ERROR ANALYSIS OF THE CUBIC FRONT TRACKING AND RKDG METHOD FOR ONE DIMENSIONAL CONSERVATION LAWS

Raju Prasad Bhusal

A Dissertation

Submitted to the Graduate College of Bowling Green State University in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY

May 2018

Committee:

Tong Sun, Advisor

Andrew Layden,
Graduate Faculty Representative

Juan Bes

So-Hsiang Chou
ABSTRACT

Tong Sun, Advisor

The error analysis for the Runge-Kutta discontinuous Galerkin (RKDG) method for solving the scalar nonlinear conservation laws for the case of a smooth solution has been given in (23) and for the case of having a fully developed shock has given in (22). We use the Cubic Front Tracking and RKDG method to obtain the solution for the case between (23) and (22) where a shock is forming but not fully developed yet. The numerical smoothness approach used in (23) is generalized for the case between (23) and (22).
I dedicate this dissertation to my mom and to my wife whose support was uniform throughout the entire process.
ACKNOWLEDGMENTS

I would like to acknowledge Dr. Sun for his valuable suggestions throughout this dissertation. His support and innovative concept of “numerical smoothness” made this dissertation possible. I expect his cooperation in research in future as well. I would like to acknowledge my seniors Dr. David Rumsey and Dr. Adamou Fode, whose works were vital for me to figure out the problem. I would like to express my special gratitude to Dr. Chou for his support throughout this process. I experienced a great time with Dr. Chou during my graduate study. I am indebted to Dr. Bes for his valuable time and effort during my preliminary examination. I would like to express my sincere thanks to my committee member Dr. Andrew Layden.

I would want to remember my colleagues Paul, David, Jake, and Daniel for their collaboration and support during my graduate study. I am thankful to the department secretaries Marcia Seubert and Anna Lynch for reminding me of all the due-dates and helping to complete required paperwork.

Finally, I am indebted to my wife, Soni without whose support and cooperation all of my works would have been meaningless and futile.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER 1</th>
<th>INTRODUCTION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The Scalar Hyperbolic Conservation Law</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>The Cubic Front Tracking and RKDG Method</td>
<td>5</td>
</tr>
</tbody>
</table>

| CHAPTER 2  | PRELIMINARIES AND LITERATURE REVIEW                | 11   |
| 2.1        | General Framework for Error Analysis of Time Dependent Differential Equations | 12   |
| 2.2        | Literature Review                                  | 18   |

| CHAPTER 3  | MAIN ALGORITHM, SMOOTHNESS INDICATORS, AND ESTIMATION OF THE TRANSITION TIME | 21   |
| 3.1        | Main Algorithm                                     | 21   |
| 3.2        | The Error Splitting                                 | 25   |
| 3.3        | Smoothness Indicators                               | 27   |
| 3.3.1      | Spatial Smoothness Indicators                       | 27   |
| 3.3.2      | Temporal Smoothness Indicators                      | 28   |
| 3.4        | Estimation of the Transition Time                   | 29   |

| CHAPTER 4  | ERROR ESTIMATION ON THE FRONT TRACKING REGION       | 36   |
| 4.1        | Conversion Error Estimation                         | 36   |
| 4.2        | Error Estimation of the Cubic Front Tracking Solution | 41   |

| CHAPTER 5  | THE SPATIAL ERROR ESTIMATION ON THE DG REGION       | 44   |
| 5.1        | Spatial Error Estimation on the First Region $\Omega_f$ | 45   |
| 5.1.1      | Error Estimation of DG Cells                        | 46   |
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The Smooth Solution</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>The Solution having a Fully Developed Shock</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>Solution at the Different Time Stages</td>
<td>9</td>
</tr>
<tr>
<td>1.4</td>
<td>Fully Developed “s-shape” Solution After Locating the Shock</td>
<td>9</td>
</tr>
<tr>
<td>1.5</td>
<td>“s-shape” Solution Satisfying the Equal Area Principle</td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td>Error Splitting in Numerical Stability</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Error Splitting in Numerical Smoothness</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Error Splitting of the DG Solution</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>The Point $\left(x_0, u^0\right)$ at the beginning and at the end of the Transition time Phase II</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>The Region $\mathcal{R}_{n,j}$</td>
<td>46</td>
</tr>
<tr>
<td>5.2</td>
<td>Detachment of the DG Solution and FT solution</td>
<td>56</td>
</tr>
<tr>
<td>6.1</td>
<td>Shock Locations and Fully Developed “s-shape” Solutions</td>
<td>68</td>
</tr>
<tr>
<td>7.1</td>
<td>Initial Solution and Solution at Different Stages of the Transition time</td>
<td>71</td>
</tr>
<tr>
<td>7.2</td>
<td>Zoom in of Computed “s-shape” Solution Before the Shock is being Located</td>
<td>72</td>
</tr>
<tr>
<td>7.3</td>
<td>Zoom in of Computed “s-shape” Solution After the Shock is being Located</td>
<td>72</td>
</tr>
<tr>
<td>7.4</td>
<td>Smoothness Indicators on the $\Omega_U$ with $h = 0.1, \tau_n = 0.005$ and $t = 0.7$</td>
<td>74</td>
</tr>
<tr>
<td>7.5</td>
<td>Smoothness Indicators on the $\Omega_U$ with $h = 0.1, \tau_n = 0.005$ and $t = 0.8$</td>
<td>74</td>
</tr>
<tr>
<td>7.6</td>
<td>Smoothness Indicators on the $\Omega_U$ with $h = 0.1, \tau_n = 0.005$ and $t = 1.0$</td>
<td>75</td>
</tr>
<tr>
<td>7.7</td>
<td>Smoothness Indicators on the $\Omega_U$ with $h = 0.1, \tau_n = 0.005$ and $t = 1.15$</td>
<td>75</td>
</tr>
<tr>
<td>7.8</td>
<td>Smoothness Indicators on the $\Omega_U$ with $h = 0.1, \tau_n = 0.0025$ and $t = 0.7$</td>
<td>76</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Function Notations</td>
<td>21</td>
</tr>
<tr>
<td>7.1</td>
<td>The Contents of the Smoothness Indicators in Figures 7.4-7.18</td>
<td>73</td>
</tr>
<tr>
<td>7.2</td>
<td>Constant $\beta \sum \tilde{D}_{n,j}$ in the Upper Bound of $</td>
<td></td>
</tr>
<tr>
<td>7.3</td>
<td>Maximum Values of the Constants $N_{n,j}^4$ in the First Region $\Omega_U$</td>
<td>82</td>
</tr>
<tr>
<td>7.4</td>
<td>Constant $\frac{h}{\pi} \sum N_{n,j}^4$ in the Upper Bound of $</td>
<td></td>
</tr>
<tr>
<td>7.5</td>
<td>Constant $\left[ \frac{C_2 + \gamma h^{1/3} C_3}{C_1} \right] C_1 e^{b C_1 \tau_n / h}$ in the Upper Bound of $</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 1 INTRODUCTION

Many real-life problems are modeled as the scalar nonlinear conservation laws. In general, these problems have not been solved analytically. We need to explore the numerical schemes to approximate the solutions. To solve the scalar nonlinear conservation law numerically, one of the most popular modern methods is the Runge Kutta Discontinuous Galerkin (RKDG) method. First, in 1973, Reed and Hill introduced the discontinuous Galerkin (DG) method in the context of neutron transport (16). Later Cockburn and his colleagues extended the DG method to the RKDG method by using Runge-Kutta method for temporal discretization (3), (4), (7), (8). Their extension was crucial to solving the nonlinear time-dependent hyperbolic partial differential equations.

An error analysis for the RKDG method can be found in the papers by Zhang and Shu (27), (28), where they have presented an error analysis under the $L^2$-norm using the traditional \textit{a priori} error analysis approach. Since the numerical scheme is employed to estimate error propagation, their error estimate contains an exponentially growing coefficient, hence are only meaningful as a convergence result, not as a practical error estimation.

The error analysis of RKDG methods has been well known to be extremely difficult, due to the nonlinearity of the conservation laws, the lack of smoothness of the solutions and the complexity of the scheme. As a consequence, very few results have been published besides the papers of Zhang and Shu (27), (28) until Sun and his students introduced the idea of “numerical smoothness” to the error analysis of the RKDG methods in (23) and (22). The advantages of this numerical smoothness approach are obvious. (1) The error estimate is of optimal order. (2) The error propagation is done by the $L^1$-contraction, also optimal. (3) It can be done for nonsmooth solutions without losing the two kinds of optimality mentioned above.

This dissertation is built on the foundation of the idea of “numerical smoothness” presented in (25), the error analysis for the RKDG method on the scalar nonlinear conservation law for the case of smooth solution presented in (23), and the error analysis of Front Tracking-RKDG method on a scalar conservation law for the case of having a fully developed shock presented in (22). In this
dissertation, the focus is on the gap left by the two results of (23) and (22). Namely, we will develop
an algorithm to compute a solution of a nonlinear conservation law during shock formation. More
precisely, we focus on the part of the solution during its transition from a smooth solution to a fully
developed shock.

In addition to the development of the algorithm, we also estimate the error of the solution in
$L^1$-norm. As in (23) and (22), we also prove the error estimate to be optimal in convergence rate
as well as in error propagation. This has never been done before.

Shock formation refers to the transition from a smooth solution to a discontinuous solution. Our
innovative algorithm to compute the shock formation is to track the so called “s-shape” solution by
using piecewise cubic polynomials. By combining this algorithm with the numerical methods in
(23) and (22), we can compute shock solutions with optimal error analysis. We will discuss more
about the “s-shape” solution of a conservation law later in this chapter. Also, more explanation
about the “s-shape” solution can be found in (15), (26).

When the solution computed by the RKDG method is close to the occurrence of a shock,
high order accuracy will be lost. Thus, there is no hope to obtain the high order solution by the
RKDG method during further sharpening and the formation of a shock. To tackle this difficulty, we
incorporate the Cubic Front Tracking scheme to evolve the steep portion of the solution computed
by the RKDG method. We will discuss in detail the method later in this chapter. Once the solution
forms a fully developed “s-shape,” we locate the shock using the equal-area principle (15). After
the shock has been located, our solution is in the case of having a fully developed shock.

During the transition time where the solution lies between the cases of (23) and (22), we split
the whole domain into three regions. The first and third regions contain the DG cells and the second
region contains the front tracking cells. During the formation of the “s-shape” solution, the solution
curve becomes multivalued as a function of $x$ and hence it is impossible to do error analysis for a
whole domain at once. Thus, we do error analysis in three different regions separately. However,
one we locate the shock using the equal-area principle (26), our solution will be in the case of
a fully developed shock (22). More explanation about the equal-area principle will be presented
later in this chapter.

During the formation of a shock, the first and third regions contain the smooth pieces of the solution and hence the proofs from the case of smooth solutions \((20)\) applies verbatim in our case as well. We just outline the ideas of proofs and refer the readers to \((20)\) for detail.

In this dissertation, FT means front tracking, DG means discontinuous Galerkin, RKDG means Runge-Kutta discontinuous Galerkin, PDE stands for a partial differential equation, ODE stands for an ordinary differential equation. Moreover, CFL condition means Courant Fridrich and Lax condition on time step size.

To illustrate our work effectively, we split this dissertation into eight chapters. The first chapter contains a detailed introduction of the one-dimensional scalar nonlinear conservation law and the Cubic Front Tracking and Runge-Kutta Discontinuous Galerkin (RKDG) method.

Chapter 2 contains some overview of methods and techniques which are used in error analysis. Chapter 3 includes a brief description of our algorithm to compute the solution during the shock formation. Moreover, smoothness indicators and estimation of the transition time are also included in Chapter 3.

Chapter 4 describes an error estimation of the Cubic Front Tracking solution. Chapter 5 and Chapter 6 includes the error estimation of the DG solution. Spatial error estimation will be focused in Chapter 5, and temporal error estimation will be focused in Chapter 6. Furthermore, shock location error will be presented in Chapter 6. Numerical experiments that support our error analysis will take place in Chapter 7. Finally, a summary of the results will be discussed in Chapter 8.

1.1 The Scalar Hyperbolic Conservation Law

We consider a hyperbolic partial differential equation of the form

\[
  u_t + f(u)_x = 0.
\]  

1.1.1 (1.1)

Here, \(u\) is a conserved quantity such as mass or energy and \(f(u)\) is the flux of the conserved quantity. We call \((1.1)\) a scalar hyperbolic conservation law. We take the upwind boundary condition
\[ u(0, t) = u_a(t) \]  

(1.2)

and the initial condition

\[ u(x, 0) = u_0(x), \]  

(1.3)

where \( \Omega = [a, b] \) is a bounded domain. If the flux function \( f(u) \) is nonlinear (linear), then (1.1) will be a nonlinear (linear) conservation law. In this dissertation, we consider \( f(u) \) to be nonlinear and also for simplicity, we assume that \( f(u) \) is a convex function, \( f''(u) > 0 \) for all \( u \). A simple example of a nonlinear hyperbolic conservation law is Burgers’ equation,

\[ u_t + uu_x = 0 \]  

(1.4)

which is obtained by substituting \( f(u) = \frac{u^2}{2} \) in (1.1). Some other examples can be found in (15). Later, for our numerical experiment, we will consider Burgers’ equation. Note that

\[
\frac{d}{dt} \int_{\alpha}^{\beta} u(x, t)dx = \int_{\alpha}^{\beta} u_t(x, t)dx \\
= -\int_{\alpha}^{\beta} f(u(x, t))_x dx \\
= f(u(\alpha, t)) - f(u(\beta, t))
\]

where \( \alpha, \beta \in [a, b] \) with \( \alpha < \beta \). This shows that amount of mass \( u \), inside the sub-domain \( [\alpha, \beta] \), can be changed only from the boundary points. Here, \( f(u(\alpha, t)) \) and \( f(u(\beta, t)) \) are inflow at \( \alpha \) and outflow at \( \beta \), respectively. One of the interesting characters of a nonlinear conservation law is that even though the initial data is smooth (continuously differentiable), smoothness can be lost as time evolves.
1.2 The Cubic Front Tracking and RKDG Method

The Discontinuous Galerkin (DG) method is a finite element method. In 1973, this method was introduced by Reed and Hill to solve the hyperbolic neutron transport equation \(\text{(16)}\). For a given partition of a space domain, in each sub-interval the solution is approximated by a smooth function, normally by a polynomial. Since for the different sub-intervals, different functions (polynomials) are allowed to approximate the solution, the method is called the Discontinuous Galerkin method.

A family of Runge-Kutta (RK) methods were introduced around 1900 by the German mathematicians C. Runge and M. W. Kutta. The combination of Runge-Kutta scheme and Discontinuous Galerkin Method is called the RKDG method. Besides this dissertation, detailed explanations of the RKDG method can be found in \(\text{(7)}\), \(\text{(8)}\). First, the front tracking method was proposed by Dafermos in the 1970s \(\text{(9)}\). His proposal was for initial value problems for a nonlinear scalar conservation law.

In 2013, Sun and Rumsey analyzed the error of the RKDG method approximating the solutions of nonlinear conservation laws for the case of a smooth solution \(\text{(23)}\). The smooth solution of scalar nonlinear conservation law is shown in Figure 1.1. After a couple of years, Sun and Adamou used the RKDG method in combination with Rankine-Hugoniot front tracking techniques. This DG-FT-RK method computes the solution of a nonlinear conservation law for the case of having a fully developed shock \(\text{(22)}\). The solution of a nonlinear conservation law for the case of having a fully developed shock is shown in Figure 1.2. The error analysis presented in both papers \(\text{(23)}\) and \(\text{(22)}\), is an \textit{a posteriori} error analysis. In this dissertation, we consider the case between \(\text{(23)}\) and \(\text{(22)}\).

Now we will write the Cubic Front Tracking and RKDG method that is used in this dissertation.
The following paragraph has detailed explanations of the DG method.

We partition the domain $\Omega = [a, b]$ into $a = x_{-\frac{1}{2}} < x_{\frac{1}{2}} < ... < x_{m-\frac{1}{2}} = b$. For convenience, we take the uniform mesh size $h = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ and $\Omega_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, $0 \leq j \leq m - 1$, where $\Omega_j$ is called a cell. Let

$$V_h = \{v : v|_{\Omega_j} \in \Pi_p, 0 \leq j \leq m - 1\}$$

be the finite element space consisting of piecewise polynomials up to degree $p$ in each cell $\Omega_j$. The semi-discrete DG scheme to solve $u_t + f(u)_x = 0$ for the case of a smooth solution is to find the unique function $u_h = u_h(t) \in \Pi_p$ for all test function $v \in \Pi_p$ and $0 \leq j \leq m - 1$ such that

$$(u_{h,t}, v)_{\Omega_j} = (f(u_h), v_x)_{\Omega_j} + f(u_h(x_{j-\frac{1}{2}}))v(x_{j-\frac{1}{2}}^+) - f(u_h(x_{j+\frac{1}{2}}))v(x_{j+\frac{1}{2}}^-),$$

(1.5)

where $u_{h,t}(x_{-\frac{1}{2}}, t) = u(a, t)$.

In the above equation (1.5), the Gudnov flux is used. However, one can use other numerical flux functions such as the Lax-Fridrich flux. When one use the Lax-Fridrichs’ flux the semi-discrete solution $u_h$ will satisfy

$$(u_{h,t}, v)_{\Omega_j} = (f(u_h), v_x)_{\Omega_j}$$

$$+ \frac{1}{2} \left\{ f(u_h(x_{j-\frac{1}{2}}^)) + f(u_h(x_{j-\frac{1}{2}}^-)) - C \left[ u_h(x_{j-\frac{1}{2}}^+ - u_h(x_{j-\frac{1}{2}}^-) \right] v(x_{j+\frac{1}{2}}^+)$$

$$- \frac{1}{2} \left\{ f(u_h(x_{j-\frac{1}{2}}^)) + f(u_h(x_{j-\frac{1}{2}}^-)) - C \left[ u_h(x_{j-\frac{1}{2}}^+ - u_h(x_{j-\frac{1}{2}}^-) \right] v(x_{j+\frac{1}{2}}^-)$$

Where, $C = \max_u |f'(u)|$ and the maximum is over the interval $[f(x_{j-\frac{1}{2}}), f(x_{j+\frac{1}{2}})]$. More
detailed comparison between the Lax-Fridrichs’ flux and the Gudnov flux can be found in (12).

At time $t = 0$, the initial condition of the semi-discrete solution $u_h(x, 0)$ is obtained by taking the $L_2$-projection of the initial solution $u(x, 0)$ onto the finite element space $V_h$.

To obtain a fully discrete scheme, we use the Total Variation Diminishing Runge-Kutta (TVD-RK) method of order three on the semi-discrete system at each time step $[t_n, t_{n+1}]$. Suppose we have a computed solution $u^c_n$ at time $t_n$, then we will compute $u^c_{n+1} \in \Pi_p$ from known $u^c_n$ in following three steps:

\begin{align*}
(u^c_n, v)_{\Omega_j} &= (u^c_n, v)_{\Omega_j} + \tau_n \mathcal{H}_j(u^c_n, v) \\
(u^c_{n+1}, v)_{\Omega_j} &= \frac{1}{3} (u^c_n, v)_{\Omega_j} + \frac{1}{4} (u^c_{n+1}, v)_{\Omega_j} + \frac{\tau_n}{4} \mathcal{H}_j(u^c_{n+1}, v) \\
(u^c_{n+1}, v)_{\Omega_j} &= \frac{1}{3} (u^c_n, v)_{\Omega_j} + \frac{2}{3} (u^c_{n+2}, v)_{\Omega_j} + \frac{2\tau_n}{3} \mathcal{H}_j(u^c_{n+2}, v)
\end{align*}

where

\begin{equation}
\mathcal{H}_j(u, v) = (f(u), v_x)_{\Omega_j} + f(u(x^{-}_{j-1/2}))v(x^{-}_{j-1/2}) - f(u(x^{+}_{j+1/2}))v(x^{+}_{j+1/2}).
\end{equation}

In all of the above equations $v \in V_h$ and $\tau_n = t_{n+1} - t_n$ must satisfy a strengthened CFL condition

$$\tau_n \leq \gamma h^{1+1/p},$$

where $p = 3$ is a degree of the polynomial in each cell $\Omega_j$. The detailed information for $p \geq 3$ can be found in (23).

Now, we want to introduce the front tracking scheme. Suppose we have a location at any point $x_n$ at time $t_n$, then the front tracking scheme computes the location at the next time step $t_{n+1}$ by
the equation
\[ x_{n+1} = x_n + \tau_n f'(u(x_n, t_n)). \] (1.10)

Since the solution along the characteristic line remains constant, \( f'(u(x_n, t_n)) = f'(u_0) \), where \( u_0 \) is the initial solution. The front tracking method is an efficient numerical method even if the initial solution is not smooth. Moreover, for this method, we do not need to impose any condition, such as the CFL condition, for the time step size. The detail about the CFL condition can be found in (15).

Starting from a smooth initial condition function \( u(x, 0) = u_0(x) \), we compute the solution by using the standard RKDG method. For a pre-determined constant \( M > 0 \), our transition time starts when the slope of the computed solution becomes less than \(-M\) at some point of \( \Omega \). At the beginning of the transition time, in each cell of the Front Tracking region \( \Omega_T \), we approximate the solution by the cubic polynomial generated by \( (u - u_l), (u - u_r), (u - u_l)(u - u_r) \), and \( (u - u_l)(u - u_r)(u - u_c) \), where \( u_l \) and \( u_r \), are u-values of left and right endpoints of each cell and \( u_c \) is the average between them. Then, we use the front tracking method to evolve the piecewise cubic polynomial. Since piecewise cubic polynomials are used to track, we call this scheme the Cubic Front Tracking scheme.

Recall that by the formation of a shock, we mean the solution gets further sharpening and is going to be a fully developed “s-shape” soon, but the shock has not been located yet. The “s-shape” solution appears until we can cut it into two lobes with equal areas. Possible solutions of the conservation law at different time stages have shown in Figure 1.3. Moreover, a possible solution of the conservation law after locating the shock has shown in Figure 1.4. To show the equal area principle in a figure, we take a picture from reference (10) and present it in Figure 1.5. In Figure 1.5, the areas of the shaded part A and part B are equal.

Our goal is to estimate the error during the formation of a shock. For this, we want to compute a solution during the shock formation. In Figure 1.3, the last three sub-figures represent the possible solutions during the shock formation.

Starting from a smooth initial condition function \( u(x, 0) = u_0(x) \), we compute the solution
Figure 1.3: Solution at the Different Time Stages

Figure 1.4: Fully Developed “s-shape” Solution After Locating the Shock

Figure 1.5: “s-shape” Solution Satisfying the Equal Area Principle
by using the standard RKDG method. The implementation and the error analysis of the standard RKDG method is given in (23). When the computed solution has sharpened to a pre-determined slope, we switch to the second stage. For a pre-determined slope $M > 0$, we assume that the slope of $u_0(x)$ is bounded by $-M$ from below: $u_0'(x) \geq -M$ for all $x \in \Omega$. The transition stage takes place at the first time when the slope of the computed solution becomes less than $-M$ at some point of $\Omega$. A shock will occur near this point soon. The time period after the transition starts and before the solution gets a fully developed “s-shape” is called the transition time. A detailed algorithm will be given later, where the domain $\Omega$ is split into three regions, $\Omega_U$, $\Omega_T$ and $\Omega_D$. Here, $\Omega_T$ is the region where the solution will be approximated by the Cubic Front Tracking scheme. The region $\Omega_U$ is upstream of the $\Omega_T$ and the region $\Omega_D$ is downstream of the $\Omega_T$. In $\Omega_U$ and $\Omega_D$, the solution can be computed by the RKDG method as usual in (23). Since we use both the RKDG and Cubic Front Tracking methods, we called this combination the Cubic Front Tracking and RKDG method.

The “s-shape” solution of a conservation law has a cut such that the area in the top and bottom lobes are equal, which is called an equal area principle. The figure representing the equal area principle was shown in Figure 1.5. After the shock is being located, the “s-shape” solution of a conservation law which satisfy the equal area principle is an entropy solution. The mathematical proof of the equal area principle can be found in (26). Also, an alternative proof can be found in (11).

We locate the shock in such a way that areas of the top and bottom lobes of the “s-shape” solution are equal. Once we locate the shock, the solution belongs to the case of a fully developed shock. The detail explanation on how to solve a fully developed shock with RKDG and front tracking can be found in (22).

In the next chapter, we will briefly discuss some ideas used to estimate the error in the past. Also, we will point out why such ideas are not sufficient to get an optimal error bound for our problem.
CHAPTER 2 PRELIMINARIES AND LITERATURE REVIEW

In this chapter, we give an overview of the approaches and techniques to do error analysis in the past and present. Many error analysis methods were developed in the last century, and some recently developed methods are also in practice. We will justify why these methods are not adequate for error analysis of the complex numerical schemes.

In this dissertation, we are considering a one-dimensional scalar nonlinear conservation law

$$u_t + f(u)_x = 0$$  \tag{2.1}

where $u$ is the conserved quantity and $f(u)$ is the flux of the conserved quantity. There are many real-life problems which are modeled by (2.1). The shallow water problems and the traffic flow problems are the examples which are modeled as a nonlinear conservation law. The following paragraph will elaborate the traffic flow problems and we refer to (15) for the detailed explanation of the shallow water problems.

Consider the flow of cars on the road. If $\rho$ and $u$ are used to denote the density of cars and the velocity of cars, respectively; then

$$\rho_t + f(\rho)_x = 0,$$

where $f(\rho) = \rho u_{\text{max}}(1 - \frac{\rho}{\rho_{\text{max}}})$ will model for the traffic flow problems. In this example $\rho$ and $u$ is restricted to a certain range $0 \leq \rho \leq \rho_{\text{max}}$, where $\rho_{\text{max}}$ is the maximum density of cars. Similarly, $0 \leq u \leq u_{\text{max}}$ where $u_{\text{max}}$ is the maximum speed of cars. The relation between $\rho$ and $u$ can be modeled by

$$u(\rho) = u_{\text{max}} \left( 1 - \frac{\rho}{\rho_{\text{max}}} \right).$$  \tag{2.2}

If the density $\rho = 0$ (empty road), cars can have maximum speed $u = u_{\text{max}}$ and at maximum density $\rho = \rho_{\text{max}}$, cars should be stop $u = 0$. These both are included in (2.2).

In general, due to lack of analytic solutions, PDE type (2.1) need to be solved by the numerical
scheme. To find a numerical solution of such problems is not an easy task, especially when the initial data is not smooth or smoothness is lost as time evolves. The solutions of a conservation law are not always smooth and hence, (2.1) will not be satisfied by the solution in the classical sense for every point in the domain. Thus, we need to study about the weak solution. For this, we multiply by a test function $\phi$, and integrate both sides of the above equation (2.1), to get
\[
\int_{0}^{+\infty} \int_{-\infty}^{+\infty} [u \phi_t + f(u) \phi_x] \, dx \, dt = - \int_{-\infty}^{+\infty} u(x,0) \phi(x,0) \, dx
\]
for all $\phi \in C^1_0(\mathbb{R} \times \mathbb{R}_+)$, where $C^1_0(\mathbb{R} \times \mathbb{R}_+)$ is the space of all continuously differentiable functions with compact support.

Equation (2.1), may have more than one weak solution, but in reality, we should have only one physical solution. Therefore, we need a condition that ensures we avoid all non-physical solutions. We call such a condition an entropy condition. Thus, a weak solution which satisfies the entropy condition is a physical solution. More detail can be found in (15).

We split this chapter into two sections. The first section discusses the general framework for error analysis of both PDEs (Partial Differential Equations) and ODEs (Ordinary Differential Equations) and the second section contains the literature review.

2.1 General Framework for Error Analysis of Time Dependent Differential Equations

In the literature of error analysis for time-dependent differential equations (ODEs and PDEs), mostly there are two types of error estimates; a priori error estimate and a posteriori error estimate. To have better understanding of these estimates, we begin our discussion with Figure 2.1 and Figure 2.2.

Assume we are solving a time-dependent ODE initial value problem for which $y(t)$ is a true solution. Here, the parameter $t$ represents ‘time’. Let $y_n^c$ denote the numerical solution approximating $y(t_n)$ and $\tilde{y}(t_n)$ be the unknown true solution passing through $y_n^c$. We denote $\tilde{y}$ be the solution obtained by the numerical scheme with initial value $y(t_n)$ (see Figure 2.1). Note that only $y_n^c, y_{n+1}^c$ and $y_{n+2}^c$ shown in Figure 2.1 and Figure 2.2 are computable. In a priori error analysis,
Figure 2.1: Error Splitting in Numerical Stability

Figure 2.2: Error Splitting in Numerical Smoothness
which can be found in most numerical analysis textbooks, the local error will be measured by how much the numerical scheme misses the original solution in one time step size. In Figure 2.1,

\[ y(t_{n+1}) - \tilde{y}_{n+1} \]

is the local error at time interval \([t_n, t_{n+1}]\). To measure this local error, smoothness of the true solution \(y(t)\) is used. The propagation error in Figure 2.1 is

\[ \tilde{y}_{n+1} - y^c_{n+1} \]

at the time \(t = t_{n+1}\). The error splitting shown in Figure 2.1 is sometimes also called numerical stability and it is used in \textit{a priori} error estimates. \textit{In a priori} error estimates, error propagation is done by the numerical scheme.

For the case of PDEs, a point on a solution curve in the figures represents a spatial solution function at that time. For PDEs, we have two parts of the local error, namely, the local spatial error and the local temporal error. Thus, for PDE cases, the estimation of a solution may be difficult when the spatial solution has bigger jumps on the cell boundaries.

Even though there are many forms of spatial and temporal local errors that are defined in textbooks, if we saw that the smoothness of the ODE or PDE’s exact solution is used to measure the local error, then the error analysis is an \textit{a priori} error analysis. The error splitting for such error analysis is shown in Figure 2.1. The main disadvantage of such an error analysis is on its numerical error propagation, known as numerical stability. This happens because such an \textit{a priori} error analysis depends on the von Neumann stability condition on the scheme itself. In this estimate, a researcher tries to prove

\[ ||\tilde{y}_{n+1} - y^c_{n+1}|| \leq |1 + \beta \tau_n| ||y(t_n) - y^c_n|| \]

for some suitable norm \(|| \cdot ||\) and a constant \(\beta\). Here, \(\tau_n = t_{n+1} - t_n\) is the time step size and \(\beta\)
depends on the differential equation as well as the numerical scheme. Unfortunately, due to the complexity of the schemes and nonlinearity of the problem, $\beta$ is often really large, and one ends up with a factor in the error bound, $e^{\beta T}$, where $T$ is the total time. Consequently, this type of error estimate becomes impractical.

On the other hand, we have the approach of error analysis as shown in Figure 2.2, where the local error has come along with the numerical solution and error propagation is done by the differential equation itself. Here, the local error at $[t_n, t_{n+1}]$ is

$$\bar{y}(t_{n+1}) - y_{n+1}^c$$

and the propagated error is

$$\bar{y}(t_{n+1}) - y(t_{n+1}).$$

Thus, in this setting as shown in Figure 2.2, error propagation is done by the differential equation, and the local error depends on the numerical solution $y_{n}^c$. Note that $a$ posteriori error analysis in the literature belongs to the framework shown in Figure 2.2, where the estimation of the local error relies on the so called error indicators of various kinds. However, if the estimation of the local error is built on the smoothness of $\bar{y}(t)$ in certain sense, we are getting into the idea of “numerical smoothness”. Some more explanation about the numerical smoothness will be presented later. Also, detailed explanations about numerical smoothness can be found in (24).

From our discussion above, we have an understanding that if we want to take advantage of the smoothness of the exact solution of the original ODE or PDE, we should study the error propagation done by the numerical scheme. On the other hand, if we refer the local error to the numerical solution, error propagation is done by the original PDE or ODE.

It turns out that the second approach, the numerical smoothness approach which uses the error splitting as in Figure 2.2 is the better approach for the error analysis of nonlinear time-dependent PDEs. In Figure 2.2, at $t = t_n$, $y_n^c$ is computed but $y(t_n)$ is unknown. The best way to compute $y_{n+1}^c$ is to follow the true solution passing through $y_n^c$. To measure the local error at $[t_n, t_{n+1}]$, under
the error splitting of Figure 2.2, we should study the smoothness of the true solution curve passing through \(y_n^c\). However, for stiff ODE systems and PDEs, it has not been done until (23). In fact, how to use the smoothness of the true solution curve passing through \(y_n^c\) was first investigated in the paper (23). For the PDE cases, most of the spatial approximations of the true solution are technically not smooth functions. This might be a reason why researchers did not study about the smoothness of the true solution curve passing through \(y_n^c\) for a long time. Moreover, we believe that most of the research done in the past was based on numerical error propagation. Also, the textbooks have been focusing on the idea of numerical stability. Thus, the idea of error splitting shown in Figure 2.2 has not been popular yet.

Before considering some important points about numerical smoothness, let us discuss the Lax Equivalence Theorem. The famous Lax Equivalence Theorem is often referred to as the fundamental theorem of numerical analysis. Originally it was about the numerical solutions of the time-dependent differential equations. It is stated as:

**Theorem 2.4.** (15) *If a linear differential equation is solved with a linear numerical scheme, and the scheme is consistent with the equation, then the scheme is convergent if and only if it is stable.*

Over the years, this specific theorem has been thought of as a general paradigm. However, the convergence result in the Lax Equivalence Theorem also depends on the smoothness of a solution of the differential equation to estimate local error, in addition to the consistency and the stability of the scheme. For the hyperbolic conservation law, a linear equation

\[
 u_t + cu_x = 0
\]

can have a discontinuous solution. Hence the “paradigm” mentioned above does not seem to fit, even in the linear case. Consider the initial condition

\[
 u_0 (x) = \begin{cases} 
 1 & x < 0 \\
 0 & x > 0 
\end{cases}
\]
We know that our exact solution is \( u_0(x - ct) \). However, numerical methods have a problem near the discontinuity. As an example, if we consider a finite difference method to approximate \( u_x \), it is approximately

\[
\frac{u(ct + h, t) - u(ct - h, t)}{2h} = -\frac{1}{2h} \rightarrow -\infty
\]

as \( h \rightarrow 0 \). Thus, the local error does not vanish as \( h \rightarrow 0 \). This trouble was created by the discontinuity of the initial data.

We already discussed that even for a linear problem, the sufficiency part of the Lax Equivalence Theorem is not applicable when the initial solution is discontinuous. For a nonlinear equation \( u_t + f(u)_x = 0 \), a smooth initial function can evolve to a shock, and the condition on the smoothness of the solution is harder to be satisfied. Therefore, the Lax stability framework does not work for the hyperbolic nonlinear conservation laws.

The bigger problem is actually error propagation. The Lax Equivalence Theorem does not cover error propagation for nonlinear problems. Moreover, even for linear problems, the error propagation in the Lax framework is done by numerical schemes. We already discussed that for complex schemes and nonlinear differential equations, numerical error propagation leads to an exponential coefficient, which is typically huge. As its effect, for practically used and affordable discretization sizes, the exponential coefficient usually shadows or even swallows convergence.

We emphasized that if the numerical schemes propagate the error, for the complex schemes an error analysis can be impractical. We prefer the framework of Figure 2.2, where error propagation is done by the exact solution of the PDE and the local error is linked to the numerical solution. This approach is called the numerical smoothness approach. We believe that this approach is the best for error analysis of the Cubic Front Tracking and RKDG method described in Chapter 1, and hence, this approach is the backbone for our error analysis in this dissertation.

Before concluding the section, we point out some important points about this new approach.

- The classical concept of numerical stability believes that stable schemes do not amplify the error by too much, but actually it is too much to allow an exponential growth, especially
when the growth depends heavily on the complexity of a scheme. In the new numerical smoothness concept, there will be no position for numerical error propagation. The error propagation is done only by the differential equation.

- PDE error propagation is always better than numerical error propagation. If a numerical scheme has a better error propagation rate than the PDE’s error propagation rate, then such a numerical scheme cannot be a convergent scheme.

- If we do not have enough numerical smoothing in a numerical scheme, we will produce a large local error at each step.

- If the computed smoothness indicators (to be defined later) are too large, then the computed numerical solution is not converging to any expected smooth function at the desired optimal rate. In (24), this is referred as the necessity of numerical smoothness.

- The boundedness of the smoothness indicators is required for the convergence (24). In fact, computing smoothness indicators is a way to overcome the difficulties of nonlinearity arising in the proof of any global property of a numerical scheme.

- The numerical smoothness approach allows for both \textit{a priori} and \textit{a posteriori} error analysis in one framework. If one can prove the boundedness of the smoothness indicators before computation, then the estimate is \textit{a priori}. Otherwise, one can always compute the smoothness indicators and perform an \textit{a posteriori} error analysis.

2.2 Literature Review

The \textit{a posteriori} error analysis idea was introduced in early 1900s and a detailed description of the development of \textit{a posteriori} error analysis can be found in (17). In the literature, researchers have been using error indicators to do \textit{a posteriori} error analysis. Cockburn, Johnson and Szepessy in (3), (14) introduced a technique for error analysis called the duality method to hyperbolic problems. In their idea, they represent error in terms of PDE solutions and incorporate it with the solution of so-called “adjoint” problem. If they were able to solve the corresponding “adjoint”
problem, then they can accomplish the error analysis. This idea is applied to reasonably simple schemes. Even though it is a famous *a posteriori* error analysis techniques, for the complex scheme and for a nonlinear problem, this approach is too costly and difficult to implement.

In the literature of the numerical solutions of conservation laws, most of the significant work of error analysis is summarized in LeVeque’s famous textbook (15). The book contains error analysis based on numerical stability of finite difference and finite volume schemes. One important theorem is the Lax-Wendroff theorem, which states that the computed solution will converge to a weak solution when the numerical method is consistent and conservative. Also, the relation between monotone methods, $L_1$-contraction schemes, and total variation diminishing (TVD) schemes is illustrated in (15).

Detailed explanation of numerical schemes for finite difference and finite volume methods to solve the scalar nonlinear conservation law can be found in (15). Moreover, the classical treatments for either overshooting oscillation or a smeared front have been done using artificial viscosity or slope limiters (15).

A Discontinuous Galerkin (DG) method is a finite element method, which was introduced by Reed and Hill to solve the hyperbolic neutron transport equation (16). The detailed survey of DG method for a linear problem can be found in (6).

For a nonlinear hyperbolic conservation law, first, Chavent and Salzano used the discontinuous Galerkin method for space discretization in (1). Their nonlinear hyperbolic conservation law was in the context of oil recovery problems. Intense study of the DG method has been done by Cockburn and his colleagues later in the series of papers (3), (4), (5), (7), (8). More literature about the Runge-Kutta Discontinuous Galerkin method can be found in the survey article (6) by Cockburn and Shu.

The first front tracking method was proposed by Dafermos in the 1970s (9). In 1992, Risebro and Tveito presented a front tracking scheme for one-dimensional conservation laws where the solution is represented by piecewise constant states separated by discontinuities, which is to approximate the Riemann solutions (18).
The error analysis for the DG method for conservation laws remained vacant for a relatively long time. In 2004, Shu and Zhang presented an *a priori* error analysis using $L_2$-norm for one-dimensional nonlinear conservation laws on the RKDG method (28). It is a crucial step in the field of error analysis of the RKDG method. In this paper, they used second-order explicit total variation diminishing (TVD) Runge Kutta method for sufficiently smooth solutions. Later in 2010, the same authors extended the result by using third order explicit total variation diminishing (TVD) Runge Kutta method for sufficiently smooth solutions (27). They used upwind fluxes to achieve the optimal order. It was another significant work in the field of error analysis of a one-dimensional conservation law. However, their results were based on von Neumann stability condition and thus contained an exponential factor in the error bound.

In 2013, Sun and Rumsey (23) published an error analysis for the RKDG method of one-dimensional scalar nonlinear conservation laws for the case of smooth solutions. Their error analysis was based on the idea of “numerical smoothness” and was innovative in this field. This paper (23), is crucial in the sense that the idea of “numerical smoothness” allowed them to achieve an optimal convergence rate as well as linear error growth for the first time in the literature. A couple of years later, Sun and Fode extended the result for the case of having a fully developed shock (22). They incorporate the front tracking techniques together with the RKDG method. These papers (23) and (22) are the starting point for this dissertation research. Since (23) dealt with smooth solutions and (22) dealt with fully developed shocks, what is missing between the two papers is the shock formation stage of a solution. We propose a cubic front tracking technique to approximate the shock formation in this work, while still using the RKDG method for the smooth pieces of solutions.
CHAPTER 3 MAIN ALGORITHM, SMOOTHNESS INDICATORS, AND ESTIMATION OF THE TRANSITION TIME

This chapter includes the brief description of our algorithm to compute the solution during the shock formation and the error analysis of the Cubic Front Tracking and RKDG method. Moreover, estimation of the transition time will be presented.

We assume throughout the dissertation that all the function notations that are written as a function of $x$ and $t$ are restricted to the DG-region and all the function notations that are written as a function of $u$ and $t$ are restricted to the FT-region.

Before presenting the algorithm to compute the fully developed “s-shape” solution of a conservation law (1.1), we will define some notations in the following Table 3.1. We use $u^c_n$ to denote the computed DG solution and $x_{ct}(u, t_n)$ for the computed cubic front tracking solution at the time $t = t_n$. The semi-discrete solution is denoted by $u^h(x, t)$. At the time of conversion, $\tilde{x}(u)$ is used to denote the inverse function of $u^c_n(x)$. Moreover, $x_t(u, t)$ is used to denote the characteristic solution in the tracking part of the solution.

3.1 Main Algorithm

Assume that at $t = T$, we do have a matured “s-shape” solution. We use the approximated transition time presented later in this chapter to compute $t = T$. To obtain the fully developed

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u(x, t)$</td>
<td>PDE’s entropy solution</td>
</tr>
<tr>
<td>$u^c_n(x, t)$</td>
<td>Computed solution at time $t = t_n$</td>
</tr>
<tr>
<td>$x_s(t)$</td>
<td>The shock location of $u(x, t)$</td>
</tr>
<tr>
<td>$u^h(x, t)$</td>
<td>Semi-discrete solution passing through $u^c_n(x, t)$</td>
</tr>
<tr>
<td>$\tilde{x}_s(t)$</td>
<td>The shock location of the computed solution</td>
</tr>
<tr>
<td>$\tilde{x}(u)$</td>
<td>Inverse function of $u^c_n$ at the time of conversion</td>
</tr>
<tr>
<td>$x_t(u, t)$</td>
<td>The exact cubic front tracking or the characteristic solution at time $t$</td>
</tr>
<tr>
<td>$x_{ct}(u, t_n)$</td>
<td>The computed cubic front tracking solution at time $t = t_n$</td>
</tr>
</tbody>
</table>
"s-shape" solution of the conservation law (1.1), we follow the following procedure:

1. For a pre-determined constant $M > 0$, use the RKDG scheme as in (23) to compute $u_n^c$ when $\frac{d}{dx}u_n^c(x) > -M$ for all $x \in \Omega$.

2. When $\frac{d}{dx}u_n^c(x) \leq -M$ for some $x \in \Omega_{im} \in \Omega$. Use approximate transition time (present later in this chapter) and the front tracking scheme to locate the mid point of $\Omega_{im}$ at the time of fully matured “s-shape” solution $t = T$. Let $x_{mid}(j)$ be the mid-point of the cell $\Omega_j$ at time $t = T$. Then the front tracking part $\Omega_T = \bigcup_{i_t}^{i_t} \Omega_j$, (is obtained by back tracing) where $i_t \leq i_m$ such that $x_{mid}(i_t + 1) > x_{mid}(i_m)$ but $x_{mid}(i_t) \leq x_{mid}(i_m)$ and $i_b \geq i_m$ such that $x_{mid}(i_b - 1) < x_{mid}(i_m)$ but $x_{mid}(i_b) \geq x_{mid}(i_m)$.

3. After the front tracking part $\Omega_T$ is determined, split $\Omega$ into three parts from the left side of the domain $\Omega$ to the right. First, the upstream part $\Omega_U = \bigcup_{j=0}^{i_t} \Omega_j$. Then the front tracking part $\Omega_T = \bigcup_{j=i_t}^{i_b} \Omega_j$, and the downstream part $\Omega_D = \bigcup_{j=i_b+1}^{m-1} \Omega_j$.

4. Once the splitting is done, for each cell $\Omega_j$, $j = i_t, i_t + 1, \ldots, i_b$ approximate $\tilde{x}(u)$, the inverse of $u_n^c$, by a piecewise cubic polynomial

$$x_c(u) = x_l \frac{(u - u_r)}{(u_l - u_r)} + x_r \frac{(u - u_l)}{(u_r - u_l)} + x_q(u - u_l)(u - u_r) + x_m(u - u_l)(u - u_r)(u - u_c),$$

where $u_l = u_n^c(x_{j-\frac{1}{2}})$ and $u_r = u_n^c(x_{j+\frac{1}{2}})$ are $u$-values at the left and right endpoints of the cell $\Omega_j$ and $u_c = (u_l + u_r)/2$. For this piecewise approximation, we call the conversion. We compute the polynomial $x_c(u)$ using the conditions

$$x_c(u_l) = \tilde{x}(u_l) = x_l = x_{j-\frac{1}{2}},$$

$$x_c(u_r) = \tilde{x}(u_r) = x_r = x_{j+\frac{1}{2}},$$

$$\int_{u_l}^{u_r} (x_c(u) - x_l) \, du = \int_{u_l}^{u_r} (\tilde{x}(u) - x_l) \, du = \int_{x_l}^{x_r} (u_n^c(x) - u_r) \, dx,$$
and

\[ x'_c(u_c) = \tilde{x}'(u_c). \]

5. After the conversion, we compute the Cubic Front Tracking solution \( x_{ct}(u, t) \) from \( x_c(u) \) by using the following interpolation conditions. For each time step interval \([t_n, t]\) during the front tracking,

\[
\begin{align*}
x_{ct}(u_l, t) &= x_{ct}(u_l, t_n) + (t - t_n) f'(u_l), \\
x_{ct}(u_r, t) &= x_{ct}(u_r, t_n) + (t - t_n) f'(u_r), \\
x_{ct}(u_c, t) &= x_{ct}(u_c, t_n) + (t - t_n) f'(u_c),
\end{align*}
\]

and

\[
\int_{u_r}^{u_l} x_{ct}(u, t) \, du = \int_{u_r}^{u_l} x_{ct}(u, t_n) \, du + (t - t_n) \int_{u_r}^{u_l} f'(u) \, du.
\]

6. For any transition time \( t = t_n \), for \( x \in \Omega_U \cup \Omega_D \), use the RKDG scheme as in (23) and for \( x \in \Omega_T \), use the Cubic Front Tracking scheme to get the solution for the new time step \( t = t_{n+1} \). The left boundary condition for the DG solution on \( \Omega_D \) is obtained by solving

\[ x_{ct}(u) - x_{i_b} - \frac{1}{2} = 0. \]

7. At the beginning of the tracking time, we start with the overlapping of two solutions \( x_c(u) \) and \( u_{n+1}^c \) on the cell \( \Omega_{i_0} \). As time evolves, the DG solution moves upward and the Cubic Front tracking solution moves forward and they will detached each other in upstream part and overlapped in downstream part.

At time \( t = t_{n+1} \), on \( \Omega_U \) if the DG solution and FT solution detached each other, set \( i_t = i_{t+1} \).

To fill in the gap resulting from the detachment, a new DG cell should be constructed as follows:

First extrapolate \( x_{ct}(u) \) to the left end of the missing part. Use \( \hat{u}(x, t_{n+1}) \) to denote the inverse of the extrapolated \( x_{ct}(u) \). Compute the cubic polynomial \( u_{n+1}^c(x) \) from \( \hat{u}(x, t_{n+1}) \).
using the following interpolation conditions:

\[ u_{n+1}^c(x_l) = \hat{u}(x_l), \]

\[ u_{n+1}^c(x_r) = \hat{u}(x_r), \]

\[ \frac{d}{dx} u_{n+1}^c \left( \frac{x_l + x_r}{2} \right) = \frac{d}{dx} \hat{u} \left( \frac{x_l + x_r}{2} \right), \]

\[ \int_{x_l}^{x_r} (\hat{u}(x) - u_r) \, dx = \int_{x_l}^{x_r} (u_{n+1}^c(x) - u_r) \, dx. \]

8. At \( t = t_{n+1}, \) on \( \Omega_D \) if the DG solution and FT solution overlapped on the whole cell \( \Omega_{ib}, \) set \( i_b = i_{b+1}. \) This means the DG solution \( u_{n+1}^c \) is omitted when the overlapping occurred in the full cell \( \Omega_{ib}. \)

9. Repeat (5)-(8), until \( t = T. \)

10. Shock Location Algorithm: At \( t = T, \) first we join the top-left end point and the bottom-right end point of the FT-solution to get the line

\[ X_L(u) = x_{ct}(u_t^c) \left( \frac{u - u_r^c}{u_t^c - u_r^c} \right) + x_{ct}(u_r^c) \left( \frac{u - u_l^c}{u_r^c - u_l^c} \right), \]

where \( u_t^c \) and \( u_r^c \) are the top and the bottom values of the shock line. Then, the shock location \( \tilde{x}_s, \) will satisfy the following system of equations:

\[ F_1 = \int_{u_r^c}^{u_t^c} (x_{ct}(u) - X_L(u)) \, du = 0 \]

\[ F_2 = x_{ct}(u_t^c) - x_{ct}(u_r^c) = 0. \]

For computation, we use the Newton’s method.
3.2 The Error Splitting

This section will focus on the error analysis of the RKDG and Cubic Front Tracking method. Recall that in the error estimation of a time step of a time-dependent problem, there are two parts of the total error, namely, the local error and the propagated error. We already discussed in Chapter 2 that there are two types of error splitting, as in Figure 2.1 and Figure 2.2. In this dissertation, we are considering \textit{a posteriori} error analysis, where error propagation is done by the PDE’s solution and the local error is linked to the numerical solution (see Figure 2.2). Thus, the PDE’s exact solution is used to estimate the propagated error. In our case, to measure the propagated error, we use the $L_1$-\textit{Contraction Property} of the scalar conservation law (1.1) (21). In order to estimate the local error, we use the smoothness indicators which are defined later in this chapter.

Note that the number of the DG cells in the first region $\Omega_U$ are increasing during the front tracking time, meanwhile the number of the DG cells are decreasing in the third region $\Omega_D$. Each initial front tracking cell is also considered as a front tracking segment when the solution is viewed as a function of $x$. Although the numerical solution in the tracking cells evolve to become a multi-valued function of $x$, it remains to be a single valued function of $u$ in all the tracking segments. Because the front tracking part of the solution originally in $\Omega_T$ becomes multi-valued as a function of $x$, we estimate its error as a function of $u$.

We discussed earlier that to estimate the local error, we use the idea of numerical smoothness. In order to define numerical smoothness rigorously, we will define the computed smoothness indicators later in this chapter. If those smoothness indicators are bounded, we call the computed solution numerically smooth. Due to the nonlinearity of the PDE $u_t + f(u)_x = 0$, and the complexity of the RKDG method, it is hard to establish an \textit{a priori} upper bound for the smoothness indicators. However, the numerical evidences in (23), (22) and later in this dissertation have shown that they do remain bounded in the computation. In fact, the boundedness of the computed smoothness indicators is a necessary condition to have the numerical solution approximating the true solution in the optimal convergence rate (25).

During the transition time, in the DG-region $\Omega_U \cup \Omega_D$, we use the idea of error splitting describe
The global error \( u(t_{n+1}) - u_{n+1}^c \) at time \( t_{n+1} \) is split into the following three parts:

\[
\| u(t_{n+1}) - u_{n+1}^c \| \leq \| u(t_{n+1}) - \tilde{u}(t_{n+1}) \| + \| \tilde{u}(t_{n+1}) - u_{n+1}^h \| + \| u_{n+1}^h - u_{n+1}^c \|. \tag{3.1}
\]

The first part \( u(t_{n+1}) - \tilde{u}(t_{n+1}) \) of the right-hand side of (3.1) is the propagation of \( u(t_n) - u_n^c \). By \( L_1 \)-contraction, we have

\[
\| u(t_{n+1}) - \tilde{u}(t_{n+1}) \| \leq \| u(t_n) - u_n^c \|.
\]

Here, we need to use \( \| \cdot \|_{L_1} \) norm as the contraction property is valid for \( \| \cdot \|_{L_1} \) norm (21).

The second part of the error is \( \tilde{u}(t_{n+1}) - u_{n+1}^h \) is a local spatial discretization error. Notice that \( u_n^c \) is an element of the discontinuous finite element space. So, it is not a smooth function in the traditional sense. Thus, the DG solution \( \tilde{u} \) contains a technical discontinuity on each cell boundary point at \( t_n \) which evolves to a shock, a contact discontinuity or a rarefaction wave. However, for the smooth solution it should not contain any big jump to maintain the numerical smoothness. To deal with this local spatial discretization error, a spatial smoothness indicator \( S_p^n \) will be defined later. Chapter 5 will cover the estimation of this part of the error.

The third part of the error is \( u_{n+1}^h - u_{n+1}^c \) is a local temporal discretization error. To estimate this part of the error, temporal smoothness indicator \( T_n^k \) is defined and then, the temporal
smoothness of \( u^h \) for \( t \in [t_n, t_{n+1}] \) is used. Chapter 6 will cover all the details to estimate this part of the error.

The conversion of \( u^c_n(x) \) to the cubic polynomial \( x_c(u) \) has been described earlier. The conversion error will be estimated in Chapter 4, and added to the global error estimate. The error due to the reconstruction of the new DG cells in the first region \( \Omega_U \) will be estimated in Chapter 5. Finally, the error of the computed shock location will be estimated in Chapter 6.

Since the idea of numerical smoothness is important to estimate our local error (part II and part III of the splitting), we need to define numerical smoothness rigorously.

3.3 Smoothness Indicators

To define numerical smoothness for the RKDG method rigorously, we need spatial and temporal smoothness indicators for each time step \([t_n, t_{n+1}]\):

- Spatial smoothness indicator: \( S^p_n \)
- Temporal smoothness indicator: \( T^k_n \)

In our case, we take the degree of the polynomial to be \( p = 3 \) and the order of Runge-Kutta scheme to be \( k = 3 \). The notation \( S \) and \( T \) stand for space and time respectively.

3.3.1 Spatial Smoothness Indicators

As in (23), we define

\[
S^p_{n,j} = (M^0_{n,j}, M^1_{n,j}, ..., M^p_{n,j}, D^0_{n,j}, ..., D^p_{n,j}),
\]

where

\[
M^l_{n,j} = \frac{\partial^l}{\partial x^l} u^c_n(x_{j-\frac{1}{2}}^+), \quad L^l_{n,j} = \frac{\partial^l}{\partial x^l} u^c_n(x_{j-\frac{1}{2}}^-),
\]

and

\[
J^l_{n,j} = M^l_{n,j} - L^l_{n,j} = D^l_{n,j} h^{p+2-l(1+\alpha)}
\]
for $\alpha = 1/3$ and $l = 0, 1, 2, 3$. The values $M_{n,j}^l$ is used to measure the smoothness of $u_n^c$ in the interior of the cell $\Omega_j$. So, in the interior of each cell $\Omega_j$, the spatial smoothness indicators $S_n^p$ contains the spatial derivatives of the computed solution $u_n^c$. Moreover, the jump $J_{n,j}^l$ carries the information of the smoothness of $u_n^c$ at each cell boundary point $x_{j-1/2}$. However, we rescaled the jumps of the derivatives to obtain the $D_{n,j}^l$ and include them in the indicator $S_n^p$. In fact, the boundedness of these $D_{n,j}^l$ is the necessary and sufficient condition for optimal convergence.

The detail about the indicator components at the upstream boundary point can be found in (23). Since they are not essential to the research of this dissertation, we omit the details to keep the section concise.

3.3.2 Temporal Smoothness Indicators

As in (23), the temporal smoothness indicator $T_n^k$ consists of temporal derivatives of the semi-discrete solution $u_n^h$ at $t_n$. Namely,

$$T_n^k = \left( u_n^c, u_t^h(t_n), u_{tt}^h(t_n), \ldots, \frac{\partial^{k+1}}{\partial t^{k+1}} u_n^h(t_n) \right),$$

where the first derivative $u_t^h(t_n)$ is computed as in the implementation of the forward Euler scheme

$$(u_t^h, v)_{\Omega_j} = (f(u_n^h), v_x)_{\Omega_j} + f(u_n^h(x_{j-1/2}^-))v(x_{j-1/2}^+) - f(u_n^h(x_{j+1/2}^-))v(x_{j+1/2}^-). \quad (3.2)$$

Now differentiating both sides with respect to $t$, we get

$$\begin{align*}
(u_{tt}^h, v)_{\Omega_j} &= (f'(u_n^h)u_t^h, v_x)_{\Omega_j} \\
&\quad + f'(u_n^h(x_{j-1/2}^-))u_t^h(x_{j-1/2}^-)v(x_{j-1/2}^+) \\
&\quad - f'(u_n^h(x_{j+1/2}^-))u_t^h(x_{j+1/2}^-)v(x_{j+1/2}^-),
\end{align*}$$

where variable $t_n$ is hiding. Now to compute $u_{tt}^h(t_n)$ from the above formula, we replace $u_n^h$ by $u_n^c$ and $u_t^h$ by $u_t^h(t_n)$. The temporal differentiation of the above equation gives the formula for $u_{ttt}^h(t_n)$. The pattern can be repeated to obtain all the components of $T_n^k$. 
The indicator $T^n_k$ reveals the smoothness of the numerical solution, the discontinuities or possible numerical oscillation. However, for the purpose of error estimation, it is only applied to the estimates of the temporal discretization error.

3.4 Estimation of the Transition Time

In this section, we describe how to determine the transition time. The transition time denotes the time interval where we perform the Cubic Front Tracking. It starts when the DG solution has self-sharpened sufficiently and presents large local error, and it ends when the Cubic Front Tracking has produced a mature “s-shape” solution to locate a fully developed shock.

For simplicity, we split the transition time into the following two phases, separated roughly by an estimated shock occurrence time.

1. **Transition time phase I**: We denote the time interval between the time of conversion to tracking and the breaking time by the transition time phase I. The length of this phase is denoted by $T_{r1}$.

2. **Transition time phase II**: We denote the time interval between the breaking time and the time of having a mature “s-shape” solution by the transition time phase II. The length of this phase is denoted by $T_{r2}$.

If we know that the shock is going to form, then the following lemma tells us explicitly the breaking time $T_b$.

**Lemma 3.3.** If we solve the one-dimensional conservation of law $u_t + f(u)_x = 0$, with smooth initial profile $u_0(x)$ for which if $u_1'(x) < 0$ at some point $x$, then there must be a discontinuity. Furthermore, if $u'(x) < 0$, for all $x \in [a, b]$ and $f''(u) > 0$ then, the breaking time $T_b$ is

$$
T_b = \min_{x \in [a, b]} \frac{-1}{f''(u_0(x))u_0'(x)}.
$$

**Proof.** Let us consider the equation $u_t + f(u)_x = 0$. The slope of the characteristic line is given by $f'(u)$. Then, equations of two characteristic lines passing through $(x_1, 0)$ and $(x_2, 0)$ for $x_1 < x_2$
$x_2, x_1, x_2 \in [a, b]$ are respectively;

$$x = x_1 + f'(u_0(x_1, 0))t$$ \hspace{1cm} (3.4)

and

$$x = x_2 + f'(u_0(x_2, 0))t. \hspace{1cm} (3.5)$$

When the breaking occurs, above equations must intersect to each other. Thus,

$$x_1 + f'(u_0(x_1, 0))t = x_2 + f'(u_0(x_2, 0))t.$$ 

Solving for $t$ gives us

$$t = \frac{-(x_2 - x_1)}{f'(u_0(x_2, 0)) - f'(u_0(x_1, 0))} = \frac{-1}{f'(u_0(x_2, 0)) - f'(u_0(x_1, 0)) \left( \frac{u_0(x_2, 0) - u_0(x_1, 0)}{x_2 - x_1} \right)}$$

which yields that the breaking time

$$T_b = \min_{x \in [a, b]} \frac{-1}{f''(u_0(x))u'_0(x)}.$$ 

Remarks: In particular, the breaking time $T_b$ for Burgers’ equation is

$$\min_{x \in [a, b]} \frac{-1}{u'_0(x)}.$$ 

Since we are interested to do the error analysis during the transition time, the first important task is to know the period of the transition time $T_r$. Let $T$ denote the final time when we have a mature “s-shape” solution. That means, $T = T_b + T_r$.
Theorem 3.6. Consider Burgers’ equation $u_t + uu_x = 0$. If there is a shock appearance, then the length of the transition time phase II ($T_{r2}$) is approximately $\frac{1}{2M}$, where $M$ is a chosen positive constant such that $|u'(x_0, T)| = M > |u_0'(x)|$ for all $x \in \Omega$ and some $x_0 \in \Omega$.

Remark: We choose a positive constant $M$ such that when $|u'(x)| \geq M$ for some $x \in \Omega$, our transition time starts. Since we start with the smooth initial solution, it is trivial that $|u_0'(x)| < M$ for all $x \in \Omega$.

Proof. Let $(x^*, u^*)$ be the breaking point at the breaking time $t = T_b$. Assume $(x_0, u^0)$ be the point on a solution curve $u(x, t)$ such that $|u'(x_0, T)| = M$. Note that at $t = T_b$, we have $x_0 < x^*$. The picture for the possible situation of the point $(x_0, u^0)$ is illustrated in Figure 3.2.

Now, consider $x$ as a function of $u$ and use the Taylor expansion about $u = u^*$, we get

$$x(u) = x(u^*) + x'(u^*)(u - u^*) + \frac{x''(u^*)(u - u^*)^2}{2!} + \frac{x'''(u^*)(u - u^*)^3}{3!} + \mathcal{O}(u - u^*)^4 \quad (3.7)$$

Here, $x'$ denotes for $\frac{dx}{du}$. Since $(x^*, u^*)$ is the inflection point of the curve, $x''(u^*) = 0$. Also, it is easy to see $x'(u^*) = 0$. Substituting $u = u^0$, in (3.7), we get

$$x(u^0) = x(u^*) + \frac{x'''(u^*)(u^0 - u^*)^3}{3!} + \mathcal{O}(u^0 - u^*)^4.$$
Now tracing back to \( x \) values and neglecting the high order terms, we get

\[
x_0 \approx x^* + \frac{x'''(u^*)(u^0 - u^*)^3}{6}.
\]

Note that \( x'''(u^*) < 0 \). Thus, we can write

\[
x_0 = x^* - c(u^0 - u^*)^3
\]

where \( c \approx \frac{-x'''(u^*)}{6} \). Notice that for \( t \in [T_b, T_b + Tr_2] \), the equation

\[
x = x_0 + (t - T_b)f'(u)
\]

(3.9)

determines the current position of \( x_0 \). Now, using Taylor series expansion of \( f'(u) \) about \( u = u^* \), we get

\[
f'(u) = f'(u^*) + f''(u^*)(u - u^*) + \frac{f'''(u^*)(u - u^*)}{2!} + \frac{f''''(u^*)(u - u^*)^3}{3!} + O(u - u^*)^4.
\]

In the case of Burgers’ equation, it reduces to

\[
f'(u) = f'(u^*) + (u - u^*).
\]

Now, (3.9) at \( u = u^0 \), reduces to

\[
x = x_0 + (t - T_b)f'(u^*) + (t - T_b)(u^0 - u^*).
\]

At \( t = T = T_b + Tr_2 \), the shock location \( x = x_s \). Hence, from the above equation,

\[
x_s = x_0 + Tr_2f'(u^*) + Tr_2(u^0 - u^*).
\]
Using (3.8), we get

\[ x_s = x^* + Tr_2 f'(u^*) - c(u^0 - u^*)^3 + Tr_2 (u^0 - u^*). \]  \hspace{1cm} (3.10)

Note that the shock location can be determined by

\[ x_s = x^* + Tr_2 f'(u^*). \]  \hspace{1cm} (3.11)

Combining (3.10) and (3.11), we get

\[ Tr_2 = c(u^0 - u^*)^2. \]

Also, from (3.8), we see

\[ x'(u^0) = -3c(u^0 - u^*)^2. \]

Hence, we get the result

\[ u'(x_0, T_b) = \frac{-1}{3Tr_2} \]  \hspace{1cm} (3.12)

On the other hand, differentiating the equation of conservation of law (1.1) with respect to \( x \), we get,

\[ u_{tx} + (f'(u))_x.u_x + f'(u).u_{xx} = 0 \]

by writing \( v = u_x \) and \( f(u) = \frac{u^2}{2} \) we get,

\[ v_t + u.v_x = -v^2 \]

Now, using the characteristic derivative yields

\[ \frac{Dv}{dt} = -v^2. \]
or, \( \frac{Dv}{v^2} = -dt \).

Then, solving this ordinary differential equation, with initial time at \( t = T_b \), we get

\[
v(t) = \frac{v(T_b)}{1 + Tr_2v(T_b)}.
\] (3.13)

At \( t = T_b + Tr_2 \), we have

\[
u'(x_0, T_b + Tr_2) = \frac{u'(x_0, T_b)}{1 + Tr_2u'(x_0, T_b)}.
\] (3.14)

Now, substituting the value of \( u'(x_0, T_b) \), from (3.12) to (3.14), we get

\[
Tr_2 = \frac{-1}{2u'(x_0, T_b + Tr_2)} = \frac{1}{2M}.
\]

\( \square \)

**Remarks:** Theorem 3.6 for a general conservation law \( u_t + f(u)_x = 0 \), will be valid with

\[
Tr_2 = \frac{-1}{2f''(u)u'(x)}
\]

at \( (x_0, T_b + Tr_2) \) in the sense of approximation. The proof is similar as in the case of Burgers’ equation but (3.12) will be replaced by

\[
f''(u)u'(x_0, T_b) = \frac{-1}{3Tr_2}.
\]

**Lemma 3.15.** The ratio between the length of the transition time phase I and phase II is 2 : 1.

**Proof.** From Lemma 3.1,

\[
T_b = \min_{x \in [a, b]} \frac{-1}{f''(u_0)u_0'(x)}.
\]
Also, from the extended version of Theorem 3.6

\[ Tr_2 = \frac{-1}{2f''(u)u'(x_0)} \]  
\[ (3.16) \]

If we consider the initial profile at the time when

\[ \min_{x \in [a,b]} u'(x) = -M, \]  
\[ (3.17) \]

then use the formula of \( T_b \) and the fact \( |u'(x_0)| = M \), yields

\[ Tr_1 = \frac{-1}{f''(u)u'(x_0)}. \]

Hence, the basic calculation yields \( Tr_1 : Tr_2 = 2 : 1. \)
CHAPTER 4 ERROR ESTIMATION ON THE FRONT TRACKING REGION

This chapter includes the error estimation of the second region $\Omega_T$ where we use the Cubic Front Tracking scheme during the transition time. Let $u_L$ be the $u$-value of the computed solution at the left end point of the first tracking cell and $u_R$ be the $u$-value of the computed solution at the right end point of the last tracking cell. Since the solution along the characteristic line remains constant, the values $u_L$ and $u_R$ remain the same throughout the transition period. Thus, $\Lambda = [u_R, u_L]$ is fixed at any time of the transition period.

4.1 Conversion Error Estimation

For a pre-determined constant $M > 0$, transition time takes the place when there is a cell $\Omega_{i_m}$, such that $|\tilde{u}'(x)| \geq M$, for some $x \in \Omega_{i_m}$ and for some $i_m \in \{0, 1, ..., m - 1\}$. Numerical experiment presented in Chapter 7 shows that for the high order estimates, the pre-determined constant $M \leq 5$. Recall that we consider the “steep curve” on the cells $\Omega_j, j = i_t, i_t + 1, ..., i_m, ..., i_b$. We use $\tilde{x}(u)$ to denote the inverse function of computed solution $\tilde{u}(x, t_n) = u_c(x)$ as a function of $u$ on the cells $\Omega_j, j = i_t, i_t + 1, ..., i_m, ..., i_b$. Let $u_l$ and $u_r$ are $u$-values at the left and right end points of a cells $\Omega_j$ and $u_c = \frac{u_l + u_r}{2}$. The interval $[u_r, u_l]$ is referred as a tracking segment, which remains invariant throughout the tracking process.

Let $x_c(u)$ be the computed cubic polynomial in $u$ spanned by the bases $(u - u_l), (u - u_r), (u - u_l)(u - u_r)$, and $(u - u_l)(u - u_r)(u - u_c)$. At the beginning of the transition time, first we approximate the solution $\tilde{x}(u)$ by $x_c(u)$ on the cells $\Omega_j, j = i_t, i_t + 1, ..., i_m, ..., i_b$ in each tracking segment. We call this computation of $x_c(u)$ from $u_c(x)$ the conversion. After the conversion is done, we use the Cubic Front Tracking scheme to evolve the converted solution in each tracking segment.

We compute the polynomial $x_c(u)$ from $\tilde{x}(u)$ using the interpolation conditions

$$x_c(u_l) = \tilde{x}(u_l) = x_{j-1/2},$$
\begin{align*}
x_c(u_r) = \bar{x}(u_r) = x_{j+1/2},
\int_{u_r}^{u_l} x_c(u) du = \int_{u_r}^{u_l} \bar{x}(u) du,
\end{align*}
and
\begin{align*}
x'_c(u_c) = \bar{x}'(u_c).
\end{align*}

Theorem 4.1, illustrate the conversion error estimate.

**Theorem 4.1.** There is a computable constant \( C_v \), such that
\begin{align}
||| \bar{x}(u) - x_c(u) |||_{L_\infty(\Lambda)} \leq C_v h^4
\end{align}

**Remark:** The rest of this section is applied to prove Theorem 4.1.

First, to estimate the error \( |\bar{x}(u) - x_c(u)| \) in each sub-interval \([u_r, u_l]\), we use the triangle inequality
\begin{align}
|\bar{x}(u) - x_c(u)| \leq |\bar{x}(u) - x_{cc}(u)| + |x_{cc}(u) - x_c(u)|,
\end{align}
where \( x_{cc}(u) \) is a cubic polynomial spanned by the same bases \((u - u_l), (u - u_r), (u - u_l)(u - u_r), (u - u_l)(u - u_r)(u - u_c)\) and the constants are determined by the following four conditions:

\begin{align*}
x_{cc}(u_l) = \bar{x}(u_l) = x_{j-1/2},
\end{align*}
\begin{align*}
x_{cc}(u_r) = \bar{x}(u_r) = x_{j+1/2},
\end{align*}
\begin{align*}
x_{cc}(u_c) = \bar{x}(u_c)
\end{align*}
and
\begin{align*}
x_{cc}'(u_c) = \bar{x}'(u_c).
\end{align*}

It is easy to see that both polynomials \( x_c(u) \) and \( x_{cc}(u) \) have the same constant coefficients except
in the quadratic term. Thus,

\[ x_c(u) - x_{cc}(u) = (x_q - x_{qq})(u - u_l)(u - u_r), \]

where \( x_q \) and \( x_{qq} \) are coefficients of the quadratic term of \( x_c(u) \) and \( x_{cc}(u) \) respectively.

**Lemma 4.4.** There exists some \( \xi \in (u_r, u_l) \) such that

\[ \ddot{x}(u) - x_{cc}(u) = \frac{x^{(4)}(\xi)}{4!}(u - u_l)(u - u_r)^2 \] (4.5)

**Proof.** Consider any \( u \in (u_r, u_c) \cup (u_c, u_l) \). Define a real valued function \( g : \mathbb{R} \rightarrow \mathbb{R} \), defined by

\[ g(\alpha) = \ddot{x}(\alpha) - x_{cc}(\alpha) - [\ddot{x}(u) - x_{cc}(u)] \left( \frac{\alpha - u_l}{u - u_l} \right) \left( \frac{\alpha - u_r}{u - u_r} \right)^2. \]

Then \( g \) is continuously differentiable on \([u_r, u_l]\). Note that \( g(u_l) = g(u_r) = g(u_c) = 0 \). Also, \( g(u) = \ddot{x}(u) - x_{cc}(u) - [\ddot{x}(u) - x_{cc}(u)] = 0 \). Thus, \( g \) has at least four zeros in \([u_r, u_l]\). Then by the application of Rolle’s theorem \( g' \) has at least three zeros in \((u_r, u_l)\) except at \( u_c \). Now, by taking the derivative of \( g(\alpha) \) and evaluate at \( u_c \), it is easy to see \( g'(u_c) = 0 \). Thus, \( g' \) has also four zeros in \([u_r, u_l]\). Now, by the repeated application of Rolle’s theorem there exists \( \xi \in (u_r, u_l) \), such that \( g^{(4)}(\xi) = 0 \). Finally, by using the fact \( x_{cc}^{(4)}(u) = 0 \), one gets (4.5). \( \square \)

**Remark:** By Lemma 4.4, we have \[ |\ddot{x}(u) - x_{cc}(u)| \leq \frac{(u_l - u_u)^4}{96} |\ddot{x}^{(4)}(\xi)|. \]

Now we are ready to estimate the error between \( x_c(u) \) and \( x_{cc}(u) \). Again, from Lemma 4.4 we have

\[ \ddot{x}(u) - x_{cc}(u) = \frac{\ddot{x}^{(4)}(\xi)}{4!}(u - u_l)(u - u_r)^2. \]

Integrating both sides from \( u_r \) to \( u_l \) we get, for some different but generic \( \xi \),

\[ \int_{u_r}^{u_t} \ddot{x}(u)du = \int_{u_r}^{u_t} x_{cc}(u)du + \frac{\ddot{x}^{(4)}(\xi)}{4!} \int_{u_r}^{u_t} (u - u_l)(u - u_r)(u - u_c)^2 du. \]
Now, using the condition $\int_{u_r}^{u_l} \bar{x}(u) du = \int_{u_r}^{u_l} x_c(u) du$ we get,

$$(x_{qq} - x_q) \int_{u_r}^{u_l} (u - u_l)(u - u_r) du = \frac{\bar{x}(4)(\xi)}{4!} \int_{u_r}^{u_l} (u - u_l)(u - u_r)(u - u_c)^2 du.$$

An elementary calculation gives $|x_{qq} - x_q| \leq \frac{\bar{x}(4)(\xi)}{480} (h_1)^2$, where $h_1 = u_l - u_r$. Consequently,

$$|x_{cc}(u) - x_c(u)| \leq \frac{|\bar{x}(4)(\xi)|}{1920} (h_1)^4.$$

Now, using (4.3) we get,

$$|\bar{x}(u) - x_c(u)| \leq |\bar{x}(u) - x_{cc}(u)| + |x_{cc}(u) - x_c(u)| \leq C|\bar{x}(4)(\xi)| (h_1)^4$$

where $C = \frac{21}{1920}$. Now, we need to established the bound on $|\bar{x}(4)(\xi)|$.

For this, let $u = u_0$ be an arbitrary point on $[u_r, u_l]$ such that $\bar{u}(x_0) = u_0$ for some $x_0 \in \Omega_j$. Then, $\bar{x}(u_0) = \bar{u}^{-1}(\bar{u}(x_0))$.

By using the Inverse Function Theorem, $|\bar{x}'(u_0)| = \left| \frac{1}{\bar{u}'(x_0)} \right|$. Since $|\bar{u}'(x_0)|$ is bounded above and also bounded away from zero, $|x'(u_0)|$ has the same property.

Now, computing high order derivatives by the quotient rule

$$\frac{\partial^2}{\partial u^2} \left[ \bar{x}(u) \right]_{u=u_0} = \frac{\partial}{\partial u} \left[ \bar{x}'(u) \right]_{u=u_0}$$

$$= \frac{\partial}{\partial x} \left[ \bar{x}'(u) \right]_{u=u_0} \left[ \frac{\partial x}{\partial u} \right]_{x=x_0}$$

$$= \frac{\partial}{\partial x} \left[ \frac{1}{\bar{u}'(x)} \right]_{x=x_0} \frac{\partial}{\partial u} \left[ \bar{x}(\bar{u}(x_0)) \right]$$

$$= -\bar{u}''(x_0) \frac{1}{(\bar{u}'(x_0))^2} \bar{u}'(x_0)$$

$$\frac{-\bar{u}''(x_0)}{(\bar{u}'(x_0))^3} \frac{1}{\bar{u}'(x_0)}$$

$$= \frac{-\bar{u}''(x_0)}{(\bar{u}'(x_0))^3}$$
Similarly, we compute

\[
\frac{\partial^3}{\partial u^3} \left[ \tilde{x}(u) \right]_{u=u_0} = \frac{\partial}{\partial u} \left[ \frac{\partial^2}{\partial u^2} \tilde{x}(u) \right]_{u=u_0} = \frac{\partial}{\partial x} \left[ \frac{-\tilde{u}''(x)}{(\tilde{u}'(x))^3} \right]_{x=x_0} \frac{\partial x}{\partial u} \bigg|_{x=x_0} = -\left[ \frac{\tilde{u}'''(x_0)(\tilde{u}'(x_0))^3 - 3(\tilde{u}'(x_0))^2(\tilde{u}''(x_0))^2}{(\tilde{u}'(x_0))^6} \right] \frac{1}{\tilde{u}'(x_0)} 
\]

\[
= 3\tilde{u}'''(x_0)^2 - \tilde{u}''''(x_0)\tilde{u}'(x_0) \quad \frac{1}{(\tilde{u}'(x_0))^5}
\]

Finally,

\[
\frac{\partial^4}{\partial u^4} \left[ \tilde{x}(u) \right]_{u=u_0} = \frac{\partial}{\partial u} \left[ \frac{\partial^3}{\partial u^3} \tilde{x}(u) \right]_{u=u_0} = \frac{\partial}{\partial x} \left[ \frac{3(\tilde{u}''(x))^2 - \tilde{u}''''(x)\tilde{u}'(x)}{(\tilde{u}'(x))^5} \right]_{x=x_0} \frac{\partial x}{\partial u} \bigg|_{x=x_0} = \frac{5(\tilde{u}'(x_0))^4\tilde{u}''''(x_0) 
- 10\tilde{u}'''(x_0)\tilde{u}''(x_0)\tilde{u}'(x_0) - 15(\tilde{u}''(x_0))^3 - \tilde{u}'''(x_0)(\tilde{u}'(x_0))^2}{(\tilde{u}'(x_0))^7} 
\]

Since \( u_0 \) and \( x_0 \) are arbitrary points, we have

\[
\frac{\partial^4}{\partial u^4} \left[ \tilde{x}(u) \right] = \frac{10\tilde{u}'''(x)\tilde{u}''(x)\tilde{u}'(x) - 15(\tilde{u}''(x))^3}{(\tilde{u}'(x))^7}. \quad (4.6)
\]

Note that each term on the numerator of the right hand side of \((4.6)\) is bounded and the denominator
is bounded away from zero. Thus, there exists a positive constant $M_4$ such that

$$|\tilde{x}^{(4)}(u)| \leq M_4$$

for all $u \in [u_r, u_l]$. In particular,

$$|\tilde{x}^{(4)}(\xi)| \leq M_4.$$  

Consequently, we have

$$||\tilde{x}(u) - x_c(u)||_{L_\infty(\Lambda)} \leq (h_1)^4 CM_4. \quad (4.7)$$

Note that $h_1 \leq Mh$ for a pre-determined positive constant $M$ for which $|\tilde{u}(x, t_n)| \leq M$ holds at the time of conversion. Thus, we can write

$$||\tilde{x}(u) - x_c(u)||_{L_\infty(\Lambda)} \leq T_1 M_4 h^4 \quad (4.8)$$

for $T_1 = \frac{21}{1920} M^4$. This completes the proof of Theorem 4.1.

**4.2 Error Estimation of the Cubic Front Tracking Solution**

After the conversion made on the cells $\Omega_j$ ($j = i_t, i_t + 1, ..., i_m, ..., i_b$), we use the Cubic Front Tracking scheme to evolve the converted solution in each tracking segment $[u_r, u_l]$. Recall that we use to denote $x_{ct}(u, t)$ for the computed tracking solution and $x_t(u, t)$ for the characteristic solution. At the time of conversion, we have a solution $x_c(u)$ in each cell $\Omega_j$ ($j = i_t, i_t + 1, ..., i_m, ..., i_b$), and the estimate

$$|\tilde{x}(u) - x_c(u)| \leq T_1 h^4 M_4. \quad (4.9)$$

If $t_n$ is the time of conversion, the characteristic solution in a tracking segment is

$$x_t(u, t) = x_c(u, t_n) + (t - t_n)f'(u). \quad (4.10)$$
As for the Cubic Front Tracking solution \( x_{ct}(u, t) \), we use the following interpolation conditions.

For each time step interval \([t_n, t]\) during the front tracking,

\[
x_{ct}(u_l, t) = x_{ct}(u_l, t_n) + (t - t_n)f'(u_l),
\]
\[
x_{ct}(u_r, t) = x_{ct}(u_r, t_n) + (t - t_n)f'(u_r),
\]
\[
x_{ct}(u_c, t) = x_{ct}(u_c, t_n) + (t - t_n)f'(u_c),
\]

and

\[
\int_{u_l}^{u_r} x_{ct}(u, t) \, du = \int_{u_l}^{u_r} x_{ct}(u, t_n) \, du + (t - t_n)\int_{u_l}^{u_r} f'(u) \, du.
\]

Differentiating both side of (4.10) with respect to \( u \), we get

\[
\frac{\partial}{\partial u} x_t(u, t) = \frac{\partial}{\partial u} x_c(u, t_n) + (t - t_n)f''(u).
\] (4.11)

Differentiating three times on both side of above equation with respect to \( u \), we get

\[
\frac{\partial^4}{\partial u^4} x_t(u, t) = \frac{\partial^4}{\partial u^4} x_c(u, t_n) + (t - t_n)f^{(5)}(u) = (t - t_n)f^{(5)}(u)
\] (4.12)

Using (4.12), and the standard interpolation error analysis, the error estimation at time \( t \), is given by

\[
|x_t(u, t) - x_{ct}(u, t)| \leq T_1 h^4 \left| \frac{\partial^4}{\partial u^4} x_t(\xi, t) \right| = T_1 h^4 (t - t_n) |f^{(5)}(\xi)|
\]

for a computable positive constant \( T_1 \). Hence, the error bound for the Cubic Front Tracking scheme is \( T_1 h^4 (t - t_n) |f^{(5)}(u)| \). Notice that \(|f^{(5)}(u)|\) is bounded, so there exists a positive constant \( M_5 \) such that \(|f^{(5)}(u)| \leq M_5\).

However, if the flux function \( f(u) \) is a polynomial of degree up to 4, then the constant \( M_5 = 0 \).

Consequently, the Cubic Front Tracking scheme is exact for the polynomial flux up to the degree 4. Thus, we prove the following theorem.
**Theorem 4.13.** For the general flux function \( f(u) \), there is a computable constant \( C_t \) such that the local error of the cubic front tracking solution has the estimate

\[
\|x_t(u, t) - x_{ct}(u, t)\|_{L_{\infty}(\Lambda)} \leq C_t(t - t_n)h^4.
\] (4.14)

Moreover, if the flux function \( f(u) \) is a polynomial of degree up to 4, then the cubic front tracking solution is exact.
CHAPTER 5 THE SPATIAL ERROR ESTIMATION ON THE DG REGION

Theorem 5.1 is the main result for the error analysis on the first region $\Omega_U$. Since the first region $\Omega_U$ contains the smooth piece of the solution, the error estimate is almost same as the main error estimate presented in (23). However, the error from the reconstruction of a new DG cell is added in our case.

**Theorem 5.1.** Let $u(x, t)$ be the entropy solution of the nonlinear conservation law $u_t + f(u)_x = 0$ satisfying initial condition $u(x, 0) = u_I(x)$ and upwind boundary condition $u(a, t) = u_L(t)$. As described in Chapter 1, let $u^c_n$ be the numerical solution computed by the TVD-RKDG scheme with piecewise polynomials of degree $p = 3$ and the TVD-RK scheme of order $k = 3$, on the partition of $\Omega_U$. Assume that $u$ and $u^c_n$ are bounded by a constant $U$ in $\Omega \times [0, T]$. Let $\beta = \max_{|w| \leq U} |f'(w)|$.

Assume that the time step size $\tau_n$ satisfies the standard CFL condition and the strengthened CFL condition $\tau_n \leq \gamma h^{1+\alpha}$, for a positive constant $\gamma$ and $\alpha = \frac{1}{3}$.

If there is a real number $M_1$, such that, for all $t_n \leq T$, all the components of $S^n_p$ and $T^n_k$ are bounded by $M_1$, then the spatial and temporal local error in $[t_n, t_{n+1}]$ satisfy

$$||\tilde{u}(t_{n+1}) - u^h(t_{n+1})||_{L_1(\Omega_U)} \leq \tau_n h^4 F_1(S^n_p) + [h_{rc}(n)] C_p h^4,$$

(5.2)

$$||u^h(t_{n+1}) - u^c_{n+1}||_{L_1(\Omega_U)} \leq \tau_n^3 G_1(T^n_k, S^n_p),$$

(5.3)

where $F_1(S^n_p)$ and $G_1(T^n_k, S^n_p)$ are computable functions of the indicators and

$$h_{rc}(n) = \begin{cases} h & \text{if reconstruction of new DG cell is required in } [t_n, t_{n+1}] \\ 0 & \text{otherwise} \end{cases}.$$

As a consequence of the above estimates and the use of the $L_1$-contraction property one can see

$$||u(t_{n+1}) - u^c_{n+1}||_{L_1(\Omega_U)} \leq ||u(t_n) - u^c_n||_{L_1(\Omega_U)} + \tau_n [h^4 F_1(S^n_p) + \tau_n^3 G_1(T^n_k, S^n_p)] + [h_{rc}(n)] C_p h^4.$$
The proof of Theorem 5.1 is split into Section 5.1 and Section 6.1.

5.1 Spatial Error Estimation on the First Region $\Omega_U$

This section includes the spatial error estimation for the first region $\Omega_U$. Note that during the transition time the number of the cells are increasing in this region. The error analysis for regular DG cells can be found in (20). To make convenience for the readers, we will summarize the main points here.

We have introduced some notations in previous chapters. We assume throughout the dissertation $\tau_n = t_{n+1} - t_n$ and $\tau \in [0, \tau_n]$. Recall that $\tilde{u}(t_{n+1})$ is obtained from the computed solution $u_n^c$ and $u_n^c$ is not smooth function. Thus, to estimate the spatial error $||\tilde{u}(t_{n+1}) - u^h(t_{n+1})||$, we need to further split it as follows:

$$
||\tilde{u}(t_{n+1}) - u^h(t_{n+1})||_{L_1(\Omega_j)} = ||\tilde{u}(t_{n+1}) - u^e(t_{n+1})||_{L_1(\Omega_j)} 
+ ||u^e(t_{n+1}) - u^p(t_{n+1})||_{L_1(\Omega_j)} 
+ ||u^p(t_{n+1}) - u^h(t_{n+1})||_{L_1(\Omega_j)}
$$

where, $u^e$ is the local auxiliary piecewise strong solution and $u^p$ is the $L_2$-projection of $u^e$ into the polynomial space. We denote $u^e_j$, for the local strong solution of the conservation law. Then the initial values of $u^e_j$ are given on the line segment $\Omega_{j-1} \cup \Omega_j \times \{t_n\}$ by

$$
u^e_j(x, t_n) = M^0_{n,j} + M^1_{n,j}(x - x_{j-1/2}) + M^2_{n,j} \frac{(x - x_{j-1/2})^2}{2!} + M^3_{n,j} \frac{(x - x_{j-1/2})^3}{3!}.
$$

In $\Omega_j$, $u^e_j(t_n) = u^c_n$. In $\Omega_{j-1}$, we have

$$
u^e_j(t_n) = u^c_n + J^0_{n,j} + J^1_{n,j}(x - x_{j-1/2}) + J^2_{n,j} \frac{(x - x_{j-1/2})^2}{2!} + J^3_{n,j} \frac{(x - x_{j-1/2})^3}{3!}.
$$
Note that in the region,

\[ \mathcal{R}_{n,j} = \{ (\tilde{x}, t) | t \in [t_n, t_{n+1}], \ x \in \Omega_{j-1} \cup \Omega_j, \ \tilde{x} = x + (t - t_n) f'(u_j(x, t_n)) \} \]

the local strong solution \( u_j^\varepsilon \), definitely exists. The picture for this region is illustrated in Figure 5.1.

If time step size \( \tau_n \) satisfies the standard CFL condition \( \beta \tau_n \leq h \), then we get

\[ \Omega_j \times [t_n, t_{n+1}] \subseteq \mathcal{R}_{n,j} \subseteq (\Omega_{j-1} \cup \Omega_j) \times [t_n, t_{n+1}] \]

and, so \( u^\varepsilon(x, t) = u_j^\varepsilon(x, t) \) for \( (x, t) \in \Omega_j \times [t_n, t_{n+1}] \).

5.1.1 Error Estimation of DG Cells

The detail error analysis of this sub-section can be found in (20). We only outline the main steps here.

**Lemma 5.7.** If \( \beta \tau_n \leq h \) and \( \tau_n \leq \gamma h^{1+1/3} \), there is a computable constant \( C(S_n^p) \) depending on the spatial smoothness indicator \( S_n^p \), such that

\[ ||\tilde{u}(t_{n+1}) - u^\varepsilon(t_{n+1})||_{L^1(\Omega_U)} \leq \tau_n h^4 C(S_n^p). \]

**Proof.** In the cell \( \Omega_j \), at time \( t_{n+1} \), both solutions \( \tilde{u} \) and \( u^\varepsilon \) depends on their initial values in
\[ \Omega_{j-1} \cup \Omega_j \] and by the characteristic line theory, \( \tilde{u} \) and \( u^e \) depend on their initial value in

\[ [x_{j-1/2} - \beta \tau_n, x_{j-1/2}] \cup \Omega_j, \]

where \( \beta = \max f'(u) \). The use of \( L_1 \)-contraction property yields

\[ ||\tilde{u}(t_{n+1}) - u^e(t_{n+1})||_{L_1(\Omega_j)} \leq ||u^c_n - u^e(t_n)||_{L_1[x_{j-1/2} - \beta \tau_n, x_{j-1/2}]} \cdot \]

Since \( u^e_j(t_n) = u^c_n \) for all \( x \in \Omega_j \), in \( [x_{j-1/2} - \beta \tau_n, x_{j-1/2}] \)

\[ u^e_j(t_n) - u^c_n = J^0_{n,j} + J^1_{n,j} (x - x_{j-1/2}) + J^2_{n,j} \frac{(x - x_{j-1/2})^2}{2!} + J^3_{n,j} \frac{(x - x_{j-1/2})^3}{3!}. \]

Hence for \( j > 0 \), and \( x \in [x_{j-1/2} - \beta \tau_n, x_{j-1/2}] \), we get

\[ |u^e_j(t_n) - u^c_n| = |D^0_{n,j} h^{3+1+1} + D^1_{n,j} h^{3+1+1-(1+1/3)} (x - x_{j-1/2}) \]
\[ + \cdots + D^3_{n,j} h^{3+1+1-(1+1/3)} \frac{(x - x_{j-1/2})^3}{3!}| \]
\[ \leq |D^0_{n,j} h^{3+1+1} + |D^1_{n,j} h^{3+1+1-(1+1/3)} \beta \tau_n + \cdots + |D^3_{n,j} h^{3+1+1-(1+1/3)} \frac{(\beta \tau_n)^3}{3!}. \]

Using \( \tau_n \leq \gamma h^{1+1/3} \), we get

\[ |u^e_j(t_n) - u^c_n| \leq |D^0_{n,j} h^{3+1+1} + |D^1_{n,j} h^{3+1+1-(1+1/3)} \beta \gamma h^{1+1/3} \]
\[ + \cdots + |D^3_{n,j} h^{3+1+1-(1+1/3)} \frac{(\beta \gamma)^3}{3!}| \]
\[ \leq h^{3+1+1} |D^0_{n,j}| + |D^1_{n,j} \beta \gamma + |D^2_{n,j} \frac{(\beta \gamma)^2}{2!} + |D^3_{n,j} \frac{(\beta \gamma)^3}{3!}| \]

put

\[ \tilde{D}_{n,j} = |D^0_{n,j}| + |D^1_{n,j} \beta \gamma + |D^2_{n,j} \frac{(\beta \gamma)^2}{2!} + |D^3_{n,j} \frac{(\beta \gamma)^3}{3!}| \]
to get
\[ |u_j^e(t_n) - u_n^c| \leq h^{3+1+1}\tilde{D}_{n,j}. \]

The area of the curve \( u_j^e(t_n) - u_n^c \), on an interval \([x_{j-1/2} - \beta \tau_n, x_{j-1/2}]\), is
\[ ||u_j^e(t_n) - u_n^c||_{L_1[x_{j-1/2} - \beta \tau_n, x_{j-1/2}]} \leq \beta \tau_n h^{(3+2)}\tilde{D}_{n,j} \]

Now, adding up the above result for all cells in the first region \( \Omega_U \) will leads to
\[ ||\tilde{u}(t_{n+1}) - u^e(t_{n+1})||_{L_1(\Omega_U)} \leq \tau_n h^4 C(S_p^p). \]

Next, to get estimate for
\[ ||u^e(t_{n+1}) - u^p(t_{n+1})||_{L_1(\Omega_U)}, \]
we need more information about the smoothness of the auxiliary solution \( u^e \). Notice that \( L_2 \)-projection of \( u^e \) into the polynomial space is given by
\[ (u^p, v)_{\Omega_j} = (u^e, v)_{\Omega_j} \]
for all \( v \) in the polynomial space. To accomplish the estimate, some inverse inequalities of the finite element space from the book (2) has been used. For any \( v \) in finite element space there exists a positive constants \( \mu_1, \mu_2, \) and \( \mu_3 \) independent of \( v \) and \( h \) such that
\[ ||v_x||_{L_2} \leq \frac{\mu_1}{h} ||v||_{L_2} \quad (5.8) \]
\[ ||v||_{\Gamma_h} \leq \frac{\mu_2}{\sqrt{h}} ||v||_{L_2} \quad (5.9) \]
\[ \|v\|_{L_\infty} \leq \frac{\mu_3}{\sqrt{h}} \|v\|_{L_2} \]  

(5.10)

where \( \|v\|_{\Gamma_h} \) is defined by

\[ \|v\|_{\Gamma_h} = \left[ \sum (v_{j+1/2}^+)^2 + (v_{j+1/2}^-)^2 \right]^{1/2}. \]

Obviously, the summation is taken over all cells in the first region \( \Omega_U \). More detail about the inverse inequalities can be found in (2). Let us denote \( u_e^j \) by \( w \) and \( \frac{\partial l}{\partial x} w = w^l \). Since \( u_e^j(t_n) \) is a polynomial in \( \Omega_{j-1} \cup \Omega_j \), for small \( \tau_n \), \( u_e \) will be smooth in \( \Omega_j \times [t_n, t_{n+1}] \).

Now, as in (20), we state the following two useful lemmas.

**Lemma 5.11.** There are constants \( I_{n,j}^l \) which can be computed from \( M_{0,n,j}^l, M_{1,n,j}^l, M_{2,n,j}^l, \) and \( M_{3,n,j}^l \) such that

\[ \|w^l(t_n)\|_{L_\infty(\Omega_j \cup \Omega_{j-1})} \leq I_{n,j}^l \quad l = 0, 1, 2, 3, \]  

(5.12)

where

\[ I_{n,j}^l = \sum_{k=l}^3 |M_{k,n,j}^l| \frac{h_k}{k!}. \]

**Lemma 5.11** is the particular case \( p = 3 \) from the general case presented in (20). So, we refer (20) for proofs.

**Lemma 5.13.** There are constants \( I_{n,j}^l \), \( (l = 0, 1, 2, 3) \) which depends on the flux function \( f \) and can be computed from \( M_{0,n,j}^l, M_{1,n,j}^l, M_{2,n,j}^l, \) and \( M_{3,n,j}^l \) such that

\[ \left\| \frac{\partial^l}{\partial x^l} w(x,t) \right\|_{L_\infty(\Omega_{n,j})} \leq N_{n,j}^l \quad l = 0, 1, 2, 3 \]  

(5.14)

moreover,

\[ \left\| \frac{\partial^4}{\partial x^4} w(x,t_{n+1}) \right\|_{L_\infty(\Omega_j)} \leq \tau_n N_{n,j}^4 \]  

(5.15)

The detail proof is available in (20). However, for reader’s convenience we outline the main points of the proof here. The proof is separated for the cases \( l = 0, 1, 2 \leq l \leq 3, \) and \( l = 4 \).
For $l = 0$, result can be obtained by using Lemma 5.11 and Theorem 16.1 of the book (21). In this case, we have
\[
||w(t_n)||_{L^\infty(\Omega_j \cup \Omega_{j-1})} \leq I^0_{n,j} = N^0_{n,j}.
\]

Now, assume $l = 1$. We take $w$ as a strong solution of the conservation law (1.1) in the region $\mathcal{R}_{n,j}$ and hence,
\[
w_t + f(w)_x = 0.
\]
Differentiating the above equation with respect to $x$ gives,
\[
w_t^{(1)} + f'(w)w_x^{(1)} + f''(w)(w^{(1)})^2 = 0.
\]
Along the characteristic line it becomes the following ODE
\[
\frac{d}{dt}w^{(1)} + f''(w)(w^{(1)})^2 = 0.
\]
Thus, we have
\[
\frac{d}{dt}w^{(1)} = -f''(w)(w^{(1)})^2.
\]
Note that the initial value of $w^{(1)}$ is bounded by $I^1_{n,j}$ and for small value of the time step size $\tau_n$, it remains bounded. Also, $w$ is bounded by the constant $N^0_{n,j}$. The flux function $f(w)$ is bounded and bound depends on $f$. Let $F_2$ is the bound for $f''(w)$. Then, by using the Lemma A.1 of (20), we get
\[
|w^{(1)}| \leq \frac{I^1_{n,j}}{1 - F_2 I^1_{n,j}\tau_n} = N^1_{n,j},
\]
For small value of $\tau_n$, $1 - F_2 I^1_{n,j}\tau_n$ is close to 1 and hence, $N^1_{n,j}$ and $I^1_{n,j}$ are close to each other.

By the same techniques, we calculate the constant $N^2_{n,j}$ and obtain
\[
|w^{(2)}| \leq N^2_{n,j} = \left(I^2_{n,j} + \frac{F_3(N^1_{n,j})^3}{3F_2N^1_{n,j}}\right) e^{(3F_2N^1_{n,j})\tau_n} - \frac{F_3(N^1_{n,j})^3}{3F_2N^1_{n,j}},
\]
where $F_3$ is the bound for $f'''(w)$. By the similar way, for

$$B = 3F_2(N_{n,j}^2)^2 + 6F_3 N_{n,j}^2 (N_{n,j}^1)^2 + F_4 (N_{n,j}^1)^4$$

and $A = 4F_2 N_{n,j}^1$

$$|w^{(3)}| \leq N_{n,j}^3 = \left( I_{n,j}^3 + \frac{B}{A} \right) e^{A\tau_n} - \frac{B}{A}$$

where $F_4$ is the bound for $f''''(w)$. Finally, $|w^{(4)}| \leq N_{n,j}^4$, where $N_{n,j}^4$ is obtained by the equation

$$N_{n,j}^4 \tau_n = \frac{C}{D} (e^{At} - 1)$$

for $C = 5F_2 N_{n,j}^1$ and $D$ is some constant bounding the function of $f^{(k)}$ and $w^{(l)}$ with $2 \leq k \leq 5$ and $1 \leq l \leq 3$.

We compute these constants in our numerical experiment section. The complexity of computing these constants depend on the complexity of the flux function $f$. For Burgers’ equation $f'''(w) = 0$, so $F_3 = 0$. Moreover all the higher derivatives vanishes and the computation is fairly easy.

**Lemma 5.16.** For sufficiently small $\tau_n$,

$$||u^e - u^p||_{L_1(\Omega_j)} \leq h^{\frac{\tau_n N_{n,j}^4}{4!}}$$ (5.17)

$$||u^e - u^p||_{L_2(\Omega_j)} \leq h^{\frac{\tau_n N_{n,j}^4}{4!}}$$ (5.18)

To get result for the first region $\Omega_U$, we simply add the error of each cell. Accumulation effect will leads the error bound $\tau_n O(h^4)$.

**Proof.** We recall that $u^p$ is the $L_2$-projection of $u^e$ into the polynomial space. Assume that $u^i$ be the Lagrange interpolation of $u^e$ with 4 points. Notice that $u^p$ is the nearest function from $u^e$ in
\[ \| u^e - u^p \|_{L^2(\Omega_j)} \leq \| u^e - u^i \|_{L^2(\Omega_j)}. \]

On the other hand, for the Lagrange interpolation we have,

\[ |u^e(x) - u^i(x)| \leq \frac{1}{4!} \max_{x \in \Omega_j} w^4(x) \max_{x \in \Omega_j} \left| (x - x_0)(x - x_1)(x - x_2)(x - x_3) \right|. \]

Since \(|x - x_k| \leq h\) for \(k = 0, 1, 2, 3\) and using the bound for \(w^4\), we get

\[ |u^e(x) - u^i(x)| \leq \frac{\tau_n N^4_{n,j} h^4}{4!}. \]

Hence,

\[ \| u^e - u^i \|_{L^\infty(\Omega_j)} \leq \frac{\tau_n N^4_{n,j} h^4}{4!}. \]

Now using the relation between \(L^p(p = \infty, 1, 2)\) spaces on the bounded domain (see (19)), we get

\[ \| u^e - u^i \|_{L^1(\Omega_j)} \leq \frac{\tau_n N^4_{n,j} h^4}{4!}. \]

\[ \| u^e - u^i \|_{L^2(\Omega_j)} \leq \sqrt{h} \frac{\tau_n N^4_{n,j} h^4}{4!}. \]

Next, using the fact \(\| u^e - u^p \|_{L^2(\Omega_j)} \leq \| u^e - u^i \|_{L^2(\Omega_j)}\),

\[ \| u^e - u^p \|_{L^2(\Omega_j)} \leq \| u^e - u^i \|_{L^2(\Omega_j)} \]

\[ \leq \sqrt{h} \frac{\tau_n N^4_{n,j} h^4}{4!}. \]

Again, using the relation between \(L^p(p = 1, 2)\) spaces in bounded domain, we get

\[ \| u^e - u^p \|_{L^1(\Omega_j)} \leq \frac{\tau_n N^4_{n,j} h^4}{4!}. \] \hspace{1cm} (5.19)

Now, simply adding over all the cells in the first region \(\Omega_U\) yields the required result. \(\square\)
Lemma 5.20. There is a computable constant $Q_n$, depending on $S^p_n$, such that

$$||u^p(t_{n+1}) - u^h(t_{n+1})||_{L^1(\Omega_U)} \leq \tau_n h^4 Q_n.$$  \hspace{1cm} (5.21)

The idea of proof is to split $||u^p(t_{n+1}) - u^h(t_{n+1})||$, into many parts and see bound for each terms. We outline only the main steps of the proof here and refer to (20) for detail.

Proof. For each cell $\Omega_j$, $u^h$ satisfies the semi-discrete DG scheme,

$$(u^h_t, v)_{\Omega_j} = (f(u^h), v_x)_{\Omega_j} + f(u^h(x_{j-1/2}^-))v(x_{j-1/2}^+) - f(u^h(x_{j+1/2}^-))v(x_{j+1/2}^-).$$  \hspace{1cm} (5.22)

Since $u^e$ is a piecewise strong solution on each cell $\Omega_j$, we have $u^e_t + f(u^e)_x = 0$. Multiplying by a test function $v$ and integrating over $\Omega_j$ gives,

$$(u^e_t, v)_{\Omega_j} = (f(u^e), v_x)_{\Omega_j} + f(u^e(x_{j-1/2}^-))v(x_{j-1/2}^+) - f(u^e(x_{j+1/2}^-))v(x_{j+1/2}^-).$$  \hspace{1cm} (5.23)

Using the fact $(u^p, v)_{\Omega_j} = (u^e, v)_{\Omega_j}$ we have, $$(u^p_t, v)_{\Omega_j} = (u^e_t, v)_{\Omega_j}.$$  By adding and subtracting terms in the above equation (5.23), we get

$$(u^p_t, v)_{\Omega_j} = (f(u^p), v_x)_{\Omega_j} + f(u^e(x_{j-1/2}^+))v(x_{j-1/2}^+) - f(u^e(x_{j+1/2}^-))v(x_{j+1/2}^-)$$

$$+ (f(u^e) - f(u^p), v_x)_{\Omega_j} - [f(u^e(x_{j+1/2}^-)) - f(u^e(x_{j+1/2}^-))]v(x_{j+1/2}^-)$$  \hspace{1cm} (5.24)
Now let $\xi = u^p - u^h$, and let $v = \xi$. From (5.22) and (5.24), we have

\[
(\xi_t, \xi)_{\Omega_j} = ((f(u^p) - f(u^h)), \xi_x)_{\Omega_j}
\]

\[
- [f(u^p(x_{j+1/2})) - f(u^h(x_{j+1/2}))]|(x_{j+1/2})
\]

\[
+ [f(u^e(x_{j-1/2})) - f(u^h(x_{j-1/2}))]|(x_{j-1/2})
\]

\[
+ ((f(u^e) - f(u^p)), \xi_x)_{\Omega_j}
\]

\[
- [f(u^e(x_{j+1/2})) - f(u^p(x_{j+1/2}))]|(x_{j+1/2}).
\]

Now, we will further split the third term of the right hand side and use the previous lemmas, Mean Value Theorem, Holder’s inequality and the inverse inequalities and the result $(\xi_t, \xi)_{\Omega_j} = \frac{d}{dt}[||\xi||_{L_2(\Omega_j)}]$ to get

\[
\frac{d}{dt}[||\xi||_{L_2(\Omega_j)}] \leq \frac{C_1}{h}||\xi||_{L_2(\Omega_j)} + \frac{C_2h^5}{h} + \frac{C_3\tau_nh^4}{h}
\]

(5.25)

where

\[
C_1 = \beta\mu_1 + \beta 2\mu_2^2
\]

\[
C_2 = \beta\mu_2\sqrt{h}\left[\sum \left(\frac{\tilde{D}_{n,j}}{1 - \tau_nF_2N_{n,j}^1}\right)^2\right]^{1/2}
\]

\[
C_3 = \beta(\mu_1 + 2\mu_2^2)\sqrt{h}\left[\sum (N_{n,j}^{(4)})^2\right]^{1/2}
\]

Now, using the fact that $\tau_n \leq \gamma h^{1 + 1/3}$, the last equation (5.25) reduces to

\[
\frac{d}{dt}[||\xi||_{L_2(\Omega_j)}] \leq \frac{C_1}{h}||\xi||_{L_2(\Omega_j)} + \frac{C_2h^5}{h} + \frac{C_3\gamma h^{5+1/3}}{h}
\]

Next, using Lemma A.3 from Appendix of (20) with zero initial condition, we get

\[
||\xi||_{L_2(\Omega_j)} \leq h^{1}\left[\frac{C_2 + \gamma h^{1/3}C_3}{C_1}\right][h(e^{C_1\gamma/h} - 1)].
\]

(5.26)
Choose some suitable $\theta$ to get

$$
||\xi||_{L_2(\Omega_U)} \leq h^4 \left[ \frac{C_2 + \gamma h^{1/3} C_3}{C_1} \right] h(e^{\theta C_1 \tau_n/h} - 0)
$$

$$
= \tau_n h^4 \left[ \frac{C_2 + \gamma h^{1/3} C_3}{C_1} \right] C_1 e^{\theta C_1 \tau_n/h}
$$

Finally, we use the fact

$$
||\xi||_{L_1(\Omega_U)} \leq \sqrt{|\Omega_U|} ||\xi||_{L_2(\Omega_U)}
$$

to get (5.21).

5.1.2 Error Estimation on the Constructed New DG cell

Assume that $\Omega_j$ is the cell where we need the reconstruction of a new DG solution at time $t = t_{n+1}$. For convenience, we denote the FT solution for the Cubic Front Tracking solution. Let $\delta_1 > 0$ be such that at $t = t_n$, and $x \in [x_{j-1} - \delta_1, x_{j-1}/2]$, we have overlapping between the DG solution and the FT solution. Similarly, let $\delta > 0$ be such that at $t = t_{n+1}$ and $x \in [x_{j-1}/2, x_{j-1}/2 + \delta]$, we have gap between the DG solution and the FT solution. As illustrated in Figure 5.2, the thick lines in the figure represent the parts of the solution where both the DG solution and FT solution overlapped. At time $t = t_n$, we extrapolate the FT solution on $\Omega_{j-1}$ by $\delta$ unit left and use the method of characteristics to get $\hat{u}(t_{n+1})$. Thus, we assume $\hat{u}(t_{n+1})$ be the characteristic solution of the extrapolated FT solution.

At time $t = t_{n+1}$, first we extrapolate the FT solution by $\delta$ unit to the left to recover the missing part. We use $\hat{u}(x, t_{n+1})$ to denote the inverse of the extrapolated FT solution.

We need to measure the error $||\hat{u}(t_{n+1}) - u_0^{n+1}||_{L_1(\Omega_j)}$. For this, we introduce the new auxiliary FT solution $u_c^t(x)$ at $t_{n+1}$ such that $u_c^t(x) = \hat{u}(x)$ for $x \in [x_{j-1}/2 + \delta, x_{j+1}/2]$. For $x \in [x_{j-1}/2, x_{j-1}/2 + \delta]$,

$$
|u_c^l(x) - \hat{u}(x)| \leq |u_c^l(x_{j-1}/2) - \hat{u}(x_{j-1}/2)|.
$$
Figure 5.2: Detachment of the DG Solution and FT solution

Here the variable $t_{n+1}$ is hiding.

To estimate $|\tilde{u}(x, t_{n+1}) - u^c_{n+1}|$, we use the triangle inequality

$$|\tilde{u}(x, t_{n+1}) - u^c_{n+1}| \leq |\tilde{u}(x, t_{n+1}) - u^e_t(x, t_{n+1})| + |u^e_t(x, t_{n+1}) - \hat{u}(x, t_{n+1})| + |\hat{u}(x, t_{n+1}) - u^c_{n+1}|.$$  (5.27)

First, consider the term $|\tilde{u}(x, t_{n+1}) - u^e_t(x, t_{n+1})|$. Note that $\tilde{u}(x, t_{n+1})$ is a characteristic solution and $u^e_t(x, t_{n+1})$ is the FT solution. By the Cubic Front Tracking error analysis presented in Chapter 4, we have

$$|\tilde{u}(x, t_{n+1}) - u^e_t(x, t_{n+1})| \leq C_t \tau_n h^4$$  (5.28)

for some constant $C_t = T_1 M_5$ used in the Cubic Front Tracking error estimate. Note that for a polynomial flux up to degree 4, $C_t = 0$.

Since $|u^e_t(x, t_{n+1}) - \hat{u}(x, t_{n+1})| \leq |u^e_t(x_{j-1/2}) - \hat{u}(x_{j-1/2})|$, there is some constant $C_s$ depending on smoothness indicators $S^p_n$ such that

$$|u^e_t(x, t_{n+1}) - \hat{u}(x, t_{n+1})| \leq C_s h^4$$  (5.29)
To estimate $|\hat{u}(x, t_{n+1}) - u_{n+1}^c|$, we use the same idea as in the conversion error estimate presented in Chapter 4. We estimate $\hat{u}(x, t_{n+1})$ by a cubic polynomial $u_{n+1}^c$ satisfying

\[ \hat{u}(x_{j-1/2}) = u_{n+1}^c(x_{j-1/2}), \hat{u}(x_{j+1/2}) = u_{n+1}^c(x_{j+1/2}), \]

\[ \frac{d}{dx} \hat{u} \left( \frac{x_{j-1/2} + x_{j+1/2}}{2} \right) = \frac{d}{dx} u_{n+1}^c \left( \frac{x_{j-1/2} + x_{j+1/2}}{2} \right), \]

and

\[ \int_{x_{j-1/2}}^{x_{j+1/2}} \hat{u}(x) dx = \int_{x_{j-1/2}}^{x_{j+1/2}} u_{n+1}^c(x) dx. \]

By the similar error estimate we presented in Section 4.1, we get

\[ |\hat{u}(x, t_{n+1}) - u_{n+1}^c| \leq C_{ss} h^4 \] (5.30)

for some constant $C_{ss}$. Now from (5.27), we get $|\bar{u}(x, t_{n+1}) - u_{n+1}^c| \leq C_{p} h^4$, where $C_{p}$ is a constant depends on interpolation constants and smoothness indicators. Finally, integrating over the cell $\Omega_j$, yields

\[ ||\bar{u}(x, t_{n+1}) - u_{n+1}^c||_{L_1(\Omega_j)} \leq C_{p} h^5 \] (5.31)

Now, combining the results of the sub-sections 5.1.1 and 5.1.2, we get the following theorem.

**Theorem 5.32.** There is a computable function $F_1(S_{n}^p)$, depending on the flux function $f$, the known constant of inverse inequalities, the known constant of interpolation/projection errors and the components of the spatial smoothness indicator such that

\[ ||\bar{u}(t_{n+1}) - u^h(t_{n+1})||_{L_1(\Omega_U)} \leq F_1(S_{n}^3) \tau_n h^4 + [h_{rc}(n)] C_{p} h^4, \] (5.33)

where

\[ h_{rc}(n) = \begin{cases} h & \text{if reconstruction of new DG cell is required in } [t_n, t_{n+1}] \\ 0 & \text{otherwise} \end{cases} \]
and $C^p_s$ is a computable constant.

**Remark:** The main error estimate for the third region $\Omega_D$ is similar to the main error estimates for the first region $\Omega_U$. 
CHAPTER 6 TEMPORAL ERROR ESTIMATION AND THE SHOCK LOCATION ERROR

6.1 Temporal Error Estimation

Now, we focus to accomplish the temporal error estimation on DG cells. This task can be done in two steps. We recall that in the assumption of the main error estimate presented in Theorem 5.1, we assume that the temporal smoothness indicator $T_n^k$ is bounded. First, we should confirm that the boundedness of smoothness indicator $T_n^k$ yields the boundedness of temporal derivatives of the semi-discrete solution $u^h(x, t)$, for $t \in [t_n, t_{n+1}]$. After we get bounds on $\frac{d^l u^h}{dt^l}, l = 0, 1, 2, 3$ and 4, we will estimate $||u^h(x, t) - u^c_{n+1}||_{L_1(\Omega_U)}$.

We notice that the temporal error estimation for DG cells can be found in (20). In our case, we apply this temporal error estimation on the first part $\Omega_U$ and the third part $\Omega_D$ of the domain $\Omega = [a, b]$. To make it convenient for the readers, we write it here as well. Recall that throughout this dissertation, we assume that $\tau_n$ satisfies the standard CFL condition $\beta \tau_n \leq h$ and the strengthened CFL condition $\tau_n \leq \gamma h^{1+1/3}$.

**Theorem 6.1.** (20) There is a computable function $G_1(T_n^k, S_n^p)$, such that

$$||u^h(x, t) - u^c_{n+1}||_{L_1(\Omega_U)} \leq (\tau_n)^4 G_1(T_n^k, S_n^p)$$

First we will illustrate the proof for the boundedness of $\frac{d^l u^h(t)}{dt^l}, l = 0, 1, 2, 3$ and 4 for $t \in [t_n, t_{n+1}]$.

6.1.1 Bound on Temporal Derivatives of $u^h(t)$

**Lemma 6.3.** There is a computable constant $K$, depending on smoothness indicators such that

$$||u^h(t)||_{L_\infty} \leq ||u^c_n||_{L_\infty} + Kh.$$  

The idea of proof is to split $u^h(t)$ further and use the result obtained in the first section of Chapter 5.
Proof. We split $u^h(t)$ for $t \in [t_n, t_{n+1}]$ further and use triangle inequality to get,

$$||u^h(t)||_{L_\infty} \leq ||\tilde{u}(t)||_{L_\infty} + ||\tilde{u}(t) - u^e(t)||_{L_\infty}$$

$$+ ||u^e(t) - u^p(t)||_{L_\infty} + ||u^p(t) - u^h(t)||_{L_\infty}.$$

By the spatial error estimation, and appropriate use of inverse inequalities from the spatial error estimation section, we get the bound for the second, third and the last term of right hand side.

Also, by the result of Theorem 16.1 of the book (21), the maximum of any entropy solution does not increase. Precisely, $||\tilde{u}(t)||_{L_\infty} \leq ||u^e_n||_{L_\infty}$. Thus, there is a computable constant $K$ such that $||u^h(t)||_{L_\infty} \leq ||u^e_n||_{L_\infty} + Kh$.

Before illustrating the proof for bound on $\frac{d^l}{dt^l} u^h(t)$ for $l \geq 1$, we write some basics results which are used in proofs.

To get the semi-discrete solution, we solve $(p + 1)(N)$ linear ODE, where $N$ is the number of cells and $p$ is the degree of a polynomial. We define

$$||A||_{\infty} = \max_j \sum_i |a_{ij}|,$$

a largest row sum. If we are dealing with a system of ODE of the form

$$y'(t) = A(t)y(t) + r(t),$$

where $A(t)$ is a square matrix and $r(t)$ is a vector. We use the following result from the page 64 of (13).

**Lemma 6.4.** Consider a system of ODE of the form

$$y'(t) = A(t)y(t) + r(t),$$

where $A(.)$ have continuous coefficients and continuous $r(.)$. Given a vector norm and the corre-
sponding induced matrix norm \( ||A|| \) is given by

\[
L(t) = \int_0^t ||A(s)|| ds,
\]

then for a solution \( y(.) \) satisfies the inequality

\[
y(t) \leq e^{L(t)} ||y(0)|| + \int_0^t e^{-L(s)} ||r(s)|| ds, t \geq 0
\]

Lemma 6.5. (23) There are computable constants \((b_l, c_l)\) depending on \( S^n \) such that for all \( t \in [t_n, t_{n+1}] \)

\[
||\frac{\partial}{\partial t} u^h(t)||_{L\infty} \leq (1 + b_l h^{1+1/p}) ||\frac{\partial}{\partial t} u_n^c||_{L\infty} + c_l h^{1/p}
\]

Proof. This proof is also available in (20). We set the notation \( z = z(\tau) = u^h(t) \) for \( t \in [t_n, t_{n+1}], \tau \in [0, \tau_n] \) and for \( z^{(l)} = \frac{d^l}{dt^l} z \) for \( l = 1, 2, 3, 4 \). Being a semi-discrete solution \( z \) satisfies

\[
(z_t, v)_{\Omega_j} = (f(z), v_x)_{\Omega_j} + f(z(x_{j-1/2}^-))v(x_{j-1/2}^+) - f(z(x_{j+1/2}^-))v(x_{j+1/2}^-)
\]  (6.6)

Differentiating with respect to \( t \) to the both sides, we get

\[
(z_t^{(1)}, v)_{\Omega_j} = (f'(z)z^{(1)}, v_x)_{\Omega_j} + f'(z(x_{j-1/2}^-))z^{(1)}(x_{j-1/2}^-)v(x_{j-1/2}^-)
\]

\[
- f'(z(x_{j+1/2}^-))z^{(1)}(x_{j+1/2}^-)v(x_{j+1/2}^-).
\]

It is important to note that the equation for \( z^{(1)} \) is linear and depends on the derivative of the flux function \( f(u) \) and \( z \). To get estimate of \( z^{(1)} \) in \( L\infty \)-norm, we can expand \( z^{(1)} \) by a basis functions \( \{\phi_{j,i} : i = 0, 1, 2, 3\} \) in each DG cell \( \Omega_j \),

\[
z^{(1)}(\tau)_{\Omega_j} = \sum_{i=0}^3 q_{j,i}^{(1)}(\tau)\phi_{j,i}.
\]
Under this basis let $q^{(1)}$ be the vector consisting of all the $q_{j,i}^{(1)}$. We can rewrite the equation for $z^{(1)}$ as

$$
B \frac{d}{dt}q^{(1)} = A^{(1)}(\tau)q^{(1)}
$$

and

$$
\frac{d}{dt}q^{(1)} = B^{-1}A^{(1)}(\tau)q^{(1)}
$$

where $A^{(1)}(\tau)$ is the matrix obtained from the right hand side of the equation for $z^{(1)}$ and $B$ is the matrix containing the entries $\phi_{j,i_1}, \phi_{j,i_2})\Omega_j$. It is important to point out that if normalized Legendre polynomials are used as the basis, then matrix $B$ reduces to a diagonal matrix with entries $\frac{h}{2}$ for all DG cells. Note that entries of $A^{(1)}(\tau)$ depend on wave speed $f'(z)$. By the fact $f$ is smooth and $\|z\|_{L_\infty}$ is bounded by Lemma 6.3, all entries of $A^{(1)}(\tau)$ are bounded. Then by the result of Lemma 6.4, there exists a constant $\tilde{A}_1$ depending on $B^{-1}A^{(1)}(\tau)$ such that

$$
\|q^{(1)}(\tau) - q^{(1)}(0)\|_{L_\infty} \leq \tilde{A}_1\|q^{(1)}(0)\|_{L_\infty}.
$$

Since $\|z\|_{L_\infty}$ and $\|q^{(1)}\|_{L_\infty}$ are equivalent, there are constants $\tilde{B}$ and $\tilde{C}$ such that

$$
\|z^{(1)}(\tau)\|_{L_\infty} \leq \|z^{(1)}(0)\|_{L_\infty} + \|z^{(1)}(\tau) - z^{(1)}(0)\|_{L_\infty}
$$

$$
\leq \|z^{(1)}(0)\|_{L_\infty} + \tilde{B}\|q^{(1)}(\tau) - q^{(1)}(0)\|_{L_\infty}
$$

$$
\leq \|z^{(1)}(0)\|_{L_\infty} + \tilde{B}\tilde{A}_1\|q^{(1)}(0)\|_{L_\infty}
$$

$$
\leq \|z^{(1)}(0)\|_{L_\infty} + \tilde{B}\tilde{A}_1\tilde{C}\|z^{(1)}(0)\|_{L_\infty}
$$

Thus, taking $b_1 = \tilde{B}\tilde{A}_1\tilde{C}$ and $c_1 = 0$ yields the proof of lemma for the case $l = 1$. The constants $\tilde{B}$ and $\tilde{C}$ depends on the choice of basis. In our case of normalized Legendre polynomials of degree $p=3$, $\tilde{B} = (p+1)\sqrt{(2p+1)/2}$ and $\tilde{C} = \sqrt{2}$. Similarly, to get the bound for $z^{(2)}$, we differentiate
the equation for $z^{(1)}$ with respect to $t$. Then we have,

\[
(z^{(2)}_t, v)_{\Omega_j} = (f'(z) z^{(2)}, v_x)_{\Omega_j} + (f''(z) [z^{(1)}]^2, v_x)_{\Omega_j} \\
+ f'(z(x_{j-1/2}^-)) z^{(2)}(x_{j-1/2}^-) v(x_{j+1/2}^+) + f''(z(x_{j-1/2}^-)) [z^{(1)}(x_{j-1/2}^-)]^2 v(x_{j-1/2}^+) \\
- f'(z(x_{j+1/2}^-)) z^{(2)}(x_{j+1/2}^-) v(x_{j+1/2}^-) - f''(z(x_{j+1/2}^-)) [z^{(1)}(x_{j+1/2}^-)]^2 v(x_{j+1/2}^-).
\]

Now, we expand $z^{(2)}$ by a basis $\{\phi_{j,i} : i = 0, 1, 2, 3\}$ to get,

\[
z^{(2)}(\tau)|_{\Omega_j} = \sum_{i=0}^{3} q^{(2)}_{j,i}(\tau) \phi_{j,i}.
\]

We denote the vector consisting all entries of the $q^{(2)}_{j,i}$ by $q^{(2)}$. Then as before, we obtain a linear ODE for $q^{(2)}$ given by

\[
B q^{(2)} = A^{(2)}(\tau) q^{(2)} + r
\]

\[
q^{(2)} = B^{-1} A^{(2)}(\tau) q^{(2)} + B^{-1} r.
\]

Here the vector $r$ may consist of some terms which are independent of $q^{(2)}$. But, all of the terms in the vector $r$ depend on the lower order derivative of $z$ and the derivatives of the flux function. Thus, we have already bounded each term of the vector $r$. Note that matrix $B$ and $A^{(2)}$ are obtained in the same way as for $z^{(1)}$. Moreover, we can see that matrix $B$ is same for both $z^{(1)}$ and $z^{(2)}$. Also, we can see that $A^{(2)} = A^{(1)}$. Hence, it suffices to concentrate only on the vector $r$. By the Lemma 6.4, there is constants $\tilde{A}_2$, and $\tilde{r}_2$ depending on $B^{-1} A^{(2)}$ and $B^{-1} r$ respectively such that

\[
||q^{(2)}(\tau) - q^{(2)}(0)||_{L_\infty} \leq \tilde{A}_2 ||q^{(2)}(0)||_{L_\infty} + \tilde{r}_2.
\]
Using the same $\tilde{B}$ and $\tilde{C}$ due to equivalence of norms we get

\[
||z^{(2)}(\tau)||_{L_\infty} \leq ||z^{(2)}(0)||_{L_\infty} + ||z^{(2)}(\tau) - z^{(2)}(0)||_{L_\infty}
\]
\[
\leq ||z^{(2)}(0)||_{L_\infty} + \tilde{B}||q^{(2)}(\tau) - q^{(2)}(0)||_{L_\infty}
\]
\[
\leq ||z^{(2)}(0)||_{L_\infty} + \tilde{B}A_2||q^{(2)}(0)||_{L_\infty} + \tilde{B}r_2
\]
\[
\leq ||z^{(2)}(0)||_{L_\infty} + \tilde{B}A_2C||z^{(2)}(0)||_{L_\infty} + \tilde{B}r_2.
\]

Put $b_2 = \tilde{B}A_2\tilde{C}$ and $c_2 = \tilde{B}r_2$, to obtain the lemma for $l = 2$. The proofs for $l = 3$ and $l = 4$ are obtained in similar way. The only difference will be the vector $r$. Now, taking the derivative of $z^{(2)}$ with respect to $t$ yields the following

\[
(z_t^{(3)}, v)_{\Omega_j} = (F(z), v_x)_{\Omega_j} + F(z)(x_{j-1/2}^-)v(x_{j-1/2}^-) - F(z)(x_{j+1/2}^-)v(x_{j+1/2}^-)
\]

where

\[
F(z) = f'(z)z^{(3)} + 3f''(z)z^{(1)}z^{(2)} + f'''(z)[z^{(1)}]^3.
\]

Similarly, for $z^{(4)}$ we have

\[
(z_t^{(4)}, v)_{\Omega_j} = (G(z), v_x)_{\Omega_j} + G(z)(x_{j-1/2}^-)v(x_{j-1/2}^-) - G(z)(x_{j+1/2}^-)v(x_{j+1/2}^-)
\]

where

\[
G(z) = f'(z)z^{(4)} + 4f''(z)z^{(1)}z^{(3)} + 3f'''(z)[z^{(2)}]^2 + 4f'''(z)[z^{(1)}]^4.
\]

Here, we left some minor details of the proof for the case of $l = 3$ and $l = 4$ and refer to (20). □

**Remark:** The complexity to obtain the bound of the temporal derivative of semi-discrete solution depends on the complexity of the flux function and the type of the basis being used.
6.1.2 Error Estimate

Given an ODE of the form

\[ y' = F(y, t) \quad y(t_n) = y_0 \]

and \( s \)-stage explicit Runge-Kutta method can be written in the form

\[ y_n^c = y_0 + \tau_n (b_1 K_1 + b_2 K_2 + \ldots + b_s K_s). \] (6.7)

Here, \( b_i \)'s are constants depending on the particular Runge-Kutta scheme being used and \( K_i \)'s are function of \( F(y, t) \) and the previous \( K \) (13). When the Runge-Kutta scheme is given in above form, we can use the following result from the page 157 of (13).

If the Runge-Kutta method is in the form (6.7), of order \( k \) and if all partial derivatives of \( F(x, y) \) up to order \( k \) exists, then the local error of (6.7) achieves the following bound:

\[ \left\| y(t_n + \tau) - y_{n+1}^c \right\| \leq \frac{1}{(k + 1)!} \max_{\xi \in [0, 1]} y^{(k+1)}(t_n + \xi \tau_n) \right\| + \frac{1}{k!} \sum_{i=1}^{s} |b_i| \max_{\xi \in [0, 1]} \| K_i^{(k)}(\xi \tau_n) \| \] (6.8)

Since we use the Runge-Kutta scheme of order 3, above inequality reduces to

\[ \left\| y(t_n + \tau) - y_{n+1}^c \right\| \leq \tau_n^3 \left( \frac{1}{(4)!} \max_{\xi \in [0, 1]} y^{(4)}(t_n + \xi \tau_n) \right\| + \frac{1}{3!} \sum_{i=1}^{3} |b_i| \max_{\xi \in [0, 1]} \| K_i^{(3)}(\xi \tau_n) \| \] (6.8)

To apply this result, first we need to write our third order Runge-Kutta scheme as in the form (6.7). By using simple back substitution, we can write our third order Runge-Kutta scheme (1.6)-(1.8) in the following form:

\[ (u_{n+1}^c, v)_{\Omega_j} = (u_n^c, v)_{\Omega_j} + \tau_n \left[ \frac{1}{6} \mathcal{H}_j(u_n^c, v) + \frac{1}{6} \mathcal{H}_j(u_{n-1}^c, v) + \frac{2}{3} \mathcal{H}_j(u_{n-2}^c, v) \right] \] (6.9)

Notice that right hand side of the error estimation (6.8) has two main components; namely fourth
derivative of $y$ and third derivative of the stages $K_i$. Since we already proved the boundedness of the semi-discrete solution $u^h(t_n + \tau_n)$ in Lemma 6.5, bound for the first term is done. To see the boundedness of the second term, assume $w_1(t) = u^h(t)$ for $t \in [t_n, t_{n+1}]$. Recall that $u^h(t_n) = u^c_n$ and $H_j(w_1(t), v) = (u_t^h, v)_{\Omega_j}$. Thus, we have

$$(K_1, v)_{\Omega_j} = H_j(w_1(t), v)$$

and, $(K_1^{(3)}, v)_{\Omega_j} = (\frac{d^3}{dt^3} u^h, v)_{\Omega_j}$. Then, by Lemma 6.5, $K_1^{(3)}$ is bounded. Assume $w_2(t) = w_1(t) + \tau_n K_1(t)$ to get

$$(K_2, v)_{\Omega_j} = H_j(w_2(t), v).$$

In similar way, assume $w_3(t) = \frac{3}{4} w_1(t) + \frac{1}{4} w_2(t) + \frac{1}{4} \tau_n K(t)$ to get

$$(K_3, v)_{\Omega_j}.$$ 

Thus, $K_i$’s are bounded. To see the boundedness of $K_i^{(3)}$ for $i \geq 2$, it suffices to see the boundedness of $w_i^{(4)}$. Now, For $w_2^{(3)}$, we have

$$w_2^{(3)} = w_1^{(3)} + \tau_n K_1^{(3)}.$$ 

To get the bound for $w_2^{(4)}$, is quite challenging. For this, we follow the similar procedure as in Lemma 6.5. For example to see bound for $w_2^{(4)}$, we have

$$(w_2^{(4)}, v)_{\Omega_j} = (G(w_2), v_x)_{\Omega_j} + G(w_2(x_{j-1/2}^-))v(x_{j-1/2}^+) - G(w_2(x_{j+1/2}^-))v(x_{j+1/2}^-)$$

where

$$G(w_2) = f'(w_2)w_2^{(4)} + 4f''(w_2)z^{(1)}w_2^{(3)} + 3f'''(w_2)[w_2^{(2)}]^2 + 4f^{(4)}(w_2)[w_1^{(1)}]^4.$$ 

Thus, by the same technique presented in Lemma 6.5 we get a bound on $w_2^{(4)}$, and hence a bound on $K_2^{(3)}$. By the same idea we can estimate for $w_3^{(4)}$, and get the bound for $K_3^{(3)}$. Hence, $K_i$’s are
bounded. Once, we get the bound on both terms of the right hand side of (6.8), we prove Theorem 6.1.

Finally, the combination of Section 5.1 and Section 6.1, yields the proof of the main error estimate for the first region $\Omega_U$ presented in Theorem 5.1.

6.2 Shock Location Error

After the computed solution reduced to a mature “s-shape”, we locate the shock by using the equal area principle. The following lemma illustrates the bound for the shock location error.

**Lemma 6.10.** If $x_s$ and $\tilde{x}_s$ be the exact and the computed shock locations respectively, then

$$|x_s - \tilde{x}_s| \leq \tilde{C}h^4$$

for some constant $\tilde{C}$.

**Proof.** Let $u^e_l$ and $u^c_l$ be the $u$-values of the top and bottom of the exact shock. Similarly, $u^e_r$ and $u^c_r$ be the top and bottom of the computed shock. Then, by using the equal area principle, we have

$$\int_{u^e_l}^{u^e_r} x(u)du = \int_{u^c_l}^{u^c_r} x_{ct}(u)du = 0 = \int_{u^e_l}^{u^e_r} (x_{ct}(u) - \tilde{x}_s)du. \quad (6.11)$$

Then, re-writing above equation gives,

$$\int_{u^e_l}^{u^e_r} x(u)du - \int_{u^c_l}^{u^c_r} x_{ct}(u)du = \int_{u^e_l}^{u^e_r} x_sdu - \int_{u^c_l}^{u^c_r} \tilde{x}_sdu. \quad (6.12)$$

We denote

$$u^{*}_l = \max\{u^e_l, u^c_l\}, \quad u^{*}_r = \min\{u^e_r, u^c_r\}, \quad u^{**}_l = \max\{u^e_r, u^c_r\},$$

and

$$u^{**}_l = \min\{u^e_l, u^c_l\}.$$
Figure 6.1: Shock Locations and Fully Developed “s-shape” Solutions

By the spatial error estimation, we have

\[ |u^*_l - u^{**}_l| \leq Ch^4 \]

and

\[ |u^*_r - u^{**}_r| \leq Ch^4, \]

for some constant \( C \). Now, from (6.12),

\[
\int_{u^*_l}^{u^*_l} x_s du - \int_{u^*_l}^{u^*_l} \bar{x}_s du = \int_{u^*_l}^{u^{**}_l} (x_s - \bar{x}_s) du + \int_{u^*_l}^{u^{**}_l} (x_s - \bar{x}_s) du + \int_{u^*_l}^{u^{**}_l} (\bar{x}_s - x_s) du. \tag{6.13}
\]

First term on the right hand side of (6.13) is the area of rectangle with wide \( x_s - \bar{x}_s \) and length \( (u^{**}_l - u^{**}_r) \). Using the results \( |u^*_l - u^{**}_l| \leq Ch^4 \) and \( |u^*_r - u^{**}_r| \leq Ch^4 \), the last two terms of right hand side of (6.13) is \( O(h^4)(x_s - \bar{x}_s) \). Thus,

\[
\left| \int_{u^*_l}^{u^*_l} x_s du - \int_{u^*_l}^{u^*_l} \bar{x}_s du \right| = |S(x_s - \bar{x}_s) + O(h^4)(x_s - \bar{x}_s)|
\]

\[ = S_1 |x_s - \bar{x}_s|,
\]
where \( S = (u_l^{**} - u_r^{**}) \) and \( S_1 = S + O(h^4) \). Note that the constant \( S_1 \) is very close to the computed shock height.

On the other hand,

\[
\left| \int_{u_l}^{u_r} x(u) du - \int_{u_l^*}^{u_r^*} x_{ct}(u) \right| \leq \left| \int_{u_l^*}^{u_r^*} [x(u) - x_{ct}(u)] du \right| \\
\leq |u_l^* - u_r^*| ||x(u) - x_{ct}(u)||_{L_{\infty}(\Lambda)} \\
\leq |u_l^* - u_r^*| C_{sh}^4.
\]

Here, the constant \( C_{sh} \) is obtained from the conversion error and the Cubic Front Tracking error estimates. Thus, from (6.12) and the above estimates

\[
S_1 |x_s - \bar{x}_s| \leq C_{sh} h^4
\]

and \( |x_s - \bar{x_s}| \leq \bar{C} h^4 \) for \( \bar{C} = \frac{C_{sh}}{S_1} \).

\( \square \)
CHAPTER 7  NUMERICAL EXPERIMENTS

This chapter provides numerical examples. First, we solve Burgers’ equation

\[ u_t + \left( \frac{u^2}{2} \right)_x = 0 \quad (7.1) \]

with the boundary condition \( u_a(t) = 2 \) and the initial condition

\[
 u_0(x) = \begin{cases} 
 2 - \int_0^x \sin^3 \left( \frac{\pi t}{3} \right) dt & x \leq 3 \\
 \frac{2(\pi - 2)}{\pi} & x > 3
\end{cases}
\]

(7.2)

We take \( \Omega = [a, b] = [0, 6] \) and use the third degree polynomials, \( p = 3 \), and a third order TVD Runge Kutta, \( k = 3 \). For \( M = 3.333333333 \), the transition time begins after \( t \approx 0.7 \), and we obtained the fully developed “s-shape” solution at \( t \approx 1.15 \). In this example, the breaking time is

\[ T_b = \min_{x \in [a, b]} \frac{-1}{u_0(x)} = 1. \]

Thus, the ratio of transition time before the breaking time and after the breaking time is \( 2 : 1 \). This supports our theoretical result from Chapter 3. We implement the RKDG method to get the numerical solution until the slope reaches \(-M\). After the slope of the computed solution reached \(-M\), we saw the further sharpening and the possible loss of smoothness through the computed smoothness indicators. To prevent possible numerical oscillations, we convert the “steep part” of the solution in the cubic polynomial in \( u \) in piecewise manner and use the Cubic Front Tracking scheme to evolve the solution on this part. After carefully implementing the our algorithm presented in Chapter 3, we get the fully developed “s-shape” solution. The computed “s-shape” solution has no numerical oscillations and is of high quality. The computed solution on different stages of the transition time is illustrated in Figure 7.1.

The zoom of the computed fully developed “s-shape” solutions before and after the shock has
been located as illustrated in Figure 7.2 and Figure 7.3, respectively. At the time of having a fully
developed shock, there is only a single cell in between the DG solution of the upstream part and
the DG solution of the downstream part. After the shock is being located, one can convert the three
cells into two cells by putting the shock as the cell boundaries. The detailed error analysis after
having a fully developed shock has been done in (12).

7.1 Error analysis on the FT Region

In this section, we present the computational results that support the error analysis presented
in Chapter 4. We recall that the error due to the conversion from $u^c_n$ to $x_c(u)$ is estimated as
$|\tilde{x}(u) - x_c(u)| \leq T_1 M_4 h^4$. The value of the constant $T_1 = \frac{21}{1980} M^4$. For our computation of
$M = 2.5$, the maximum value of the constant $T_1 = \frac{21}{1980} \times (2.5)^4 = 0.4142$. Moreover, in most of
the cells the slope is even smaller and taking the fourth power makes much smaller. Note that the
constant $M_4$ has different values on the different converted cells. Recall that in the denominator
of the bound of the constant $M_4$ has the term $\tilde{u}'(x,t_n)^7$. Thus, when $|\tilde{u}'(x,t_n)| = M > 1$, the
constant $M_4$ has much smaller upper bound. This shows that our converted solution is of high
order.

In our example of the Burgers’ equation, $f(u) = \frac{u^2}{2}$, and hence $M_5 = |f^{(5)}(u)| = 0$. This
Figure 7.2: Zoom in of Computed “s-shape” Solution Before the Shock is being Located

Figure 7.3: Zoom in of Computed “s-shape” Solution After the Shock is being Located
shows that the constant $C_t$ used in Cubic Front Tracking error estimate is zero. Thus, there is no Cubic Front Tracking error for this case.

7.2 Smoothness Indicators

We claimed at the beginning of the dissertation that our error analysis relies heavily on the computed smoothness indicators. In this section, we present all the computed smoothness indicators for both the first and third regions. Table 7.1 represents the content of the smoothness indicators.

Table 7.1: The Contents of the Smoothness Indicators in Figures 7.4-7.18

<table>
<thead>
<tr>
<th>$M^0_{n,j}$</th>
<th>$M^1_{n,j}$</th>
<th>$M^2_{n,j}$</th>
<th>$M^3_{n,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_t^n(x,t_n)$</td>
<td>$u_t^n(x,t_n)$</td>
<td>$u_{ttt}^n(x,t_n)$</td>
<td>$u_{tttt}^n(x,t_n)$</td>
</tr>
<tr>
<td>$J^0_{n,j}$</td>
<td>$J^1_{n,j}$</td>
<td>$J^2_{n,j}$</td>
<td>$J^3_{n,j}$</td>
</tr>
<tr>
<td>$\log h J^0_{n,j}$</td>
<td>$\log h J^1_{n,j}$</td>
<td>$\log h J^2_{n,j}$</td>
<td>$\log h J^3_{n,j}$</td>
</tr>
</tbody>
</table>

The computed smoothness indicators are obtained by using the third order normalized Legendre polynomial and third order TVD Runge-Kutta scheme. We compute the indicators at different stages of the transition time and for the different values of the time step size $\tau_n$. It is important to recall that the number of cells in the first region $\Omega_U$ are increasing as time $t$ increases. In this numerical experiment for $M = 2.5$, at the beginning of the transition time, the first region $\Omega_U$ contains 19 cells and the number of cells increases as time increases and reached the number 29.

In the pictures of the smoothness indicators, Figures 7.4-7.18, the first rows have the computed solution and their derivatives $M^l_{n,j}$, $l = 0, 1, 2, 3$. The second rows have the temporal derivatives. In Figures 7.4-7.7, we can see that for the bigger time step size $\tau_n$ the temporal derivatives are increasing in a narrow sub-domain. For the bigger time step size $\tau_n$ optimal error order has been lost. The third row contains $J^l_{n,j}$, the jumps of the computed solution and their derivatives at the cell boundaries and the last row illustrates $\log h |J^{(l)}_{n,j}|$ for $l = 0, 1, 2, 3$. We saw that both the jump and temporal derivatives are bigger at the right end of the first region $\Omega_U$. It is expected that at the end of the first region, we have sharper solution. Moreover, we add more cells in the first region by constructing the new DG cells and in the process of reconstruction, we create slightly larger jumps than in the normal cells. Indeed, the bigger jump is created by space conversion from $u$-
direction to $x$-direction in the reconstructed new DG cell. However, our solution is still satisfying the numerical smoothness. The temporal derivatives are also significantly larger at the end of the first region $\Omega_U$. This indicates that numerical instability can occur if we enlarge the size of the first region $\Omega_U$ to the right end.

In the third region $\Omega_D$, (Figures 7.13-7.16) we compute the smoothness indicators in a similar way. We omit the first cell in presented figures of the smoothness indicators for the third region $\Omega_D$. 

Figure 7.4: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 0.7$

Figure 7.5: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 0.8$
Figure 7.6: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 1.0$

Figure 7.7: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 1.15$
Figure 7.8: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.0025$ and $t = 0.7$

Figure 7.9: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.0025$ and $t = 1.0$
Figure 7.10: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.0025$ and $t = 0.115$

Figure 7.11: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.00125$ and $t = 0.7$
Figure 7.12: Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.00125$ and $t = 1.0$

Figure 7.13: Smoothness Indicators on the $\Omega_D$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 0.7$
Figure 7.14: Smoothness Indicators on the $\Omega_D$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 0.8$

Figure 7.15: Smoothness Indicators on the $\Omega_D$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 1.0$
7.3 Spatial Error Analysis on the DG region

In this section, we present all computational results that support our spatial error analysis of the first region $\Omega_U$ and the third region $\Omega_D$. We compute all the constants that are used in the proofs of the spatial error analysis at different stages of the transition time and for different values of the time step size $\tau_n$. We recall that $\beta = \max |f'(u)|$ and for our numerical experiment of Burgers’ equation, we take $\beta = \max |u^c_n|$. The time step size $\tau_n$ has satisfied the strengthened CFL condition $\gamma \leq \frac{\tau_n h^{1+1/\beta}}{h^{1+1/3}}$. We take $\gamma = \frac{\tau_n h^{1+1/3}}{h^{1+1/3}}$.

Recall that the spatial error analysis uses bounds for three terms, $||\tilde{u}(t_{n+1}) - u^e(t_{n+1})||_{L^1(\Omega_U)}$, $||u^e(t_{n+1}) - u^p(t_{n+1})||_{L^1(\Omega_U)}$, and $||u^p(t_{n+1}) - u^h(t_{n+1})||_{L^1(\Omega_U)}$. We recall from Lemma 5.1 that $||\tilde{u}(t_{n+1}) - u^e(t_{n+1})||_{L^1(\Omega_U)} \leq \tau h^4 \beta \sum h \tilde{D}_{n,j}$, where

$$\tilde{D}_{n,j} = D^0_{n,j} + |D^1_{n,j}| \beta \gamma + |D^2_{n,j}| \frac{(\beta \gamma)^2}{2!} + |D^3_{n,j}| \frac{(\beta \gamma)^3}{3!}.$$  

The computed values of $\beta \sum h \tilde{D}_{n,j}$ at different stages of the transition time and for different values of the time step size $\tau_n$ are presented in Table 7.2. After multiplying by the term $\tau_n h^4$ in these constants we get the error order of $10^{-6}$ to $10^{-9}$ depending on the value $\tau_n$.

The second term used in the proofs was $||u^e(t_{n+1}) - u^p(t_{n+1})||_{L^1(\Omega_U)}$, where $u^p$ is the projection.

Figure 7.16: Smoothness Indicators on the $\Omega_D$ with $h = 0.1$, $\tau_n = 0.005$ and $t = 1.15$
Table 7.2: Constant $\beta \sum h \tilde{D}_{n,j}$ in the Upper Bound of $||u - u^e||_{(\Omega_U)}$

<table>
<thead>
<tr>
<th>$t$</th>
<th>0.7</th>
<th>0.8</th>
<th>1.0</th>
<th>1.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_n = 0.005$</td>
<td>0.4152</td>
<td>0.5317</td>
<td>1.2737</td>
<td>2.0431</td>
</tr>
<tr>
<td>$\tau_n = 0.0025$</td>
<td>0.2687</td>
<td>0.2989</td>
<td>0.7058</td>
<td>1.1665</td>
</tr>
<tr>
<td>$\tau_n = 0.00125$</td>
<td>0.2213</td>
<td>0.2371</td>
<td>0.5055</td>
<td>0.8086</td>
</tr>
</tbody>
</table>

of $u^e$ onto the polynomial space. The bound of $u^e$ is obtained by the proof of Lemma 5.11. These values are the smoothness indicators $M^0_{n,j}$, $M^1_{n,j}$, $M^2_{n,j}$, and $M^3_{n,j}$. These constants are explicitly shown in the figures of the smoothness indicators. To find the bound on $u^e$, we need to compute the constants $N^l_{n,j}$ for $l = 0, 1, \ldots, 4$. The

$$N^0_{n,j} = |M^0_{n,j}| + |M^1_{n,j}| h + |M^2_{n,j}| \frac{h^2}{2!} + |M^3_{n,j}| \frac{h^3}{3!},$$

is easy to calculate. Basic calculation gives $N^1_{n,j} = \frac{I^1_{n,j}}{1 - F_2 I^1_{n,j} \tau_n}$, where $F_2 = 1$ for Burgers’ equation. For the small value of $\tau_n$, $1 - I^1_{n,j} \tau_n$ is close to 1 and hence, $N^1_{n,j}$ is close to $I^1_{n,j}$. Similarly, we compute the constants

$$N^2_{n,j} = I^2_{n,j} e^{3 I^1_{n,j} \tau_n},$$

$$N^3_{n,j} = (I^3_{n,j} + \frac{3 (N^2_{n,j})^2}{4 N^1_{n,j}}) e^{4 N^1_{n,j} \tau_n} - \frac{3 (N^2_{n,j})^2}{4 N^1_{n,j}},$$

and

$$N^4_{n,j} \tau_n = \frac{10 N^3_{n,j} N^2_{n,j}}{5 N^1_{n,j}} (e^{5 N^1_{n,j} \tau_n} - 1).$$

We report maximum values of the constant $N^l_{n,j}$ for $l = 0, 1, \ldots, 4$ in Table 7.3 at the different time stages of the transition time and different values of the time step size $\tau_n$.

Once we calculate the values of $N^l_{n,j}, l = 0, 1, \ldots, 4$, it is trivial to calculate the constant $\frac{h^4}{4!} \sum N^4_{n,j}$. After multiplying by the factor $\tau_n h^4$, we get the error order of $10^{-6}$ to $10^{-8}$. The detailed computational result is presented in Table 7.4.

Finally, to compute the constant bound for the term $||u^p(t_{n+1}) - u^h(t_{n+1})||_{L_1(\Omega_U)}$ is relatively easy. The computed constants are reported in Table 7.5. For $\tau_n = 0.005$, the constants are much
Table 7.3: Maximum Values of the Constants $N_{n,j}^{l}$ in the First Region $\Omega_U$

<table>
<thead>
<tr>
<th>$\tau_n$</th>
<th>$N_{n,j}^{0}$</th>
<th>$N_{n,j}^{1}$</th>
<th>$N_{n,j}^{2}$</th>
<th>$N_{n,j}^{3}$</th>
<th>$N_{n,j}^{4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0.7$</td>
<td>2.0034</td>
<td>0.2703</td>
<td>1.5050</td>
<td>5.5208</td>
<td>83.37</td>
</tr>
<tr>
<td>$t = 0.8$</td>
<td>2.0036</td>
<td>0.2837</td>
<td>1.6237</td>
<td>6.2201</td>
<td>101.81</td>
</tr>
<tr>
<td>$t = 1.0$</td>
<td>2.0043</td>
<td>0.3158</td>
<td>1.9975</td>
<td>9.2146</td>
<td>184.78</td>
</tr>
</tbody>
</table>

| $\tau_n = 0.0025$, $t = 0.7$ | 2.0034 | 0.2706 | 1.5304 | 6.0334 | 92.4937 |
| $t = 0.8$ | 2.0036 | 0.2846 | 1.6566 | 6.60 | 109.68 |
| $t = 1.0$ | 2.0042 | 0.3170 | 2.0310 | 9.5283 | 193.9015 |

| $\tau_n = 0.00125$, $t = 0.7$ | 2.0034 | 0.2706 | 1.5366 | 6.1872 | 95.1507 |
| $t = 0.8$ | 2.0035 | 0.2844 | 1.6718 | 6.89 | 115.34 |
| $t = 1.0$ | 2.0043 | 0.3176 | 2.0569 | 9.8994 | 203.8235 |

Table 7.4: Constant $\frac{h}{4!} \sum N_{n,j}^{4}$ in the Upper Bound of $||u^e - u^p||_{(\Omega_U)}$

<table>
<thead>
<tr>
<th>$t$</th>
<th>0.7</th>
<th>0.8</th>
<th>1.0</th>
<th>1.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_n = 0.005$</td>
<td>0.5927</td>
<td>0.7007</td>
<td>1.1346</td>
<td>1.83</td>
</tr>
<tr>
<td>$\tau_n = 0.0025$</td>
<td>0.6297</td>
<td>0.7304</td>
<td>1.1650</td>
<td>1.8206</td>
</tr>
<tr>
<td>$\tau_n = 0.00125$</td>
<td>0.6403</td>
<td>0.7532</td>
<td>1.2023</td>
<td>1.8659</td>
</tr>
</tbody>
</table>

bigger and the error is about $10^{-5}$, whereas for $\tau_n = 0.0025$ and $\tau_n = 0.00125$, the error order is of $10^{-6}$ to $10^{-9}$.

Table 7.5: Constant $C_1 e^{\theta C_1 \tau_n / h} \left[ \frac{C_2 + \gamma h^{1/3} C_3}{C_1} \right]$ in the Upper Bound of $||u^p - u^h||_{(\Omega_U)}$

<table>
<thead>
<tr>
<th>$t$</th>
<th>0.7</th>
<th>0.8</th>
<th>1.0</th>
<th>1.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_n = 0.005$</td>
<td>343.00</td>
<td>427.39</td>
<td>914.07</td>
<td>1.6345e+03</td>
</tr>
<tr>
<td>$\tau_n = 0.0025$</td>
<td>25.7184</td>
<td>30.3826</td>
<td>63.0125</td>
<td>112.42</td>
</tr>
<tr>
<td>$\tau_n = 0.00125$</td>
<td>6.5901</td>
<td>7.6550</td>
<td>15.9980</td>
<td>28.011</td>
</tr>
</tbody>
</table>

7.4 Temporal Error Analysis on the DG region

In this section, we will discuss about the temporal error of the numerical example. The detailed derivation of the temporal error for the smooth solution can be found in (20). In our case, for the first region $\Omega_U$ and the third region $\Omega_D$, we apply the same temporal error analysis presented in (20).
Recall the notation from the theoretical section of the temporal error analysis $z = z(\tau_n)$, and $z^l = \frac{d}{d\tau}$. Lemma 6.5 has given the bound for the term $z(\tau_n)$. We compute $z^{(1)}(\tau_n)$, such that it satisfies the equation

$$
(z_t^{(1)}, v)_{\Omega_j} = (f'(z)z^{(1)}, v_x)_{\Omega_j} \\
+ f'(z(x^-_{j-1/2}))z^{(1)}(x^-_{j-1/2})v(x^+_{j-1/2}) \\
- f'(z(x^-_{j+1/2}))z^{(1)}(x^-_{j+1/2})v(x^+_{j+1/2}).
$$

(7.3)

Expanding $z^{(1)}$ by the basis function $\{\phi_{j,i} : i = 0, 1, 2, 3\}$ in each DG cell $\Omega_j$,

$$
z^{(1)}(\tau)|_{\Omega_j} = \sum_{i=0}^3 q^{(1)}_{j,i}(\tau)\phi_{j,i}.
$$

Dropping the subscript $j$ and focusing on the specific cell $\Omega_j$, we get

$$
z^{(1)} = q^{(1)}_3 \phi_3 + q^{(1)}_2 \phi_2 + q^{(1)}_1 \phi_1 + q^{(1)}_0 \phi_0.
$$

Plugging the values of $z^{(1)}$ in (7.3), yields

$$
\left(\frac{d}{dt}(q^{(1)}_3 \phi_3 + q^{(1)}_2 \phi_2 + q^{(1)}_1 \phi_1 + q^{(1)}_0 \phi_0), v\right)_{\Omega_j} \\
= \left(f'(z)(q^{(1)}_3 \phi_3 + q^{(1)}_2 \phi_2 + q^{(1)}_1 \phi_1 + q^{(1)}_0 \phi_0), v_x\right)_{\Omega_j} \\
+ f'(z(x^-_{j-1/2}))(\tilde{q}^{(1)}_3 \phi_3 + \tilde{q}^{(1)}_2 \phi_2 + \tilde{q}^{(1)}_1 \phi_1 + \tilde{q}^{(1)}_0 \phi_0)v(x^+_{j-1/2}) \\
- f'(z(x^-_{j+1/2}))(q^{(1)}_3 \phi_3 + q^{(1)}_2 \phi_2 + q^{(1)}_1 \phi_1 + q^{(1)}_0 \phi_0)v(x^+_{j+1/2}).
$$

(7.4)

Here the notation $\tilde{q}^{(1)}_i$ is denoting that values are obtained from the boundary of the cell $\Omega_{j-1}$. Substituting $v = \phi_0, \phi_1, \phi_2$, and $\phi_3$ we get,
\[
\begin{bmatrix} B \end{bmatrix} \frac{d}{dt} \begin{bmatrix} q_0^{(1)} \\ q_1^{(1)} \\ q_2^{(1)} \\ q_3^{(1)} \end{bmatrix} = \begin{bmatrix} A_1 \end{bmatrix} \begin{bmatrix} q_0^{(1)} \\ q_1^{(1)} \\ q_2^{(1)} \\ q_3^{(1)} \end{bmatrix} + \begin{bmatrix} A_2 \end{bmatrix} \begin{bmatrix} q_0^{(1)} \\ q_1^{(1)} \\ q_2^{(1)} \\ q_3^{(1)} \end{bmatrix} + \begin{bmatrix} A_3 \end{bmatrix} \begin{bmatrix} q_0^{(1)} \\ q_1^{(1)} \\ q_2^{(1)} \\ q_3^{(1)} \end{bmatrix}. \tag{7.5}
\]

To estimate \( q^{(1)} \) and hence for \( z^{(1)} \), we need to get the bound for each term of the matrices of the equation (7.5). From the computational point of view, the matrix \( B \) is not hard to calculate and is simply \( (\phi_i, \phi_j)_{\Omega_j} \). Thus, the difficulty of the computation of the matrix \( B \) depends on the choice of the basis function \( \phi_j \). In our case, we choose the normalized Legendre polynomial as our basis. Due to orthogonality of the basis, the matrix \( B \) reduces to the diagonal matrix. Moreover, orthogonality made it easier to estimate the values of \( A_1 \). We omit the detailed discussion about the impact of the choice of basis here and refer to (20). Moreover, we do not go in depth of computing all constants and refer to (20).

We simply present the graph of the temporal derivatives and shown they are bounded for the both regions \( \Omega_U \) and \( \Omega_D \). As it has shown in the pictures of the indicators in Figure 7.4-7.12, at the right end of the first region \( \Omega_U \), the temporal derivatives are increasing significantly. It is expected as the solution becomes sharper and sharper at the right end side. However, still it has maintained numerical smoothness.

In the third region \( \Omega_D \), we saw that the temporal derivatives are larger in the first couple of cells. We believe this is due to the impact of the flux being estimated by the interpolation result of the cubic front tracking solution. However, due to the fact that \( \tau_n \) is very small and multiplying the constants by \( \tau_n^4 \) leads the error estimate being reasonable. Needless to say, it has lost the numerical smoothness locally and it seems some adaptive treatment is required on the first couple of cells of the third region \( \Omega_D \). However, our numerical solution is stable and has no oscillations.
7.5 Second Numerical Example

For simplicity, we choose Burgers’ equation

\[ u_t + \left( \frac{u^2}{2} \right)_x = 0 \]  
(7.6)

with the boundary condition \( u_0(t) = 2 \) and the initial condition

\[ u_0(x) = 1 + e^{-x^2} \]  
(7.7)

We take \( \Omega = [a, b] = [0, 5] \) and use the third-degree polynomials and a third order TVD Runge-Kutta. It is not necessary to use the third-degree polynomials, but for simplicity, we use the third degree Legendre polynomials. Thus, in our numerical example \( p = 3 \) and \( k = 3 \). Similar to the first example we apply the RKDG scheme until the slope of the numerical solution reaches \(-M\).

In this example, the breaking time is

\[ T_b = \min_{x \in [a, b]} \frac{-1}{u_0'(x)} = \sqrt{\frac{e}{2}}. \]

This numerical value is approximately 1.165. We apply the algorithm presented in Chapter 3 to obtain the numerical solution. It is numerically smooth and has no numerical oscillations.

7.5.1 Error Analysis of the FT region

The error due to conversion was estimated by \( |\bar{x}(u) - x_c(u)| \leq T_1 M_4 h^4 \). The value of the constant \( T_1 = \frac{21}{1980} M^4 \). For our computation, we consider \( M = 2.333 \) and hence the maximum value of the constant \( T_1 = \frac{21}{1980} \times (2.33)^4 = 0.324 \). Moreover, in most of the cells \( M \) is actually smaller than 2.33 and taking the fourth power makes it much smaller. Recall that the constant \( M_4 \) has different values on the different converted cells. Also, in the denominator of the bound of the constant \( M_4 \) has the term \( \bar{u}'(x, t_n) \). When \( |\bar{u}'(x, t_n)| = M > 1 \), the constant \( M_4 \) has a much smaller upper bound. This justifies that the upper bound in the error estimate is reasonable and our
converted solution $x_c(u)$ is of high order.

For our experiment of Burgers’ equation, $M_5 = |f^{(5)}(u)| = 0$. Thus, the constant of the upper bound of the Cubic Front Tracking error estimate is zero. Consequently, our Cubic Front Tracking solution has no more error after the conversion is done. Thus, the numerical solution is of high order in the FT region.

7.5.2 Error Analysis of the DG region

Similar to the discussion we made on the first numerical experiment, our error bound for both spatial error estimation and temporal error estimation will be based on computed smoothness indicators. We have the first region and third region where we use the RKDG method. On those regions, we compute the smoothness indicators separately. We saw that at the end of the first region indicators are increasing rapidly in a narrow sub-domain. This was expected as the solution curve became very steep in that part and to avoid possible numerical instability, we use the Cubic Front Tracking scheme in the second region.

Recall that for the spatial error analysis for the first region; we have bounds for three terms, $||\tilde{u}(t_{n+1}) - u^e(t_{n+1})||_{L_1(\Omega_U)}$, $||u^e(t_{n+1}) - u^p(t_{n+1})||_{L_1(\Omega_U)}$, and $||u^p(t_{n+1}) - u^h(t_{n+1})||_{L_1(\Omega_U)}$. All bounds for these estimates depend on computed smoothness indicators. Since we present detailed values of these bound for the first numerical experiments, we skip these detailed values here. We simply illustrate Figures 7.17-7.18, of computed smoothness indicators to justify that our error bounds are reasonable and our computed solution is of high order.

Next, the temporal error bound written in the temporal error analysis section depends on temporal smoothness indicators. Again, we don’t repeat the meticulous calculation illustrated in the first numerical example and illustrates in Figure 7.17 and in Figure 7.18 of the temporal smoothness indicators. We saw that at the first cell of the third region in Figure 7.17 and in Figure 7.18, temporal derivatives are higher and temporal smoothness seems lost locally. We believe that some adaptive treatment is needed locally. However, $\tau_n$ is very small, and error bound has the term $\tau_n^4$. So, our numerical solution is stable and has no oscillations.
Figure 7.17: Second Numerical Example Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.00125$ and $t = 1.0$

Figure 7.18: Second Numerical Example Smoothness Indicators on the $\Omega_U$ with $h = 0.1$, $\tau_n = 0.0025$ and $t = 1.0$
CHAPTER 8 CONCLUSIONS

The error analysis for the RKDG method to solve the one-dimensional conservation law for the case of smooth solutions has been done in (23) and for the case of having a fully developed shock has been done in (22). In this dissertation, we have completed the error analysis for the Cubic Front Tracking and RKDG method for the case between (23) and (22). Thus, our work fills the gap of the case of the smooth solution and the case of having a fully developed shock. This work completes the whole picture of the error analysis for solving the one-dimensional nonlinear conservation law.

In the proofs presented in Chapter 4, we use the $L_{\infty}$-norm as the Cubic Front Tracking scheme is used to evolve the solution. On the other hand, for the DG region, we use the $L_1$-norm to take the advantage of $L_1$-contraction property (21). When the solution gets sharper and sharper, there is no hope to obtain the high order solution by the RKDG method. Thus, we convert the computed solution $u_c^n$ to $\tilde{x}(u)$ and incorporate the Cubic Front Tracking scheme. We implement the algorithm presented in Chapter 3 to produce the high order solution during the shock formation.

We use the cubic polynomial in the space conversion. However, our idea can be extended for the case of any polynomial of degree $p$ with $p \geq 3$. Since we consider only the “s-shape” solution, there is significant work still needed to handle the case of having any shape of the solution during the shock formation. If the shock appearance time is far apart, then our algorithm to compute the “s-shape” solution can be still applied for multiple times for the multiple shocks. However, more work is needed to handle the case of multiple shocks appearing at the same time. Needless to say, work remains to extend the result to higher dimensional conservation laws. Moreover, our work did not address the famous opinion (15) to use the shock capturing method, instead of the front tracking method to estimate the error. However, our work has made a significant step to compute the high order solution during the shock formation. It has exemplified how the Cubic Front Tracking and RKDG scheme can be used to produce the high order solution even in the case of further sharpening and the formation of a shock.
BIBLIOGRAPHY


York, 1999.

Galerkin finite element method for conservation laws III: One dimensional systems. *J. Com-


