5-2016

Definition and Construction of Entropy Satisfying Multiresolution Analysis (MRA)

Ju Y. Yi
Utah State University

DEFINITION AND CONSTRUCTION OF ENTROPY SATISFYING

MULTIRESOLUTION ANALYSIS (MRA)

by

Ju Y. Yi

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mathematical Sciences

Approved:

Joseph Koebbe Major Professor	Luis Gordillo Committee Member
Nghiem Nguyen Committee Member	Zhaohu Nie Committee Member
Todd Moon Committee Member	Mark McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2016

ABSTRACT

DEFINITION AND CONSTRUCTION OF ENTROPY SATISFYING

MULTIRESOLUTION ANALYSIS(MRA)

by

Ju Y. Yi, Doctorate of Philosophy

Utah State University, 2016

Major Professor: Dr. Joseph Koebbe
Department: Mathematics and Statistics

This paper considers some numerical schemes for the approximate solution of conservation laws and various wavelet methods are reviewed. This is followed by the construction of wavelet spaces based on a polynomial framework for the approximate solution of conservation laws. Construction of a representation of the approximate solution in terms of an entropy satisfying Multiresolution Analysis (MRA) is defined. Finally, a proof of convergence of the approximate solution of conservation laws using the characterization provided by the basis functions in the MRA will be given.

(110 pages)

PUBLIC ABSTRACT

DEFINITION AND CONSTRUCTION OF ENTROPY SATISFYING MULTIRESOLUTION ANALYSIS(MRA)

by

Ju Y. Yi, Doctorate of Philosophy

Utah State University, 2016

Major Professor: Dr. Joseph Koebbe
Department: Mathematics and Statistics

This paper considers some numerical schemes for the approximate solution of conservation laws and various wavelet methods are reviewed. This is followed by the construction of wavelet spaces based on a polynomial framework for the approximate solution of conservation laws. Construction of a representation of the approximate solution in terms of an entropy satisfying Multiresolution Analysis (MRA) is defined. Finally, a proof of convergence of the approximate solution will be given.

(110 pages)

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my major professor, Dr. Joseph Koebbe, for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. Besides my advisor, I would like to thank the rest of my thesis committee members, Drs. Luis Gordillo, Nighiem Nguyen, Zhaohu Nie, and Todd Moon, for their support and assistance while I pursued studies and research at Utah State University.

I would like to thank my family: my parents Gwi Yi and Tae Yi, my sisters Hyun Yi, Joice Yi, and brother Harris Yi, for their love and support from the beginning of my research to this final document. I could not have done it without all of you.

Ju Y. Yi

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The equation,

$$u_t + f(u)_x = 0 \quad -\infty < x < \infty, \;\; t > 0 \tag{1.1}$$

is called a conservation law. Conservation laws are partial differential equations that describe the time evolution of some quantity conserved in time. Conservation laws have been a concern of mathematicians, scientists, and engineers for a long time because it is not possible to write down the exact solutions of equation (1.1) for all $t > 0$.

In this paper, we will build wavelets conditioned on differential operators of the form (1.1) and construct an approximate solution for the conservation law using wavelets based on a polynomial framework. Then we will prove the convergence of our approximate solution to a unique weak solution of the conservation law. Our approximate solution is of the form

$$u(x, t_n) = u^n(x) = \alpha_0^n \varphi_0(x) + \sum_{l=0}^{m} \sum_{k=0}^{2^m-1} \beta_{l,k}^n \psi_{l,k}(x) \tag{1.2}$$

$$\alpha_0^n = \langle u^n(x), \varphi_0(x) \rangle$$
$$\beta_{l,k}^n = \langle u^n(x), \psi_{l,k}(x) \rangle$$

where $\varphi_0(x)$ is a shape function and $\psi_{l,k}(x)$ is an associated wavelet function.

This work treats conservation laws of the form (1.1), the development of an entropy condition that can be used on the conservation law and our representation of the approximate solution (1.2). From the construction, a definition is given for an entropy satisfying multiresolution analysis or MRA. In Smoller [15] existence and uniqueness of a weak solution for (1.1) using the Lax-Friedrichs method was presented. In contrast, we will develop an example of an approximate solution using the finite volume method (FVM) and Simpson's rule. Our goal in this paper is to prove that our approximate solution converges if it satisfies the entropy condition. The rest of the paper is organized as follows

In Chapter 2, we will start with a brief literature review of numerical methods for conservation laws, which provides useful concepts for weak solutions of conservation laws. Also we will test some numerical methods (classical methods and modern methods) for the approximate solution of a scalar conservation law in linear and nonlinear cases.

In Chapter 3, we will review a polynomial framework for construction of approximate solutions scalar conservation laws based on [4,5]. We will define grid-line values

$$U^n_{j+\frac{1}{2}} \approx u(x_{j+\frac{1}{2}}, t_n) = u(x_j + \frac{\Delta x}{2}, t_n),$$

used in the determination of numerical fluxes and explain some examples of the use of the polynomial definition of $U^n_{j\pm\frac{1}{2}}$ in linear and nonlinear cases for conservation laws.

In Chapter 4, we will present a brief review of the history of wavelets and describe the lifting scheme. Also we will review and define the multiresolution analysis (MRA). Then we will build shape functions based on a polynomial framework lifting approaching for specific numerical methods; Upwind, Lax-Wendroff, Beam-Warming, TVD and Lax-Friedrichs schemes.

In Chapter 5, we will construct wavelet functions from numerical methods for conservation laws. We will use Upwind, Lax-Wendroff, Beam-Warming, TVD and Lax-Friedrichs schemes as examples.

In Chapter 6, we will construct an approximate solution representation of the form in (1.2) for conservation laws. As an example of a representative solution, (1.2), we will compute $U(x, \Delta t)$ using a finite volume method [10] and also compute the numerical flux values $F^n_{j\pm\frac{1}{2}}$ via Simpson's rule for linear and nonlinear conservation laws.

In Chapter 7, we will show that our approximate method converges to a unique weak solution. This is the main goal of the work in this paper. Note that the analysis follows the work of Oleinik,[13], as presented in Smoller, [15].

Chapter 8 contains conclusions and a discussion of future work.

CHAPTER 2
REVIEW OF NUMERICAL METHODS FOR THE APPROXIMATE SOLUTION OF SCALAR HYPERBOLIC CONSERVATION LAWS

In this chapter, we will present definitions and examples of conservation laws. For completeness definitions related to weak solutions and entropy conditions are presented. Next, we will review and test various numerical methods for approximate solution of conservation laws.

2.1 Scalar Conservation Laws: Definition and Examples

A conservation law is a time dependent system of the form

$$u_t + f(u)_x = 0 \quad -\infty < x < \infty, \;\; t > 0 \tag{2.1}$$

with initial data

$$u(x, 0) = u_0(x) \qquad -\infty < x < \infty \tag{2.2}$$

where the subscripts indicate differentiation with respect to the time variable, t, and the spatial variable, x. The function $f(u)$ is called a flux function and it is assumed $u_0(x)$ is in $L^\infty(-\infty, \infty)$. (2.1) and (2.2) are an example of a hyperbolic partial differential equation. Hyperbolic equations appear commonly in the physical world. The propagation of acoustic waves and electromagnetic waves obey hyperbolic equations. Hyperbolic equations with compactly-supported initial data yield compactly-supported solutions for all time. For the work in this paper we consider two categories of hyperbolic equations in one-spatial dimension, linear and nonlinear conservation laws, as defined below.

2.1.1 Linear Conservation Laws: $f(u) = au, a \neq 0$

We first consider the linear scalar advection equation:

$$\begin{aligned} u_t + au_x &= 0, \\ u(x,0) &= u_0(x) \end{aligned} \tag{2.3}$$

It is assumed that $a \neq 0$ is a constant. The analytic solution of (2.3) is $u(x,t) = u_0(x - at)$. This solution can be obtained by an application of the method of characteristics [11].

2.1.2 An Example of a Nonlinear Conservation Law: Burgers' Equation

In this case, we consider a flux function which is nonlinear in the unknown function u. For example, if $f(u) = \frac{1}{2}u^2$, then

$$u_t + uu_x = 0 \tag{2.4}$$

This is a famous example of a nonlinear conservation law. The solution of (2.4) is $u(x,t) = u(\xi, 0)$ where $x = \xi + u(\xi, 0)t$. This solution is not necessarily unique [11]. Equation (2.4) can be written in conservation form as follows,

$$u_t + \left(\frac{u^2}{2}\right)_x = 0. \tag{2.5}$$

We call (2.5) the inviscid Burgers' equation [2].

2.1.3 Weak Solutions and Entropy Conditions

Physical models usually involve some dissipation that may be neglected in an idealized approximation. For example,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \epsilon\frac{\partial^2 u}{\partial x^2}$$

includes a dissipation term on the right hand side. In an idealized model where we assume $0 < \epsilon << 1$, it may be advantageous to neglect the dissipation term and the result is the model defined in (2.3).

Definition 2.1. Weak solution [11]

A weak solution for (2.1) can be obtained by multiplying $u_t + f(u)_x = 0$ by $\phi(x,t) \in C_0^1(\mathbb{R} \times \mathbb{R}^+)$. Then, integrating in x and t, to obtain

$$\int_0^{\infty} \int_{-\infty}^{\infty} \phi[u_t + f(u)_x] \, dx \, dt = 0$$

Using integration by parts, we obtain

$$\int_0^{\infty} \int_{-\infty}^{\infty} [\phi_t u + \phi_x f(u)] \, dx \, dt + \int_{-\infty}^{\infty} \phi(x,0) u(x,0) \, dx = 0. \tag{2.6}$$

Stated another way, we want to find $u(x,t)$ such that (2.6) is satisfied for all $\phi(x,t) \in C_0^1(\mathbb{R} \times \mathbb{R}^+)$. If such a solution exists, $u(x,t)$ is called a ***weak solution*** of (2.1).

Weak solutions of conservation laws are not unique in general. We can use a vanishing viscosity solution or an entropy condition to obtain uniqueness in some cases. It is well-known that if f is strictly convex, then there exists a unique weak solution $u(x,t)$ of (2.1) satisfying a physical entropy condition [2]. The vanishing viscosity solution involves taking the limit as ϵ goes to zero in the equation

$$u_t + a u_x = \epsilon u_{xx}$$

with $0 < \epsilon << 1$ as discussed above.

Let us introduce some examples of entropy conditions.(see [11]):
Consider

$$u_t + f(u)_x = 0$$

on $t > 0$, $-\infty < x < \infty$ with piecewise constant initial data

$$u(x,0) = \begin{cases} u_L & : x < 0 \\ u_R & : x > 0 \end{cases}$$

(i) (Entropy condition 1 and Scalar case):

For $u_t + f(u)_x = 0$, a discontinuity propagating with shock speed,

$$s = \frac{f(u_L) - f(u_R)}{u_L - u_R} = \frac{[f]}{[u]}$$

satisfies an entropy condition if

$$f'(u_L) > s > f'(u_R) \tag{2.7}$$

We note that $f'(u)$ is the characteristic speed ($u_t + f(u)_x = u_t + f'(u)u_x = 0$) [11].

(ii) (Entropy condition 2 and Non-convex scalar case):

$u(x,t)$ is the entropy solution condition if all discontinuities obey

$$\frac{f(u) - f(u_L)}{u - u_L} \geq s \geq \frac{f(u) - f(u_R)}{u - u_R} \tag{2.8}$$

for all $u_L \geq u \geq u_R$. This condition is due to Oleinik [13].

(iii) (Entropy condition 3 and Rarefaction case):

$u(x,t)$ is an entropy solution of (2.1) if there exists $E > 0$ such that for all $a > 0, t > 0, x \in \mathbb{R}$

$$\frac{u(x+a,t) - u(x,t)}{a} < \frac{E}{t} \tag{2.9}$$

Oleinik's original proof that an entropy solution to (2.1) satisfying (2.8) always exists proceeds by defining a discrete approximation and then taking limits as the resolution of the finite difference increases. This is the condition used in Theorem 16.1 (see [15]). For the discrete approximation, the Lax-Friedrichs approximations used in Oleinik's proof (see [15]).

2.2 Numerical Methods for Approximately Solving Scalar Conservation Laws

In this section, examples of numerical methods for the approximate solution of (2.1) and (2.2) are presented.

2.2.1 Some Definitions for Numerical Methods

We discretize the upper half of the $x - t$ plane by choosing a uniform spatial mesh size $\Delta x = x_{j+1} - x_j$ and a constant time step $\Delta t = t_{n+1} - t_n$, and define the discrete mesh points (x_j, t_n) by

$$x_j = j\Delta x, \quad j = \cdots, -1, 0, 1, 2, \cdots$$

$$t_n = n\Delta t, \quad n = 0, 1, 2, \cdots$$

Figure 2.1 shows a representation of the discrete mesh defined above.

The finite difference methods we will develop produce approximations $U_j^n \in \mathbb{R}^m$ to the solution $u(x_j, t_n)$ at the discrete mesh points. The point-wise values of the true solution will be denoted by $U_j^n \approx u(x_j, t_n)$. Using the point-wise approximation, U_j^n, we can define errors in the approximation to compute methods. In addition, the concept of convergence to a unique solution can be defined.

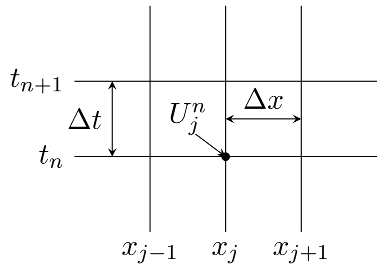


FIGURE 2.1: A representation of the mesh in space-time

The following presents some standard definitions used in the analysis of methods for approximation of the solutions of differential equations. These are included for completeness and for defining notation used in this work.

Definition 2.2. The Local Error and Convergence [11]

1. Local Error: The local error is the difference between the computed and true solutions

$$E_j^n = \underbrace{U_j^n}_{\text{computed solution}} - \underbrace{u_j^n}_{\text{true solution}}$$

2. Convergence: The approximate solution converges to the true solution if

$$\left\|E_j^n\right\|_1 \to 0$$

as $\Delta x, \Delta t \to 0$ where $\|E_k(\cdot,t)\|_1 = \int_{-\infty}^{\infty} |E_k(x,t)| \; dt$

For linear differential equations, convergence of a method will produce a consistent approximation that is stable. The following gives an intuitive definition for consistency and stability.

Definition 2.3. Consistency and Stability [2],[3]

1. Consistency: A numerical scheme is consistent if its discrete operator with finite differences converges to the continuous operator associated with the PDE as $\Delta x, \Delta t \to 0$.

2. Stability: The errors from any source will not grow without bound in time. That is, small perturbations due to round off and truncation errors are bounded.

Combining consistency with stability allows the statement and proof of the well known Lax Equivalence Theorem. The theorem is stated without proof (see [12]).

Theorem 2.4. ***Lax Equivalence Theorem*** *[12]*
For any discrete approximation the following is true.

$$\textit{Consistency + Stability} \Leftrightarrow \textit{Convergence}$$

As a result of the intuitive statement of the Lax Equivalence Theorem, we see that convergence is obtained when a numerical method produces a consistent and stable approximation. For the problems considered in this work, stability of the method results in the following condition.

Theorem 2.5. ***The Courant-Friedrichs-Lewy (CFL) condition*** *[6]*
For each x, t the mathematical domain of dependence must be contained in the numerical domain of dependence (see Figure 2.2):

$$X(x,t) \subseteq X_0(x,t)$$

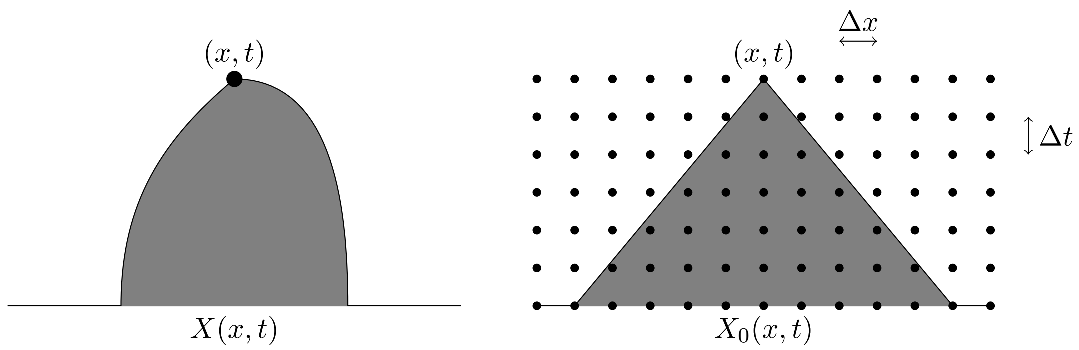


Figure 2.2: Mathematical and numerical domains of dependence

For example, for the linear advection equation (2.1), the discrete approximation

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{\Delta x}\left(U_j^n - U_{j-1}^n\right)$$

with $a > 0$, results in the CFL condition for the numerical method (referred to as the Upwind approximation) is

$$0 \leq \frac{a\Delta t}{\Delta x} \leq 1.$$

The Upwind method is one of the classical methods described in the next section.

2.2.2 Examples of Classical Methods

A general form for numerical methods for the approximate solution of conservation laws is given by

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}(\underbrace{F(U_j^n, U_{j+1}^n)}_{f(U^n_{j+\frac{1}{2}})} - \underbrace{F(U_{j-1}^n, U_j^n)}_{f(U^n_{j-\frac{1}{2}})})$$

Most approximate numerical methods for hyperbolic equations can be written in this form. From LeVeque ([11]), we list some classical/standard finite difference schemes for the scalar advection equation:

- Backward Euler:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{2\Delta x}a(U_{j+1}^n - U_{j-1}^n)$$

- Upwind:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}a(U_j^n - U_{j-1}^n)$$

- Downwind:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}a(U_{j+1}^n - U_j^n)$$

- Lax-Friedrichs:

$$U_j^{n+1} = \frac{1}{2}(U_{j+1}^n + U_{j-1}^n) - \frac{\Delta t}{2\Delta x}a(U_{j+1}^n - U_{j-1}^n)$$

- Leapfrog:

$$U_j^{n+1} = U_j^{n-1} - \frac{\Delta t}{2\Delta x}a(U_{j+1}^n - U_{j-1}^n)$$

- Lax-Wendroff:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{2\Delta x}a(U_{j+1}^n - U_{j-1}^n) + \frac{(\Delta t)^2}{2(\Delta x)^2}a^2(U_{j+1}^n - 2U_j^n + U_{j-1}^n)$$

- Beam-Warming:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{2\Delta x}a(3U_j^n - 4U_{j-1}^n + U_{j-2}^n) + \frac{(\Delta t)^2}{2(\Delta x)^2}a^2(U_j^n - 2U_{j-1}^n + U_{j-2}^n)$$

Note that these methods provide only a small subset of methods proposed over the past several decades.

2.2.3 Examples of Modern Methods

Modern schemes for hyperbolic conservation laws can be classified into following two categories [14]:

- **Flux-Splitting Methods** (characterized by algebraic construction): The basic idea is to add a switch such that the scheme becomes first order near discontinuities and maintains high order accuracy in smooth regions. Examples include *the Artificial Viscosity Method, Flux-Correction Transport (FCT), and Total Variation Diminishing (TVD) method.*
- **High-Order Godunov methods** (characterized by geometrical construction): The basic idea of slope-limiter methods is to generalize Godunov's method by replacing the piecewise constant representation of the solution by a more accurate representation. Examples include *Monotone Upstream Conservative Limited (MUSCL) schemes, the Piecewise Parabolic Method (PPM), and Essentially Non Oscillatory (ENO) schemes.*

For the first category, we will only describe TVD schemes and we will describe the MUSCL and ENO schemes for the second type of modern method.

2.2.3.1 Total Variation Diminishing (TVD) Scheme

Consider the linear advection equation where $f(u) = au, a > 0$. The numerical flux for a TVD scheme is

$$F(U_j^n, U_{j+1}^n) = f(u_{j+1/2}^n) = aU_j + C_{j+\frac{1}{2}} \left(\frac{1}{2} a(1 - a\frac{\Delta t}{\Delta x})(U_{j+1} - U_j) \right)$$

where $C_{j+\frac{1}{2}} := C(\theta_{j+\frac{1}{2}})$ is called a flux-limiter and depends on

$$\theta_{j+\frac{1}{2}} := \frac{U_j - U_{j-1}}{U_{j+1} - U_j}.$$

It has been proven,[16], that a scheme is TVD if $0 \leq \frac{C(\theta)}{\theta} \leq 2$ and $0 \leq C(\theta) \leq 2$. As $C(\theta)$ changes, a continuous family of methods can be defined. The following lists some examples:

- $C(\theta) = 1$ gives the Lax-Wendroff scheme.
- $C(\theta) = \theta$ gives the Beam-Warming scheme.
- Van Leer Limiter: $C(\theta) = \frac{\theta+|\theta|}{1+|\theta|}$
- Monotone central limiter: $C(\theta) = \max[0, \min(2\theta, \frac{(1+\theta)}{2}, 2)]$.
- Superbee limiter: $C(\theta) = \max(0, \min(1, 2\theta), \min(\theta, 2))$

Specifically the conditions necessary for a method to be TVD can be stated as follows.

Definition 2.6. A numerical method is said to be total variation diminishing (TVD) [11], if

$$TV(U^{n+1}) \leq TV(U^n).$$

The following gives a definition of the total variation

Definition 2.7. Denote the total variation (TV),

$$TV(v) = \sup \sum_{j=1}^{N} |v(x_j) - v(x_{j-1}|)$$

where the supremum is taken over all subdivisions of the real line $-\infty = x_0 < x_1 < \cdots < x_N = \infty$.

Another concept of importance is contained in the following definition

Definition 2.8. A scheme is said to be **monotone** if for two initial conditions u_j^0, v_j^0 with $u_j^0 \geq v_j^0$, then

$$u_j^n \geq v_j^n, \ \textit{for all } n \ .$$

A monotone scheme for a scalar conservation law can be shown to converge to a unique entropy satisfying solution.

Definition 2.9. A scheme is said to be monotonicity preserving if u^n is a monotone mesh function, then u^{n+1} is a monotonicity preserving schemes.

To relate TVD methods to other modern methods, Harten [9] has shown the following:

$$\text{Monotone scheme} \Rightarrow \text{TVD scheme} \Rightarrow \text{Monotonicity preserving scheme}$$

Figures 2.3 to 2.6 show various regions [16] related to TVD schemes.

- Figure 2.3: The shaded region shows where function values must lie for the method to be TVD. Second order linear methods have function $C(\theta)$ that leave this region.
- Figure 2.4: The shaded region is the Sweby region for second order TVD methods do not satisfy the constraints specified by the region in Figure 2.3.
- Figure 2.5: When the Superbee limiter is applied the result minimizes the effect of the limiter and maximizing steepening of shocks while maintaining the TVD constraints. The implication is $C(\theta)$ lies along the upper boundary of the TVD regions.
- Figure 2.6: Van Leer Limiter $C(\theta) = \frac{\theta+|\theta|}{1+|\theta|}$ defines a smooth variation.

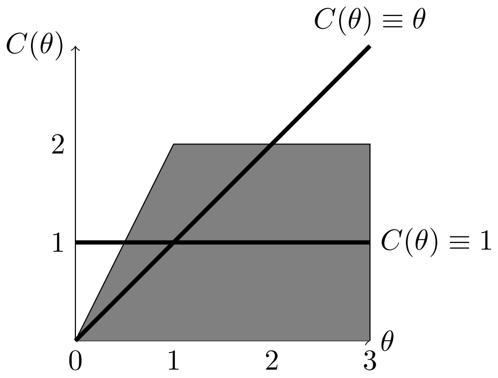


Figure 2.3: TVD region

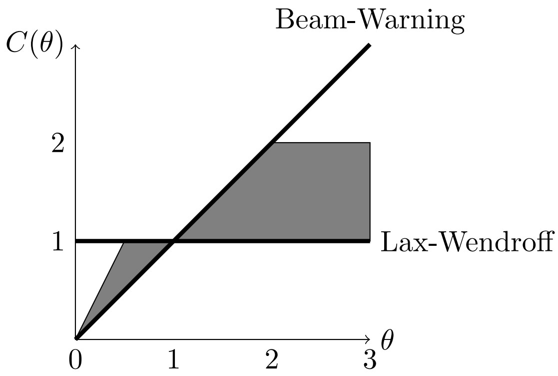


Figure 2.4: Second order TVD region

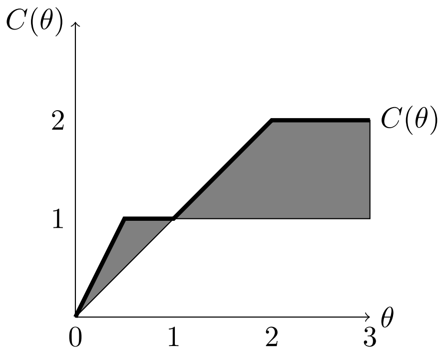


Figure 2.5: Superbee limiter

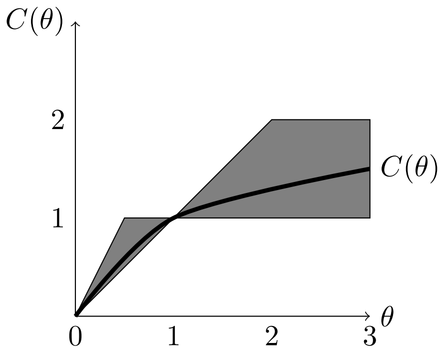


Figure 2.6: Van Leer Limiter

2.2.3.2 Monotone Upstream Conservative Limited Scheme

Finite volume methods can provide highly accurate numerical solutions for given systems, even in cases where the solution exhibits shocks, discontinuities, or large gradients. MUSCL [11] based numerical schemes extend the idea of using a linear piecewise approximation to each cell by using slope limited left and right extrapolated states. Figure 2.7 illustrates this idea.

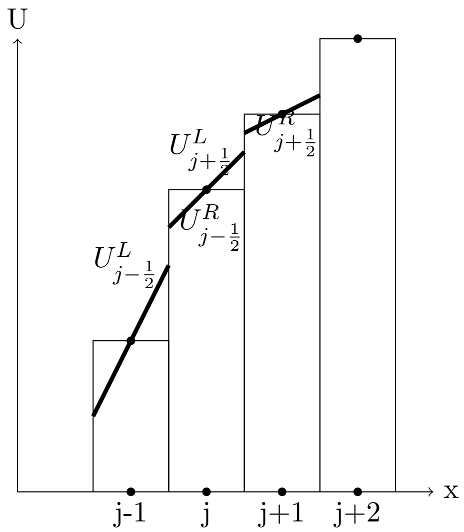


FIGURE 2.7: MUSCL scheme type left and right state linear extrapolation

The numerical flux for the MUSCL scheme [11] involves a nonlinear combination of first and second order approximations to the continuous flux function. The following steps can be used to compute values in the MUSCL scheme.

1. $U^n_{j+\frac{1}{2}} = U_{j+\frac{1}{2}}\left(U^L_{j+\frac{1}{2}}, U^R_{j+\frac{1}{2}}\right), \quad U^n_{j-\frac{1}{2}} = U_{j-\frac{1}{2}}\left(U^L_{j-\frac{1}{2}}, U^R_{j-\frac{1}{2}}\right)$
2. $U^L_{j+\frac{1}{2}} = U_j + 0.5\, C(\theta_j)(U_{j+1} - U_j), \quad U^R_{j+\frac{1}{2}} = U_{j+1} - 0.5\, C(\theta_{j+1})(U_{j+2} - U_{j+1})$
3. $U^L_{j-\frac{1}{2}} = U_{j-1} + 0.5\, C(\theta_{j-1})(U_j - U_{j-1}), \quad U^R_{j-\frac{1}{2}} = U_j - 0.5\, C(\theta_j)(U_{j+1} - U_j)$
4. $\theta_j = \frac{U_j - U_{j-1}}{U_{j+1} - U_j}$

The function $C(\theta_j)$ is a limiter function that limits the slope of the piecewise approximations to ensure the solution is TVD, thereby avoiding the spurious oscillations that would otherwise occur around discontinuities or shocks.

2.2.3.3 Essentially Non-Oscillatory (ENO) scheme

This type of scheme [14] goes back to the idea of the MUSCL schemes. The following reconstruction steps can be used in this case. The steps are based on the basic idea illustrated in Figure 2.7 and Figure 2.8.

1. Construct left and right slopes by connecting the average values in adjacent cells.
2. Select the downstream flux by using the smaller slope.

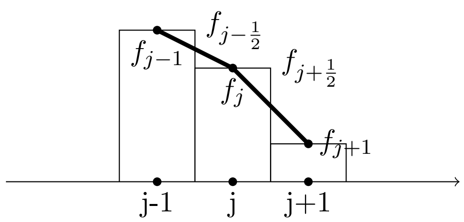


FIGURE 2.8: Example of ENO and Slope

This is an example of the steps needed to compute the second order ENO scheme for the linear advection equation $\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial x} = 0$.

1.
$$f_j^{n+1} = f_j^n - \frac{\Delta t}{\Delta x}a(f_{j+\frac{1}{2}}^n - f_{j-\frac{1}{2}}^n)$$

2.
$$\Delta f_j^+ = f_{j+1} - f_j, \quad \Delta f_j^- = f_j - f_{j-1}$$

3.
$$f_{j+\frac{1}{2}} = \begin{cases} f_j + \frac{1}{2}amin\left(\Delta f_j^+, \Delta f_j^-\right), & \textbf{if } \frac{1}{2}\left(U_j + U_{j+1}\right) > 0 \\ f_j - \frac{1}{2}amin\left(\Delta f_{j+1}^+, \Delta f_{j+1}^-\right), & \textbf{if } \frac{1}{2}\left(U_j + U_{j+1}\right) < 0 \end{cases}$$

4.
$$amin(a,b) = \begin{cases} a & , |a| < |b| \\ b & , |b| \le |a| \end{cases}$$

2.3 Example Results from Numerical Methods

In this section, results for a single problem are modeled to illustrate the behavior from various methods described in Section 2.2. The results are organized by the types and categories of problems described in Section 2.2.

2.3.1 Numerical Examples: Linear Scalar Conservation Laws

We consider the Riemann problem for the scalar advection equation

$$u_t + au_x = 0, \quad -\infty < x < \infty, \;\; t \geq 0, \tag{2.10}$$

with initial condition,

$$u_0(x) = \begin{cases} 1 & \text{if } x < 0, \\ 0 & \text{if } x > 0. \end{cases} \tag{2.11}$$

Figures 2.9, 2.10, and 2.11 show numerical and exact solutions to the problem defined by (2.10) and (2.11) computed with various methods. In all cases $a = 1$ and the results are plotted at time $t = 0.5$. We can see that the first order methods (Upwind and Lax-Friedrichs) give very smeared out solutions, and the second order methods (Lax-Wendroff and Beam-Warming) exhibit spurious oscillations. The best method for the scalar advection equation is a TVD scheme.

In Figure 2.9, $\Delta x = 0.01$ while a finer grid with $\Delta x = 0.002$ is used in Figure 2.10 for the Upwind, Lax-Friedrichs, Lax-Wendroff, and Beam-Warming schemes. Also, in Figure 2.11, $\Delta x = 0.01$ and $\Delta x = 0.002$ for the TVD scheme.

2.3.2 Numerical Examples: Nonlinear Conservation Laws

The simplest nonlinear example of a conservation law is the inviscid Burgers' equation with $f(u) = \frac{1}{2}u^2$. That is,

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0.$$

The inviscid Burgers' equation is an example of nonlinear conservation law. Any conservation law that involves a nonlinear flux function $f(u)$, will produce a nonlinear conservation law.

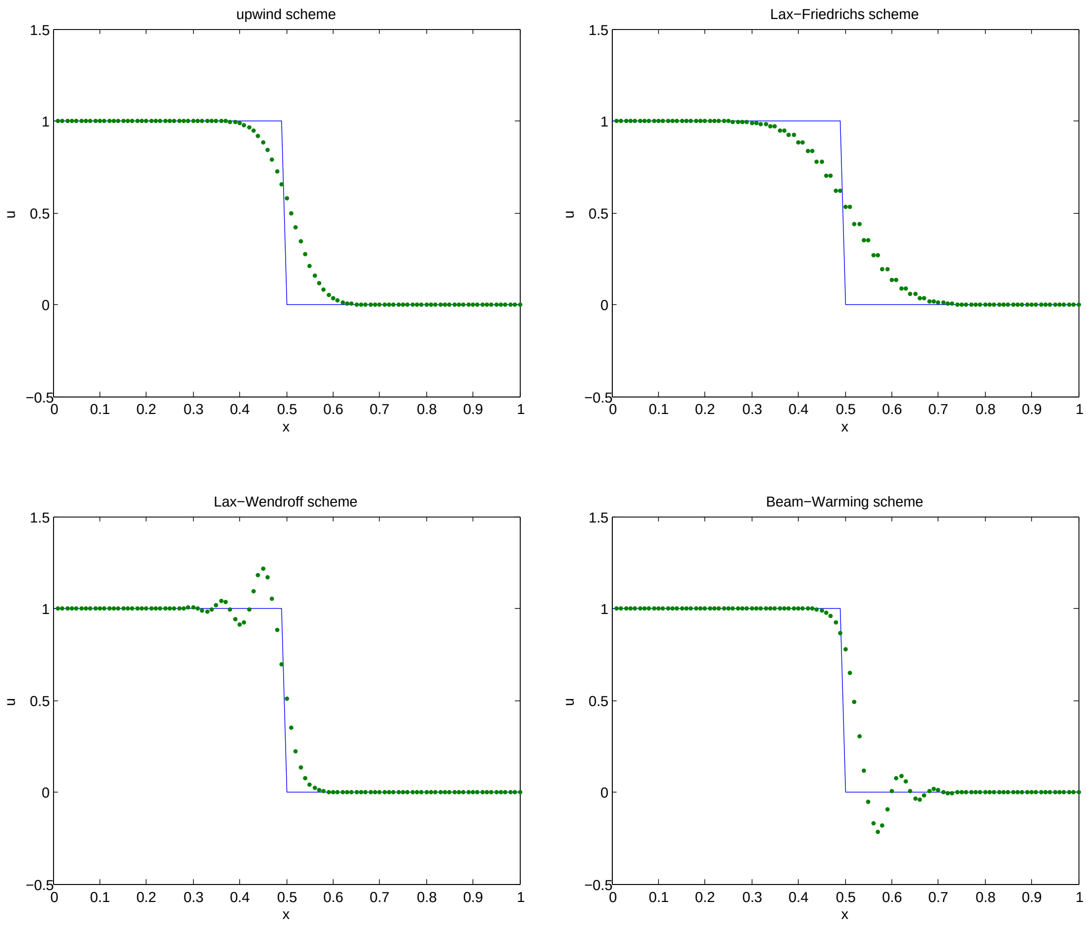


FIGURE 2.9: Numerical and exact solution to (2.10) with $\Delta x = 0.01$

From [2] and [21], we list some finite difference schemes for Burger's equation:

- Upwind:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}(f(U_j^n) - f(U_{j-1}^n)).$$

  For Burgers' equation we have

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\left[\frac{1}{2}(U_j^n)^2 - \frac{1}{2}(U_{j-1}^n)^2\right]$$

- Lax-Friedrichs:

$$U_j^{n+1} = \frac{1}{2}(U_{j+1}^n + U_{j-1}^n) - \frac{\Delta t}{2\Delta x}(f(U_{j+1}^n) - f(U_{j-1}^n)),$$

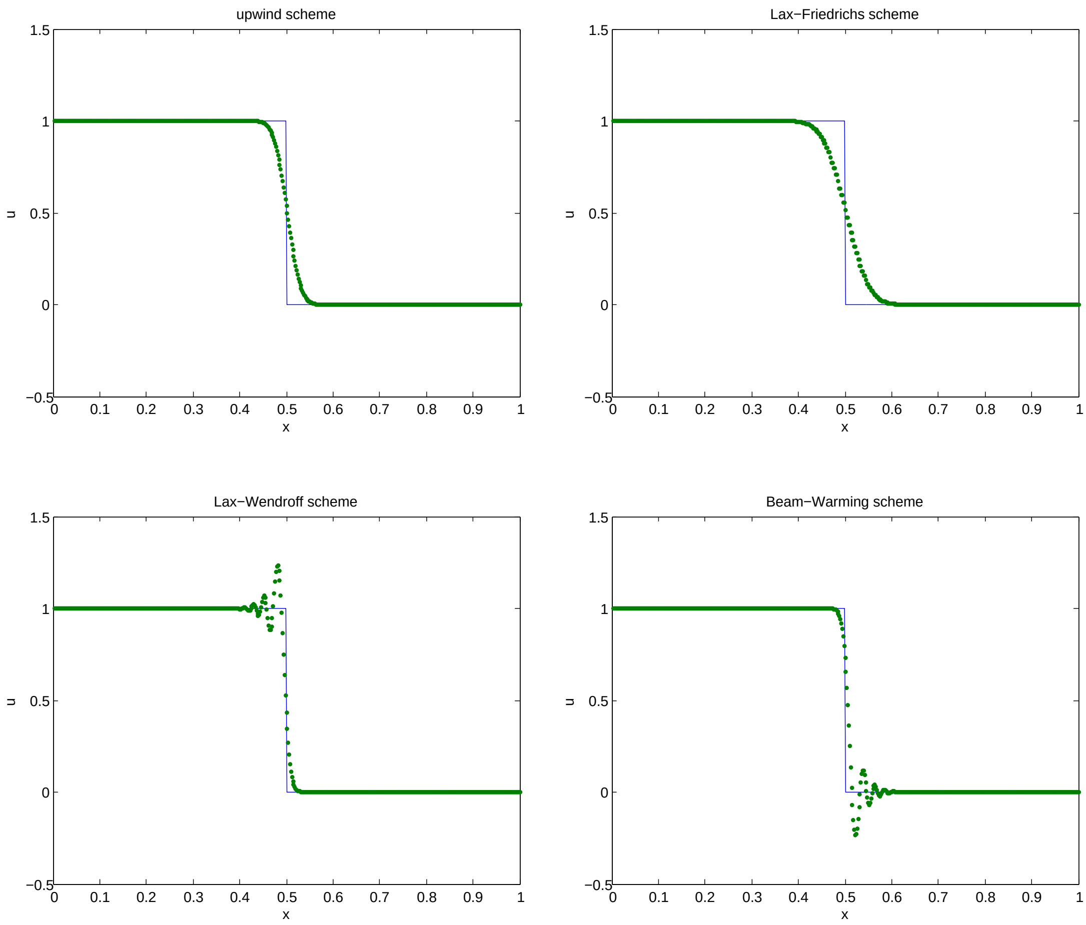


FIGURE 2.10: Numerical and exact solution to (2.10) with $\Delta x = 0.002$

For Burgers' equation we have

$$U_j^{n+1} = \frac{1}{2}(U_{j+1}^n + U_{j-1}^n) - \frac{\Delta t}{2\Delta x}\left[\frac{1}{2}(U_{j+1}^n)^2 - \frac{1}{2}(U_{j-1}^n)^2\right].$$

- Lax-Wendroff:

$$\begin{aligned} U_j^{n+1} = U_{j+1}^n &- \frac{\Delta t}{2\Delta x}(f(U_{j+1}^n) - f(U_{j-1}^n)) \\ &+ \frac{(\Delta t)^2}{2(\Delta x)^2}\left[A_{j+\frac{1}{2}}(f(U_{j+1}^n) - f(U_j^n)) - A_{j-\frac{1}{2}}(f(U_j^n) - f(U_{j-1}^n))\right] \end{aligned}$$

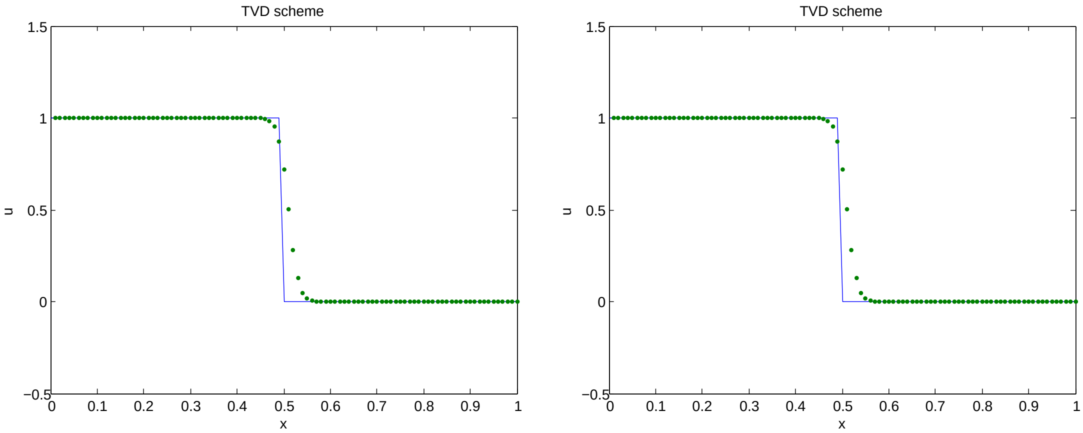


FIGURE 2.11: Numerical and exact solution to (2.10) with $\Delta x = 0.01$ and $\Delta x = 0.002$ for TVD scheme

where $A_{j\pm\frac{1}{2}}$ is evaluated at $\frac{1}{2}(U_j^n + U_{j\pm1}^n)$. For Burgers' equation we have $f'(u) = u$ so

$$\begin{aligned}
U_j^{n+1} =& U_{j+1}^n - \frac{\Delta t}{2\Delta x}(\frac{1}{2}(U_{j+1}^n)^2 - \frac{1}{2}(U_{j-1}^n)^2) \\
&+ \frac{(\Delta t)^2}{2(\Delta x)^2}[(\frac{1}{2}(U_j^n + U_{j+1}^n))(\frac{1}{2}(U_{j+1}^n)^2 - \frac{1}{2}(U_j^n)^2) \\
&- (\frac{1}{2}(U_j^n + U_{j-1}^n))(\frac{1}{2}(U_j^n)^2 - \frac{1}{2}(U_{j-1}^n)^2)]
\end{aligned}$$

- Beam-Warming:

$$\begin{aligned}
U_j^{n+1} &= U_{j+1}^n - \frac{\Delta t}{2\Delta x}(3f(U_j^n) - 4f(U_{j-1}^n) + f(U_{j-2}^n)) \\
&+ \frac{(\Delta t)^2}{2(\Delta x)^2}\left[A_{j+\frac{1}{2}}(f(U_j^n) - f(U_{j-1}^n)) - A_{j-\frac{1}{2}}(f(U_{j-1}^n) - f(U_{j-2}^n))\right]
\end{aligned}$$

where $A_{j\pm\frac{1}{2}}$ is evaluated at $\frac{1}{2}(U_j^n + u_{j\pm1}^n)$. For Burgers' equation we have $f'(u) = u$ so

$$\begin{aligned}
U_j^{n+1} &= U_{j+1}^n - \frac{\Delta t}{2\Delta x}(\frac{1}{2}3(U_j^n)^2 - \frac{1}{2}4(U_{j-1}^n)^2 + \frac{1}{2}(U_{j-2}^n)^2) \\
&+ \frac{(\Delta t)^2}{2(\Delta x)^2}[(\frac{1}{2}(U_j^n + U_{j-1}^n))(\frac{1}{2}(U_j^n)^2 - \frac{1}{2}(U_{j-1}^n)^2) \\
&- (\frac{1}{2}(U_{j-2}^n + U_{j-1}^n))(\frac{1}{2}(U_{j-1}^n)^2 - \frac{1}{2}(U_{j-2}^n)^2)]
\end{aligned}$$

For the examples in this section, consider Burgers' equation:

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0, \tag{2.12}$$

with initial condition,

$$u_0(x) = \begin{cases} 1 & \text{if } x < 0, \\ 0 & \text{if } x > 0. \end{cases} \tag{2.13}$$

Figures 2.12 and 2.13, show numerical approximations and exact solutions to Burgers' equation (2.12) and (2.13) computed using same Δx values of advection equation case. Figure 2.12 and 2.13 used $\Delta x = 0.01$ and $\Delta x = 0.002$. We see the same type of results when the first order methods an applied (Upwind and Lax-Friedrichs). These methods produce smeared solutions. The second order methods (Lax-Wendroff and Beam-Warming) produce spurious oscillation in the solutions.

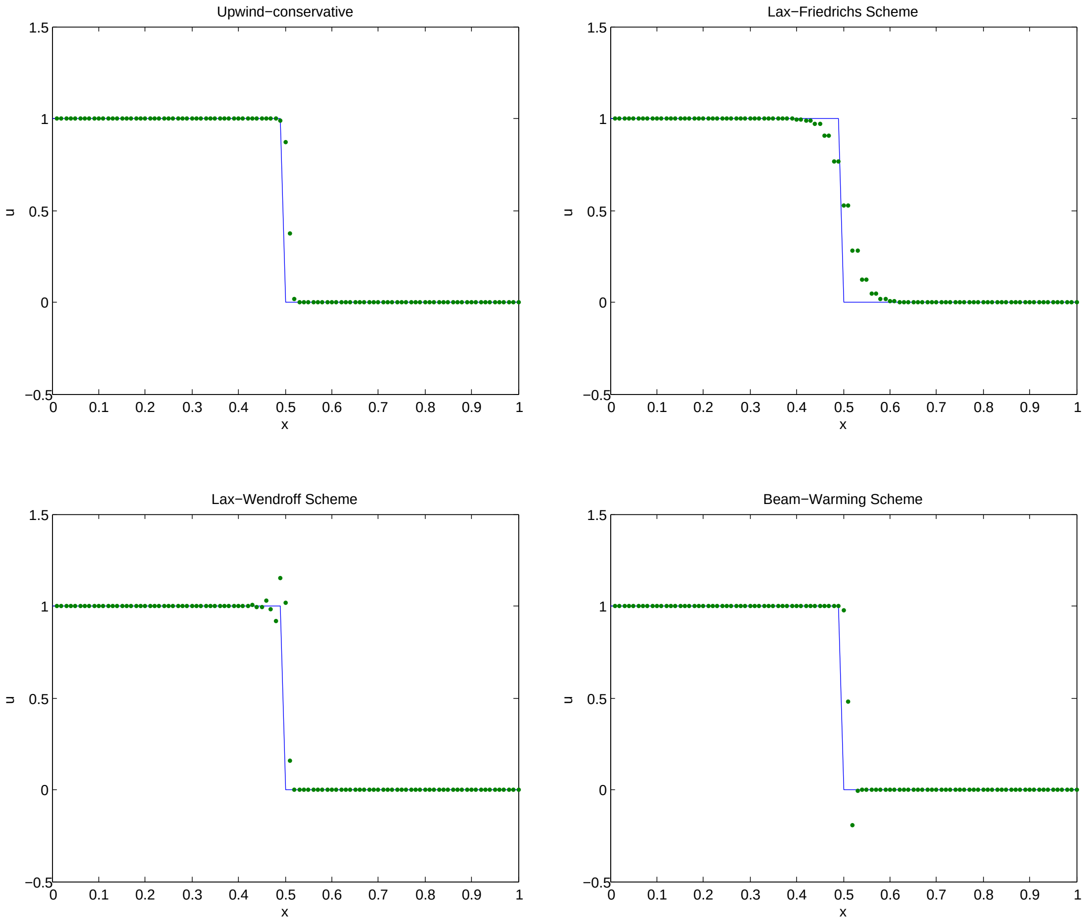


FIGURE 2.12: Numerical and exact solution to (2.11) with $\Delta x = 0.01$

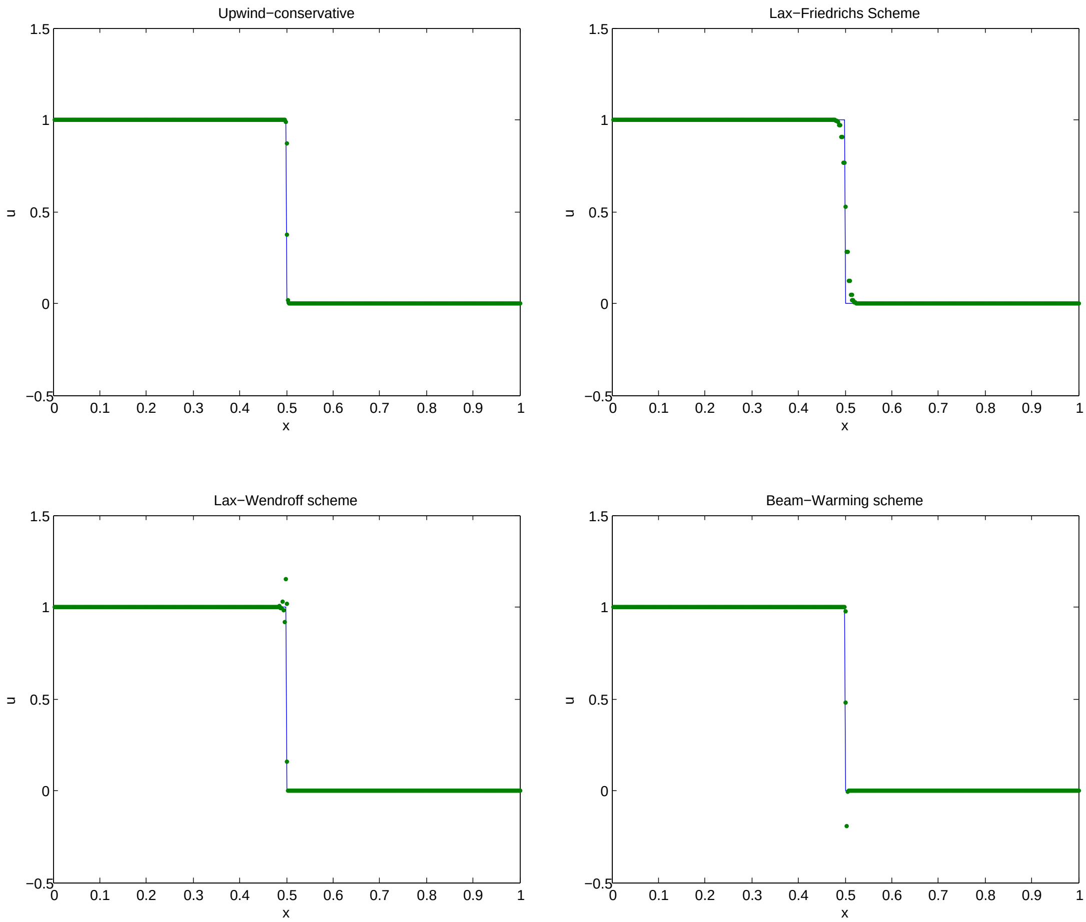


FIGURE 2.13: Numerical and exact solution to (2.11) with $\Delta x = 0.002$

CHAPTER 3
POLYNOMIAL FRAMEWORK FOR CONSTRUCTING APPROXIMATE SOLUTION METHODS

This chapter covers yet another representation of numerical methods for conservation laws. In this chapter, we will consider a polynomial framework based on [4],[5]. We start with some definitions and examples using the polynomial framework of $U^n_{j\pm\frac{1}{2}}$.

3.1 Definition and Notation

Definition 3.1. From [14], a scheme for approximately solving conservation laws is called conservative if and only if it can be written as

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}(\underbrace{f(U^n_{j+\frac{1}{2}})}_{F^n_{j+\frac{1}{2}}} - \underbrace{f(U^n_{j-\frac{1}{2}})}_{F^n_{j-\frac{1}{2}}}), \tag{3.1}$$

where f is

1. Lipschitz continuous, and
2. $f(u, \cdots, u) = f(u)$ (consistency)

Note. A function f from $S \subset \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous at $x \in S$ if there is a constant C such that

$$\|f(y) - f(x)\| \leq C\,\|y - x\|$$

for all $y \in S$ sufficiently near x [2].

From [12] and [14], examples of conservative methods of the form (3.1) for approximately solving conservation laws include the following.

- Godunov scheme:

$$F^n_{j+\frac{1}{2}} = \begin{cases} \min\limits_{U_j \le U \le U_{j+1}} f(U) & U_j < U_{j+1} \\ \max\limits_{U_j \le U \le U_{j+1}} f(U) & U_j \ge U_{j+1} \end{cases}$$

- Lax-Friedrichs scheme:

$$F^n_{j+\frac{1}{2}} = \frac{1}{2}\left[f(U_j) + f(U_{j+1}) - \alpha(U_{j+1} - U_j)\right]$$

where $\alpha = \max\limits_U \left|f'(U)\right|$

- Roe scheme:

$$F^n_{j+\frac{1}{2}} = \begin{cases} f(U^n_j) & a_{j+\frac{1}{2}} \ge 0 \\ f(U^n_{j+1}) & a_{j+\frac{1}{2}} < 0 \end{cases}$$

where $a_{j+\frac{1}{2}} = \frac{f(U_{j+1}) - f(U_j)}{U_{j+1} - U_j}$

- Engquist-Osher scheme:

$$F^n_{j+\frac{1}{2}} = f^+(U_j) + f^-(U_{j+1})$$

where

$$f^+(U) = \int_0^U \max(f'(U), 0)\, dU + f(0),$$

$$f^-(U) = \int_0^U \min(f'(U), 0)\, dU$$

- Lax-Wendroff scheme:

$$F^n_{j+\frac{1}{2}} = \frac{1}{2}\left[f(U_j) + f(U_{j+1}) - \frac{\Delta t}{\Delta x} f'(U_{j+\frac{1}{2}})\left(f(U_{j+1}) - f(U_j)\right)\right]$$

- MacCormark scheme:

$$U^{n+\frac{1}{2}}_j = U^n_j - \frac{\Delta t}{\Delta x}\left(f(U^n_j) - f(U^n_{j-1})\right)$$

$$U^{n+1}_j = \frac{1}{2}\left[U^n_j + U^{n+\frac{1}{2}}_j + \frac{\Delta t}{\Delta x}\left[f(U^{n+\frac{1}{2}}_{j+1}) - f(U^{n+\frac{1}{2}}_j)\right]\right]$$

$$f(U^n_{j+\frac{1}{2}}) = \frac{1}{2}\left[f(U^n_j) + f(U^n_j - \frac{\Delta t}{\Delta x}\left(f(U^n_j) - f(U^n_{j-1})\right))\right]$$

Note that many schemes (e.g., Upwind scheme) are also conservative.

3.2 Description of a Polynomial Framework for Approximate Solution of Conservation Laws

In this section we focus on a representation of the numerical flux, $F^n_{j+\frac{1}{2}} = f(U^n_{j+\frac{1}{2}})$ given by

$$U^n_{j+\frac{1}{2}} \approx u(x_{j+\frac{1}{2}}, t_n) = u(x_j + \frac{\Delta x}{2}, t_n).$$

By choosing a single representation between the two values U^n_j and U^n_{j+1}, the numerical scheme will always be conservation if we use

$$F^n_{j\pm\frac{1}{2}} = f(U^n_{j\pm\frac{1}{2}}).$$

The framework in the next section assumes the representation is in terms of a local polynomial function.

3.2.1 Definition of Grid-line Values: $U^n_{j\pm\frac{1}{2}}$

Using the ideas in [4],[5] to define $U^n_{j\pm\frac{1}{2}}$,

$$U^n_{j\pm\frac{1}{2}} = m^n_{j\pm\frac{1}{2}}(\alpha^n_{j\pm\frac{1}{2}}) = \sum_{i=0}^{l} a^n_{i,j\pm\frac{1}{2}}(\alpha^n_{j\pm\frac{1}{2}})^i,$$

where $m_{j\pm\frac{1}{2}}$ is a function chosen to represent the unknown u between neighboring nodes. Also, $\alpha^n_{j\pm\frac{1}{2}}$ is chosen as a point which is some distance upwind from the grid line $x_{j\pm\frac{1}{2}}$. This idea is illustrated in Figure 2.1. Thus the grid line values $U^n_{j\pm\frac{1}{2}}$ are in fact approximations of the unknown at the grid lines $x_{j\pm\frac{1}{2}}$ for the next time level, t_{n+1}. We use an evaluation point $\alpha_{j\pm\frac{1}{2}}$ of the form

$$\alpha^n_{j\pm\frac{1}{2}} = \gamma_1 + \gamma_2 \lambda f'(u^n_{i,j\pm\frac{1}{2}})$$

where $\lambda = \frac{\Delta t}{\Delta x}$ and the parameters γ_1, γ_2 are given. The coefficients, $a_{i,j\pm\frac{1}{2}}$, can be chosen so that the resulting numerical scheme satisfies given accuracy, stability, consistency, symmetry, and other conditions (e.g., TVD conditions), see [4],[5].

3.2.2 Examples Using the Polynomial Definition for $U^n_{j\pm\frac{1}{2}}$

3.2.2.1 Linear Conservation Law Examples

For the linear advection case, we can use a linear polynomial to represent the solution between nodes [4],

$$U^n_{j+\frac{1}{2}} = m^n_{j+\frac{1}{2}}(\alpha^n_{j+\frac{1}{2}}) = U^n_{j+1} + (U^n_j - U^n_{j+1})\alpha^n_{j+\frac{1}{2}}$$

$$U^n_{j-\frac{1}{2}} = m^n_{j-\frac{1}{2}}(\alpha^n_{j-\frac{1}{2}}) = U^n_j + (U^n_{j-1} - U^n_j)\alpha^n_{j-\frac{1}{2}}$$

This defines a continuous family of numerical approximation of the solution at the grid lines in the mesh. The following list of methods shows the relationship between the linear polynomial representation and some well known methods.

- When $\alpha^n_{j\pm\frac{1}{2}} = 0$:

  $$U^n_{j+\frac{1}{2}} = U^n_{j+1}$$

  $$U^n_{j-\frac{1}{2}} = U^n_j$$

  This choice generates a downwind scheme.

  $$U^{n+1}_j = U^n_j - \frac{\Delta t}{\Delta x} a(U^n_{j+1} - U^n_j)$$

  This method is clearly not stable and is of no real use.

- When $\alpha^n_{j\pm\frac{1}{2}} = 1$:

  $$U^n_{j+\frac{1}{2}} = U^n_j$$

  $$U^n_{j-\frac{1}{2}} = U^n_{j-1}$$

  This choice generates an upwind scheme.

  $$U^{n+1}_j = U^n_j - \frac{\Delta t}{\Delta x} a(U^n_j - U^n_{j-1})$$

- When $\alpha^n_{j\pm\frac{1}{2}} = \frac{1}{2} + \frac{1}{2}a\frac{\Delta t}{\Delta x}$,

$$\begin{aligned} U^n_{j+\frac{1}{2}} &= m^n_{j+\frac{1}{2}}(\alpha^n_{j+\frac{1}{2}}) \\ &= U^n_{j+1} + (U^n_j - U^n_{j+1})(\frac{1}{2} + \frac{1}{2}a\frac{\Delta t}{\Delta x}) \\ &= \frac{1}{2}(U^n_{j+1} + U^n_j) + \frac{\Delta t}{2\Delta x}a(U^n_j - U^n_{j+1}), \end{aligned}$$

$$\begin{aligned} U^n_{j-\frac{1}{2}} &= m^n_{j-\frac{1}{2}}(\alpha^n_{j-\frac{1}{2}}) \\ &= U^n_j + (U^n_{j-1} - U^n_j)(\frac{1}{2} + \frac{1}{2}a\frac{\Delta t}{\Delta x}) \\ &= \frac{1}{2}(U^n_j + U^n_{j-1}) + \frac{\Delta t}{2\Delta x}a(U^n_{j-1} - U^n_j). \end{aligned}$$

  This generates the Lax-Wendroff scheme.

$$U^{n+1}_j = U^n_j - a\frac{\Delta t}{\Delta x}\left[\frac{1}{2}(U^n_{j+1} - U^n_{j-1}) - \frac{\Delta t}{2\Delta x}a(U^n_{j+1} - 2U^n_j + U^n_{j-1})\right]$$

- When $\alpha^n_{j\pm\frac{1}{2}} = \frac{1}{2} - \frac{1}{2}a\frac{\Delta t}{\Delta x}$,

$$\begin{aligned} U^n_{j+\frac{1}{2}} &= m^n_{j+\frac{1}{2}}(\alpha^n_{j+\frac{1}{2}}) \\ &= U^n_j - (U^n_{j-1} - U^n_j)(\frac{1}{2} - \frac{1}{2}a\frac{\Delta t}{\Delta x}) \\ &= \frac{1}{2}(3U^n_j - 4U^n_{j-1}) + \frac{\Delta t}{2\Delta x}a(U^n_{j-1} - U^n_j), \end{aligned}$$

$$\begin{aligned} U^n_{j-\frac{1}{2}} &= m^n_{j-\frac{1}{2}}(\alpha^n_{j-\frac{1}{2}}) \\ &= U^n_{j-1} - (U^n_{j-2} - U^n_{j-1})(\frac{1}{2} - \frac{1}{2}a\frac{\Delta t}{\Delta x}) \\ &= \frac{1}{2}(3U^n_{j-1} + U^n_{j-2}) + \frac{\Delta t}{2\Delta x}a(U^n_{j-2} - U^n_{j-1}). \end{aligned}$$

  This generates the Beam-Warming scheme.

$$U^{n+1}_j = U^n_j - a\frac{\Delta t}{\Delta x}\left[\frac{1}{2}(3U^n_j - 4U^n_{j-1} + U^n_{j-2}) - \frac{\Delta t}{2\Delta x}a(U^n_j - 2U^n_{j-1} + U^n_{j-2})\right]$$

From these examples, it is clear there is a one to one correspondence between well known methods and specific choices of parameters in this polynomial based method.

3.2.2.2 Nonlinear Conservation Law Examples

If f is nonlinear with $f' > 0$, then by the results in section 3.2.1,

$$U^n_{j\pm\frac{1}{2}} = m^n_{j\pm\frac{1}{2}}(\alpha^n_{j\pm\frac{1}{2}}) = \sum_{i=0}^{l} a^n_{i,j\pm\frac{1}{2}}(\alpha^n_{j\pm\frac{1}{2}})^i,$$

where we choose to represent the evaluation point

$$\alpha^n_{j\pm\frac{1}{2}} = \gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f'(U^n_{j\pm\frac{1}{2}}).$$

So,

$$U^n_{j\pm\frac{1}{2}} = m^n_{j\pm\frac{1}{2}}(\alpha^n_{j\pm\frac{1}{2}}) = \sum_{i=0}^{l} a_{i,j\pm\frac{1}{2}}(\gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f'(U^n_{j\pm\frac{1}{2}}))^i. \tag{3.2}$$

The results is an implicit relationship for $U^n_{j\pm\frac{1}{2}}$. This relationship is a nonlinear relationship which requires a solution or an approximate solution. In general, an algebraic solution will not be available. So, it will be necessary to approximate the solution of the nonlinear equation. Newton's method provides one way to approximate the solution of the nonlinear equation for $U^n_{j\pm\frac{1}{2}}$. To that end, what follows describes an implementation of Newton's method for the specific problem of approximating $U^n_{j\pm\frac{1}{2}}$. We define

$$G(U^n_{j\pm\frac{1}{2}}) = U^n_{j\pm\frac{1}{2}} - \sum_{i=0}^{l} a_{i,j\pm\frac{1}{2}}(\gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f'(U^n_{j\pm\frac{1}{2}}))^i.$$

If $G(U^n_{j\pm\frac{1}{2}}) = 0$, then (3.2) is satisfied.

Let $v = U^n_{j\pm\frac{1}{2}}$, then

$$G(v) = v - \sum_{i=0}^{l} a_{i,j\pm\frac{1}{2}}(\gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f'(v))^i.$$

Applying Newton's method generates the following iteration formula.

$$v_{k+1} = v_k - \frac{v_k - \sum_{i=0}^{l} a_{i,j\pm\frac{1}{2}}(\gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f'(v))^i}{1 - \sum_{i=0}^{l} a_{i,j\pm\frac{1}{2}}(\gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f''(v))^i}$$

Namely,

$$\left(U^n_{j+\frac{1}{2}}\right)_{k+1} = \left(U^n_{j+\frac{1}{2}}\right)_k - \frac{\left(U^n_{j+\frac{1}{2}}\right)_k - \sum_{i=0}^{l} a_{i,j+\frac{1}{2}}(\gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f'(\left(U^n_{j+\frac{1}{2}}\right)_k))^i}{1 - \sum_{i=0}^{l} a_{i,j+\frac{1}{2}}(\gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f''(\left(U^n_{j+\frac{1}{2}}\right)_k))^i}.$$

For example, in the case of the Lax-Wendroff method for nonlinear problems,

$$U^n_{j+\frac{1}{2}} = U^n_{j+1} + (U^n_j - U^n_{j+1})(\frac{1}{2} + \frac{1}{2}\frac{\Delta t}{\Delta x} f'(U^n_{j+\frac{1}{2}}))$$

From above, we can define,

$$G(v) = v - U^n_{j+1} + (U^n_j - U^n_{j+1})(\frac{1}{2} + \frac{1}{2}\frac{\Delta t}{\Delta x} f'(v)),$$

$$G'(v) = 1 - U^n_{j+1} + (U^n_j - U^n_{j+1})(\frac{1}{2}\frac{\Delta t}{\Delta x} f''(v)).$$

Thus if bounds are known on the function f'' a range of $\frac{\Delta t}{\Delta x}$ can be chosen so that the denominator is never zero. Therefore, we can easily solve problems for nonlinear conservation laws. Note that we have assumed f'' is convex or $f'' > 0$ for all $U \in \mathbb{R}$.

CHAPTER 4
SHAPE FUNCTION CONSTRUCTION FOR MULTIRESOLUTION ANALYSIS (MRA)

We now explain a relationship between the construction of a Multiresolution Analysis (MRA) and the polynomial framework from the previous chapter. In this section, a historical perspective about wavelets is given. Also, we will develop an algorithm for use in the construction of MRA shape functions using the polynomial framework approach for conservation laws.

4.1 History of Wavelets and Multiresolution Analysis

A wavelet is a mathematical function useful in digital signal processing, image compression and many other applications. The use of wavelets for these purposes is a recent development, although the theory is not new.

4.1.1 Prior to 1930

The principles are similar to those of Fourier analysis, which was first developed in the early part of the 19th century [1]. Joseph Fourier introduced Fourier analysis in 1805. His work is based on the fact that certain classes of functions can be represented as infinite sums of sine and cosine functions. Simply stated any 2π periodic function $f(x)$ can be represented by

$$f(x) = a_0 + \sum_{k=1}^{\infty}(a_k \cos(kx) + b_k \sin(kx))$$

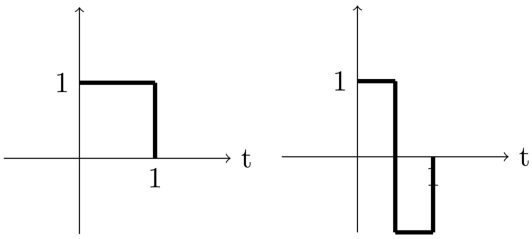
where the coefficients are defined using orthogonality of trigonometric functions and the formulas for the coefficients are given by

$a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(x)\ \ dx$

$a_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos(kx)\ \ dx$

$b_k = \int_0^{2\pi} f(x)\sin(kx)\ \ dx$

Alfred Haar(1909) found another set of basis functions to analyze signals. One property of the Haar wavelet is that it has compact support, which means that it vanishes outside a finite interval. In comparison the trigonometric basis functions in the Fourier series have nonzero support defined on the entire real line. Haar wavelets are not continuously differentiable which somewhat limits their application. In contrast the trigonometric functions are analytic. Figure 4.1 shows (a) a scaling or shape function and (b) a detail



(a) (b)

FIGURE 4.1: Haar basis function

function used as part of a basis for the Haar representation of a function. The definitions for these functions are

(a)

$$\varphi(t) = \begin{cases} 1 & t \in [0,1), \\ 0 & t \notin [0,1). \end{cases}$$

(b)

$$\psi(t) = \begin{cases} 1 & t \in [0,1/2), \\ -1 & t \in [1/2,1), \\ 0 & t \notin [0,1). \end{cases}$$

4.1.2 1930 through 1940

In the 1930s, several groups working independently researched the representation of functions using scale-varying basis functions. Paul Levy found Haar basis functions superior to the Fourier basis function for studying small complicated details in Brownian motion. He discovered that the scale-varying basis functions created by Haar formed a better basis than the Fourier basis functions.

Remark 4.1. Scale-varying basis functions [20]
A basis function varies in scale by chopping up the same function or data space using different scale sizes. For example, if we have a signal over the domain from 0 to 1, we can divide the signal with two step functions that range from 0 to 1/2 and 1/2 to 1. Then we can divide the original signal again using four step functions from 0 to 1/4, 1/4 to 1/2, 1/2 to 3/4, and 3/4 to 1. This can be repeated through all dyadic scales.

4.1.3 1940 through 1980

In 1946, Dennis Gabor introduced a time-localization or "window function" to extract local information of the Fourier transform of the signal. He suggested a signal with time and frequency as coordinates. The method is a short time Fourier transform (STFT).

$$F(\omega, \tau) = \int_{-\infty}^{\infty} f(t) g(t - \tau) e^{j\omega t} \, dt,$$

$g(t)$ is called a window function. Between 1960 and 1980, the mathematicians Guido Weiss and Ronald R. Coifman studied the simplest elements of a function space, called atoms, with the goal of finding the atoms for a common function and finding the "assembly rules" that allow the reconstruction of all the elements of the function space using these atoms.

4.1.4 1980 to the Present Day

In 1984, Morlet and Grossman developed a continuous wavelet transform. Meyer (1985) discovered the first smooth orthogonal wavelet basis functions with better time and frequency localization. In 1986, Mallat collaborated with Meyer to develop mutiresolution analysis theory (MRA), discrete wavelet transform (DWT), wavelet construction techniques. In the most recent stage of developing wavelets, Ingrid Daubechies found compact and orthogonal wavelets. Daubechies wavelets are a family of orthogonal wavelets that define a discrete wavelet transform, characterized by a maximal number of vanishing moments for some given support. In 1994, Swelden developed the lifting scheme for the construction of wavelet shape functions. In the next section, we consider the lifting scheme.

4.2 The Lifting Scheme

Swelden ([17],[18] and [19]) developed a general framework for the construction of fast wavelet transforms of signals referred to as lifting. The algorithm has two advantages; first, it doesn't require the machinery of Fourier analysis as a prerequisite and it can easily be generalized to complex geometric situations which typically occur in computer graphics and other real world problems. The lifting scheme has three operations:

1. Spliting: The signal is split into even and odd indexed samples.

$$(\{U_{k,2j}\}, \{U_{k,2j+1}\}) := S(\{U_{k,j}\})$$

   where S is the splitting operator. This step identifies the odd and even indexed samples in the signal.

2. Prediction: In this operation the odd samples are predicted using the even samples.

$$d_{k-1,j} = U_{k,2j+1} - P(\{U_{k,2j}\})$$

   This step replaces the odd samples with the detail given by the difference between the sample values and the predicted value.

3. Update: In this operation the even samples, $U_{k,2j}$, are updated using the details computed in the previous step. (e.g., overall signal average)

$$U_{k-1,j} = U_{k,2j} + Q(\{d_{k-1,j}\})$$

   This step replaces the even indexed samples with an average value.

k indicates the refinement level of the signal and j indicates the j^{th} sample at the k^{th} level. A simple lifting scheme forward transform is diagrammed in Figure 4.2.

4.2.1 Construction of the Haar Multiresolution Analysis via Lifting

In the lifting scheme version of the Haar transform (see [17],[18]), the prediction step predicts that the odd element will be equal to the even element. The difference between the predicted value and the actual value of the odd element replaces the odd

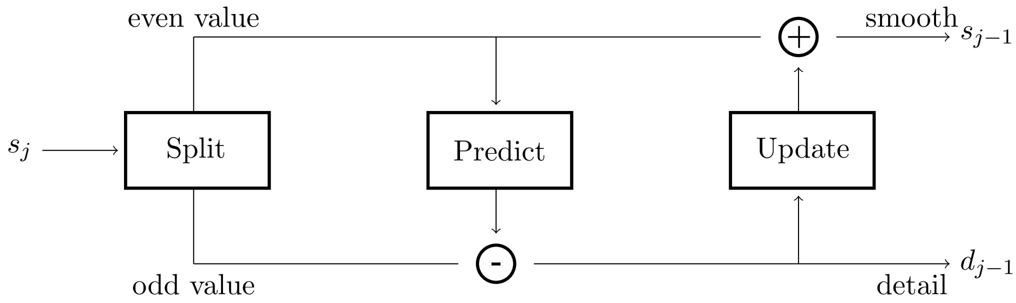


FIGURE 4.2: Lifting scheme forward wavelet transform

element

$$U^*_{2j+1} = U_{2j+1} - U_{2j}.$$

In the lifting scheme version of the Haar transform, the update step replaces an even element with the average of even/odd pair. The original odd element has been replaced by the difference between this element and its even predecessor.

$$U^*_{2j+1} = U_{2j+1} - U_{2j} \Rightarrow U^*_{2j+1} + U_{2j} = U_{2j+1}$$

$$U_{2j} = \frac{U_{2j} + U_{2j+1}}{2} = U_{2j} + U_{2j+1}/2.$$

In Figure 4.3 and 4.4, we can see Haar transform in-place. The Haar transform will replace the values ($s_{2,0} = 9$, $s_{2,1} = 7$, $s_{2,2} = 3$, $s_{2,3} = 5$) by average ($s_{1,0} = \frac{9+7}{2} = 8$, $s_{1,1} = \frac{3+5}{2} = 4$) and difference($d_{1,0} = 7 - 9 = -2$, $d_{1,1} = 5 - 3 = 2$), then once again, replace the new average values by average ($s_{0,0} = \frac{8+4}{2} = 6$) and difference ($d_{0,0} = 4 - 8 = -4$).

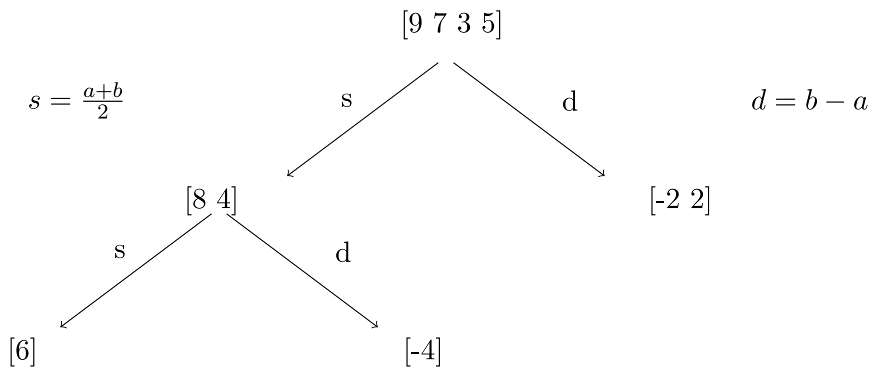


FIGURE 4.3: Example of Haar (Lifting)

$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	[9 7 3 5]
$s_{1,0}$	$d_{1,0}$	$s_{1,1}$	$d_{1,1}$	[8 -2 4 2]
$s_{0,0}$	$d_{1,0}$	$d_{0,0}$	$d_{1,1}$	[6 -2 -4 2]

FIGURE 4.4: Haar lifting ordering

4.2.2 Construction of Linear Interpolating Wavelets

Another example of a simple transform can be computed using a linear prediction step (see [18],[19]). The difference between the predicted value and the actual value of the odd element replaces the odd element with

$$U_{2j+1} = U_{2j+1} - \frac{1}{2}(U_{2j} + U_{2j+2}).$$

The linear interpolation update step replaces the even elements with an average of the data being processed. The update involves a combination of the two neighboring detail values computed in the previous step and stored in U_{2j+1} and U_{2j-1}

$$U_{2j} = U_{2j} + \frac{1}{4}(U_{2j+1} + U_{2j-1}).$$

This example is included due to its use in the development of Multiresolution analysis using numerical methods for hyperbolic partial differential equations. In fact, this transform is related to the central difference method for conservation laws.

4.3 Multiresolution Analysis (MRA): Definitions and Notation

Multiresolution analysis (MRA) was formulated based on the study of orthonormal, compactly supported wavelet bases. The following definition will be used in this work.

Definition 4.2. Let V_j, $j = \cdots, -2, -1, 0, 1, 2, \cdots$ be a sequence of subspaces of functions in $L^2(\mathbb{R})$. The collection of spaces $\{V_j, j \in \mathbb{Z}\}$ is called a **multiresolution analysis**

with scaling function φ if the following conditions hold. (see [1])

1. For all $j \in \mathbb{Z}, \quad V_j \subseteq V_{j+1}$
2. $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$. That is, the set $\bigcup_{j\in\mathbb{Z}} V_j$ is dense in $L^2(\mathbb{R})$.
3. $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$
4. The function $f(x)$ belongs to V_j if and only if the function $f(2^{-j}x)$ belongs to V_0.
5. The function φ belongs to V_0 and the set $\{\varphi(x-k), k \in \mathbb{Z}\}$ is an orthonormal basis for V_0.

Remark 4.3. The following well known results are stated for completeness and without proof.

- The V_j's are called **approximation spaces**.
- Every $f \in L^2$ can be approximated as closely as one likes by a function in a V_j, provided that j is large enough.
- $f \in \overline{\bigcup_{j\in\mathbb{Z}} V_j}$ if and only if for every $\epsilon > 0$ one can find j such that there is an $f_j \in V_j$ for which $\|f - f_j\| < \epsilon$
- We say that a sequence f_n of functions in $L^2([0,1))$ converges to a function $f(x) \in L^2([0,1))$, if
$$\lim_{n\to\infty} \|f_n - f\| = 0.$$

With this definition of a MRA, we need some definitions and notation for the scaling and translation concepts within any MRA.

Definition 4.4. A set of basis functions can be defined by translating and stretching (or compressing) a function $\varphi(x)$, called the scaling function. This is done through the definition of a doubly indexed set of functions defined by

$$\varphi_{j,k} = 2^{j/2}\varphi(2^j x - k)$$

for all integers j, k.

A simple example is provided by the function

$$\varphi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & elsewhere \end{cases}$$

This is called the Haar scaling function. Using the definitions above $\varphi(x)$ will general the Haar basis.

Definition 4.5. MRA Equation [1]
The set of scaling functions at any level j can be used to express functions that form the linear space V_j. This means

$$f(x) = \sum \alpha_k \varphi_{j,k}(x)$$

The representation of a function in V_j by an expansion in V_{j+1} is given by

$$\varphi_{j,k}(x) = \sum_n \alpha_n \varphi_{j+1,n}(x)$$

for any $f \in V_j$. Substituting the definition

$$\varphi_{j,n} = 2^{j/2}\varphi(2^j x - n)$$

and replacing the coefficients with the notation $h_\varphi(n) = \alpha_n$.

$$\varphi_{j,k}(x) = \sum_n h_\varphi(n) 2^{(j+1)/2} \varphi(2^{j+1}x - n)$$

By setting $(j,k) = (0,0)$ $\varphi(x) = \varphi_{0,0}(x)$ and

$$\varphi(x) = \sum_n h_\varphi(n)\sqrt{2}\varphi(2x - n)$$

This recursive equation is called the MRA equation.

In general, $V_{j+1} = V_j \oplus W_j$. The functions in W_j can be expanded in terms of a set of wavelets $\{\psi_{j,k}\}$ and the wavelets must be orthogonal to the scaling functions $\{\varphi_{j,k}\}$. Any function can be represented by a sequence of approximations which contain more

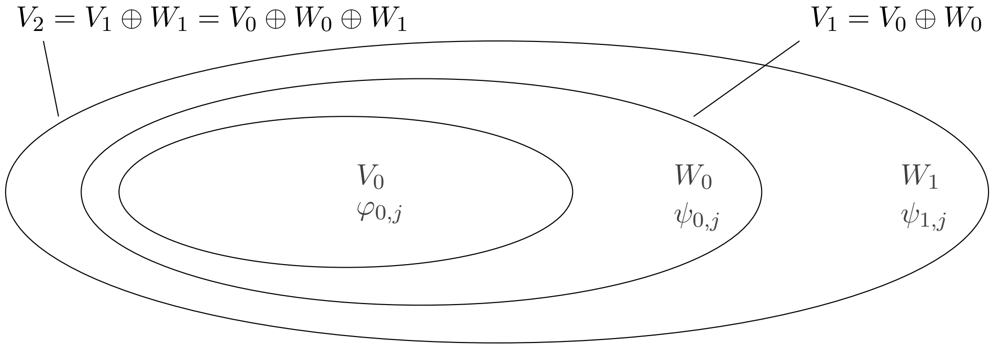


FIGURE 4.5: The relationship between scaling and wavelet function spaces [20]

and more detail (See Figure 4.5).

$$L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \cdots$$

Remark 4.6. The function $\varphi \in V$ is sometimes called the father wavelet for the MRA and the function $\psi \in W$ is sometimes called the mother wavelet for the MRA.

4.4 Polynomial Framework Lifting Approach

We can develop an algorithm for creating wavelets and associated MRAs using a combination of lifting with conservative numerical schemes for conservation laws. Given a discrete point source, $\delta(j)$, defined by

$$\delta(j) = \begin{cases} 1, & \text{if } j = 0 \\ 0, & \text{if } j \neq 0 \end{cases}$$

we can compute values at the half-integers or midpoints between integers using interpolation formulas or using grid-line values predicted by a numerical method for conservation laws. The steps to do this are to assume

1. $U_j^0 = \delta(j), \qquad j = \cdots, -2, -1, 0, 1, 2, \cdots$
2. $U_{j+1/2} = \sum_{i=0}^{l} a_{i,j+1/2}\alpha_{j+1/2}^{i}, \qquad j = \cdots, -2, -1, 0, 1, 2, \cdots$

Any numerical scheme that can be written in the polynomial framework for conservation laws can be used along with lifting to generate a future wavelet and thus an associated MRA.

4.4.1 Upwind Scheme

As an example a MRA can be created using the upwind scheme to compute the values at the half integers in the lifting algorithm. From the section 3.2.2,

$$U^n_{j+\frac{1}{2}} = m^n_{j+\frac{1}{2}}(\alpha^n_{j+\frac{1}{2}}) = U^n_{j+1} + (U^n_j - U^n_{j+1})\alpha^n_{j+\frac{1}{2}}$$

$$U^n_{j-\frac{1}{2}} = m^n_{j-\frac{1}{2}}(\alpha^n_{j-\frac{1}{2}}) = U^n_j + (U^n_{j-1} - U^n_j)\alpha^n_{j-\frac{1}{2}},$$

and we know that for upwinding, $\alpha^n_{j\pm\frac{1}{2}} = 1$. Then we have

$$U^n_{j+\frac{1}{2}} = U^n_j,$$

$$U^n_{j-\frac{1}{2}} = U^n_{j-1}$$

where we have assumed $f'(U) > 0$.
Recall that the general form of $\alpha^n_{j\pm\frac{1}{2}}$ is

$$\alpha^n_{j\pm\frac{1}{2}} = \gamma_1 + \gamma_2 \frac{\Delta t}{\Delta x} f'(U^n_{j\pm\frac{1}{2}}).$$

For upwinding we fix $\alpha^n_{j\pm\frac{1}{2}} = 1$ and if we assume $f' > 0$ then the steps of the lifting result in the following.

$$U_j = \begin{cases} 1, & \text{if } j = 0 \\ 0, & \text{if } j \neq 0 \end{cases}$$

The midpoints using the grid-line values predicted by upwind method are

$$U_{-\frac{1}{2}} = U_{-1} = 0,$$

$$U_{\frac{1}{2}} = U_0 = 1.$$

In the same way, we can find

$$U_{-\frac{1}{4}} = 0,$$

$$U_{\frac{1}{4}} = 1.$$

Figure 4.6, shows the resulting upwind shape function with only the discrete points from the lifting plotted. There are 17,129, 4097 points plotted. From top to bottom the graphs in Figure 4.6 use 17,129, and 4097 points. This is done using $n = 4$, $n = 7$, and $n = 12$ recursive steps in the algorithm.

4.4.2 Lax-Wendroff Scheme

Another example of a MRA can be defined using the Lax-Wendroff method. To compute the values at the half integers in the lifting algorithm we can use the polynomial framework. From Section 3.2.2, the Lax-Wendroff scheme has $\alpha^n_{j+\frac{1}{2}} = \frac{1}{2} + \frac{1}{2}a\frac{\Delta t}{\Delta x}$. Then

$$\begin{aligned} U^n_{j+\frac{1}{2}} &= U^n_{j+1} + (U^n_j - U^n_{j+1})\alpha^n_{j+\frac{1}{2}} \\ &= U^n_{j+1} + (U^n_j - U^n_{j+1})(\frac{1}{2} + \frac{1}{2}a\frac{\Delta t}{\Delta x}) \\ &= \frac{1}{2}(U^n_{j+1} + U^n_j) + \frac{\Delta t}{2\Delta x}a(U^n_j - U^n_{j+1}), \end{aligned}$$

$$U^n_{j-\frac{1}{2}} = \frac{1}{2}(U^n_j + U^n_{j-1}) + \frac{\Delta t}{2\Delta x}a(U^n_{j-1} - U^n_j).$$

Therefore,

$$U^n_{j+\frac{1}{2}} = \underbrace{\frac{1}{2}(U^n_{j+1} + U^n_j)}_{\text{average of two values}} + \frac{\Delta t}{2\Delta x}a \underbrace{(U^n_j - U^n_{j+1})}_{\text{difference of two values}} .$$

$$U^n_{j-\frac{1}{2}} = \underbrace{\frac{1}{2}(U^n_j + U^n_{j-1})}_{\text{average of two values}} + \frac{\Delta t}{2\Delta x}a \underbrace{(U^n_{j-1} - U^n_j)}_{\text{difference of two values}} .$$

We know that the Courant-Fridrichs-Lewy (CFL) condition is $0 < \frac{a\Delta t}{\Delta x} < 1$, so the midpoints using the grid-line values predicted by Lax-Wendroff method are

$$U_{-\frac{1}{2}} = \frac{1}{2} + \frac{\Delta t}{2\Delta x}a(-1) < \frac{1}{2}$$

$$U_{\frac{1}{2}} = \frac{1}{2} + \frac{\Delta t}{2\Delta x}a(1) > \frac{1}{2}$$

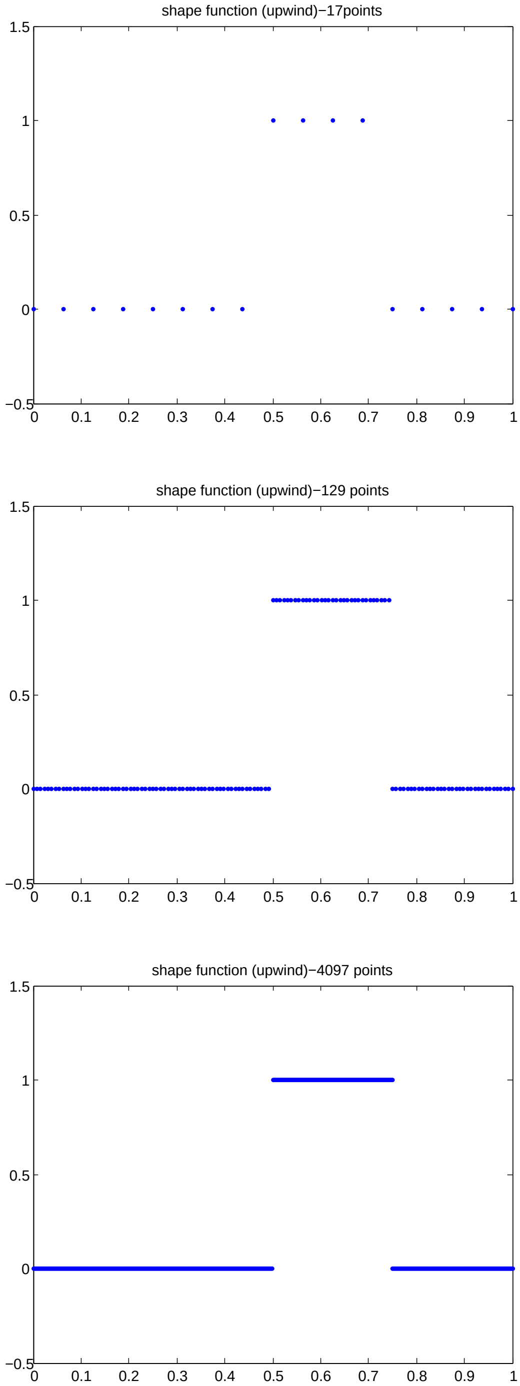


FIGURE 4.6: Upwind scheme with lifting framework

Figure 4.7, shows the result of applying the lifting algorithm to the Lax-Wendroff shape function. From top to bottom the graphs in Figure 4.7 use 17,129, and 4097 points. This is done using $n = 4$, $n = 7$, and $n = 12$ recursive steps in the algorithm. We use the CFL $\frac{\Delta t}{\Delta x} a = 0.5$.

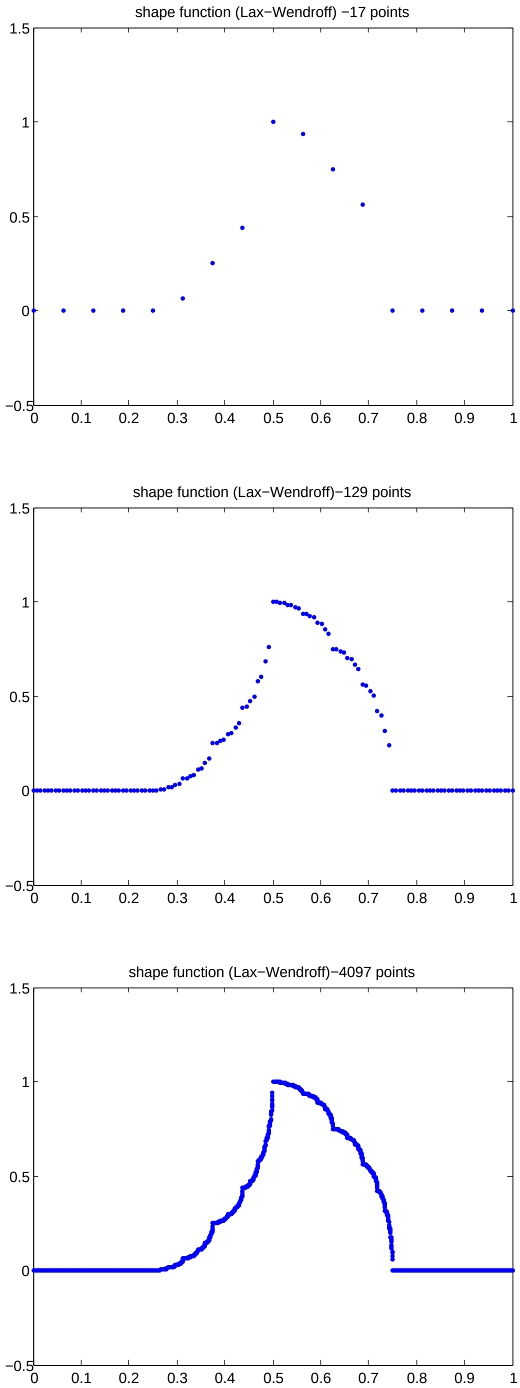


FIGURE 4.7: Lax-Wendroff scheme with lifting framework

4.4.3 Beam-Warming Scheme

Another example of a MRA can be defined using the Beam-Warming method. To compute the values at the half integers in the lifting algorithm. We can use (See Section 3.2.2) $\alpha^n_{j+\frac{1}{2}} = \frac{1}{2} - \frac{1}{2}a\frac{\Delta t}{\Delta x}$. Then

$$\begin{aligned} U^n_{j+\frac{1}{2}} &= U^n_{j+1} + (U^n_j - U^n_{j+1})\alpha^n_{j+\frac{1}{2}} \\ &= U^n_{j+1} + (U^n_j - U^n_{j+1})(\frac{1}{2} - \frac{1}{2}a\frac{\Delta t}{\Delta x}) \\ &= \frac{1}{2}(U^n_{j+1} + U^n_j) - \frac{\Delta t}{2\Delta x}a(U^n_j - U^n_{j+1}), \end{aligned}$$

$$U^n_{j-\frac{1}{2}} = \frac{1}{2}(U^n_j + U^n_{j-1}) - \frac{\Delta t}{2\Delta x}a(U^n_{j-1} - U^n_j).$$

Therefore,

$$U^n_{j+\frac{1}{2}} = \underbrace{\frac{1}{2}(U^n_{j+1} + U^n_j)}_{\text{average of two values}} - \frac{\Delta t}{2\Delta x}a \underbrace{(U^n_j - U^n_{j+1})}_{\text{difference of two values}},$$

$$U^n_{j-\frac{1}{2}} = \underbrace{\frac{1}{2}(U^n_j + U^n_{j-1})}_{\text{average of two values}} - \frac{\Delta t}{2\Delta x}a \underbrace{(U^n_{j-1} - U^n_j)}_{\text{difference of two values}}.$$

The midpoints using the grid-line values predicted by Beam-Warming method are

$$U_{-\frac{1}{2}} = \frac{1}{2} + \frac{\Delta t}{2\Delta x}a(1) > \frac{1}{2}$$

$$U_{\frac{1}{2}} = \frac{1}{2} + \frac{\Delta t}{2\Delta x}a(-1) < \frac{1}{2}$$

Figure 4.8, shows the result of applying the lifting algorithm to the Beam-Warming shape function. From top to bottom the graphs in Figure 4.8 use 17,129, and 4097 points. This is done using $n = 4$, $n = 7$, and $n = 12$ recursive steps in the algorithm. We use the CFL $\frac{\Delta t}{\Delta x}a = 0.5$.

4.4.4 TVD Scheme

In Section 2.2.3.1, the numerical flux for a TVD scheme was given by

$$f(u^n_{j+\frac{1}{2}}) = aU_j + C_{j+\frac{1}{2}}\left(\tfrac{1}{2}a(1 - a\tfrac{\Delta t}{\Delta x})(U_{j+1} - U_j)\right)$$

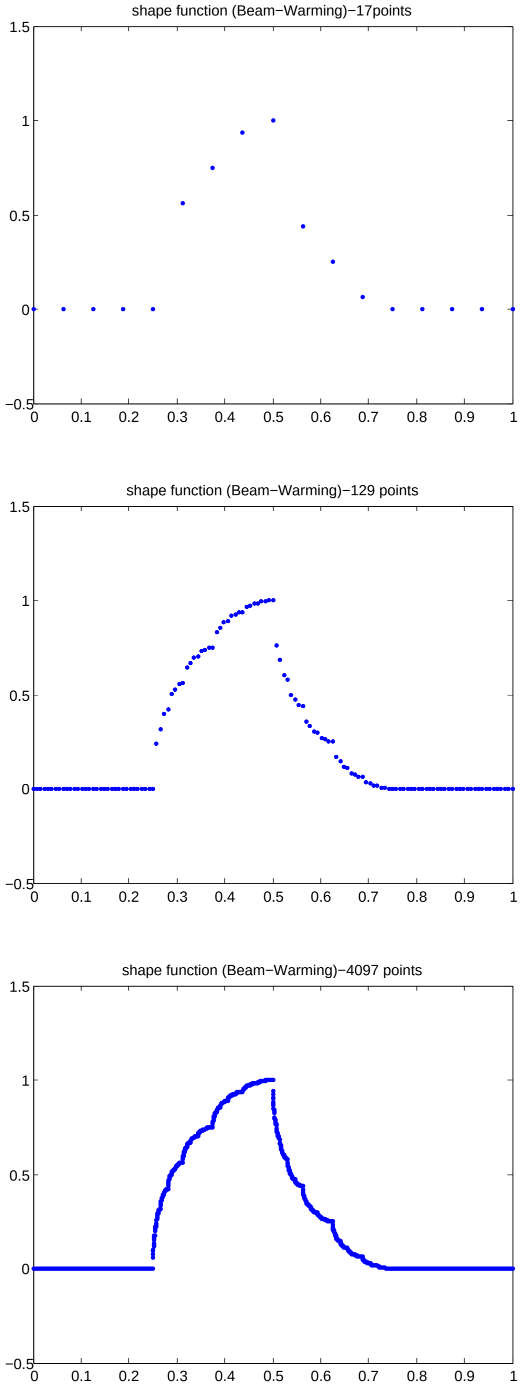


FIGURE 4.8: Beam-Warming scheme with lifting framework

where $C_{j+\frac{1}{2}} := C(\theta_{j+\frac{1}{2}})$ is a flux-limiter that depends on $\theta_{j+\frac{1}{2}} := \frac{U_{j-1}-U_j}{U_j-U_{j+1}}$. Below we will consider one example of a flux limited TVD scheme called the Super-Bee limiter. In the linear advection problem the flux-limited methods by presented in Sweby [16], we choose (see [4])

$$\alpha^n_{j+\frac{1}{2}} = \frac{1}{2}(1 + a\lambda) + \frac{1}{2}(1 - a\lambda)(1 - C^n_{j+\frac{1}{2}})$$

where $\lambda = \frac{\Delta t}{\Delta x}$ and $C^n_{j+\frac{1}{2}}$ is a flux limiter as defined in Sweby [16].

We can compute the sign of the ratio of $(U_{j-1} - U_j)$ and $(U_j - U_{j+1})$. If the sign is a negative number, then $C_{j+\frac{1}{2}}$ should be zero. If the sign is a positive number and $(U_j - U_{j+1}) = 0$, then $C_{j+\frac{1}{2}}$ should be 2. By a "Super-Bee" flux limiter [16], the rest of the cases are

$$C(\theta^n_{j+\frac{1}{2}}) = \begin{cases} 0, & \text{if } \theta_{j+\frac{1}{2}} \leq 0 \\ 2\theta_{j+\frac{1}{2}}, & \text{if } 0 < \theta_{j+\frac{1}{2}} < \frac{1}{2} \\ 1, & \text{if } \frac{1}{2} \leq \theta_{j+\frac{1}{2}} < 1 \\ \theta_{j+\frac{1}{2}}, & \text{if } 1 \leq \theta_{j+\frac{1}{2}} < 2 \\ 2, & \text{if } \theta_{j+\frac{1}{2}} \geq 2 \end{cases}$$

If $C(\theta^n_{j+\frac{1}{2}}) = 0$, then

$$\alpha^n_{j+\frac{1}{2}} = \frac{1}{2}(1 + a\lambda) + \frac{1}{2}(1 - a\lambda)(1 - 0) = \frac{1}{2}(1 + a\lambda) + \frac{1}{2}(1 - a\lambda) = 1,$$

so

$$U^n_{j+1/2} = U^n_{j+1} + (U^n_j - U^n_{j+1})\alpha^n_{j+1/2} = U^n_{j+1} + (U^n_j - U^n_{j+1}) = U^n_j,$$

$$U^n_{j-1/2} = U^n_j + (U^n_{j-1} - U^n_j)\alpha^n_{j-1/2} = U^n_j + (U^n_{j-1} - U^n_j) = U^n_{j-1}.$$

Note that this gives the upwind value if $C(\theta^n_{j+\frac{1}{2}}) = 1$, then

$$\alpha^n_{j+\frac{1}{2}} = \frac{1}{2}(1 + a\lambda) + \frac{1}{2}(1 - a\lambda)(1 - 1) = \frac{1}{2}(1 + a\lambda).$$

This is the same as the value returned by the Lax-Wendroff if $C(\theta^n_{j+\frac{1}{2}}) = 2$, then

$$\alpha^n_{j+\frac{1}{2}} = \frac{1}{2}(1 + a\lambda) + \frac{1}{2}(1 - a\lambda)(1 - 2) = \frac{1}{2}(1 + a\lambda) - \frac{1}{2}(1 - a\lambda) = a\lambda$$

Figure 4.9, 4.10, and 4.11 show the result of applying the lifting algorithm to define the TVD shape function. From top to bottom the graphs in Figure 4.9, 4.10, and 4.11 use 17,129, and 4097 points. This is done using $n = 4$, $n = 7$, and $n = 12$ recursive steps in the algorithm. Figures 4.9, 4.10, and 4.11 show shape function where we use different values for $\frac{\Delta t}{\Delta x} a = 0.1, 0.5, 0.9$.

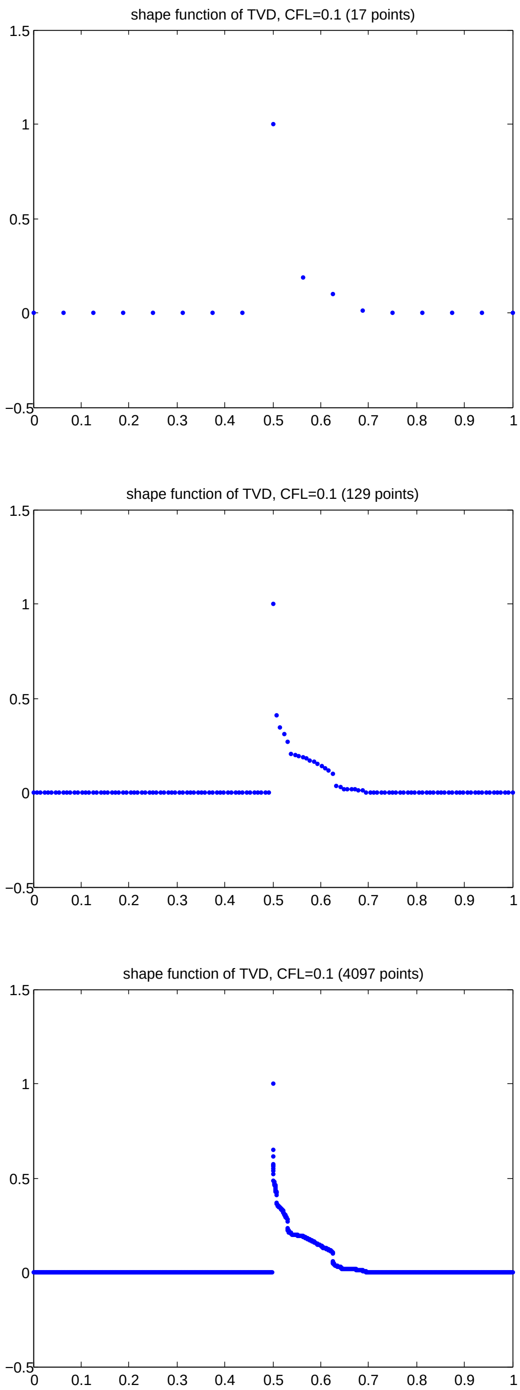


FIGURE 4.9: TVD scheme with lifting framework, CFL=0.1

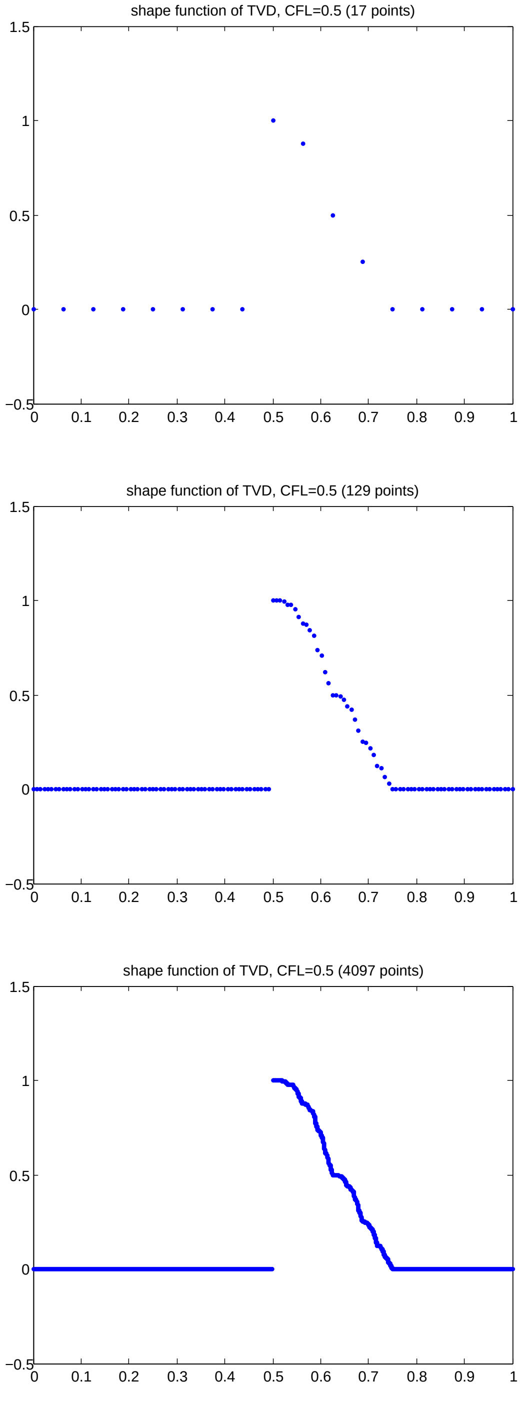


FIGURE 4.10: TVD scheme with lifting framework, CFL=0.5

4.4.5 Lax-Friedrichs Scheme

The general numerical flux for the Lax-Friedrichs scheme takes the form

$$U_j^{n+1} = \frac{1}{2}\left(U_{j-1}^n + U_{j+1}^n\right) - \frac{\Delta t}{2\Delta x}\left(f(U_{j+1}^n) - f(U_{j-1}^n)\right).$$

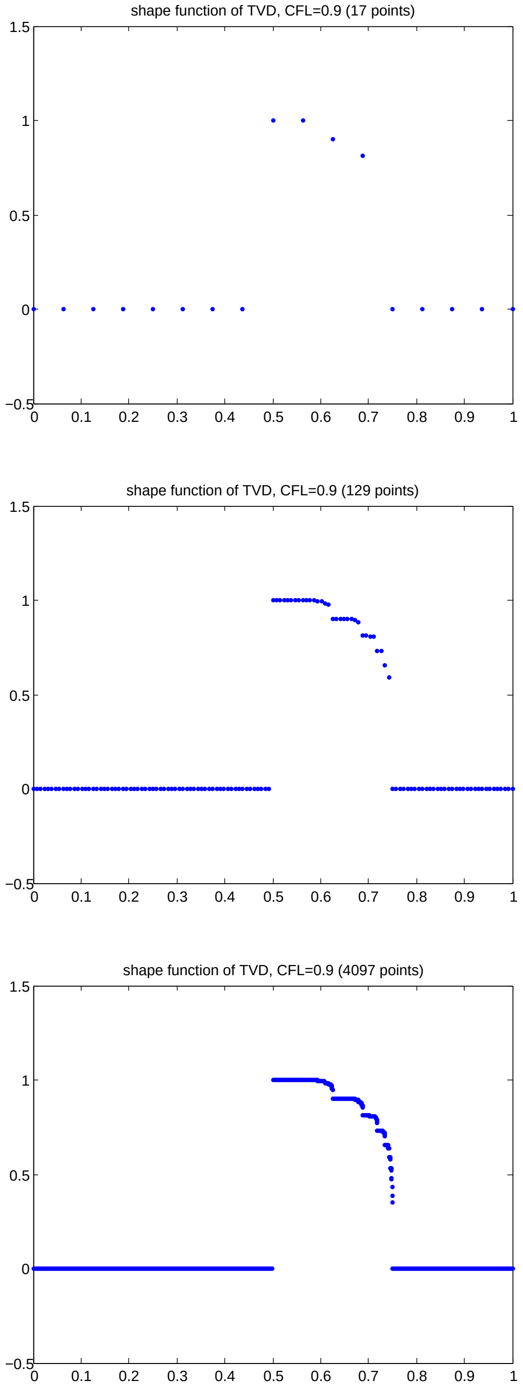


FIGURE 4.11: TVD scheme with lifting framework, CFL=0.9

This method can be written in a conservation form (3.1) by taking

$$f(U^n_{j+\frac{1}{2}}) = \frac{\Delta x}{2\Delta t}\left(U^n_{j+1} - U^n_j\right) + \frac{1}{2}\left(f(U^n_j) + f(U^n_{j+1})\right).$$

If the flux function is linear and $f(u) = au$, then

$$\begin{aligned}
U^n_{j+\frac{1}{2}} &= \frac{1}{2}\left(f(U^n_j) + f(U^n_{j+1})\right) + \frac{\Delta x}{2a\Delta t}\left(U^n_{j+1} - U^n_j\right) \\
&= \frac{1}{2}\frac{\Delta x}{a\Delta t}\left(\frac{a\Delta t}{\Delta x}(U^n_{j+1} + U^n_j) + (U^n_{j+1} - U^n_j)\right) \\
&= \frac{\Delta x}{a\Delta t}\left(\frac{1}{2}\left(1 + \frac{a\Delta t}{\Delta x}\right)U^n_{j+1} + \frac{1}{2}\left(1 - \frac{a\Delta t}{\Delta x}\right)U^n_j\right)
\end{aligned}$$

So,

$$U^n_{j+\frac{1}{2}} = U^n_{j+1} + \left(\frac{1}{2} + \frac{\Delta x}{2a\Delta t}\right)(U^n_j - U^n_{j+1})$$

$$U^n_{j-\frac{1}{2}} = U^n_j + \left(\frac{1}{2} + \frac{\Delta x}{2a\Delta t}\right)(U^n_{j-1} - U^n_j)$$

Figure 4.12, shows the result of applying the lifting algorithm to the Lax-Friedrichs shape function. From top to bottom the graphs in Figure 4.12 use 17,129, and 4097 points. This is done using $n = 4$, $n = 7$, and $n = 12$ recursive steps in the algorithm.

Remark 4.7. This example has been included due to the use of the Lax-Friedrichs method in the proof of existence and uniqueness given by Oleinik [13].

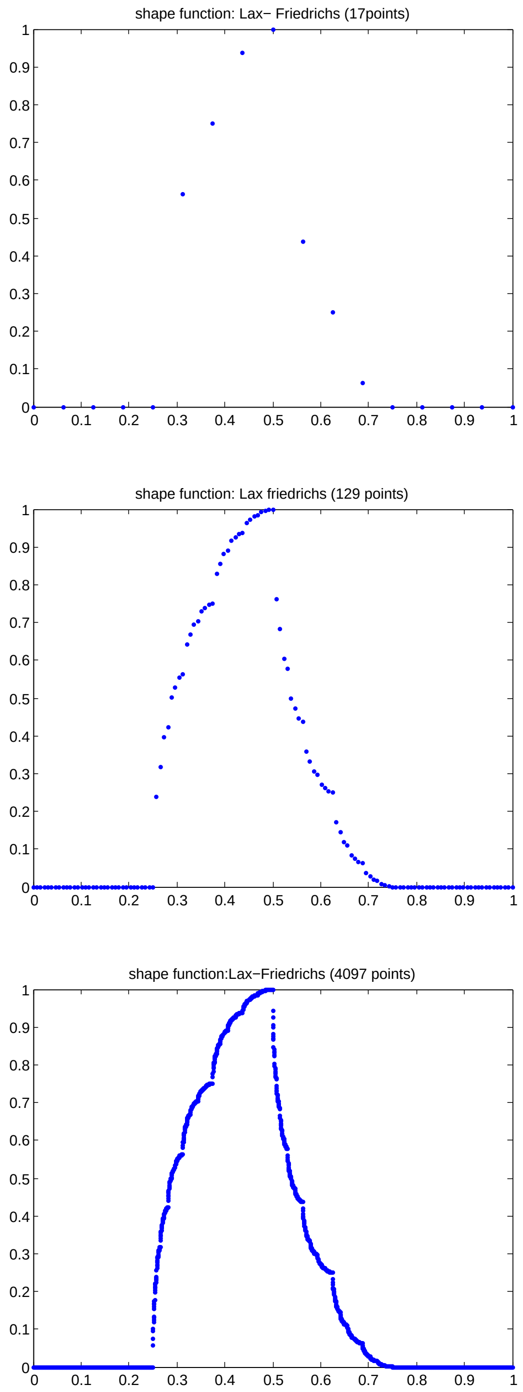


FIGURE 4.12: Lax-Friedrichs scheme with lifting framework

CHAPTER 5
CONSTRUCTION OF WAVELETS BASES CONDITIONED ON HYPERBOLIC CONSERVATION LAWS

In Section 4.4, we generated shape functions conditioned on various approximation schemes conservation law via the lifting algorithm of Sweldens. Now we build the wavelet functions from the shape functions. Through Figure 5.1, we learn how to construct the wavelet shape function based on the central difference method.

First, we set our shape function (see Figure 5.1 (a)) in the interval $[0, 1]$. Then we compress two shape functions (see Figure 5.1 (b)) in the same interval $[0, 1]$. Finally, the functions shown in Figure 5.1 (b) are subtracted from the function shown in Figure 5.1 (a). The result is shown in Figure 5.1(c). The function graphed in Figure 5.1 (c) is our wavelet function.
If the wavelet function is $f(x)$, then

$$\int_0^1 f(x)\, dx = 0.$$

Note that the shape function in Figure 5.1 (a) is supported on the unit interval $[0, 1]$. Figure 5.1 (b) shows two scaled shape functions, one supported on the interval $[0, \frac{1}{2}]$ and the other supported on $[\frac{1}{2}, 1]$. Finally, the function graphed in Figure 5.1 (c) is supported on the unit interval $[0, 1]$.

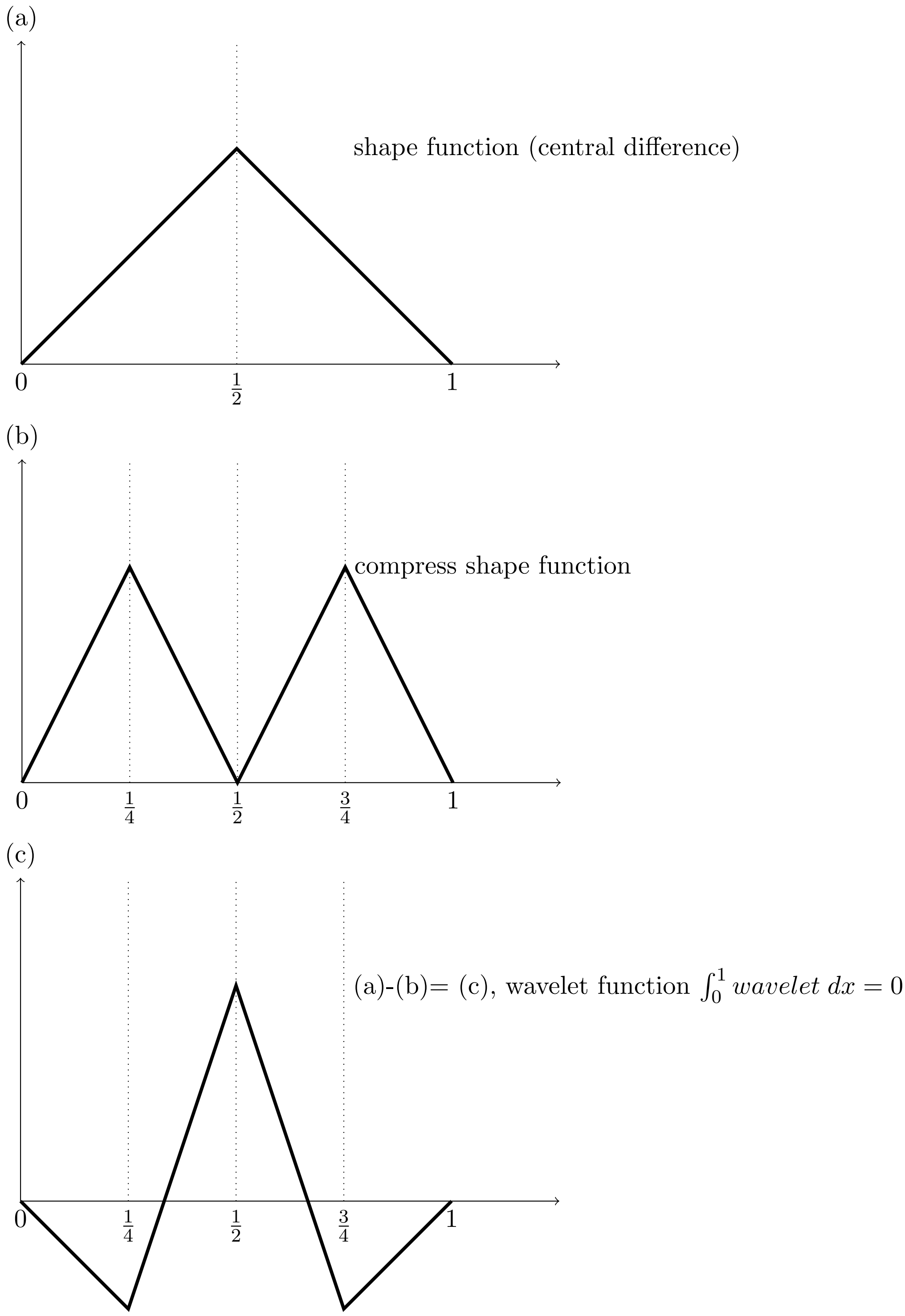


FIGURE 5.1: Build the wavelet function from shape function

5.1 Upwind Scheme Example

Figure 5.2 shows how the wavelets associated with the Upwind scheme are created.

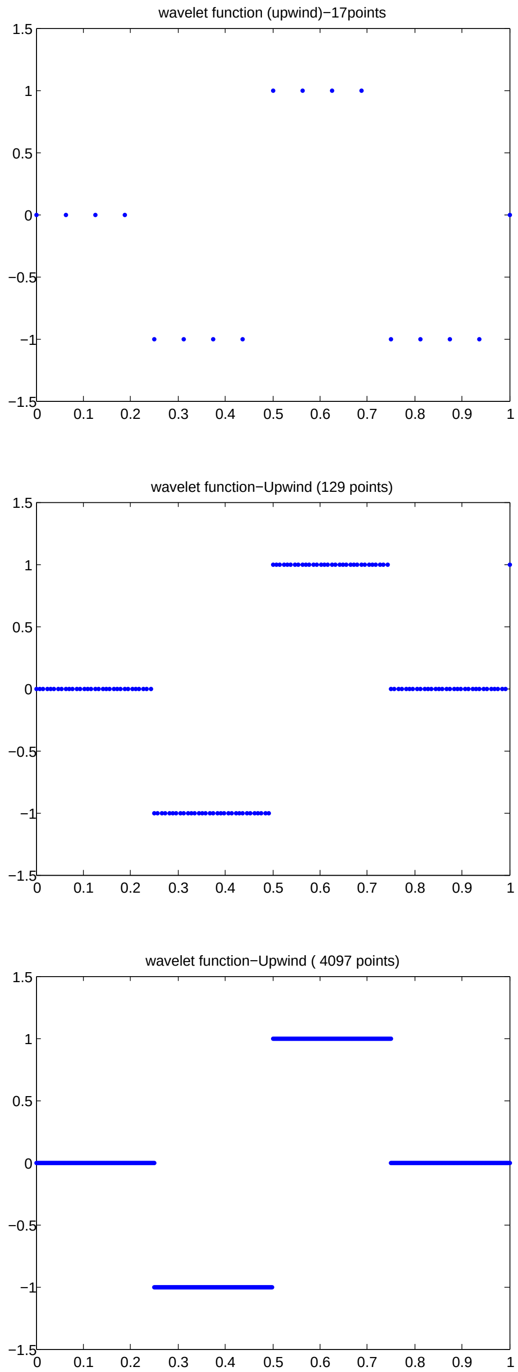


FIGURE 5.2: The wavelet function for the Upwind scheme, CFL=0.5

5.2 Lax-Wendroff Scheme Example

Figure 5.3 shows the construction of wavelets based on the Lax-Wendroff method for approximate solution of scalar conservation laws.

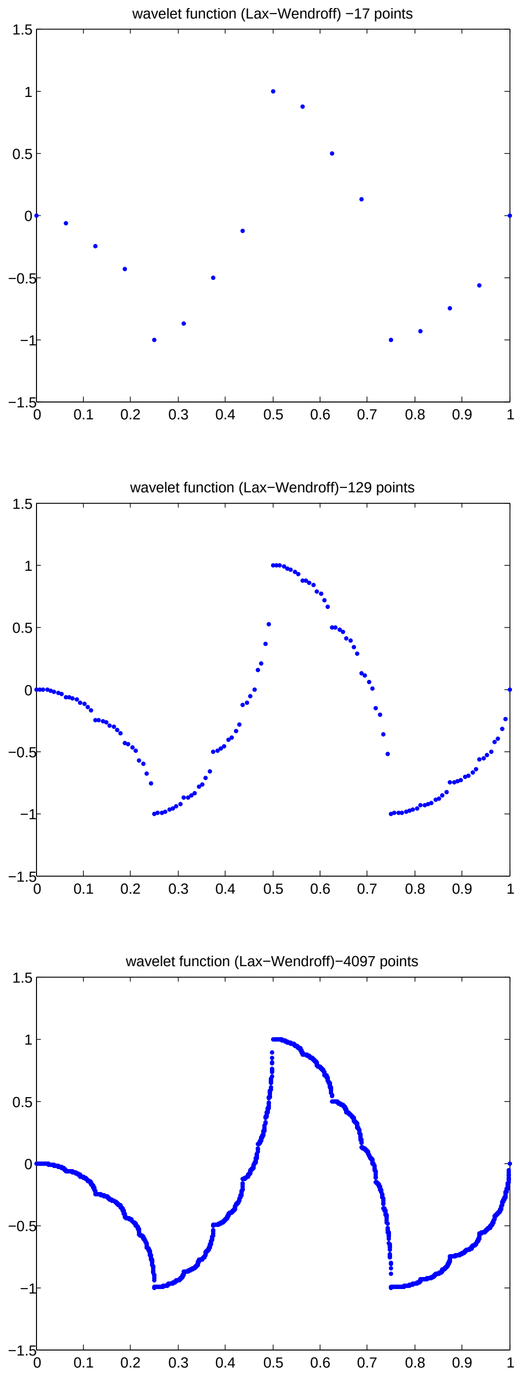


FIGURE 5.3: The wavelet function for the Lax-Wendroff scheme, CFL=0.5

5.3 Beam-Warming Scheme Example

Figure 5.4 shows the construction of wavelets based on the Beam-Warming method for approximate solution of scalar conservation laws.

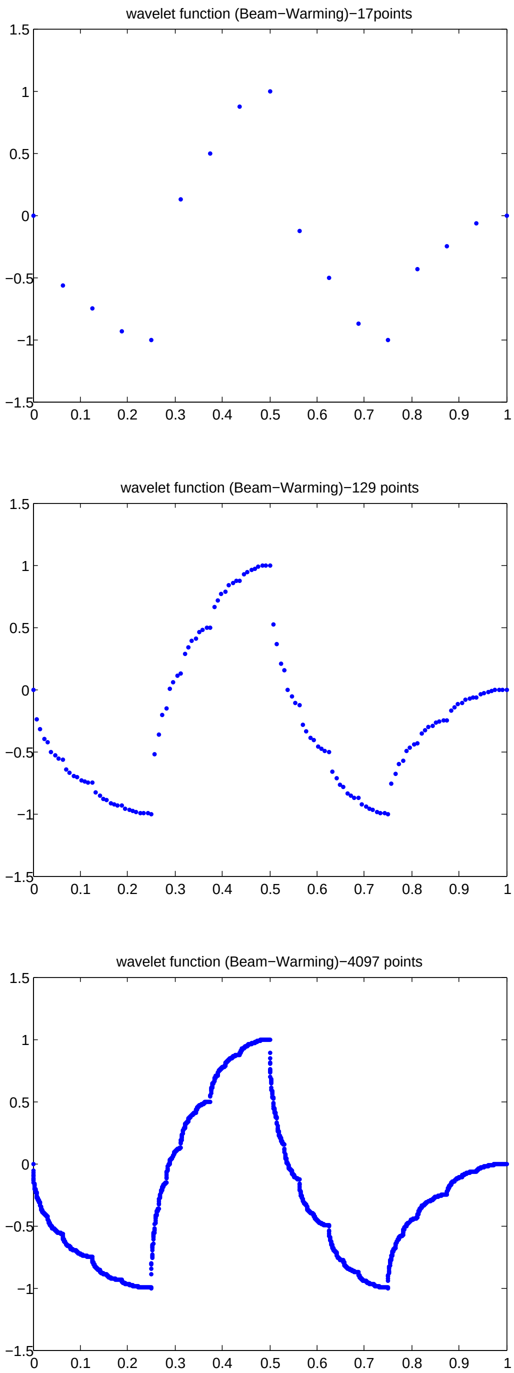


FIGURE 5.4: The wavelet for the Beam-Warming scheme, CFL=0.5

5.4 TVD Scheme Example

Figure 5.5 shows the construction of wavelets based on the TVD method for approximate solution of scalar conservation laws.

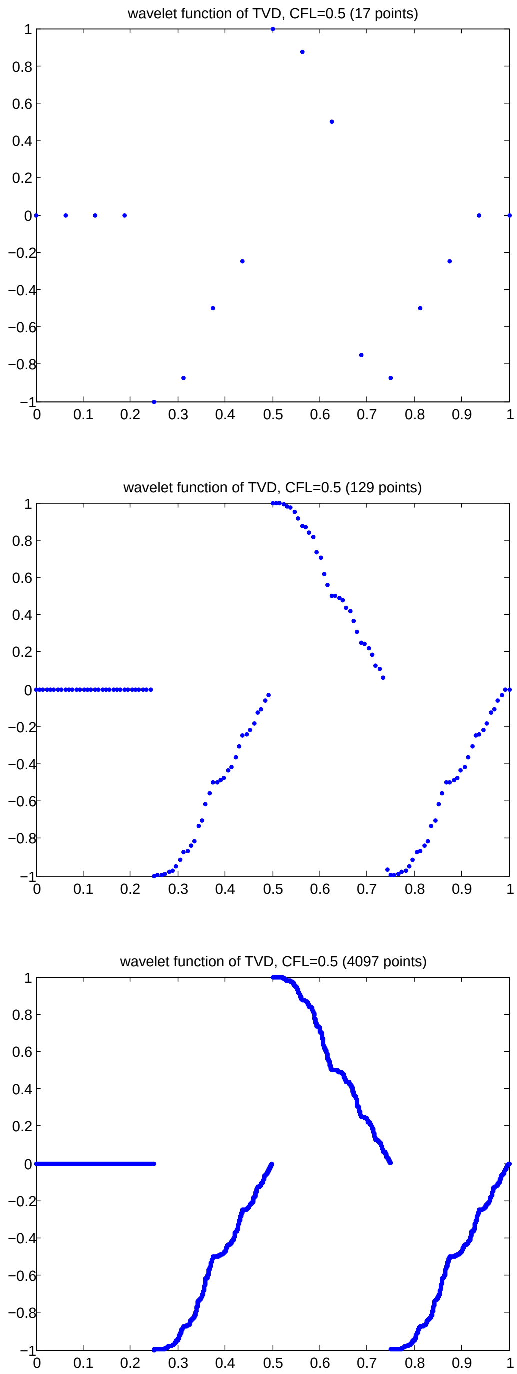


FIGURE 5.5: The wavelet function for the TVD scheme, CFL=0.5

5.5 Lax-Friedrichs Scheme

Figure 5.6 shows the construction of wavelets based on the Lax-Friedrichs method for approximate solution of scalar conservation laws.

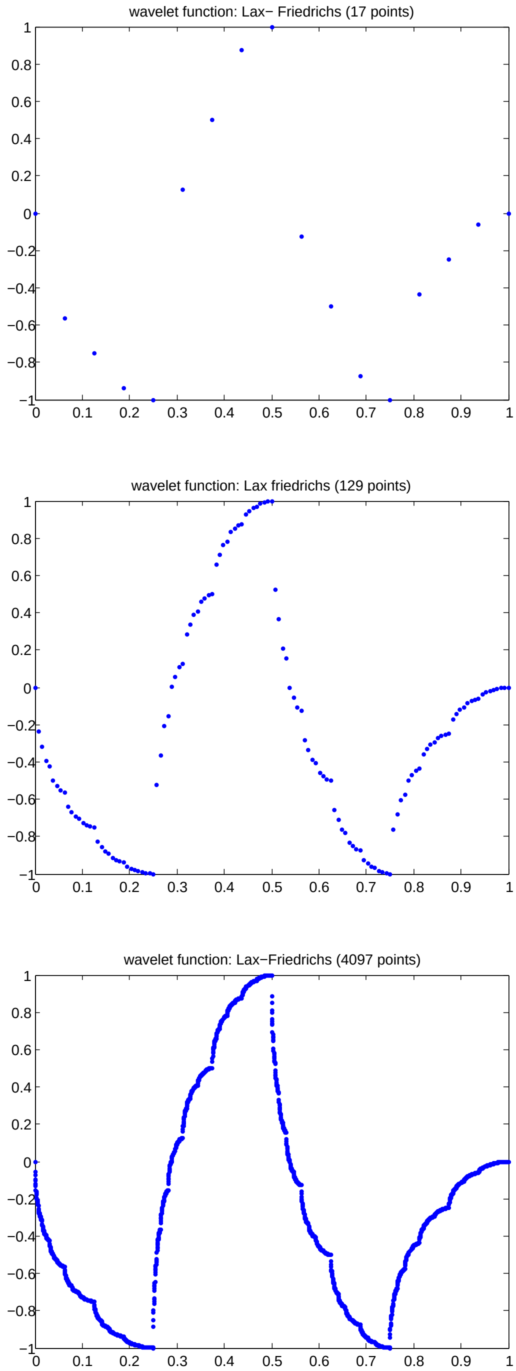


FIGURE 5.6: The wavelet function for the Lax-Friedrichs, CFL=0.5

5.6 How to use the MRA in Applications

Once the shape and wavelet functions are known, how can they be of use in applications. In the next chapter, we will introduce numerical methods for conservation laws based on the representations provided by the MRAs built in this and previous chapters.

CHAPTER 6
CONSTRUCTION OF APPROXIMATE SOLUTIONS OF CONSERVATION LAWS VIA A FINITE VOLUME APPROACH

In this chapter, we will build an approximate solution method for conservation laws from the work in previous chapters. Also, we will give an example of our approximation method for conservation laws. The example will use the finite volume method (FVM) approach along with a high order quadrature rule-Simpson's Rule.

6.1 Building a Representation of Solution

In previous chapters, we created shape functions and wavelet functions condition on numerical methods for conservation laws. In the next step, we build an approximate solution of conservation laws.

Suppose we are given 2^m samples ($u^n = \underbrace{\{u_0, u_1, u_2, \cdots, u_{N-1}\}}_{2^m}$, $N = 2^m$) that represent the solution of a conservation law at some time t_n. That is,

$$u(x, t_n) = u^n(x) = \alpha_0^n \varphi_0(x) + \sum_{l=0}^{m} \sum_{k=0}^{N-1} \beta_{l,k}^n \psi_{l,k}(x) \tag{6.1}$$

$$\alpha_0^n = \langle u^n(x), \varphi_0(x) \rangle \tag{6.2}$$

$$\beta_{l,k}^n = \langle u^n(x), \psi_{l,k}(x) \rangle \tag{6.3}$$

where $\varphi_0(x)$ is the shape function and $\psi_{l,k}(x)$ is the wavelet function. Note that the shape and wavelet functions are defined at discrete points. This means the inner products in (6.2) and (6.3) are discrete sums.

As shown in Figure 6.1 we can construct basis functions in our MRA for the representation given in Equations (6.1), (6.2), and (6.3) via lifting. Figure 6.1 shows one example of the process.

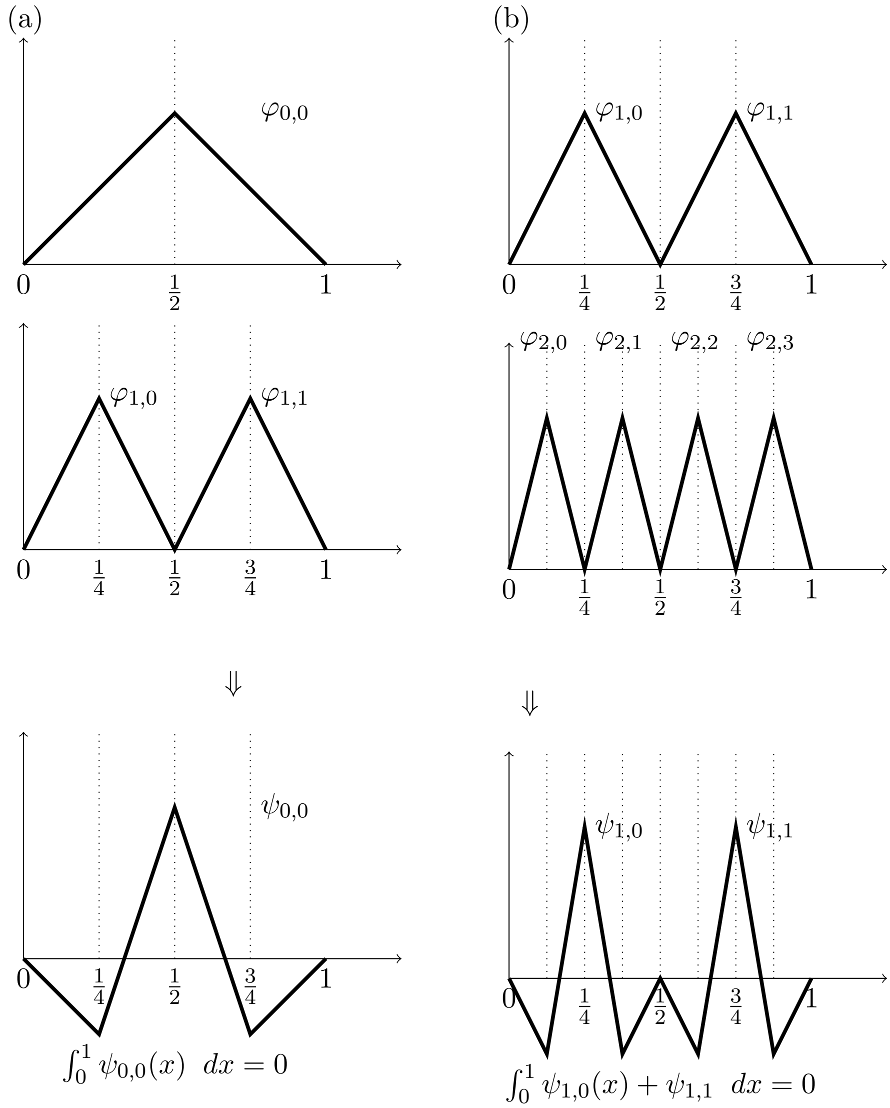


FIGURE 6.1: Relationship of the shape function and the wavelet function for conservation law

6.2 An Example: Finite Volume Approach and Simpson's Rule

First, we compute our approximate solution $u(x, t_n)$. When $n = 0$,

$$u^0(x) = \alpha_0^0 \varphi_0(x) + \sum_{l=0}^{m} \sum_{k=0}^{2^m-1} \beta_{l,k}^0 \psi_{l,k}(x),$$

$$\alpha_0^0 = \langle u^0(x), \varphi_0(x) \rangle$$

$$\beta_{l,k}^0 = \langle u^0(x), \psi_{l,k}(x) \rangle$$

When $n = 1, 2, \cdots,$ we use th finite volume method (FVM)[10]. The finite volume method is commonly used to discretize conservation laws. The following steps provide a very brief review of this technique. Starting with the conservation law

$$u_t + f(u)_x = 0,$$

we integrate over a finite volume in the x-t plane. Given the rectangle defined by $[t_n, t_{n+1}] \times [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, we integrate as follows

$$\int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} (u_t + f(u)_x) \; dx \; dt = 0$$

This can be rewritten as

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} (u(x, t_{n+1}) - u(x, t_n)) \; dx + \int_{t_n}^{t_{n+1}} \Big(f(u(x_{j+\frac{1}{2}}, t)) - f(u(x_{j-\frac{1}{2}}, t)) \Big) \; dt = 0$$

Let

$$U_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_n) \; dx.$$

So,

$$\Big(U_j^{n+1} - U_j^n \Big) \cdot \Delta x + \int_{t_n}^{t_{n+1}} \Big(f(u(x_{j+\frac{1}{2}}, t)) - f(u(x_{j-\frac{1}{2}}, t)) \Big) \; dt = 0.$$

Next, let

$$F_{j\pm\frac{1}{2}}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(u(x_{j\pm\frac{1}{2}}, t)) \; dt.$$

The end result is the following

$$\Big(U_j^{n+1} - U_j^n \Big) \cdot \Delta x + \Big(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \Big) \cdot \Delta t = 0$$

or

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} \Big(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \Big).$$

Remark 6.1. This form has been used throughout this work. The computations to get to this point in the FVM approach are illustrated in Figure 6.2.

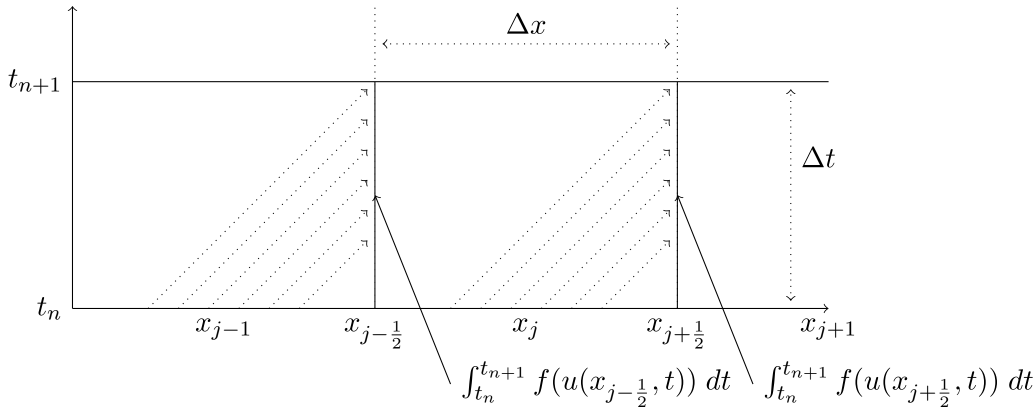


FIGURE 6.2: Finite Volume Method

Next, we have to compute approximations for $F^n_{j\pm\frac{1}{2}} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(u(x_{j\pm\frac{1}{2}}, t))\ dt$. This integral can be approximated by a quadrature method. The quadrature method provides a numerical approximation of the value of a definite integral $\int_a^b f(x)\ dx$. In this example, we use Simpson's rule. It is one quadrature method of many that could be used. Simpson's rule is

$$\int_a^b f(x)\ dx \approx \frac{h}{3}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right],$$

where $h = \frac{b-a}{2}$. Now we need to determine values for $f(a)$, $f(\frac{(}{a} + b2)$ and $f(b)$ within the context of the MRAs being used.

6.2.1 Linear Conservation Law Case

If $f(u) = au$, then an approximation of Simpson's rule results in the following,

$$\begin{aligned} F^n_{j\pm\frac{1}{2}} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(u(x_{j\pm\frac{1}{2}}, \tau))\ d\tau \\ &\approx \frac{1}{\Delta t}\left(\frac{\Delta t}{6}a\left[u(x_{j\pm\frac{1}{2}}, t_n) + 4u(x_{j\pm\frac{1}{2}}, t_{n+\frac{1}{2}}) + u(x_{j\pm\frac{1}{2}}, t_{n+1})\right]\right) \\ &= \frac{1}{6}a(U^n_{j\pm\frac{1}{2}} + 4U^{n+\frac{1}{2}}_{j\pm\frac{1}{2}} + U^{n+1}_{j\pm\frac{1}{2}}) \end{aligned}$$

where $U^n_{j\pm\frac{1}{2}} = u(x_{j\pm\frac{1}{2}}, t_n)$, $U^{n+\frac{1}{2}}_{j\pm\frac{1}{2}} = u(x_{j\pm\frac{1}{2}}, t_{n+\frac{1}{2}})$, and $U^{n+1}_{j\pm\frac{1}{2}} = u(x_{j\pm\frac{1}{2}}, t_{n+1})$. By an application of the method of characteristics, we can compute $U^n_{j\pm\frac{1}{2}}$,$U^{n+\frac{1}{2}}_{j\pm\frac{1}{2}}$ and $U^{n+1}_{j\pm\frac{1}{2}}$ for our approximate solution, as follows

$$\begin{aligned}
U_{j\pm\frac{1}{2}}^{n+\frac{1}{2}} &= u(x_{j\pm\frac{1}{2}}, t_{n+\frac{1}{2}}) \\
&= u(x_{j\pm\frac{1}{2}} - a(t_{n+\frac{1}{2}} - t_n), t_n) \\
&= u(x_{j\pm\frac{1}{2}} - a\frac{\Delta t}{2}, t_n) \\
&= \alpha_0^n \varphi_0(x_{j\pm\frac{1}{2}} - a\frac{\Delta t}{2}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1} \beta_{l,k}^n \psi_{l,k}(x_{j\pm\frac{1}{2}} - a\frac{\Delta t}{2}),
\end{aligned}$$

$$\begin{aligned}
U_{j\pm\frac{1}{2}}^{n+1} &= u(x_{j\pm\frac{1}{2}}, t_{n+1}) \\
&= u(x_{j\pm\frac{1}{2}} - a(t_{n+1} - t_n), t_n) \\
&= u(x_{j\pm\frac{1}{2}} - a\Delta t, t_n) \\
&= \alpha_0^n \varphi_0(x_{j\pm\frac{1}{2}} - a\Delta t) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1} \beta_{l,k}^n \psi_{l,k}(x_{j\pm\frac{1}{2}} - a\Delta t),
\end{aligned}$$

and

$$\begin{aligned}
U_{j\pm\frac{1}{2}}^{n} &= u(x_{j\pm\frac{1}{2}}, t_n) \\
&= u(x_{j\pm\frac{1}{2}} - a(t_n - t_n), t_n) \\
&= u(x_{j\pm\frac{1}{2}}, t_n) \\
&= \alpha_0^n \varphi_0(x_{j\pm\frac{1}{2}}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1} \beta_{l,k}^n \psi_{l,k}(x_{j\pm\frac{1}{2}}).
\end{aligned}$$

So,

$$\begin{aligned}
F_{j\pm\frac{1}{2}}^n \approx \frac{1}{6}a(\alpha_0^n &\left[\varphi_0(x_{j\pm\frac{1}{2}}) + 4\varphi_0(x_{j\pm\frac{1}{2}} - a\frac{\Delta t}{2}) + \varphi_0(x_{j\pm\frac{1}{2}} - a\Delta t)\right] \\
&+ \sum_{l=0}^{m}\sum_{k=0}^{2^m-1} \beta_{l,k}^n \left[\psi_{l,k}(x_{j\pm\frac{1}{2}}) + 4\psi_{l,k}(x_{j\pm\frac{1}{2}} - a\frac{\Delta t}{2}) + \psi_{l,k}(x_{j\pm\frac{1}{2}} - a\Delta t)\right])
\end{aligned}$$

Therefore, our numerical approximations for linear conservation laws are,

$$\begin{aligned}
U_j^{n+1} &= U_j^n - \frac{\Delta t}{\Delta x}[F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n] \\
&= U_j^n - \frac{a\Delta t}{6\Delta x}((U_{j+\frac{1}{2}}^n - U_{j-\frac{1}{2}}^n) + 4(U_{j+\frac{1}{2}}^{n+\frac{1}{2}} - U_{j-\frac{1}{2}}^{n+\frac{1}{2}}) + (U_{j+\frac{1}{2}}^{n+1} - U_{j+\frac{1}{2}}^{n+1}))
\end{aligned}$$

with the following definitions,

$$U_j^n = \alpha_0^n\varphi_0(x_j) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_j),$$

$$U_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \alpha_0^n\varphi_0(x_{j+\frac{1}{2}} - a\tfrac{\Delta t}{2}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j+\frac{1}{2}} - a\frac{\Delta t}{2}),$$

$$U_{j-\frac{1}{2}}^{n+\frac{1}{2}} = \alpha_0^n\varphi_0(x_{j-\frac{1}{2}} - a\tfrac{\Delta t}{2}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j-\frac{1}{2}} - a\frac{\Delta t}{2}),$$

$$U_{j+\frac{1}{2}}^{n+1} = \alpha_0^n\varphi_0(x_{j+\frac{1}{2}} - a\Delta t) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j+\frac{1}{2}} - a\Delta t),$$

$$U_{j-\frac{1}{2}}^{n+1} = \alpha_0^n\varphi_0(x_{j-\frac{1}{2}} - a\Delta t) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j-\frac{1}{2}} - a\Delta t),$$

$$U_{j+\frac{1}{2}}^{n} = \alpha_0^n\varphi_0(x_{j+\frac{1}{2}}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j+\frac{1}{2}}),$$

$$U_{j-\frac{1}{2}}^{n} = \alpha_0^n\varphi_0(x_{j-\frac{1}{2}}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j-\frac{1}{2}}).$$

6.2.2 Nonlinear Conservation Law

If $f(u)$ is nonlinear, the analogous formula is the following,

$$\begin{aligned} F_{j\pm\frac{1}{2}}^n &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(u(x_{j\pm\frac{1}{2}},\tau))\,d\tau \\ &\approx \frac{1}{\Delta t}\left(\frac{\Delta t}{6}\left[f(u(x_{j\pm\frac{1}{2}},t_n)) + 4f(u(x_{j\pm\frac{1}{2}},t_{n+\frac{1}{2}})) + f(u(x_{j\pm\frac{1}{2}},t_{n+1}))\right]\right) \end{aligned}$$

Therefore, our numerical approximations for nonlinear conservation laws are,

$$\begin{aligned} U_j^{n+1} &= U_j^n - \frac{\Delta t}{\Delta x}[F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n] \\ &= U_j^n - \frac{\Delta t}{6\Delta x}(f(U_{j+\frac{1}{2}}^n) - f(U_{j-\frac{1}{2}}^n) + 4(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})) + f(U_{j+\frac{1}{2}}^{n+1}) - f(U_{j+\frac{1}{2}}^{n+1})) \end{aligned}$$

with the following definitions, $U_j^n = \alpha_0^n\varphi_0(x_j) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_j)$,

$$U_{j\pm\frac{1}{2}}^{n+\frac{1}{2}} = \alpha_0^n\varphi_0(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\tfrac{\Delta t}{2}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\frac{\Delta t}{2}),$$

$$U_{j\pm\frac{1}{2}}^{n+1} = \alpha_0^n\varphi_0(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\Delta t) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\Delta t),$$

$$U_{j\pm\frac{1}{2}}^{n} = \alpha_0^n\varphi_0(x_{j\pm\frac{1}{2}}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta_{l,k}^n\psi_{l,k}(x_{j\pm\frac{1}{2}}).$$

Figure 6.3 illustrates the relationship between the application of the method of characteristics, Simpson's rule, and the finite volume method (FVM).

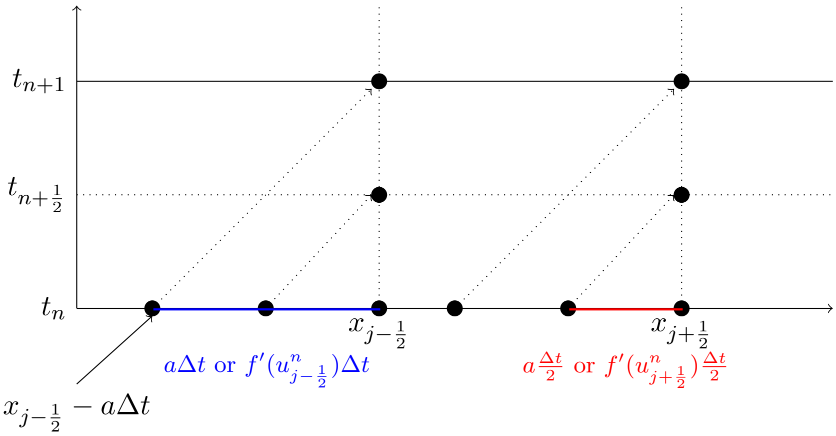


FIGURE 6.3: Use of Simpson's Rule to compute numerical fluxes in the Finite Volume Method

CHAPTER 7
CONVERGENCE OF THE APPROXIMATE SOLUTIONS TO A UNIQUE WEAK SOLUTION OF THE CONSERVATION LAW

In this chapter, we will prove convergence of our approximate solution method for conservation laws. Convergence of our approximation scheme will requires the resulting finite difference scheme satisfies an entropy condition (Smoller's book [15]).

As an example of how this can be achieved, we consider the following finite difference discretization of the equation
$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\left[F^n_{j+\frac{1}{2}} - F^n_{j-\frac{1}{2}}\right],$
with the following definitions,
$F^n_{j+\frac{1}{2}} = \frac{1}{6}\left(f(U^n_{j+\frac{1}{2}}) + 4f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}}) + f(U^{n+1}_{j+\frac{1}{2}})\right),$
$F^n_{j-\frac{1}{2}} = \frac{1}{6}\left(f(U^n_{j-\frac{1}{2}}) + 4f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}}) + f(U^{n+1}_{j-\frac{1}{2}})\right),$
$U_j^n = \alpha_0^n\varphi_0(x_j) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta^n_{l,k}\psi_{l,k}(x_j),$
$U^{n+\frac{1}{2}}_{j\pm\frac{1}{2}} = \alpha_0^n\varphi_0(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\frac{\Delta t}{2}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta^n_{l,k}\psi_{l,k}(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\frac{\Delta t}{2}),$
$U^{n+1}_{j\pm\frac{1}{2}} = \alpha_0^n\varphi_0(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\Delta t) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta^n_{l,k}\psi_{l,k}(x_{j\pm\frac{1}{2}} - f'(u(x_{j\pm\frac{1}{2}}))\Delta t),$
$U^n_{j\pm\frac{1}{2}} = \alpha_0^n\varphi_0(x_{j\pm\frac{1}{2}}) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta^n_{l,k}\psi_{l,k}(x_{j\pm\frac{1}{2}}),$
and with the initial data

$$u^0(x) = \alpha_0^0\varphi_0(x) + \sum_{l=0}^{m}\sum_{k=0}^{2^m-1}\beta^0_{l,k}\psi_{l,k}(x),$$

$$\alpha_0^0 = \langle u^0(x), \varphi_0(x)\rangle$$

$$\beta^0_{l,k} = \left\langle u^0(x), \psi_{l,k}(x) \right\rangle .$$

and we show that

1. Our approximate solution satisfies (a),(b), and (c) of Theorem 16.1 [15]. That is, let $u^0 \in L_\infty(\mathbb{R})$ and let $f \in C^2(\mathbb{R})$ with $f'' > 0$ on $\left\{u : |u| \leq \left\|u^0\right\|_\infty\right\}$. Then there exists a weak solution u with the following properties:
   (a) $|u(x,t)| \leq \left\|u^0\right\|_\infty \equiv M, (x,t) \in \mathbb{R} \times \mathbb{R}_+$.
   (b) There is a constant $E > 0$, depending only on M, $\mu = \min\left\{f''(u) : |u| \leq \left\|u^0\right\|_\infty\right\}$ and $A = \max\left\{f'(u) : |u| \leq \left\|u^0\right\|_\infty\right\}$, such that for every $a > 0$, $t > 0$, and $x \in \mathbb{R}$,
   $$\frac{u(x+a,t) - u(x,t)}{a} < \frac{E}{t}$$
   (c) u is stable and depends continuously on u^0 in the following sense: If $u^0, v^0 \in L_\infty(\mathbb{R}) \cap L_1(\mathbb{R})$ with $\left\|v^0\right\|_\infty \leq \left\|u^0\right\|_\infty$ and v is the corresponding solution constructed from the process in the proof, then for every $x_1, x_2 \in \mathbb{R}$, with $x_1 < x_2$, and every $t > 0$,
   $$\int_{x_1}^{x_2} |u(x,t) - v(x,t)| \; dx \leq \int_{x_1 - At}^{x_2 + At} \left|u^0(x) - v^0(x)\right| \; dx$$
2. As $\Delta t, \Delta x \searrow 0$, the solution converges to a weak solution satisfying (a),(b), and (c). The convergence is a weak version of $u^n_j \longrightarrow u(j\Delta x, n\Delta t)$-convergence to an entropy solution.

We will use the notation

$$M = \left\|u^0\right\|_{L_\infty},$$

$$\mu = \min\left\{f''(u) : |u| \leq \left\|u^0\right\|_\infty\right\}$$

and

$$A = \max\left\{f'(u) : |u| \leq \left\|u^0\right\|_\infty\right\}.$$

in the rest of this section and in the proofs that follow. We will also use the Lax-Friedrichs MRA as constructed in our work.

7.1 Uniform Bound on the Approximate Solution

Lemma 7.1. *We show that if* $\left|U_j^0\right| \leq M$*, then*

$$\left|U_j^n\right| \leq M \text{ for all } j \in \mathbb{R}, \quad n \in \mathbb{R}_+.$$

Proof. We write our difference approximation as following,

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}[F^n_{j+\frac{1}{2}} - F^n_{j-\frac{1}{2}}]$$

with

$$\begin{aligned} F^n_{j+\frac{1}{2}} &= \frac{1}{6}(f(U^n_{j+\frac{1}{2}}) + 4f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}}) + f(U^{n+1}_{j+\frac{1}{2}})) \\ F^n_{j-\frac{1}{2}} &= \frac{1}{6}(f(U^n_{j-\frac{1}{2}}) + 4f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}}) + f(U^{n+1}_{j-\frac{1}{2}})). \end{aligned}$$

So that,

$$\begin{aligned} U_j^{n+1} &= U_j^n \\ &- \frac{\Delta t}{6\Delta x}\left(f(U^n_{j+\frac{1}{2}}) - f(U^n_{j-\frac{1}{2}}) + 4\left(f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}}) - f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\right) + f(U^{n+1}_{j+\frac{1}{2}}) - f(U^{n+1}_{j+\frac{1}{2}})\right) \\ &= \frac{1}{6}\left(U_j^n - \frac{\Delta t}{\Delta x}\left[f(U^n_{j+\frac{1}{2}}) - f(U^n_{j-\frac{1}{2}})\right]\right) + \frac{4}{6}\left(U_j^n - \frac{\Delta t}{\Delta x}\left[f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}}) - f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\right]\right) \\ &+ \frac{1}{6}\left(U_j^n - \frac{\Delta t}{\Delta x}\left[f(U^{n+1}_{j+\frac{1}{2}}) - f(U^{n+1}_{j+\frac{1}{2}})\right]\right) \end{aligned}$$

By the mean value theorem,

$$\begin{aligned} U_j^{n+1} &= \frac{1}{6}\left(U_j^n - \frac{\Delta t}{\Delta x}f'(\theta_1)(U^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}})\right) \\ &+ \frac{4}{6}\left(U_j^n - \frac{\Delta t}{\Delta x}f'(\theta_2)(U^{n+\frac{1}{2}}_{j+\frac{1}{2}} - U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\right) \\ &+ \frac{1}{6}\left(U_j^n - \frac{\Delta t}{\Delta x}f'(\theta_3)(U^{n+1}_{j+\frac{1}{2}} - U^{n+1}_{j-\frac{1}{2}})\right) \qquad (7.1) \end{aligned}$$

$$(7.2)$$

where $\theta_1 \in \left[U^n_{j-\frac{1}{2}}, U^n_{j+\frac{1}{2}}\right]$, $\theta_2 \in \left[U^{n+\frac{1}{2}}_{j-\frac{1}{2}}, U^{n+\frac{1}{2}}_{j+\frac{1}{2}}\right]$, and $\theta_3 \in \left[U^{n+1}_{j-\frac{1}{2}}, U^{n+1}_{j+\frac{1}{2}}\right]$.
Next, we use the polynomial framework definition (see section 3.2.2).

$$U^n_{j+\frac{1}{2}} = m^n_{j+\frac{1}{2}}(\alpha^n_{j+\frac{1}{2}}) = U^n_{j+1} + (U^n_j - U^n_{j+1})\alpha^n_{j+\frac{1}{2}},$$

$$U^n_{j-\frac{1}{2}} = m^n_{j-\frac{1}{2}}(\alpha^n_{j-\frac{1}{2}}) = U^n_j + (U^n_{j-1} - U^n_j)\alpha^n_{j-\frac{1}{2}}.$$

Also, we choose $\alpha^n_{j\pm\frac{1}{2}}$ from the Lax-Friedrichs MRA as constructed in Section 4.4.5,

$$\begin{aligned} U^n_{j+\frac{1}{2}} &= U^n_{j+1} + \left(\frac{1}{2} + \frac{1}{2}\frac{\Delta x}{a_+\Delta t}\right)(U^n_j - U^n_{j+1}), \\ U^n_{j-\frac{1}{2}} &= U^n_j + \left(\frac{1}{2} + \frac{1}{2}\frac{\Delta x}{a_-\Delta t}\right)(U^n_{j-1} - U^n_j), \end{aligned}$$

with $a_+ = f'(U^n_{j+\frac{1}{2}})$ and $a_- = f'(U^n_{j-\frac{1}{2}})$.

When we use these definitions we get a form that approximates the Lax-Friedrichs scheme used by Oleinik (see [13]). That is,

$$\begin{aligned} U^n_j - \frac{\Delta t}{\Delta x}f'(\theta_1)(U^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}}) &\approx \frac{1}{2}\left(U^n_{j+1} + U^n_{j-1}\right) - \frac{\Delta t}{2\Delta x}\left(f(U^n_{j+1}) - f(U^n_{j-1})\right) \\ &= \frac{1}{2}\left(U^n_{j+1} + U^n_{j-1}\right) - \frac{\Delta t}{2\Delta x}\left(f'(\theta_4)(U^n_{j+1} - U^n_{j-1})\right) \\ &= \left(\frac{1}{2} - \frac{\Delta t}{2\Delta x}f'(\theta_4)\right)U^n_{j+1} + \left(\frac{1}{2} + \frac{\Delta t}{2\Delta x}f'(\theta_4)\right)U^n_{j-1}. \end{aligned}$$

If we assume

$$\begin{aligned} \frac{1}{2} + \frac{\Delta t}{2\Delta x}f'(\theta) \geq 0, \\ \frac{1}{2} - \frac{\Delta t}{2\Delta x}f'(\theta) \geq 0, \end{aligned}$$

then by a simple induction the first term in (7.1) will give

$$\begin{aligned} &\left|\frac{1}{6}\left(\left(\frac{1}{2} - \frac{\Delta t}{2\Delta x}f'(\theta_4)\right)U^n_{j+1} + \left(\frac{1}{2} + \frac{\Delta t}{2\Delta x}f'(\theta_4)\right)U^n_{j-1}\right)\right| \\ &\qquad \leq \frac{1}{6}\left(\left[\frac{1}{2} - \frac{\Delta t}{2\Delta x}f'(\theta_4)\right]M + \left[\frac{1}{2} + \frac{\Delta t}{2\Delta x}f'(\theta_4)\right]M\right) = \frac{1}{6}M. \end{aligned}$$

Now we do the same for the other two terms in Simpson's rule. Therefore, we have

$$\begin{aligned} \left|U^{n+1}_j\right| &\leq \frac{1}{6}\left(\left[\frac{1}{2} - \frac{\Delta t}{2\Delta x}f'(\theta_4)\right]M + \left[\frac{1}{2} + \frac{\Delta t}{2\Delta x}f'(\theta_4)\right]M\right) \\ &+ \frac{4}{6}\left(\left[\frac{1}{2} - \frac{\Delta x}{2\Delta t}f'(\theta_5)\right]M + \left[\frac{1}{2} + \frac{\Delta x}{2\Delta t}f'(\theta_5)\right]M\right) \\ &+ \frac{1}{6}\left(\left[\frac{1}{2} - \frac{\Delta x}{2\Delta t}f'(\theta_6)\right]M + \left[\frac{1}{2} + \frac{\Delta x}{2\Delta t}f'(\theta_6)\right]M\right) \\ &\leq \frac{1}{6}M + \frac{4}{6}M + \frac{1}{6}M \\ &= M. \end{aligned}$$

This implies

$$\left|U_j^{n+1}\right| \le M$$

whenever

$$\left|U_j^n\right| \le M$$

and the result we need.

7.2 A Discrete Entropy Condition on the Approximate Solution

The next result we need to prove is the following.

Lemma 7.2. *If* $C = \min\left(\frac{\mu}{2}, \frac{A}{4M}\right)$, *then*

$$\frac{U_j^n - U_{j-2}^n}{2\Delta x} \le \frac{E}{n\Delta t}$$

where $E = \frac{9}{C}$.

Proof. We define the following difference

$$z_j^n = \frac{U_j^n - U_{j-2}^n}{2\Delta x}.$$

Then our difference scheme implies

$$\begin{aligned}\frac{z_j^{n+1} - z_j^n}{\Delta t} + \frac{1}{36(\Delta x)^2}[f(U_{j+\frac{1}{2}}^n) - 2f(U_{j-\frac{1}{2}}^n) + f(U_{j-\frac{3}{2}}^n)\\ + 4(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - 2f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})\\ + f(U_{j-\frac{3}{2}}^{n+\frac{1}{2}})) + f(U_{j+\frac{1}{2}}^{n+1})\\ - 2f(U_{j+\frac{1}{2}}^{n+1}) + f(U_{j+\frac{3}{2}}^{n+1})] = 0.\end{aligned}$$

Substituting the definition above into our finite difference scheme results in

$$
\begin{aligned}
z_j^{n+1} &= z_j^n - \frac{\Delta t}{36(\Delta x)^2}[f(U_{j+\frac{1}{2}}^n) - 2f(U_{j-\frac{1}{2}}^n) + f(U_{j-\frac{3}{2}}^n) \\
&+ 4(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - 2f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}) + f(U_{j-\frac{3}{2}}^{n+\frac{1}{2}})) \\
&+ f(U_{j+\frac{1}{2}}^{n+1}) - 2f(U_{j+\frac{1}{2}}^{n+1}) + f(U_{j+\frac{3}{2}}^{n+1})] \\
&= \frac{z_{j+\frac{1}{2}}^n + z_{j-\frac{1}{2}}^n}{2} - \frac{\Delta t}{36(\Delta x)^2}[f(U_{j+\frac{1}{2}}^n) \\
&- 2f(U_{j-\frac{1}{2}}^n) + f(U_{j-\frac{3}{2}}^n) + 4(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) \\
&- 2f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}) + f(U_{j-\frac{3}{2}}^{n+\frac{1}{2}})) + f(U_{j+\frac{1}{2}}^{n+1}) \\
&- 2f(U_{j+\frac{1}{2}}^{n+1}) + f(U_{j+\frac{3}{2}}^{n+1})] \\
&= \frac{1}{2}\left(z_{j+\frac{1}{2}}^n + z_{j-\frac{1}{2}}^n\right) - \frac{\Delta t}{36(\Delta x)^2}[\left(f(U_{j+\frac{1}{2}}^n) - f(U_{j-\frac{1}{2}}^n)\right) \\
&+ \left(f(U_{j-\frac{3}{2}}^n) - f(U_{j-\frac{1}{2}}^n)\right) \\
&+ 4\left(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})\right) + 4\left(f(U_{j-\frac{3}{2}}^{n+\frac{1}{2}})\right) \\
&- f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}) + \left(f(U_{j+\frac{1}{2}}^{n+1}) - f(U_{j+\frac{1}{2}}^{n+1})\right) \\
&+ \left(f(U_{j+\frac{3}{2}}^{n+1}) - f(U_{j+\frac{1}{2}}^{n+1})\right)]
\end{aligned}
$$

Using Taylor expansions result in

$$f(U_{j+\frac{1}{2}}^n) = f(U_{j-\frac{1}{2}}^n) + f'(U_{j-\frac{1}{2}}^n)\left(U_{j+\frac{1}{2}}^n - U_{j-\frac{1}{2}}^n\right) + \frac{f''(\xi)}{2!}\left(U_{j+\frac{1}{2}}^n - U_{j-\frac{1}{2}}^n\right)^2$$

and

$$f(U_{j+\frac{1}{2}}^n) - f(U_{j-\frac{1}{2}}^n) = f'(U_{j-\frac{1}{2}}^n)\left(U_{j+\frac{1}{2}}^n - U_{j-\frac{1}{2}}^n\right) + \frac{f''(\theta_1)}{2!}\left(U_{j+\frac{1}{2}}^n - U_{j-\frac{1}{2}}^n\right)^2$$

where θ_1 is between $U_{j+\frac{1}{2}}^n$ and $U_{j-\frac{1}{2}}^n$.

Also,

$$f(U_{j-\frac{3}{2}}^n) = f(U_{j-\frac{3}{2}}^n) + f'(U_{j-\frac{1}{2}}^n)\left(U_{j-\frac{3}{2}}^n - U_{j-\frac{1}{2}}^n\right) + \frac{f''(\xi)}{2!}\left(U_{j-\frac{3}{2}}^n - U_{j-\frac{1}{2}}^n\right)^2$$

and

$$f(U_{j-\frac{3}{2}}^n) - f(U_{j-\frac{1}{2}}^n) = f'(U_{j-\frac{1}{2}}^n)\left(U_{j-\frac{3}{2}}^n - U_{j-\frac{1}{2}}^n\right) + \frac{f''(\theta_2)}{2!}\left(U_{j-\frac{3}{2}}^n - U_{j-\frac{1}{2}}^n\right)^2$$

where θ_2 is between $U^n_{j-\frac{1}{2}}$ and $U^n_{j-\frac{3}{2}}$.

Therefore,

$$\begin{aligned} z_j^{n+1} = {} & \frac{1}{2}\left(z^n_{j+\frac{1}{2}} + z^n_{j-\frac{1}{2}}\right) - \frac{\Delta t}{36(\Delta x)^2}[f'(U^n_{j-\frac{1}{2}})\left(U^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}}\right) && (7.3)\\ & + \frac{f''(\theta_1)}{2!}\left(U^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}}\right)^2 \\ & + f'(U^n_{j-\frac{1}{2}})\left(U^n_{j-\frac{3}{2}} - U^n_{j-\frac{1}{2}}\right) + \frac{f''(\theta_2)}{2!}\left(U^n_{j-\frac{3}{2}} - U^n_{j-\frac{1}{2}}\right)^2 \\ & + 4\left(f'(U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\left(U^{n+\frac{1}{2}}_{j+\frac{1}{2}} - U^{n+\frac{1}{2}}_{j-\frac{1}{2}}\right) + \frac{f''(\theta_3)}{2!}\left(U^{n+\frac{1}{2}}_{j+\frac{1}{2}} - U^{n+\frac{1}{2}}_{j-\frac{1}{2}}\right)^2\right) \\ & + 4\left(f'(U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\left(U^{n+\frac{1}{2}}_{j-\frac{3}{2}} - U^{n+\frac{1}{2}}_{j-\frac{1}{2}}\right) + \frac{f''(\theta_4)}{2!}\left(U^{n+\frac{1}{2}}_{j-\frac{3}{2}} - U^{n+\frac{1}{2}}_{j-\frac{1}{2}}\right)^2\right) \\ & + f'(U^{n+1}_{j-\frac{1}{2}})\left(U^{n+1}_{j+\frac{1}{2}} - U^{n+1}_{j-\frac{1}{2}}\right) + \frac{f''(\theta_5)}{2!}\left(U^{n+1}_{j+\frac{1}{2}} - U^{n+1}_{j-\frac{1}{2}}\right)^2 \\ & + f'(U^{n+1}_{j-\frac{1}{2}})\left(U^{n+1}_{j-\frac{3}{2}} - U^{n+1}_{j-\frac{1}{2}}\right) + \frac{f''(\theta_6)}{2!}\left(U^{n+1}_{j-\frac{3}{2}} - U^{n+1}_{j-\frac{1}{2}}\right)^2] && (7.4) \end{aligned}$$

where

$$\theta_1 \in [U^n_{j-\frac{1}{2}}, U^n_{j+\frac{1}{2}}], \theta_2 \in [U^n_{j-\frac{3}{2}}, U^n_{j-\frac{1}{2}}], \theta_3 \in [U^{n+\frac{1}{2}}_{j-\frac{1}{2}}, U^{n+\frac{1}{2}}_{j+\frac{1}{2}}],$$

$$\theta_4 \in [U^{n+\frac{1}{2}}_{j-\frac{3}{2}}, U^{n+\frac{1}{2}}_{j-\frac{1}{2}}], \theta_5 \in [U^{n+1}_{j-\frac{1}{2}}, U^{n+1}_{j+\frac{1}{2}}], \theta_6 \in [U^{n+1}_{j-\frac{3}{2}}, U^{n+1}_{j-\frac{1}{2}}].$$

We use

$$z_j^n = \frac{U_j^n - U_{j-2}^n}{2\Delta x},$$

so

$$z^n_{j+\frac{1}{2}} = \frac{U^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}}}{2\Delta x} \Longrightarrow 2\Delta x z^n_{j+\frac{1}{2}} = U^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}}$$

and

$$z^n_{j-\frac{1}{2}} = \frac{U^n_{j-\frac{1}{2}} - U^n_{j-\frac{3}{2}}}{2\Delta x} \Longrightarrow 2\Delta x z^n_{j-\frac{1}{2}} = U^n_{j-\frac{1}{2}} - U^n_{j-\frac{3}{2}}.$$

Substituting the double line differences above into (7.2) gives the following expression.

$$
\begin{aligned}
z_j^{n+1} &= \frac{1}{2}\left(z_{j+\frac{1}{2}}^n + z_{j-\frac{1}{2}}^n\right) - \frac{\Delta t}{36(\Delta x)^2}[f'(U_{j-\frac{1}{2}}^n)2\Delta x z_{j+\frac{1}{2}}^n - f'(U_{j-\frac{1}{2}}^n)2\Delta x z_{j-\frac{1}{2}}^n \\
&+ \frac{f''(\theta_1)}{2!}(2\Delta x z_{j+\frac{1}{2}}^n)^2 + \frac{f''(\theta_2)}{2!}(2\Delta x z_{j-\frac{1}{2}}^n)^2 \\
&+ 4(f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})(2\Delta x z_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})(2\Delta x z_{j-\frac{1}{2}}^{n+\frac{1}{2}}) + \frac{f''(\theta_3)}{2!}(2\Delta x z_{j+\frac{1}{2}}^{n+\frac{1}{2}})^2 \\
&+ \frac{f''(\theta_4)}{2!}(2\Delta x z_{j-\frac{1}{2}}^{n+\frac{1}{2}})^2) + f'(U_{j-\frac{1}{2}}^{n+1})(2\Delta x z_{j+\frac{1}{2}}^{n+1}) - f'(U_{j-\frac{1}{2}}^{n+1})(2\Delta x z_{j-\frac{1}{2}}^{n+1}) \\
&+ \frac{f''(\theta_5)}{2!}(2\Delta x z_{j+\frac{1}{2}}^{n+1})^2 + \frac{f''(\theta_6)}{2!}(2\Delta x z_{j-\frac{1}{2}}^{n+1})^2] \\
&= \left(\frac{1}{2} - \frac{\Delta t}{18\Delta x} f'(U_{j-\frac{1}{2}}^n)\right) z_{j+\frac{1}{2}}^n + \left(\frac{1}{2} + \frac{\Delta t}{18\Delta x} f'(U_{j-\frac{1}{2}}^n)\right) z_{j-\frac{1}{2}}^n \\
&- \frac{\Delta t}{18}\left(f''(\theta_1)(z_{j+\frac{1}{2}}^n)^2 + f''(\theta_2)(z_{j-\frac{1}{2}}^n)^2\right) \\
&- \frac{2\Delta t}{9\Delta x}\left(f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})z_{j+\frac{1}{2}}^{n+\frac{1}{2}} - f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})z_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right) \\
&- \frac{2\Delta t}{9}\left(f''(\theta_3)(z_{j+\frac{1}{2}}^{n+\frac{1}{2}})^2 + f''(\theta_4)(z_{j-\frac{1}{2}}^{n+\frac{1}{2}})^2\right) \\
&- \frac{\Delta t}{18\Delta x}\left(f'(U_{j-\frac{1}{2}}^{n+1})z_{j+\frac{1}{2}}^{n+1} - f'(U_{j-\frac{1}{2}}^{n+1})z_{j-\frac{1}{2}}^{n+1}\right) \\
&- \frac{\Delta t}{18}\left(f''(\theta_5)(z_{j+\frac{1}{2}}^{n+1})^2 + f''(\theta_6)(z_{j-\frac{1}{2}}^{n+1})^2\right)
\end{aligned}
$$

We define $\widetilde{z}_j^n = \max\left\{\left|z_{j+\frac{1}{2}}^n\right|, \left|z_{j-\frac{1}{2}}^n\right|, 0\right\}$. If $\widetilde{z}_j^n = 0$, then $z_j^n \le \widetilde{z}_{j-1}^n = 0 \le \frac{E}{n\Delta t}$.

We can assume $\widetilde{z}_j^n \ne 0$, and we suppose $\widetilde{z}_j^n = z_{j+\frac{1}{2}}$ or $\widetilde{z}_j^n = z_{j-\frac{1}{2}}$.

Then, we have

$$
\begin{aligned}
z_j^{n+1} &\le \left(\frac{1}{2} - \frac{\Delta t}{18\Delta x} f'(U_{j-\frac{1}{2}}^n)\right) z_{j+\frac{1}{2}}^n + \left(\frac{1}{2} + \frac{\Delta t}{18\Delta x} f'(U_{j-\frac{1}{2}}^n)\right) z_{j-\frac{1}{2}}^n \\
&- \frac{2\Delta t}{9\Delta x}\left(f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})z_{j+\frac{1}{2}}^{n+\frac{1}{2}} - f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})z_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right) - \frac{\Delta t}{18\Delta x}\left(f'(U_{j-\frac{1}{2}}^{n+1})z_{j+\frac{1}{2}}^{n+1} - f'(U_{j-\frac{1}{2}}^{n+1})z_{j-\frac{1}{2}}^{n+1}\right) \\
&- \frac{\Delta t}{9}C\left((z_{j+\frac{1}{2}}^n)^2 + (z_{j-\frac{1}{2}}^n)^2\right) - \frac{4\Delta t}{9}C\left((z_{j+\frac{1}{2}}^{n+\frac{1}{2}})^2 + (z_{j-\frac{1}{2}}^{n+\frac{1}{2}})^2\right) \\
&- \frac{\Delta t}{9}C\left((z_{j+\frac{1}{2}}^{n+1})^2 + (z_{j-\frac{1}{2}}^{n+1})^2\right) \\
&\le \left(\frac{1}{2} - \frac{\Delta t}{18\Delta x} f'(U_{j-\frac{1}{2}}^n)\right) \widetilde{z}_j^n + \left(\frac{1}{2} + \frac{\Delta t}{18\Delta x} f'(U_{j-\frac{1}{2}}^n)\right) \widetilde{z}_j^n \\
&- \frac{2\Delta t}{9\Delta x}\left(f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})\widetilde{z}_j^{n+\frac{1}{2}} - f'(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})\widetilde{z}_j^{n+\frac{1}{2}}\right) - \frac{\Delta t}{18\Delta x}\left(f'(U_{j-\frac{1}{2}}^{n+1})\widetilde{z}_j^{n+1} - f'(U_{j-\frac{1}{2}}^{n+1})\widetilde{z}_j^{n+1}\right) \\
&- \frac{\Delta t}{9}C\left((z_j^n)^2\right) - \frac{4\Delta t}{9}C\left((z_j^{n+\frac{1}{2}})^2\right) - \frac{\Delta t}{9}C\left((z_j^{n+1})^2\right) \\
&= \frac{1}{2}\widetilde{z}_j^n + \frac{1}{2}\widetilde{z}_j^n - \frac{\Delta t}{9}C\left((z_j^n)^2\right) - \frac{4\Delta t}{9}C\left((z_j^{n+\frac{1}{2}})^2\right) - \frac{\Delta t}{9}C\left((z_j^{n+1})^2\right)
\end{aligned}
$$

$$\therefore z_j^{n+1} \leq \widetilde{z}_j^n - \frac{\Delta t}{3} C\left((\widetilde{z}_j^n)^2\right) - \frac{4\Delta t}{3} C\left((\widetilde{z}_j^{n+\frac{1}{2}})^2\right) - \frac{\Delta t}{3} C\left((\widetilde{z}_j^{n+1})^2\right).$$

We know that $\left|U_j^n\right| \leq M$ from the result in Section 7.1. So,

$$z_j^n = \frac{U_j^n - U_{j-2}^n}{2\Delta x} \Longrightarrow z_j^n \leq \frac{2M}{2\Delta x} = \frac{M}{\Delta x}.$$

Also, we have assumed

$$\frac{A\Delta t}{\Delta x} \leq 1 \Longrightarrow A\Delta t \leq \Delta x \Longrightarrow \frac{1}{A\Delta t} \geq \frac{1}{\Delta x},$$

and the fact that

$$C \leq \frac{A}{4M} \Longrightarrow 4CM \leq A \Longrightarrow \frac{1}{4CM} \geq \frac{1}{A}.$$

Thus we have

$$z_j^n \leq \left|z_j^n\right| \leq \frac{M}{\Delta x} \leq \frac{M}{A\Delta t} \leq \frac{M}{\Delta t}\frac{1}{4CM} = \frac{1}{4C\Delta t}.$$

This implies

$$z_j^n \leq \frac{1}{4C\Delta t}.$$

Let

$$M^n = \max\left\{\widetilde{z}_j^n\right\}.$$

Then, $M^n \geq 0$. Also, let $\phi(y) = y - C\Delta t(y)^2$.

Then, $\phi'(y) = 1 - 2C\Delta t y > 0 \Longrightarrow \phi$ is a increasing function if $\frac{1}{2C\Delta t} > y$.

Therefore,

$$z_j^n \leq \frac{1}{4C\Delta t} < \frac{1}{2C\Delta t},$$

so that $\phi(\widetilde{z}_j^n) \leq \phi(M^n)$, and this gives

$$\begin{aligned} z_j^{n+1} &\leq \widetilde{z}_j^n - \frac{\Delta t}{9} C\left((\widetilde{z}_j^n)^2\right) - \frac{4\Delta t}{9} C\left((\widetilde{z}_j^{n+\frac{1}{2}})^2\right) - \frac{\Delta t}{9} C\left((\widetilde{z}_j^{n+1})^2\right) \\ &\leq \widetilde{z}_j^n - \frac{\Delta t}{9} C\left((\widetilde{z}_j^n)^2\right) \\ &\leq M^n - \frac{\Delta t}{9} C(M^n)^2. \end{aligned}$$

As in Oleinik's proof we arrive at

$$z_j^{n+1} \leq M^n - \frac{\Delta t}{9} C(M^n)^2,$$

and this implies

$$M^{n+1} \leq M^n - \frac{\Delta t}{9} C (M^n)^2.$$

We shall show that this necessarily implies

$$M^n \leq \frac{1}{\frac{nC\Delta t}{9} + \frac{1}{M^0}}. \tag{7.5}$$

Assuming (7.3), we have

$$z_j^n \leq M^n \leq \frac{1}{\frac{nC\Delta t}{9} + \frac{1}{M^0}} \leq \frac{9}{Cn\Delta t} = \frac{E}{n\Delta t},$$

and this gives

$$\frac{U_j^n - U_{j-2}^n}{2\Delta x} \leq \frac{E}{n\Delta t}.$$

We have to prove

$$M^n \leq \frac{1}{\frac{nC\Delta t}{9} + \frac{1}{M^0}}.$$

We proceed by induction. The case $n = 0$ is trivial. So we show that the result holds when n is replaced by $n + 1$. From

$$M^n \leq \frac{1}{\frac{nC\Delta t}{9} + \frac{1}{M^0}},$$

$$\frac{1}{M^n} \geq \frac{nC\Delta t}{9} + \frac{1}{M^0}.$$

So

$$1 - \frac{C\Delta t M^n}{9} \geq 1 - \frac{Cn\Delta t M^n}{9} \geq \frac{1}{M^0} \geq 0$$

and

$$1 - \left(\frac{C\Delta t M^n}{9}\right)^2 \geq 0.$$

This gives the inequality

$$\underbrace{\left(1 - \frac{C\Delta t M^n}{9}\right)}_{\geq 0} \underbrace{\left(1 + \frac{C\Delta t M^n}{9}\right)}_{\geq 0} \geq 0.$$

From $M^{n+1} \leq M^n - \frac{\Delta t}{9} C (M^n)^2$, we have

$$M^{n+1} \leq M^n \left(1 - \frac{\Delta t}{9} C M^n\right),$$

and

$$\frac{M^{n+1}}{1-\frac{\Delta t}{9}CM^n} \leq M^n \leq \frac{M^n}{1-\left(\frac{\Delta t}{9}CM^n\right)^2}.$$

From

$$\frac{M^{n+1}}{1-\frac{\Delta t}{9}CM^n} \leq \frac{M^n}{1-\left(\frac{\Delta t}{9}CM^n\right)^2},$$

we can write

$$\begin{aligned} M^{n+1} &\leq \frac{M^n}{1+\frac{C\Delta t}{9}M^n} = \frac{1}{\frac{1}{M^n}+\frac{C\Delta t}{9}} \\ &\leq \frac{1}{\frac{Cn\Delta t}{9}+\frac{1}{M^0}+\frac{C\Delta t}{9}} = \frac{1}{\frac{C\Delta t}{9}(n+1)+\frac{1}{M^0}}. \end{aligned}$$

Therefore,

$$M^n \leq \frac{1}{\frac{nC\Delta t}{9}+\frac{1}{M^0}},$$

and

$$z_j^n \leq M^n \leq \frac{1}{\frac{nC\Delta t}{9}+\frac{1}{M^0}} \leq \frac{9}{Cn\Delta t} = \frac{E}{n\Delta t}.$$

Thus

$$M^n \leq \frac{1}{\frac{nC\Delta t}{9}+\frac{1}{M^0}}$$

holds for all n. This proves the result.

7.3 Stability of the Approximate Solution

7.3.1 A Total Variation Bound

Following along the proof in Smoller [15], the next result shows that the variation of the difference method including Simpson's rule quadrature is locally bounded whenever $n\Delta t \geq 0$. The result we need is

Lemma 7.3. *Let* $\left\{U_j^n\right\}$ *and* $\left\{V_j^n\right\}$ *be solutions corresponding to the initial values* $\left\{U_j^0\right\}$ *and* $\left\{V_j^0\right\}$ *, respectively, with* $\sup\left|U_j^0\right| \leq M$ *and* $\sup\left|V_j^0\right| \leq M$*. Then for* $n > 0$ *we have*

$$\sum_{|j|\leq J}\left|U_j^n - V_j^n\right|\Delta x \leq \sum_{|j|\leq J+n}\left|U_j^0 - V_j^0\right|\Delta x.$$

Proof. Let $W_j^n = U_j^n - V_j^n$, then we have

$$
\begin{aligned}
W_j^{n+1} &= U_j^{n+1} - V_j^{n+1} \\
&= U_j^n - \frac{\Delta t}{6\Delta x}(f(U_{j+\frac{1}{2}}^n) - f(U_{j-\frac{1}{2}}^n) + 4(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) \\
&- f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}})) + f(U_{j+\frac{1}{2}}^{n+1}) - f(U_{j+\frac{1}{2}}^{n+1})) \\
&- V_j^n + \frac{\Delta t}{6\Delta x}(f(V_{j+\frac{1}{2}}^n) - f(V_{j-\frac{1}{2}}^n) + 4(f(V_{j+\frac{1}{2}}^{n+\frac{1}{2}}) \\
&- f(V_{j-\frac{1}{2}}^{n+\frac{1}{2}})) + f(V_{j+\frac{1}{2}}^{n+1}) - f(V_{j+\frac{1}{2}}^{n+1})) \\
&= \left(U_j^n - V_j^n\right) - \frac{\Delta t}{6\Delta x}\left(f(U_{j+\frac{1}{2}}^n) - f(V_{j+\frac{1}{2}}^n)\right) + \frac{\Delta t}{6\Delta x}\left(f(U_{j-\frac{1}{2}}^n) - f(V_{j-\frac{1}{2}}^n)\right) \\
&- \frac{4\Delta t}{6\Delta x}\left(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - f(V_{j+\frac{1}{2}}^{n+\frac{1}{2}})\right) + \frac{4\Delta t}{6\Delta x}\left(f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}) - f(V_{j-\frac{1}{2}}^{n+\frac{1}{2}})\right) \\
&- \frac{\Delta t}{6\Delta x}\left(f(U_{j+\frac{1}{2}}^{n+1}) - f(V_{j+\frac{1}{2}}^{n+1})\right) + \frac{\Delta t}{6\Delta x}\left(f(U_{j-\frac{1}{2}}^{n+1}) - f(V_{j-\frac{1}{2}}^{n+1})\right) \\
&= \frac{1}{6}\left[\left(U_j^n - V_j^n\right) - \frac{\Delta t}{\Delta x}\left(\left(f(U_{j+\frac{1}{2}}^n) - f(V_{j+\frac{1}{2}}^n)\right) - \left(f(U_{j-\frac{1}{2}}^n) - f(V_{j-\frac{1}{2}}^n)\right)\right)\right] \\
&+ \frac{4}{6}\left[\left(U_j^n - V_j^n\right) - \frac{\Delta t}{\Delta x}\left(\left(f(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - f(V_{j+\frac{1}{2}}^{n+\frac{1}{2}})\right) - \left(f(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}) - f(V_{j-\frac{1}{2}}^{n+\frac{1}{2}})\right)\right)\right] \\
&+ \frac{1}{6}\left[\left(U_j^n - V_j^n\right) - \frac{\Delta t}{\Delta x}\left(\left(f(U_{j+\frac{1}{2}}^{n+1}) - f(V_{j+\frac{1}{2}}^{n+1})\right) - \left(f(U_{j-\frac{1}{2}}^{n+1}) - f(V_{j-\frac{1}{2}}^{n+1})\right)\right)\right] \\
&= \frac{1}{6}\left[\left(U_j^n - V_j^n\right) - \frac{\Delta t}{\Delta x}f'(\theta_1)\left((U_{j+\frac{1}{2}}^n - V_{j+\frac{1}{2}}^n) - (U_{j-\frac{1}{2}}^n - V_{j-\frac{1}{2}}^n)\right)\right] \\
&+ \frac{4}{6}\left[\left(U_j^n - V_j^n\right) - \frac{\Delta t}{\Delta x}f'(\theta_2)\left((U_{j+\frac{1}{2}}^{n+\frac{1}{2}} - V_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - (U_{j-\frac{1}{2}}^{n+\frac{1}{2}} - V_{j-\frac{1}{2}}^{n+\frac{1}{2}})\right)\right] \\
&+ \frac{1}{6}\left[\left(U_j^n - V_j^n\right) - \frac{\Delta t}{\Delta x}f'(\theta_3)\left((U_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^{n+1}) - (U_{j-\frac{1}{2}}^{n+1} - V_{j-\frac{1}{2}}^{n+1})\right)\right]
\end{aligned}
$$

This result is obtained by the same approach as Section 7.1.

- We use the polynomial framework definition (see Section 3.2.2),

$$U_{j+\frac{1}{2}}^n = m_{j+\frac{1}{2}}^n(\alpha_{j+\frac{1}{2}}^n) = U_{j+1}^n + (U_j^n - U_{j+1}^n)\alpha_{j+\frac{1}{2}}^n,$$

  and

$$U_{j-\frac{1}{2}}^n = m_{j-\frac{1}{2}}^n(\alpha_{j-\frac{1}{2}}^n) = U_j^n + (U_{j-1}^n - U_j^n)\alpha_{j-\frac{1}{2}}^n.$$

- We choose $\alpha_{j\pm\frac{1}{2}}^n$ for Lax-Friedrichs method (see Section 4.4.5),

$$U_{j+\frac{1}{2}}^n = U_{j+1}^n + \left(\frac{1}{2} + \frac{1}{2}\frac{\Delta x}{a_+\Delta t}\right)(U_j^n - U_{j+1}^n),$$

and

$$U^n_{j-\frac{1}{2}} = U^n_j + \left(\frac{1}{2} + \frac{1}{2}\frac{\Delta x}{a_-\Delta t}\right)(U^n_{j-1} - U^n_j),$$

with

$$a_+ = f'(U^n_{j+\frac{1}{2}})$$

and $a_- = f'(U^n_{j-\frac{1}{2}})$.

- When we use these forms we get a form that approximates Lax-Friedrichs. That is,

$$\begin{gathered}
\left(U^n_j - V^n_j\right) - \tfrac{\Delta t}{\Delta x} f'(\theta_1)\left((U^n_{j+\frac{1}{2}} - V^n_{j+\frac{1}{2}}) - (U^n_{j-\frac{1}{2}} - V^n_{j-\frac{1}{2}})\right) \\
\approx \tfrac{1}{2}((U^n_{j+1} - V^n_{j+1}) + (U^n_{j-1} - V^n_{j-1})) \\
-\tfrac{\Delta t}{2\Delta x}\left((f(U^n_{j+1}) - f(V^n_{j+1})) - (f(U^n_{j-1}) - f(V^n_{j-1}))\right) \\
= (\tfrac{1}{2} - \tfrac{\Delta t}{2\Delta x} f'(\theta_4))(U^n_{j+1} - V^n_{j+1}) \\
+(\tfrac{1}{2} + \tfrac{\Delta t}{2\Delta x} f'(\theta_4))(U^n_{j-1} - V^n_{j-1}) \\
= (\tfrac{1}{2} - \tfrac{\Delta t}{2\Delta x} f'(\theta_4))W^n_{j+1} + (\tfrac{1}{2} + \tfrac{\Delta t}{2\Delta x} f'(\theta_4))W^n_{j-1}
\end{gathered}$$

Therefore,

$$\begin{aligned}
W^{n+1}_j = &\frac{1}{6}\left[(\frac{1}{2} - \frac{\Delta t}{2\Delta x} f'(\theta_4))W^n_{j+1} + (\frac{1}{2} + \frac{\Delta t}{2\Delta x} f'(\theta_4))W^n_{j-1}\right] \\
&+ \frac{4}{6}\left[(\frac{1}{2} - \frac{\Delta t}{2\Delta x} f'(\theta_5))W^{n+\frac{1}{2}}_{j+1} + (\frac{1}{2} + \frac{\Delta t}{2\Delta x} f'(\theta_5))W^{n+\frac{1}{2}}_{j-1}\right] \\
&+ \frac{1}{6}\left[(\frac{1}{2} - \frac{\Delta t}{2\Delta x} f'(\theta_6))W^{n+1}_{j+1} + (\frac{1}{2} + \frac{\Delta t}{2\Delta x} f'(\theta_6))W^{n+1}_{j-1}\right]
\end{aligned}$$

Since the stability condition $\frac{A\Delta t}{\Delta x} \leq 1$ is assumed true, the coefficients of $W^n_{j\pm1}$,$W^{n+\frac{1}{2}}_{j\pm1}$, and $W^{n+1}_{j\pm1}$ are non-negative; using this we have

$$\begin{aligned}
\sum_{|j|\le J}\left|W_j^{n+1}\right| &\le \sum_{|j|\le J}\frac{1}{6}\left[(\frac{1}{2}-\frac{\Delta t}{2\Delta x}f'(\theta_4))\left|W_{j+1}^n\right| + (\frac{1}{2}+\frac{\Delta t}{2\Delta x}f'(\theta_4))\left|W_{j-1}^n\right|\right] \\
&+ \sum_{|j|\le J}\frac{4}{6}\left[(\frac{1}{2}-\frac{\Delta t}{2\Delta x}f'(\theta_5))\left|W_{j+1}^{n+\frac{1}{2}}\right| + (\frac{1}{2}+\frac{\Delta t}{2\Delta x}f'(\theta_5))\left|W_{j-1}^{n+\frac{1}{2}}\right|\right] \\
&+ \sum_{|j|\le J}\frac{1}{6}\left[(\frac{1}{2}-\frac{\Delta t}{2\Delta x}f'(\theta_6))\left|W_{j+1}^{n+1}\right| + (\frac{1}{2}+\frac{\Delta t}{2\Delta x}f'(\theta_6))\left|W_{j-1}^{n+1}\right|\right] \\
&\le \sum_{|m|\le J+1}\frac{1}{6}\left[(\frac{1}{2}-\frac{\Delta t}{2\Delta x}f'(\theta_4))\left|W_m^n\right| + (\frac{1}{2}+\frac{\Delta t}{2\Delta x}f'(\theta_4))\left|W_m^n\right|\right] \\
&+ \sum_{|m|\le J+1}\frac{4}{6}\left[(\frac{1}{2}-\frac{\Delta t}{2\Delta x}f'(\theta_5))\left|W_m^{n+\frac{1}{2}}\right| + (\frac{1}{2}+\frac{\Delta t}{2\Delta x}f'(\theta_5))\left|W_m^{n+\frac{1}{2}}\right|\right] \\
&+ \sum_{|m|\le J+1}\frac{1}{6}\left[(\frac{1}{2}-\frac{\Delta t}{2\Delta x}f'(\theta_6))\left|W_m^{n+1}\right| + (\frac{1}{2}+\frac{\Delta t}{2\Delta x}f'(\theta_6))\left|W_m^{n+1}\right|\right] \\
&\le \sum_{|m|\le J+1}\frac{1}{6}\left|W_m^n\right| + \sum_{|m|\le J+1}\frac{4}{6}\left|W_m^n\right| + \sum_{|m|\le J+1}\frac{1}{6}\left|W_m^n\right| \\
&= \sum_{|m|\le J+1}\left|W_m^n\right|.
\end{aligned}$$

Therefore,

$$\sum_{|j|\le J}\left|W_j^{n+1}\right| \le \sum_{|m|\le J+1}\left|W_m^n\right|,$$

$$\sum_{|j|\le J}\left|W_j^{n}\right| \le \sum_{|m|\le J+1}\left|W_m^{n-1}\right|,$$

$$\sum_{|j|\le J}\left|W_j^{n}\right|\Delta x \le \sum_{|j|\le J+n}\left|W_j^{0}\right|\Delta x,$$

and

$$\sum_{|j|\le J}\left|U_j^n - V_j^n\right|\Delta x \le \sum_{|j|\le J+n}\left|U_j^0 - V_j^0\right|\Delta x.$$

It follows from this by a sample induction argument, that the stability holds. See [15].

7.3.2 Convergence to a Weak Solution

We show the convergence of our difference approximation. We know that

1. U_j^n is uniformly bounded. (Section 7.1),
2. our approximate solution satisfies the entropy condition. (Section 7.2), and
3. our approximate solution is stable. (Section 7.3.1)

From this, we must show that our approximate solution is a weak solution.

Let $\Delta x_i \to 0$ as $i \to \infty$, and suppose that for $\phi \in C_0^3$

$$\lim_{i\to\infty} \int_{-\infty}^{\infty} \left[U_i(x,0) - u_0(x)\right] \phi(x,0) \;\; dx = 0.$$

Then u is satisfies

$$\int\int_{t>0} (u\phi_t + f(u)\phi_x) \;\; dx\; dt + \int_{t=0} u_0\phi \;\; dx = 0$$

as described in Section 2.3.1.

Proof, Our approximate solution can be written in the form

$$\begin{aligned}\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{1}{6\Delta x}\left[f(U^n_{j+\frac{1}{2}}) - f(U^n_{j-\frac{1}{2}})\right] + \frac{4}{6\Delta x}\left[f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}}) - f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\right] \\ + \frac{1}{6\Delta x}\left[f(U^{n+1}_{j+\frac{1}{2}}) - f(U^{n+1}_{j+\frac{1}{2}})\right] = 0.\end{aligned}$$

First, we multiply ϕ_j^n, then we have

$$\begin{aligned}\underbrace{\frac{\phi_j^{n+1}U_j^{n+1} - \phi_j^nU_j^n}{\Delta t}}_{①} - U_j^{n+1}\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + \underbrace{\frac{1}{6\Delta x}\left[\phi^n_{j+\frac{1}{2}} f(U^n_{j+\frac{1}{2}}) - \phi^n_{j-\frac{1}{2}} f(U^n_{j-\frac{1}{2}})\right]}_{②} \\ - f(U^n_{j+\frac{1}{2}})\left(\frac{\phi^n_{j+\frac{1}{2}} - \phi^n_j}{6\Delta x}\right) - f(U^n_{j-\frac{1}{2}})\left(\frac{\phi^n_j - \phi^n_{j-\frac{1}{2}}}{6\Delta x}\right) \\ + \underbrace{\frac{4}{6\Delta x}\left[\phi^{n+\frac{1}{2}}_{j+\frac{1}{2}} f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}}) - \phi^{n+\frac{1}{2}}_{j-\frac{1}{2}} f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\right]}_{③} - f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}})\left(\frac{4(\phi^{n+\frac{1}{2}}_{j+\frac{1}{2}} - \phi^n_j)}{6\Delta x}\right) \\ - f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}})\left(\frac{4(\phi^n_j - \phi^{n+\frac{1}{2}}_{j-\frac{1}{2}})}{6\Delta x}\right) + \underbrace{\frac{1}{6\Delta x}\left[\phi^{n+1}_{j+\frac{1}{2}} f(U^{n+1}_{j+\frac{1}{2}}) - \phi^{n+1}_{j-\frac{1}{2}} f(U^{n+1}_{j-\frac{1}{2}})\right]}_{④} \\ - f(U^{n+1}_{j+\frac{1}{2}})\left(\frac{\phi^{n+1}_{j+\frac{1}{2}} - \phi^n_j}{6\Delta x}\right) - f(U^{n+1}_{j-\frac{1}{2}})\left(\frac{\phi^n_j - \phi^{n+1}_{j-\frac{1}{2}}}{6\Delta x}\right) = 0.\end{aligned}$$

Since ϕ has compact support, we may assume that $\phi_j^n = 0$ if $n \geq \frac{T}{\Delta t}$. We multiply by $\Delta t \Delta x$ and sum over all $j \in \mathbb{R}$ and $n \in \mathbb{R}^+$. ①, ②, ③, and ④ are "telescoping" and they cancel, except for the first term with $n = 0$. So we have

$$
\begin{aligned}
&-\Delta x \sum_j U_j^0 \phi_j^0 + \Delta t \Delta x (-\sum_{j,n} U_j^{n+1} \frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} - \sum_{j,n} f(U^n_{j+\frac{1}{2}}) \left(\frac{\phi^n_{j+\frac{1}{2}} - \phi^n_j}{6\Delta x} \right) \\
&\quad - \sum_{j,n} f(U^n_{j-\frac{1}{2}}) \left(\frac{\phi^n_j - \phi^n_{j-\frac{1}{2}}}{6\Delta x} \right) - \sum_{j,n} f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}}) \left(\frac{4(\phi^{n+\frac{1}{2}}_{j+\frac{1}{2}} - \phi^n_j)}{6\Delta x} \right) \\
&\quad - \sum_{j,n} f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}}) \left(\frac{4(\phi^n_j - \phi^{n+\frac{1}{2}}_{j-\frac{1}{2}})}{6\Delta x} \right) - \sum_{j,n} f(U^{n+1}_{j+\frac{1}{2}}) \left(\frac{\phi^{n+1}_{j+\frac{1}{2}} - \phi^n_j}{6\Delta x} \right) \\
&\qquad\qquad - \sum_{j,n} f(U^{n+1}_{j-\frac{1}{2}}) \left(\frac{\phi^n_j - \phi^{n+1}_{j-\frac{1}{2}}}{6\Delta x} \right)) = 0
\end{aligned}
$$

Since $U_{\Delta t,\Delta x}$ is a piecewise constant, ϕ is smooth and the integrals are limits of step functions, we have

$$
\begin{aligned}
&-\int_{t=0} U_{\Delta t,\Delta x}\phi + \delta_1 - \int\int_{t\geq 0} U_{\Delta t,\Delta x}\phi_t + \delta_2 - \frac{1}{6}\int\int_{t\geq 0} f(U_{\Delta t,\Delta x})\phi_x + \delta_3 \\
&-\frac{1}{6}\int\int_{t\geq 0} f(U_{\Delta t,\Delta x})\phi_x + \delta_4 - \frac{1}{6}\int\int_{t\geq 0} f(U_{\Delta t,\Delta x})\phi_x + \delta_5 - \frac{1}{6}\int\int_{t\geq 0} f(U_{\Delta t,\Delta x})\phi_x \\
&\qquad + \delta_6 - \frac{1}{6}\int\int_{t\geq 0} f(U_{\Delta t,\Delta x})\phi_x + \delta_7 - \frac{1}{6}\int\int_{t\geq 0} f(U_{\Delta t,\Delta x})\phi_x + \delta_8 = 0,
\end{aligned}
$$

where $\delta_i \longrightarrow 0$, as $\Delta t, \Delta x \longrightarrow 0$. We replace $U_{\Delta t,\Delta x}$ by U_i. Then

$$\int\int_{t\geq 0} U_i\phi_t + f(U_i)\phi_x + \int_{t=0} U_i\phi = \delta(\Delta x_i, \Delta t_i),$$

where $\delta(\Delta x_i, \Delta t_i) \longrightarrow 0$ as $i \longrightarrow \infty$. We know that $U_i \longrightarrow u$ as $i \longrightarrow \infty$:

$$\int\int_{t\geq 0} U_i\phi_t + f(U_i)\phi_x \Longrightarrow \int\int_{t\geq 0} u\phi_t + f(u)\phi_x.$$

By the choice of the initial values:

$$\int_{t=0} U_i\phi \Longrightarrow \int_{t=0} u_0\phi$$

Therefore,

$$\int\int_{t\geq 0} u\phi_t + f(u)\phi_x + \int_{t=0} u_0\phi = 0.$$

This proves our solution of the numerical approximation is a weak solution to the conservation law. So, as $\Delta t, \Delta x$ converge to zero while satisfying the CFL condition our solution converge to a weak solution satisfying (a),(b), and (c).

7.4 Uniqueness

Diperna [7,8] has, it is shown that an entropy condition and bounds on the total variation are sufficient to guarantee that the numerical approximation converges to a unique limit function, u. Also, the paper gives us the function is the desired solution of our conservation law and guarantees the uniqueness of u. This completes the proof that our approximate solution method converges to a unique weak solution of the conservation laws.

CHAPTER 8
CONCLUSION AND FUTURE WORK

This chapter summarizes to work done in this dissertation. We have

1. a frame work for the development of Multiresolution Analysis (MRA) conditioned on numerical methods used to approximate the solution of conservation laws,
2. a way to construct the shape function and wavelet bases for an MRA via a polynomial framework and lifting schemes of Sweldens [17],
3. an example of how these pieces fit together in a finite volume method approach using a Simpson's rule approximation for computing numerical fluxes in the approximate solution of conservation laws,
4. the development of entropy conditions that can be used to condition the associated MRA with the entropy condition, which results in the definition of an entropy satisfying MRA, and
5. a way to prove the convergence of numerical schemes for conservation laws based on combing the steps in this work.

Note that the method presented in this dissertation combing a finite volume method approach with Simpson's rule for computing numerical fluxes is one example that can be created and analyzed using the work in this dissertation.

In the future it is hoped that the techniques developed in this dissertation can be used to classify any Multiresolution Analysis (MRA) conditioned on numerical fluxes for conservation laws. In turn it is hoped that an a-priori condition can be obtained that will guarantee the convergence of a given numerical method applied to conservation laws to a unique entropy satisfying solution to the original partial differential equation.

BIBLIOGRAPHY

[1] Albert Bogess and F.J Narcowch , *A first course in wavelets with Fourier Analysis* Hoboken, NJ: John Wiley & Sons, Inc., 2009

[2] Alberto Bressan, *Numerical methods for conservation laws and related equation,* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.2558&rep=rep1 &type=pdf

[3] I-Liang Chern, *Finite difference methods for solving differential equations,* http://scicomp.math.ntu.edu.tw/wiki/images/6/62/FD.pdf

[4] C.S. Coray and J.V. Koebbe, *Accuracy optimized methods for constrained numerical solutions of hyperbolic conservation laws,* J. Comp. Phys, **109:115-132**, Nov.1993.

[5] C.S. Coray and J.V. Koebbe, *High order accuracy optimized methods for constrained numerical solutions of hyperbolic conservation laws,* SIAM J. Sci. Comput, **Vol 15:846-865**, July.1994.

[6] R.Courant, K.O.Friedrichs, and H.Lewy, *On the partial difference equations of mathematical physics,* Math. Ann.,100(1928), pp.32-74.

[7] R. Diperna, *Convergence of approximate solutions to conservation laws,* Arch. Rat. Mech. Anay. ,**V82:27-70**, 1983.

[8] R. Diperna, *Measure-valued solutions to conservation laws,* Arch. Rat. Mech. Anay. ,**V88:223-270**, 1985.

[9] A.Harten, *High resolution schemes for hyperbolic conservation laws,* J. Comp. Phys.,**V49:357-393**, 1983.

[10] R.J LeVeque, *Finite volume methods for Hyperbolic problems,* Cambridge, UK: Cambridge University Press, 2004.

[11] R. J LeVeque, *Numerical methods for Conservation laws,* Lecture Notes in mathematics, ETH Zurich, Birkhauser, 1992.

[12] Siddhartha Mishra, *Numerical methods for conservation laws and related equation,* https://www.math.ethz.ch/education/bachelor/lectures/fs2013/math/nhdgl/ numcl_notes_HOMEPAGE.pdf

[13] O. Oleinik, *Discontinuous solutions of nonlinear differential equation,* Amer. Math. Soc. Transl. ,**Ser.2,26: 95-172**, 1957.

[14] Chi-Wang Shu, *Numerical methods for hyperbolic conservation laws (AM 257),* http://mathema.tician.de/dl/academic/notes/257/257.pdf

[15] J. Smoller, *Shock Waves and Reaction-Diffusion Equations,* Springer-Verlag, 1993.

[16] P.K. Sweby *High resolution schemes using flux limiters for hyperbolic conservation laws,* SIAM J. on Numerical Anay. ,**V21: No.5. pp 995-1011**, Oct. 1984.

[17] W. Sweldens and Peter Schroder, *Building your own wavelets at home.*

[18] W. Sweldens, *The lifting scheme: A construction of second generation wavelets,* Technical Report 1995:6, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995.

[19] W. Sweldens, *The lifting scheme: A custom-design construction of biothogonal wavelets,* Appl. Comput. Harmon. Anay, **3(2):186-200,** 1995.

[20] David F. Walnut, *An introduction to wavelet analysis,* New York, NY: Birkhauser Boston, 2002

[21] *Hyperbolic equations: Scalar one-dimensional conservation laws-lecture 11* http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-920j-numerical-methods-for-partial-differential-equations-sma-5212-spring-2003/lecture-notes/lec11_notes.pdf

APPENDICES

Appendix A.

Matlab Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%1D advection equation: du/dt+a du/dx=0%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%upwind scheme
a=1;
amin=min(0,a);
amax=max(0,a);
dx=0.01;
dt=0.5*dx;
tend=0.5;

%%%Discretization of domain
x=-1:dx:1;
t=0:dt:tend;

%%%initial condition
n=length(x);
u0=zeros(1,n);
u0(1:ceil(n/2))= 1;
u0((ceil(n/2)+1):n)=0;
unext=zeros(1,n);
u=u0;

%%%upwind
for k=t
    for j=2:n-1
        unext(j)=u(j)-a*dt/dx*(u(j)-u(j-1));
        unext(1)=unext(2);
        unext(n)=unext(n-1);
    end
        u=unext(1:n);
end
%%%%plot
%%%exact solution and numerical solution
y = zeros(size(x));
region1 =(-1<=x) & (x<=0.5) ;
y(region1) =1;
region2 = (0.5<=x) & (x<=1);
y(region2) = 0;

plot(x,y,x,u,'.'); xlim([0,1]); ylim([-.5,1.5]);
title('upwind scheme');
xlabel('x');
ylabel('u');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%Lax-Wendroff
```

```
for k=t
    for j=2:n-1
        unext(j)=u(j)-dt/(2*dx)*a*(u(j+1)-u(j-1))+
(dt^2/2)/(dx^2)*(a^2)*(u(j+1)-2*u(j)+u(j-1));
        unext(1)=unext(2);
        unext(n)=unext(n-1);
    end
        u=unext(1:n);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%Lax-Friendrichs
for k=t
    for j=2:n-1
        unext(j)=.5*(u(j+1)+u(j-1))-(a*dt/dx)/2*(u(j+1)-u(j-1));
        unext(1)=unext(2);
        unext(n)=unext(n-1);
    end
        u=unext(1:n);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%Beam-Warming
for k=t
    for j=3:n
        unext(j)=u(j)-(dt/(2*dx))*a*(3*u(j)-4*u(j-1)+u(j-2))
+(dt^2)/(2*dx^2)*a^2*(u(j)-2*u(j-1)+u(j-2));
        unext(1)=unext(2);
        unext(2)=unext(3);
    end
        u=unext(1:n);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%TVD scheme

r=zeros(1,n);
frl=zeros(1,n);
frh=zeros(1,n);
fll=zeros(1,n);
flh=zeros(1,n);
fright=zeros(1,n);
fleft=zeros(1,n);
unext=zeros(1,n);
u=u0;

for k=t
    for j=2:n-1
        if u(j)==u(j+1)
            r(j)=1;
        elseif a>0
            r(j)=(u(j)-u(j-1))/(u(j+1)-u(j));
        elseif a<0
            r(j)=(u(j+2)-u(j+1))/(u(j+1)-u(j));
        end
        r(1)=1; r(n)=1;
    end
    %%%flux limiter function
    phi=(r+abs(r))./(1+abs(r));

    for j=2:n-1
        frl(j)=amax*u(j)+amin*u(j+1);
        frh(j)=(1/2)*a*(u(j)+u(j+1))-(1/2)*(a^2)*(dt/dx)*(u(j+1)-u(j));
        fll(j)=amax*u(j-1)+amin*u(j);
        flh(j)=(1/2)*a*(u(j-1)+u(j))-(1/2)*(a^2)*(dt/dx)*(u(j)-u(j-1));
```

```
        fright(j)=frl(j)+phi(j)*(frh(j)-frl(j));
        fleft(j)=fll(j)+phi(j-1)*(flh(j)-fll(j));
        unext(j)=u(j)-dt/dx*(fright(j)-fleft(j));
    end
    unext(1)=unext(2);
    unext(n)=unext(n-1);
    u=unext(1:n);
end
%%%%plot
plot(x,y,x,u,'.'); xlim([0,1]); ylim([-.5,1.5]); title('TVD scheme');
xlabel('x');
ylabel('u');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Burgers' equation u_t+(1/2 u^2)_x=0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
%%%%selection of equation,method, and initial condition
type_burger=menu('Choose the equation:',...
'Inviscid Burgers equation', 'Viscid Burgers equation');
if (type_burger==1)
method=menu('Choose a numerical method:',...
'Upwind conservative','Lax-Friedrichs', 'Lax-Wendroff','Beam-Warming');
else
    method=menu('Choose a numerical method:',...
        'Parabolic Method');
end

ictype=menu('Choose the initial condition type:',...
    'shock','expansion');
%%%%Selection of numerical parameters
xend=1;
tend=0.5;
dx=0.01;
dt=0.5*dx;
x=-1:dx:xend;
t=0:dt:tend;
n=length(x);

%%%% set-up initial solution
u0=init(x,ictype);
u=u0;
unew=0*u;

%%%%numerical method
if (type_burger==1)
    for i=1:n,
    switch method
    case 1 %%%upwind conservative
    unew(2:end)=u(2:end)-dt/dx*(f(u(2:end))-f(u(1:end-1)));
    unew(1)=u(1);

    case 2 %%%Lax-Friedrichs
    unew(2:end-1)=0.5*(u(3:end)+u(1:end-2))-0.5*dt/dx* ...
    (f(u(3:end))-f(u(1:end-2)));
    unew(1)=u(1);
    unew(end)=u(end);

    case 3 %%%Lax-Wendroff
    unew(2:end-1)=u(2:end-1) ...
    -0.5 *dt/dx*(f(u(3:end))-f(u(1:end-2)))...
    +0.5*(dt/dx)^2*...
```

```
   (df(0.5*(u(3:end)+u(2:end-1))).*(f(u(3:end))-f(u(2:end-1)))-...
   df(0.5*(u(2:end-1)+u(1:end-2))).*(f(u(2:end-1))-f(u(1:end-2))) );
   unew(1)=u(1);
   unew(end)=u(end);

    case 4%%%Beam-Warming
    unew(3:end)= u(3:end)-0.5*(dt/dx)*(3*f(u(3:end))-4*f(u(2:end-1))
+...f(u(1:end-2)))...
    +0.5*(dt/dx)^2*...
   (df(0.5*(u(3:end)+u(2:end-1))).*(f(u(3:end))-f(u(2:end-1)))- ...
    df(0.5*(u(2:end-1)+u(1:end-2))).*(f(u(2:end-1))-f(u(1:end-2))));
    unew(1)=u(1);
    unew(2)=u(2);


    end
    u=unew;

    end
else
    for i=1:n,
    switch method
    case 1 %%%Parabolic method
    D=0.01;
   fminus=0.5*(f(u(2:end-1))+f(u(1:end-2)));
   fplus=0.5*(f(u(2:end-1))+f(u(3:end)));
unew(2:end-1)=u(2:end-1)+dt*(D*(u(3:end)-2*u(2:end-1)
+u(1:end-2))/(dx)^2
 ...-(fplus-fminus)/dx );
  unew(1)=u(1);
  unew(end)=u(end);
    end
    u=unew;

    end
end

y = zeros(size(x));
region1 =(-1<=x) & (x<=0.5) ;
y(region1) =1;

region2 = (0.5<=x) & (x<=1);
y(region2) = 0;

plot(x,y,x,u,'.'); xlim([0,1]); ylim([-.5,1.5]);
xlabel('x');
ylabel('u');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% initial condition for Burgers' equation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function ui=init(x,ictype)
xshift=0;
if(ictype==1)
    uL=1;
    uR=0;
    ui=uR+(uL-uR)*((x-xshift)<=0.0);
elseif(ictype==2)
    uL=0.5;
    uR=1;
    ui=uR+(uL-uR)*((x-xshift)<=0.0);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% f(u)=1/2 u^2, f'(u)=u
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ret=f(u)
        ret=0.5*u.^2;

function ret=df(u)
            ret=u;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%figure for Haar wavelet
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function  Haar
iter=8;
[phi,psi,x]=wavefun('db1',iter);
figure(1),
subplot(2,1,1), plot(x,phi), hold on
plot(-0.5+2*x, zeros(size(x))), hold off
title('phi Haar')
axis([-0.5,1.5,-1.5,1.5])
subplot(2,1,2),plot(x,psi), hold on
plot(-0.5+2*x, zeros(size(x))), hold off
title('psi Haar')
axis([-0.5,1.5,-1.5,1.5])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function upshape(n,nval)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%This method will generate the shape function data for
%%%various conservation law schemes
%%%INPUT: n=2^x (x= any integer; n=4,8,16,....)
%%%         nval=compute the number of points to update
%%%(nval=1,then your point is 2*n+1)
%%%Output:Upwind method with lifting schemes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alpha=1;
%%%initial shapefunction
for i=1:n+1
    shapefunc(1,i)=0;
end
%%Initialize the discrete dirac function value and the number of
%%values to the left and right of the center
indmid=n/2+1;
shapefunc(1,indmid)=1;

%%%Find new(1,i)=U_{i+1/2};
%%%U_{i+1/2}=U{i+1)+(U(i)-U(i+1))*alpha
%%% lifting framework
for l=1:nval
for i=1:2^(l-1)*n
new(1,i)=shapefunc(1,i+1)+(shapefunc(1,i)-shapefunc(1,i+1))*alpha;
end
for j=1:2^(l-1)*n
newshape(1,2*j-1)=shapefunc(1,j);
newshape(1,2*j)=new(1,j);
end
newshape(1,2^(l)*n+1)=shapefunc(1,2*(l-1)*n+1);
shapefunc=newshape;
end
x(1)=0;
dx=1/(2^nval*n);
for i=1:2^nval*n
    x(i+1)=x(i)+dx;
end
```

```
figure (1);
plot(x,shapefunc,'.');
axis([0 1 -0.5 1.5]);
%%%%%%Find the new shape function
na=2^nval*n/2;
for i=1:na
    tempfunc(i)=shapefunc(i+na/2);
end
for j=1:na+1
    tempfunc(j+na)=shapefunc(j+na/2);
end
newshapefunc=tempfunc;
%%%find the wavelet function
for l=1:2^nval*n+1
    wvletfunc(l)=shapefunc(l)-newshapefunc(l);
end
figure (2)
plot(x,wvletfunc,'.'); %%%wavelet function graph
axis([0 1 -1.5 1.5]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function TVDshape(n,nval,cfl)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%This method will generate the shape function data for various
%%conservation law schemes by TVD scheme
%%Input:n=2^x (x= any integer; n=4,8,16,....)
%%%     nval=compute the number of points to update
%%%     cfl=cfl condition number, 0<cfl<1
%%Output:TVD method with lifting schemes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%initial shapefunction
for i=1:n+1
    shapefunc(1,i)=0;
end
%%Initialize the discrete dirac function value and the number
%%of values to the left and right of the center
indmid=n/2+1;
shapefunc(1,indmid)=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Find new(1,i)=U_{i+1/2};
%%%U_{i+1/2}=U_{i+1)+(U_(i)-U_(i+1))*alpha
%%% Lifting framework%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for l=1:nval
%%%%flux limiter work of Sweby
num(1)=0; den(1)=0; num(2^(l-1)*n+1)=0; den(2^(l-1)*n+1)=0;
for i=2:2^(l-1)*n
num(i)=shapefunc(i-1)-shapefunc(i);
den(i)=shapefunc(i)-shapefunc(i+1);
end
%%% compute the sign of the ratio

for i=1:2^(l-1)*n+1
    signn(i)=1;
    if num(i)<0
        signn(i)=-1;
    end
    signd(i)=1;
    if den(i)<0
        signn(i)=-1;
    end
   signr(i)=signn(i)*signd(i);
```

```
end


for i=1:2^(l-1)*n+1
    if signr(i)<0
        c(i)=0;
    else
        if den(i)==0
            c(i)=2;
        else
        theta(i)=num(i)/den(i);

       if( 0< theta(i)) &&(theta(i)< 1/2)
        c(i)=2*theta(i);
        elseif (1/2<=theta(i)) &&(theta(i)<1)
        c(i)=1;
        elseif (1<=theta(i))&& (theta(i)<2)
        c(i)=theta(i);
        else
        c(i)=2;
        end
     end
    end
end

%%% alpha value for TVD
for i=1:2^(l-1)*n+1
    alpha(i)=(1/2)*(1+cfl)+(1/2)*(1-cfl)*(1-c(i));
end
%%%Lifting frame work
for i=1:2^(l-1)*n
new(1,i)=shapefunc(1,i+1)+(shapefunc(1,i)
-shapefunc(1,i+1))*alpha(i+1);
end
for j=1:2^(l-1)*n
    newshape(1,2*j-1)=shapefunc(1,j);
    newshape(1,2*j)=new(1,j);
end
newshape(2^l*n+1)=0;
shapefunc=newshape;
end

x(1)=0;
dx=1/(2^nval*n);
for i=1:2^nval*n
    x(i+1)=x(i)+dx;
end
figure (1) %%%shape function graph
plot(x,shapefunc,'.');
axis([0 1 -0.5 1.5]);
%%%%%%Find the new shape function
na=2^nval*n/2;
for i=1:na
    tempfunc(i)=shapefunc(i+na/2);
end
for j=1:na+1
    tempfunc(j+na)=shapefunc(j+na/2+1);
end
newshapefunc=tempfunc;
%%%find the wavelet function
for l=1:2^nval*n+1
    wvletfunc(l)=shapefunc(l)-newshapefunc(l);
```

```
end
figure (2)
plot(x,wvletfunc,'.'); %%%wavelet function graph
axis([0 1 -inf inf]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function sampleshapefunction(n,nval,gamma1,gamma2,ncfl)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%This method will generate the shape function data for
%%various conservation law schemes
%%Input:n=2^x (x= any integer; n=4,8,16,....)
%%%     nval=compute the number of points to update
%%%     gamma1,gamma2
%%%     ncfl=cfl(i) (cfl(1)=0, cfl(2)=0.1,...cfl(11)=1)
%%Output:Numerical method with lifting schemes
%%Using Lax-Wendroff (gamma1=1/2, gamm2=1/2),
%%Beam-Warming(gamma1=1/2, gamma2=-1/2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nstep=10;
dcfl=1/nstep;
cfl(1)=0;
for i=1:nstep
    cfl(i+1)=cfl(i)+dcfl;
    alpha(i)=gamma1+gamma2*cfl(i);
end
%%%initial shapefunction
for i=1:n+1
    shapefunc(1,i)=0;
end
%%Initialize the discrete dirac function value and the
%%number of values to the left and right of the center
indmid=n/2+1;
shapefunc(1,indmid)=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Find new(1,i)=U_{i+1/2};
%%%U_{i+1/2}=U_{i+1)+(U_(i)-U_(i+1))*alpha
%%% Lifting framework%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for l=1:nval
for i=1:2^(l-1)*n
new(1,i)=shapefunc(1,i+1)+
(shapefunc(1,i)-shapefunc(1,i+1))
*alpha(ncfl);
end
for j=1:2^(l-1)*n
newshape(1,2*j-1)=shapefunc(1,j);
newshape(1,2*j)=new(1,j);
end
newshape(1,2^(l)*n+1)=0;
shapefunc=newshape;
end
%%%%%Find the new shape function
na=2^nval*n/2;
for i=1:na
    tempfunc(i)=shapefunc(i+na/2);
end
for j=1:na+1
    tempfunc(j+na)=shapefunc(j+na/2);
end
newshapefunc=tempfunc;
%%%%%interval of x
x(1)=0;
```

```
dx=1/(2^nval*n);
for i=1:2^nval*n
    x(i+1)=x(i)+dx;
end
figure (1) %%%shape function graph
plot(x,shapefunc,'.');
axis([0 1 -0.5 1.5]);

%%%find the wavelet function
for l=1:2^nval*n+1
    wvletfunc(l)=shapefunc(l)-newshapefunc(l);
end
figure (2)
plot(x,wvletfunc,'.'); %%%wavelet function graph
axis([0 1 -1.5 1.5]);
```

CURRICULUM VITAE

Ju Y. Yi

EDUCATION:

Doctor of Philosophy, Mathematical Science, 2016
Utah State University, Logan, UT
Adviser: Dr. Joe Koebbe

Master of Art, Mathematics (Applied), 2000
University of California, San Diego (UCSD), La Jolla, CA

Bachelor of Science in Mathematics, 1998
University of California, Los Angeles (UCLA), Los Angeles, CA

Glendale community college, 1992-1995

RESEARCH INTERESTS:

Computational and applied mathematics, with special emphasis on wavelet construction and analysis of numerical schemes

TEACHING INTERESTS:

Undergraduate and graduate numerical analysis courses, PDE, ODE and numerical optimization courses

SKILLS:

Problem Solving: Excellent analytical and logical reasoning skills. Able to multitask. Can learn new skills quickly. Able to lead or work within a group environment.

Computer Languages and Math Program:

- Many years experience with **C++**
- Matlab (Fluent), Maple, Mathematica
- OpenCL, OpenMP

Other:

- Creative, motivated, innovative, stunningly beautiful. LaTeX, mathematical ability (obviously). Teaching skills.
- Korean (Fluent)
- Statistical Program: R
- Pass Actuarial Exam 1 (Exam P)-2003

Employment, Ventura College

Instructor for College Algebra, Pre-Algebra *Fall 2016*
Preparation and presentation of lectures, supervision of group work, writing and grading tests,grading projects and papers, preparing and grading the final exam for the course.

Employment, Utah State University

Graduate Instructor for College Algebra *Summer 2012, Summer 2015*
Graduate Instructor for Trigonometry *Summer 2014*
Preparation and presentation of lectures, supervision of group work, writing and grading tests,grading projects and papers, preparing and grading the final exam for the course. Set-up online program (WillyPlus):homework, quiz, etc.

Recitation leader for Calculus 1 *Fall 2015*
Recitation leader for College Algebra *Fall 2011, Spring 2012, Fall 2014, Spring 2015*
Recitation leader for Calculus Techniques *Spring2014*
Ran discussion/problem/review sessions twice a week that augmented the business calculus course, graded exams, some guest lecturing.

Graduate Instructor for Calculus 2 *Spring 2013, Fall 2013*
Graduate Instructor for Calculus 1 *Fall 2012, Summer 2013*
Preparation and presentation of lectures, supervision of group work, writing and grading tests,grading projects and papers, preparing and grading the final exam for the course.

Grader for Theory of Linear Algebra (graduate course) *fall 2014*

Employment, Thinkers' Club, LLC

Director *2003-2009*

- Teaching K-12th grade for mathematics, SAT, ACT
- Personal Management
- Control after school program

Employment, University of California, Los Angeles

Teaching Assistant *2001-2002*
Ran discussion/problem/review sessions twice a week that augmented the course, graded papers and quizzes, some guest lecturing.

- Calculus 1
- Vector Calculus
- Ordinary Differential Equation

ACTIVITIES:

USU SIAM Co-Founded the student chapter, 2012-2015

Science Unwrapped, SIAM chapter presentation-“image and data compression”, March 21, 2014 at USU

AMS student chapter member, 2013-2015