the endpoints. In equation (3.4) we should determine $F_i(t)$ in terms of $v(x,t)$ so that we can get an equation to solve for density of substance, $v(x,t)$. For our problem, the flux at a given point *x* and time *t* is just the product of the density $v(x,t)$ and the velocity $c(x,t)$. We can confirm this in terms of units. That is the unit of density is gram per meter and the unit of velocity is meter per second. The product of two units gives us gram per second which is the unit of flux. Thus we can write

$$\text{flux} = f(q,x,t) = c(x,t)q(x,t). \tag{3.5}$$

If the velocity is constant with respect to time and space, then we can write

$$\text{flux} = f(q) = cq. \tag{3.6}$$

For the case of constant speed, flux at any point can only depend on the density. It does not change owing to location of point and time. For such cases we can rewrite the basic form of conservation law in (3.4) as

$$\frac{d}{dt}\int_{x_1}^{x_2} v(x,t)\,dx = f\left(v(x_1,t)\right) - f\left(v(x_2,t)\right). \tag{3.7}$$

We can rewrite this as

$$\frac{d}{dt}\int_{x_1}^{x_2} v(x,t)\,dx = f\left(v(x,t)\right)\Big|_{x_1}^{x_2}. \tag{3.8}$$

If the functions $v(x,t)$ and f are sufficiently smooth we can write the equation (3.8) as

$$\frac{d}{dt}\int_{x_1}^{x_2} v(x,t)\,dx = -\int_{x_1}^{x_2} \frac{\partial}{\partial x} f\left(v(x,t)\right)dx. \tag{3.9}$$

If we pick up the terms in a single integral,

$$\int_{x_1}^{x_2} \frac{\partial}{\partial t} v(x,t) + \frac{\partial}{\partial x} f\left(v(x,t)\right)dx = 0. \tag{3.10}$$

To handle this equality, the integrand of integral in equation (3.10) must be zero. Therefore,

$$\frac{\partial}{\partial t} v(x,t) + \frac{\partial}{\partial x} f\left(v(x,t)\right) = 0. \tag{3.11}$$

This is called the differential form of the conservation laws. We will built our schemes on this basic form and develop them from this fundamental pattern. As we will see later that it is very important for a scheme to be in conservation form because otherwise numerical methods most often do not capture the speed of discontinuity.

3.2 The Riemann Problem

The Riemann problem is the initial value problem which has a special initial condition. The initial data consist of two constant values v_l and v_r separated by a jump discontinuity at the point $x = 0$. That is, the hyperbolic equation

$$v_t + \left(f(v) \right)_x = 0 \tag{3.12}$$

and initial condition

$$v(x,0) = v_0(x) \tag{3.13}$$

where

$$v_0(x) = \begin{cases} v_l & \text{if } x < 0, \\ v_r & \text{if } x > 0. \end{cases}$$

For the scalar advection equation $v_t + cv_x = 0$, the solution to the Riemann problem is formed by the discontinuity propagating along the characteristic with speed *c* and the solution is

$$v(x,t) = \begin{cases} v_l & \text{if } x - ct < 0, \\ v_r & \text{if } x - ct > 0. \end{cases} \tag{3.14}$$

The Riemann problem is significant structure to build Godunov's method which is the main building block for construction of high resolution methods. We will mention about this in chapter 4.

3.3 Finite Volume Methods

Finite volume methods are similar to finite difference methods, and finite volume methods can be considered as a finite difference approximation to the differential equation. However, there are important differences that provide advantages to finite volume method. It is the fact that finite volume methods are derived on the basis of the integral form of the conservation law.

For finite volume method, we divide the spatial domain into subintervals and try to approximate to the value of integral of *v* over each of these grid cells. We recalculate these values by approximating to the flux throughout the endpoints of each interval. We do the same for each time step (Figure 3.1). If we denote the *i*th interval by

$C_i = \left(x_{i-1/2}, x_{i+1/2}\right)$ and denote the average value of $v\left(i\Delta x, \mathrm{n}\,\Delta t\right)$ over the C_i at time t_n by u_i^n then we can write the following expression.

$$u_i^n \approx \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} v\left(x,t_n\right)dx \equiv \frac{1}{\Delta x}\int_{C_i} v\left(x,t_n\right)dx\,. \tag{3.15}$$

Although it is not necessary to assume that the grid is uniform, we will accept that the grid is uniform for ease.

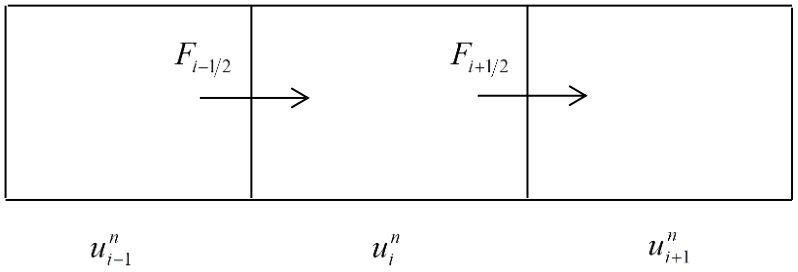

Figure 3. 1 Updating the cell average with the fluxes throughout the endpoints for finite volume method.

The integral in (3.15) approximate the value of $v\left(x,t\right)$ at the midpoint of the grid cell with the order of $O\left(\Delta x^2\right)$, if $v\left(x,t\right)$ is a smooth function. This situation may not be superb, but it enables to use the significant properties of the conservation law in driving numerical methods to work with cell averages. In fact, we can guarantee that the numerical method is conservative. That is, it imitates the true solution, and this property of numerical method is crucial for calculating shock waves accurately. This property of numerical methods comes from the fact that $\sum_{i=1}^{N} u_i^n \Delta x$ approximates the integral of $v\left(x,t\right)$ over the interval $\left[a,b\right]$ (assume that we have an interval $\left[a,b\right]$ and divide it into *N* subinterval), and if we work with a method that is in conservation form (we will just mention it below), then this approximate sum will vary only because of fluxes at the endpoints of interval, namely *a* and *b*. Therefore, the total value of discrete sum will maintain the same, or at least change correctly if we impose the boundary conditions agreeably.

Now applying the integral form of the conservation law (3.4) to the grid cell C_i, we get

$$\frac{d}{dt}\int_{C_i} v(x,t)\,dx = f\left(v\left(x_{i-1/2},t\right)\right) - f\left(v\left(x_{i+1/2},t\right)\right). \tag{3.16}$$

By using (3.16), we can improve an explicit numerical method. If we know the cell averages u_i^n at time t_n, we can approximate the cell averages u_i^{n+1} at time t_{n+1}. Integrating the equation in (3.16) from t_n to t_{n+1} we have

$$\int_{C_i} v(x,t_{n+1})\,dx - \int_{C_i} v(x,t_n)\,dx = \int_{t_n}^{t_{n+1}} f\left(v\left(x_{i-1/2},t\right)\right)dt - \int_{t_n}^{t_{n+1}} f\left(v\left(x_{i+1/2},t\right)\right)dt\,. \tag{3.17}$$

By dividing (3.17) with Δx gives

$$\frac{1}{\Delta x}\int_{C_i} v(x,t_{n+1})\,dx = \frac{1}{\Delta x}\int_{C_i} v(x,t_n)\,dx$$

$$-\frac{1}{\Delta x}\left[\int_{t_n}^{t_{n+1}} f\left(v\left(x_{i+1/2},t\right)\right)dt - \int_{t_n}^{t_{n+1}} f\left(v\left(x_{i-1/2},t\right)\right)dt\right]. \tag{3.18}$$

We can conclude from the above equation that we can update the average value of $v(x,t)$ for next time step from the integral form of conservation law (3.4). We can also deduce that we should work on numerical methods which are the following form

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right), \tag{3.19}$$

where $F_{i+1/2}^n$ is an approximation to the flux at point $x_{i+1/2}$. That is,

$$F_{i+1/2}^n \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f\left(v\left(x_{i+1/2},t\right)\right)dt\,. \tag{3.20}$$

Now to arrive a completely discrete method, we should approximate the above average flux in terms of u^n.

We can realize that value of $F_{i+1/2}^n$ depends on the approximate values to the average value of $v(x,t)$ on both sides of the point $x_{i+1/2}$. Thus, it is logical to estimate the $F_{i+1/2}^n$ from the value of u_{i+1}^n and u_i^n. From this idea, we can write

$$F_{i+1/2}^n = G\left(u_i^n, u_{i+1}^n\right), \tag{3.21}$$

where *G* is some numerical flux function. If we insert this term into (3.19), we have

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(G\left(u_{i+1}^n, u_i^n\right) - G\left(u_i^n, u_{i-1}^n\right)\right). \tag{3.22}$$

Because we drive the above equation from the equation (3.18), which is the integral form of conservation law, and it imitates the property of conservation low, the equation in (3.22) is in conservation form.

At the beginning of this section we said that finite volume method can be considered as same with finite difference method. To demonstrate this, if we rearrange the term of equation (3.22), we have

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{F_{i+1/2}^n - F_{i-1/2}^n}{\Delta x} = 0. \tag{3.23}$$

This is the difference scheme of conservation law $v_t + f_x(v) = 0$.

In equation (3.21), we gave the general form of flux functions. We know that it depends on u_{i+1}^n and u_i^n, but what the function *G* can be is still undetermined. To determine *G*, the first idea that comes in mind may be the simple arithmetic average. That is,

$$F_{i+1/2}^n = G\left(u_{i+1}^n, u_i^n\right) = \frac{1}{2}\left(f\left(u_{i+1}^n\right) + f\left(u_i^n\right)\right), \tag{3.24}$$

and inserting this into (3.19), we get

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{2\Delta x}\left(f\left(u_{i+1}^n\right) - f\left(u_{i-1}^n\right)\right). \tag{3.25}$$

Be aware that this method is in conservation form, but it is ordinarily unstable for hyperbolic equation.

An example of finite volume method in conservation form is the classical Lax-Friedrichs method which has the form

$$u_i^{n+1} = \frac{1}{2}\left(u_{i+1}^n - u_{i-1}^n\right) - \frac{\Delta t}{2\Delta x}\left(f\left(u_{i+1}^n\right) - f\left(u_{i-1}^n\right)\right). \tag{3.26}$$

This method looks like unstable method in (3.25), but we replace u_i^n by $\frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right)$ and this change enables method to be stable for a linear hyperbolic equation if the necessary condition $cfl \le 1$ is fulfilled.

Notice that the Lax-Friedrichs method is of the form (3.19) if we define the numerical flux as

$$F_{i-1/2}^{n} = \frac{1}{2}\left(f\left(u_{i-1}^{n}\right) + f\left(u_{i}^{n}\right)\right) - \frac{\Delta x}{2\Delta t}\left(u_{i}^{n} - u_{i-1}^{n}\right). \tag{3.27}$$

3.4 REA Algorithm

To solve nonlinear Euler equations of gas dynamics, Godunov suggested an approach [3]. This approach is called REA algorithm which is the short writing of reconstruct-evolve-average. As we said before, this algorithm would be the base for huge amount of new algorithms which are modern, high order, improvable etc.

The algorithm consists of three steps:

1. Reconstruct a function $\tilde{v}^{n}\left(x,t_{n}\right)$ which is piecewise polynomial defined for each x, from the cell averages u_{i}^{n}. For straightforward situation, one can define $\tilde{v}^{n}\left(x,t_{n}\right)$ as a piecewise constant function that takes the value u_{i}^{n} for the *i*th interval. That is,

   $$\tilde{v}^{n}\left(x,t_{n}\right) = u_{i}^{n} \qquad \text{for all } x \in C_{i} \tag{3.28}$$

2. Evolve the hyperbolic equation exactly or approximately with this initial data to achieve $\tilde{v}^{n}\left(x,t_{n+1}\right)$ for the next time step.

3. Average this piecewise function over each interval to get new cell averages, i.e.,

   $$u_{i}^{n+1} = \frac{1}{\Delta x}\int_{C_{i}} \tilde{v}^{n}\left(x,t_{n+1}\right)dx \tag{3.29}$$

By applying these three steps, we get the value for the next time step (Δt times later). To find the values for given time, the algorithm will be repeated.

To evolve the hyperbolic equation in step 2, we should use the theory of Riemann problems introduced in section 3.2 since our initial data consists of piecewise constant functions.

We build a function $\tilde{v}^{n}\left(x,t_{n}\right)$ from the u_{i}^{n} in step 1. At first time when Godunov had been used the REA algorithm, he has reconstructed $\tilde{v}^{n}\left(x,t_{n}\right)$ as a simple piecewise

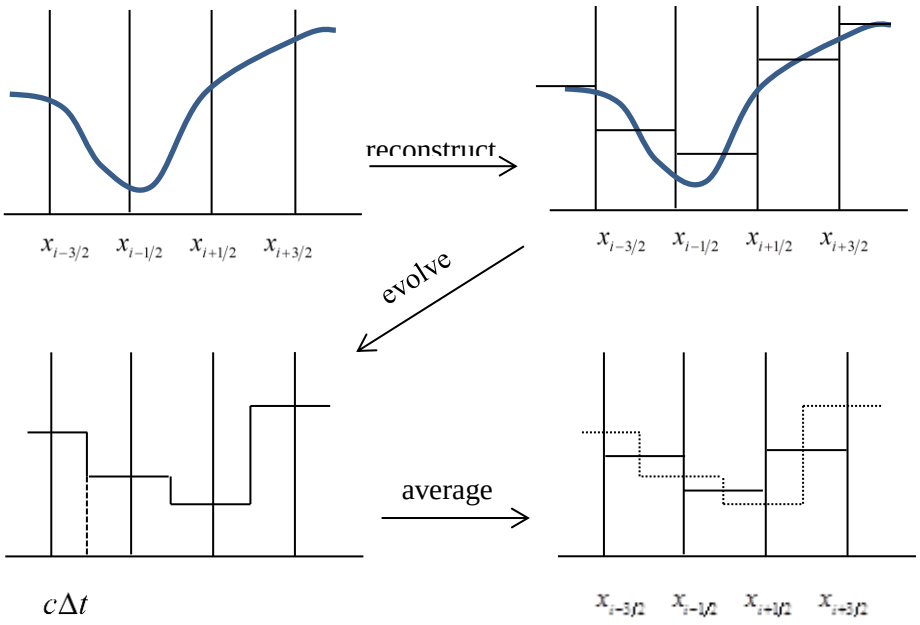

Figure 3. 2 Representation of REA algorithm

constant function. Using such a reconstruction leads to Riemann problem which is easy to solve, yet it gives just first-order accuracy. In order to achieve better accuracy, we may consider utilize a better reconstruction. That is instead of using piecewise constant function, we can use piecewise linear function as an initial data. This way of thinking establishes the basis for the Godunov type high resolution schemes that we will mention in next chapter.

Now, it is time to improve a finite volume method which can be easily performed in practice, based on the REA algorithm. In step 3, in order to determine the new cell average u_i^{n+1}, we should compute the integral of $\tilde{v}^n\left(x,t_{n+1}\right)$. Because the function $\tilde{v}^n\left(x,t_{n+1}\right)$ contains a lot of discontinuities, it is hard to implement. However, there is an easy way to find the cell averages. Instead of calculating integral, we can determine the numerical flux function for each cell and using this, we can easily compute the new cell averages.

Remember that, we define the numerical flux $F_{i-1/2}^n$ as an approximation to the time average of the flux at $x_{i-1/2}$ from t_n to t_{n+1} in (3.20). That is,

$$F_{i-1/2}^n \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f\left(v\left(x_{i-1/2},t\right)\right)dt.$$

Normally, the function $v\left(x_{i-1/2},t\right)$ changes with *t*, and we do not know this change of the analytic solution exactly. However, if we replace $v\left(x,t\right)$ by the function $\tilde{v}^n\left(x,t\right)$ defined in the REA algorithm using piecewise constant reconstruction, we can calculate the integral certainly. Because $\tilde{v}\left(x_{i-1/2},t\right)$ is constant over the time interval $\left(t_n,t_{n+1}\right)$, its value is equal to the solution of Riemann problem at centered $x_{i-1/2}$. Then, we define $F_{i-1/2}^n$ as

$$F_{i-1/2}^n=\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f\left(v^{Rim}(u_{i-1}^n,u_i^n)\right)dt=f\left(v^{Rim}(u_{i-1}^n,u_i^n)\right). \tag{3.30}$$

Here, $v^{Rim}(u_{i-1}^n,u_i^n)$ represents the solution of Riemann problem at the point $x_{i-1/2}$.

Therefore, Godunov's method for conservation laws has the following way of implementation:

- Solve the Riemann problem at $x_{i-1/2}$ to obtain $v^{Rim}(u_{i-1}^n,u_i^n)$.
- Define the flux $F_{i-1/2}$ as a function of u_{i-1}^n and u_i^n.
- Apply the flux-differencing formula (3.19).

We will generally use this format to state the methods.

CHAPTER 4

HIGH RESOLUTION METHODS

High resolution methods are built for solving conservation laws equations that have discontinuous solutions, such as gas dynamics equations. Von Neumann, Richtmyer and Lax have been studied for the numerical solution of partial differential equations. Godunov developed their methods greatly and he applied his methods to a lot of problems in one-dimensional gas dynamics, in which contact discontinuities and shock waves arise [3].

In this chapter, we will introduce the methods that are used to compare and contrast. In section 4.1, we will mention the classical high resolution methods. Although they have some problems, as will be mentioned later in this chapter, they give significant information to build more efficient methods. For next sections, we will introduce the concept of limiter and slope, we will upgrade the reconstruction of the piecewise polynomial function in step 1 of the REA algorithm from constant to linear to get more adequate methods. After that, we will talk about advanced high resolution methods.

We will use the following test problem to evaluate the methods.

$$v_t + cv_x = 0, \tag{4.1}$$

$$v(x,0) = \begin{cases} \exp\left(-200\left(x-0.3\right)^2\right) & \text{for } 0 \le x < 0.6 \\ 1 & \text{for } 0.6 \le x \le 0.8 \\ 0 & \text{for } 0.8 < x \le 1.0\ , \end{cases} \tag{4.2}$$

$$v(0,t) = v(1,t). \tag{4.3}$$

Clearly, (4.1) is an advection equation. This seems a trivial equation, but it contains the core of the hardship encountered in numerical approaches to hyperbolic problems [25]. Therefore it is very important to understand the methods properly for the advection

equation. The initial condition for (4.1) consists of a smooth pulse named as Gaussian hump and a square pulse. We choose such an initial condition because some methods are perfect for smooth solution but they fail for the solutions which have discontinuity and some other methods are good enough for discontinuous case but they have some drawback for the smooth case. The boundary condition in equation (4.3) is periodic. Therefore when the front of the solution goes out from the point $x=1$, it will come in from the point $x=0$.

We carry out the calculations on a uniform grid of 200 intervals, our speed, *c*, is 1, we take the *cfl* as 0.8. That is, $dt/dx=0.8$. We study the results at $t=1.0$ as short time evaluation and $t=5.0$ as a long term evaluation. We use these informations for all the methods in this chapter and for the advection equation in (4.1)-(4.3).

4.1 Classical High Resolution Methods

4.1.1 The Upwind Method

In the first half of the twentieth century, difference methods especially worked by von Neumann, Richtmyer, and Lax had all been centered methods and symmetric about the point where the solution is updated. However for hyperbolic problems, information propagates as waves moving along characteristics, so it can be found better numerical flux functions. From this idea, upwind method is developed. Courant, Isaacson and Rees published a paper in 1952. In their publication, they choose upwind-biased stencil which follows from the backward variant of the method of characteristics [26].

The idea behind the upwind method is that the information for each characteristic variable (we only have one variable for scalar advection equation.) is obtained by looking in the direction from which this information should be coming. For the one-way wave equation there is only one characteristic, so there is only one speed which goes to the right or left. Remember that for the equation in (4.1), if the constant *c* is positive then the wave goes to the right and if *c* is negative then the wave goes to the left. That is why; the upwind method is defined as follows

$$u_i^{n+1}=u_i^n-\frac{c\Delta t}{\Delta x}\left(u_i^n-u_{i-1}^n\right) \text{ for } c>0, \tag{4.4}$$

$$u_i^{n+1}=u_i^n-\frac{c\Delta t}{\Delta x}\left(u_{i+1}^n-u_i^n\right) \text{ for } c<0 \text{ [1].} \tag{4.5}$$

We can write the equation (4.4) and (4.5) with together. To do this first lets define

$$c^{+} = \max(c,0), \quad c^{-} = \min(c,0). \tag{4.6}$$

Then,

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left[c^{+}\left(u_i^n - u_{i-1}^n\right) + c^{-}\left(u_{i+1}^n - u_i^n\right)\right]. \tag{4.7}$$

The difference method (4.7) is first-order accurate for smooth initial data and stable for $-1 \le cfl \le 1$. When we apply the method to the test problem (4.1)-(4.3), we reach the following graphical results.

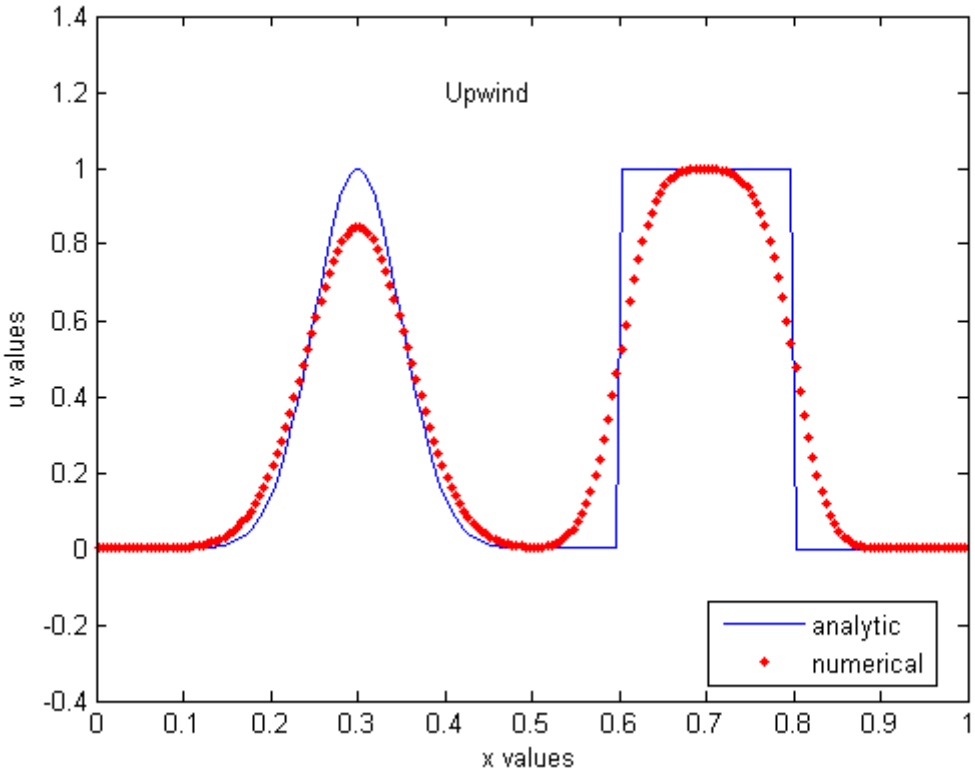

Figure 4. 1 Upwind method applied to the test problem (4.1)-(4.3) at time t=1.0

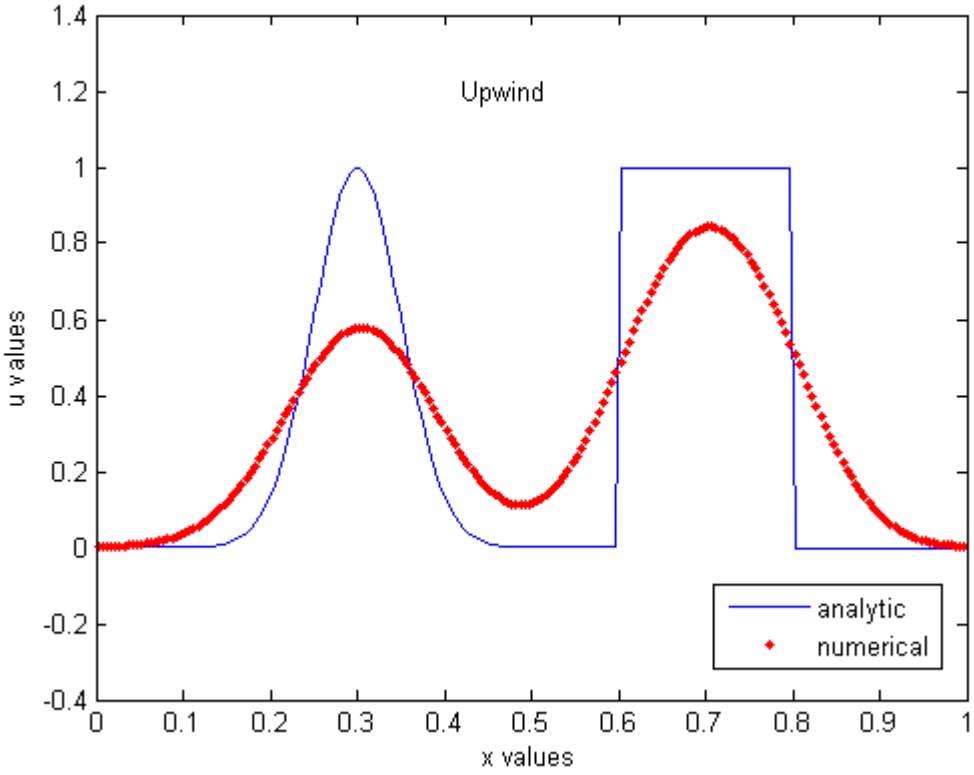

Figure 4. 2 Upwind method applied to the test problem (4.1)-(4.3) at time t=5.0

We can conclude from the graphs that it cannot capture the solution well. The numerical results of the method have great dissipation that cannot be negligible. The accuracy of the method is poor so much, especially time evolves. Even though these drawbacks, the upwind method enables to capture shock waves without oscillations. We will see in later sections that some methods oscillate around the discontinuity.

4.1.2 The Lax-Wendroff Method

Remember that the upwind method in the previous section is only first-order accurate. As we saw in the figure 4.1 that its result is so poor, it is more clearly seen in long time evaluation (Figure 4.2). Improvement of the upwind method by adding correction terms is done and named as Lax-Wendroff method. This method is second-order accurate in both space and time where the solution is smooth and stable for $-1 < cfl < 1$.

Base of the Lax-Wendroff method for the advection equation $v_t + cv_x = 0$ is the Taylor series expansion. Note that since $v_t = -cv_x$, then

$$v_{tt} = \left(-cv_x\right)_t = -cv_{xt} = -cv_{tx} = -c\left(v_t\right)_x = -c\left(-cv_x\right)_x = c^2 v_{xx}. \tag{4.8}$$

Expanding $v\left(i\Delta x,(\mathrm{n}+1)\Delta t\right)$ with respect to time, we get

$$\begin{aligned} &v\left(i\Delta x,(\mathrm{n}+1)\Delta t\right)=v\left(i\Delta x,\mathrm{n}\,\Delta t\right)+\Delta t v_t\left(i\Delta x,\mathrm{n}\,\Delta t\right)\\ &+\frac{1}{2}\left(\Delta t\right)^2 v_{tt}\left(i\Delta x,\mathrm{n}\,\Delta t\right)+O\left(\Delta t^3\right) \end{aligned}. \tag{4.9}$$

Using the equality (4.8), we have

$$\begin{aligned} &v\left(i\Delta x,(\mathrm{n}+1)\Delta t\right)=v\left(i\Delta x,\mathrm{n}\,\Delta t\right)-\Delta t c v_x\left(i\Delta x,\mathrm{n}\,\Delta t\right)\\ &+\frac{1}{2}\left(\Delta t\right)^2 c^2 v_{xx}\left(i\Delta x,\mathrm{n}\,\Delta t\right)+O(\Delta t^3) \end{aligned}. \tag{4.10}$$

Using the central difference approximation for the spatial derivative in the equation, we reach

$$\begin{aligned} &v_i^{n+1}=v_i^n-c\Delta t\left(\frac{v_{i+1}^n-v_{i-1}^n}{2\Delta x}+O\left(\Delta x^2\right)\right)\\ &+\frac{1}{2}c^2\left(\Delta t\right)^2\left(\frac{v_{i+1}^n-2v_i^n+v_{i-1}^n}{\Delta x^2}+O\left(\Delta x^2\right)\right)+O(\Delta t^3) \end{aligned}. \tag{4.11}$$

Rearranging this, we get

$$\begin{aligned} &v_i^{n+1}=v_i^n-\frac{c\Delta t}{2\Delta x}\left(v_{i+1}^n-v_{i-1}^n\right)+\frac{1}{2}c^2\left(\frac{\Delta t}{\Delta x}\right)^2\left(v_{i+1}^n-2v_i^n+v_{i-1}^n\right)\\ &+O(\Delta t^3)+O\left(\Delta t\Delta x^2\right) \end{aligned}. \tag{4.12}$$

Therefore we approximate the advection equation $v_t+cv_x=0$ by the difference method

$$u_i^{n+1}=u_i^n-\frac{c\Delta t}{2\Delta x}\left(u_{i+1}^n-u_{i-1}^n\right)+\frac{1}{2}c^2\left(\frac{\Delta t}{\Delta x}\right)^2\left(u_{i+1}^n-2u_i^n+u_{i-1}^n\right). \tag{4.13}$$

Difference scheme (4.13) is called the Lax-Wendroff method. We can see that it has 3 stencil points. Numerical solutions of the Lax-Wendroff method in graphical form are below.

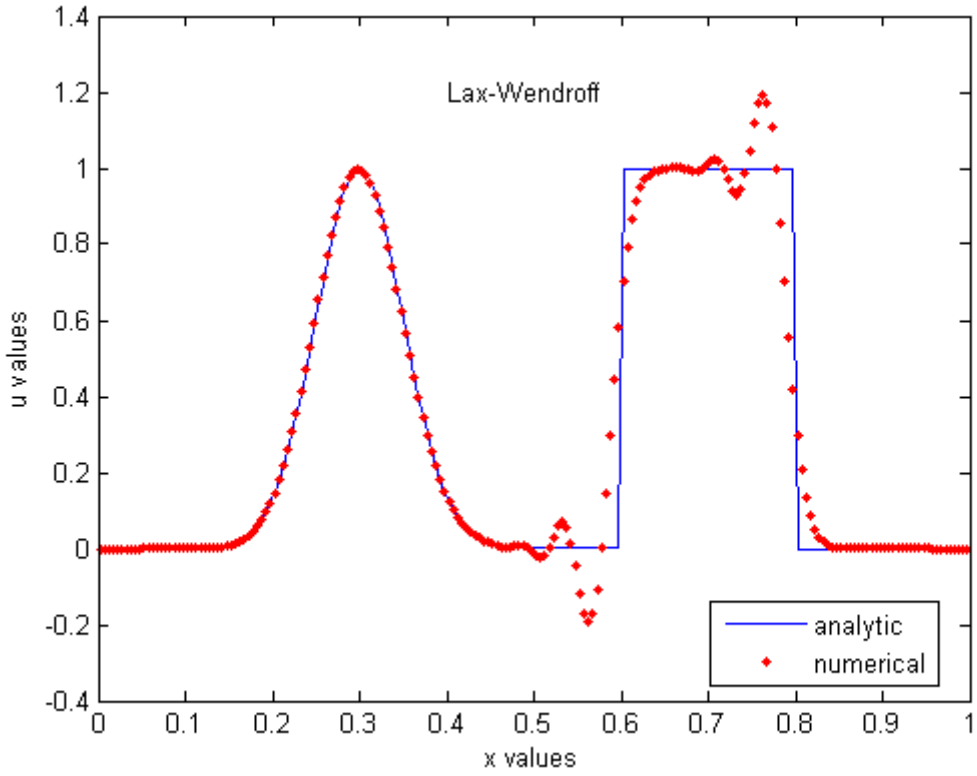

Figure 4. 3 Lax-Wendroff method applied to the test problem (4.1)-(4.3) at time t=1.0

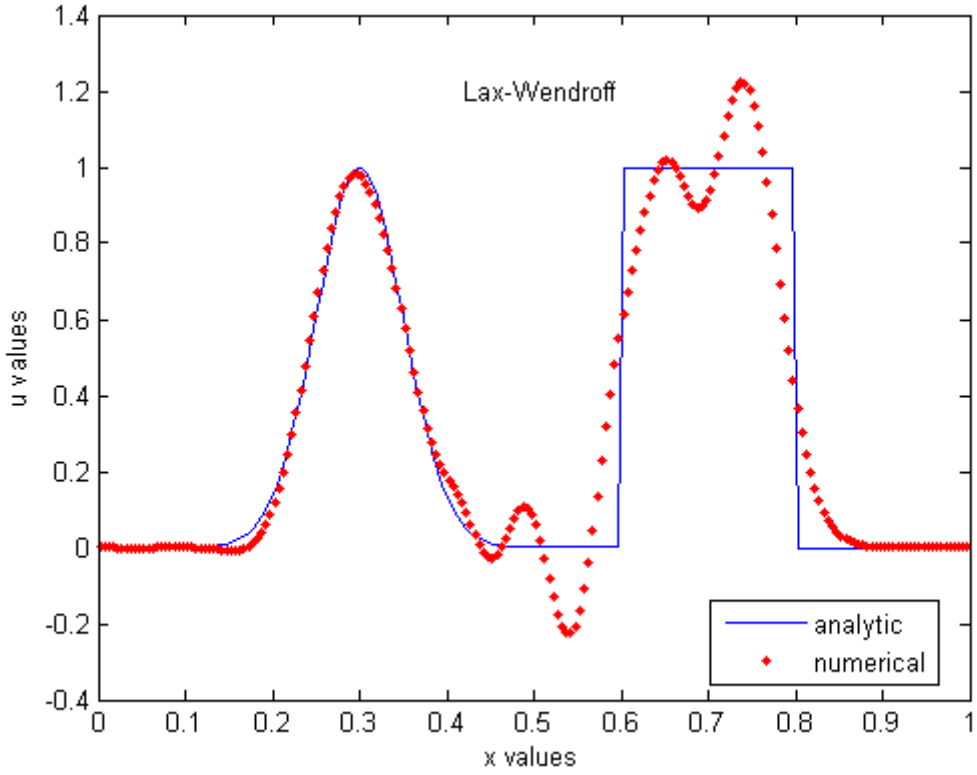

Figure 4. 4 Lax-Wendroff method applied to the test problem (4.1)-(4.3) at time t=5.0

When we analyze the figures to compare with the upwind method, we see that the smooth pulse, namely Gaussian hump, captured much better than the upwind method. In terms of square wave, the Lax-Wendroff method also has problem, it cause oscillations around discontinuities and these oscillations appear behind of the discontinuities. One more note on the Lax-Wendroff method is that it has a phase error. That is, it causes a slight shift in the location of the Gaussian hump. This is clearer in figure 4.4.

4.1.3 The Beam-Warming Method

In order to reach the Lax-Wendroff method, we use the central difference approximation in (4.10) for the spatial derivative. That is why; the method is a centered method. Either c is positive or negative, we can use this scheme. However if we know that $c>0$, than it may be logical to use a one-sided formula to get more correct answer. Instead of central difference approximation, if we use

$$\begin{aligned} v_x\left(i\Delta x, n\Delta t\right) &= \frac{1}{2\Delta x}\left(3v_i^n - 4v_{i-1}^n + v_{i-2}^n\right) + O\left(\Delta x^2\right), \\ v_{xx}\left(i\Delta x, n\Delta t\right) &= \frac{1}{\left(\Delta x\right)^2}\left(v_i^n - 2v_{i-1}^n + v_{i-2}^n\right) + O\left(\Delta x\right). \end{aligned} \tag{4.14}$$

in the equation (4.10), we have

$$u_i^{n+1} = u_i^n - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(3u_i^n - 4u_{i-1}^n + u_{i-2}^n\right) + \frac{1}{2}\left(\frac{c\Delta t}{\Delta x}\right)^2\left(u_i^n - 2u_{i-1}^n + u_{i-2}^n\right). \tag{4.15}$$

This method is called as the Beam-Warming method [7]. This method is again second order accurate and stable for $0 \le cfl \le 2$. (Remember that this scheme and stability conditions are valid for $c>0$, for the case $c<0$, it is easy to derive the scheme and its stability conditions.) Similar to Lax-Wendroff, this method also has 3 stencil points.

When we apply the method to the test problem (4.1)-(4.3) we have the following graphs.

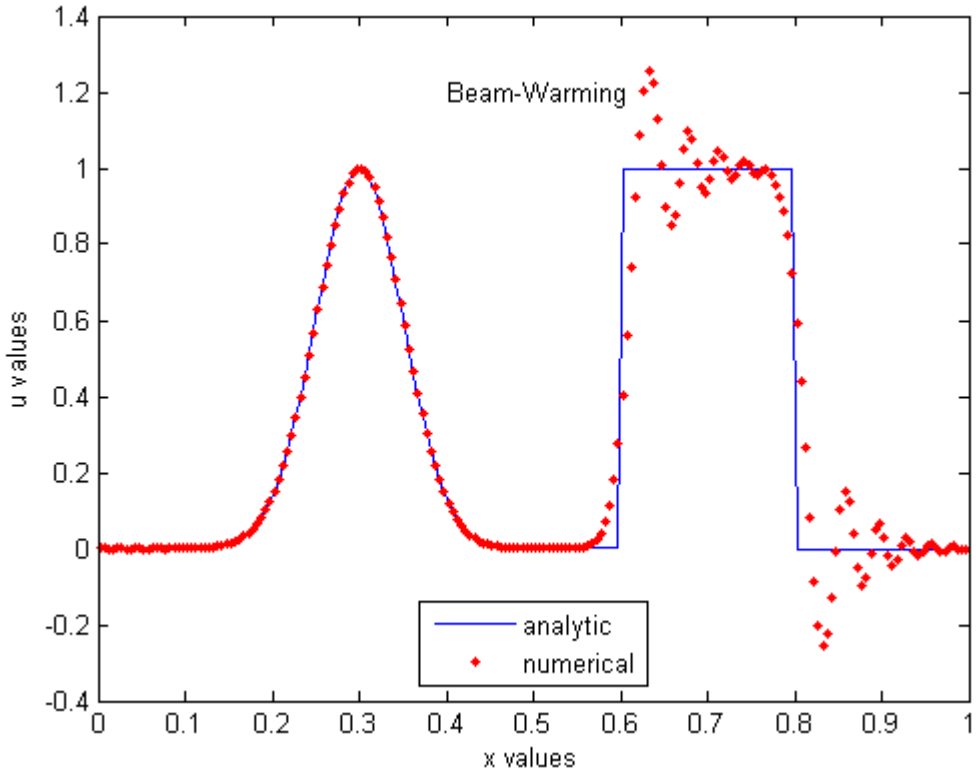

Figure 4. 5 Beam-Warming method applied to the test problem (4.1)-(4.3) at time t=1.0

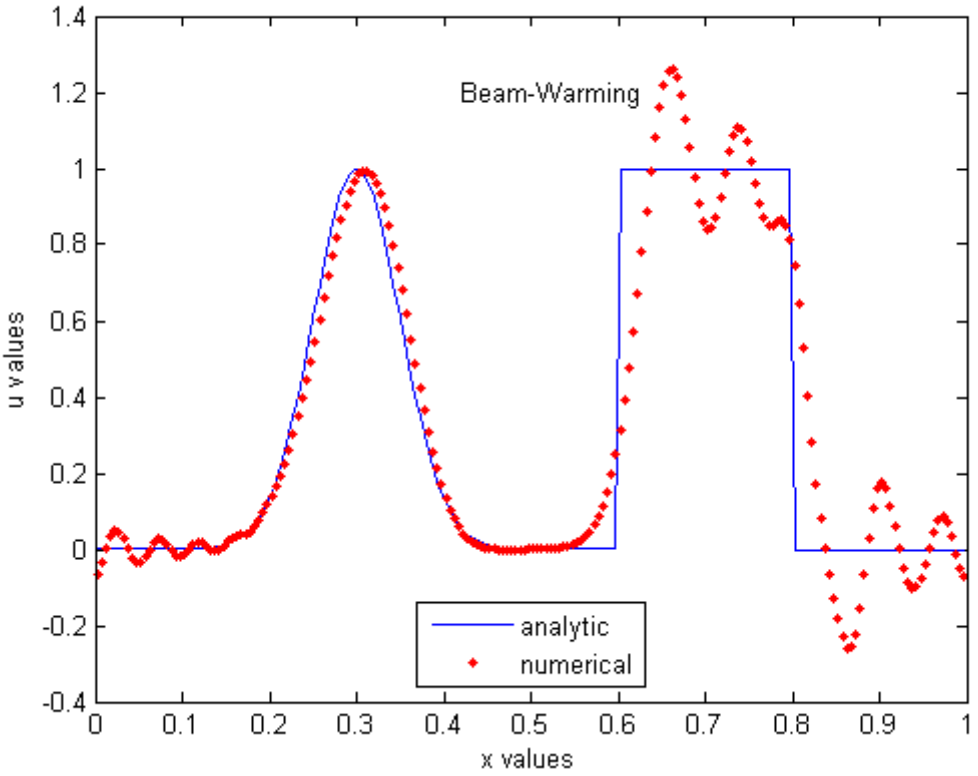

Figure 4. 6 Beam-Warming method applied to the test problem (4.1)-(4.3) at time t=5.0

We can notice that the Beam-Warming method also oscillates around the discontinuities like the Lax-Wendroff method. Furthermore it has oscillations at the beginning of the graph, especially at time t=5.

4.1.4 Fromm's Method

Another second-order method is the Fromm's method. It has the following formula [6].

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{4\Delta x}\left(u_{i+1}^n + 3u_i^n - 5u_{i-1}^n + u_{i-2}^n\right) - \frac{1}{4}\left(\frac{c\Delta t}{\Delta x}\right)^2\left(u_{i+1}^n - u_i^n - u_{i-1}^n + u_{i-2}^n\right) \tag{4.16}$$

Unlike the Lax-Wendroff and Beam-Warming, the Fromm's method has four stencil points. The results obtaining by the application of Fromm's method to the test problem (4.1)-(4.3) are bellowing.

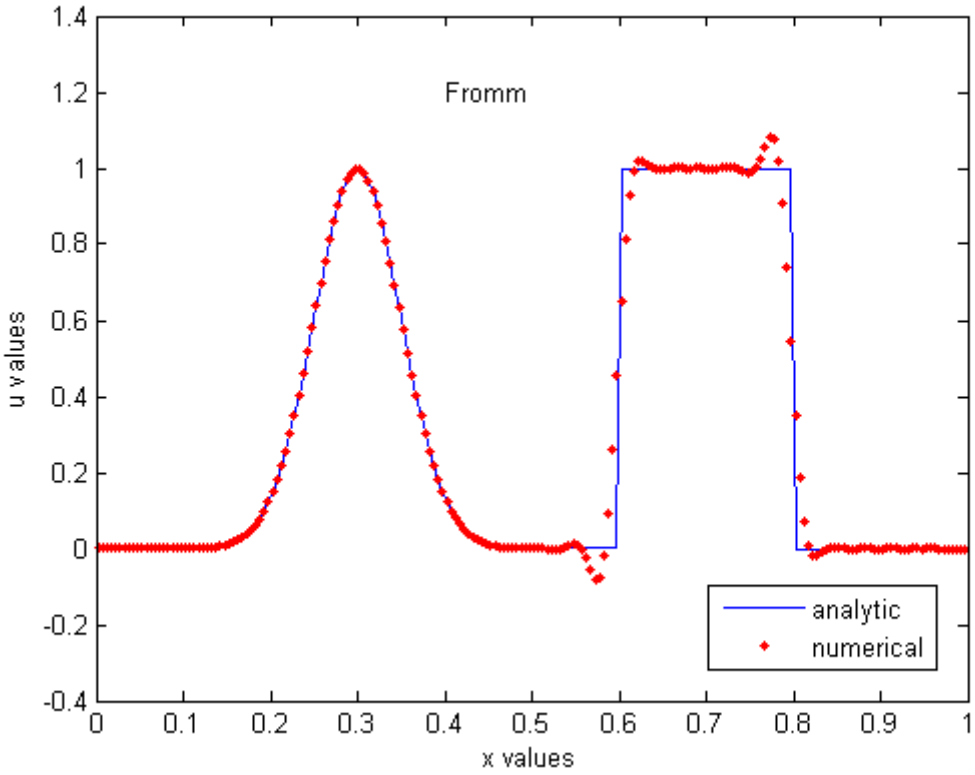

Figure 4. 7 Fromm's method applied to the test problem (4.1)-(4.3) at time t=5.0

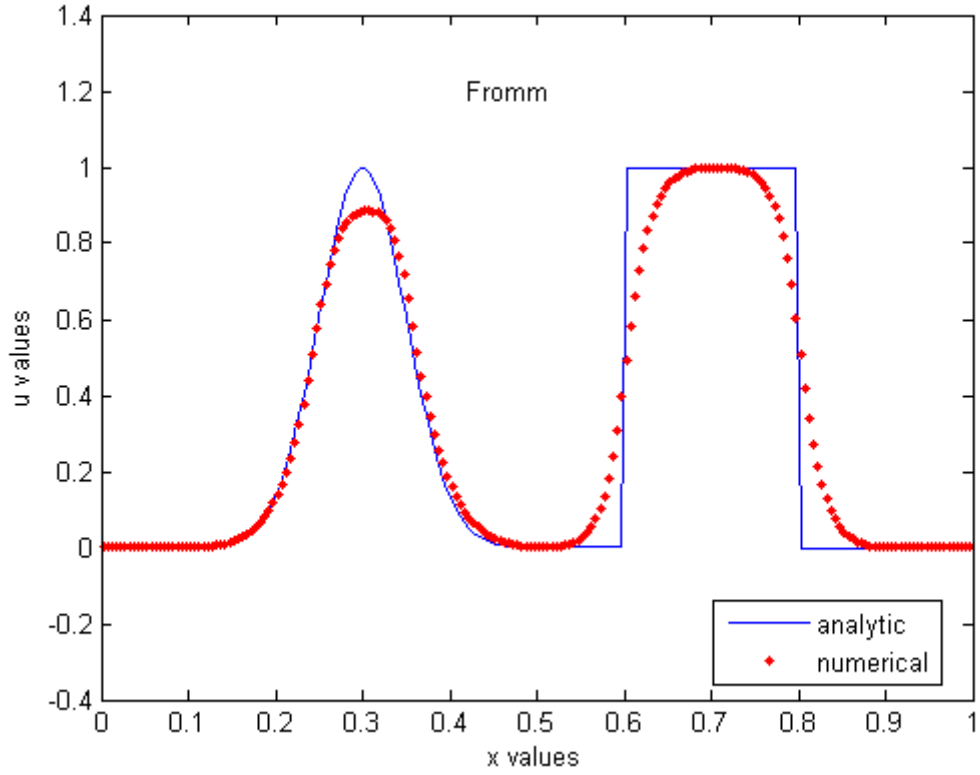

Figure 4. 8 Fromm's method applied to the test problem (4.1)-(4.3) at time t=5.0

In terms of steep gradient, Fromm's method has interesting result. While it has an spurious oscillation around discontinuity at time $t=1.0$, it vanishes at later time ($t=5.0$). Expectedly, the accuracy for the smooth data decreases as time evolves. Comparing to the previous two second-order methods, Fromm's method is quite well in terms of oscillations.

The characteristic of high resolution schemes is actually choosing the advantages of methods and combining them to obtain more sufficient method. For instance, if it is possible, we reach second order accuracy, but we do not insist on it if the solution does not behave smoothly for some region. To achieve this, let have a look at the REA algorithm once more.

4.2 The REA Algorithm Revisited

Remember that we gave the REA algorithm in section 3.4. By constructing a constant piecewise function $\tilde{v}^n(x,t)$ from the cell average u_i^n, we obtain the first order Godunov type method, the upwind method, to solve the one way wave equation. To improve the

accuracy, we must utilize a better reconstruction for $\tilde{v}^n(x,t)$ then a piecewise constant data, namely value of cell average. We can build a piecewise linear function from u_i^n which has the form

$$\tilde{v}^n(x,t_n) = u_i^n + \sigma_i^n(x - x_i) \quad \text{for } x_{i-1/2} \le x \le x_{i+1/2}, \tag{4.17}$$

where

$$x_i = \frac{1}{2}\left(x_{i-1/2} + x_{i+1/2}\right). \tag{4.18}$$

is the center of cell C_i and σ_i^n is the slope on the same grid cell. The features of this definition are that its value at the point x_i is equal to the value of cell average, u_i^n and the average value of this linear piecewise function $\tilde{v}^n(x,t)$ over C_i is equal again to the value of cell average, u_i^n. The latter property of this reconstruction is very important in building up new conservative methods for conservation laws.

Now using this reconstruction, let build the REA algorithm again for scalar advection equation. Without loss of generality, let assume that c in the equation $v_t + cv_x = 0$ is positive and also assume that $c\Delta t/\Delta x \le 1$ which is necessary to hold the convergence. Then reconstruction of the REA algorithm, the upwind method, becomes

$$u_i^{n+1} = \left(u_i^n - \frac{1}{2}c\Delta t\sigma_i^n\right) - \frac{c\Delta t}{\Delta x}\left[\left(u_i^n - \frac{1}{2}c\Delta t\sigma_i^n\right) - \left(u_{i-1}^n + \frac{1}{2}(\Delta x - c\Delta t)\sigma_{i-1}^n\right)\right]. \tag{4.19}$$

Rearranging this, we obtain

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right) - \frac{1}{2}\frac{c\Delta t}{\Delta x}(\Delta x - c\Delta t)\left(\sigma_i^n - \sigma_{i-1}^n\right). \tag{4.20}$$

Notice that this is the upwind method, but together with a term that relies on the slopes.

4.3 Limiters

In order to get rid of the drawbacks of methods that have been mentioned up to now, we can use the limiters. Limiters enable us to eliminate phase error, and to dispose of oscillations. To understand the concept of limiters, we will begin by analyzing the upwind method flux and the Lax-Wendroff flux. For the upwind method in (4.4) (remember that we assume $c > 0$), we can write the flux as

$$F_{i-1/2}^{n} = cu_{i-1}^{n}. \tag{4.21}$$

For the Lax-Wendroff method in (4.13), the flux is

$$F_{i-1/2}^{n} = \frac{c}{2}\left(u_{i-1}^{n} + u_{i}^{n}\right) - \frac{c^2}{2}\frac{\Delta t}{\Delta x}\left(u_{i}^{n} - u_{i-1}^{n}\right). \tag{4.22}$$

If we rewrite the equation (4.22) with arranging the terms, we have

$$F_{i-1/2}^{n} = cu_{i-1}^{n} + \frac{c}{2}\left(1 - \frac{c\Delta t}{\Delta x}\right)\left(u_{i}^{n} - u_{i-1}^{n}\right). \tag{4.23}$$

Notice that, the flux in (4.23) has the form of the upwind flux with an additional term. This term can be considered as the correction term for the upwind method. we should be aware that although the correction term in (4.23) resembles a diffusive flux since it relies on $u_{i}^{n} - u_{i-1}^{n}$, it is an anti-diffusive flux because the coefficient is positive when the *cfl* condition is satisfied [1]. By anti-diffusive flux, we mean that it has sharpening influence for very diffusive upwind methods. That is why; the Lax-Wendroff method has oscillations even for the smooth data. In order to prevent these wiggles we should modify the correction term by using some form of limiter. This limiter changes the magnitude of correction by taking into account of the behavior of solution. As a result, when upgrading methods from first-order to second-order, we can prevent oscillations thanks to limiter.

4.4 Different Slopes

When we build the REA algorithm with piecewise linear function, the limiting process can be considered as the limiting the slope. For the equation (4.20), if we choose the slope as zero, this means that we construct the REA algorithm with piecewise constant functions and this gives the upwind method. In fact, if we put zero for the slope in equation (4.20), we will get the equation (4.4) which is exactly the upwind method. To reach a second-order accurate method, we should use nonzero slope, σ_{i}^{n}, and this slope should approximate the derivative of $v(x,t)$ over the C_i. When we choose

$$\sigma_{i}^{n} = \frac{u_{i+1}^{n} - u_{i}^{n}}{\Delta x}, \tag{4.24}$$

and put it in the equation (4.20), we have

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right) - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(\Delta x - c\Delta t\right)\left(\frac{u_{i+1}^n - u_i^n}{\Delta x} - \frac{u_i^n - u_{i-1}^n}{\Delta x}\right). \tag{4.25}$$

Rearranging the above equation gives

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{2\Delta x}\left(u_{i+1}^n - u_{i-1}^n\right) + \frac{1}{2}\left(\frac{c\Delta t}{\Delta x}\right)^2\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right), \tag{4.26}$$

which is the Lax-Wendroff method. This means that the Lax-Wendroff is the Godunov type method with second-order accuracy.

Similarly, if we choose the slope as

$$\sigma_i^n = \frac{u_i^n - u_{i-1}^n}{\Delta x} \tag{4.27}$$

and apply it to the equation (4.20), we get

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right) - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(\Delta x - c\Delta t\right)\left(\frac{u_i^n - u_{i-1}^n}{\Delta x} - \frac{u_{i-1}^n - u_{i-2}^n}{\Delta x}\right). \tag{4.28}$$

Rearranging this yields

$$u_i^{n+1} = u_i^n - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(3u_i^n - 4u_{i-1}^n + u_{i-2}^n\right) + \frac{1}{2}\left(\frac{c\Delta t}{\Delta x}\right)^2\left(u_i^n - 2u_{i-1}^n + u_{i-2}^n\right). \tag{4.29}$$

This is the Beam-Warming method. we can say then, that the Beam-Warming method is also Godunov type method.

These are the natural choice that first comes into mind when we try to approximate the $v_x(x,t)$. Other choices give other methods, but we will not mention about them.

Now, we will analyze why the second-order Godunov type methods give oscillatory approximations to solution near discontinuities, by keeping the REA algorithm in mind. Let us consider the Beam-Warming method applied to piecewise constant initial condition

$$u_i^0 = \begin{cases} 1 & \text{if } i < k, \\ 0 & \text{if } i \geq k. \end{cases} \tag{4.30}$$

When we calculate the slope according to (4.27), we will get piecewise linear function shown in figure 4.9(i). The slope is zero for all values of i except for $i = k$. The

constructed function $\tilde{v}(x,t_n)$ has an undershoot with a minimum value of -0.5 regardless of spatial step size. For the next time step, when we compute the average of cell C_{k+1} (Figure 4.9(ii)), the value will be less than 0 for any Δt with $0 < c\Delta t < \Delta x$ and this will cause oscillation. Furthermore, since the slope for C_{k+1} becomes negative it triggers the cell C_{k+2} to became negative also. Therefore oscillation will spread out the cells. Also with time, this oscillation will grow.

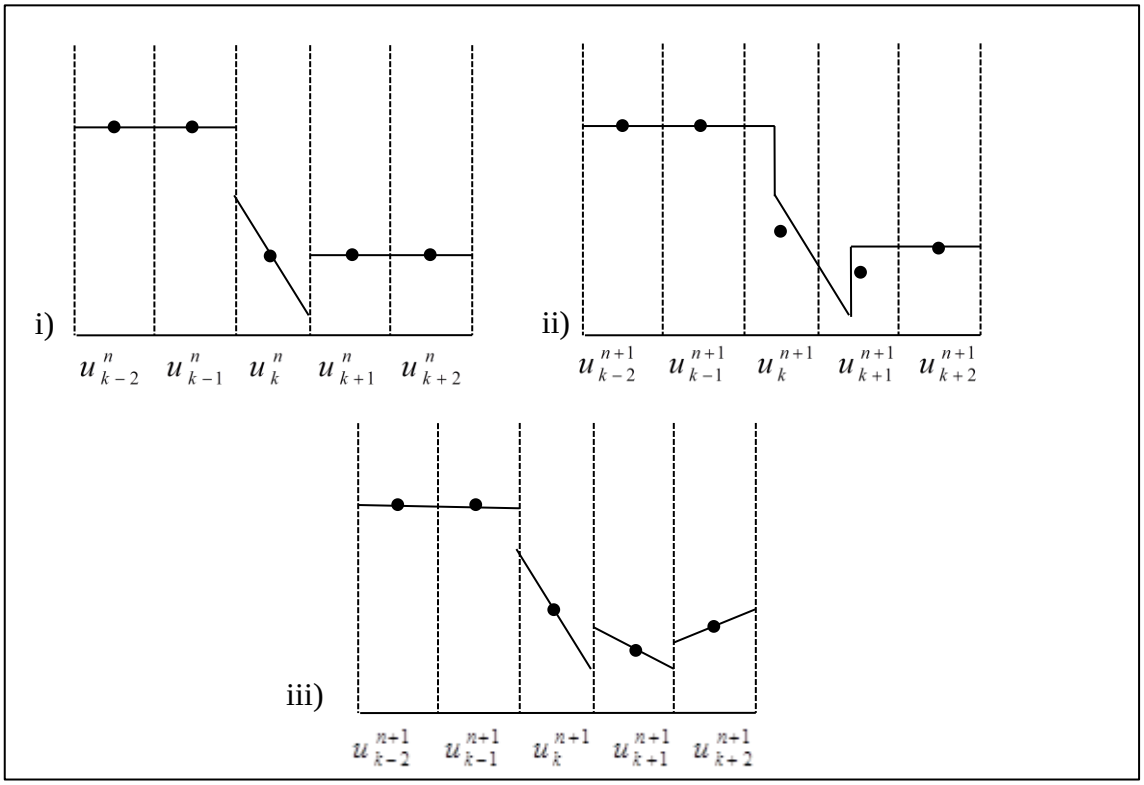

Figure 4. 9 i) Construction of $\tilde{v}(.,t_n)$ from cell averages by Beam-Warming slope. ii) Δt time later. iii) New cell averages (dots) and reconstruction of $\tilde{v}(.,t_{n+1})$

We saw that near a discontinuity it does not improve the accuracy to introduce slope that mentioned above. Moreover, if we want to prevent nonphysical oscillations we should not introduce any nonzero slope for the *k*th cell because any slope $\sigma_k^n < 0$ causes $u_{k+1}^{n+1} < 0$ and so oscillations (note that positive slope is meaningless for this cell). On the other hand, if we set the entire slope to 0, then we have just first-order accuracy. This is what we do not want where the solution is smooth. In addition to this, constructing nonzero slope can enable to prevent solution from smearing out too far and enable discontinuity to become sharp effectively.

When choosing our formula for slope σ_i^n, if we take into account that how the solution is behaving around the discontinuity then we can get rid of the oscillation, we can derive fairly sharp solution to approximate the discontinuity and for smooth solution we can get second-order accuracy. For smooth solution, we want to choose something like the Beam-Warming slope. Around a discontinuity, to prevent appearance of oscillation, we want to limit this slope by using a smaller value in magnitude. Methods building up from this opinion are called as slop-limiter methods.

4.5 Advanced High Resolution Methods

4.5.1 Minmod Slope-Limiter Method

We saw in the previous section that the Lax-Wendroff and the Beam-Warming methods are the second type Godunov method with downwind slope defined in (4.24) and upwind slope defined in (4.27). The minmod slope method is also second type Godunov method but it is also slope-limiter methods that is mentioned above. For this method we define the slope as follows

$$\sigma_i^n = \min \mathrm{mod}\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}, \frac{u_{i+1}^n - u_i^n}{\Delta x} \right) \text{[9]}. \tag{4.31}$$

Here, the Minmod function is defined by

$$\min \mathrm{mod}(x, y) = \begin{cases} x & \text{if } |x| < |y| \text{ and } xy > 0, \\ y & \text{if } |y| < |x| \text{ and } xy > 0, \\ 0 & \text{if } xy \le 0. \end{cases} \tag{4.32}$$

We can consider the minmod function as follow: if x and y have the same sign, then the minmod function returns to the one that is smaller in absolute value. If x and y have different sign then it returns to 0.

The Lax-Wendroff scheme uses always downwind slope and the Beam-Warming scheme utilize always upwind slope. On the other hand the minmod slop method compares the two slopes and selects the one that is smaller in modulus. If the two values have different sign, it choose the zero slope. This is logical since if the two slopes have different sign, it means that there is a local minimum or local maximum of solution (Remember that for the local minimum or maximum, v_x is zero).

When we apply the method to the test problem (4.1)-(4.3), we have following graphical results.

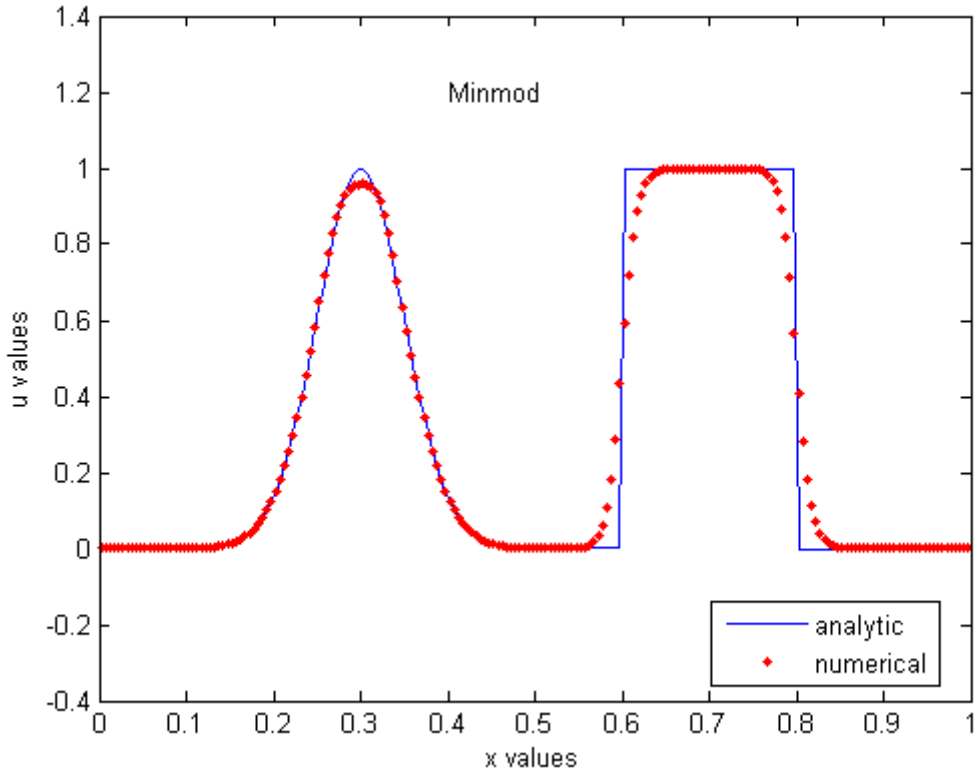

Figure 4. 10 Minmod slope-limiter method applied to the test problem (4.1)-(4.3) at time $t = 1.0$

When we compare this graph with the graph of previous methods, we can say that the minmod method is really better than the previous ones. The accuracy of minmod method is at least as good as the accuracy of previous ones for smooth hump while for the square wave; it is perfect in terms of capturing the discontinuity compared to the Lax-Wendroff and Beam-Warming. For the long term evaluation, it become worse, but it is acceptable and understandable. Note that still there is no unphysical oscillation in the numerical solution.

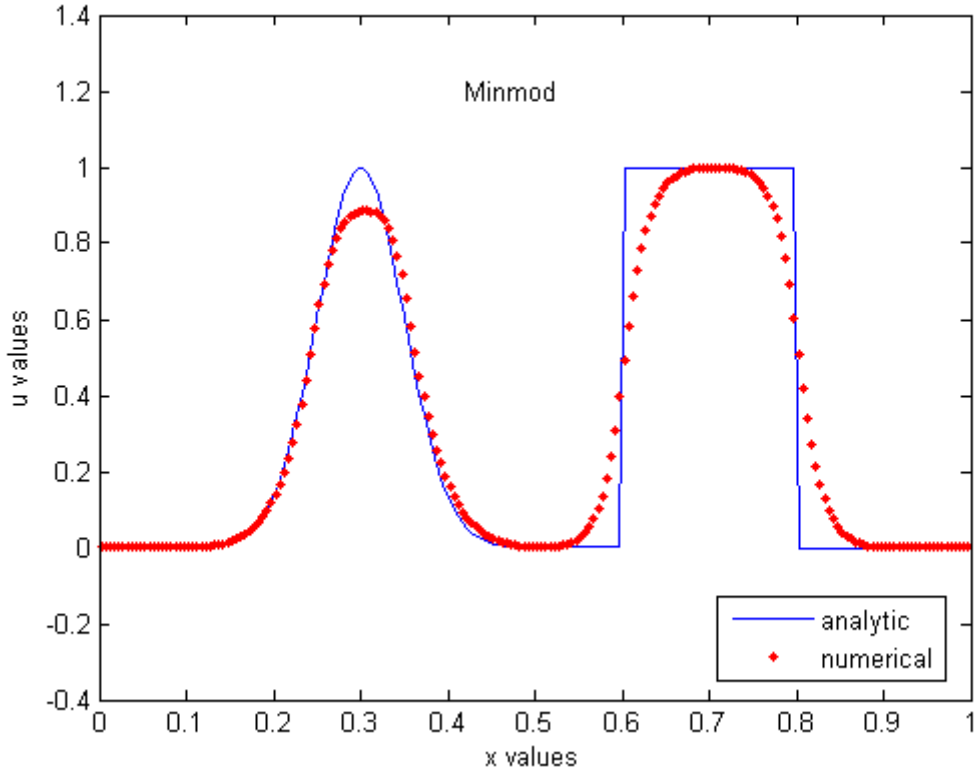

Figure 4. 11 minmod slope-limiter method applied to the test problem (4.1)-(4.3) at time t=5.0

4.5.2 Superbee Slope-Limiter Method

One of the other slope limiter methods, which is also second type of Godunov method, is the superbee limiter method. In addition to the minmod method, this method is also second order accuracy for smooth solutions. The method is introduced by Roe [14] and has the following slope.

$$\sigma_i^n = \max \mathrm{mod}(\mathrm{x}, \mathrm{y}), \tag{4.33}$$

where

$$\begin{aligned} x &= \min \mathrm{mod}\left(\left(\frac{u_{i+1}^n - u_i^n}{\Delta x}\right), 2\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}\right)\right) \\ y &= \min \mathrm{mod}\left(2\left(\frac{u_{i+1}^n - u_i^n}{\Delta x}\right), \left(\frac{u_i^n - u_{i-1}^n}{\Delta x}\right)\right) \end{aligned}. \tag{4.34}$$

We can view this slope as in that way: downwind slope is compared with twice the upwind slope in terms of minmod and vice versa. From this process we have two values and we select the value that has bigger in magnitude.

Results of application of this method to the test problem (4.1)-(4.3) are below.

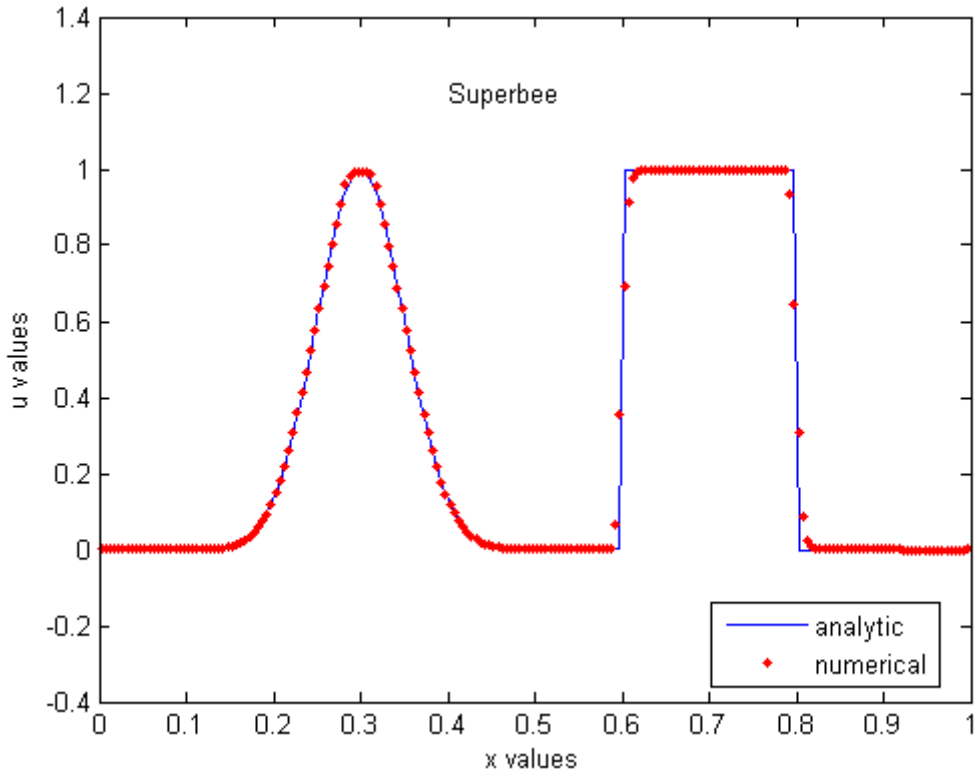

Figure 4. 12 Superbee slope-limiter method applied to the test problem at time t=1.0

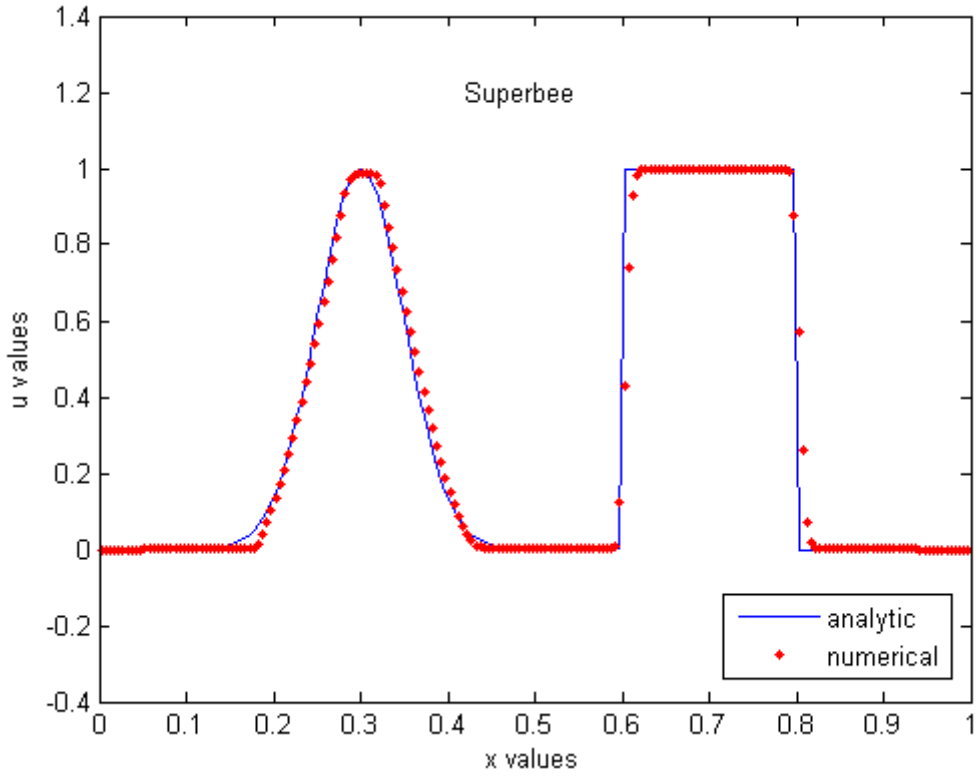

Figure 4. 13 Superbee slope-limiter method applied to the test problem at time t=5.0

We can derive from figure 4.12 and figure 4.13 that the superbee slope-limiter method is much better compared to the minmod slope-limiter method in terms of amplitude of the solution and catching the discontinuity. On the other hand, for the Gaussian hump, it gives like a horizontal line where the solution resembles a concave curve. This may be problematic especially if the solution has inflection points [1].

4.5.3 Van Leer Slope-Limiter Method

In 1974, Bram van Leer published a paper and he introduced a new method. we will give the formulation after the section 4.6. The graphs of van Leer method solution to the test problem are the following figures.

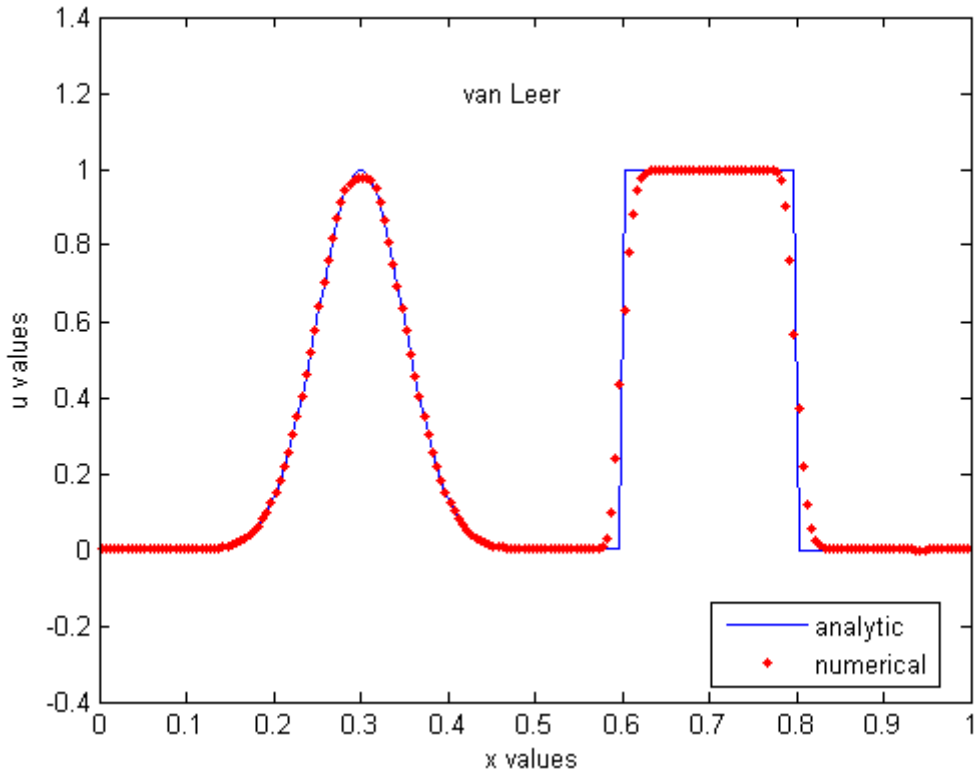

Figure 4. 14 Van Leer method applied to (4.1)-(4.3) at time t=1.0

In figure 4.14, we see that van Leer method captures the solution as well as superbee method. However, for long term evaluation (figure 4.15), accuracy of van Leer method decreases much compared to superbee method.

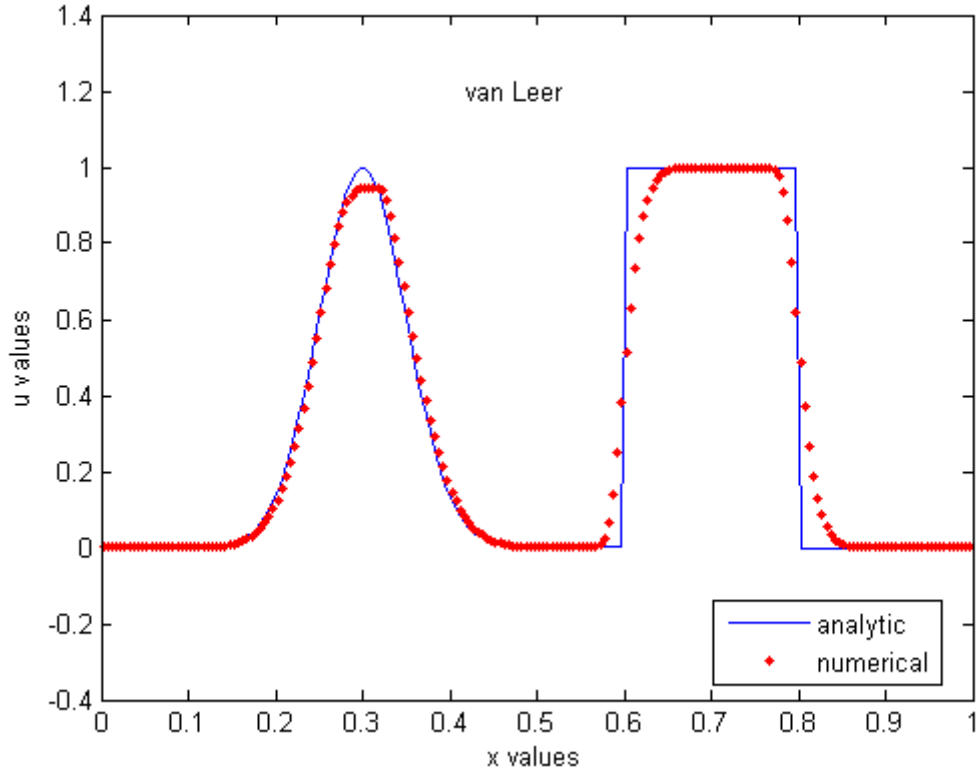

Figure 4. 15 Van Leer method applied to (4.1)-(4.3) at time t=5.0

4.5.4 MC Slope-Limiter Method

The MC slope-limiter which is short name of monotonized central-difference limiter and introduced by van Leer [12] has the following slope:

$$\sigma_i^n = \min \operatorname{mod}\left(\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right), 2\left(\frac{u_{i+1}^n - u_i^n}{\Delta x}\right), 2\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}\right)\right). \tag{4.35}$$

We can interpret this slope as comparing the three values, the central difference, two times the upwind slope and twice the downwind slope, along each other and taking the one that minimum in absolute value if the three values have the same sign. Otherwise, slope becomes zero. Applying of the method to the test problem (4.1)-(4.3), we have the following graphs.

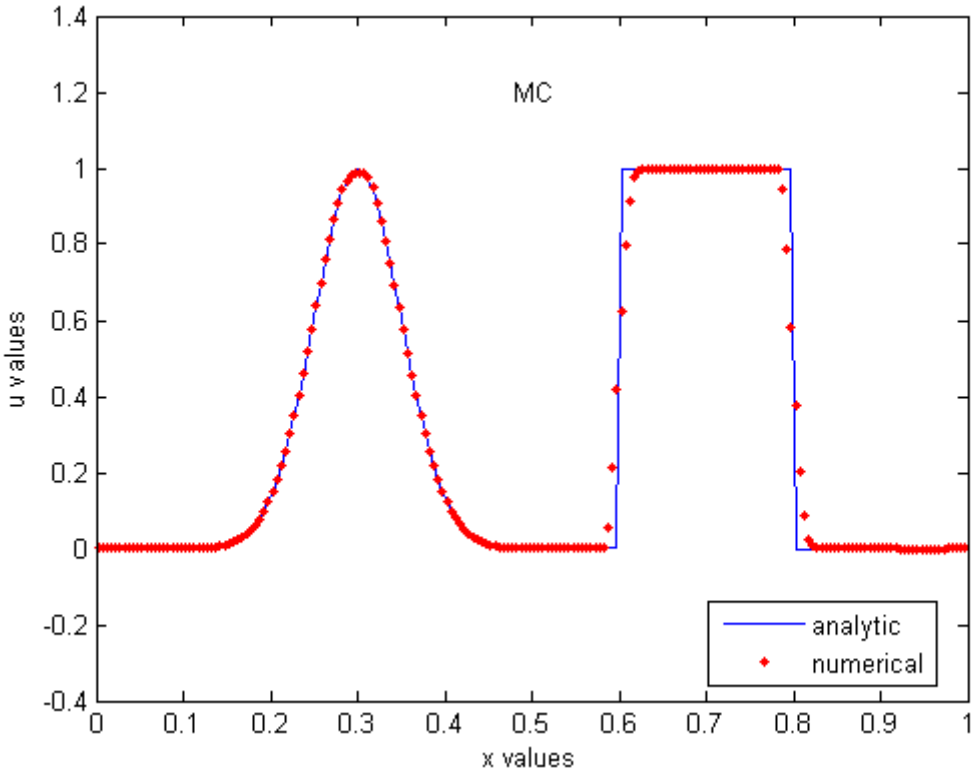

Figure 4. 16 MC slope-limiter method applied to the test problem at time t=1.0

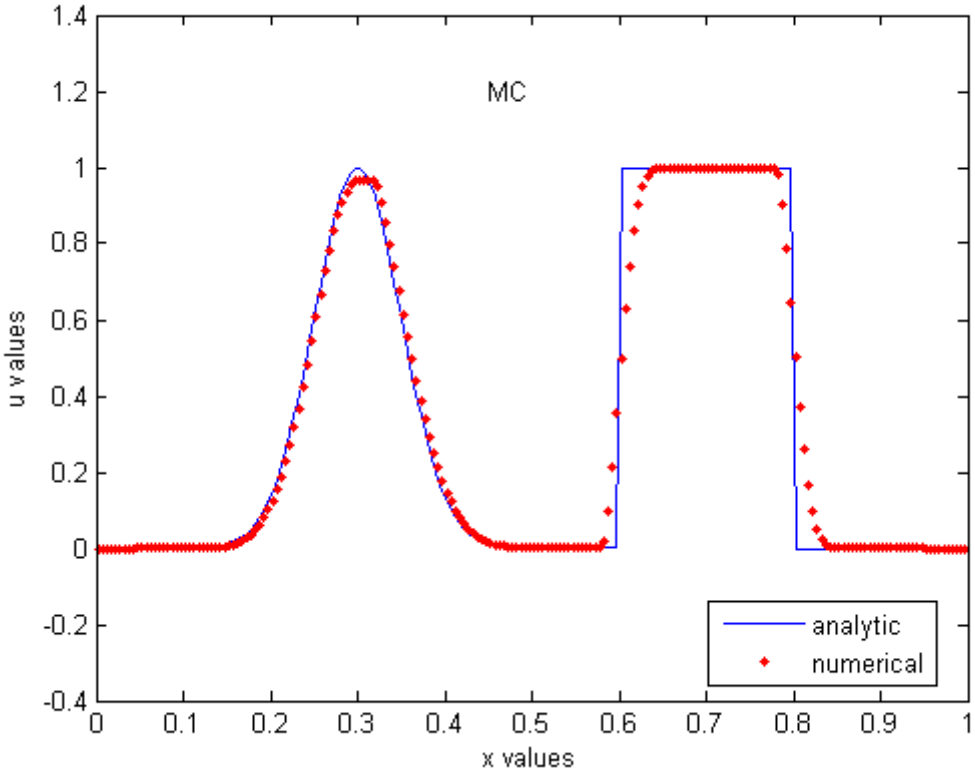

Figure 4. 17 MC slope-limiter method applied to the test problem at time t=5.0

MC slope-limiter method has the similar graph with the superbee slope-limiter method. Contrary to similarity, it has also diversities some of which are good and some of others are bad. The disadvantage of MC compared to superbee is that, it is less sharp around the discontinuity. On the other hand, as the advantage of MC, it resolves the default of superbee which is becoming squared off at the top of the smooth hump.

4.6 Flux-Differencing Form of Methods

The methods we have seen up to now can be also written in the form of flux-differencing defined in (3.19). Writing in such a form enables us to conclude that the methods are in conservation laws form. We can do this issue by algebraically manipulating the equation (4.20) to find the flux function or by computing the flux at the interface using the piecewise linear reconstruction. Both ways will give the same flux function. For the advection equation with $c > 0$, we have

$$F_{i-1/2}^n = cu_{i-1}^n + \frac{1}{2}c\left(\Delta x - c\Delta t\right)\sigma_{i-1}^n \text{ [1].} \tag{4.36}$$

Using this function in the flux-differencing formula (3.19), we have

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right) - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(\Delta x - c\Delta t\right)\left(\sigma_i^n - \sigma_{i-1}^n\right). \tag{4.37}$$

Notice that this is the same with the equation (4.20) expectedly. If we write the flux function for positive and negative *c*, we get

$$F_{i-1/2}^n = \begin{cases} cu_{i-1}^n + \frac{1}{2}c\left(\Delta x - c\Delta t\right)\sigma_{i-1}^n & \text{if } c \geq 0 \\ cu_i^n - \frac{1}{2}c\left(\Delta x + c\Delta t\right)\sigma_i^n & \text{if } c \leq 0 \end{cases}. \tag{4.38}$$

Here, we write the flux function with slope, σ_i^n for the cell C_i. However, writing the methods in terms of flux function, it is more logical to correlate our approximation to $v_x\left(x,t\right)$ with the cell interface at $x_{i-1/2}$ rather than the cell C_i because we define the flux $F_{i-1/2}^n$ at the cell edge $x_{i-1/2}$. If we define the jump between two successive cells as

$$\Delta u_{i-1/2}^n = u_i^n - u_{i-1}^n, \tag{4.39}$$

and if we divide this difference by Δx, then we attain an approximation to $v_x\left(x,t\right)$. Therefore we can rewrite the flux in (4.36) as

$$F_{i-1/2}^{n} = cu_{i-1}^{n} + \frac{1}{2}c\left(1 - \frac{c\Delta t}{\Delta x}\right)\delta_{i-1/2}^{n}. \tag{4.40}$$

Here, $\delta_{i-1/2}^{n}$ is the function of $\Delta u_{i-1/2}^{n}$. (Notice that this flux formula is for $c > 0$, for the negative *c*, it can be rewritten easily.)

Now, for $\delta_{i-1/2}^{n} = \Delta u_{i-1/2}^{n}$, (4.40) gives the Lax-Wendroff flux function and so the Lax-Wendroff method. Since this is the basic selection for $\delta_{i-1/2}^{n}$, we can say that the Lax-Wendroff method is the fundamental second-order method depended on piecewise linear reconstruction. Furthermore, some other choices of $\delta_{i-1/2}^{n}$ give the other methods some of which are our methods that we have mentioned. Therefore, we can consider the slope-limiter methods as also flux-limiter methods.

Table 4. 1 Flux-limiter function of the methods [1]

Name of Method	**Flux-Limiter Function**
Upwind	$\phi(\theta) = 0$
Lax-Wendroff	$\phi(\theta) = 1$
Beam-Warming	$\phi(\theta) = \theta$
Fromm	$\phi(\theta) = (1+\theta)/2$
Minmod slope-limiter	$\phi(\theta) = \min \mathrm{mod}(1, \theta)$
Superbee slope-limiter	$\phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta))$
MC slope-limiter	$\phi(\theta) = \max\left(0, \min\left((1+\theta)/2, 2, 2\theta\right)\right)$
van Leer	$\phi(\theta) = \left(\theta + \lvert\theta\rvert\right)/(1+\theta)$

If we define $\delta_{i-1/2}^{n}$ as

$$\delta_{i-1/2}^{n} = \phi\left(\theta_{i-1/2}^{n}\right)\Delta u_{i-1/2}^{n}, \tag{4.41}$$

where

$$\theta_{i-1/2}^{n}=\begin{cases}\Delta u_{i-1-1/2}^{n}\big/\Delta u_{i-1/2}^{n} & \text{if } c>0\\ \Delta u_{i+1/2}^{n}\big/\Delta u_{i-1/2}^{n} & \text{if } c<0\end{cases}, \tag{4.42}$$

then for $\phi(\theta)=0$ we have

$$\delta_{i-1/2}^{n}=0\Rightarrow F_{i-1/2}^{n}=cu_{i-1}^{n}. \tag{4.43}$$

This is the flux of upwind method for positive *c*. Similarly, if we choose $\phi(\theta)=1$, we have

$$\delta_{i-1/2}^{n}=\Delta u_{i-1/2}^{n}\Rightarrow F_{i-1/2}^{n}=cu_{i-1}^{n}+\frac{1}{2}c\left(1-\frac{c\Delta t}{\Delta x}\right)\Delta u_{i-1/2}^{n}, \tag{4.44}$$

which is the Lax-Wendroff flux function for $c>0$. Table 4.1 demonstrates which choice of flux-limiter function corresponds to which methods that we have discussed so far.

We can write conservation form of the methods for negative and positive speed as follows.

$$u_{i}^{n+1}=u_{i}^{n}-\frac{\Delta t}{\Delta x}\left(A^{+}\Delta u_{i-1/2}^{n}+A^{-}\Delta u_{i+1/2}^{n}\right)-\frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2}-\tilde{F}_{i-1/2}\right), \tag{4.45}$$

where $A^{+}\Delta u_{i-1/2}^{n}$ is the net effect of right going waves and $A^{-}\Delta u_{i+1/2}^{n}$ is the net effect of left going waves. That is,

$$A^{+}\Delta u_{i-1/2}^{n}=c^{+}\left(u_{i}^{n}-u_{i-1}^{n}\right), \tag{4.46}$$

and

$$A^{-}\Delta u_{i+1/2}^{n}=c^{-}\left(u_{i+1}^{n}-u_{i}^{n}\right). \tag{4.47}$$

Furthermore, we define $\tilde{F}_{i+1/2}$ and $\tilde{F}_{i-1/2}$ as

$$\tilde{F}_{i+1/2}=\frac{1}{2}|c|\left(1-\left|\frac{c\Delta t}{\Delta x}\right|\right)\delta_{i+1/2}^{n}, \tag{4.48}$$

$$\tilde{F}_{i-1/2}=\frac{1}{2}|c|\left(1-\left|\frac{c\Delta t}{\Delta x}\right|\right)\delta_{i-1/2}^{n}. \tag{4.49}$$

CHAPTER 5

APPLICATIONS TO BURGERS EQUATION

In previous chapter, we introduced the methods and evaluated them for advection equation which is linear and has scalar coefficient. For this chapter we will use the Burgers equation, which is nonlinear hyperbolic partial differential equation, as a test problem. We will give the result of methods in the graphical form.

5.1 Burgers Equation

There are two different versions of Burgers equation; one of them is the nonhomogeneous and nonlinear parabolic partial differential equation written as

$$v_t + vv_x = \varepsilon v_{xx}, \tag{5.1}$$

where $\varepsilon > 0$ and constant. This equation is generally called the viscid Burgers equation, since in fluid dynamics εv_{xx} corresponds to viscosity. The other equation is the homogenous and nonlinear hyperbolic partial differential equation and it can be stated as

$$v_t + vv_x = 0. \tag{5.2}$$

This is usually considered as the inviscid Burgers equation. (5.2) can also be written in the scalar conservation law form.

$$v_t + f(v)_x = 0, \tag{5.3}$$

where $f(v) = \frac{1}{2}v^2$.

The equation in (5.1) comes out usually as a simplification of a more sophisticated model. Thus, it is generally considered as a toy model. By saying toy model, we mean that it is a tool that is use to make clear some of the inside behavior of the general

problem. We can give the Navier Stokes equation for such a general problem as an example [27]. Moreover, Burgers equation models directly physical phenomena; one of them is the traffic flow [1], [27]. In numerical partial differential equation, it is also important due to the fact that its solution may contain discontinuities.

Note that there is a relationship between (5.1) and (5.2). When we take the limit of (5.1) as $\varepsilon \to 0$, we reach the equation (5.2). This approach is correct in the mathematical sense and also important for finding the approximate solution of inviscid Burgers equation. Be aware that the equation (5.2) has no analytic solution.

When we write the numerical methods for the Burgers equation, we will use the equation in (4.45). That is,

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(A^+ \Delta u_{i-1/2}^n + A^- \Delta u_{i+1/2}^n\right) - \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2} - \tilde{F}_{i-1/2}\right). \tag{5.4}$$

There are a few differences between scalar linear equation and nonlinear equation to write the equation in (5.4). First of all, for advection equation, the flux function is $f(v) = cv$. On the other hand, the flux function for Burgers equation is $f(v) = v^2/2$. Another difference is that for advection equation the characteristic speed is constant, but for Burgers equation it changes in time. We will define speed as follows.

$$s_{i-1/2} = \begin{cases} \left(f(u_i) - f(u_{i-1})\right)/u_i - u_{i-1} & \text{if } u_{i-1} \neq u_i \\ f'(u_i) & \text{if } u_{i-1} = u_i \end{cases} \text{ [1].} \tag{5.5}$$

Then $A^+ \Delta u_{i-1/2}^n$ and $A^- \Delta u_{i-1/2}^n$ become

$$A^+ \Delta u_{i-1/2}^n = s_{i-1/2}^+ \Delta u_{i-1/2}^n, \tag{5.6}$$

and

$$A^- \Delta u_{i-1/2}^n = s_{i-1/2}^- \Delta u_{i-1/2}^n. \tag{5.7}$$

As we can guess, $s_{i-1/2}^+ = \max\left(0, s_{i-1/2}\right)$ (and $s_{i-1/2}^- = \min\left(0, s_{i-1/2}\right)$). If we put this into (5.6) and do necessary calculations, we get

$$A^+ \Delta u_{i-1/2}^n = \begin{cases} f\left(u_i^n\right) - f\left(u_{i-1}^n\right) & \text{if } u_i^n > u_{i-1}^n \\ 0 & \text{if } u_i^n \leq u_{i-1}^n \end{cases}. \tag{5.8}$$

There is an important detail when defining fluctuations $A^{+}\Delta u_{i-1/2}^{n}$ and $A^{-}\Delta u_{i-1/2}^{n}$. If $f'(u_{i-1}) < 0 < f'(u_i)$ then fluctuations become

$$A^{+}\Delta u_{i-1/2}^{n} = \begin{cases} f(u_i^n) - f(v_s) & \text{if } u_i^n > u_{i-1}^n \\ 0 & \text{if } u_i^n \le u_{i-1}^n \end{cases}, \tag{5.9}$$

and

$$A^{-}\Delta u_{i-1/2}^{n} = \begin{cases} f(v_s) - f(u_{i-1}^n) & \text{if } u_i^n > u_{i-1}^n \\ 0 & \text{if } u_i^n \le u_{i-1}^n \end{cases}. \tag{5.10}$$

Here v_s is called stagnation point (or sonic point) and it is the value of *v* for which $u_{i-1} < v_s < u_i$ and $f'(v_s) = 0$. This modification for fluctuations is necessary due to entropy condition [1].

Now, we set $\tilde{F}_{i-1/2}$ as follows.

$$\tilde{F}_{i-1/2} = \frac{1}{2}\left|s_{i-1/2}\right|\left(1 - \frac{\Delta t}{\Delta x}\left|s_{i-1/2}\right|\right)\delta_{i-1/2}^{n}. \tag{5.11}$$

Remember that $\delta_{i-1/2}^{n}$ and other related concepts are defined in section 4.6. However, for the completeness of chapter we will give them again. $\delta_{i-1/2}^{n}$ is defined as

$$\delta_{i-1/2}^{n} = \phi\left(\theta_{i-1/2}^{n}\right)\Delta u_{i-1/2}^{n}, \tag{5.12}$$

where

$$\theta_{i-1/2}^{n} = \begin{cases} \Delta u_{i-1-1/2}^{n} \big/ \Delta u_{i-1/2}^{n} & \text{if } c > 0 \\ \Delta u_{i-1/2}^{n} \big/ \Delta u_{i-1/2}^{n} & \text{if } c > 0 \end{cases}. \tag{5.13}$$

We calculate the results of numerical methods for Burgers equation according to the table 5.1.

Table 5. 1 Flux-limiter function of the methods (revisited)

Name of Method	**Flux-Limiter Function**
Upwind	$\phi(\theta)=0$
Lax-Wendroff	$\phi(\theta)=1$
Beam-Warming	$\phi(\theta)=\theta$
Fromm	$\phi(\theta)=(1+\theta)/2$
Minmod slope-limiter	$\phi(\theta)=\min\text{mod}(1,\theta)$
Superbee slope-limiter	$\phi(\theta)=\max(0,\min(1,2\theta),\min(2,\theta))$
MC slope-limiter	$\phi(\theta)=\max\left(0,\min\left((1+\theta)/2,2,2\theta\right)\right)$
van Leer	$\phi(\theta)=\left(\theta+\lvert\theta\rvert\right)/(1+\theta)$

5.2 Numerical Results of the Methods

We will use the following equation, initial condition and boundary condition to test the methods.

$$v_t + f(v)_x = 0, \tag{5.14}$$

$$v(x,0)=\begin{cases}\min\left(0,(x-1.5)(x-2.5)\right) & \text{if } x\le 2.5\\ -v(5-x,0) & \text{if } x>2.5\end{cases}, \tag{5.15}$$

$$v(0,t)=v(5,t). \tag{5.16}$$

Graph of the initial data is shown in figure 5.1.

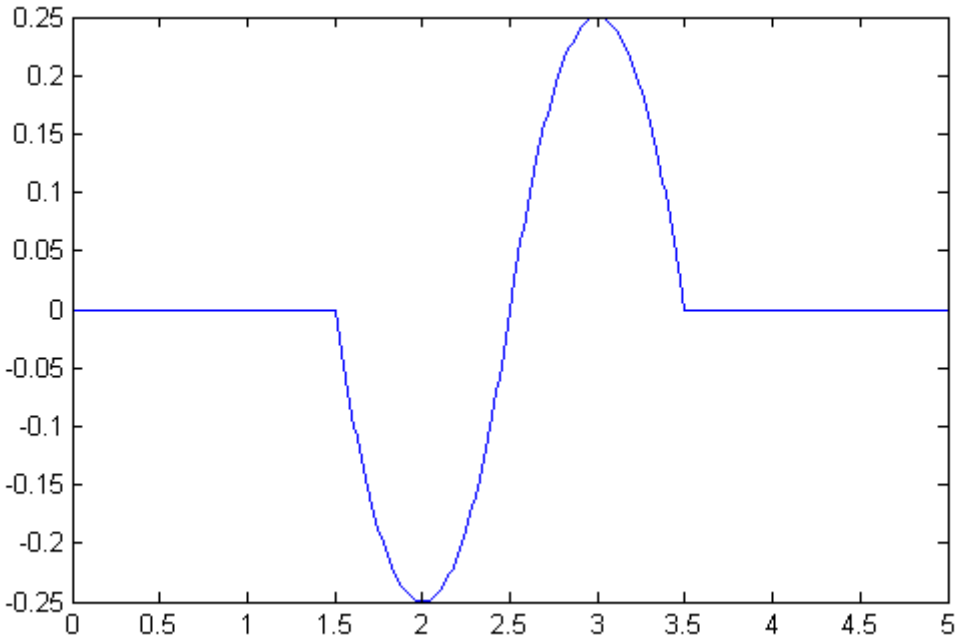

Figure 5. 1 Graph of the initial data in the equation (5.15)

For $3.0 < x < 3.5$ the characteristic speed decreases with x thus, top of the pulse moves faster than the below region of the pulse. This causes formation of shock. In the same way, a left-going shock also forms. At the end, the initial data becomes an N-wave. We will do our calculation on 40 uniform grids and we set $dt = 0.4dx$ so that $|cfl| \le 0.1$. We will give the results at time $t = 2.0$ which is just after the formation of shock and at time $t = 12.0$ which is the long-term evaluation. Note that because inviscid Burgers equation has no analytical solution, we use MC slope-limiter method with a 2000 grid points to draw the analytical solution for graphs. This is the way that most of the scientist use when drawing analytical solution.

Our numerical results are produced based on a general time loop that internally uses a specific high resolution method. Below we outline the basic structure of the algorithm. We note that we used fortran 95 to program the schemes that used in this study. Please refer to Appendix for the fortran code.

5.2.1 Algorithm of Methods for Burgers Equation

Step 1: Set *cfl*, Δx, Δt and *t_final* .

Step 2: Store x values between the initial and endpoint of given interval with step size Δx .

Step 3: For $t = 0$, store the cell average value for each grid cell from the initial data.

Step 4: Set the boundary condition.

Step 5: Do calculation below for each grid cell.

Calculate

$\phi(\theta)$ for right and left endpoints of grid cell according to the used method.

Flux for right and left endpoints of grid cell.

Fluctuation for right and left endpoints of grid cell.

Calculate the new cell value from the finite volume scheme

Step 6: Transfer the data to another variable and increase the time up to Δt.

Step 7: Repeat step 5 and step 6 until time becomes equal to t_final.

Step 8: Store the cell to plot the numerical results.

5.2.2 Results of Methods for $t = 2.0$

In this section we will give the results of all methods for time $t = 2.0$ and will discuss the results.

5.2.2.1 Graphs for Classical Methods

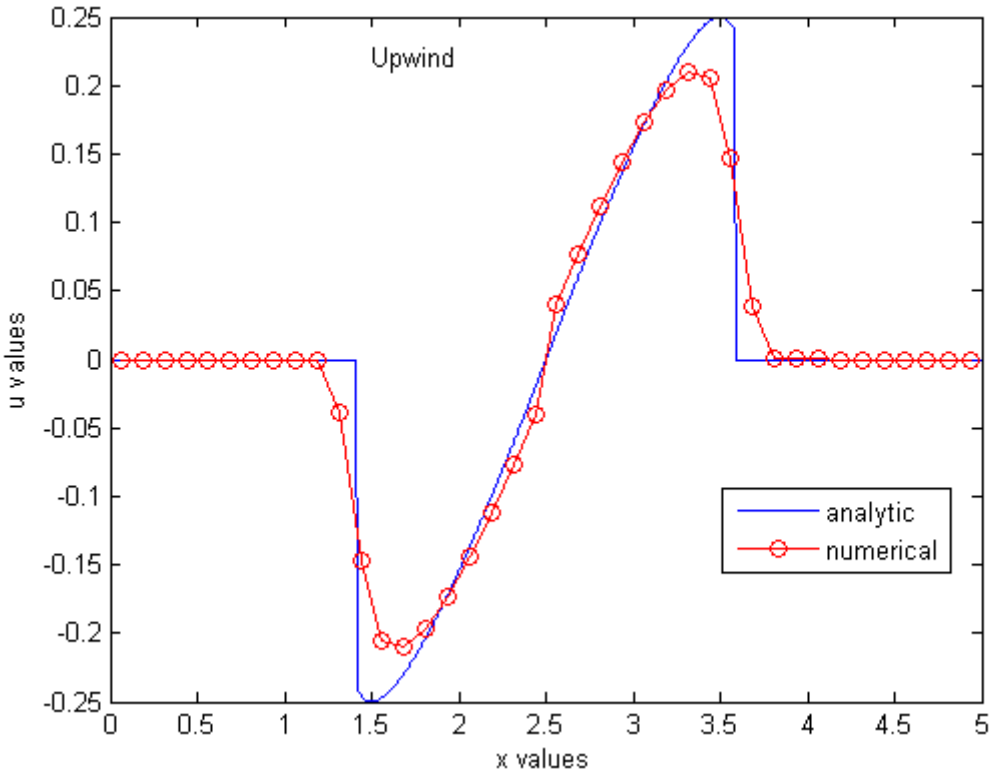

Figure 5. 2 Upwind method at time $t = 2.0$

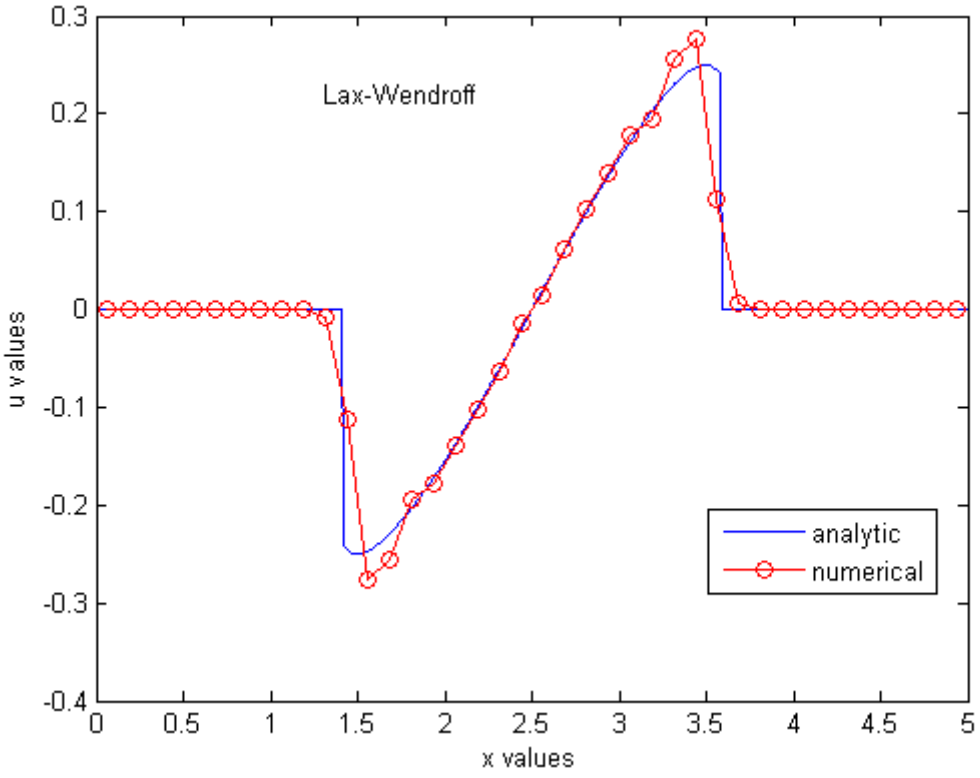

Figure 5. 3 Lax-Wendroff method at time $t = 2.0$

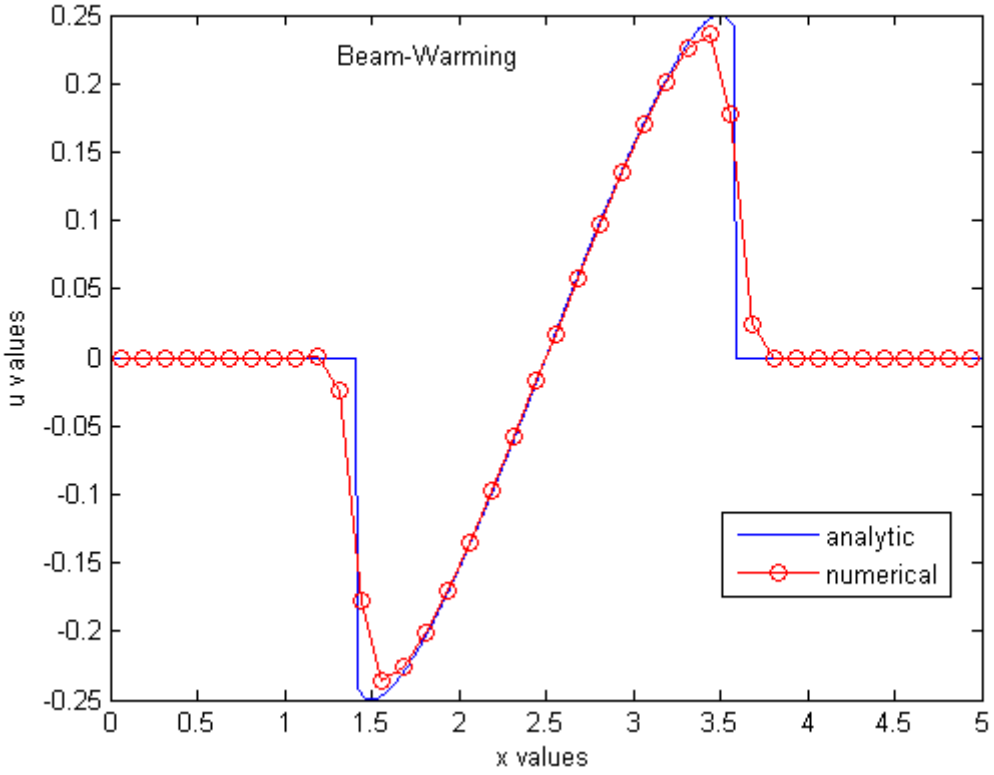

Figure 5. 4 Beam-Warming method at time $t = 2.0$

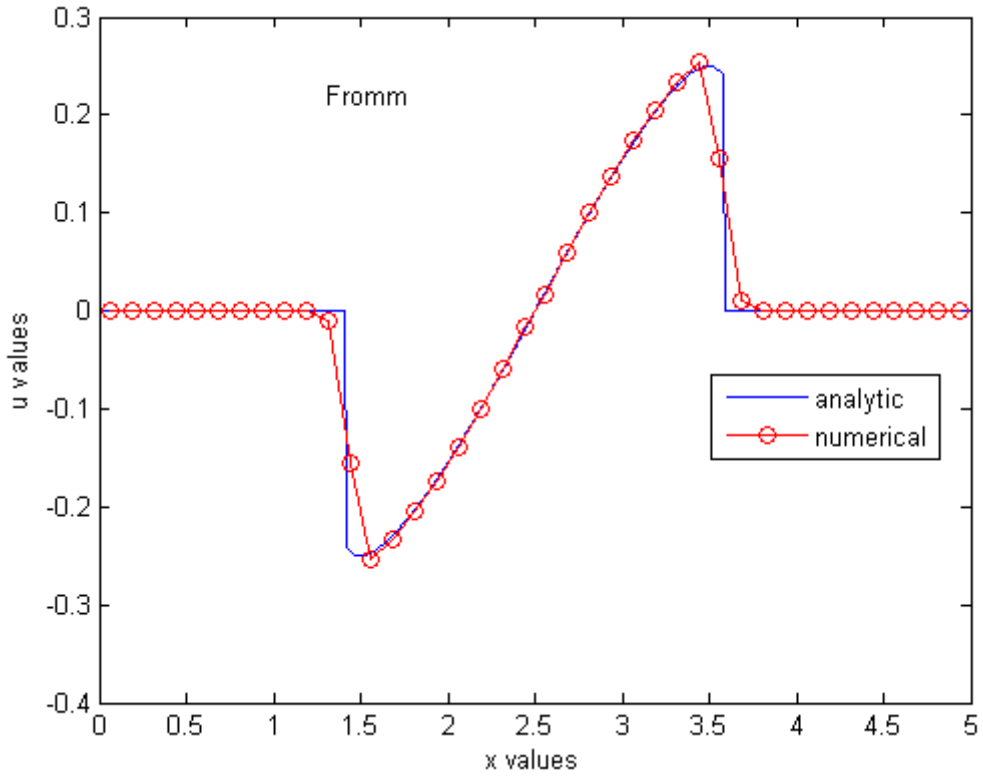

Figure 5. 5 Fromm's method at time $t = 2.0$

In figure 5.2, the upwind method gives dissipative result, particularly near the local minimum and local maximum points. This result is similar to the conclusion obtained by the advection equation. In figure 5.3, except for the extreme points, the Lax-Wendroff method gives good result; it captures the solution well compare to the upwind method. The Beam-Warming method gives the similar result with the Lax-Wendroff method for the smooth regions, but it is relatively well around the extreme points (figure 5.4). Compared to the Beam-Warming method, we can say that Fromm's method is more adequate in terms of capturing the amplitude of the wave. Overall, we can conclude that Fromm's method is the best in classical methods for $t = 2.0$.

5.2.2.2 Graphs for Advanced Methods

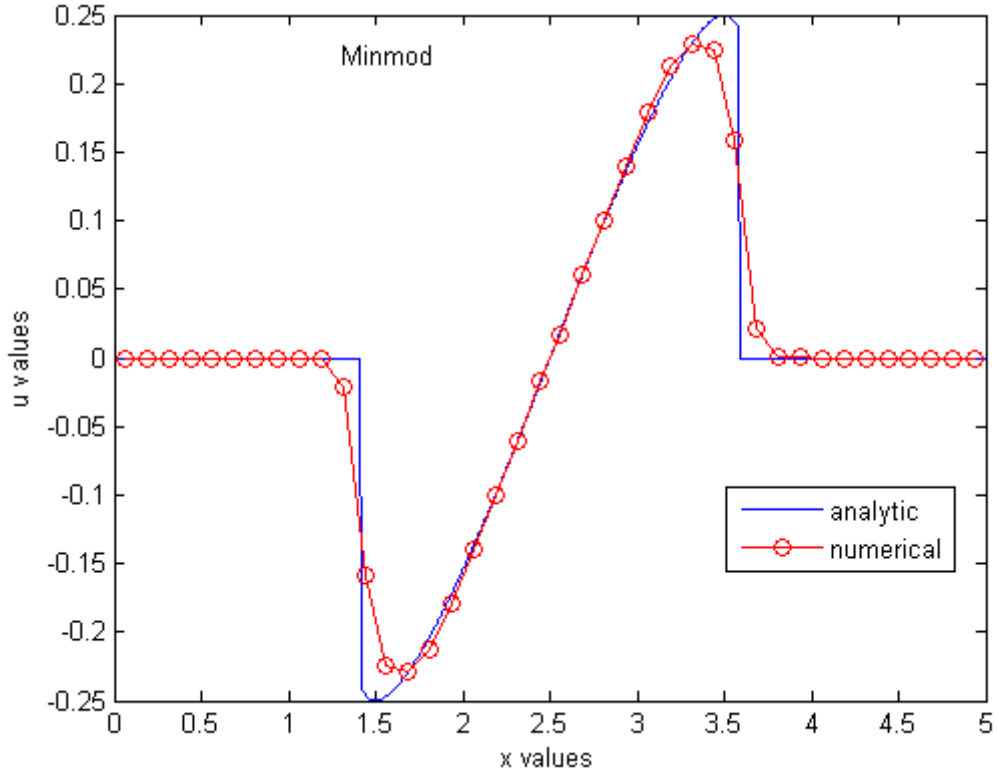

Figure 5. 6 Minmod method at time $t = 2.0$

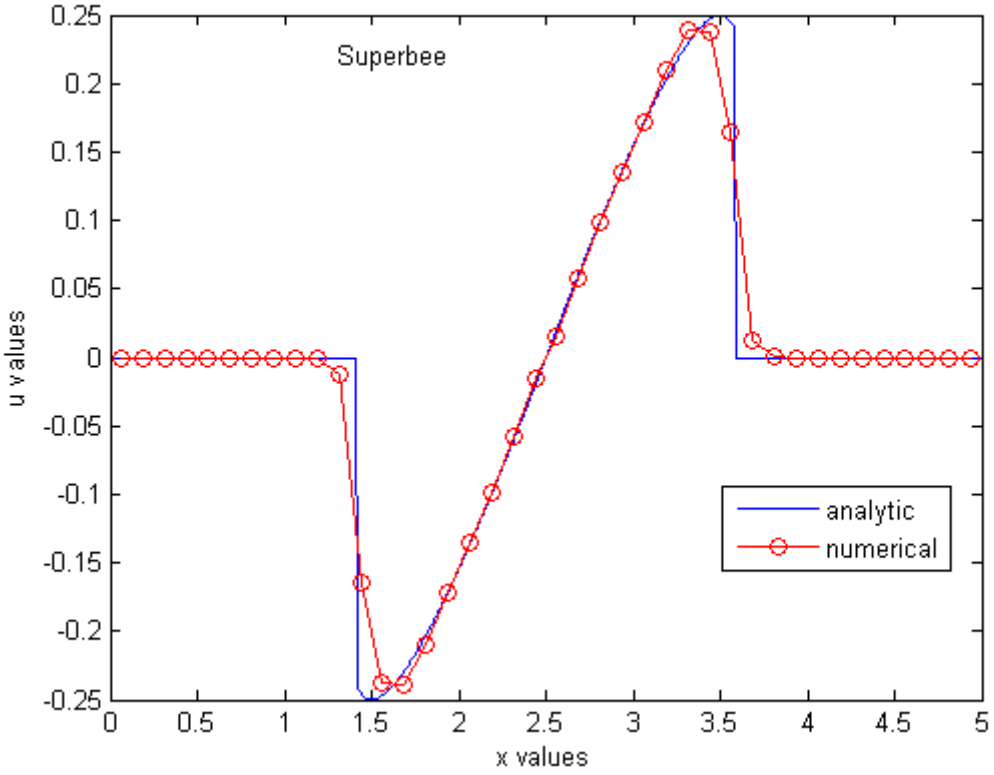

Figure 5. 7 Superbee method at time $t = 2.0$

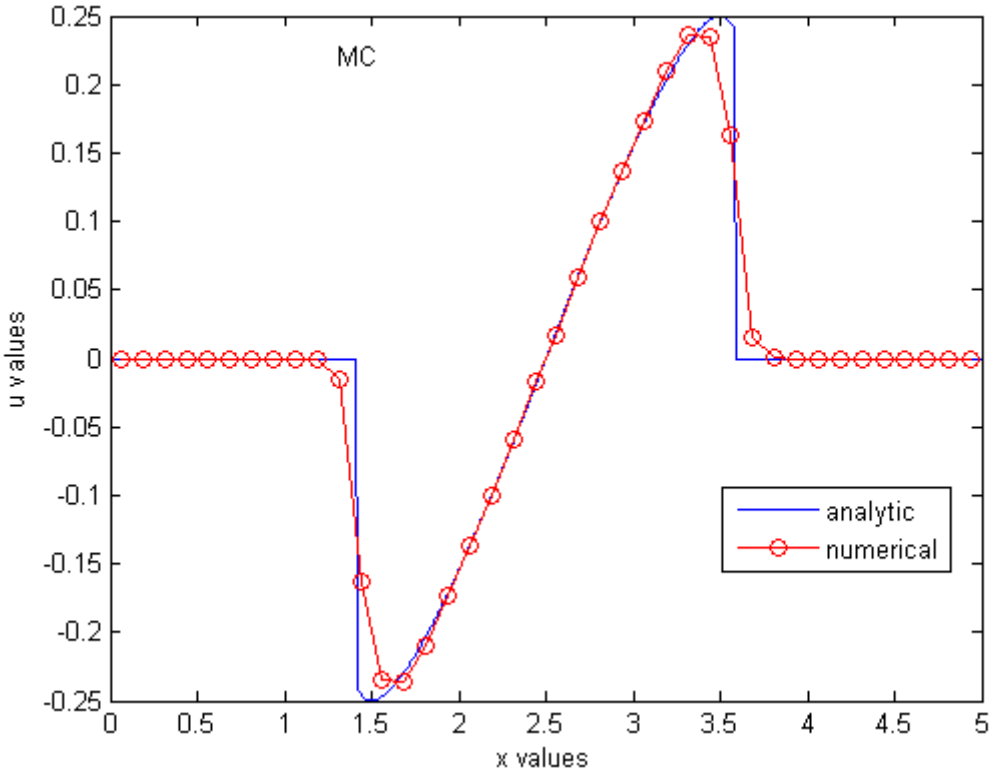

Figure 5. 8 MC method at time $t = 2.0$

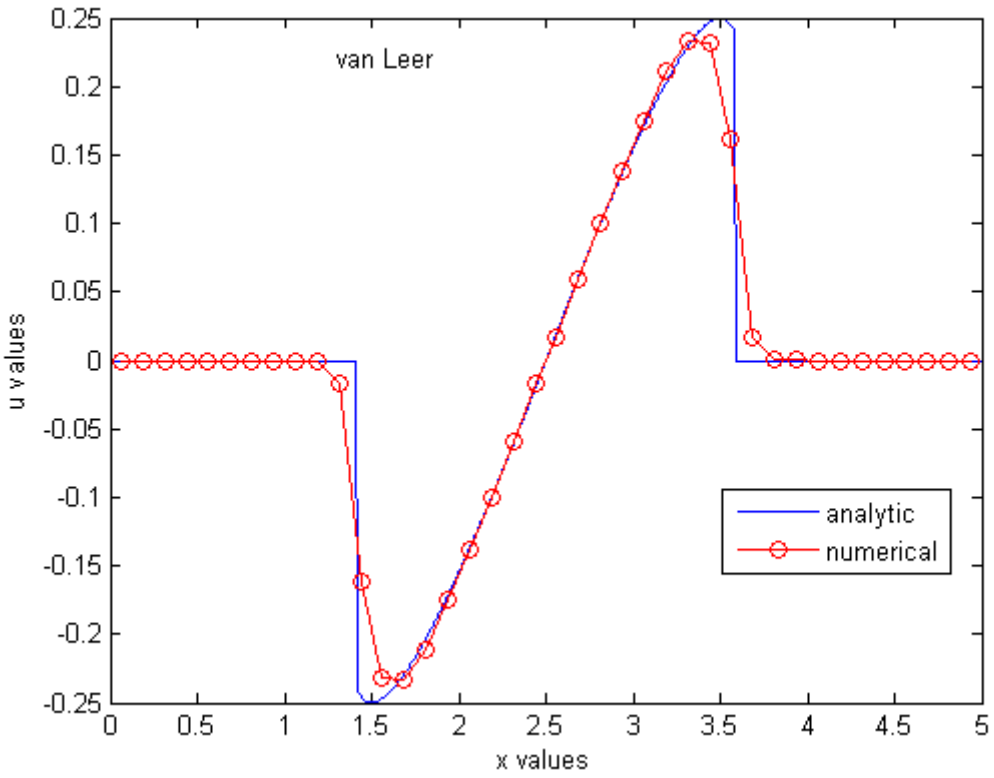

Figure 5. 9 Van Leer method at time $t = 2.0$

Result of minmod method in figure 5.6 is similar to the Beam-Warming method, but it is relatively dissipative especially near discontinuities. For $t = 2.0$, superbee method (figure 5.7) is better than both MC method (figure 5.8) and van Leer method (figure 5.9) in terms of the approximating the amplitude of the wave and capturing the steep gradient. In terms of amplitude, MC is better than van Leer.

5.2.3 Results of Methods at $t = 12.0$

Most of the time, results at $t = 12.0$ is more important than the results at $t = 2.0$ because they tell us the long behavior of the methods. As we will see in this section although some methods seem good for $t = 2.0$, its accuracy has deteriorated more than expected.

5.2.3.1 Graphs for Classical Methods

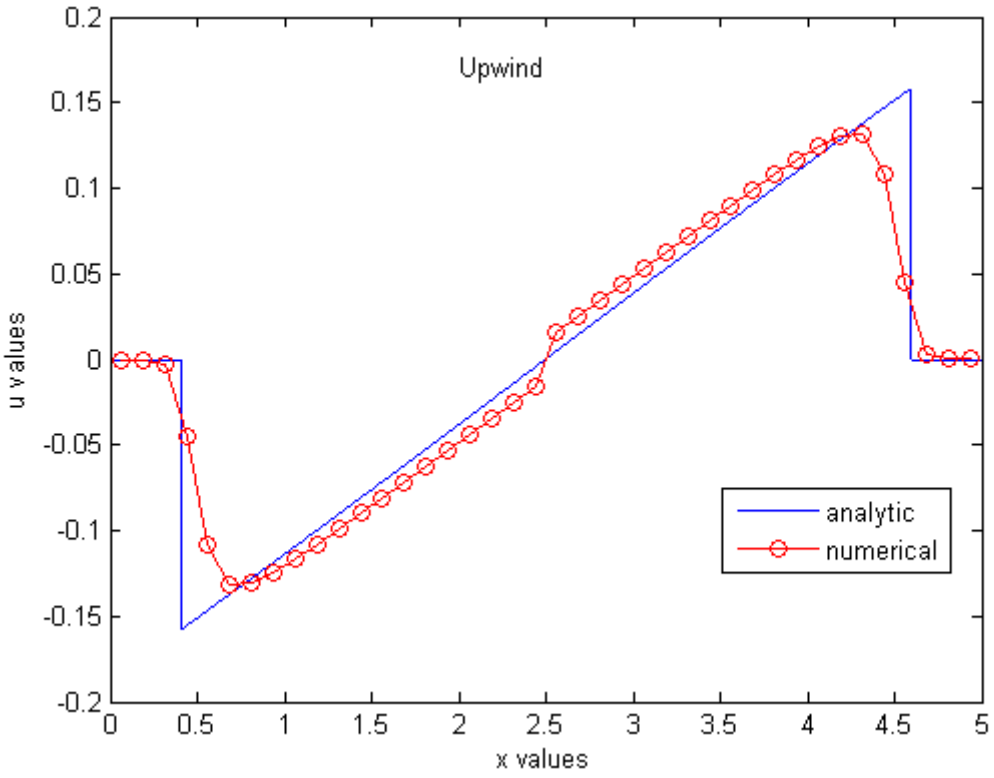

Figure 5. 10 Upwind method at time $t = 12.0$

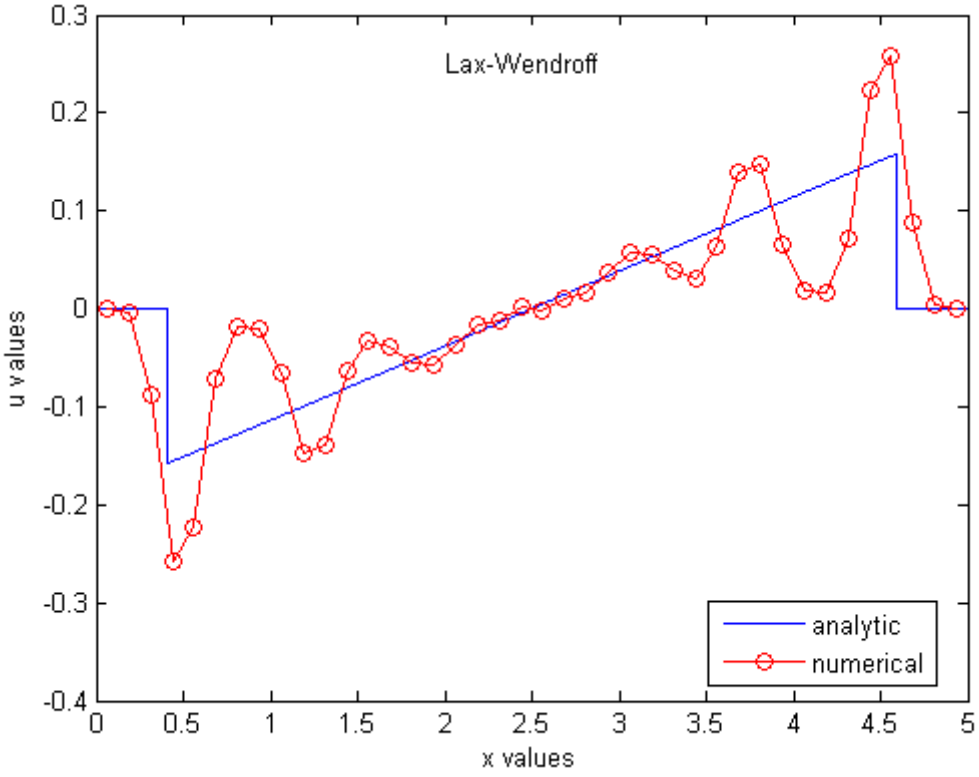

Figure 5. 11 Lax-Wendroff method at time $t = 12.0$

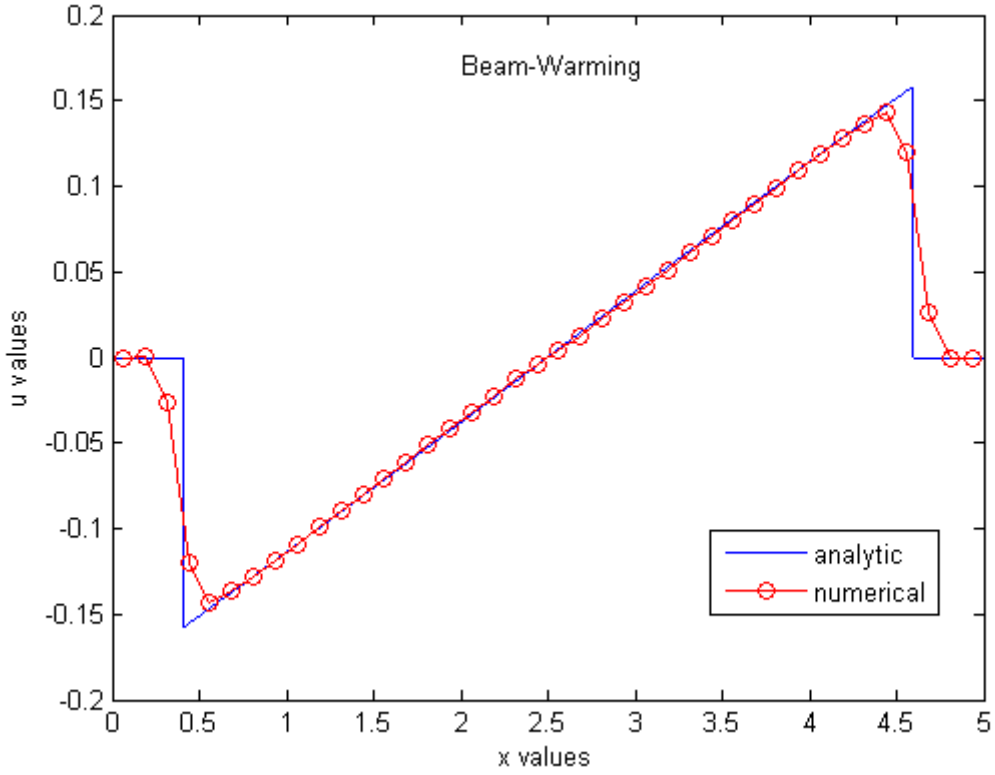

Figure 5. 12 Beam-Warming method at time $t = 12.0$

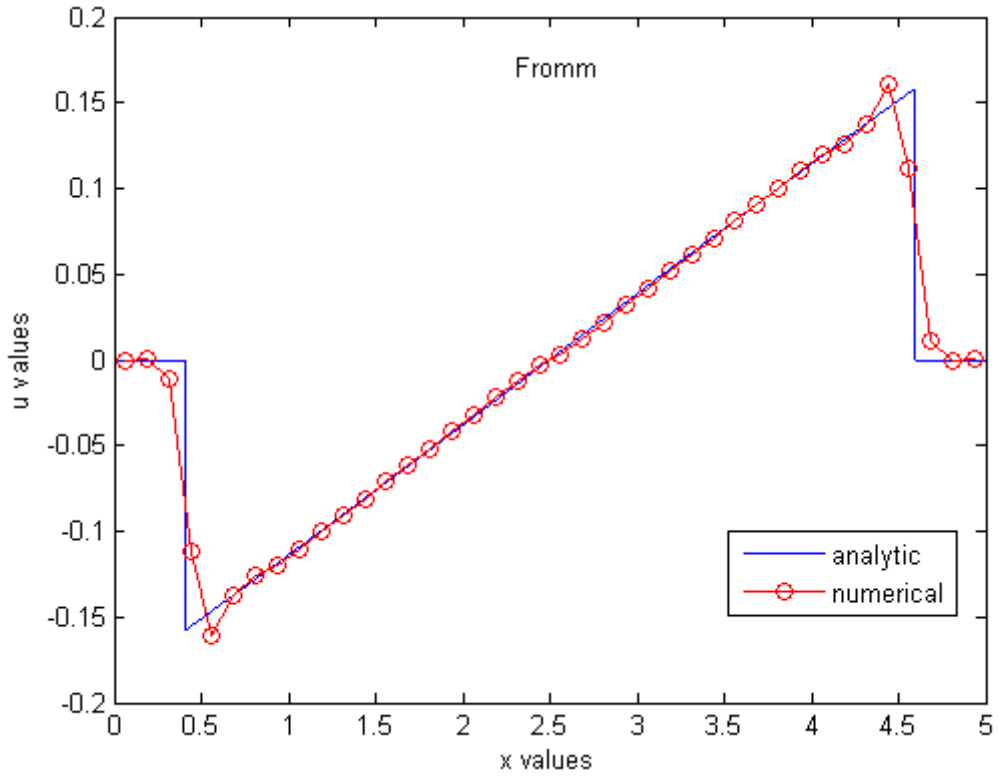

Figure 5. 13 Fromm method at time $t = 12.0$

When we analyze the graphs for $t = 12.0$, upwind method (figure 5.10) get worse to catching the solution even for smooth data. Lax-Wendroff method demonstrates its oscillatory characteristic much clearly (figure 5.11). Its result is unacceptable. In figure 5.12, surprisingly Beam-Warming method gives a nice solution though its oscillatory feature. It is also good to approximating the local minimum and local maximum. We see in figure 5.13 that, although Fromm's method is acceptable for smooth region, it tends to oscillate around discontinuity. This may cause problems in time. We can conclude that for long term evaluation, Beam-Warming method is the most satisfactory along the classical high resolution methods.

5.2.3.2 Graphs for Advanced Methods

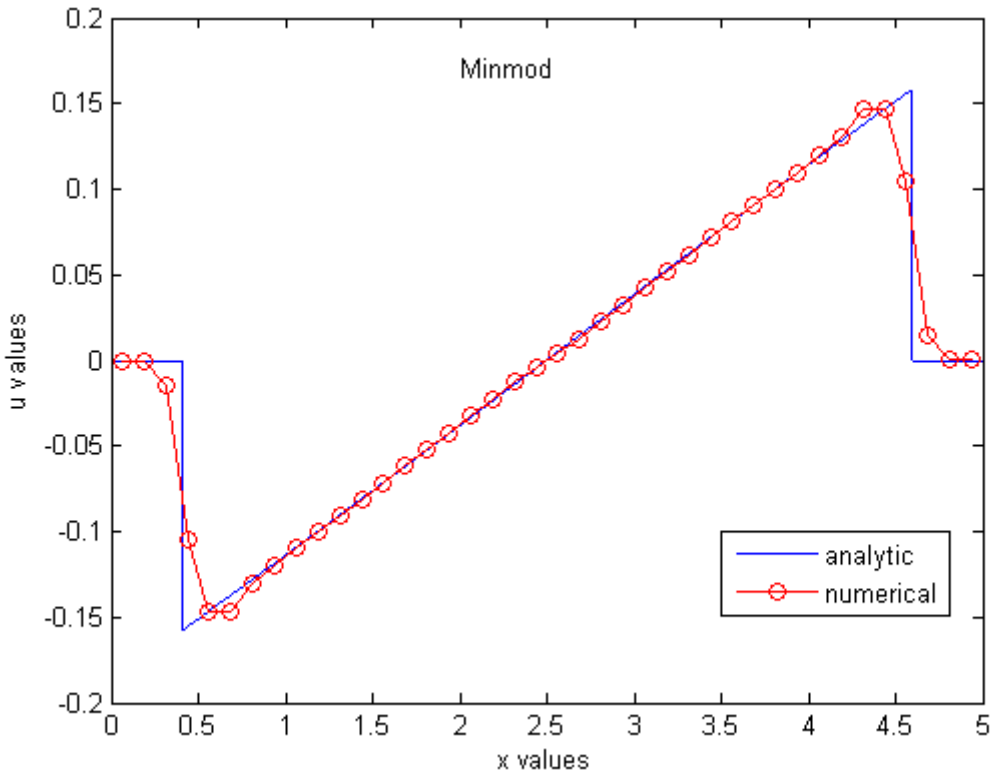

Figure 5. 14 Minmod method at time $t = 12.0$

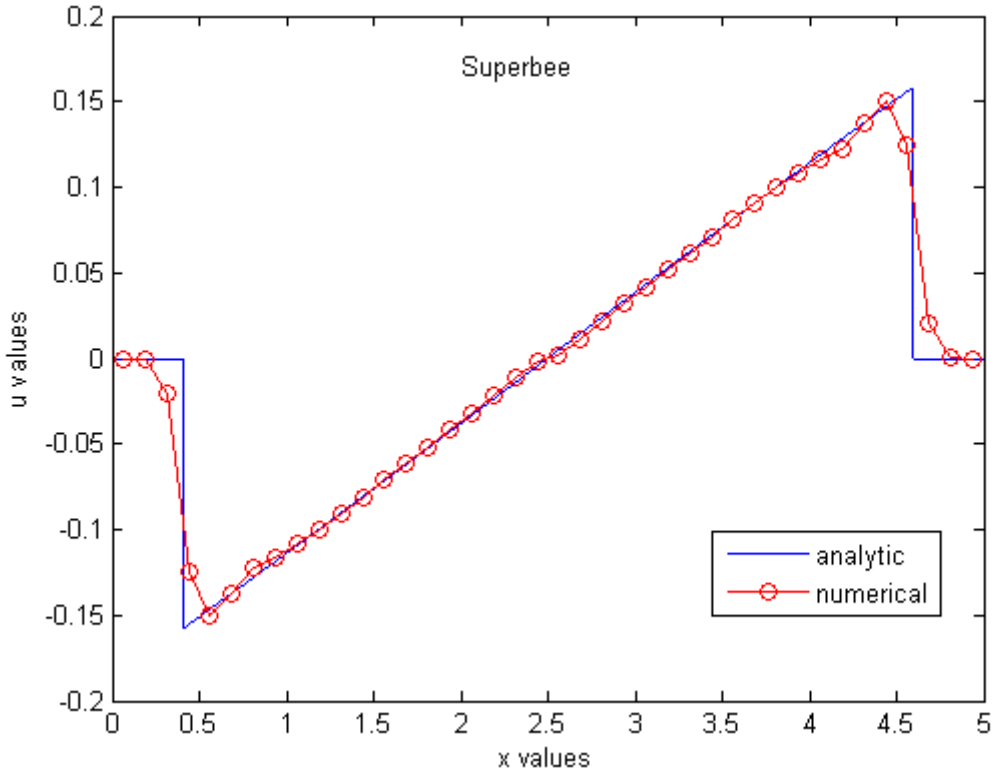

Figure 5. 15 Superbee method at time $t = 12.0$

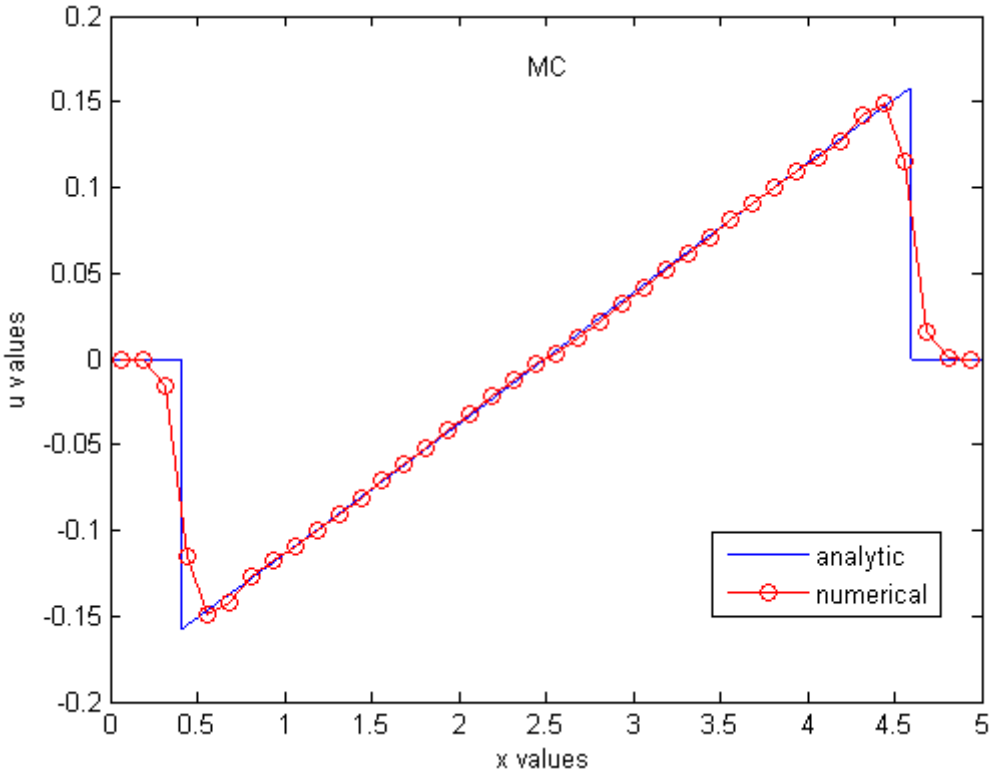

Figure 5. 16 MC method at time $t = 12.0$

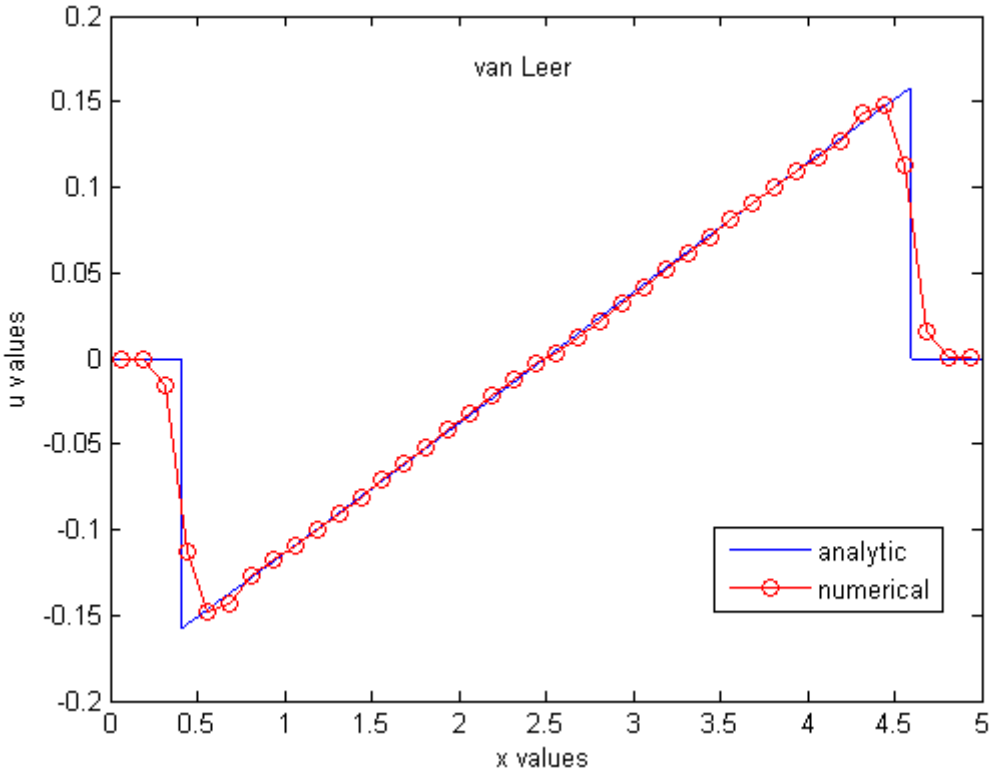

Figure 5. 17 Van Leer method at time $t = 12.0$

Even though its dissipative nature relative to advanced shock capturing method, minmod method in figure 5.14 is acceptable according to amplitude of the wave. Superbee (figure 5.15), MC (figure 5.16) and van Leer (figure 5.17) methods are quite similar to each other when we look at the results roughly. On the other hand, when we analyze deeply, we see that superbee method has a little bit deviation for smooth data. Furthermore, van Leer method is worse than the other two, in terms of accuracy near discontinuity. We can derive from these graphs that MC method is the best along the eight methods for long-term evaluation.

CHAPTER 6

RESULTS AND DISCUSSION

In this thesis, we have compared well-known high resolution methods in terms of their accuracy and stability for smooth and discontinuous problems. In this thesis, we have first given a discussion about some theoretical background. Then, we have provided more detailed discussions regarding the numerical methods.

We have discussed finite volume methods, conservation law, Riemann problems which are necessary tools to understand the high resolution methods clearly. Then, we have described the high resolution schemes with their important features and mathematical theory. We have applied above mentioned high resolution methods to a scalar, linear one-way wave equation. This has given us the opportunity to perform some theoretical analysis such as accuracy and stability. Then, we have applied the high resolution methods to the Burgers Equation. By solving this equation with different methods, we have gained a lot of insights about the stability accuracy and therefore the suitability of used high resolution methods for nonlinear hyperbolic partial differential equations.

In our conclusion, advanced high resolution methods have provided reasonable results for both smooth and discontinuous problems. Among the advanced high resolution schemes, MC slope-limiter method has been shown to be superior to the others. This thesis can be considered as an initial step towards understanding the fundamentals of high resolution methods. We wish have deeper understanding and explore further about these special group of numerical methods, with the aim of possible original contributions.

REFERENCES

[1] Leveque, R.J., (2002). Finite-Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge.

[2] Thomas J.W., (1999). Numerical Partial Differential Equations Conservation Laws and Elliptic Equations, First Edition, Springer-Verlag, New York.

[3] Godunov, S.K., (1959). "A Difference Method for Numerical Calculation of Discontinuous Solutions of the Equations of Hydrodynamics", Mat. Sb., 47(89): 271-306.

[4] Kadioglu, S.Y., (2011). "A Gas Dynamics Method Based on the Spectral Deferred Corrections (SDC) Time Integration Technique and the Piecewise Parabolic Method (PPM)", American Journal of Computational Mathematics, 1: 303-317.

[5] Lax, P. and Wendroff B., (1960). "Systems of Conservation Laws", Communications on Pure and Applied Mathematics, 13: 217-237; Editor: Sweby, P.K., (1984). "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws", SIAM Journal on Numerical Analysis, 21(5): 995-1011.

[6] Fromm, J.E., (1968). "A Method for Reducing Dispersion in Connective Difference Schemes", Journal of Computational Physics, 3(2): 176-189.

[7] Warming, B.E. and Beam, R.M., (1975). "Upwind Second-order Difference Schemes and Applications in Unsteady Aerodynamics Flows", AIAA Journal, 14(9): 1241-1249.

[8] Yang, H.Q. and Przekwas, A.J., (1990). "A Comparative Study of Advanced Shock-Capturing Schemes Applied to Burgers' Equation", AIAA 21st Fluid Dynamics, Plasma Dynamics and Lasers Conference, 18-20 June, Seattle WA.

[9] Leer, V.B., (1973). "Towards the Ultimate Conservative Difference Scheme I. The Quest of Monotonicity", Springer Lecture Notes Physics, 18: 163-168.

[10] Leer, V.B., (1974). "Towards the Ultimate Conservative Difference Scheme II. Monotonicity and conservation combined in a second order scheme", Journal of Computational Physics, 14: 361-370.

[11] Leer, V.B., (1977). "Towards the Ultimate Conservative Difference Scheme III. Upstream-Centered Finite Difference Schemes for Ideal Compressible Flow", Journal of Computational Physics, 23: 263-275.

[12] Leer, V.B., (1977). "Towards the Ultimate Conservative Difference Scheme IV. A New Approach to Numerical Convection", Journal of Computational Physics, 23: 276-299.

[13] Leer, V.B., (1979). “Towards the Ultimate Conservative Difference Scheme V. A second Order Sequel to Godunov’s Method”, Journal of Computational Physics, 32: 101-136.

[14] Roe, P.L., (1985). “Some Contributions to the Modeling of Discontinuous Flows”, Lecture Notes in Applied Mathematics, 22: 163-193.

[15] Sweby, P.K., (1984). “High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws”, SIAM Journal on Numerical Analysis, 21: 995-1011.

[16] Farthing, M.W. and Miller, C.T., (2001). “A Comparison of High-Resolution, Finite-Volume, Adaptive-Stencil Schemes for Simulating Advective-Dispersive Transport”, Advances in Water Resources, 24: 29-48.

[17] Harten, A., (1997). “High Resolution Schemes for Hyperbolic Conservation Laws”, Journal of Computational Physics, 135(2): 260-278.

[18] Maciel, E.S., (2012). “Explicit and Implicit TVD High Resolution Schemes in 2D”, Wseas Transactions on Applied and Theoretical Mechanics, 7(3): 182-209.

[19] Tenaud, C., Garnier, E. and Sagaut, P., (2000). “Evaluation of Some High-Order Shock Capturing Schemes for Direct Numerical Simulation of Unsteady Two-Dimensional Free Flows”, International Journal for Numerical Methods in Fluids, 33: 249-278.

[20] Daru, V. and Tenaud, C., (2001). “Evaluation of TVD High Resolution Schemes for Unsteady Viscous Shocked Flows”, Computers and Fluids, 30: 89-113.

[21] Thomas J.W., (1995). Numerical Partial Differential Equations: Finite Difference Methods, First Edition, Springer-Verlag, New York.

[22] Leveque, R.J., (1992). Numerical Methods for Conservation Laws, Second Edition, Birkhäuser, Basel.

[23] Debnath, L., Bhatta, D., (2007). Integral Transforms and Their Applications, Second Edition, Taylor & Francis Group, Boca Raton.

[24] Strikwerda, J.C., (2004). Finite Difference Schemes and Partial differential Equations, Second Edition, Society for Industrial and Applied Mathematics (SIAM), Philadelphia.

[25] Hussaini, M.Y., Leer, B.V. and Rosendale, J.V., (1997). Upwind and High-Resolution Schemes, First Edition, Springer-Verlag, Berlin.

[26] Courant, R., Isaacson, E. and Rees, M., (1952). “On the solution of non-linear hyperbolic differential equations by finite differences”, Communications on Pure and Applied Mathematics, 5: 243-255; Editor: Leer, B.V., (2005). “Upwind and High-Resolution Methods for Compressible Flow: From Donor Cell to Residual-Distribution Schemes”, Communications in Computational Physics, 1(2): 192-206.

[27] Landajuela, M., (2011). “Burgers Equation”, Basque Center for Applied Mathematics, Summer 2011, Spain.

APPENDIX

FORTRAN CODE FOR BURGERS EQUATION

```
program HRMforBurgersEquation
implicit none
integer::i,mt
integer, parameter :: M = 40, ng=2
double precision   xedge(0 : M) , xcell(0 : M-1)
double precision Uext(0-ng : M-1+ng)
double precision Uold(0-ng : M-1+ng)
double precision Unew(0-ng : M-1+ng)
double precision:: t_final, time, dt, xa, xb, dx, cfl,a
double precision A_plus, A_minus
double precision w_imh, w_iph
double precision s_minus_imh,s_minus_iph,s_plus_imh,s_plus_iph
double precision s_imh,s_iph
double precision wtilda_imh, wtilda_iph
double precision f_imh, f_iph
double precision minmod
double precision teta_imh, teta_iph
double precision fiteta_imh, fiteta_iph
double precision, parameter:: ep=0.001d0
character*2:: label = 'AA'

!set mt value for methods that you want to use
! 1 for upwind; 2 for l-w; 3 for b-w; 4 for fromm; 5 for minmod
! 6 for superbee; 7 for MC; 8 for van leer

mt=1

cfl = 0.4d0
t_final = 12.0d0
time = 0.0d0
xa = 0.0d0 ; xb = 5.0d0
dx = (xb - xa)/M
dt =  cfl * dx

do i = 0, M-1
   xedge(i) = xa + i*dx
   xcell(i) = xa + (i+1./2.)*dx
enddo

   xedge(M) = xa + M*dx
```

```
do i= 0, M-1
  uext(i) = dmax1(0.0d0, (xcell(i)-2.5d0) * (3.5d0-xcell(i)) )
enddo

do i= 0, int(M/2)-1
   uext(i)=-uext(M-1-i)
enddo

do i=0, M-1
  uold(i)=uext(i)
enddo

Uold(-2) = Uold(M-2)
Uold(-1) = Uold(M-1)
Uold(M)  = Uold(0)
Uold(M+1)  = Uold(1)

  do while (time <= t_final)
    time = time + dt
    do i = 0 , M-1

      s_imh = 0.5d0*(Uold(i)   + Uold(i-1))
      s_iph = 0.5d0*(Uold(i+1) + Uold(i  ))

      s_minus_imh = dmin1(0.d0, s_imh)
      s_minus_iph = dmin1(0.d0, s_iph)
      s_plus_imh  = dmax1(0.d0, s_imh)
      s_plus_iph  = dmin1(0.d0, s_iph)

      w_imh = Uold(i)-Uold(i-1)
      w_iph = Uold(i+1)-Uold(i)

      if( (Uold(i-1).LT.0.d0) .AND. (Uold(i).GT.0.d0) ) then
        A_plus = 0.5d0 * ( Uold(i)**2 )
      else
        A_plus  = s_plus_imh *w_imh
      endif

      if( (Uold(i).LT.0.d0) .AND. (Uold(i+1).GT.0.d0) ) then
        A_minus = -0.5d0 * ( Uold(i)**2 )
      else
        A_minus = s_minus_iph*w_iph
      endif



      methodtype: select case (mt)
      case (1)   !upwind
        label = 'Up'
        teta_imh = 0.0d0
        teta_iph = 0.0d0
        fiteta_imh = 0.0d0
        fiteta_iph = 0.0d0
```

```
        wtilda_imh = fiteta_imh*w_imh
        wtilda_iph = fiteta_iph*w_iph

      case (2)    !LW
        label = 'LW'
        teta_imh = 1.0d0
        teta_iph = 1.0d0
        fiteta_imh = 1.0d0
        fiteta_iph = 1.0d0

        wtilda_imh = fiteta_imh*w_imh
        wtilda_iph = fiteta_iph*w_iph

      case (3)    !BW
        label = 'BW'
        call BWfi (teta_imh, teta_iph, w_imh,w_iph, s_imh,s_iph,
     &            Uold(i-2),Uold(i-1), Uold(i),Uold(i+1),Uold(i+2))

        fiteta_imh = teta_imh
        fiteta_iph = teta_iph

        if (w_imh == 0.0d0) then
          wtilda_imh = fiteta_imh*ep
        else
          wtilda_imh = fiteta_imh*w_imh
        endif

        if (w_iph == 0.0d0) then
          wtilda_iph = fiteta_iph*ep
        else
          wtilda_iph = fiteta_iph*w_iph
        endif

      case (4)   ! Fromm
        label = 'FR'
        call BWfi (teta_imh, teta_iph, w_imh,w_iph, s_imh,s_iph,
     &            Uold(i-2),Uold(i-1), Uold(i),Uold(i+1),Uold(i+2))

        fiteta_imh = 0.5d0*( 1.0d0 + teta_imh )
        fiteta_iph = 0.5d0*( 1.0d0 + teta_iph )

        wtilda_imh = fiteta_imh*w_imh
        wtilda_iph = fiteta_iph*w_iph


      case (5)    !minmod
        label = 'MM'
        call BWfi (teta_imh, teta_iph, w_imh,w_iph, s_imh,s_iph,
     &            Uold(i-2),Uold(i-1), Uold(i),Uold(i+1),Uold(i+2))

        fiteta_imh = minmod(1.0d0 ,teta_imh )
        fiteta_iph = minmod(1.0d0 ,teta_iph )

        wtilda_imh = fiteta_imh*w_imh
        wtilda_iph = fiteta_iph*w_iph
```

```
      case (6)   !superbee
        label = 'SB'
        call BWfi (teta_imh, teta_iph, w_imh,w_iph, s_imh,s_iph,
&              Uold(i-2),Uold(i-1), Uold(i),Uold(i+1),Uold(i+2))

        fiteta_imh = dmax1(0.0d0 ,dmin1(1.0d0,2*teta_imh),
&                                  dmin1(2.0d0,teta_imh ) )
        fiteta_iph = dmax1(0.0d0 ,dmin1(1.0d0,2*teta_iph),
&                                  dmin1(2.0d0,teta_iph) )

        wtilda_imh = fiteta_imh*w_imh
        wtilda_iph = fiteta_iph*w_iph

      case (7)   !MC
        label = 'MC'
        call BWfi (teta_imh, teta_iph, w_imh,w_iph, s_imh,s_iph,
&              Uold(i-2),Uold(i-1), Uold(i),Uold(i+1),Uold(i+2))

        fiteta_imh = dmax1(0.0d0 ,dmin1((1.0d0+teta_imh)/2.0d0 ,
&                                  2.0d0, 2*teta_imh ) )
        fiteta_iph = dmax1(0.0d0 ,dmin1((1.0d0+teta_iph)/2.0d0 ,
&                                   2.0d0, 2*teta_iph) )

        wtilda_imh = fiteta_imh*w_imh
        wtilda_iph = fiteta_iph*w_iph

      case (8)  !van Leer
        label = 'VL'
        call BWfi (teta_imh, teta_iph, w_imh,w_iph, s_imh,s_iph,
&              Uold(i-2),Uold(i-1), Uold(i),Uold(i+1),Uold(i+2))

        fiteta_imh= (teta_imh+dabs(teta_imh))/(1 + dabs(teta_imh))
        fiteta_iph= (teta_iph+dabs(teta_iph))/(1 + dabs(teta_iph))

        wtilda_imh = fiteta_imh*w_imh
        wtilda_iph = fiteta_iph*w_iph



      case default
        print*, "wrong case"

      end select methodtype


      f_imh = 0.5d0*dabs(s_imh)*(1.d0 - (dt/dx)*dabs(s_imh))
&                                    *wtilda_imh
      f_iph = 0.5d0*dabs(s_iph)*(1.d0 - (dt/dx)*dabs(s_iph))
&                                    *wtilda_iph

      Unew(i) = Uold(i) - (dt/dx) * (A_plus + A_minus)
&                                  -(dt/dx)*(f_iph - f_imh)
```

```
    enddo

    Unew(-1) = Unew(M-1)
    Unew(M)  = Unew(0)
    Unew(-2) = Unew(M-2)
    Unew(M+1)  = Unew(1)

    Uold = Unew
  enddo

 do i=0, M-1
    open(13,file = 't12'//label//'.txt')
    write(13,*) xcell(i), Unew(i) !,dabs(Uext(i)- Unew(i))

 enddo
 close(13)
 END



 subroutine BWfi (teta_imh, teta_iph, w_imh,w_iph,s_imh,s_iph
&                                  ,Um2,Um1,Uo,Up1,Up2)

  double precision teta_imh,teta_iph,s_imh,s_iph
  double precision w_imh,w_iph ,Um2,Um1,Uo,Up1,Up2
  double precision, parameter:: ep=0.001d0

  if(w_imh == 0.0d0) then
    if ( s_imh .GT. 0.0d0) then
      teta_imh = (Um1-Um2)/ep
    else
      teta_imh = (Up1-Uo)/ep
    endif

  else
    if ( s_imh .GT. 0.0d0) then
      teta_imh = (Um1-Um2)/w_imh
    else
      teta_imh = (Up1-Uo)/w_imh
    endif
  endif

  if(w_iph == 0.0d0) then
    if ( s_iph .GT. 0.0d0) then
      teta_iph = (Uo-Um1)/ep
    else
      teta_iph = (Up2-Up1)/ep
    endif

  else
    if ( s_iph .GT. 0.0d0) then
      teta_iph = (Uo-Um1)/w_iph
    else
      teta_iph = (Up2-Up1)/w_iph
    endif
```

```
  endif

end subroutine


double precision FUNCTION minmod(a,b)
  double precision,intent(in)::a,b
  double precision c

  if (a*b>0) then
     if (dabs(a)<dabs(b)) then
        c=a
     else
        c=b
     endif
  else
     c=0
  endif
  minmod=c
end
```

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Veli ÇOLAK

Date of birth and place : 01.01.1990

Foreign Languages : English

E-mail : velicolak2010@hotmail.com

EDUCATION

Degree	Department	University	Date of Graduation
Undergraduate	Mathematics	Boğaziçi University	2011
High School		Kılıçaslan High School	2006

WORK EXPERIENCE

Year	Corporation/Institute	Enrollment
2012	Yıldız Technical University	Research Assistant