Wave Resolution Properties and Weighted Essentially Non-Oscillatory Limiter for Discontinuous Galerkin Methods

by

Xinghui Zhong

B.S., University of Science and Technology of China, Hefei, China, 2007

M.S., Brown University, RI, 2009

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2012

This dissertation by Xinghui Zhong is accepted in its present form
by Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date_____ _____

Chi-Wang Shu, Ph.D., Advisor

Recommended to the Graduate Council

Date_____ _____

Johnny Guzmán, Ph.D., Reader

Date_____ _____

Jan Hesthaven, Ph.D., Reader

Approved by the Graduate Council

Date_____ _____

Sheila Bonde, Dean of the Graduate School

Vitae

Biographical Information

Born on August 7th, 1984 in *Puyang, China*

Education

Brown University, Providence, RI, USA
Ph.D. in Applied Mathematics, May 2012 (expected)
Advisor: Chi-Wang Shu

Brown University, Providence, RI, USA
M.S. in Applied Mathematics, May 2009

University of Science and Technology of China, Hefei, China
B.S. with distinction (top 3%), in Mathematics and Applied Mathematics, 2007

Research Experience

Brown University, Providence, RI, USA, 2009-present
Research Assistant,
Dissertation: Wave resolution properties and weighted ENO limiters for discontinuous Galerkin methods.

Institute for Mathematics and its Applications (IMA), Minneapolis, MN, USA, August, 2009
Participated a 10-day workshop on Mathematical Modelling in Industry XIII.
Conducted research on fast computational methods for reservoir flow models under the direction of Dr. Robert Shuttleworth from ExxonMobil Upstream Research.

University of Science and Technology of China, Hefei, China, July-October 2005
Participated the "Undergraduate Research Project" program.
Conducted research on boundary evaluation of interval vector operations under the direction of Prof. Jiansong Deng.

Publications/Preprints

- X. Zhong and C.-W. Shu, Numerical resolution of discontinuous Galerkin methods for time dependent wave equations, *Computer Methods in Applied Mechanics and Engineering*, v200 (2011), pp. 2814-2827.
- X. Zhong and C.-W. Shu, A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods, submitted to *Journal of Computational Physics*.
- J. Zhu, X. Zhong, C.-W. Shu and J.X. Qiu, Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes, submitted to SIAM Journal on Scientific Computing.
- W. Guo, X. Zhong and J.-M. Qiu, Superconvergence of discontinuous Galerkin method: eigen-structures analysis based on Fourier approach, in preparation.

Teaching Experience

Brown University
Teaching Assistant
Methods of Applied Mathematics - II, Spring 2009
Methods of Applied Mathematics - I, Fall 2008

Math Resource Center Tutor, 2007-2011

Certificate III, Sheridan center for Teaching and Learning, 2011-2012

Certificate I, Sheridan Center for Teaching and Learning, 2009- 2010

Awards

Brown University
Dunmu Ji Award, 2012 AMS Graduate Student Travel Grant, January 2012
NSF Travel Support, June 2011
Graduate Student Fellowships in Fall 2007, Spring 2008, Fall 2011

University of Science and Technology of China
Outstanding Undergraduate Research Project Award, 2005
Cyrus Tang Scholarship, 2004-2006
Outstanding Freshman Scholarship, 2003

Acknowledgements

First and foremost, I owe my deepest gratitude to my advisor Prof. Chi-Wang Shu for his patience, inspiration and enthusiasm. His wide knowlege has helped me in all the time of research and writing of this thesis whilst allowing me the room to work in my own way. His guidance has helped me to see life and science in their full depth. His detailed and constructive comments have had a remarkable influence on my entire study. One simply could not wish for a better or friendlier supervisor. I also wish to express my warm and sincere thanks to his wife Din-Sui Loh for he ever smiling face and encouraging words.

It is an honor for me to thank the members of my committee, Prof. Jan Hesthaven and Prof. Johnny Guzman for their detailed review, constructive criticism and excellent advice during the preparation of this thesis.

I am grateful to the Division of Applied Mathematics and graduate school of Brown University for generous financial support in the past five years.

I am indebted to many of my close friends with whom I have been able to share the joys during the five long years of my PhD studies. I cannot thank enough my best friend in high school (Xiaoping Cheng, Suhong Yang) and my best friends as an

undergraduate (Liquan Huang, Han Lv) for all the emotional support, comraderie, entertainment, and caring they provided. Many friends at Brown University have helped me a lot, too. It is a pleasure to thank my former roommates and friends (Yanan Liu, Xiaomin Ma) and my present roommate Luan Lin, whose words and encouragements have always been very special to me. I am thankful to all my previous and current group members for methodically looking over my work and helping me at every instance.

Last but definitely not the least, I owe my loving thanks to my parents for giving birth to me at the first place and supporting me spiritually throughout my life. My special gratitude is due to my brothers for their unflagging love. I could never finish this dissertation without them.

I finish with a final silence of gratitude for my life.

Abstract of "Wave Resolution Properties and Weighted Essentially Non-Oscillatory Limiter for Discontinuous Galerkin Methods" by Xinghui Zhong, Ph.D., Brown University, May 2012

This dissertation presents wave resolution properties and weighted essentially non-oscillatory limiter for discontinuous Galerkin methods solving hyperbolic conservation laws.

In this dissertation, using Fourier analysis, we provide a quantitative error analysis for the semi-discrete DG method applied to time dependent linear convection equations with periodic boundary conditions. We apply the same technique to show that the error is of order $k+2$ superconvergent at Radau points on each element and of order $2k+1$ superconvergent at the downwind point of each element, when using piecewise polynomials of degree k. An analysis of the fully discretized approximation is also provided. We compute the number of points per wavelength required to obtain a fixed error for several fully discrete schemes.

We also investigate a simple limiter using weighted essentially non-oscillatory (WENO) methodology for the Runge-Kutta discontinuous Galerkin (RKDG) methods solving conservation laws, with the goal of obtaining a robust and high order limiting procedure to simultaneously achieve uniform high order accuracy and sharp, non-oscillatory shock transitions. The idea of this limiter is to reconstruct the entire polynomial, instead of reconstructing point values or moments in the classical WENO reconstructions. That is, the reconstruction polynomial on the target cell is a convex combination of polynomials on this cell and its neighboring cells and the nonlinear weights of the convex combination follow the classical WENO procedure. The main advantage of this limiter is its simplicity in implementation, especially for multi-dimensional meshes.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Discontinuous Galerkin (DG) methods are a class of finite element methods using completely discontinuous basis functions, which are usually chosen as piecewise polynomials.

The first discontinuous Galerkin method was introduced by Reed and Hill [53] in 1973, to solve the neutron transport equation, i.e. a time independent linear hyperbolic equation. The type of DG methods we will discuss in this dissertation is the Runge-Kutta discontinuous Galerkin (RKDG) method [24, 23, 21, 20, 25], which is using explicit and nonlinearly stable high order Runge-Kutta method for time discretization and the DG method for space discretization. This method has several advantages, such as local conservation, the allowance of arbitrary triangulation, excellent parallel efficiency, the capability in h-p adaptivity and the ability to sharply capture discontinuities (especially contact discontinuities) in the solution. Furthermore, it has certain superconvergence properties.

DG methods are known to provide good wave resolution properties, especially for long time simulation. The critical issue for wave phenomena is how many points are needed to resolve a wave, namely to make sure that the error is smaller than a given tolerance. In 1972, Kreiss and Oliger [42] introduced the error analysis of the wave problem using the finite difference method and developed the estimates of the necessary points per wavelength. 1974, Swartz and Wendroff [65] extended these ideas to the finite element method. In this dissertation, we combine and extend these ideas to DG methods. Using Fourier analysis, we develop a quantitative error estimate for DG methods and an estimate of the number of points per wavelength required to obtain a fixed error. The quantitative analysis provided in this dissertation can be used in the application to guide the choice of mesh size and polynomial degree in the DG method. Moreover, by investigating the quantitative error at Radau points using the same technique, we show that the errors at Radau points are of order $k + 2$

with the exception of the Radau point at the downwind end of the element, which is of order $2k+1$, when using the DG method with piecewise polynomials of degree k.

DG methods have a lot of advantages, one of which is that they can compute solutions to hyperbolic conservation laws, which are either smooth or have weak shocks and other discontinuities, without further modification. However, if the discontinuities are strong, the polynomial representation will generate significant oscillations and even nonlinear instability. To avoid such difficulties, a slope limiter is borrowed from the finite volume methodology to control the numerical solution. Many such limiters exist in the literature, all with advantages and disadvantages. It is no exaggeration to state that the design of good, robust limiters remains one of the bottlenecks to the development of DG methods for solving conservation laws.

In this dissertation, we use the WENO methodology to design a new and simpler limiter for the RKDG methods. The idea of this simple WENO limiter is that the reconstruction polynomial on the target cell is a convex combination of the polynomials on this cell and its immediate neighboring cells, with the nonlinear weights of the linear combination following the classical WENO procedure. Since this procedure uses information only from immediate neighbors and simple positive linear weights, it is significantly simpler to implement than previous WENO type limiters for RKDG methods.

This dissertation is organized as follows: we first present an overview of DG methods in Chapter 2. Then we investigate the wave resolution property and superconvergence property of DG methods in Chapter 3. In Chapter 4, we design a new and simple WENO limiter for RKDG methods. Concluding remarks are given in Chapter 5.

Chapter 2

Review of Discontinuous Galerkin Methods for Conservation Laws

The discontinuous Galerkin method was first designed as an effective numerical methods for solving hyperbolic conservation laws, which may have discontinuous solutions. In this chapter, we will review the algorithm formulation, error estimates and minmod type limiters for the DG method solving hyperbolic conservation laws.

2.1 Algorithm Formulation

In this section, we give an overview of the algorithm formulation of the RKDG method solving one-dimensional conservation law

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

To define the DG method for (2.1), we consider a partition of the computational domain $[a, b]$ in N cells as follows

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = b.$$

We denote

$$I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}], \quad x_j = \frac{1}{2}\left(x_{j+\frac{1}{2}} + x_{j-\frac{1}{2}}\right)$$

as the cell and cell center, respectively. We again denote

$$\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}; \quad h = \max_{1\le j\le N} \Delta x_j.$$

We assume that the mesh is regular, namely there is a constant $c > 0$ independent of h such that

$$\Delta x_j \ge ch, \quad 1 \le j \le N.$$

Define the approximation space as

$$V_h^k = \left\{ v : v|_{I_j} \in P^k(I_j);\ 1 \le j \le N \right\}$$

where $P^k(I_j)$ denotes the set of polynomials of degree up to k defined on the cell I_j. The semi-discrete DG method for solving (2.1) is defined as follows: find the unique function $u_h = u_h(t) \in V_h^k$ such that, for $j = 1, \cdots, N$,

$$\int_{I_j} (u_h)_t v_h \, dx - \int_{I_j} f(u_h)(v_h)_x \, dx + \hat{f}_{j+\frac{1}{2}} v_h(x^-_{j+\frac{1}{2}}) - \hat{f}_{j-\frac{1}{2}} v_h(x^+_{j-\frac{1}{2}}) = 0 \qquad (2.2)$$

holds for all test functions $v_h \in V_h^k$. $\hat{f}_{j+\frac{1}{2}}$ is the numerical flux, which is a single valued function defined at the cell interface and in general depends on the values of the numerical solution u_h from both sides of the interface

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}\left(u_h(x^-_{j+\frac{1}{2}}, t), u_h(x^+_{j+\frac{1}{2}}, t) \right).$$

We use monotone numerical fluxes from finite difference and finite volume schemes for solving conservation laws, which satisfy the following conditions:

- Consistency: $\hat{f}(u, u) = f(u)$.
- Continuity: $\hat{f}(u^-, u^+)$ is at least Lipschitz continuous with respect to both arguments u^+ and u^-.
- Monotonicity: $\hat{f}(u^-, u^+)$ is a non-decreasing function of its first argument u^- and a non-increasing function of its second argument u^+. Symbolically $\hat{f}(\uparrow, \downarrow)$.

Here and below u^-, u^+ denote the left and right limits of the function u at the cell interface, respectively.

There is a class of monotone fluxes called upwind fluxes, which satisfies

$$\hat{f}(u^-,u^+)=\begin{cases} f(u^-), & \text{if } f'(u)\geq 0,\ \ \forall u\in[\min(u^-,u^+),\max(u^-,u^+)],\\ \hat{f}(u^+), & \text{if } f'(u)<0,\ \ \forall u\in[\min(u^-,u^+),\max(u^-,u^+)].\end{cases}$$

Well know monotone fluxes include the Lax-Friedrichs flux

$$\hat{f}^{LF}\left(u^-,u^+\right)=\frac{1}{2}\left(f\left(u^-\right)+f\left(u^+\right)-\alpha\left(u^+-u^-\right)\right),\quad \alpha=\max_u\left|f'(u)\right|;$$

the Godunov flux

$$\hat{f}^{God}\left(u^-,u^+\right)=\begin{cases}\min_{u^-\leq u\leq u^+} f(u), & \text{if } u^-<u^+,\\ \max_{u^+\leq u\leq u^-} f(u), & \text{if } u^-\geq u^+;\end{cases}$$

and the Engquist-Osher flux

$$\hat{f}^{EO}=\int_0^{u^-}\max\left(f'(u),0\right)du+\int_0^{u^+}\min\left(f'(u),0\right)du+f(0).$$

Clearly, $\hat{f}^{God}$ and $\hat{f}^{EO}$ are upwind fluxes. For more details about monotone fluxes, see, e.g., [45].

2.2 Time Discretization

The semi-discrete scheme (2.2) can be written as

$$u_t=L(u),$$

where $L(u)$ is the spatial discretization operator. To discretize the time variable, we often use a class of high order nonlinearly stable Runge-Kutta methods, which are convex combinations of first order forward Euler steps. Therefore they maintain strong stability properties of the forward Euler step in any semi-norm.

These methods were first developed in [62, 58], and later generalized in [31, 32]. The most popular scheme in this class is the following total variation diminishing (TVD) third order Runge-Kutta method [62]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}). \end{aligned} \tag{2.3}$$

Other TVD, or strong stability preserving (SSP) time discretizations can of course also be used. For details, see the survey paper [60] and the review paper [32].

2.3 Error Estimates for Smooth Solutions

If we assume the exact solution of (2.1) is smooth, we can obtain optimal L^2 error estimates. The first error estimate of the DG method of Reed and Hill [53] was done by Lesaint and Raviart [44] in 1974. They proved that this method is convergent with optimal order of accuracy, namely $O\left(h^{k+1}\right)$, in L^2 norm, when piecewise tensor product polynomials of degree k are used as basis functions. Later in 1986, Johnson and Pitkäranta [39] proved that L^2 error estimate of this method is $O\left(h^{k+\frac{1}{2}}\right)$ where k is again the polynomial degree and h is the mesh size, when the solution is sufficiently smooth for arbitrary meshes.

Such error estimates can be obtained for the general conservation law (2.1):

Proposition 2.3.1. The solution u_h of the DG scheme (2.2) for (2.1) with a sufficiently smooth solution u satisfies the following error estimate

$$\|u - u_h\| \leq Ch^{k+1}, \quad \text{if } \hat{f} \text{ is an upwind flux;} \tag{2.4}$$

$$\|u - u_h\| \leq Ch^{k+\frac{1}{2}}, \quad \text{if } \hat{f} \text{ is general monotone flux,} \tag{2.5}$$

where C depends on u and its derivatives but is independent of h.

For the proof of Proposition 2.3.1, see [71, 72]. In actual numerical computations, one almost always observe the optimal $O(h^{k+1})$ accuracy.

2.4 Minmod Limiters

In practice, especially for problems containing strong discontinuities, the DG scheme is significant oscillatory near discontinuities and even nonlinear instable. We often need to apply nonlinear limiters to control these oscillations. In this section, we discuss the minmod type limiter [24, 23, 21, 20, 25].

For simplicity, we consider the forward Euler time discretization of the semi-discrete scheme (2.2). Starting from a solution $u_h^n \in V_h^k$ at time level n (for the initial condition, u_h^0 is taken as the L^2 projection of the analytical initial condition into V_h^k). We would like to "limit" it to obtain a new function $u_h^{n,new}$ before advancing it to next time level. That is: to find $u_h^{n+1} \in V_h^k$, such that, for $j = 1, \cdots, N$,

$$\int_{I_j} \frac{u_h^{n+1} - u_h^{n,new}}{\Delta t} v_h \, dx - \int_{I_j} f(u_h^{n,new})(v_h)_x \, dx + \hat{f}_{j+\frac{1}{2}}^{n,new} v_h(x_{j+\frac{1}{2}}^-) - \hat{f}_{j-\frac{1}{2}}^{n,new} v_h(x_{j-\frac{1}{2}}^+) = 0$$

holds for all test functions $v_h \in V_h^k$.

The limiting procedure to go from u^n to $u^{n,new}$ should satisfy the following conditions:

- The cell averages of $u_h^{n,new}$ and u_h^n are the same. That is to ensure the conservation.
- In the smooth regions this limiter does not change the solution, $u_h^{n,new}(x) = u_h^n(x)$.

There are many limiters discussed in the literature. We provide here an overview of the minmod limiter [23] in the one-dimensional scalar case. Denote the cell average of the solution u as

$$\bar{u}_j = \frac{1}{\Delta x_j} \int_{I_j} u_h \, dx \tag{2.6}$$

and further denote

$$\tilde{u}_j = u^-_{j+\frac{1}{2}} - \bar{u}_j, \quad \tilde{\tilde{u}}_j = \bar{u}_j - u^+_{j-\frac{1}{2}}. \tag{2.7}$$

$\tilde{u}_j$ and $\tilde{\tilde{u}}_j$ are modified by the minmod limiter [33],

$$\tilde{u}_j^{(mod)} = m\left(\tilde{u}_j, \Delta_+ \bar{u}_j, \Delta_- \bar{u}_j\right), \quad \tilde{\tilde{u}}_j^{(mod)} = m\left(\tilde{\tilde{u}}_j, \Delta_+ \bar{u}_j, \Delta_- \bar{u}_j\right), \tag{2.8}$$

where

$$\Delta_+ \bar{u}_j = \bar{u}_{j+1} - \bar{u}_j, \quad \Delta_- \bar{u}_j = \bar{u}_j - \bar{u}_{j-1},$$

and the minmod function m is defined by

$$m(a_1, \cdots, a_l) = \begin{cases} s \min_{1 \le i \le l} |a_i| & \text{if } s = sign(a_1) = \cdots = sign(a_l), \\ 0, & \text{otherwise.} \end{cases} \tag{2.9}$$

Then the new point values at the cell boundary are

$$u_h^{(mod)}\left(x^-_{j+\frac{1}{2}}\right) = \bar{u}_j + \tilde{u}_j^{(mod)}, \quad u_h^{(mod)}\left(x^+_{j-\frac{1}{2}}\right) = \bar{u}_j - \tilde{\tilde{u}}_j^{(mod)} \tag{2.10}$$

by definition (2.7). The limited solution $u_h^{(mod)}$ is then reconstructed to maintain the old cell average (2.6) and the new point values (2.10). Clearly, this reconstruction is unique for $k \le 2$. For $k > 2$, we have extra freedom in obtaining $u_h^{(mod)}$. It has been proved that the DG scheme with this "pre-processing" limiter is TVD by using Harten's Lemma [33].

We now need to verify that this limiter does not affect accuracy in smooth regions. After a simply Taylor expansion, we have

$$\tilde{u}_j = \frac{1}{2} u_x(x_j)\Delta x_j + O(h^2), \quad \tilde{\tilde{u}}_j = \frac{1}{2} u_x(x_j)\Delta x_j + O(h^2),$$

$$\Delta_+ \bar{u}_j = \frac{1}{2} u_x(x_j)(\Delta x_j + \Delta x_{j+1}) + O(h^2), \quad \Delta_- \bar{u}_j = \frac{1}{2} u_x(x_j)(\Delta x_j + \Delta x_{j-1}) + O(h^2).$$

Clearly, if we are in a smooth and monotone domain, i.e. $u_x(x_j)$ is away from zero, the first argument in the minmod function (2.8) is of the same sigh as the second and third argument and is smaller in magnitude, when h is small. Therefore, the modified values $\tilde{u}_j^{(mod)}$ and $\tilde{\tilde{u}}_j^{(mod)}$ in (2.8) will take the unmodified values $\tilde{u}_j$ and $\tilde{\tilde{u}}_j$, respectively. That is, this limiter does not affect accuracy in smooth monotone regions. However, from numerical results, this TVD limiter does kill accuracy at smooth extrema. Therefore, in practice, we often use a total variation bounded

(TVB) modified minmod limiter [57]

$$\tilde{m}(a_1, \cdots, a_l) = \begin{cases} a_1 & \text{if } |a_1| \leq Mh^2, \\ m(a_1, \cdots, a_l), & \text{otherwise,} \end{cases} \tag{2.11}$$

where the TVB parameter M depends on the solution of the problem. For scalar problems it is possible to estimate M by the initial condition (M is proportional to the second derivative of the initial condition at smooth extrema); however it is more difficult to estimate M for the system case. If M is chosen too small, accuracy may degenerate at smooth extrema of the solution; however, if M is chosen too large, oscillations will appear. For more details, see [23].

2.5 Numerical Examples

In this section, we demonstrate the performance of the DG method, using the linear version of (2.1):

$$u_t + u_x = 0, \quad 0 \leq x \leq 2\pi \tag{2.12}$$

with the initial condition $u(x, 0) = \sin x$ and periodic boundary conditions. The exact solution of (2.12) is

$$u(x, t) = \sin(x - t).$$

The monotone flux is taken as the simple upwind flux $\hat{f}(u^-, u^+) = u^-$. We use the 3rd order RK method discussed in Section 2.1 for time discretization and DG methods with polynomials of degree k for space discretization. Table 2.1 lists the CFL number, which are chosen according to the CFL table in [26]. For the P^k case

with $k \geq 3$, Δt is further reduced in the accuracy test. Table 2.2 lists the errors and

Table 2.1: The CFL numbers for polynomials of degree k

k	0	1	2	3	4	5	6
RK3	1.20	0.40	0.20	0.14	0.08	0.06	0.05

orders of the accuracy for the RKDG method with P^k polynomials, for $0 \leq k \leq 6$. We observe that the error is of order $k+1$. Note here we collect L^2, L^1, L^∞ errors. For the error tables in the following chapters, we only collect L^1 and L^∞ errors, since

$$\begin{aligned}
\|f\|_{L^1} &= \frac{1}{2\pi}\int_0^{2\pi} |f(x)|dx \\
&\leq \frac{1}{2\pi}\sqrt{\int_0^{2\pi} |f(x)|^2 dx}\sqrt{\int_0^{2\pi} 1 dx} \\
&= \sqrt{\frac{1}{2\pi}\int_0^{2\pi} |f(x)|^2 dx} \\
&= \|f\|_{L^2} \\
&\leq \|f\|_{L^\infty}.
\end{aligned}$$

For more numerical examples about DG methods and minmod limiters, see, e.g. [24, 23, 21, 20, 25].

Table 2.2: $u_t + u_x = 0$ with $u(x,0) = \sin x$ at $t = 2\pi$ with RKDG method

	N	L^2 error	order	L^1 error	order	L^∞ error	order
P^0	20	4.50E-01		4.05E-01		6.43E-01	
	40	2.78E-01	0.70	2.50E-01	0.70	3.98E-01	0.69
	80	1.55E-01	0.84	1.40E-01	0.84	2.22E-01	0.84
	160	8.25E-02	0.91	7.42E-02	0.91	1.18E-01	0.92
	320	4.25E-02	0.96	3.83E-02	0.96	6.05E-02	0.96
P^1	10	2.41E-02		1.99E-02		4.44E-02	
	20	4.56E-03	2.40	3.76E-03	2.40	9.84E-03	2.17
	40	1.02E-03	2.16	8.20E-04	2.20	2.90E-03	1.76
	80	2.46E-04	2.05	1.91E-04	2.10	7.79E-04	1.90
	160	6.07E-05	2.02	4.61E-05	2.05	2.01E-04	1.95
P^2	10	8.63E-04		7.28E-04		2.33E-03	
	20	1.05E-04	3.04	8.70E-05	3.06	3.05E-04	2.94
	40	1.30E-05	3.01	1.07E-05	3.02	3.86E-05	2.98
	80	1.62E-06	3.00	1.33E-06	3.01	4.87E-06	2.99
	160	2.02E-07	3.00	1.66E-07	3.00	6.10E-07	3.00
P^3	10	8.12E-05		6.90E-05		1.71E-04	
	20	5.08E-06	4.00	4.31E-06	4.00	1.08E-05	3.98
	40	3.17E-07	4.00	2.69E-07	4.00	6.82E-07	3.99
	80	1.98E-08	4.00	1.68E-08	4.00	4.27E-08	4.00
	160	1.24E-09	4.00	1.05E-09	4.00	2.67E-09	4.00
P^4	10	9.27E-06		8.33E-06		1.32E-05	
	20	2.91E-07	5.00	2.61E-07	4.99	4.16E-07	4.99
	40	9.09E-09	5.00	8.19E-09	5.00	1.29E-08	5.01
	80	2.85E-10	5.00	2.56E-10	5.00	4.05E-10	4.99
P^5	10	2.46E-06		2.21E-06		3.51E-06	
	20	3.85E-08	6.00	3.46E-08	6.00	5.55E-08	5.98
	30	3.38E-09	6.00	3.04E-09	6.00	4.86E-09	6.01
	40	6.01E-10	6.00	5.41E-10	6.00	8.68E-10	5.99
	50	1.58E-10	5.99	1.42E-10	5.99	2.28E-10	6.00
P^6	10	8.93E-07		8.04E-07		1.26E-06	
	20	6.99E-09	7.00	6.29E-09	7.00	9.88E-09	7.00
	30	4.09E-10	7.00	3.68E-10	7.00	5.79E-10	7.00
	40	5.50E-11	6.98	4.95E-11	6.98	7.77E-11	6.98

Chapter 3

Numerical Resolution of Discontinuous Galerkin Methods for Time Dependent Wave Equations

3.1 Introduction

In this chapter, we consider the following time dependent linear wave problem

$$\begin{aligned} u_t + a u_x &= 0, \quad x \in [0, 2\pi],\ t > 0 \\ u(x, 0) &= e^{i\omega x},\ x \in [0, 2\pi] \end{aligned} \tag{3.1}$$

with periodic boundary conditions, where a is the phase speed (for simplicity, assume that $a > 0$), $i = \sqrt{-1}$ and ω is the wave number (for convenience, assume that $\omega > 0$). We develop a quantitative error analysis for the discontinuous Galerkin (DG) solution via Fourier analysis and study the superconvergence property of the DG solution. Because the equation (3.1) as well as the DG scheme are linear, a general L^2 initial condition can be written as a sum of the simple waves $e^{i\omega x}$ (Fourier series) and the numerical solution for such general initial condition is just a superposition of the numerical solution of the equation (3.1) with the simple wave initial condition. The results of this chapter is therefore applicable to such general initial conditions when we have a clear understanding of the wave numbers we would like to resolve. The quantitative analysis provided in this chapter can be used to guide the choice of mesh size and polynomial degree in the DG method, when we are solving a linear wave equation, and would like to resolve the first k waves, to a given threshold of error, up to a certain time t. The results of this chapter is also valid for one dimensional linear hyperbolic system, and can provide useful guidelines when the DG method is used to solve more complicated linear and nonlinear wave equations.

The superconvergence behavior of the classical finite element method has been analyzed for many years. For example, in [27], Douglas and Dupont showed that for a general class of two-point boundary value problems, the rate of convergence at the

mesh points is of order $2k$ when polynomials of degree k are used. In [10], Bakker proved that the C^0 Galerkin solution of a two-point boundary value problem using piecewise P^k polynomials, has superconvergence of order $2k$ at the knots and of order $k+2$ at the Lobatto points of each segment, and the gradient is superconvergent of order $k+1$ at the zeros of a Legendre polynomial shifted to the elements of the partition. In [11], the same author proved that for two classes of Galerkin methods: the Ritz-Galerkin method for $2m$-th order self-adjoint boundary value problems and the collocation method for arbitrary m-th order boundary value problems, the solution exhibits superconvergence of order $k+2$ at the zeros of the Jacobi polynomial $P_n^{m,m}(\sigma)$ shifted to the elements of the partition, and the derivative of the solution is superconvergent of order $k+1$ at the zeros of the Jacobi polynomial $P_{n+1}^{m-1,m-1}(\sigma)$ shifted to the elements, where $n = k+1-2m$ and k is the degree of the finite element space.

Based on the results of Adjerid et al. [1] for singularly-perturbed parabolic systems, Biswas et al. [13] used the assumption that the DG solutions of hyperbolic conservation laws using k-degree polynomial approximation exhibit superconvergence at the roots of Radau polynomial of degree $k+1$ (Radau points), to construct a posteriori estimate of spatial discretization error.

Adjerid et al. [2] proved that the DG solution of the ordinary differential equation (ODE) $u' - f(u) = 0$ is superconvergent of order $2k+1$ at the downwind end of each element while maintaining an order of $k+2$ at the remaining Radau points. Numerical examples for the partial differential equations (PDEs) were also shown without analysis. Later, these results were extended to two-dimensional problems on rectangular meshes [4] and nonlinear hyperbolic problems [5]. Castillo [15] investigated the existence of superconvergent points for DG methods applied to elliptic problems and showed that on each element the k-degree local discontinuous Galerkin

(LDG) solution gradient is superconvergent of order $k+1$ at the shifted roots of the k-degree Legendre polynomial. For these cases, the model problems are independent of time.

In [3], Adjerid et al. showed that the LDG solutions of convection-dominated problems are superconvergent of order $k+2$ at the shifted Radau points on each element. For diffusion-dominated problems, the derivative of the LDG solution is superconvergent of order $k+2$ at Radau points. Later, Adjerid et al. proved that the DG solution is superconvergent of order $k+2$ at Radau points for linear symmetric hyperbolic systems [7] and for linear symmetrizable hyperbolic systems [6], when $t = O(1)$. In this chapter, by investigating the quantitative error at Radau points using Fourier analysis, we show that, even for t greater than $O(1)$, the errors at Radau points are of order $k+2$ with the exception of the Radau point at the downwind end of the element, which is of order $2k+1$, when using DG method with piecewise polynomials of degree k. See, for example [40, 16, 17, 18, 19, 22] for more superconvergence results of the DG method.

3.2 Motivation

This work was motivated by [42], where the error analysis is given using the finite difference method. We first provide an overview of this method solving the linear wave problem (3.1). The exact solution of (3.1) is

$$U(x,t) = e^{i\omega(x-at)}. \tag{3.2}$$

Assume a uniform grid with $\Delta x = \frac{2\pi}{N}$. Using the following second order central

finite difference method to approximate the spatial derivative in (3.1)

$$D_0 u = \frac{u_{j+1} - u_{j-1}}{2\Delta x},$$

yields a semi-discrete version of (3.1) with a system of differential-difference equations given by

$$\begin{aligned} \frac{du_j(t)}{dt} + a\frac{u_{j+1}(t) - u_{j-1}(t)}{2\Delta x} &= 0, \\ u_j(0) &= e^{i\omega x_j}. \end{aligned} \tag{3.3}$$

It is easy to compute the solution of (3.3), which is

$$u_j(t) = e^{i\omega(x_j - ct)}$$

with the numerical phase speed

$$c = \frac{a\sin\xi}{\xi}, \quad \xi = \omega\Delta x.$$

By Taylor expansion, the leading term in the error between the exact solution $U(x,t)$ and the approximate solution $u(x,t)$ is

$$e = \|U - u\|_\infty := \max_{0\le j\le N} |U(x_j,t) - u_j(t)| \simeq \frac{\omega a t \xi^2}{6}. \tag{3.4}$$

It is evident that the error is a function of time. Under the assumption of periodicity, the important quantity is not the time elapsed, but rather the number of time periods, which is denoted by q, that is,

$$q = \frac{\overline{t}}{2\pi}, \quad \overline{t} = \omega a t. \tag{3.5}$$

Now the critical issue is: how many grid points are needed to resolve a wave, to make sure that the error is smaller than a given tolerance ε? We introduce the number of points per wavelength, M, which is given by

$$M = \frac{N}{\omega} = \frac{2\pi}{\xi}. \tag{3.6}$$

Note that M has a theoretical minimum of 2, since it takes a minimum of two points per wavelength to uniquely specify a wave.

Rewriting the error (3.4) in terms of M and q, yields

$$e \simeq \frac{\pi q}{3}\left(\frac{2\pi}{M}\right)^2. \tag{3.7}$$

Then, the lower bound on M

$$M \geq 2\pi\left(\frac{\pi}{3}\right)^{1/2}\left(\frac{q}{\varepsilon}\right)^{1/2},$$

required to ensure a specific error ε can be obtained from (3.7).

According to the analysis above, assume that, for a given problem, we know the maximum wave number ω needed to adequately resolve the solution. Then we can choose the number of points to ensure that the wave is well resolved. This can reduce the cost of computation while preserving the accuracy. If more detailed information is known about the spectral distribution of the Fourier coefficients, then this can be used to obtain sharper comparisons of efficiency by weighting the error function appropriately.

In 1974, Swartz and Wendroff [65] extended the ideas of [42] to the finite element method, using smooth splines as basis functions and presented an analysis of the

fully discretized approximation. Our approach is to combine and extend the ideas of [42] and [65] to the DG method, using the techniques introduced in [69] (see also [68, 70]) where the explicit formula of the DG solution is given by Fourier analysis. See, for example [55, 36, 37, 8, 9] for recent development and studies of the dispersion and dissipation errors of DG methods using Fourier analysis.

3.3 Implementation of DG Algorithm

In this section, we present the implementation of the DG method solving the model problem (3.8):

$$\begin{aligned} u_t + au_x = 0, \quad x \in [0, 2\pi], t > 0 \\ u(x, 0) = \sin \omega x, \ x \in [0, 2\pi] \end{aligned} \tag{3.8}$$

with periodic boundary conditions. The exact solution to (3.8) is

$$U(x, t) = \sin(\omega(x - at)). \tag{3.9}$$

As discussed in Section 2.1, to define the DG method for the model problem, we first consider a uniform partition of the computational domain $[0, 2\pi]$ in N cells of size $\Delta x = \frac{2\pi}{N}$. Denote the cell by $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ and the cell center by $x_j = \frac{1}{2}\left(x_{j+\frac{1}{2}} + x_{j-\frac{1}{2}}\right)$, $j = 1, \cdots, N$, where

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = 2\pi$$

and denote the approximation space as

$$V_h^k = \left\{ v : v|_{I_j} \in P^k(I_j);\ 1 \le j \le N \right\} \tag{3.10}$$

where $P^k(I_j)$ denotes the set of polynomials of degree up to k defined on the cell I_j. The semi-discrete DG method using the upwind flux for solving (3.8) is defined as follows: find the unique function $u = u(t) \in V_h^k$ such that, for $j = 1, \cdots, N$,

$$\int_{I_j} u_t v \, dx - a \int_{I_j} u v_x \, dx + a u^-_{j+\frac{1}{2}} v(x^-_{j+\frac{1}{2}}) - a u^-_{j-\frac{1}{2}} v(x^+_{j-\frac{1}{2}}) = 0 \tag{3.11}$$

holds for all test functions $v \in V_h^k$. Here and below u^+, u^- denote the left and right limits of the function u at the cell interface, respectively.

We now look at the implementation of the scheme (3.11). If a local basis of $P^k(I_j)$ is chosen and denoted as $\phi_j^l(x)$ for $l = 1, 2, \cdots, k+1$, then the numerical solution can be represented as

$$u(x) = \sum_{l=1}^{k+1} u_j^l \phi_j^l(x), \quad x \in I_j. \tag{3.12}$$

After substituting (3.12) into (3.11), we have

$$\sum_{l=1}^{k+1} \underbrace{\left(\int_{I_j} \phi_j^l(x) \phi_j^m(x) dx \right)}_{\boldsymbol{M}_{m,l}} (u_j^l)_t$$

$$= \sum_{l=1}^{k+1} a \left\{ \underbrace{\left(\int_{I_j} \phi_j^l(x) (\phi_j^m(x))_x \, dx \right) u_j^l + \left(\phi_j^l \phi_j^m \big|_{x^-_{j+\frac{1}{2}}} \right) u_j^l}_{A_{1m,l} u_j^l} \underbrace{- \left(\phi_{j-1}^l \phi_j^m \big|_{x^+_{j-\frac{1}{2}}} \right) u_{j-1}^l}_{B_{1m,l} u_{j-1}^l} \right\},$$

which can be written as the following matrix form

$$\boldsymbol{M}\frac{d\boldsymbol{u}_j}{dt} = aA_1\boldsymbol{u}_j + aB_1\boldsymbol{u}_{j-1} \quad \boldsymbol{u}_j = \left(u_j^1, \cdots, u_j^{k+1}\right)^T.$$

After inverting a local $(k+1)\times(k+1)$ mass matrix $\boldsymbol{M}$, the DG scheme (3.11) can be written as

$$\frac{d\boldsymbol{u}_j}{dt} = \frac{a}{\Delta x}\left(A\boldsymbol{u}_j + B\boldsymbol{u}_{j-1}\right), \tag{3.13}$$

where $\boldsymbol{u}_j = \left(u_j^1, \cdots, u_j^{k+1}\right)^T$, A and B are $(k+1)\times(k+1)$ constant matrices.

As in [69], the local basis of $P^k(I_j)$ is chosen to be the Lagrangian polynomials, based on the following $k+1$ equally spaced points

$$x_{j+\frac{2l-k}{2(k+1)}} = x_j + \left(\frac{2l-k}{2(k+1)}\right)\Delta x, \;\; l = 0, \cdots, k.$$

In this way, $\boldsymbol{u}_j$, the coefficients of the solution u inside the cell I_j is a vector of length $k+1$ containing the values of the solution at these points. The DG finite element scheme (3.13) becomes a finite difference scheme on a globally uniform mesh (with a mesh size $\frac{\Delta x}{k+1}$). However, it is not a standard finite difference scheme because each point in the group of $k+1$ points belonging to the cell I_j obeys a different form.

3.4 Error Analysis

In this section, we present the details of the DG algorithm formulation with the basis functions discussed in Section 3.3 and derive the error estimation of the semi-discrete scheme using piecewise P^k polynomials, with $k = 1$, 2, 3. At the end of this section,

We provide an estimate of the necessary number of points per wavelength required to obtain a fixed error.

3.4.1 The Case of P^1

In this subsection, we consider the piecewise linear case, i.e. $k = 1$. We present the details of the error analysis for this case.

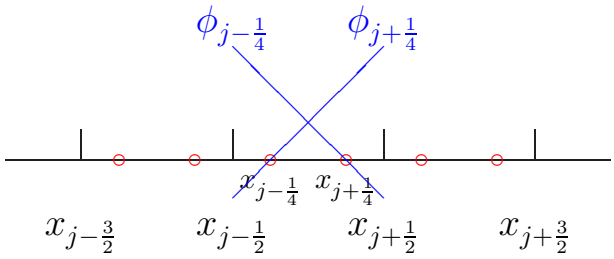


Figure 3.1: Local basis functions of P^1 case.

In this case, the local basis functions inside cell I_j are $\phi_{j-\frac{1}{4}}(x)$, $\phi_{j+\frac{1}{4}}(x)$, which are Lagrangian polynomials based on the points $x_{j-\frac{1}{4}}$, $x_{j+\frac{1}{4}}$ (Figure 3.1). With these basis functions, the solution inside the cell I_j is then represented by

$$u(x) = u_{j-\frac{1}{4}}\phi_{j-\frac{1}{4}}(x) + u_{j+\frac{1}{4}}\phi_{j+\frac{1}{4}}(x). \tag{3.14}$$

After substituting (3.14) into (3.11), we get the finite difference representation of the DG method

$$\begin{aligned} u'_{j-\frac{1}{4}} &= \frac{a}{4\Delta x}\left(-5u_{j-\frac{5}{4}} + 15u_{j-\frac{3}{4}} - \ 7u_{j-\frac{1}{4}} + 3u_{j+\frac{1}{4}}\right) \\ u'_{j+\frac{1}{4}} &= \frac{a}{4\Delta x}\left(\ \ u_{j-\frac{5}{4}} - \ 3u_{j-\frac{3}{4}} + 11u_{j-\frac{1}{4}} - 9u_{j+\frac{1}{4}}\right). \end{aligned} \tag{3.15}$$

The scheme (3.15) can be rewritten into the matrix form

$$\frac{d\boldsymbol{u}_j}{dt} = \frac{a}{\Delta x}\left(A\boldsymbol{u}_j + B\boldsymbol{u}_{j-1}\right), \tag{3.16}$$

where

$$\boldsymbol{u}_j = \begin{pmatrix} u_{j-\frac{1}{4}} \\ \\ u_{j+\frac{1}{4}} \end{pmatrix}, \quad A = \frac{1}{4}\begin{pmatrix} -7 & -3 \\ \\ 11 & -9 \end{pmatrix}, \quad B = \frac{1}{4}\begin{pmatrix} -5 & 15 \\ \\ 1 & -3 \end{pmatrix}. \tag{3.17}$$

To solve (3.16), the standard Fourier analysis is used here. This analysis depends heavily on the assumption of uniform mesh and periodic boundary conditions. Assume

$$\begin{pmatrix} u_{j-\frac{1}{4}}(t) \\ \\ u_{j+\frac{1}{4}}(t) \end{pmatrix} = \begin{pmatrix} \hat{u}_{-\frac{1}{4}}(t) \\ \\ \hat{u}_{\frac{1}{4}}(t) \end{pmatrix} e^{i\omega x_j}, \tag{3.18}$$

and after substituting (3.18) into the DG scheme (3.16), the coefficient vector satisfies the following ODE system

$$\begin{pmatrix} \hat{u}'_{-\frac{1}{4}}(t) \\ \\ \hat{u}'_{\frac{1}{4}}(t) \end{pmatrix} = G\begin{pmatrix} \hat{u}_{-\frac{1}{4}}(t) \\ \\ \hat{u}_{\frac{1}{4}}(t) \end{pmatrix}, \tag{3.19}$$

where G is the amplification matrix, given by

$$G = \frac{a}{\Delta x}(A + Be^{-i\xi}), \quad \xi = \omega\Delta x. \tag{3.20}$$

The two eigenvalues of G are

$$\lambda_{1,2} = \frac{a}{\Delta x}\left(-e^{-i\xi} - 2 \mp \sqrt{e^{-2i\xi} + 10e^{-i\xi} - 2}\right), \tag{3.21}$$

and the corresponding eigenvectors are

$$V_{1,2} = \begin{pmatrix} -1 + e^{i\xi} \mp 4\sqrt{1 + 10e^{i\xi} - 2e^{2i\xi}} \\ \\ 1 + 11e^{i\xi} \end{pmatrix}. \tag{3.22}$$

Then the general solution of the ODE system (3.19) is

$$\begin{pmatrix} \hat{u}_{-\frac{1}{4}}(t) \\ \\ \hat{u}_{\frac{1}{4}}(t) \end{pmatrix} = C_{11}\, e^{\lambda_1 t} V_1 + C_{12}\, e^{\lambda_2 t} V_2. \tag{3.23}$$

The coefficients C_{11} and C_{12} in (3.23) can be determined by the initial condition

$$\begin{pmatrix} \hat{u}_{-\frac{1}{4}}(0) \\ \\ \hat{u}_{\frac{1}{4}}(0) \end{pmatrix} = \begin{pmatrix} e^{-\frac{i\,\xi}{4}} \\ \\ e^{\frac{i\,\xi}{4}} \end{pmatrix}. \tag{3.24}$$

We thus have the explicit solution of the DG scheme with piecewise linear polynomials for solving (3.8). Comparing this DG approximate solution with the exact solution (3.9) will give us the quantitative error. The error analysis procedure can be summarized in Figure 3.2.

Now let us first introduce some notations:

$$\begin{aligned} \|e_{-\frac{1}{4}}\|_\infty &:= \max_{1\le j\le N} \left| U(x_{j-\frac{1}{4}}, t) - u_{j-\frac{1}{4}}(t) \right|, \\ \|e_{+\frac{1}{4}}\|_\infty &:= \max_{1\le j\le N} \left| U(x_{j+\frac{1}{4}}, t) - u_{j+\frac{1}{4}}(t) \right|. \end{aligned} \tag{3.25}$$

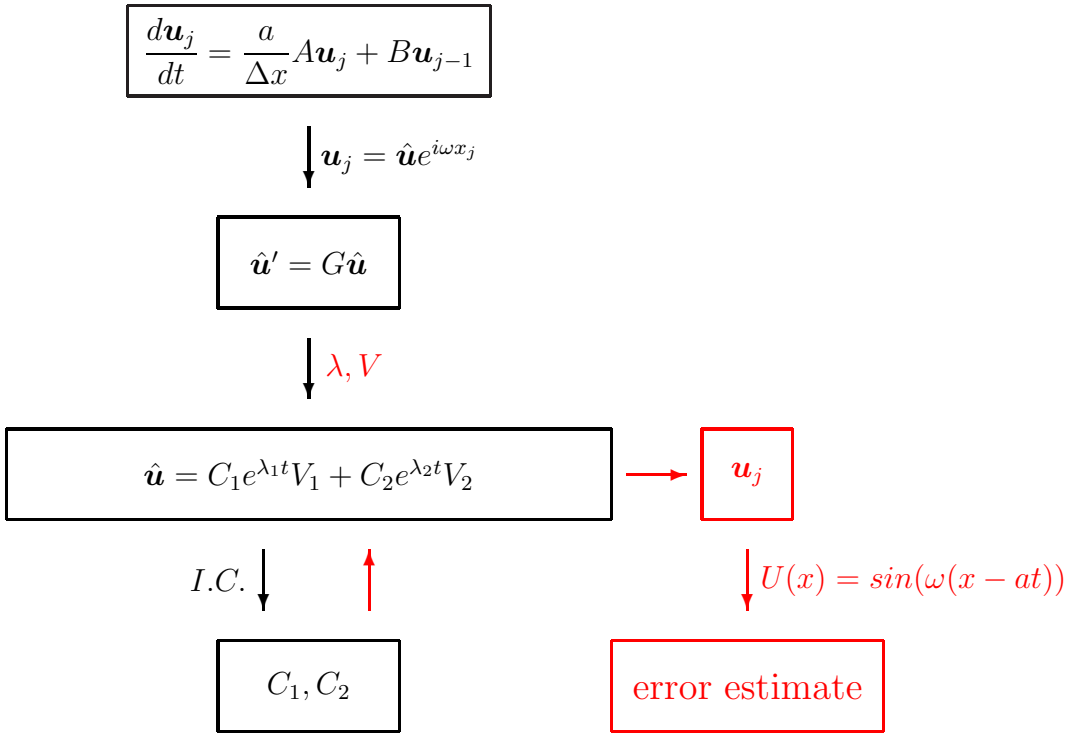


Figure 3.2: Fourier analysis procedure for P^1 case.

From the error analysis procedure, we have

$$
\begin{aligned}
\|e_{+\frac{1}{4}}\|_\infty &= \max_{1\le j\le N}\left|U(x_{j+\frac{1}{4}},t)-u_{j+\frac{1}{4}}(t)\right| \\
&= \max_{1\le j\le N}\left|\sin\underbrace{\left(i\omega(x_{j+\frac{1}{4}}-at)\right)}_{\Theta}-\mathrm{Im}\left\{e^{i\omega x_j}\hat{u}_{+\frac{1}{4}}(t)\right\}\right| \\
&= \max_{1\le j\le N}\left|\sin\Theta-\mathrm{Im}\left\{e^{i\Theta}\underbrace{\exp\left\{i\omega\left(at-\frac{\Delta x}{4}\right)\right\}\hat{u}_{+\frac{1}{4}}(t)}_{\text{Taylor expansion}}\right\}\right| \\
&= \max_{1\le j\le N}\left|\sin\Theta-\mathrm{Im}\left\{e^{i\Theta}\left(1-\frac{1}{24}\xi^2-\frac{9i+8\omega t}{576}\xi^3+\cdots\right)\right\}\right| \\
&= \max_{1\le j\le N}\left|\sin\Theta-\mathrm{Im}\left\{e^{i\Theta}\underbrace{\left(1-\frac{1}{24}\xi^2-\frac{9i+8\omega t}{576}\xi^3+\cdots\right)}_{1+C_3+iC_4}\right\}\right| \\
&= \max_{1\le j\le N}\left|\sin\Theta-\left((1+C_3)\sin\Theta+C_4\cos\Theta\right)\right| \\
&= \max_{1\le j\le N}\left|-C_3\sin\Theta+C_4\cos\Theta\right| \\
&= \sqrt{C_3^2+C_4^2}.
\end{aligned}
$$

Notice that (Taylor expansion part):

$$
\left|\exp\left\{i\omega\left(at-\frac{\Delta x}{4}\right)\right\}\hat{u}_{+\frac{1}{4}}(t)-1\right| = |1+C_3+iC_4-1| = \sqrt{C_3^2+C_4^2}.
$$

Therefore

$$
\|e_{+\frac{1}{4}}\|_\infty = \left|\exp\left\{i\omega\left(at-\frac{\Delta x}{4}\right)\right\}\hat{u}_{+\frac{1}{4}}(t)-1\right|
$$

By a simple Taylor expansion, we get

$$\begin{aligned}\|e_{+\frac{1}{4}}\|_\infty =&\frac{1}{24}\xi^2+\frac{\overline{t}}{72}\xi^3-\frac{211}{27648}\xi^4-\frac{1861\,\overline{t}}{414720}\xi^5\\&+\left(\frac{2808495+1364224\,\overline{t}^2}{1592524800}\right)\xi^6\\&-\left(\frac{11(-2854295\,\overline{t}+965888\,\overline{t}^3)}{33443020800}\right)\xi^7\\&+\left(\frac{-6386427605+581426944\,\overline{t}^2+1359970304\,\overline{t}^4}{12842119987200}\right)\xi^8\\&-\left(\frac{48576117895\,\overline{t}-7643182848\,\overline{t}^3+6799851520\,\overline{t}^5}{192631799808000}\right)\xi^9\\&+O(\xi^{10})\end{aligned}\tag{3.26}$$

where $\overline{t}=\omega at$.

From (3.26), for short time $\overline{t}$, the first term $\frac{1}{24}\xi^2$, which is independent of $\overline{t}$, dominates. As $\overline{t}$ increases to $O\left(\frac{1}{\xi}\right)$, the second term $\frac{\overline{t}}{72}\xi^3$ begins to dominate. If $\overline{t}$ continues increasing, another term with higher degree of $\overline{t}$ as coefficient may dominate. This means, during different time intervals, the dominant terms are different. At $\overline{t}=O\left(\frac{1}{\xi}\right)$, the error can be approximated by the first two terms, i.e.

$$\begin{aligned}\|e_{-\frac{1}{4}}\|_\infty &\simeq \left|\frac{1}{24}\xi^2-\frac{1}{72}\,\overline{t}\xi^3\right|,\\ \|e_{+\frac{1}{4}}\|_\infty &\simeq \frac{1}{24}\xi^2+\frac{1}{72}\,\overline{t}\xi^3.\end{aligned}\tag{3.27}$$

Denote $e_1=\max\left\{\|e_{-\frac{1}{4}}\|_\infty,\,\|e_{+\frac{1}{4}}\|_\infty\right\}$, where the subscript 1 is used here to indicate

the case $k = 1$. Clearly, the error e_1 satisfies

$$e_1 = \max\left\{\|e_{-\frac{1}{4}}\|_\infty, \|e_{+\frac{1}{4}}\|_\infty\right\} \simeq \frac{1}{24}\xi^2 + \frac{1}{72}\,\bar{t}\xi^3. \tag{3.28}$$

Similar to the finite difference case, we would like to rewrite the error in terms of the number of points per wavelength M_1 and the number of time periods $q = \frac{\bar{t}}{2\pi}$. Notice that two points $\{x_{j-\frac{1}{4}},\, x_{j+\frac{1}{4}}\}$ are used for each cell, therefore,

$$M_1 = \frac{2N}{\omega} = \frac{4\pi}{\xi}. \tag{3.29}$$

Substituting (3.29) and (3.5) into (3.28), we get

$$e_1 \simeq \frac{1}{24}\left(\frac{4\pi}{M_1}\right)^2 + \frac{\pi q}{36}\left(\frac{4\pi}{M_1}\right)^3. \tag{3.30}$$

Then the number of points per wavelength required to guarantee the error $e_1 \leq \varepsilon$ satisfies the following inequality

$$\frac{1}{24}\left(\frac{4\pi}{M_1}\right)^2 + \frac{\pi q}{36}\left(\frac{4\pi}{M_1}\right)^3 \leq \varepsilon. \tag{3.31}$$

3.4.2 The Case of P^2

In this subsection, we use the same procedure as we did in the piecewise linear case to present the details of the error estimation of the DG method using piecewise P^2 polynomials.

In this case, the local basis functions $\left\{\phi_{j-\frac{1}{3}}(x),\, \phi_j(x),\, \phi_{j+\frac{1}{3}}(x)\right\}$ inside cell I_j are the Lagrangian polynomials based on the points $x_{j-\frac{1}{3}}$, x_j, $x_{j+\frac{1}{3}}$ (Figure 3.3). The

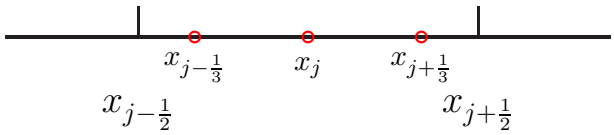


Figure 3.3: Mesh points related to the local basis functions of P^2 case.

solution inside the cell I_j is then represented by

$$u(x) = u_{j-\frac{1}{3}}\phi_{j-\frac{1}{3}}(x) + u_j\phi_j(x) + u_{j+\frac{1}{3}}\phi_{j+\frac{1}{3}}(x).$$

Denote

$$\boldsymbol{u}_j = \begin{pmatrix} u_{j-\frac{1}{3}} & u_j & u_{j+\frac{1}{3}} \end{pmatrix}^T,$$

then the DG scheme can be written into the matrix form (3.13) with

$$A = \begin{pmatrix} -\frac{43}{16} & -\frac{29}{24} & \frac{1}{16} \\ \frac{69}{16} & -\frac{15}{8} & -\frac{15}{16} \\ -\frac{19}{16} & \frac{139}{24} & -\frac{71}{16} \end{pmatrix}, \quad B = \begin{pmatrix} \frac{23}{16} & -\frac{115}{24} & \frac{115}{16} \\ -\frac{9}{16} & \frac{15}{8} & -\frac{45}{16} \\ -\frac{1}{16} & \frac{5}{24} & -\frac{5}{16} \end{pmatrix}. \tag{3.32}$$

The standard Fourier analysis is again used here. First we make an ansatz of the form

$$\begin{pmatrix} u_{j-\frac{1}{3}}(t) \\ u_j(t) \\ u_{j+\frac{1}{3}}(t) \end{pmatrix} = \begin{pmatrix} \hat{u}_{-\frac{1}{3}}(t) \\ \hat{u}_0(t) \\ \hat{u}_{\frac{1}{3}}(t) \end{pmatrix} e^{i\omega x_j}. \tag{3.33}$$

After substituting (3.33) into the DG scheme (3.13) with (3.32), the coefficient vector

satisfies

$$\begin{pmatrix} \hat{u}'_{-\frac{1}{3}}(t) \\ \hat{u}'_0(t) \\ \hat{u}'_{\frac{1}{3}}(t) \end{pmatrix} = G \begin{pmatrix} \hat{u}_{-\frac{1}{3}}(t) \\ \hat{u}_0(t) \\ \hat{u}_{\frac{1}{3}}(t) \end{pmatrix}, \tag{3.34}$$

where G is given by (3.20) with A and B defined in (3.32). Then with three eigenvalues of G and the corresponding eigenvectors computed by Mathematica (Here we use Cardano's method to obtain the eigenvalues, see Appendix A.1), the general solution of the ODE system (3.34) is

$$\begin{pmatrix} \hat{u}_{-\frac{1}{3}}(t) \\ \hat{u}_0(t) \\ \hat{u}_{\frac{1}{3}}(t) \end{pmatrix} = C_{21}\, e^{\lambda_1 t} V_1 + C_{22}\, e^{\lambda_2 t} V_2 + C_{23}\, e^{\lambda_3 t} V_3. \tag{3.35}$$

The coefficients C_{21}, C_{22} and C_{23} in (3.35) can be determined by the initial condition

$$\begin{pmatrix} \hat{u}_{-\frac{1}{3}}(0) \\ \hat{u}_0(t) \\ \hat{u}_{\frac{1}{3}}(0) \end{pmatrix} = \begin{pmatrix} e^{-\frac{i\xi}{3}} \\ 1 \\ e^{\frac{i\xi}{3}} \end{pmatrix}. \tag{3.36}$$

We thus have the explicit solution of the DG scheme with P^2 polynomials for solving (3.8). Comparing this with the exact solution (3.9) will give us the quantitative error

estimates.

$$\begin{aligned}\|e_{-\frac{1}{3}}\|_\infty :=& \max_{1\le j\le N}\left|U(x_{j-\frac{1}{3}},t)-u_{j-\frac{1}{3}}(t)\right|\\ =&\frac{\xi^3}{1296}+\frac{3223\xi^5}{3110400}-\frac{241\,\overline{t}\xi^6}{1008000}+\left(-\frac{110175241}{134369280000}+\frac{\overline{t}^2}{80000}\right)\xi^7+O(\xi^8),\end{aligned}$$

$$\begin{aligned}\|e_0\|_\infty :=& \max_{1\le j\le N}|U(x_j,t)-u_j(t)|\\ =&\frac{\xi^3}{240}-\frac{4979\xi^5}{15552000}-\frac{11\,\overline{t}\xi^6}{1814400}+\left(\frac{28245223}{2015539200000}+\frac{\overline{t}^2}{432000}\right)\xi^7+O(\xi^8),\end{aligned}$$

$$\begin{aligned}\|e_{+\frac{1}{3}}\|_\infty :=& \max_{1\le j\le N}\left|U(x_{j+\frac{1}{3}},t)-u_{j+\frac{1}{3}}(t)\right|\\ =&\frac{23\xi^3}{6480}+\frac{67567\xi^5}{357696000}+\frac{263\,\overline{t}\xi^6}{4636800}-\left(\frac{451421052697}{8174355148800000}-\frac{\overline{t}^2}{368000}\right)\xi^7+O(\xi^8).\end{aligned}$$

Clearly, the coefficients of high order terms depend on $\overline{t}$. Similarly, during different time intervals, the dominant terms are different. Denote $e_2 := \max\left\{\|e_{-\frac{1}{3}}\|_\infty,\ \|e_0\|_\infty,\ \|e_{+\frac{1}{3}}\|_\infty\right\}$. At $\overline{t}=O\left(\frac{1}{\xi}\right)$, the error e_2 satisfies

$$e_2 \simeq \frac{\xi^3}{240}. \tag{3.37}$$

At $\overline{t}=O\left(\frac{1}{\xi^2}\right)$, the error e_2 satisfies

$$e_2 \simeq \frac{\xi^3}{240}-\frac{4979\xi^5}{15552000}-\frac{11\,\overline{t}\xi^6}{1814400}+\left(\frac{28245223}{2015539200000}+\frac{\overline{t}^2}{432000}\right)\xi^7. \tag{3.38}$$

In this case, the number of points per wavelength M_2 satisfies

$$M_2=\frac{3N}{\omega}=\frac{6\pi}{\xi}, \tag{3.39}$$

since three points are used in each cell. Similarly, by rewriting (3.37) and (3.38) in terms of M_2 and $q=\frac{\overline{t}}{2\pi}$, we can obtain the number of points per wavelength M_2

required to ensure that $e_2 \leq \varepsilon$.

3.4.3 The Case of P^3

In this subsection, we apply the same procedure discussed in the previous subsections to the DG method using piecewise P^3 polynomials. The local basis functions

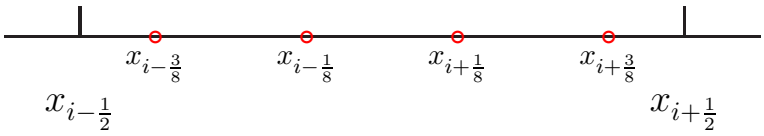


Figure 3.4: Mesh points related to the local basis functions of P^3 case.

$\left\{\phi_{j-\frac{3}{8}}(x),\ \phi_{j-\frac{1}{8}}(x),\ \phi_{j+\frac{1}{8}}(x),\ \phi_{j+\frac{3}{8}}(x)\right\}$ inside cell I_j are the Lagrangian polynomials based on the points $x_{j-\frac{3}{8}}$, $x_{j-\frac{1}{8}}$, $x_{j+\frac{1}{8}}$, $x_{j+\frac{3}{8}}$ (Figure 3.4). With these basis functions, the DG scheme can be written in the matrix form (3.13) with

$$A = \begin{pmatrix} -\frac{15109}{6144} & -\frac{4521}{2048} & \frac{255}{2048} & \frac{403}{6144} \\ \frac{43577}{6144} & -\frac{7699}{2048} & -\frac{1115}{2048} & -\frac{959}{6144} \\ -\frac{11761}{6144} & \frac{10747}{2048} & -\frac{5629}{2048} & -\frac{7097}{6144} \\ -\frac{4723}{6144} & -\frac{7983}{2048} & \frac{21993}{2048} & -\frac{43211}{6144} \end{pmatrix}$$

and

$$B = \begin{pmatrix} -\frac{2865}{2048} & \frac{12033}{2048} & -\frac{20055}{2048} & \frac{20055}{2048} \\ \frac{1685}{2048} & -\frac{7077}{2048} & \frac{11795}{2048} & -\frac{11795}{2048} \\ -\frac{365}{2048} & \frac{1533}{2048} & -\frac{2555}{2048} & \frac{2555}{2048} \\ -\frac{615}{2048} & \frac{2583}{2048} & -\frac{4305}{2048} & \frac{4305}{2048} \end{pmatrix}.$$

The standard Fourier analysis is once again used here to obtain the explicit solution of the DG scheme (Here we use Ferrari's method to solve the eigenvalues of the amplification matrix, see Appendix A.2). The error is also approximated by the dominant terms of Taylor series. Denote $e_3 := \max\left\{\|e_{-\frac{3}{8}}\|_\infty, \|e_{-\frac{1}{8}}\|_\infty, \|e_{+\frac{1}{8}}\|_\infty, \|e_{+\frac{3}{8}}\|_\infty\right\}$. At $\overline{t} = O\left(\frac{1}{\xi^2}\right)$,

$$e_3 \simeq 0.000363692026289682539682539682 54\xi^4. \tag{3.40}$$

At $\overline{t} = O\left(\frac{1}{\xi^3}\right)$,

$$\begin{aligned} e_3 \simeq & 0.000363692026289682539682539682 54\xi^4 \\ & - 7.253199544232434643589615700996 21 \times 10^{-6}\xi^6 \\ & + 7.086167800453514739229024944331066 \times 10^{-7}\overline{t}\xi^7. \end{aligned} \tag{3.41}$$

In this case, the number of points per wavelength M_3 satisfies

$$M_3 = \frac{4N}{\omega} = \frac{8\pi}{\xi}, \tag{3.42}$$

since four points are used in each cell. Similarly, the number of points per wavelength M_3 required to ensure that $e_3 \leq \varepsilon$ can be obtained by rewriting (3.40) and (3.41) in terms of M_3 and q.

Table 3.1 shows the lower bound of M_k to ensure the specific error ε, which is a function of q. we only present the leading term here. The leading term is computed using asymptotic analysis. Take (3.31) and $\varepsilon = 0.1$ as an example:

$$\frac{1}{24}\left(\frac{4\pi}{M_1}\right)^2 + \frac{\pi q}{36}\left(\frac{4\pi}{M_1}\right)^3 \leq 0.1. \tag{3.43}$$

We look for a solution of the form $M_1 = Cq^\alpha$. Using this expansion in the equation, we have

$$-0.1C^3q^{3\alpha} + \frac{2\pi^2}{3}Cq^\alpha + \frac{16\pi^4}{9}q = 0. \tag{3.44}$$

Then equating coefficients of q with highest degree to zero, we get

(1) $3\alpha = \alpha$, then $\alpha = 0$. However, $\frac{16\pi^4}{9} \neq 0$.

(2) $\alpha = 1$, then $-0.1C^3$ should be 0. $C = 0$.

(3) $3\alpha = 1$, then $\alpha = \frac{1}{3}$. And $-0.1C^3 = \frac{16\pi^4}{9} \Rightarrow C \approx 12$.

Therefore, the leading term of M_1 is $12q^{\frac{1}{3}}$.

Table 3.1: Leading term of the lower bound of M_k as a function of q

	M_1	M_2		M_3	
$\overline{t}$	$O\left(\frac{1}{\xi}\right)$	$O\left(\frac{1}{\xi}\right)$	$O\left(\frac{1}{\xi^2}\right)$	$O\left(\frac{1}{\xi^2}\right)$	$O\left(\frac{1}{\xi^3}\right)$
$\varepsilon = 0.1$	$12q^{\frac{1}{3}}$	6.53	$7q^{\frac{2}{7}}$	6.17	$6q^{\frac{1}{7}}$
$\varepsilon = 0.01$	$26q^{\frac{1}{3}}$	14.08	$10q^{\frac{2}{7}}$	10.98	$8q^{\frac{1}{7}}$
$\varepsilon = 0.001$	$56q^{\frac{1}{3}}$	30.33	$13q^{\frac{2}{7}}$	19.52	$12q^{\frac{1}{7}}$

3.5 Superconvergence

In this section, we study the superconvergence property of the DG solution at Radau points.

3.5.1 Superconvergence at Radau points

In this subsection, we study the superconvergence property of DG solutions at Radau points. The local basis functions of $P^k(I_j)$ are chosen to be the Lagrangian polynomials, based on the following $k+1$ points

$$x_{j+r_l} = x_j + \frac{\zeta_{k,l}}{2}\Delta x, \quad l = 0, \cdots, k$$

where $\{\zeta_{k,l}\}$ are the roots of the Radau polynomial $P_{k+1}(x) - P_k(x)$ and $P_k(x)$ is the Legendre polynomial of degree k. Figure 3.5 shows the local basis functions of P^1 based on Radau points.

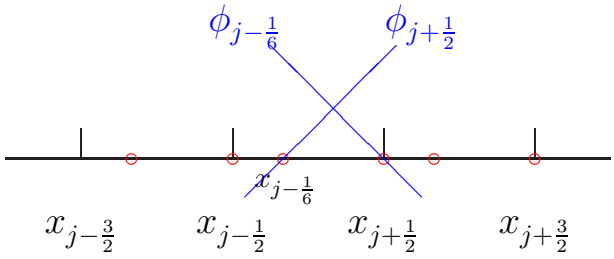


Figure 3.5: Local basis function of P^1 based on Radau points.

Table 3.2 displays $\{\zeta_{k,l}\}$ for $k = 1, 2, 3$. Note that $x_{j+r_k} = x_{j+\frac{1}{2}}$ is the downwind point of the element I_j.

Remark 3.5.1. The basis functions in this section are different from those discussed

Table 3.2: $\{\zeta_{k,l}\}_{l=0}^{k}$: the roots of $P_{k+1}(x) - P_k(x)$

	$\zeta_{k,l}$
$k=1$	$\zeta_{1,0} = -\frac{1}{3}, \qquad \zeta_{1,1} = 1$
$k=2$	$\zeta_{2,0} = -\frac{1+\sqrt{6}}{5}, \quad \zeta_{2,1} = -\frac{1-\sqrt{6}}{5}, \quad \zeta_{2,2} = 1$
$k=3$	$\zeta_{3,0} = -0.82282408097459210520890771246109$ $\zeta_{3,1} = -0.18106627111853057827014749586234$ $\zeta_{3,2} = \quad 0.57531892352169411205048377975200$ $\zeta_{3,3} = \quad 1$

in Section 3.4 and the amplification matrices are different, but they have the same eigenvalues. These eigenvalues are also the same as the ones derived in [55] using an orthonormal basis of Legendre polynomials. This can be proved as follows: Assume the amplification matrices with the basis functions $\left\{\phi_j^l\right\}_{l=1}^{k+1}$ and $\left\{\varphi_j^l\right\}_{l=1}^{k+1}$ are G_1 and G_2, respectively. Then

$$G_1 = T^{-1} G_2 T$$

where T is the change-of-basis matrix from $\left\{\phi_j^l\right\}_{l=1}^{k+1}$ to $\left\{\varphi_j^l\right\}_{l=1}^{k+1}$ and obviously T is invertible. G_1 and G_2 are similar matrices and thus they have the same characteristic polynomials.

As in Section 3.3, the DG method with the above basis can be written in a matrix form. The standard Fourier analysis is used to find the explicit solution of the DG scheme as discussed in Section 3.4. We do not repeat the details of this analysis procedure. Denote

$$\|e_{r_l}\|_\infty := \max_{1\le j\le N} \left|U(x_{j+r_l}, t) - u(x_{j+r_l}, t)\right|,$$

$$\|e^-_{+\frac{1}{2}}\|_\infty := \max_{1\le j\le N} \left|U(x^-_{j+\frac{1}{2}}, t) - u(x^-_{j+\frac{1}{2}}, t)\right|.$$

The followings are the error estimate results.

- $k = 1$

$$\|e_{r_0}\|_\infty = \frac{1}{72}\sqrt{\frac{16}{81} + \overline{t}^2}\,\xi^3 + O(\xi^4),$$
$$\|e^-_{+\frac{1}{2}}\|_\infty = \frac{1}{72}\sqrt{\frac{16}{9} + \overline{t}^2}\,\xi^3 + O(\xi^4);$$

- $k = 2$

$$\|e_{r_0}\|_\infty = \left|0.00112979589711327123927891362988\xi^4 - \frac{\overline{t}}{7200}\xi^5\right| + O(\xi^6),$$
$$\|e_{r_1}\|_\infty = 0.00082979589711327123927891362988\xi^4 + \frac{\overline{t}}{7200}\xi^5 + O(\xi^6),$$
$$\|e^-_{+\frac{1}{2}}\|_\infty = \frac{\sqrt{\overline{t}^2 + 144/25}}{7200}\xi^5 + O(\xi^6);$$

- $k = 3$

$$\begin{aligned}
\|e_{r_0}\|_\infty =& \big|4.58526548614820367915101051257673 \times 10^{-5}\xi^5 \\
&+ 1.48442413127661313280202433713330 \times 10^{-6}\xi^7 \\
&-2.95523546192529651029864723686194 \times 10^{-7}\overline{t}\xi^8\big| + O(\xi^9), \\
\|e_{r_1}\|_\infty =& 4.81347804829324175268316712031863 \times 10^{-5}\xi^5 \\
&- 1.74662334589086702133200039227532 \times 10^{-6}\xi^7 \\
&+ 1.12983256993051227300311527010632 \times 10^{-8}\overline{t}\xi^8 + O(\xi^9), \\
\|e_{r_2}\|_\infty =& 2.60817329279298238245105944584508 \times 10^{-5}\xi^5 \\
&+ 1.86838652276109018545310450926279 \times 10^{-6}\xi^7 \\
&+ 3.23693699988152665996155459585616 \times 10^{-7}\overline{t}\xi^8 + O(\xi^9), \\
\|e^-_{+\frac{1}{2}}\|_\infty =& \frac{\sqrt{\overline{t}^2 + 576/49}}{1412200}\xi^7 \\
&+ \frac{3.20791891379522690550503036041469 \times 10^{-7}\overline{t}}{\sqrt{576 + 49\overline{t}^2}}\xi^8 + O(\xi^9).
\end{aligned}$$

3.5.2 Superconvergence at the Downwind Point

In this subsection, we take a particular look at the error at the downwind point of each cell.

Let M_k^f be the number of points per wavelength needed to guarantee the error $\|e^-_{+\frac{1}{2}}\|_\infty \le \varepsilon$ when using P^k polynomials. According to the results derived in the previous subsection, even when $\overline{t} \gg 1$, we have

- $k = 1$

$$\|e^-_{+\frac{1}{2}}\|_\infty = \frac{1}{72}\sqrt{\frac{16}{9} + \overline{t}^2}\,\xi^3 + O(\xi^4) \simeq \frac{\overline{t}}{72}\xi = \frac{\pi q}{36}\left(\frac{4\pi}{M_1^f}\right)^3 \le \varepsilon.$$

- $k = 2$

$$\|e^-_{+\frac{1}{2}}\|_\infty = \frac{\sqrt{\overline{t}^2 + 144/25}}{7200}\xi^5 + O(\xi^6) \simeq \frac{\overline{t}}{7200}\xi^5 = \frac{\pi q}{3600}\left(\frac{6\pi}{M_2^f}\right)^5 \le \varepsilon.$$

- $k = 3$

$$\|e^-_{+\frac{1}{2}}\|_\infty = \frac{\sqrt{\overline{t}^2 + 576/49}}{1412200}\xi^7 + O(\xi^8) \simeq \frac{\overline{t}}{1411200}\xi^7 = \frac{\pi q}{705600}\left(\frac{8\pi}{M_3^f}\right)^7 \le \varepsilon.$$

Therefore,

$$M_1^f \ge 4\pi\left(\frac{\pi}{36}\right)^{\frac{1}{3}}\left(\frac{q}{\varepsilon}\right)^{\frac{1}{3}},$$

$$M_2^f \ge 6\pi\left(\frac{\pi}{3600}\right)^{\frac{1}{5}}\left(\frac{q}{\varepsilon}\right)^{\frac{1}{5}},$$

$$M_3^f \ge 8\pi\left(\frac{\pi}{705600}\right)^{\frac{1}{7}}\left(\frac{q}{\varepsilon}\right)^{\frac{1}{7}}.$$

Table 3.3 shows the lower bound of M_k^f as a function of q for three representative values of ε.

Table 3.3: Lower bound of M_p^f as a function of q

	M_1^f	M_2^f	M_3^f
$\varepsilon = 0.1$	$12q^{\frac{1}{3}}$	$7q^{\frac{1}{5}}$	$6q^{\frac{1}{7}}$
$\varepsilon = 0.01$	$25q^{\frac{1}{3}}$	$11q^{\frac{1}{5}}$	$8q^{\frac{1}{7}}$
$\varepsilon = 0.001$	$55q^{\frac{1}{3}}$	$18q^{\frac{1}{5}}$	$11q^{\frac{1}{7}}$

3.6 Fully Discrete Schemes

The previous sections deal with the semi-discrete schemes. In practice, the time discretization also plays an important role. An analysis of fully discretized approximation will be presented in this section with the basis functions discussed in Section 3.4.

3.6.1 The Case of P^1

In this subsection, we present the details of the fully discretized error analysis for the piecewise linear case.

Let $\tilde{u}$ be the approximate solution of the differential-difference equations (3.11) obtained by using p-stage explicit Runge-Kutta methods of order p (RK(p, p)) [32]. Then

$$\begin{pmatrix} \tilde{u}_{j-\frac{1}{4}} \\ \\ \tilde{u}_{j+\frac{1}{4}} \end{pmatrix} = R^n \begin{pmatrix} e^{i\omega x_{j-\frac{1}{4}}} \\ \\ e^{i\omega x_{j+\frac{1}{4}}} \end{pmatrix}, \quad n = \frac{t}{\Delta t} \tag{3.45}$$

where

$$R = 1 + \Delta t G + \frac{\Delta t^2}{2!} G^2 + \cdots + \frac{\Delta t^p}{p!} G^p \tag{3.46}$$

with the amplification matrix G defined in (3.20). Let $Q = [V_1, V_2]$ be the matrix with G's eigenvectors as columns. Clearly, we have

$$Q^{-1} G Q = \Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}. \tag{3.47}$$

Denote $\widetilde{R} = Q^{-1} R Q$. From (3.47) and (3.46), we have

$$\begin{aligned} \widetilde{R} = Q^{-1} R Q &= 1 + \Delta t \Lambda + \cdots + \frac{\Delta t^p}{p!} \Lambda^p \\ &= \begin{pmatrix} 1 + \Delta t \lambda_1 + \cdots \dfrac{\Delta t^p}{p!} \lambda_1^p & 0 \\ 0 & 1 + \Delta t \lambda_2 + \cdots \dfrac{\Delta t^p}{p!} \lambda_2^p \end{pmatrix}. \end{aligned}$$

Thus,

$$\begin{aligned} R^n = (Q \widetilde{R} Q^{-1})^n &= Q \widetilde{R}^n Q^{-1} \\ &= Q \begin{pmatrix} 1 + \Delta t \lambda_1 + \cdots \dfrac{\Delta t^p}{p!} \lambda_1^p & 0 \\ 0 & 1 + \Delta t \lambda_2 + \cdots \dfrac{\Delta t^p}{p!} \lambda_2^p \end{pmatrix}^n Q^{-1}. \end{aligned} \tag{3.48}$$

Substituting (3.48) into (3.45), we can determine the explicit solution of the RKDG scheme for solving (3.8). Comparing this with the exact solution (3.9) will give us the quantitative error estimates, based on the assumption cfl $= a\frac{\Delta t}{\Delta x}$.

By a simple Taylor expansion, the error between the DG solution with an RK(2,2)

time discretization and the exact solution is

$$\begin{aligned}\tilde{u}_{j-\frac{1}{4}} - U(x_{j-\frac{1}{4}},t) =& \frac{\xi^2}{24}\left(1-4i\mathrm{cfl}^2\overline{t}\right)e^{-i\overline{t}} - \frac{\xi^2}{24}\left(1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2\right)^{\frac{at}{\mathrm{cfl}\,\Delta x}}\, e^{-\frac{3i(-1+6\,\mathrm{cfl})\overline{t}}{1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2}} \\ &+ \frac{\xi^3}{576}\left(-i-8\overline{t}-24\mathrm{cfl}^2+72\mathrm{cfl}^3\overline{t}\right)e^{-i\overline{t}} \\ &+ \frac{\xi^3}{576}i\left(1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2\right)^{\frac{at}{\mathrm{cfl}\,\Delta x}}\, e^{-\frac{3i(-1+6\,\mathrm{cfl})\overline{t}}{1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2}} \\ &+ \frac{\xi^3}{24}\overline{t}\left(1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2\right)^{\frac{at}{\mathrm{cfl}\,\Delta x}}\, e^{-\frac{3i(-1+6\,\mathrm{cfl})\overline{t}}{1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2}} + \cdots\end{aligned}$$

where $\xi = \omega\Delta x$ and $\overline{t} = \omega at$.

In order to let the terms with coefficients $\left(1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2\right)^{\frac{at}{\mathrm{cfl}\,\Delta x}}$ go to 0 as $\Delta x \to 0$, we assume $\left|1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2\right| < 1$, i.e. $\mathrm{cfl} < \frac{1}{3}$. Then

$$\tilde{u}_{j-\frac{1}{4}} - U(x_{j-\frac{1}{4}},t) = e^{-i\overline{t}}\left(\frac{\xi^2}{24} - \frac{i\xi^2}{6}\,\mathrm{cfl}^2\overline{t} - \frac{i\xi^3}{576} - \frac{\xi^3}{72}\overline{t} - \frac{\xi^3}{24}\,\mathrm{cfl}^2\overline{t} + \frac{\xi^3}{8}\mathrm{cfl}^3\overline{t} + O(\xi^4)\right).$$

Thus

$$\begin{aligned}\|e_{-\frac{1}{4}}\|_\infty &= \max_{1\le j\le N}\left|\tilde{u}_{j-\frac{1}{4}} - U(x_{j-\frac{1}{4}},t)\right| \\ &= \left|\frac{1}{24}\sqrt{1+16\mathrm{cfl}^4\overline{t}^2}\xi^2 - \frac{\left(2\overline{t}+7\mathrm{cfl}^2\overline{t}-18\mathrm{cfl}^3\overline{t}\right)\xi^3}{144\sqrt{1+16\mathrm{cfl}^4\overline{t}^2}}\right| + O(\xi^4).\end{aligned} \tag{3.49}$$

We keep the same notations here as in the semi-discrete case. Similarly,

$$\|e_{+\frac{1}{4}}\|_\infty = \frac{1}{24}\sqrt{1+16\mathrm{cfl}^4\overline{t}^2}\xi^2 + \frac{\left(2\overline{t}+7\mathrm{cfl}^2\overline{t}-18\mathrm{cfl}^3\overline{t}\right)\xi^3}{144\sqrt{1+16\mathrm{cfl}^4\overline{t}^2}} + O(\xi^4) \tag{3.50}$$

and

$$\begin{aligned} e_1 &= \max\{\|e_{-\frac{1}{4}}\|_\infty, \|e_{+\frac{1}{4}}\|_\infty\} \\ &= \frac{1}{24}\sqrt{1+16\mathrm{cfl}^4\overline{t}^2}\xi^2 + \frac{\left(2\overline{t}+7\mathrm{cfl}^2\overline{t}-18\mathrm{cfl}^3\overline{t}\right)\xi^3}{144\sqrt{1+16\mathrm{cfl}^4\overline{t}^2}} + O(\xi^4). \end{aligned} \tag{3.51}$$

Since it takes a minimum of two points per wavelength to uniquely specify a wave, the largest wave number that can be represented on the N cells (reminder: two points are used for each cell) is $\omega = \frac{2N}{2} = N$. Therefore, we consider simple waves with $0 < \omega \leq N$. If ω is much less than N in magnitude, then there are many grid points per wavelength, and the solution is well resolved. In this case, ξ is a small number, and we can neglect $O(\xi^3)$ and obtain

$$e_1 \simeq \frac{1}{24}\sqrt{1+16\mathrm{cfl}^4\overline{t}^2}\xi^2. \tag{3.52}$$

If ξ is large, corresponding to fewer grid points per wavelength in the solution, then the P^1 solution is not a good approximation of the exact solution and approximation polynomials of degree > 1 are required.

Similarly, when using an RK(3,3) time discretization, the error estimates under the assumption[1] cfl < 0.418,

$$\|e_{-\frac{1}{4}}\|_\infty = \left|\frac{\xi^2}{24} - \frac{\xi^3}{72}\overline{t}\left(1+3\mathrm{cfl}^3\right)\right| + O(\xi^4), \tag{3.53}$$

$$\|e_{+\frac{1}{4}}\|_\infty = \frac{\xi^2}{24} + \frac{\xi^3}{72}\overline{t}\left(1+3\mathrm{cfl}^3\right) + O(\xi^4). \tag{3.54}$$

[1] in order to let the terms with coefficients $\left(1-6\,\mathrm{cfl}+18\,\mathrm{cfl}^2-36\mathrm{cfl}^3\right)^{-2+\frac{at}{\mathrm{cfl}\,\Delta x}} \to 0$ as $\Delta x \to 0$.

At $\overline{t} = O\left(\frac{1}{\xi}\right)$,

$$e_1 \simeq \frac{\xi^2}{24} + \frac{\xi^3}{72}\overline{t}\left(1 + 3\text{cfl}^3\right). \tag{3.55}$$

The necessary number of points per wavelength required to ensure a specific error ε can be obtained by rewriting (3.52) and(3.55) in terms of M_1 and q and setting $e_1 \leq \varepsilon$.

3.6.2 The Case of P^2

For $k = 2$ and $k = 3$, the analysis is similar.

In this subsection, we present the fully discretized error estimates when piecewise P^2 polynomials are used.

When using an RK(3,3) time discretization, under the assumption cfl < 0.327, we have

$$\|e_{-\frac{1}{3}}\|_\infty = \frac{1}{1296}\sqrt{1 + \frac{7222932}{3125}\text{cfl}^6\overline{t}^2}\ \xi^3 + O(\xi^4). \tag{3.56}$$

$$\|e_0\|_\infty = \frac{1}{240}\sqrt{1 + 100\text{cfl}^6\overline{t}^2}\ \xi^3 + O(\xi^4), \tag{3.57}$$

$$\|e_{+\frac{1}{3}}\|_\infty = \frac{23}{6480}\sqrt{1 + \frac{72900}{529}\text{cfl}^6\overline{t}^2}\ \xi^3 + O(\xi^4). \tag{3.58}$$

Thus,

$$e_2 = \max\{\|e_{-\frac{1}{3}}\|_\infty, \|e_0\|_\infty, \|e_{+\frac{1}{3}}\|_\infty\} \simeq \frac{1}{240}\sqrt{1 + 100\text{cfl}^6\overline{t}^2}\ \xi^3. \tag{3.59}$$

When using an RK(4,4) time discretization, under the assumption cfl < 0.351, we have

$$\|e_{-\frac{1}{3}}\|_\infty = \frac{1}{1296}\xi^3 + \frac{\overline{t}}{120}\mathrm{cfl}^4\xi^4 + O(\xi^5), \tag{3.60}$$

$$\|e_0\|_\infty = \frac{1}{240}\xi^3 + \frac{\overline{t}}{120}\mathrm{cfl}^4\xi^4 + O(\xi^5), \tag{3.61}$$

$$\|e_{+\frac{1}{3}}\|_\infty = \left|\frac{23}{6480}\xi^3 - \frac{\overline{t}}{120}\mathrm{cfl}^4\xi^4\right| + O(\xi^5). \tag{3.62}$$

At $\overline{t} = O\left(\frac{1}{\xi}\right)$,

$$e_2 \simeq \frac{1}{240}\xi^3 + \frac{\overline{t}}{120}\mathrm{cfl}^4\xi^4. \tag{3.63}$$

While at $\overline{t} = O\left(\frac{1}{\xi^2}\right)$, we need more terms of the Taylor series to approximate e_2. In this case,

$$\begin{aligned} e_2 \simeq \Big| & 0.0041666666666667\xi^3 + 0.0083333333333333\mathrm{cfl}^4\xi^4\overline{t} \\ & - 0.0003201517489712\xi^5 \\ & + \xi^6\left(-6.0626102292768959\times 10^{-6}\overline{t} - 0.0000680298353909\mathrm{cfl}^4\overline{t}\right. \\ & \left. +0.0008873456790123\mathrm{cfl}^5\overline{t} - 0.0029761904761905\mathrm{cfl}^6\overline{t}\right) \\ & + \xi^7\left(-0.00013365952729388 + 2.3148148148148148\times 10^{-6}\overline{t}^2\right. \\ & -0.0000354938271605\mathrm{cfl}^4\overline{t}^2 + 0.0002314814814815\mathrm{cfl}^5\overline{t}^2 \\ & +0.0001360596707819\mathrm{cfl}^8\overline{t}^2 - 0.0017746913580247\mathrm{cfl}^9\overline{t}^2 \\ & \left. +0.005787037037037\mathrm{cfl}^{10}\overline{t}^2\right)\Big|. \end{aligned} \tag{3.64}$$

The necessary number of points per wavelength required to ensure a specific error ε can be obtained by rewriting (3.59), (3.63) and (3.64) in terms of M_2 and q and setting $e_2 \leq \varepsilon$.

3.6.3 The Case of P^3

For the cases $k = 1$ and $k = 2$, the cfl condition is consistent with the CFL_{L^2} table in [26]. However, for $k = 3$, we cannot obtain the cfl condition using the same procedure by Taylor expansion and Mathematica due to limited memory. In this subsection, we provide the error estimates based on a fixed cfl $= 0.1$.

When using an RK(4,4) time discretization, we have

$$\|e_{-\frac{3}{8}}\|_\infty = \xi^4\sqrt{3.2541560538021135 \times 10^{-8} + 6.944444444444444 \times 10^{-13}\overline{t}^2} + O(\xi^5),$$

$$\|e_{-\frac{1}{8}}\|_\infty = \xi^4\sqrt{1.3128715187393769 \times 10^{-9} + 6.944444444444444 \times 10^{-13}\overline{t}^2} + O(\xi^5),$$

$$\|e_{+\frac{1}{8}}\|_\infty = \xi^4\sqrt{1.3227188998669514 \times 10^{-7} + 6.944444444444444 \times 10^{-13}\overline{t}^2} + O(\xi^5),$$

$$\|e_{+\frac{3}{8}}\|_\infty = \xi^4\sqrt{9.3484606662364834 \times 10^{-9} + 6.944444444444444 \times 10^{-13}\overline{t}^2} + O(\xi^5).$$

And

$$e_3 \simeq \xi^4\sqrt{1.3227188998669514 \times 10^{-7} + 6.9444444444444444 \times 10^{-13}\overline{t}^2}. \qquad (3.65)$$

When using an RK(5,5) time discretization, we have

$$\|e_{-\frac{3}{8}}\|_\infty = 0.0001803927951389\xi^4 + 1.3888888888888889 \times 10^{-8}\overline{t}\xi^5 + O(\xi^6),$$

$$\|e_{-\frac{1}{8}}\|_\infty = 0.0000362335689484\xi^4 + 1.3888888888888889 \times 10^{-8}\overline{t}\xi^5 + O(\xi^6),$$

$$\|e_{+\frac{1}{8}}\|_\infty = 0.0003636920262897\xi^4 + 1.3888888888888889 \times 10^{-8}\overline{t}\xi^5 + O(\xi^6),$$

$$\|e_{+\frac{3}{8}}\|_\infty = 0.0000966874379960\xi^4 + 1.3888888888888889 \times 10^{-8}\overline{t}\xi^5 + O(\xi^6).$$

At $\overline{t} = O\left(\frac{1}{\xi^2}\right)$,

$$e_3 \simeq 0.0003636920262897\xi^4 + 1.3888888888888889 \times 10^{-8}\overline{t}\xi^5. \tag{3.66}$$

The necessary number of points per wavelength required to ensure a specific error ε can be obtained by rewriting (3.65) and (3.66) in terms of M_3 and q and setting $e_3 \leq \varepsilon$.

Table 3.4 shows the M_k as a function of q (we only present the leading term here), derived from the error estimates in each subsection. These estimates is computed at $\overline{t} = O\left(\frac{1}{\xi}\right)$. Here we fix cfl $= 0.3$ for $k = 1$, cfl $= 0.2$ for $k = 2$ and cfl $= 0.1$ for $k = 3$.

Table 3.4: Leading term of the lower bound of M_k as a function of q

	M_1		M_2		M_3	
RK(p, p)	RK(2,2)	RK(3,3)	RK(3,3)	RK(4,4)	RK(4,4)	RK(5,5)
$\varepsilon = 0.1$	$12q^{\frac{1}{2}}$	$12q^{\frac{1}{3}}$	$5q^{\frac{1}{3}}$	$3q^{\frac{1}{4}}$	$2q^{\frac{1}{4}}$	$2q^{\frac{1}{5}}$
$\varepsilon = 0.01$	$39q^{\frac{1}{2}}$	$26q^{\frac{1}{3}}$	$11q^{\frac{1}{3}}$	$8q^{\frac{1}{4}}$	$4q^{\frac{1}{4}}$	$2q^{\frac{1}{5}}$
$\varepsilon = 0.001$	$122q^{\frac{1}{2}}$	$57q^{\frac{1}{3}}$	$24q^{\frac{1}{3}}$	$10q^{\frac{1}{4}}$	$8q^{\frac{1}{4}}$	$4q^{\frac{1}{5}}$

3.6.4 A Measure of Work per Wavelength

So far, we have expressed e_k as a function of M_k and q. Now the work per wavelength to obtain the error e_k when integrating to time t using RK(p, p) is given by

$$W_k = 2(k+1) \times M_k \times \frac{t}{\Delta t} \times p,$$

which is a product of the number of operations per mesh point per time step $2(k+1)$, the number of points per wavelength $M_k = \frac{2\pi(k+1)}{\omega \Delta x}$, the total number of time steps

$\frac{t}{\Delta t}$ and the number of stages per time step p. Using $\omega at = 2\pi q$ and $\text{cfl} = a\frac{\Delta t}{\Delta x}$, we obtain

$$\begin{aligned} W_k &= 2(k+1)M_k\, p\, \frac{t}{\Delta t} \\ &= 2(k+1)M_k\, p\, \frac{2\pi q}{\omega a \Delta t} \\ &= 2(k+1)M_k\, p\, \frac{2\pi q}{\omega \Delta x\, \text{cfl}} \\ &= 2(k+1)M_k\, p\, \frac{q}{\text{cfl}} \frac{2\pi}{\omega \Delta x} \\ &= 2(k+1)M_k\, p\, \frac{q}{\text{cfl}} \frac{M_k}{k+1} \\ &= \frac{2}{\text{cfl}}\, p\, q M_k^2. \end{aligned}$$

3.7 Numerical Results

In this section, we provide numerical experiments to demonstrate the predicted results presented in the previous sections.

To verify the predicted results for the semi-discrete scheme, we adopt SSPRK(9,9) [32] to make the temporal error negligible compared to the spatial error. This time discretization method for solving $du/dt = Lu$, where L is a spatial discretization operator (for this SSP method to be ninth order, L needs to be linear) is defined as

follows:

$$u^{(i)} = u^{(i-1)} + \Delta t\ Lu^{(i-1)}, \quad i = 1, \cdots, 8$$

$$u^{(9)} = \sum_{k=0}^{7} \alpha_{9,k} u^{(k)} + \alpha_{9,8} \left(u^{(8)} + \Delta t\, L\, u^{(8)} \right),$$

where

$$\alpha_{9,0} = \frac{16687}{45360}, \quad \alpha_{9,0} = \frac{2119}{5760}, \quad \alpha_{9,2} = \frac{103}{560},$$
$$\alpha_{9,3} = \frac{53}{864}, \quad \alpha_{9,4} = \frac{11}{720}, \quad \alpha_{9,5} = \frac{1}{320},$$
$$\alpha_{9,6} = \frac{1}{2160}, \quad \alpha_{9,7} = \frac{1}{10080}, \quad \alpha_{9,8} = \frac{1}{362880}.$$

Uniform meshes are used in the calculation and the CFL condition $\text{cfl} = a\frac{\Delta t}{\Delta x}$ is chosen from the table in [26].

Tables 3.5-3.7 list the error between DG solution and exact solution for fixed $\overline{t}$ with different cell size $\overline{N}$, where $\overline{N} = \frac{N}{\omega}$. The numerical results are computed using P^k polynomials and RK(9,9) time discretization. The predicted results are computed using the formulas derived in Section 3.4. The predicted errors agree with the computed results very well. These tables also show that the orders of the errors are different at different final time $\overline{t}$. This is because the higher order terms of the error dominate for large $\overline{t}$.

Figures 3.6-3.8 show the time evolution of the L^∞ error and e_k error when using P^k polynomials with three different time discretization methods. One is the same order as the spatial error, another one is one order higher than the spatial error and the last one is the RK(9,9) which makes the temporal error negligible comparing with

Table 3.5: When using P^k polynomials on a uniform mesh of $\overline{N}$ cells, $\overline{t} = 1$

N	Numerical results		Predicted by analysis	
$k = 1$	e_1	order	e_1	order
20	4.4639E-03		4.5430E-03	
40	1.0763E-03	2.05	1.0819E-03	2.07
80	2.6333E-04	2.03	2.6375E-04	2.04
160	6.5072E-05	2.02	6.5096E-05	2.02
$k = 2$	e_2	order	e_2	order
20	1.2796E-04		1.2825E-04	
40	1.6139E-05	2.99	1.6119E-05	2.99
80	2.0179E-06	3.00	2.0177E-06	3.00
160	2.5231E-07	3.00	2.5230E-07	3.00
$k = 3$	e_3	order	e_3	order
20	3.6972E-06		3.5497E-06	
40	2.2928E-07	4.01	2.2153E-07	4.00
80	1.3858E-08	4.05	1.3840E-08	4.00
160	8.6484E-10	4.00	8.6494E-10	4.00

Table 3.6: When using P^k polynomials on a uniform mesh of $\overline{N}$ cells, $\overline{t} = 100$

N	Numerical results		Predicted by analysis	
$k = 1$	e_1	order	e_1	order
20	4.5526E-02		4.7177E-02	
40	6.3632E-03	2.84	6.4111E-03	2.88
80	9.2878E-04	2.78	9.2990E-04	2.79
160	1.4831E-04	2.65	1.4837E-04	2.65
$k = 2$	e_2	order	e_2	order
20	1.3370E-04		1.3463E-04	
40	1.6120E-05	3.05	1.6164E-05	3.06
80	2.0172E-06	3.00	2.0180E-06	3.00
160	2.5226E-07	3.00	2.5230E-07	3.00
$k = 3$	e_3	order	e_3	order
20	3.5443E-06		3.5570E-06	
40	2.2147E-07	4.00	2.2148E-07	4.01
80	1.3831E-08	4.00	1.3838E-08	4.00
160	8.6229E-10	4.00	8.6490E-10	4.00

Table 3.7: When using P^k polynomials on a uniform mesh of $\overline{N}$ cells, $\overline{t} = 1000$

N	Numerical results		Predicted by analysis	
$k=1$	e_1	order	e_1	order
20	3.5039E-01		4.3476E-01	
40	5.3261E-02	2.72	5.4858E-02	2.99
80	6.9551E-03	2.94	6.9858E-03	2.97
160	9.0479E-04	2.94	9.0535E-04	2.95
$k=2$	e_2	order	e_2	order
20	4.4844E-04		8.2153E-04	
40	2.0799E-05	4.43	2.1489E-05	5.26
80	2.0575E-06	3.34	2.0589E-06	3.38
160	2.5111E-07	3.03	2.5261E-07	3.03
$k=3$	e_3	order	e_3	order
20	3.7047E-06		3.7487E-06	
40	2.2252E-07	4.06	2.2298E-07	4.07
80	1.3902E-08	4.00	1.3850E-08	4.01
160	8.6496E-10	4.01	8.6499E-10	4.00

the spatial error. The numerical L^∞ error is computed by taking a uniform partition of each element with 20 points. These figures show that the time discretization method needs to be at least one order higher than the DG method, in order to get the same result as the semi-discrete case. The order of the RK method higher than $k+1$ makes little difference when the DG method is using P^k polynomials. Also, the figures demonstrate that the predicted errors in general agree with the computed results very well.

Tables 3.8-3.10 list the numerical errors and their orders at Radau points for different final time $\overline{t}$. These tables verify the results derived in Section 3.5: the errors at the downwind point of each element are superconvergent of order $2k+1$ and at other Radau points are superconvergent of order $k+2$ when using piecewise P^k polynomials. These tables also show the relation between the error and $\overline{t}$: for $k=1$, errors are proportional to $\overline{t}$ while for $k=2,3$, the leading terms of the errors do not depend on $\overline{t}$, thus the errors are almost the same for $\overline{t}$ under $O\left(\frac{1}{\xi}\right)$. We have

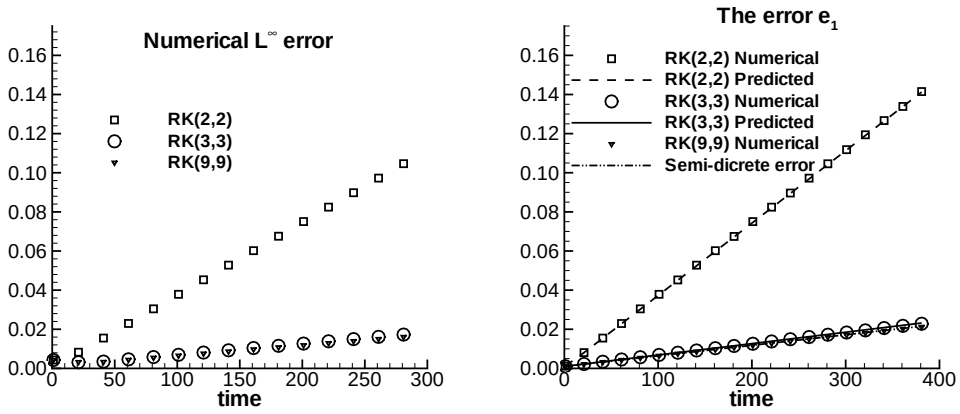


Figure 3.6: Time evolution of L^∞ error (left) and e_1 error (right) for P^1 DG solution with $\overline{N} = 40$, cfl $= 0.3$.

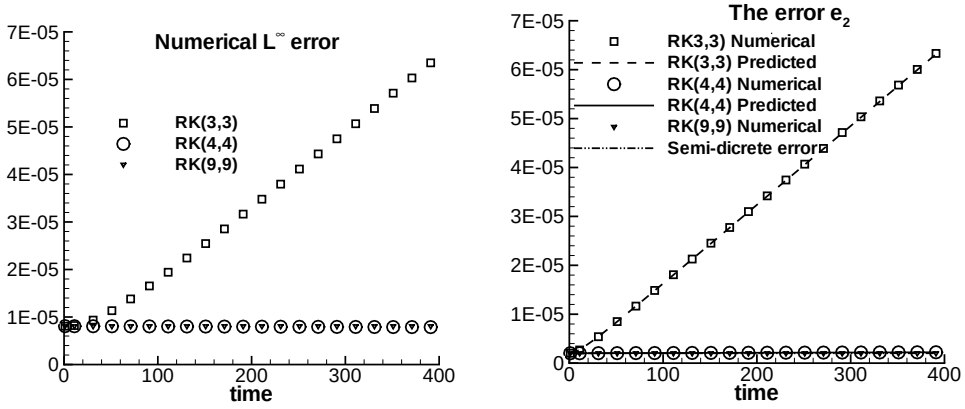


Figure 3.7: Time evolution of L^∞ error (left) and e_2 error (right) for P^2 DG solution with $\overline{N} = 80$, cfl $= 0.2$.

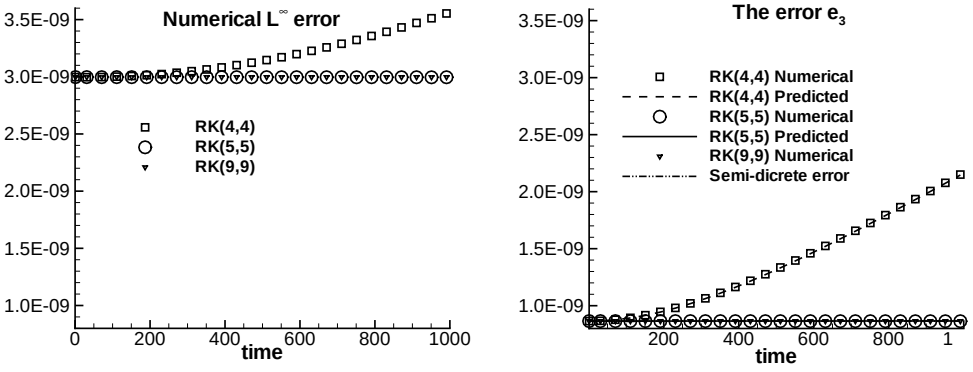


Figure 3.8: Time evolution of L^∞ error (left) and e_3 error (right) for P^3 DG solution with $\overline{N} = 160$, cfl $= 0.1$.

also tested the P^2 case at final time $\overline{t} = 500$, the error $\max_{0\le l\le 1}\{\|e_{r_l}\|_\infty\}$ achieves fifth order. This is because as $\overline{t}$ increases to 500, the fifth order term dominates. This verifies the fact that during different time intervals, the dominant terms are different.

We have also used a non-uniform mesh which is 60% random perturbation of the uniform mesh. For example, the right end of the cell I_j is now $x_{j+\frac{1}{2}} + 60\%(r_{j+\frac{1}{2}} - 0.5)\Delta x$, where $x_{j+\frac{1}{2}}$, Δx are taking values under the uniform mesh and $r_{j+\frac{1}{2}}$ is a random number from the uniform distribution over the range $(0, 1)$. Tables 3.11-3.13 list the errors and their orders at Radau points in this case for different final time $\overline{t}$, when using piecewise P^k polynomials. We can see that the error at Radau points is of order around $k + 2$, including the downwind point. We have also tested the cases of P^2 and P^3 using 1% random perturbation of the uniform perturbation. The numerical results still do not show the strong superconvergence of order $2k + 1$.

Table 3.8: Error at Radau points using P^1 DG and RK(9,9) on a uniform mesh of $\overline{N}$ cells

	$\overline{N}$	$\overline{t} = 1$		$\overline{t} = 10$		$\overline{t} = 100$	
		error	order	error	order	error	order
$\|e_{r_0}\|_\infty$	20	4.75E-04		4.25E-03		4.17E-02	
	40	5.91E-05	3.01	5.37E-04	2.98	5.36E-03	2.96
	80	7.38E-06	3.00	6.73E-05	3.00	6.72E-04	3.00
	160	9.22E-07	3.00	8.42E-06	3.00	8.41E-05	3.00
$\|e^-_{+\frac{1}{2}}\|_\infty$	20	6.97E-04		4.28E-03		4.18E-02	
	40	8.87E-05	2.97	5.39E-04	2.99	5.35E-03	2.97
	80	1.12E-05	2.99	6.77E-05	2.99	6.72E-04	2.99
	160	1.40E-06	3.00	8.48E-06	3.00	8.41E-05	3.00

Tables 3.14 and 3.15 list the necessary number of points per wavelength required to guarantee $e_k \le \varepsilon$ for various RKDG schemes and values of ε. Denote RKDG$(p, k+1)$ to be the scheme using the RK(p, p) method for time discretization and the DG method with piecewise P^k polynomials for space discretization. For these schemes with RK(9,9) method, the results of semi-discrete case are used as the predicted results. These two tables also verify our conclusions about the error estimates with

Table 3.9: Error at Radau points using P^2 DG and RK(9,9) on a uniform mesh of $\overline{N}$ cells

	$\overline{N}$	$\overline{t}=1$		$\overline{t}=10$		$\overline{t}=100$	
		error	order	error	order	error	order
$\max_{0\le l\le 1}\{\|e_{r_l}\|_\infty\}$	20	1.05E-05		1.23E-05		4.97E-05	
	40	6.73E-07	3.96	6.37E-07	4.27	1.83E-06	4.76
	80	4.25E-08	3.98	3.88E-08	4.07	7.30E-08	4.65
	160	2.67E-09	3.99	2.56E-09	3.92	3.27E-09	4.48
$\|e^-_{+\frac{1}{2}}\|_\infty$	20	1.09E-06		4.34E-06		4.23E-05	
	40	3.44E-08	4.98	1.36E-07	4.99	1.33E-06	5.00
	80	1.08E-09	5.00	4.26E-09	5.00	4.15E-08	5.00
	160	3.37E-11	5.00	1.33E-10	5.00	1.30E-09	5.00

Table 3.10: Error at Radau points using P^3 DG and RK(9,9) on a uniform mesh of $\overline{N}$ cells

	$\overline{N}$	$\overline{t}=1$		$\overline{t}=10$		$\overline{t}=100$	
		error	order	error	order	error	order
$\max_{0\le l\le 2}\|e_{r_l}\|_\infty$	20	1.64E-07		1.46E-07		1.48E-07	
	40	4.70E-09	5.12	4.60E-09	4.99	4.60E-09	5.01
	80	1.44E-10	5.02	1.44E-10	5.00	1.44E-10	5.00
	160	4.50E-12	5.00	4.50E-12	5.00	4.50E-12	5.00
$\|e^-_{+\frac{1}{2}}\|_\infty$	20	1.86E-08		2.24E-09		2.13E-08	
	40	2.12E-10	6.45	1.76E-11	6.99	1.67E-10	7.00
	80	5.09E-13	8.70	1.38E-13	7.00	1.31E-12	7.00
	160	3.75E-16	10.41	1.08E-15	7.00	1.02E-14	7.00
	320	2.85E-18	7.04	8.43E-18	7.00	7.98E-17	7.00
	640	2.23E-20	7.00	6.61E-20	6.99	6.23E-19	7.00

Table 3.11: Error at Radau points using P^1 DG and RK(9,9) on a random mesh of $\overline{N}$ cells

	$\overline{N}$	$\overline{t}=1$		$\overline{t}=10$		$\overline{t}=100$	
		error	order	error	order	error	order
$\|e_{r_0}\|_\infty$	20	1.82E-03		6.73E-03		6.33E-02	
	40	2.39E-04	2.93	7.68E-04	3.13	7.50E-03	3.08
	80	3.23E-05	2.89	9.05E-05	3.08	8.87E-04	3.08
	160	3.11E-06	3.37	1.22E-05	2.89	1.20E-04	2.89
	320	4.98E-07	2.64	1.49E-06	3.03	1.42E-05	3.08
$\|e^-_{+\frac{1}{2}}\|_\infty$	20	1.19E-03		6.81E-03		6.34E-02	
	40	1.50E-04	2.98	7.80E-04	3.13	7.51E-03	3.08
	80	2.27E-05	2.72	8.96E-05	3.12	8.87E-04	3.08
	160	2.36E-06	3.27	1.22E-05	2.88	1.20E-04	2.89
	320	3.51E-07	2.75	1.43E-06	3.09	1.42E-05	3.08

Table 3.12: Error at Radau points using P^2 DG and RK(9,9) on a random mesh of $\overline{N}$ cells

	$\overline{N}$	$\overline{t}=1$		$\overline{t}=10$		$\overline{t}=100$	
		error	order	error	order	error	order
$\|e_{r_0}\|_\infty$	20	3.20E-05		3.14E-05		1.31E-04	
	40	2.63E-06	3.61	2.58E-06	3.61	4.63E-06	4.82
	80	1.55E-07	4.08	1.99E-07	3.70	2.11E-07	4.45
	160	1.32E-08	3.56	1.37E-08	3.86	1.11E-08	4.25
	320	7.70E-10	4.10	7.85E-10	4.12	7.25E-10	3.94
$\|e^-_{+\frac{1}{2}}\|_\infty$	20	1.81E-05		1.39E-05		9.31E-05	
	40	4.87E-07	5.21	5.20E-07	4.74	2.81E-06	5.05
	80	6.06E-08	3.01	3.66E-08	3.83	9.28E-08	4.92
	160	2.34E-09	4.69	1.25E-09	4.87	2.74E-09	5.08
	320	1.33E-10	4.14	6.55E-11	4.26	1.01E-10	4.76

Table 3.13: Error at Radau points using P^3 DG and RK(9,9) on a random mesh of $\overline{N}$ cells

	$\overline{N}$	$\overline{t}=1$		$\overline{t}=10$		$\overline{t}=100$	
		error	order	error	order	error	order
$\|e_{r_0}\|_\infty$	20	3.90E-07		4.16E-07		4.71E-07	
	40	4.73E-08	3.04	3.55E-08	3.55	3.74E-08	3.65
	80	1.10E-09	5.43	8.65E-10	5.36	1.04E-09	5.17
	160	2.71E-11	5.34	3.18E-11	4.76	2.94E-11	5.14
	320	1.25E-12	4.44	1.43E-12	4.47	1.40E-12	4.39
$\|e^-_{+\frac{1}{2}}\|_\infty$	20	2.29E-07		2.71E-08		4.51E-08	
	40	1.80E-08	3.67	8.42E-09	1.69	3.91E-09	3.53
	80	2.97E-10	5.92	1.16E-10	6.19	4.84E-11	6.34
	160	8.42E-12	5.14	3.56E-12	5.02	1.25E-12	5.27
	320	2.49E-13	5.08	1.05E-13	5.08	4.71E-14	4.73

different time discretization methods, which states that the RK method needs to be at least one order higher than the DG method, in order to get the same result as the semi-discrete case; and there is little difference among the RK methods of order higher than $k+1$ when the DG method is using piecewise P^k polynomials.

Table 3.14: Necessary number of points per wavelength for $e_k \leq \varepsilon$ when time period $q = 1$

RKDG$(p, k+1)$	$\varepsilon = 0.1$		$\varepsilon = 0.01$		$\varepsilon = 0.001$	
	Numerical	Predicted	Numerical	Predicted	Numerical	Predicted
RKDG(2,2)	15.02	14.63	42.87	42.47	130.06	129.81
RKDG(3,2)	13.43	14.09	34.32	34.62	92.60	92.73
RKDG(9,2)	13.14	13.82	33.83	34.14	91.86	91.99
RKDG(3,3)	8.00	6.78	15.74	14.62	31.69	31.49
RKDG(4,3)	7.61	6.66	14.37	14.20	30.17	30.46
RKDG(9,3)	7.63	6.53	14.46	14.08	30.00	30.33
RKDG(4,4)	6.16	6.17	11.10	10.98	19.25	19.52
RKDG(5,4)	6.16	6.17	11.10	10.98	19.24	19.52
RKDG(9,4)	6.16	6.17	11.10	10.98	19.24	19.52

Table 3.15: Necessary number of points per wavelength for $e_k \leq \varepsilon$ when time period $q = 10$

RKDG$(p, k+1)$	$\varepsilon = 0.1$		$\varepsilon = 0.01$		$\varepsilon = 0.001$	
	Numerical	Predicted	Numerical	Predicted	Numerical	Predicted
RKDG(2,2)	39.46	38.60	122.60	122.06	386.36	385.97
RKDG(3,2)	26.74	27.38	60.85	61.03	140.86	140.93
RKDG(9,2)	26.08	26.72	59.48	59.67	138.16	138.23
RKDG(3,3)	13.47	11.27	26.07	24.27	53.43	52.30
RKDG(4,3)	11.38	13.58	18.83	19.86	32.85	33.11
RKDG(9,3)	11.51	13.53	18.94	19.64	32.47	32.32
RKDG(4,4)	8.00	6.19	12.65	11.00	20.45	19.57
RKDG(5,4)	8.00	6.19	12.64	10.99	20.35	19.53
RKDG(9,4)	8.00	8.70	12.64	12.65	20.36	20.35

Tables 3.16-3.17 list the work per wavelength of the scheme RKDG$(p, k+1)$, w_k, required to obtain the error ε when time period $q = 1$ and $q = 10$. We should point out that we could have increased the order of the RK method used in the scheme RKDG$(k+2, k+1)$ without significantly affecting the number of points per

wavelength, but the numbers of stages per timestep would increase considerably and so does the work per wavelength.

It is clear that even for short time period, high order methods are the most appropriate choice when accuracy is the primary consideration. When using P^k polynomials for the DG method, RK$(k+1, k+1)$ needs less work than the RK$(k+2, k+2)$ for the short time period. While for long time periods, the better choice is RK$(k+2, k+2)$.

Table 3.16: When time period $q = 1$

Type of Scheme	Order	Multiplies per Meshpoint	Stages per Timestep	W_k		
				$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$
RKDG(2,2)	2	4	2	2,900	24,000	224,700
RKDG(3,2)	2	4	3	4,000	24,000	172,000
RKDG(3,3)	3	6	3	1,400	6,400	29,800
RKDG(4,3)	3	6	4	1,800	8,100	37,100
RKDG(4,4)	4	8	4	3,000	9,600	30,500
RKDG(5,4)	4	8	5	3,800	12,000	38,100

Table 3.17: When time period $q = 10$

Type of Scheme	Order	Multiplies per Meshpoint	Stages per Timestep	W_k		
				$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$
RKDG(2,2)	2	4	2	201,400	1,995,300	19,891,700
RKDG(3,2)	2	4	3	149,900	745,000	3,972,500
RKDG(3,3)	3	6	3	38,100	176,800	820,400
RKDG(4,3)	3	6	4	51,800	157,800	438,600
RKDG(4,4)	4	8	4	30,600	96,900	306,300
RKDG(5,4)	4	8	5	38,300	120,800	381,500

In Figure 3.9, we compare the necessary number of points per wavelength required to obtain a fixed error 0.01 between two second order accurate methods. One is the DG method with P^1 polynomials and the other is the second order finite difference method. It can be seen clearly that, even for a short time period, the DG method needs much fewer points per wavelength. Also, the necessary number of points per

wavelength of the finite difference scheme grows faster with time than the RKDG method. This is because the leading term of the error for the RKDG method does not depend on $\overline{t}$, while the finite difference method does. This conclusion also holds for higher order schemes between the DG method and the finite difference method with the same order. In [65], Swartz and Wendroff computed the necessary number of intervals per wavelength required to obtain a fixed error for the finite element method using smooth splines as basis functions and various time discretization methods, such as the trapezoidal method and the leap-frog method. With the same order of time discretization method and space discretization method, the RKDG method requires fewer intervals per wavelength than the finite element method with smooth splines as basis functions to obtain a specified error. Moreover, the leading term of the error of this finite element method using smooth splines depends on $\overline{t}$. This means the necessary number of intervals per wavelength for the finite element method using smooth splines also grows faster with time than the RKDG method.

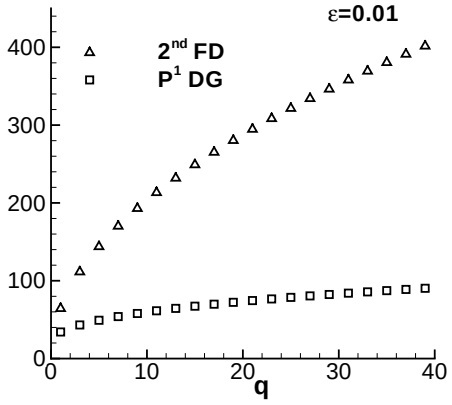


Figure 3.9: Necessary points per wave for $\|e\|_\infty \leq \varepsilon$

Remark 3.7.1. All the numerical results and the predicted results take point value collocation as initial conditions in this chapter. The usual way of taking initial conditions in a finite element method is via an L^2 projection. However this does not affect the results in our chapter. See Figure 3.10. The following are quatitative eror

for $k = 1$:

$$\|e_{-\frac{1}{4}}\|_\infty = \left|\frac{1}{32}\xi^2 - \frac{\bar{t}}{72}\xi^3\right| + O(\xi^4)$$
$$\|e_{+\frac{1}{4}}\|_\infty = \frac{5}{96}\xi^2 - \frac{\bar{t}}{72}\xi^3 + O(\xi^4)$$

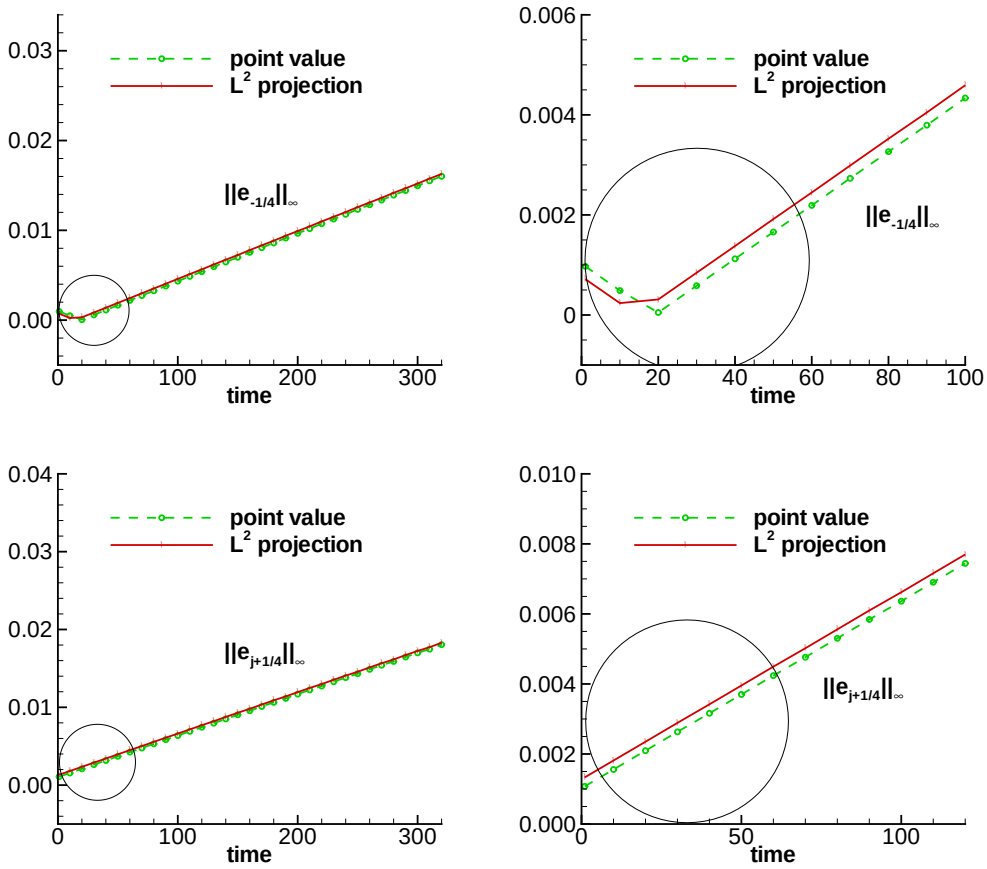


Figure 3.10: Comparison of time evolution of the errors with two different initial interpolations.

Remark 3.7.2. We also did this Fourier analysis procedure using central flux. Both our theoritical results and numerical results is only k-th order accurate when using DG methods with piecewise P^k polynomials.

3.8 Concluding Remarks

In this chapter, by Fourier analysis, we have derived the quantitative error estimates of the DG methods using piecewise P^k polynomials with $1 \leq k \leq 3$, for solving time dependent linear wave problems. We have proved the superconvergence property of the DG solution at Radau points. The error of the DG solution shows order $2k+1$ superconvergence at the downwind point of each element and order $k+2$ superconvergence at other Radau points. We also provide a fully discretized error analysis with various Runge-Kutta methods. We have computed the necessary number of points per wavelength required to obtain a fixed error for several RKDG schemes. A very important implication of the discussion in this chapter is that the dominant terms of the error for the RKDG schemes are different during different time intervals. This further justifies the advantage of choosing DG methods for long time simulation of linear wave equations.

The technique of Fourier analysis discussed in this chapter can be extended to the hyperbolic systems with constant coefficients. Theoretically, it can also be extended to the multidimensional case, however the algebraic manipulations may become prohibitively complicated.

Chapter 4

A Simple Weighted Essentially Non-Oscillatory Limiter for Runge-Kutta Discontinuous Galerkin Methods

4.1 Introduction

In this chapter, we consider the following hyperbolic conservation law

$$\begin{aligned} u_t + f(u)_x &= 0, \\ u(x,0) &= u_0(x), \end{aligned} \tag{4.1}$$

and its two-dimensional version, where u and $f(u)$ can be either scalars or vectors. We investigate a simple limiter using weighted essentially non-oscillatory (WENO) methodology for the Runge-Kutta discontinuous Galerkin (RKDG) methods, with the goal of obtaining a robust and high order limiting procedure to simultaneously maintain uniform high order accuracy in smooth regions and control spurious numerical oscillations near discontinuities. The idea of this limiter is to reconstruct the entire polynomial based on the polynomials of the DG solution in the target and neighboring cells, instead of reconstructing point values or moments based on cell averages or lower order moments.

The main difficulty in solving (4.1) is that solutions may contain discontinuities even if the initial conditions are smooth. DG methods can compute solutions to (4.1), which are either smooth or have weak shocks and other discontinuities, without further modification. However, for problems containing strong discontinuities, the scheme will generate significant oscillations and even nonlinear instabilities. We often need to apply nonlinear limiters to control these oscillations. Many such limiters exist in the literature, such as the minmod type total variation bounded (TVB) limiter [24, 23, 21, 20, 25], the moment-based limiter [13] and the more recent improved moment limiter [14]. Although these limiters can control spurious numerical oscillations near discontinuities, they tend to degrade accuracy when mistakenly used

in smooth regions of the solution. It is usually difficult to design limiters to achieve both high order accuracy and a non-oscillatory property near discontinuities. Qiu and Shu [52] and Zhu et al. [76] have made such an attempt using WENO methodology [34, 47, 38, 29, 35, 46, 48, 56] as limiters for the DG methods. They use the usual WENO reconstruction based on cell averages of neighboring cells as in [38, 35, 56], to reconstruct the values of the solutions at certain Gaussian quadrature points in the target cells, and then rebuild the solution polynomials from the original cell average and the reconstructed values at the Gaussian quadrature points through numerical integration of the moments. This limiter needs to use the information from not only the immediate neighboring cells but also neighbors' neighbors, making it complicated to implement in multi-dimensions, especially for unstructured meshes [76, 35, 73]. The effort in [49, 51] attempts to construct Hermite type WENO approximations, which use the information of not only the cell averages but also the lower order moments such as slopes, to reduce the spread of reconstruction stencils. However for higher order methods the information of neighbors' neighbors is still needed.

In this chapter, we use the WENO methodology to design a new and simpler limiter for the RKDG methods. We do not reconstruct the point values or moments individually and separately, but attempt to reconstruct the entire polynomial in one shot, using the information only from the target cell and its immediate neighbors. As we will see later, this approach simultaneously removes the problem of negative weights and reduces considerably the complexity of implementation, especially for multi-dimensional meshes including unstructured meshes. Comparing with previous WENO type limiters for DG schemes, the limiter developed in this chapter uses most fully the information of the complete polynomials which are already available for DG methods in the target and neighboring cells. Because of this richness of available information, the choice of linear weights are much less restrictive. Essentially, any

choice of positive linear weights which add up to one is adequate for accuracy. We refer to [29, 41, 28] and the review paper [61] for a detailed discussion of such practice in choosing linear weights for WENO procedures in finite volume WENO schemes. For our new WENO limiter, following the practice in [28], we give the central cell with a larger linear weight compared to the neighboring cells since in smooth regions the central stencil should provide the most stable reconstruction together with the highest quality in accuracy.

4.2 DG Algorithm Formulation

In this section, we give an overview of the algorithm formulation of the RKDG method for solving (4.1) in the one-dimensional case.

As discussed in Section 2.1, with a slight abuse of notation, the semi-discrete DG method for solving (4.1) is defined as follows: find the unique function $u = u(t) \in V_h^k$ such that, for $j = 1, \cdots, N$,

$$\int_{I_j} u_t v \, dx - \int_{I_j} f(u) v_x \, dx + \hat{f}_{j+\frac{1}{2}} v(x_{j+\frac{1}{2}}^-) - \hat{f}_{j-\frac{1}{2}} v(x_{j-\frac{1}{2}}^+) = 0 \tag{4.2}$$

holds for all test functions $v \in V_h^k$. $\hat{f}_{j+\frac{1}{2}}$ is the the so-called monotone numerical fluxes (approximate or exact Riemann solvers in the system case). To discretize the time variable, we use the TVD third order Runge-Kutta method (2.3). Other TVD, or SSP time discretizations [30] can of course also be used.

For simplicity, we consider the forward Euler time discretization of the semi-discrete scheme (4.2). Starting from a solution $u^n \in V_h^k$ at time level n (for the initial condition, u^0 is taken as the L^2 projection of the analytical initial condition

into V_h^k). We would like to "limit" it to obtain a new function $u^{n,new}$ before advancing it to next time level. That is: find $u^{n+1} \in V_h^k$, such that, for $j = 1, \cdots, N$,

$$\int_{I_j} \frac{u^{n+1} - u^{n,new}}{\Delta t} v \, dx - \int_{I_j} f(u^{n,new}) v_x \, dx + \hat{f}^{n,new}_{j+\frac{1}{2}} v(x^-_{j+\frac{1}{2}}) - \hat{f}^{n,new}_{j-\frac{1}{2}} v(x^+_{j-\frac{1}{2}}) = 0$$

holds for all test functions $v \in V_h^k$. The limiting procedure to go from u^n to $u^{n,new}$ will be discussed in the following section.

4.3 A New WENO Limiter

In this section, we present the details of our new WENO limiting procedure for the RKDG methods. As in [52], we also adopt the following framework:

1. Identify the troubled cells, namely, those cells which might need the limiting procedure.

2. Replace the solution polynomials in the troubled cells with reconstructed polynomials, which keep the original cell averages, maintain the original high order of accuracy, but are less oscillatory.

4.3.1 Identify the Troubled Cells

In this subsection, we discuss the identification of troubled cells. This part is not the emphasis of our chapter and we do not attempt to compare the pros and cons of various troubled cell identification procedures. We will simply use the TVB minmod limiter [24, 23, 21, 20, 25] to identify troubled cells. We emphasize that the goal of our

limiter is to be insensitive to the troubled-cell indicators. That is, if more troubled cells are identified than they actually exist, we will increase the computational cost of the algorithm but should otherwise maintain the original high order accuracy.

To detect the troubled cells using the limiter described above, we first denote the cell average of the solution u as

$$\bar{u}_j = \frac{1}{\Delta x_j} \int_{I_j} u \, dx \tag{4.3}$$

and further denote

$$\tilde{u}_j = u^-_{j+\frac{1}{2}} - \bar{u}_j, \quad \tilde{\tilde{u}}_j = \bar{u}_j - u^+_{j-\frac{1}{2}}. \tag{4.4}$$

$\tilde{u}_j$ and $\tilde{\tilde{u}}_j$ are modified either by the usual minmod limiter [33],

$$\tilde{u}^{(mod)}_j = m\left(\tilde{u}_j, \Delta_+\bar{u}_j, \Delta_-\bar{u}_j\right), \quad \tilde{\tilde{u}}^{(mod)}_j = m\left(\tilde{\tilde{u}}_j, \Delta_+\bar{u}_j, \Delta_-\bar{u}_j\right), \tag{4.5}$$

where

$$\Delta_+\bar{u}_j = \bar{u}_{j+1} - \bar{u}_j, \quad \Delta_-\bar{u}_j = \bar{u}_j - \bar{u}_{j-1},$$

and the minmod function m is defined in (2.9) or by the TVB modified minmod function (2.11).

Then, we declare that whenever one of the minmod functions (4.5) gets enacted (returns other than the first argument), this cell is marked as a troubled cell and subject to WENO reconstructions. Of course, if too few cells are identified as troubled cells, oscillations and possible instability may not be avoided. If too many cells are identified as troubled cells, the computational cost associated with the second step will increase. Therefore, troubled-cell indicator is a very important issue for

WENO limiters. However our main concern in this chapter is how to design the new WENO limiter. We refer the readers to [50] about the comparison among different troubled-cell indicators.

We have given the details of identifying troubled cells using the TVB minmod limiters for the one-dimensional scalar case. For two-dimensional scalar case, we use the TVB minmod limiter defined in [20]; for the one- and two-dimensional systems, we use the characteristic-wise TVB minmod limiters defined in [21] and [25], respectively.

4.3.2 Reconstruction of the New Polynomials in the Troubled Cells Using a WENO Limiter: Scalar Case

In this subsection, we present the details of the reconstruction procedure for the new polynomials in the troubled cells using our WENO limiter for scalar conservation laws. The idea of this WENO limiter is to reconstruct a new polynomial on the troubled cell I_j which is a convex combination of polynomials on this cell and its immediate neighboring cells, with necessary adjustments to keep the original cell average on the target cell. The nonlinear weights in the convex combination coefficients follows the classical WENO procedure.

We start with the one-dimensional scalar case. Assume that the cell I_j is a troubled cell. Denote the DG solution polynomial of u on the cells I_{j-1}, I_j, I_{j+1} as $p_0(x)$, $p_1(x)$, $p_2(x)$, respectively (Figure 4.1). In order to make sure that the reconstructed polynomial maintains the original cell average of p_1 in the target cell

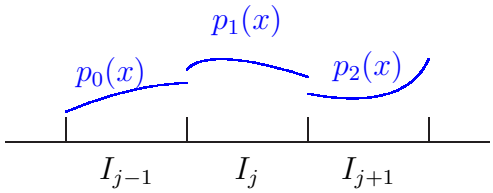


Figure 4.1: DG polynomials.

I_j, we make the following modifications:

$$\tilde{p}_0(x) = p_0(x) - \bar{\bar{p}}_0 + \bar{\bar{p}}_1, \qquad \tilde{p}_2(x) = p_2(x) - \bar{\bar{p}}_2 + \bar{\bar{p}}_1, \tag{4.6}$$

where

$$\bar{\bar{p}}_0 = \frac{1}{\Delta x_j}\int_{I_j} p_0(x)\,dx, \quad \bar{\bar{p}}_1 = \frac{1}{\Delta x_j}\int_{I_j} p_1(x)\,dx, \quad \bar{\bar{p}}_2 = \frac{1}{\Delta x_j}\int_{I_j} p_2(x)\,dx.$$

The final nonlinear WENO reconstruction polynomial $p_1^{new}(x)$ is now defined by a convex combination of these modified polynomials:

$$p_1^{new}(x) = \omega_0\tilde{p}_0(x) + \omega_1 p_1(x) + \omega_2\tilde{p}_2(x). \tag{4.7}$$

From (4.6) and (4.7), it is easy to prove that p_1^{new} has the same cell average as p_1 if the weights satisfy $\omega_0 + \omega_1 + \omega_2 = 1$. p_1^{new} also maintains the order of accuracy of p_1 can be directly proved by the following lemma.

Lemma 4.3.1. Assume u is sufficiently smooth in $[a, b]$. Denote a partition of $[a, b]$ in N cells as follows:

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = b.$$

Denote $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, $x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})$, $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$and $h = \max_{1\le j\le N}\Delta x_j$. We assume that the mesh is regular, namely, there is a constant

$c > 0$ independent of h such that

$$\Delta x_j \geq ch, 1 \leq j \leq N. \tag{4.8}$$

If

$$|u(x) - p(x)| \leq C_1 h^{k+1}, \quad \forall x \in I_j$$

where $p(x)$ is a polynomial of degree k, then

$$|u(x) - p(x)| \leq C_2 h^{k+1}, \quad \forall x \in I_{j+1}.$$

Here C_1 and C_2 depend on u and its derivatives but are independent of h.

Proof Denote $\{\xi_i\}_{i=0}^k$ as the set of the following $k+1$ equally spaced points in the cell I_j:

$$\xi_i = x_j + \left(\frac{2i-k}{2(k+2)}\right)\Delta x_j, \quad i = 0, \cdots, k. \tag{4.9}$$

Denote $u_I(x)$ to be the Lagrange interpolation polynomial of $u(x)$ based on $\{\xi_i\}$. Then

$$u_I(x) = \sum_{i=0}^{k} u(\xi_i) l_i(x),$$

where

$$l_i(x) = \prod_{m=0,\, m\neq i}^{k} \frac{x - \xi_i}{\xi_m - \xi_i}.$$

By Theorem B.2.1 in Appendix B.1, we have

$$u(x) - u_I(x) = \frac{u^{(k+1)}(\zeta)}{(k+1)!} \prod_{i=0}^{k} (x - \xi_i) \tag{4.10}$$

where

$$\min\{\xi_0, \xi_1, \cdots, \xi_n, x\} \leq \zeta \leq \max\{\xi_0, \xi_1, \cdots, \xi_n, x\}.$$

On the other hand, $p(x)$ can be written in the following Lagrange form

$$p(x) = \sum_{i=0}^{k} p(\xi_i) l_i(x)$$

Notice that, by the assumption, we have

$$|u(\xi_i) - p(\xi_i)| \leq C_1 h^{k+1}, \quad i = 0, \cdots, k \tag{4.11}$$

Therefore, according to (4.8), (4.9), (4.10) and (4.11), for $x \in I_{j+1}$, we have

$$\begin{aligned}
|u(x) - p(x)| &\leq |u(x) - u_I(x)| + |u_I(x) - p(x)| \\
&= \left| \frac{u^{(k+1)}(\zeta)}{(k+1)!} \prod_{i=0}^{k} (x - \xi_i) \right| + \left| \sum_{i=0}^{k} (u(\xi_i) - p(\xi_i)) l_i(x) \right| \\
&\leq \left| \frac{u^{(k+1)}(\zeta)}{(k+1)!} 2^{k+1} h^{k+1} \right| + C_1 h^{k+1} \sum_{i=0}^{k} |l_i(x)| \\
&\leq \frac{\left|u^{(k+1)}(\zeta)\right|}{(k+1)!} 2^{k+1} h^{k+1} + C_1 h^{k+1} \sum_{i=0}^{k} \left| \prod_{m=0,\, m \neq i}^{k} \frac{x - \xi_i}{\xi_m - \xi_i} \right| \\
&\leq \frac{|M|}{(k+1)!} 2^{k+1} h^{k+1} + C_1 h^{k+1} \sum_{i=0}^{k} \frac{2^k h^k}{\left(\frac{\Delta x_j}{k+2}\right)^k} \\
&\leq \frac{|M|}{(k+1)!} 2^{k+1} h^{k+1} + C_1 h^{k+1} \sum_{i=0}^{k} \frac{2^k h^k}{\left(\frac{ch}{k+2}\right)^k} \\
&= \left(\frac{2^{k+1} M}{(k+1)!} + \frac{C_1 (k+1)(2k+4)^k}{c^k} \right) h^{k+1}
\end{aligned}$$

where $M = \max_{x \in [a,b]} u^{(k+1)}(x)$.

This proves the lemma with $C_2 = \frac{2^{k+1} M}{(k+1)!} + \frac{C_1 (k+1)(2k+4)^k}{c^k}$. ■

Lemma 4.3.2. Under the same assumption as Lemma 4.3.1. If

$$\|u(\cdot) - p(\cdot)\|_{L^2(I_j)} \leq C_1 h^{k+\frac{3}{2}},$$

where $p(x)$ is a polynomial of degree k, then

$$\|u(\cdot) - p(\cdot)\|_{L^2(I_{j+1})} \leq C_2 h^{k+\frac{3}{2}}.$$

Here C_1 and C_2 depend on u and its derivatives but are independent of h.

Proof Denote $\{\xi_i\}_{i=0}^k$ as the set of the shifted $k+1$ Gauss points in the cell I_j:

$$\xi_i = x_j + \frac{z_i}{2}\Delta x_j, \quad i = 0, \cdots, k \tag{4.12}$$

where $\{z_i\}_{i=0}^k$ are the roots of the Legendre polynomial of degree k. Here we also denote $u_I(x)$ to be the Lagrange interpolation polynomial of $u(x)$ based on $\{\xi_i\}$. Then

$$u_I(x) = \sum_{i=0}^{k} u(\xi_i) l_i(x),$$

and

$$u(x) - u_I(x) = \frac{u^{k+1}(\zeta_1)}{(k+1)!} \prod_{i=0}^{k} (x - \xi_i) \tag{4.13}$$

where

$$\min\{\xi_0, \xi_1, \cdots, \xi_n, x\} \leq \zeta_1 \leq \max\{\xi_0, \xi_1, \cdots, \xi_n, x\}.$$

We also rewrite $p(x)$ in the following Lagrange form

$$p(x) = \sum_{i=0}^{k} p(\xi_i) l_i(x)$$

Notice that, by the error estimate of Gauss quadrature formula, we have

$$\begin{aligned}\|u(\cdot)-p(\cdot)\|^2_{L^2(I_j)} &= \int_{I_j}(u(x)-p(x))^2dx \\ &= \sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\omega_i + \frac{(u^2)^{(2k+2)}(\zeta_2)((k+1)!)^4}{(2k+3)[(2k+2)!]^3}\Delta x_j^{2k+3}\end{aligned} \tag{4.14}$$

where ω_i is the correponding weights of ξ_i for Legendre Gauss quadrature rule.

$$\|u(\cdot)-p(\cdot)\|_{L^2(I_{j+1})} \le \|u(\cdot)-u_I(\cdot)\|_{L^2(I_{j+1})} + \|u_I(\cdot)-p(\cdot)\|_{L^2(I_{j+1})} := I + II$$

By (4.13), we have

$$\begin{aligned}I^2 &= \int_{I_{j+1}}(u(x)-u_I(x))^2dx \\ &\le \int_{I_{j+1}}\left(\frac{u^{(k+1)}(\zeta_1)}{(k+1)!}\prod_{i=0}^{k}(x-\xi_i)\right)^2dx \\ &\le \frac{2^{2(k+1)}M^2}{((k+1)!)^2}h^{2k+3}\end{aligned}$$

This gives us

$$I \le \frac{2^{k+1}M}{(k+1)!}h^{k+\frac{3}{2}} \tag{4.15}$$

By (4.8),(4.12) and (4.14),

$$\begin{aligned}II^2 &= \int_{I_{j+1}}(u_I(x)-p(x))^2dx \\ &= \sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\int_{I_{j+1}}l_i^2(x)dx \\ &= \sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\int_{I_{j+1}}\left(\prod_{m=0,\,m\neq i}^{k}\frac{x-\xi_i}{\xi_m-\xi_i}\right)^2dx\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\int_{I_{j+1}}\left(\prod_{m=0,\,m\neq i}^{k}\frac{2h}{\xi_m-\xi_i}\right)^2 dx\\
&= \sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\left(\prod_{m=0,\,m\neq i}^{k}\frac{2h}{\frac{z_m-z_i}{2}\Delta x_j}\right)^2 \Delta x_j\\
&\leq \sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\left(\prod_{m=0,\,m\neq i}^{k}\frac{4}{c(z_m-z_i)}\right)^2 h\\
&= \sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\omega_i\left(\prod_{m=0,\,m\neq i}^{k}\frac{4}{c\sqrt{\omega_i}(z_m-z_i)}\right)^2 h\\
&\leq M^2h\sum_{i=0}^{k}(u(\xi_i)-p(\xi_i))^2\omega_i\\
&= M_2^2h\left(\|u(x)-p(x)\|^2_{L^2(I_j)}-\frac{(u^2)^{(2k+2)}(\zeta_2)((k+1)!)^4}{(2k+3)[(2k+2)!]^3}\Delta x_j^{2k+3}\right)\\
&\leq M_2^2h\left(\|u(x)-p(x)\|^2_{L^2(I_j)}+\left|\frac{(u^2)^{(2k+2)}(\zeta_2)((k+1)!)^4}{(2k+3)[(2k+2)!]^3}\Delta x_j^{2k+3}\right|\right)\\
&\leq M_2^2h\left(C_1^2h^{2k+3}+\frac{M_3((k+1)!)^4}{(2k+3)[(2k+2)!]^3}h^{2k+3}\right)\\
&= \underbrace{\left(M_2^2C_1^2+\frac{M_2^2M_3((k+1)!)^4}{(2k+3)[(2k+2)!]^3}\right)}_{C_3^2}h^{2k+4}
\end{aligned}
$$

where $M_2=\max_{0\leq i\leq k}\prod_{m=0,\,m\neq i}^{k}\frac{4}{c\sqrt{\omega_i}(z_m-z_i)}$ and $M_3=\max_{x\in I_j}(u^2)^{(2k+2)}(x)$. Hence

$$II\leq C_3h^{k+2}. \tag{4.16}$$

The conclusion of this lemma can be obtained by (4.15) and (4.16). ■

Following [38, 35, 12, 41], the normalized nonlinear weights are defined as

$$\omega_j = \frac{\bar{\omega}_j}{\sum_l \bar{w}_l}, \tag{4.17}$$

where the non-normalized nonlinear weights $\bar{w}_j$ are functions of the linear weights γ_j and the so-called smoothness indicators β_j as follows:

$$\bar{\omega}_j = \frac{\gamma_j}{(\varepsilon + \beta_j)^r}. \tag{4.18}$$

We use $\varepsilon = 10^{-6}$ and $r = 2$ in all computations in this chapter. As in [38, 12], we use the following smoothness indicator for the one-dimensional case:

$$\beta_j = \sum_{l=1}^{k} \int_{I_j} \Delta x_j^{2l-1} \left(\frac{d^l}{dx^l} p_j(x) \right)^2 dx. \tag{4.19}$$

For the two-dimensional case and more details about this smoothness indicator, we refer to [38, 12, 59]. Notice that, because we have used the complete information of the three polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$ in the three cells I_{j-1}, I_j, I_{j+1}, we do not have extra requirements on the linear weights in order to maintain the original high order accuracy. The linear weights can be chosen to be any set of positive numbers adding up to one. Since for smooth solutions the central cell is usually the best one, we put a larger linear weight on the central cell than on the neighboring cells, i.e.

$$\gamma_1 \gg \gamma_0, \qquad \gamma_1 \gg \gamma_2.$$

Lower values of the ratio $\frac{\gamma_1}{\gamma_0}$, $\frac{\gamma_1}{\gamma_2}$ yield better results on discontinuities while larger values are usually better for smooth solutions. In our numerical tests we take

$$\gamma_0 = 0.001,\ \gamma_1 = 0.998,\ \gamma_2 = 0.001, \tag{4.20}$$

which maintains the original high order in smooth regions and can keep essentially non-oscillatory shock transition.

We summarize the WENO limiting procedure for one-dimensional scalar conservation laws as follows. Assuming that DG solution at time level n is u^n, for $j = 1, \cdots, N$,

1. Use the minmod limiter described in Subsection 4.3.1 to detect whether I_j is a troubled cell or not.

2. If I_j is not a troubled cell, then $u^{n,new}|_{I_j} = u^n|_{I_j}$.

   If I_j is a troubled cell, then

   (a) Denote u^n on the cells I_{j-1}, I_j, I_{j+1} as $p_0(x)$, $p_1(x)$, $p_2(x)$, respectively and modify $p_0(x)$, $p_2(x)$ to $\tilde{p}_0(x)$, $\tilde{p}_2(x)$ using (4.6).

   (b) Determine the linear weights by (4.20).

   (c) Compute the smoothness indicators β_l using (4.19) for $l = 0, 1, 2$.

   (d) Compute the normalized nonlinear weights w_l using (4.17) and (4.18) for $l = 0, 1, 2$.

   (e) The reconstruction polynomial is given by (4.7), i.e. $u^{n,new}|_{I_j} = \omega_0\tilde{p}_0(x) + \omega_1 p_1(x) + \omega_2\tilde{p}_2(x)$.

For the two-dimensional scalar case with rectangular meshes considered in this chapter, the limiting procedure is similar as described above. The WENO reconstruction polynomial on the troubled cell $I_{i,j}$ is a convex combination of the polynomials on this cell and its four neighboring cells $\{I_{i,j-1}, I_{i,j+1}, I_{i-1,j}, I_{i+1,j}\}$ suitably modified to maintain the original cell average on the target cell $I_{i,j}$. For our numerical tests, we put a larger linear weight 0.996 on the troubled cell $I_{i,j}$ and the neighboring

cells $\{I_{i,j-1},\ I_{i,j+1},\ I_{i-1,j},\ I_{i+1,j}\}$ get the smaller linear weight 0.001 (Figure 4.2). The nonlinear weights of the convex combination follow the classical WENO procedure, with smoothness indicators still computed as the sums of L^2-norm squares of all the derivatives of the respective polynomials, as defined in (4.21)

$$\beta_l = \sum_{1\le|\alpha|\le k} \int_{I_{i,j}} (\Delta x_i \Delta y_j)^{|\alpha|-1} \left(D^\alpha p_l(x,y)\right)^2 dx\, dy \tag{4.21}$$

where $\alpha = (\alpha_1, \alpha_2)$, $|\alpha| = \alpha_1 + \alpha_2$ and $D^\alpha = \dfrac{\partial^\alpha}{\partial x^{\alpha_1} \partial y^{\alpha_2}}$.

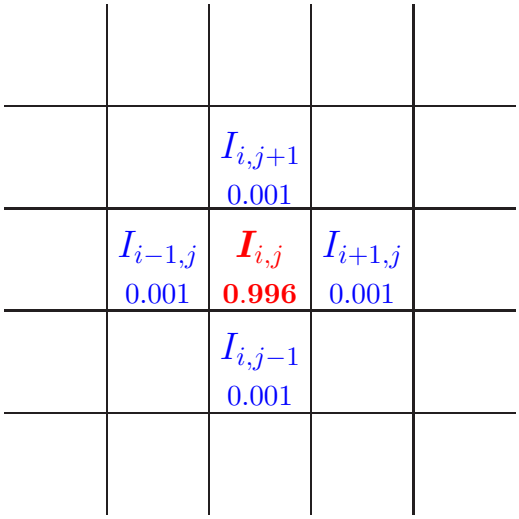


Figure 4.2: Linear weights for two dimensional rectangular mesh.

4.3.3 WENO Limiting Procedure for Systems

In this subsection, we present the details of the WENO limiting procedure for systems.

One-dimensional systems. Consider equation (4.1) where u and f are vectors

with m components. In order to achieve better non-oscillatory qualities, the WENO reconstruction limiter is applied with a local characteristic field decompositions, see, e.g. [59] for more details.

Denote the Jacobian matrix by $A_j = \frac{\partial f}{\partial u}\big|_{\bar{u}_j}$. Denote the left and right eigenvectors of A_j by $l_j^{(p)}$, $r_j^{(p)}$, $p = 1, \cdots, m$, normalized so that $l_j^{(p)} \cdot r_j^{(q)} = \delta_{pq}$. Let $R(\bar{u}_j)$ be the $m \times m$ matrix with the right eigenvectors as columns, i.e.

$$R(\bar{u}_j) = \left(r_j^{(1)}, r_j^{(2)}, \cdots, r_j^{(m)}\right). \tag{4.22}$$

Clearly, $R^{-1}(\bar{u}_j)$ is a $m \times m$ matrix with the left eigenvectors as rows, that is

$$R^{-1}(\bar{u}_j) = \left(l_j^{(1)}, l_j^{(2)}, \cdots, l_j^{(m)}\right)^T. \tag{4.23}$$

The WENO limiting procedure for one-dimensional system case is given as follows. Assume that the DG solution at time level n is u^n (for simplicity, we will use the notation u for the following description), for $j = 1, \cdots, N$,

1. Compute $R = R(\bar{u}_j)$ and R^{-1} as defined in (4.22) and (4.23).

2. Compute $\Delta_+\bar{v}_j = R^{-1}\left(\bar{u}_{j+1} - \bar{u_j}\right)$, $\quad \Delta_-\bar{v}_j = R^{-1}\left(\bar{u}_j - \bar{u}_{j-1}\right)$, $\tilde{v}_j = R^{-1}\left(u^-_{j+\frac{1}{2}} - \bar{u}_j\right)$ and $\tilde{\tilde{v}}_j = R^{-1}\left(\bar{u}_j - u^+_{j-\frac{1}{2}}\right)$ respectively.

3. Compute $\tilde{v}_j^{(mod)} = \tilde{m}(\tilde{v}_j, \Delta_+\bar{v}_j, \Delta_-\bar{v}_j)$ and $\tilde{\tilde{v}}_j^{(mod)} = \tilde{m}(\tilde{\tilde{v}}_j, \Delta_+\bar{v}_j, \Delta_-\bar{v}_j)$ with the modified minmod function $\tilde{m}$ define in (2.11) for each component of the vectors.

4. If

$$\tilde{v}_j^{(mod)} = \tilde{v}_j \quad \& \quad \tilde{\tilde{v}}_j^{(mod)} = \tilde{\tilde{v}}_j, \tag{4.24}$$

then I_j is not a troubled cell and $u^{new}|_{I_j} = u|_{I_j}$. Otherwise,

(i) Denote the DG polynomial u on the cells I_{j+l} as u_{j+l}. Project u_{j+l} into the characteristic fields $v_{j+l} = R^{-1}u_{j+l}$, for $l = -1, 0, 1$. Note that v_{j+l} is a m-component vector and each component is a polynomial.

(ii) Perform Steps (a), (b), (c), (d), (e) as in the one-dimensional scalar case for each component of v_j which is a troubled component, namely the corresponding component makes the condition (4.24) not satisfied. The updated vector v_j is denoted as v_j^{new}.

(iii) $u^{new}|_{I_j} = Rv_j^{new}$.

Two-dimensional systems. Considering the following two-dimensional system:

$$u_t + f(u)_x + g(u)_y = 0, \tag{4.25}$$

where u, $f(u)$ and $g(u)$ are vectors. The DG scheme with Euler forward time discretization for solving (4.25) on a rectangular mesh is

$$\begin{aligned}
&\int_{I_{i,j}} \frac{u^{n+1} - u}{\Delta t} v \, dx \, dy = \\
&\int_{I_{i,j}} f(u) v_x \, dx \, dy - \int_{I_j} \hat{f}_{i+\frac{1}{2}}(y) v(x^-_{i+\frac{1}{2}}, y) \, dy + \int_{I_j} \hat{f}_{i-\frac{1}{2}}(y) v(x^+_{i-\frac{1}{2}}, y) \, dy \\
&+ \int_{I_{i,j}} g(u) v_y \, dx \, dy - \int_{I_i} \hat{g}_{j+\frac{1}{2}}(x) v(x, y^-_{j+\frac{1}{2}}) \, dx + \int_{I_i} \hat{g}_{i-\frac{1}{2}}(x) v(x, y^+_{j-\frac{1}{2}}) \, dx,
\end{aligned} \tag{4.26}$$

where $u = u^n$ and $\hat{f}$ and $\hat{g}$ are monotone numerical fluxes.

For two-dimensional systems, we need to be more careful when using the WENO limiting procedure with a local characteristic field decompositions, since there are two Jacobian matrices corresponding to fluxes in the x and y directions respectively, and therefore two sets of eigenspaces. We perform the characteristic-wise WENO reconstruction in the x-direction and y-direction separately.

Assume that $I_{i,j}$ is a troubled cell,

1. In the x-direction, we choose the polynomials on the cells $I_{i-1,j}$, $I_{i,j}$, $I_{i+1,j}$ to reconstruct a new polynomial $u_{i,j}^{x,new}$ using the characteristic-wise WENO limiting procedure with the Jacobian matrix $\frac{\partial f}{\partial u}$ as in the one-dimensional system case.

2. Similarly, in the y-direction, we choose the polynomials on the cells $I_{i,j-1}$, $I_{i,j}$, $I_{i,j+1}$ to reconstruct a new polynomial $u_{i,j}^{y,new}$ using the characteristic-wise WENO limiting procedure with the Jacobian matrix $\frac{\partial g}{\partial u}$.

3. $u^{new}|_{I_{i,j}} = \frac{1}{2}\left(u_{i,j}^{x,new} + u_{i,j}^{y,new}\right)$.

After limiting the DG solution, we advance it to the next time level for the Euler forward time discretization by

$$\int_{I_{i,j}} \frac{u^{n+1} - u^{new}}{\Delta t} v \, dx \, dy =$$
$$\int_{I_{i,j}} f(u^{x,new}) v_x \, dx \, dy + \int_{I_{i,j}} g(u^{y,new}) v_y \, dx \, dy$$
$$- \int_{I_j} \hat{f}(u^{x,new})_{i+\frac{1}{2}}(y) v(x_{i+\frac{1}{2}}^-, y) \, dy + \int_{I_j} \hat{f}(u^{x,new})_{i-\frac{1}{2}}(y) v(x_{i-\frac{1}{2}}^+, y) \, dy$$
$$- \int_{I_i} \hat{g}(u^{y,new})_{j+\frac{1}{2}}(x) v(x, y_{j+\frac{1}{2}}^-) \, dx + \int_{I_i} \hat{g}(u^{y,new})_{i-\frac{1}{2}}(x) v(x, y_{j-\frac{1}{2}}^+) \, dx.$$

The SSP Runge-Kutta time discretization is just a convex combination of such Euler forward steps.

4.4 Numerical Experiments

In this section, we provide numerical experiments to demonstrate the performance of the WENO limiter for RKDG methods described in Section 4.3. Even though the advantage of simplicity is most evident for higher-dimensional unstructured meshes, we show in this chapter only results for one and two dimensional structured meshes.

For all the computational results, we use the local Lax-Friedrichs flux. For the one-dimensional examples, the CFL number is set to be 0.3 for the P^1 case, 0.15 for the P^2 case and 0.1 for the P^3 case (for the P^3 case Δt is further reduced in the accuracy test).

We have used both uniform and nonuniform meshes in the numerical experiments, obtaining similar results. The nonuniform meshes are obtained from a 20% random perturbation of each node of the uniform mesh. Take the one-dimensional case as an example. The cell boundary point is now $x_{j+\frac{1}{2}} + 20\%(r_{j+\frac{1}{2}} - 0.5)\Delta x$, where $x_{j+\frac{1}{2}}$ and Δx are taking values from the uniform mesh and $r_{j+\frac{1}{2}}$ is a random number from the uniform distribution over the range $(0, 1)$. We will only show results with some accuracy tests on nonuniform meshes as representative tests.

For all the accuracy tests (Tables 4.1-4.12), in order to see the effect of the WENO limiter on the accuracy of the RKDG method, we use the TVB minmod limiter with a small TVB constant $M = 0.01$ to identify troubled cells, resulting in many good cells being identified as troubled cells. For Figures 4.3-4.14, the solid lines are for

the exact solutions or grid converged solutions, and the symbols "+" are for the numerical solutions (just one point per cell is plotted).

4.4.1 Scalar Conservation Laws

Example 4.4.1. We consider the 1D transport equation:

$$u_t + u_x = 0, \quad 0 \le x \le 2\pi \tag{4.27}$$

with the initial condition $u(x,0) = \sin x$ and periodic boundary conditions. The exact solution of (4.27) is

$$u(x,t) = \sin(x-t).$$

Table 4.1 lists the errors and orders of the accuracy for the RKDG method with a WENO limiter compared to the original RKDG method. We can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original RKDG method.

Table 4.1: 1D Transport equation. $t = 2\pi$. Uniform mesh with N cells

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	3.20E-03		1.73E-02		2.66E-02		7.33E-02	
	40	7.82E-04	2.03	4.76E-03	1.87	5.05E-03	2.40	2.90E-02	1.34
	80	1.98E-04	1.98	1.24E-03	1.94	4.76E-04	3.41	4.28E-03	2.76
	160	5.02E-05	1.98	3.15E-04	1.97	5.33E-05	3.16	4.23E-04	3.34
	320	1.27E-05	1.99	7.96E-05	1.99	1.27E-05	2.07	8.27E-05	2.35
P^2	20	8.73E-05		5.09E-04		1.21E-04		5.10E-04	
	40	1.10E-05	2.99	6.44E-05	2.98	1.30E-05	3.21	6.44E-05	2.99
	80	1.38E-06	2.99	8.07E-06	3.00	1.49E-06	3.13	8.07E-06	3.00
	160	1.73E-07	3.00	1.01E-06	3.00	1.79E-07	3.06	1.01E-06	3.00
	320	2.17E-08	3.00	1.26E-07	3.00	2.20E-08	3.03	1.26E-07	3.00

Example 4.4.2. We consider the 2D transport equation:

$$u_t + u_x + u_y = 0, \quad 0 \le x, y \le 2\pi \tag{4.28}$$

with the initial condition $u(x, y, 0) = \sin(x + y)$ and periodic boundary conditions. The exact solution of (4.28) is

$$u(x, y, t) = \sin(x + y - 2t).$$

Table 4.2 lists the errors and orders of the accuracy for the RKDG method with a WENO limiter compared to the original RKDG method. Similarly, we can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of accuracy of the original RKDG method.

Table 4.2: 2D Transport equation. $t = \pi$. Uniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	6.69E-03		4.68E-02		1.76E-02		6.40E-02	
	40	1.30E-03	2.37	1.34E-02	1.80	3.77E-03	2.22	2.41E-02	1.41
	80	3.09E-04	2.07	3.56E-03	1.91	7.16E-04	2.40	6.94E-03	1.80
	160	7.86E-05	1.98	9.15E-04	1.96	8.28E-05	3.11	1.03E-03	2.75
	320	2.01E-05	1.97	2.32E-04	1.98	2.01E-05	2.04	2.34E-04	2.15
P^2	10	2.79E-03		3.16E-02		2.69E-03		2.99E-02	
	20	3.40E-04	3.04	4.03E-03	2.97	3.36E-04	3.00	4.05E-03	2.89
	40	4.29E-05	2.99	5.07E-04	2.99	4.28E-05	2.97	5.07E-04	3.00
	80	5.40E-06	2.99	6.35E-05	3.00	5.40E-06	2.99	6.35E-05	3.00
	160	6.79E-07	2.99	7.95E-06	3.00	6.79E-07	2.99	7.95E-06	3.00
	320	8.51E-08	3.00	9.95E-07	3.00	8.51E-08	3.00	9.95E-07	3.00

Example 4.4.3. We consider the Burgers equation:

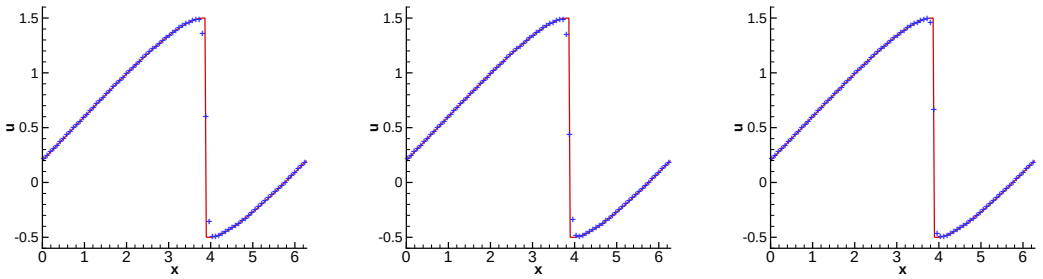
$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad 0 \le x \le 2\pi \tag{4.29}$$

with the initial condition $u(x, 0) = 0.5 + \sin x$ and periodic boundary conditions. The exact solution is smooth up to $t = 1$, then it develops a moving shock which also

acts with the rarefaction waves. We can get the exact solution by Newton iteration. For details, see [34]. The errors at t = 0.5 when the solution is smooth are listed in Tables 4.3 and 4.4. We see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original RKDG method. In Figure 4.3, we show the RKDG solutions with a WENO limiter at $t = 1.5$ using 80 cells. We can see that the schemes of all orders perform well in capturing this discontinuity without oscillations.

Table 4.3: 1D Burgers equation. $t = 0.5$. Uniform mesh with N cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	3.81E-03		3.82E-02		7.65E-03		6.86E-02	
	40	9.46E-04	2.01	1.03E-02	1.90	1.81E-03	2.08	2.01E-02	1.77
	80	2.35E-04	2.01	2.65E-03	1.95	3.06E-04	2.57	3.10E-03	2.70
	160	5.89E-05	2.00	6.73E-04	1.98	5.93E-05	2.37	6.73E-04	2.20
	320	1.47E-05	2.00	1.70E-04	1.98	1.47E-05	2.01	1.70E-04	1.98
P^2	20	2.73E-04		5.07E-03		2.68E-04		5.08E-03	
	40	4.22E-05	2.69	8.96E-04	2.50	4.17E-05	2.68	8.96E-04	2.50
	80	6.17E-06	2.77	1.60E-04	2.48	6.17E-06	2.76	1.60E-04	2.48
	160	8.86E-07	2.80	2.55E-05	2.65	8.92E-07	2.79	2.55E-05	2.65
	320	1.25E-07	2.82	3.79E-06	2.75	1.27E-07	2.81	3.79E-06	2.75
P^3	20	1.85E-05		3.64E-04		2.17E-05		3.66E-04	
	40	1.00E-06	4.20	3.69E-05	3.30	1.03E-06	4.40	3.69E-05	3.31
	80	6.08E-08	4.05	2.22E-06	4.06	6.17E-08	4.06	2.22E-06	4.06
	160	3.76E-09	4.02	1.44E-07	3.95	3.83E-09	4.01	1.44E-07	3.95
	320	2.33E-10	4.01	9.19E-09	3.97	2.45E-10	3.97	9.19E-09	3.97



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.3: Burgers equation at $t = 1.5$ with $N = 80$ cells.

Table 4.4: 1D Burgers equation. $t = 0.5$. Nonuniform mesh with N cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	3.84E-03		4.01E-02		6.35E-03		7.39E-02	
	40	9.85E-04	1.96	1.28E-02	1.65	1.93E-03	1.72	2.07E-02	1.84
	80	2.44E-04	2.02	3.09E-03	2.05	3.16E-04	2.61	3.16E-03	2.71
	160	6.14E-05	1.99	8.28E-04	1.90	6.22E-05	2.35	8.28E-04	1.93
	320	1.53E-05	2.00	2.36E-04	1.81	1.53E-05	2.02	2.36E-04	1.81
P^2	20	2.86E-04		4.87E-03		2.79E-04		4.88E-03	
	40	4.43E-05	2.69	7.68E-04	2.66	4.38E-05	2.67	7.68E-04	2.67
	80	6.37E-06	2.80	1.61E-04	2.25	6.37E-06	2.78	1.61E-04	2.25
	160	9.08E-07	2.81	2.83E-05	2.51	9.16E-07	2.80	2.83E-05	2.51
	320	1.30E-07	2.81	4.12E-06	2.78	1.32E-07	2.80	4.12E-06	2.78
P^3	20	1.85E-05		3.91E-04		1.85E-05		3.95E-04	
	40	1.01E-06	4.20	3.20E-05	3.61	1.03E-06	4.17	3.20E-05	3.62
	80	6.42E-08	3.97	2.79E-06	3.52	6.55E-08	3.97	2.79E-06	3.52
	160	4.40E-09	3.87	2.80E-07	3.32	4.42E-09	3.89	2.80E-07	3.32
	320	2.72E-10	4.02	1.66E-08	4.07	2.77E-10	4.00	1.66E-08	4.07

Example 4.4.4. A two-dimensional version of Example 4.4.3

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, \quad 0 \le x, y \le 2\pi \tag{4.30}$$

is tested with initial condition $u(x, y, 0) = 0.5 + \sin(x + y)$ and periodic boundary conditions. The exact solution is one-dimensional depending only on $\xi = x + y$; however, our meshes are rectangular in the (x, y) coordinates, and thus this example is a truly two-dimensional test problem. As in Example 4.4.3, we collect the L_1 and L_∞ errors at $t = 0.25$ (smooth solution) in Tables 4.5 and 4.6. At $t = 0.5$, a shock begins to form. We compute the solutions of the RKDG methods using P^k polynomials with a WENO limiter with 80×80 meshes until $t = 0.75$ and plot the solution on the diagonal cells in Figures 4.4 and 4.5. Again, we can see that the WENO limiter obtains uniform high order accuracy and sharp, non-oscillatory shock transitions for the RKDG methods.

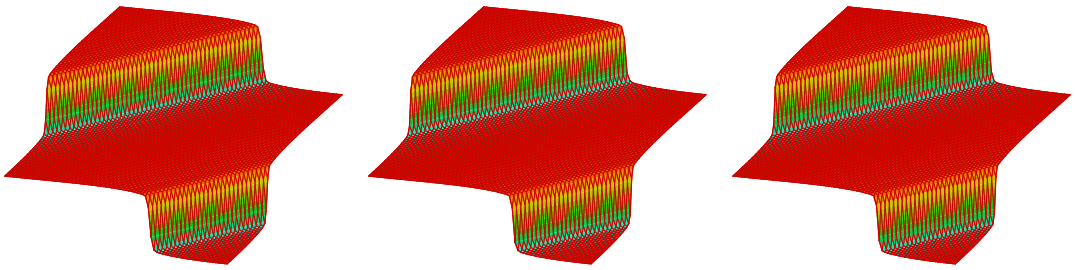
Example 4.4.5. Our last scalar example is the Buckley-Leverett problem that is

Table 4.5: 2D Burgers equation at $t = 0.25$. Uniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
	20	7.56E-03		1.22E-01		1.11E-02		2.07E-01	
	40	1.92E-03	1.98	3.55E-02	1.78	3.09E-03	1.85	6.57E-02	1.65
P^1	80	4.79E-04	2.00	9.36E-03	1.92	6.60E-04	2.23	9.36E-03	2.81
	160	1.20E-04	2.00	2.39E-03	1.97	1.23E-04	2.43	2.39E-03	1.97
	320	2.99E-05	2.00	6.04E-04	1.98	2.99E-05	2.04	6.04E-04	1.98
	20	8.62E-04		4.37E-02		8.62E-04		4.37E-02	
	40	1.16E-04	2.89	6.08E-03	2.84	1.16E-04	2.89	6.08E-03	2.84
P^2	80	1.50E-05	2.96	9.60E-04	2.66	1.50E-05	2.96	9.60E-04	2.66
	160	1.90E-06	2.98	1.35E-04	2.83	1.90E-06	2.98	1.35E-04	2.83
	320	2.43E-07	2.97	1.83E-05	2.88	2.44E-07	2.96	1.83E-05	2.88
	20	1.30E-04		9.32E-03		1.56E-04		9.03E-03	
	40	9.28E-06	3.80	7.54E-04	3.63	1.00E-05	3.96	7.54E-04	3.58
P^3	80	5.91E-07	3.97	6.05E-05	3.64	6.21E-07	4.01	6.05E-05	3.64
	160	3.73E-08	3.99	3.98E-06	3.92	4.50E-08	3.79	3.98E-06	3.92
	320	2.35E-09	3.99	2.56E-07	3.96	2.66E-09	3.83	2.56E-07	3.96

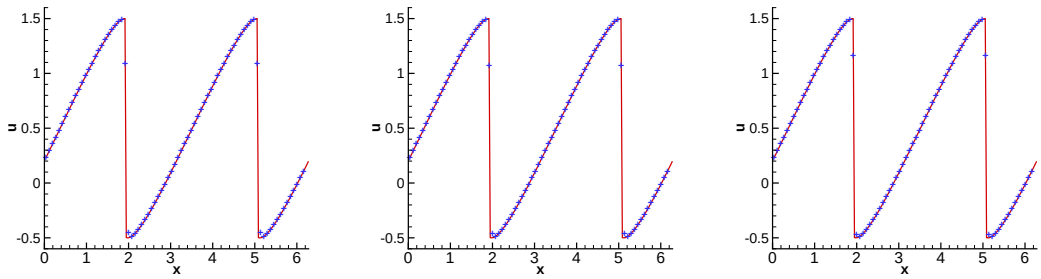
Table 4.6: 2D Burgers equation at $t = 0.25$. Nonuniform mesh with $N \times N$ cells

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
	20	7.70E-03		1.38E-01		1.24E-02		2.82E-01	
	40	1.99E-03	1.96	4.62E-02	1.58	3.48E-03	1.84	8.75E-02	1.69
P^1	80	4.93E-04	2.01	1.21E-02	1.93	6.86E-04	2.35	1.21E-02	2.85
	160	1.24E-04	2.00	3.38E-03	1.84	1.28E-04	2.42	3.38E-03	1.84
	320	3.10E-05	1.99	8.50E-04	1.99	3.10E-05	2.05	8.50E-04	1.99
	20	8.95E-04		5.48E-02		8.92E-04		5.48E-02	
	40	1.21E-04	2.89	8.18E-03	2.74	1.21E-04	2.88	8.18E-03	2.74
P^2	80	1.55E-05	2.96	1.37E-03	2.57	1.57E-05	2.95	1.37E-03	2.57
	160	1.95E-06	3.00	1.98E-04	2.79	1.99E-06	2.97	1.98E-04	2.79
	320	2.50E-07	2.96	2.77E-05	2.84	2.60E-07	2.94	2.77E-05	2.84
	20	1.35E-04		1.27E-02		1.60E-04		1.27E-02	
	40	9.63E-06	3.81	1.08E-03	3.56	1.04E-05	3.95	1.08E-03	3.56
P^3	80	6.18E-07	3.96	1.04E-04	3.37	6.89E-07	3.91	1.04E-04	3.37
	160	3.90E-08	3.99	6.99E-06	3.90	6.18E-08	3.48	1.25E-05	3.05
	320	2.35E-09	3.99	2.56E-07	3.96	2.66E-09	3.83	2.56E-07	3.96



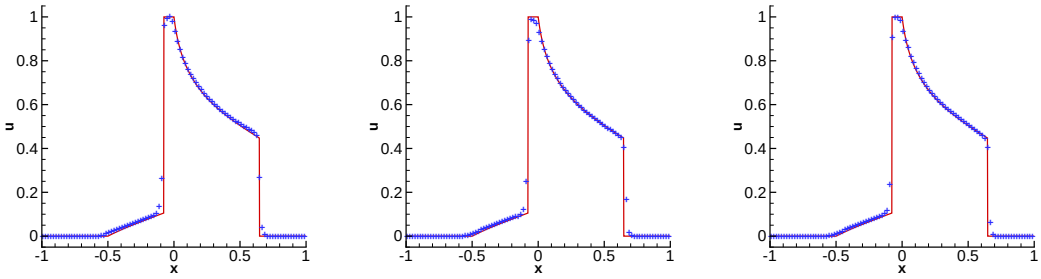
(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.4: 2D Burgers solution at $t = 0.75$ with 80×80 cells.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.5: 2D Burgers solution that cuts along the diagonal at $t = 0.75$ with 80×80 cells.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.6: Buckley-Leverett problem at $t = 0.4$ with 80 cells.

governed by the equation

$$u_t + \left(\frac{4u^2}{4u^2 + (1-u)^2} \right)_x = 0, \tag{4.31}$$

with the initial condition $u = 1$ for $-\frac{1}{2} \leq x \leq 0$ and $u = 0$ elsewhere. The exact solution is a shock-rarefaction-contact discontinuity mixture. The solution is computed up to $t = 0.4$. Figure 4.6 shows the numerical solutions of RKDG methods with a WENO limiter using $N = 80$ cells. Again, all schemes perform similarly well for this example.

4.4.2 Euler System in One Dimension

The 1D Euler system is given by

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x = 0, \tag{4.32}$$

where $\boldsymbol{u} = (\rho,\ \rho v,\ E)^T$ and $\boldsymbol{f}(\boldsymbol{u}) = (\rho v, \rho v^2 + p, v(E+p))^T$. Here ρ is the density, v is the velocity, E is the total energy, and p is the pressure, with

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho v^2\right); \tag{4.33}$$

$\gamma = 1.4$ is used in the computation. For details of the Jacobian, its eigenvalues, eigenvectors, etc., see [34, 54]. We consider the following typical examples.

Example 4.4.6. This example is to test the order of accuracy for the RKDG methods with our WENO limiter. The initial condition is set to be $\rho(x,0) = 1 + 0.2\sin x$, $v(x,0) = 1$, $p(x,0) = 1$ and the boundaries are periodic. The exactly solution is $\rho(x,t) = 1 + 0.2\sin(x-t)$, $v(x,t) = 1$ and $p(x,t) = 1$. Tables 4.7 and 4.8 list the errors and orders of accuracy for the RKDG method with a WENO limiter compared to the RKDG method without a limiter. Again, we can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of accuracy of the original RKDG method.

Table 4.7: 1D Euler system with initial condition $\rho(x,0) = 1 + 0.2\sin x$, $v(x,0) = 1$, $p(x,0) = 1$ at $t = 2\pi$. Uniform mesh with N cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	4.68E-04		2.98E-03		6.81E-03		1.60E-02	
	40	1.23E-04	1.93	7.80E-04	1.93	5.10E-04	3.74	3.10E-03	2.37
	80	3.20E-05	1.95	1.99E-04	1.97	3.52E-05	3.86	2.63E-04	3.56
	160	8.17E-06	1.97	5.04E-05	1.98	8.20E-06	2.10	5.27E-05	2.32
	320	2.07E-06	1.98	1.27E-05	1.99	2.07E-06	1.99	1.27E-05	2.05
P^2	20	2.81E-05		1.56E-04		4.45E-05		1.95E-04	
	40	3.63E-06	2.95	2.04E-05	2.94	4.69E-06	3.25	2.04E-05	3.26
	80	4.57E-07	2.99	2.57E-06	2.99	5.25E-07	3.16	2.57E-06	2.99
	160	5.72E-08	3.00	3.22E-07	3.00	6.15E-08	3.09	3.22E-07	3.00
	320	7.15E-09	3.00	4.02E-08	3.00	7.41E-09	3.05	4.02E-08	3.00

Example 4.4.7. We consider here two well-known problems of the Euler equation

Table 4.8: 1D Euler system with initial condition $\rho(x,0) = 1 + 0.2\sin x$, $v(x,0) = 1$, $p(x,0) = 1$ at $t = 2\pi$. Nonuniform mesh with N cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	4.99E-04		3.88E-03		6.75E-03		1.59E-02	
	40	1.36E-04	1.88	1.08E-03	1.84	5.57E-04	3.60	3.24E-03	2.30
	80	3.49E-05	1.96	2.83E-04	1.94	4.09E-05	3.77	3.23E-04	3.32
	160	8.96E-06	1.96	6.81E-05	2.05	9.00E-06	2.18	7.19E-05	2.17
	320	2.28E-06	1.98	1.73E-05	1.98	2.28E-06	1.98	1.73E-05	2.06
P^2	20	3.05E-05		1.85E-04		5.40E-05		2.35E-04	
	40	3.95E-06	2.95	2.57E-05	2.85	5.45E-06	3.31	2.57E-05	3.20
	80	4.99E-07	2.98	3.28E-06	2.97	5.06E-07	3.43	3.28E-06	2.97
	160	6.20E-08	3.01	4.13E-07	2.99	6.86E-08	2.88	4.13E-07	2.99
	320	7.80E-09	2.99	5.42E-08	2.93	8.34E-09	3.04	5.42E-08	2.93

(4.32) which have the following Riemann type initial conditions:

$$\boldsymbol{u}(x,0) = \begin{cases} \boldsymbol{u}_L, & x < 0, \\ \boldsymbol{u}_R, & x > 0. \end{cases} \tag{4.34}$$

The first one is the Sod problem [64]. The initial data are

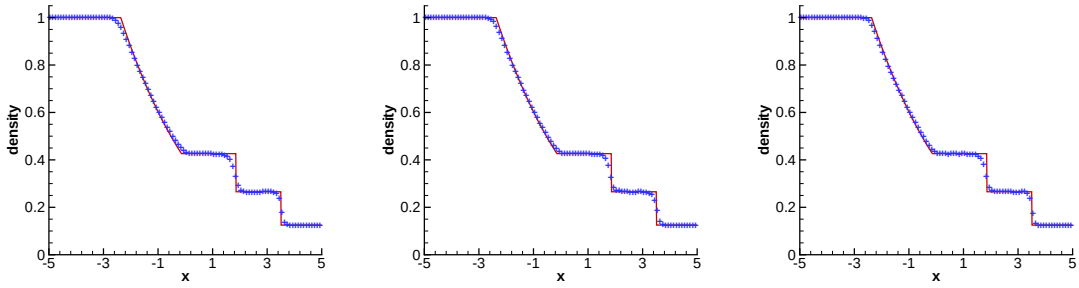
$$(\rho_L,\, v_L,\, p_L) = (1,\, 0,\, 1); \quad (\rho_R,\, v_R,\, p_R) = (0.125,\, 0,\, 0.1). \tag{4.35}$$

The second one is the Lax problem [43], with the initial data

$$(\rho_L,\, v_L,\, p_L) = (0.445,\, 0.698,\, 3.528); \quad (\rho_R,\, v_R,\, p_R) = (0.5,\, 0,\, 0.571). \tag{4.36}$$

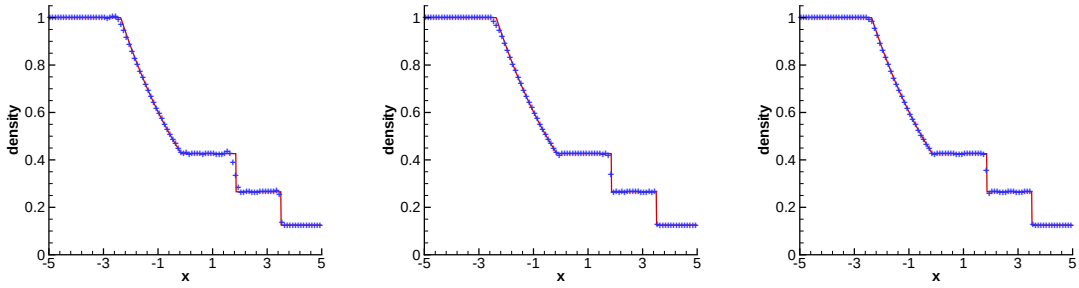
The numerical results with the WENO limiter are in Figures 4.7-4.10. To save space, we only show the plots for density. The figures for velocity and pressure are not shown. For Figures 4.7 and 4.9, we use the TVB constant $M = 0.01$. We can see that there are no oscillations near the discontinuities, however we also observe rather severe smearing, especially for the contact discontinuity, due to this strong limiter. For the Sod problem, relaxing the limiting by taking $M = 30$ improves the

smearing at the price of slight over- and under-shoots, comparing Figure 4.8 with 4.7. The Lax problem is more sensitive to the parameter M, we relax the limiting by taking $M = 7$, 30, 50 for $k = 1, k = 2, k = 3$ respectively, see, Figure 4.10.



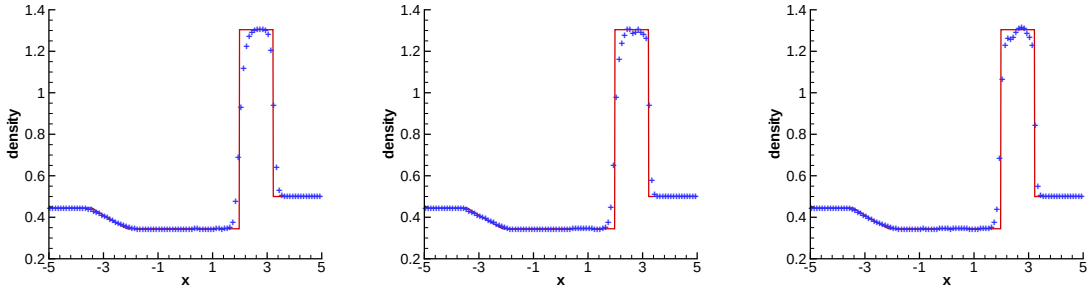
(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.7: Sod problem. $t = 2$. $M = 0.01$. $N = 100$. Density.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

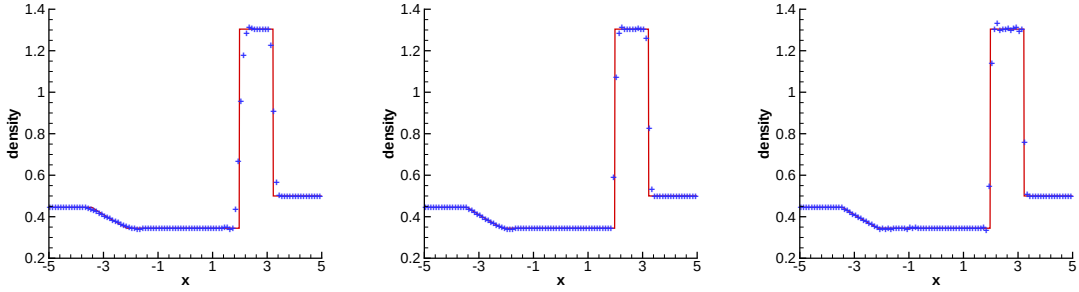
Figure 4.8: Sod problem. $t = 2$. $M = 30$. $N = 100$. Density.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.9: Lax problem. $t = 1.3$. $M = 0.01$. $N = 100$. Density.

Example 4.4.8. To demonstrate the advantage of higher order methods, we use



(a) $k = 1, M = 7$ (b) $k = 2, M = 30$ (c) $k = 3, M = 50$

Figure 4.10: Lax problem. $t = 2$. $N = 100$. Density.

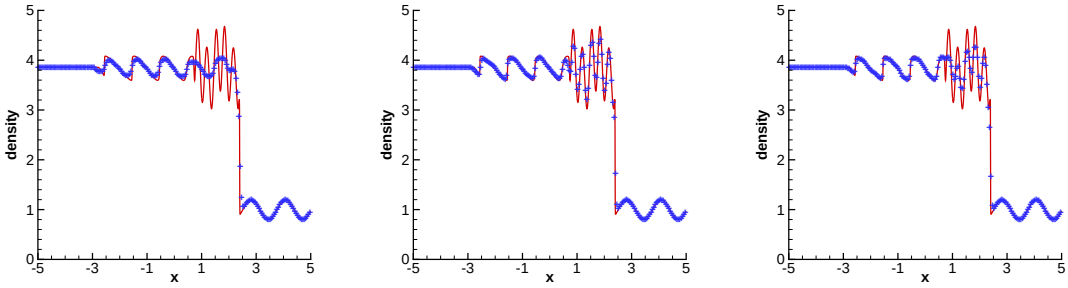
the Euler equation (4.32) with initial condition

$$\begin{aligned} &(\rho_L,\, v_L,\, p_L) = (3.857143,\, 2.629369,\, 10.333333), && \text{when } x < -4, \\ &(\rho_R,\, v_R,\, p_R) = (1 + 0.2\sin(5x),\, 0,\, 1), && \text{when } x \geq -4. \end{aligned} \tag{4.37}$$

This example was used in [63]. It describes the interaction of a Mach 3 shock with a density wave. A Mach 3 shock is initially located at $x = -4$ and moves to the right. A sine wave is superimposed to the density to the right region to the shock. It contains both shocks and fine structures in smooth regions. Our results are shown in Figures 4.11-4.12. The solid lines are the referenced "exact" solution, which is a converged solution computed by the fifth order finite difference WENO scheme [38] with 2000 grid points. Again, we explore the effect of the TVB constant M in the minmod limiter to identify troubled cells. If M is adjusted adequately, schemes of all orders can perform extremely well, see Figure 4.12.

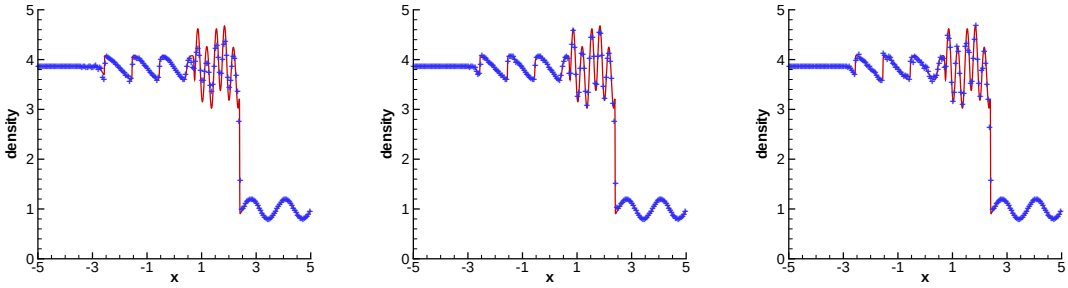
Example 4.4.9. We consider the interaction of blast waves of the Euler equation (4.32) with the initial condition

$$\boldsymbol{u}(x,0) = \begin{cases} \boldsymbol{u}_L, & 0 \leq x < 0.1, \\ \boldsymbol{u}_M, & 0.1 \leq x < 0.9, \\ \boldsymbol{u}_R, & 0.9 \leq x < 1, \end{cases} \tag{4.38}$$



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

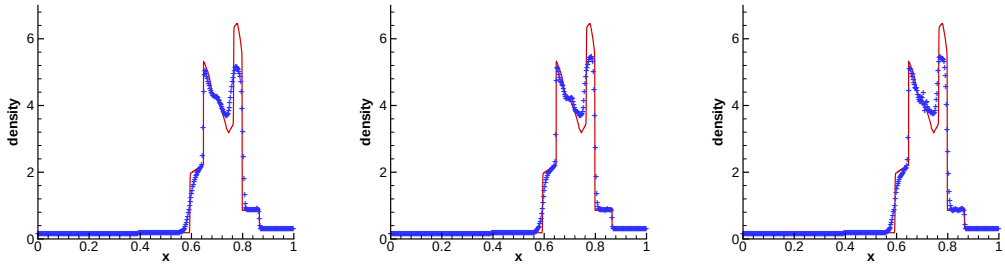
Figure 4.11: The shock density wave interaction problem. $t = 1.8$. $M = 0.01$. $N = 200$.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

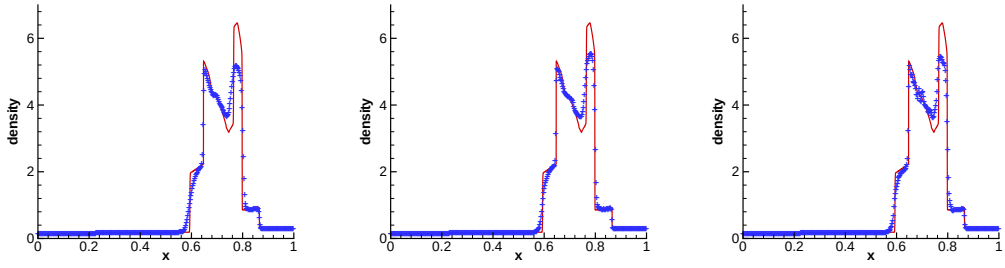
Figure 4.12: The shock density wave interaction problem. $t = 1.8$. $M = 300$. $N = 200$.

where $\rho_L = \rho_M = \rho_R = 1$, $v_L = v_M = v_R = 0$, $p_L = 10^3$, $p_M = 10^{-2}$, $p_R = 10^2$. A reflecting boundary condition is applied to both ends. See [67, 34]. The computed density ρ is plotted at $t = 0.038$ against the reference "exact" solution, which is a converged solution computed by the fifth order finite difference WENO scheme [38] with 16000 grid points. The results are in Figures 4.13-4.14. For the P^3 case, we add the positivity-preserving limiter [66] to avoid negative density or negative pressure during the time evolution. We can see that the pictures are satisfactory, except for the smearing of contact discontinuities, which seems more serious for this problem. For this problem, we have not noticed any significant difference for taking M from 0.01 to 300.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.13: The blast wave problem. $t = 0.038$. $M = 0.01$. $N = 400$.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.14: The blast wave problem. $t = 0.038$. $M = 200$. $N = 400$.

4.4.3 Euler System in Two Dimension

The 2D Euler system is given by

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x + \boldsymbol{g}(\boldsymbol{u})_y = 0,$$

$$\boldsymbol{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{u}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \boldsymbol{g}(\boldsymbol{u}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}. \tag{4.39}$$

Here ρ is the density, (u, v) is the velocity, E is the total energy, and p is the pressure, with

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right); \tag{4.40}$$

$\gamma = 1.4$ is used in the computation.

Example 4.4.10. This example is to test the order of accuracy for RKDG methods with our WENO limiter. The initial condition is set to be $\rho(x, y, 0) = 1 + 0.2\sin(x + y)$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$, $p(x, y, 0) = 1$ and the boundary conditions are periodic. The exactly solution is $\rho(x, y, t) = 1 + 0.2\sin(x + y - t)$, $u(x, y, t) = 0.7$, $v(x, y, t) = 0.3$ and $p(x, y, t) = 1$. We collect the L_1 and L_∞ errors at $t = 2\pi$ in Tables 4.9 and 4.10. Again, we can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the error of the original RKDG method.

Example 4.4.11. We consider the two-dimensional vortex evolution problem [35, 59], which is an idealized problem for the 2D Euler equations (4.39). The set up of this problem is as follows: The mean flow is $\rho = 1$, $p = 1$, and $(u, v) = (1, 1)$

Table 4.9: 2D Euler equation with initial condition $\rho(x,y,0) = 1 + 0.2\sin(x+y)$, $u(x,y,0) = 0.7$, $v(x,y0) = 0.3$, $p(x,y,0) = 1$ at $t = 2\pi$. Uniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	2.67E-03		6.33E-03	2.68	8.31E-03		2.36E-02	
	40	3.35E-04	3.00	1.84E-03	1.78	7.63E-04	3.45	3.92E-03	2.59
	80	5.92E-05	2.50	5.38E-04	1.77	6.20E-05	3.62	5.97E-04	2.72
	160	1.40E-05	2.08	1.44E-04	1.90	1.64E-05	1.92	1.54E-04	1.95
P^2	20	8.92E-05		7.41E-04		9.73E-05		7.42E-04	
	40	1.08E-05	3.04	1.06E-04	2.81	1.14E-05	3.10	1.06E-04	2.81
	80	1.29E-06	3.08	1.39E-05	2.93	1.33E-06	3.09	1.39E-05	2.93
	160	1.56E-07	3.05	1.76E-06	2.98	1.60E-07	3.06	1.76E-06	2.98

Table 4.10: 2D Euler equation with initial condition $\rho(x,y,0) = 1 + 0.2\sin(x+y)$, $u(x,y,0) = 0.7$, $v(x,y0) = 0.3$, $p(x,y,0) = 1$ at $t = 2\pi$. Nonuniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	2.76E-03		7.20E-03		8.14E-03		2.62E-02	
	40	3.52E-04	2.97	2.70E-03	1.41	8.28E-04	3.30	4.51E-03	2.54
	80	6.21E-05	2.50	6.87E-04	1.98	7.01E-05	3.56	7.60E-04	2.57
	160	1.49E-05	2.06	2.09E-04	1.72	1.49E-05	2.23	2.12E-04	1.85
P^2	20	9.25E-05		8.32E-04		1.08E-04		8.53E-04	
	40	1.12E-05	3.05	1.27E-04	2.71	1.22E-05	3.14	1.27E-04	2.74
	80	1.31E-06	3.09	1.59E-05	3.01	1.40E-06	3.13	1.59E-05	3.01
	160	1.58E-007	3.05	2.05E-06	2.95	1.65E-07	3.09	2.05E-06	2.95

(diagonal flow). We add, to this mean flow, an isentropic vortex (perturbations in (u, v) and the temperature $T = \frac{p}{\rho}$, no perturbation in the entropy $S = \frac{p}{\rho^\gamma}$:

$$(\delta u, \delta v) = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(-\bar{y}, \bar{x}), \quad \delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad \delta S = 0, \tag{4.41}$$

where $(\bar{x}, \bar{y}) = (x - 5, y - 5), r^2 = \bar{x}^2 + \bar{y}^2$, and the vortex strength $\epsilon = 5$. Since the mean flow is in the diagonal direction, the vortex movement is not aligned with the mesh direction. The computational domain is taken as $[-5, 15] \times [-5, 15]$, extended periodically in both directions. It is clear that the exact solution of the Euler equation with the above initial and boundary conditions is just the passive convection of the vortex with the mean velocity. We compute the solution to $t = 2$ for the accuracy test. The accuracy results are shown in Tables 4.11 and 4.12. Again, we can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the error of the original RKDG method.

Table 4.11: 2D Euler System of Smooth Vortex Evolution at $t = 2$. Uniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	2.11E-03		1.63E-01		4.05E-03		3.33E-01	
	40	4.57E-04	2.20	4.04E-02	2.01	8.74E-04	2.21	7.52E-02	2.15
	80	8.88E-05	2.36	1.28E-02	1.66	1.37E-04	2.67	1.54E-02	2.29
	160	1.67E-05	2.41	2.90E-03	2.14	2.05E-05	2.74	3.24E-03	2.25
P^2	20	5.44E-04		1.01E-01		1.95E-03		1.18E-01	
	40	6.19E-05	3.14	9.99E-03	3.34	8.72E-05	4.48	9.87E-03	3.58
	80	7.70E-06	3.01	1.28E-03	2.97	1.24E-05	2.81	1.59E-03	2.63
	160	1.08E-06	2.84	1.78E-04	2.85	1.41E-06	3.14	2.21E-04	2.85

Example 4.4.12. We consider the double Mach reflection problem [67]. It contains strong shock waves and contact discontinuity which is a good example to test the numerical scheme to show the ability to capture strong shock wave and the resolution for small scale structure. The computational domain for this problem is chosen to be $[0, 4] \times [0, 1]$. The reflecting wall lies at the bottom, starting from $x = \frac{1}{6}$. Initially

Table 4.12: 2D Euler System of Smooth Vortex Evolution at $t = 2$. Nonuniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	2.15E-03		1.67E-01		4.16E-03		3.31E-01	
	40	4.81E-04	2.16	4.08E-02	2.04	9.29E-04	2.16	7.69E-02	2.10
	80	9.89E-05	2.28	1.21E-02	1.75	1.51E-04	2.62	1.80E-02	2.10
	160	1.91E-05	2.38	2.79E-03	2.12	2.34E-05	2.69	3.29E-03	2.45
P^2	20	5.68E-04		9.37E-02		2.37E-03		1.46E-01	
	40	6.67E-05	3.09	1.10E-02	3.09	1.03E-04	4.53	1.14E-02	3.68
	80	8.09E-06	3.04	1.13E-03	3.28	1.52E-05	2.75	1.63E-03	2.80
	160	1.11E-06	2.86	1.67E-04	2.77	1.62E-06	3.23	2.42E-04	2.76

a right-moving Mach 10 shock is positioned at $x = \frac{1}{6}$, $y = 0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact post-shock condition is imposed for the part from $x = 0$ to $x = \frac{1}{6}$, and a reflective boundary condition is used for the rest. At the top boundary, the flow values are set to describe the exact motion of a Mach 10 shock. We compute the solution up to $t = 0.2$. As in [67], only results in $[0, 3] \times [0, 1]$ are shown. Two different uniform meshes, with 480×120 and 960 cells, and three different values of the TVB constant, $M = 0.01$, $M = 100$ and $M = 200$, are used in the numerical experiments. The density is plotted in Figure 4.15 for $M = 0.01$ and in Figure 4.16 for $M = 100$. In all the plots, we use 32 contours equally distributed from $\rho = 1.35$ to 23. It is not easy to observe any significant difference among these results in the picture. However, if we show a "blown-up" portion around the double Mach region, as in Figures 4.17 and 4.18, it is clear that one observes an increased resolution with an increasing k on the same mesh. Also, the resolution is slightly better as M increases from $M = 0.01$ to $M = 200$; however, this difference is not significant. Finally, we list in Table 4.13 the percentage of troubled-cells among all the cells.

Table 4.13: Percentage of troubled cells in the double Mach problem

N	$k=1$			$k=2$		
	$M=0.01$	$M=100$	$M=200$	$M=0.01$	$M=100$	$M=200$
480×120	20.31	5.65	4.53	25.84	18.78	16.11
960×240	14.35	4.69	3.79	23.31	19.62	18.05

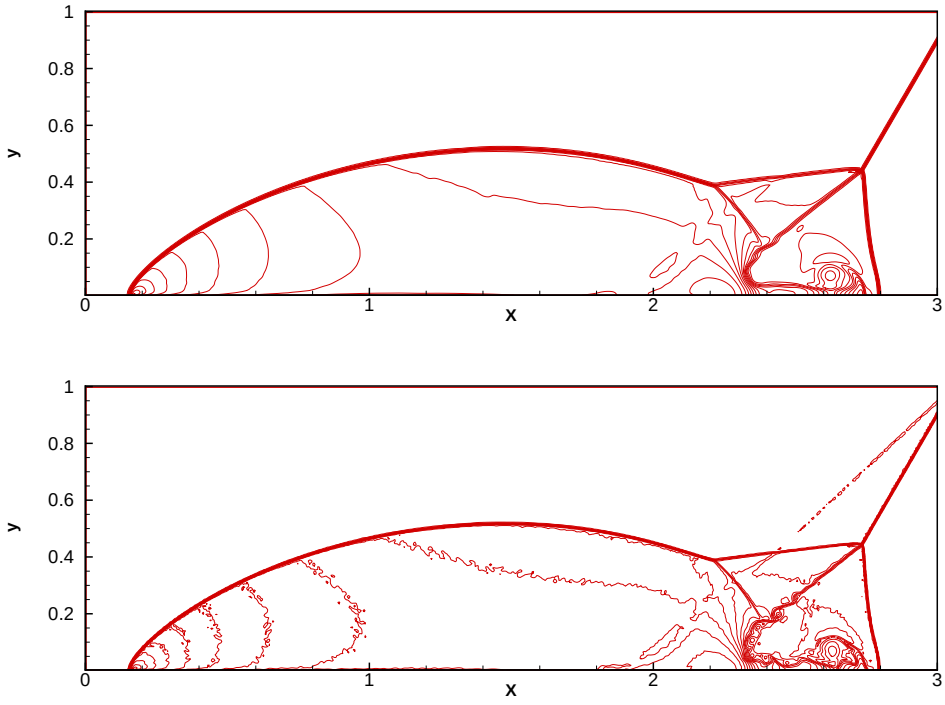


Figure 4.15: Double Mach reflection problem. $M = 0.01$. 960×240 cells. Thirty two equally spaced density contours from 1.35 to 23. Top: $k = 1$. Bottom: $k = 2$.

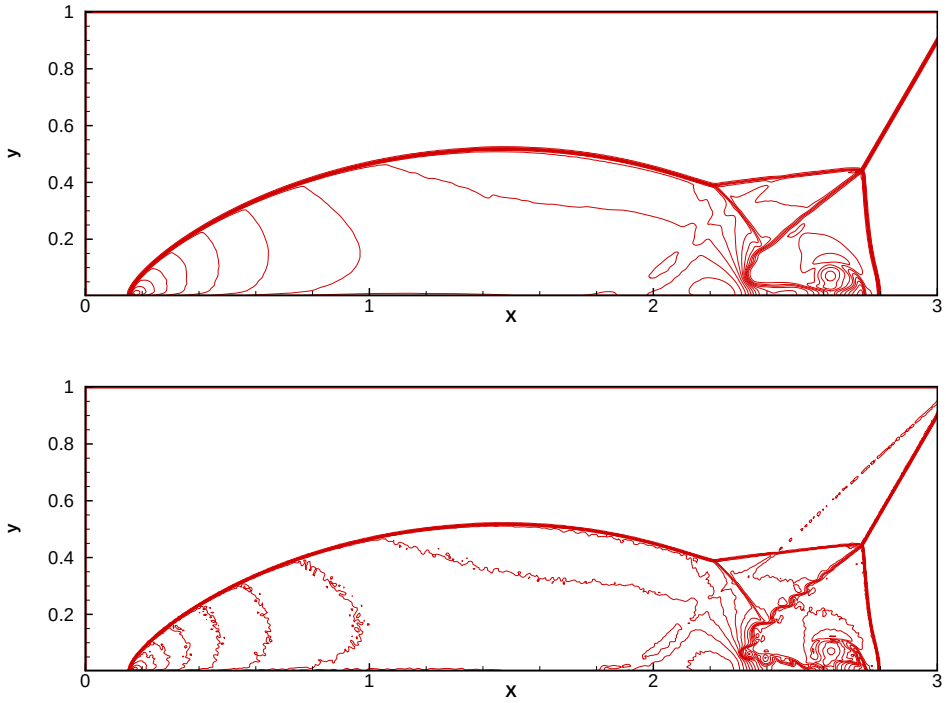


Figure 4.16: Double Mach reflection problem. $M = 100$. 960×240 cells. Thirty two equally spaced density contours from 1.35 to 23. Top: $k = 1$. Bottom: $k = 2$.

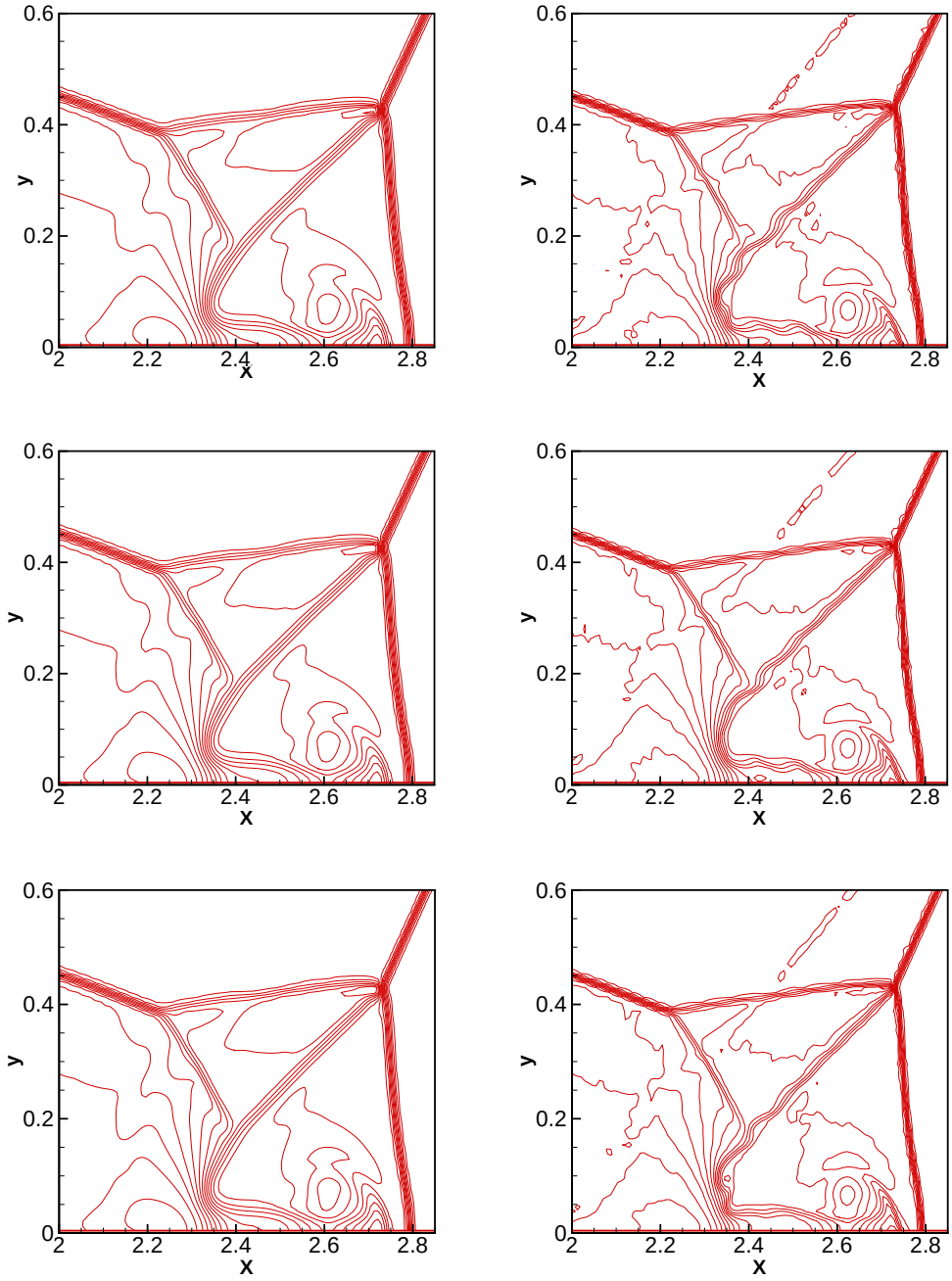


Figure 4.17: Double Mach reflection problem. 480×120 cells. $M = 0.01$ (top), $M = 100$ (middle) and $M = 200$ (bottom). Thirty two equally spaced density contours from 1.35 to 23. Left: $k = 1$. Right: $k = 2$.

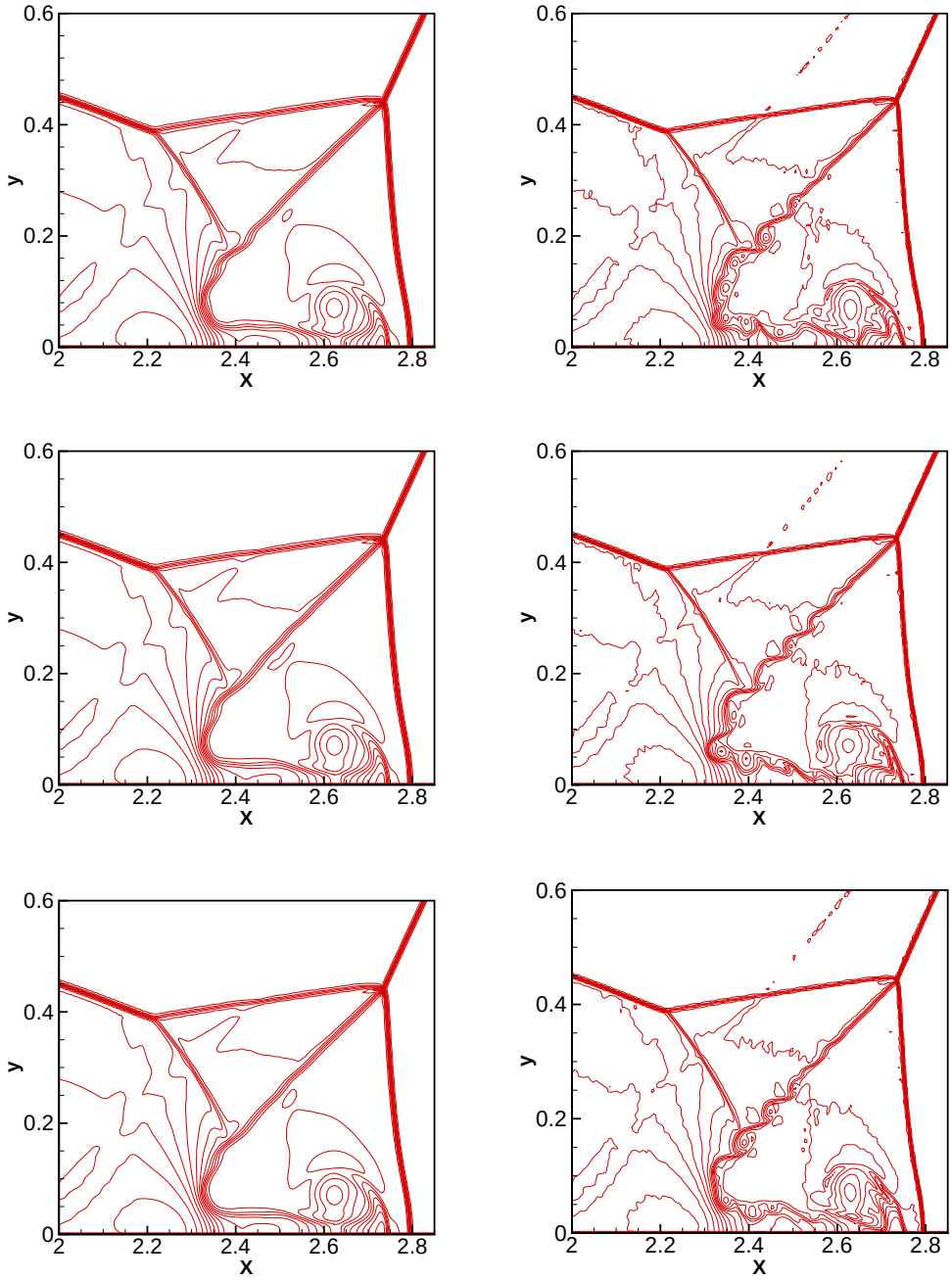


Figure 4.18: Double Mach reflection problem. 960×240 cells. $M = 0.01$ (top), $M = 100$ (middle) and $M = 200$ (bottom). Thirty two equally spaced density contours from 1.35 to 23. Left: $k = 1$. Right: $k = 2$.

Chapter 5

Conclusion

This dissertation is based on [26, 74, 75] and focuses on the numerical solutions of conservation laws using discontinuous Galerkin methods. It contains three parts.

In the first part, we have given an overview of DG methods for solving conservation laws. Since the basis functions can be completely discontinuous, these methods have flexibility which is not shared by typical finite element methods, such as the allowance of arbitrary triangulation with hanging nodes, complete freedom in changing the polynomial degrees in each element independent of that in the neighbor (p adaptivity), and extremely local data structure (elements only communicate with immediate neighbors regardless of the order of accuracy of the scheme) and the resulting embarrassingly high parallel efficiency.

In the second part, using Fourier analysis, we have developed quantitative error estimates for semi-discrete and fully discrete DG schemes with various Runge-Kutta methods, and investigated superconvergence properties at Radau points. Based on these quantitative errors, we have computed the necessary number of points per wavelength required to obtain a fixed error for several RKDG schemes. According to this analysis, assume that we know for a given problem the maximum wave number needed to adequately describe the solution, then we can choose the number of points such that the wave is well resolved. This can reduce the cost of computation while preserving the accuracy. If more detailed information is known about the spectral distribution of the Fourier coefficients, then this can be used to obtain sharper comparisons of efficiency by weighting the error function appropriately.

In the third part, we have designed a new and simpler limiter using WENO methodology for DG methods, with the goal of obtaining a robust limiting procedure to simultaneously obtain uniform high order accuracy and sharp, non-oscillatory shock transitions. The idea of this simple WENO limiter is that the reconstruction

polynomial on the troubled cell is a convex combination of the polynomials on this cell and its immediate neighboring cells, with the nonlinear weights of the linear combination following the classical WENO procedure. This limiting procedure can obtain both uniform high order accuracy and sharp, non-oscillatory shock transitions for the RKDG methods. The numerical performance is similar to previous WENO limiters which are much more complicated to implement. Improving the procedure for identifying troubled cells and implementing the limiter for more general meshes and three-dimensional problems constitute ongoing work.

Appendix A

A.1 Cardano's Method

This method[1] is to find the roots of a cubic function in the following form

$$x^3 + ax^2 + bx + c = 0. \tag{A.1}$$

The three roots are

$$x_1 = u_1 + u_2 - \frac{a}{3}, \quad x_2 = \omega u_1 + \bar{\omega} u_2 - \frac{a}{3}, \quad x_2 = \bar{\omega} u_1 + \omega u_2 - \frac{a}{3}$$

where

$$\omega = -\frac{1}{2} + \frac{\sqrt{3}}{2} i, \quad u_{1,2} = \sqrt[3]{-\frac{q}{2} \pm \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}}$$

with

$$p = b - \frac{a^2}{3}, \quad q = c + \frac{2a^3 - 9ab}{27a^3}.$$

[1] http://en.wikipedia.org/wiki/Cubic_equation

A.2 Ferrari's Method

This method[2] is to find the roots of a quartic function in the following form

$$x^4 + ax^3 + bx^2 + cx + d = 0. \tag{A.2}$$

Its solution can be found by means of the following calculations:

$$\alpha = -\frac{3a^2}{8} + b, \quad \beta = \frac{a^3}{8} - \frac{ac}{2} + c, \quad \gamma = -\frac{3a^4}{256} + \frac{a^2 b}{16} - \frac{ac}{4} + d.$$

If $\beta = 0$, then

$$x = -\frac{a}{4} \pm_s \sqrt{\frac{-\alpha \pm_t \sqrt{\alpha^2 - 4\gamma}}{2}} \qquad \text{(for } \beta = 0 \text{ only)}. \tag{A.3}$$

Otherwise, continue with

$$P = -\frac{\alpha^2}{12} - \gamma, \quad Q = -\frac{\alpha^3}{108} + \frac{\alpha\gamma}{3} - \frac{\beta^2}{8},$$

$$R = -\frac{Q}{2} \pm \sqrt{\frac{Q^2}{4} + \frac{P^3}{27}}, \quad \text{(either sign of the square root will do)}$$

$$U = \sqrt[3]{R}, \qquad \text{(there are 3 complex roots, any one of them will do)}$$

$$y = \begin{cases} -\frac{5}{6}\alpha + U - \frac{P}{3U} & \texttt{if } U \neq 0 \\ -\frac{5}{6}\alpha + U - \sqrt{3}Q & \texttt{if } U = 0 \end{cases},$$

$$W = \sqrt{\alpha + 2y}.$$

$$x = -\frac{a}{4} + \frac{\pm_s W \mp_t \sqrt{-\left(3\alpha + 2y \pm_s \frac{2\beta}{W}\right)}}{2}.$$

[2] http://en.wikipedia.org/wiki/Quartic_function

Appendix B

B.1 Lagrange Interpolation polynomial

This section[1] documents the interpolation polynomial in the Lagrange form.

B.1.1 Interpolation

Given a set of $(n+1)$ data points $\{(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)\}$, the points defined by $(x_i)_{0 \leq i \leq n}$ are called *points of interpolation*. The points $(y_i)_{0 \leq i \leq n}$ are the *values of interpolation*. To interpolate a function f, we define the values of interpolation as follows:

$$y_i = f(x_i), \quad \forall i = 0, \cdots, n.$$

[1] `http://www.math-linux.com/spip.php?article71`

B.2 Lagrange interpolation polynomial

The purpose here is to determine the unique polynomial of degree n, P_n which verifies

$$P_n(x_i) = f(x_i), \quad \forall i = 0, \ldots, n.$$

The polynomial which meets this equality is Lagrange interpolation polynomial

$$P_n(x) = \sum_{k=0}^{n} l_k(x) f(x_k)$$

where l_k are polynomials of degree n that form a basis of $\mathcal{P}_n$

$$l_k(x) = \prod_{i=0,\, i\neq k}^{n} \frac{x - x_i}{x_k - x_i} = \frac{x - x_0}{x_k - x_0} \cdots \frac{x - x_{k-1}}{x_k - x_{k-1}} \frac{x - x_{k+1}}{x_k - x_{k+1}} \cdots \frac{x - x_n}{x_k - x_n}$$

B.2.1 Properties of Lagrange interpolation polynomial and Lagrange basis

- They are the l_k polynomials which verify the following property:

$$l_k(x_i) = \delta_{ki} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}, \quad \forall i = 0, \ldots, n.$$

- They form a basis of the vector space $\mathcal{P}_n$ of polynomials of degree at most equal to n.

$$\sum_{k=0}^{n} \alpha_k l_k(x) = 0.$$

By setting: $x = x_i$, we obtain:

$$\sum_{k=0}^{n} \alpha_k l_k(x_i) = \sum_{k=0}^{n} \alpha_k \delta_{ki} = 0 \Longrightarrow \alpha_i = 0.$$

The set $(l_k)_{0 \leq k \leq n}$ is linearly independent and consists of $n+1$ vectors. It is thus a basis of $\mathcal{P}_n$.

- Finally, we can easily see that:

$$P_n(x_i) = \sum_{k=0}^{n} l_k(x_i) f(x_k) = \sum_{k=0}^{n} \delta_{ki} f(x_k) = f(x_i).$$

B.2.2 Existence and uniqueness of Lagrange interpolation polynomials

Existence. The proof is shown above. Actually, it corresponds to the construction of Lagrange interpolation polynomial with regard to the basis $(l_k)_{0 \leq k \leq n}$ of $\mathcal{P}_n$.
Uniqueness. Consider two elements P_n and Q_n of $\mathcal{P}_n$ which verify

$$P_n(x_i) = Q_n(x_i) = f(x_i), \quad \forall i = 0, \ldots, n.$$

Let $R_n = (P_n - Q_n) \in \mathcal{P}_n$. The polynomial R_n has $(n+1)$ roots which are exactly the $(x_i)_{0 \leq i \leq n}$ since

$$R_n(x_i) = P_n(x_i) - Q_n(x_i) = f(x_i) - f(x_i) = 0, \quad \forall i = 0, \ldots, n.$$

R_n has then $(n+1)$ roots and $R_n \in \mathcal{P}_n$. Therefore,

$$R_n = 0 \Longrightarrow P_n = Q_n.$$

We have thus shown the existence and uniqueness of Lagrange interpolation polynomial.

B.2.3 Error in Lagrange interpolation

Assume that $f \in \mathcal{C}^{n+1}([a,b])$ and $x \in [a,b]$. Let I be the closed set defined by $I = [\min(x, x_0), \max(x, x_n)]$ (the smallest closed set containing x and $x_i,\ \forall\ i$).

Theorem B.2.1.

$$\forall x \in [a,b], \quad f(x) - P_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i),$$

where

$$\min\{x_0, x_1, \cdots, x_n, x\} \leq \xi = \xi(x) \leq \max\{x_0, x_1, \cdots, x_n, x\}.$$

Proof: There are two possible ways to prove this theorem: 1) the one encountered in the polynomial interpolation in the form of Newton and, 2) the following proof. If $x = x_i$, the problem is over:

$$f(x) - P_n(x) = f(x_i) - P_n(x_i) = 0.$$

This means $f(x) - P_n(x)$ has $n+1$ roots. Therefore, we can rewrite it in the following form:

$$f(x) - P_n(x) = \Phi(x) \prod_{i=0}^{n} (x - x_i).$$

We also define the function g:

$$g(x,t) = f(t) - P_n(t) - \prod_{i=0}^{n} (t - x_i)\Phi(x).$$

$g(x,\cdot)$ is $(n+1)$ times differentiable and evaluates to zero at the $(n+2)$ points $x_0, x_1, \ldots, x_n, x$ of the interval I. By successively applying Rolles theorem, $g^{(n+1)}(x,\cdot)$ evaluates to zero at a point $\xi \in I$:

$$g^{(n+1)}(x,\xi) = 0.$$

The $(n+1)^{th}$ derivative of $g(x,\cdot)$ can be easily calculated:

$$g^{(n+1)}(x,t) = f^{(n+1)}(t) - (n+1)!\Phi(x).$$

By setting $t = \xi$, we have:

$$g^{(n+1)}(x,\xi) = f^{(n+1)}(\xi) - (n+1)!\Phi(x) = 0.$$

Therefore

$$\Phi(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

We finally conclude that

$$f(x) - P_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i).$$

Since ξ is dependent on $x, x_0, \cdots, x_n$, ξ is a function of x, i.e. $\xi = \xi(x)$. Furthermore, by Rolle's theorem, we have

$$\min\{x_0, x_1, \cdots, x_n, x\} \leq \xi = \xi(x) \leq \max\{x_0, x_1, \cdots, x_n, x\}.$$

Corollary B.2.1. Assume that $f \in \mathcal{C}^{n+1}([a,b])$ and $x \in [a,b]$.

$$\forall x \in [a,b], \quad |f(x) - P_n(x)| \leq \frac{\left|\prod_{i=0}^{n}(x - x_i)\right|}{(n+1)!} \sup_{x \in [a,b]} |f^{n+1}(x)|.$$

Alternatively stated:

$$\forall x \in [a,b], \quad |f(x) - P_n(x)| \leq \frac{(b-a)^{n+1}}{(n+1)!} \sup_{x \in [a,b]} |f^{n+1}(x)|.$$

Bibliography

[1] S. Adjerid, M. Aiffa and J. Flaherty. High-order finite element methods for singularly perturbed elliptic and parabolic problems. *SIAM Journal on Applied Mathematics*, 55:520–543, 1995.

[2] S. Adjerid, K. Devine, J. Flaherty and L. Krivodonova. A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 191:1097–1112, 2002.

[3] S. Adjerid and A. Klauser. Superconvergence of discontinuous finite element solutions for transient convection-diffusion problems. *Journal of Scientific Computing*, 22–23:5–24, 2005.

[4] S. Adjerid and T.C. Massey. A posteriori discontinuous finite element error estimation for two-dimensional hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 191:5877–5897, 2002.

[5] S. Adjerid and T.C. Massey. Superconvergence of discontinuous Galerkin solutions for a nonlinear scalar hyperbolic problem. *Computer Methods in Applied Mechanics and Engineering*, 195:3331–3346, 2006.

[6] S. Adjerid and T. Weinhart. Discontinuous Galerkin error estimation for linear symmetrizable hyperbolic systems. *Mathematics of Computation*, 80:1335–1367, 2011.

[7] S. Adjerid and T. Weinhart. Discontinuous Galerkin error estimation for linear symmetric hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 198:3113–3129, 2009.

[8] M. Ainsworth. Dispersive and dissipative behavior of high order discontinuous Galerkin finite element methods. *Journal of Computational Physics*, 198:106–130, 2004.

[9] M. Ainsworth, P. Monk and W. Muniz. Dispersive and dissipative properties of discontinuous Galerkin finite element methods for the second-order wave equation. *Journal of Scientific Computing*, 27:5–40, 2006.

[10] M. Bakker. A note on C^0 Galerkin methods for two-point boundary problems. *Numerische Mathematik*, 38:447–452, 1982.

[11] M. Bakker. One-dimensional Galerkin methods and superconvergence at interior nodal points. *SIAM Journal on Numerical Analysis*, 21:101–110, 1984.

[12] D.S. Balsara and C.-W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemeswith increasingly high order of accuracy. *Journal of Computational Physics*, 160:405–452, 2000.

[13] R. Biswas, K. Devine and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–283, 1994.

[14] A. Burbeau, P. Sagaut and Ch.H. Bruneau. A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods. *Journal of Computational Physics*, 169:111–150, 2001.

[15] P. Castillo. A superconvergence result for discontinuous Galerkin methods applied to elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 192:4675–4685, 2003.

[16] F. Celiker and B. Cockburn. Superconvergence of the numerical traces for discontinuous Galerkin and hybridized methods for convection-diffusion problems in one space dimension. *Mathematics of Computation*, 76:67–96, 2007.

[17] Y. Cheng and C.-W. Shu. Superconvergence and time evolution of discontinuous Galerkin finite element solutions. *Journal of Computational Physics*, 227:9612–9627, 2008.

[18] Y. Cheng and C.-W. Shu. Superconvergence of local discontinuous Galerkin methods for one-dimensional convection-diffusion equations. *Computers and Structures*, 87:630–641, 2009.

[19] Y. Cheng and C.-W. Shu. Superconvergence of discontinuous Galerkin and local discontinuous Galerkin schemes for linear hyperbolic and convection-diffusion equations in one space dimension. *SIAM Journal on Numerical Analysis*, 47:4044–4072, 2010.

[20] B. Cockburn, S. Hou and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.

[21] B. Cockburn, S.-Y. Lin and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.

[22] B. Cockburn, M. Luskin, C.-W. Shu and E. Süli. Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. *Mathematics of Computation*, 72:577–606, 2003.

[23] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.

[24] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection P^1 discontinuous Galerkin method for scalar conservation laws. *Mathematical Modelling and Numerical Analysis*, 25:337–361, 1991.

[25] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.

[26] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16:173–261, 2001.

[27] J. Douglas and T. Dupont. Some superconvergence results for Galerkin methods for the approximate solution of two-point boundary value problems. In J.J. Miller, editor, *Topics in Numerical Analysis II*, Academic Press, 1973, 89–93.

[28] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693–723, 2007.

[29] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144:194–212, 1998.

[30] S. Gottlieb, D.I. Ketcheson and C.-W. Shu. High order strong stability preserving time discretizations. *Journal of Scientific Computing*, 38:251–289, 2009.

[31] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation*, 67:73–85, 1998.

[32] S. Gottlieb, C.-W. Shu and E. Tadmor. Strong stability preserving high order time discretization methods. *SIAM Review*, 43:89–112, 2001.

[33] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49:257–393, 1983.

[34] A. Harten, B. Engquist, S. Osher and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[35] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150:97–127, 1999.

[36] F.Q. Hu and H.L. Atkins. Eigensolution analysis of the discontinuous Galerkin method with nonuniform grids: I. one space dimension. *Journal of Computational Physics*, 182:516–545, 2002.

[37] F.Q. Hu and H.L. Atkins. Two-dimensional wave analysis of the discontinuous Galerkin method with non-uniform grids and boundaryconditions. In *Proceedings of the Eighth AIAA/CEAS Aeronautics Conference and Exhibit, Breckenridge, Colorado*. 2012. AIAA paper no. 2002-2514.

[38] G. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1995.

[39] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 46:1–26, 1986.

[40] O. Karakashian and C. Makridakis. A space-time finite element method for the nonlinear Schrödinger equation: the discontinuous Galerkin method. *Mathematics of Computation*, 67:479–499, 1998.

[41] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205:486–508, 2005.

[42] H.-O. Kreiss and J. Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24:199–215, 1972.

[43] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7:159–193, 1954.

[44] P. Lesaint and P.A. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, Academic Press, 1974, 89–145.

[45] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhauser Verlag, Basel, 1990.

[46] D. Levy, G. Puppo and G. Russo. Central WENO Schemes for hyperbolic systems of conservation laws. *Mathematical Modelling and Numerical Analysis (M^2AN)*, 33:547–571, 2001.

[47] X. Liu, S. Osher and T. Chen. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115:200–212, 1994.

[48] J. Qiu and C.-W. Shu. On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes. *Journal of Computational Physics*, 183:187–209, 2002.

[49] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, 193:115–135, 2003.

[50] J. Qiu and C.-W. Shu. A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters. *SIAM Journal on Scientific Computing*, 27:995–1013, 2005.

[51] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two dimensional case. *Computers and Fluids*, 34:642–663, 2005.

[52] J. Qiu and C.-W. Shu. Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing*, 26:907–929, 2005.

[53] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, 1973.

[54] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 135:250–258, 1997.

[55] S.J. Sherwin. Dispersion analysis of the continuous and discontinuous Galerkin formulations. In B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin methods: Theory, Computational and Applications*, Springer, 2000, 425–431.

[56] J. Shi, C. Hu and C.-W. Shu. A technique of treating negative weights in WENO schemes. *Journal of Computational Physics*, 175:108–127, 2002.

[57] C.-W. Shu. TVB uniformly high-order schemes for conservation laws. *Mathematics of Computation*, 49:105–121, 1987.

[58] C.-W. Shu. Total-Variation-Diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing*, 9:1073–1084, 1988.

[59] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu

and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, volume 1697, Springer, 1998, 325-432.

[60] C.-W. Shu. A survey of strong stability preserving high order time discretization methods. In D. Estep and S. Tavener, editors, *Collected Lectures on the Preservation of Stability under Discretization*, SIAM: Philadelphia, PA, 2002, 51–65.

[61] C.-W. Shu. High order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Review*, 51:82–126, 2009.

[62] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–471, 1988.

[63] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83:32–78, 1989.

[64] G. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1978.

[65] B. Swartz and B. Wendroff. The relative efficiency of finite difference and finite element methods. *SIAM Journal on Numerical Analysis*, 11:979–993, 1974.

[66] C. Wang, X. Zhang, C.-W. Shu and J. Ning. Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations. *Journal of Computational Physics*, 231:653–665, 2012.

[67] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:115 – 173, 1984.

[68] M. Zhang and C.-W. Shu. An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations. *Mathematical Models and Methods in Applied Sciences (M^3AS)*, 13:395–413, 2003.

[69] M. Zhang and C.-W. Shu. An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods. *Computers and Fluids*, 34:581–592, 2005.

[70] M. Zhang and C.-W. Shu. Fourier analysis for discontinuous Galerkin and related methods. *Chinese Science Bulletin*, 54:1809–1816, 2009.

[71] Q. Zhang and C.-W. Shu. Error estimates to smooth solutions of Runge–Kutta discontinuous Galerkin methods for scalar conservation laws. *SIAM Journal on Numerical Analysis*, 42:641–666, 2004.

[72] Q. Zhang and C.-W. Shu. Error estimates to smooth solutions of Runge-Kutta discontinuous Galerkin method for symmetrizable systems of conservation laws. *SIAM Journal on Numerical Analysis*, 44:1703–1720, 2006.

[73] Y.-T. Zhang and C.-W. Shu. Third order WENO scheme on three dimensional tetrahedral meshes. *Communications in Computational Physics*, 5:836–848, 2009.

[74] X.H. Zhong and C.-W. Shu. Numerical resolution of discontinuous Galerkin methods for time dependent wave equations. *Computer Methods in Applied Mechanics and Engineering*, 200:2814–2827, 2011.

[75] X.H. Zhong and C.-W. Shu. A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods. *Submitted to Journal of Computational Physics*.

[76] J. Zhu, J. Qiu, C.-W. Shu and M. Dumbser. Runge-Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes. *Journal of Computational Physics*, 227:4330–4353, 2008.