Positivity-Preserving High-Order Discontinuous Galerkin Methods: Implicit Time Stepping and Applications to Relativistic Hydrodynamics

by

Tong Qin

B.Sc., University of Science and Technology of China, Hefei, P. R. China, 2011

M.Sc., University of British Columbia, Vancouver, BC, 2013

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

PROVIDENCE, RHODE ISLAND

May 2017

This dissertation by Tong Qin is accepted in its present form
by the 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_____ _____

Jerome Darbon, Ph.D., Reader

Approved by the Graduate Council

Date_____ _____

Andrew G. Campbell, Dean of the Graduate School

Vita

Education

Brown University, Providence, RI, U.S.A.,
Ph. D. in Applied Mathematics, May, 2017 (expected),
Advisor: Prof. Chi-Wang Shu.

University of British Columbia (UBC), Vancouver, Canada,
M. Sc. in Applied Mathematics, Nov., 2013,
Advisor: Prof. Dominik Schötzau.

University of Science and Technology of China (USTC), Hefei, China,
B. Sc. in Mathematics and Applied Mathematics, Jul. 2011,
Advisor: Prof. Yan Xu.

Publications and Preprints

1. T. Qin and C.-W. Shu, *Implicit positivity preserving high order discontinuous Galerkin methods for conservation laws*, submitted.

2. T. Qin, C.-W. Shu and Y. Yang, *Bound-preserving discontinuous Galerkin methods for relativistic hydrodynamics*, Journal of Computational Physics, vol. 315, pp. 323-347, 2016.

3. R. Oyarzúa, T. Qin and D. Schötzau, *An exactly divergence-free finite element method for a generalized Boussinesq problem*, IMA Journal of Numerical Analysis, vol. 34 (3), pp. 1104-1135, 2014.

Honors and Awards

- 2010, Outstanding Undergraduate Student Research Project, USTC.
- 2007-2011, Outstanding Student Scholarship, USTC.
- 2010, Second prize in Statistical Modelling Competition, USTC.
- 2007, Second prize in National Olympia Physics Competition.

Teaching Experience

At Brown

- Spring 2015, Teaching Assistant,
  APMA 1200 Operational Analysis: Probabilistic Models.
- Fall 2014, Teaching Assistant,
  APMA 0360 Methods of Applied Mathematics.

At UBC

- Spring 2013/2012, Teaching Assistant,
  MATH 210 Introduction to Mathematical Computing.
- Fall 2012/2011, Teaching Assistant,
  MATH 255 Ordinary Differential Equations.
- 2011-2013, Graduate Tutor in the Math Learning Center.

Acknowledgments

First of all, I would like to express my deepest gratitude to my supervisor Prof. Chi-Wang Shu for his advice and encouragement when I got stuck and depressed, for his bluntly pointing out of my shortcomings, for his generous sharing his life stories and experience with me and for his thoughtful advice for both of my career and my family life. He not only showed me how to do high-standard and useful research but also taught me how to work efficiently and how to be a self-disciplined, responsible and integral person. The most valuable thing I learned from him is to be generous to others while being strict with ourselves. All in all, he has set me a life-long role model.

Secondly, I would like to thank Prof. Johnny Guzmán and Prof. Jerome Darbon who have graciously agreed to spend their time in reading this dissertation and providing invaluable feedbacks. I also owe my thanks to all the other professors who have once shared their wisdoms with me, either in class or in research projects, including Prof. Mark Ainsworth, Prof. Yan Guo, Prof. Walter Strauss, Prof. Hongjie Dong, Prof. Paul Dupuis at Brown as well as Prof. Dominik Schötzau, Prof. Michael Ward and Prof. Uri Ascher at UBC, Prof. Zhiming Chen at CAS and Prof. Yan Xu at USTC.

Thirdly, my thanks also go to Dr. Xiangxiong Zhang and Dr. Yang Yang. Dr. Xiangxiong Zhang's intellectual work in his thesis laid a foundation for this

dissertation. The second part of the dissertation is based on the work [55], in which Dr. Yang Yang has made a great contribution. Moreover, their stories at Brown constantly remind me how to be a hard-working and resilient student.

Next, all my friends at Brown have helped me a lot during my four years' graduate study. Among them I would thank Chengke Shi, for numerous enthusiastic discussions on math, life and many other topics. Thanks to Zheng Sun, for being a mirror reflecting many of my shortcomings. Thanks to Tianheng Chen, Guosheng Fu and Juntao Huang, for sharing their research with me and for their help with my life. Thanks to Weifeng Sun for being a considerate roommate and a reliable friend. Thanks to Bingyu Zhao and Kunrui Wang, for organizing our weekly academic family gathering. Thanks to my academic buddy Christian Glusa, for his advice on my preliminary exam and for many conversations on each other's research. Thanks to Dr. Jeffery Miller for kindly sharing this thesis template around. At last, thanks to Prof. Henry Majewski for hosting me and providing me with help during my very first year in Providence.

All these could not happen without the generous support from the Brown University and the Graduate School at Brown as well as the service of all the staff members of DAM at Brown.

Last but not the least, I will keep my special gratitude to my family. Thanks to my parents for their unconditional support and sacrifice. Thanks to my wife, my beautiful sunshine, Mrs. Yang Chen, for her constant support and tolerance.

Abstract of "Positivity-Preserving High-Order Discontinuous Galerkin Methods: Implicit Time Stepping and Applications to Relativistic Hydrodynamics", by Tong Qin, Ph.D., Brown University, May 2017

The positivity-preserving property is a highly desirable property when designing high order numerical methods for hyperbolic conservation laws, since negative values sometimes cause ill-posedness of the problem and blow-ups of the algorithms. The general framework for constructing positivity-preserving schemes for solving hyperbolic conservation laws have been proposed in (X. Zhang and C.-W. Shu, Journal of Computational Physics, 229 (2010), pp. 3091–3120) and (X. Zhang and C.-W. Shu, Journal of Computational Physics, 229 (2010), pp. 8918–8934). In this dissertation, we extend this framework to DG methods with implicit discretizations and to DG methods for solving the relativistic hydrodynamics (RHD).

Due to the the Courant-Friedrichs-Levis (CFL) number constraint, explicit DG methods are impractical for problems involving unstructured and extremely varying meshes or long-time simulations. Instead, implicit DG schemes are often popular in practice, especially in the computational fluid dynamics (CFD) community. In the first part of this dissertation, we develop a high-order positivity-preserving DG method with the backward Euler time discretization for solving conservation laws, basing on a generalization of the Zhang-Shu positivity-preserving limiter. Both the analysis and numerical experiments indicate that a lower bound for the CFL number is required to obtain the positivity-preserving property for the numerical schemes. The proposed method not only preserves the positivity of the numerical approximation without compromising the designed high-order accuracy, but also helps accelerate the convergence towards the steady-state solution and add robustness to the nonlinear solver.

For RHD systems, the density and the pressure are positive physical quantities

and the velocity is bounded by the speed of light. The violation of these bounds will result in ill-posedness of the problem and blow-up of the code, especially in extreme relativistic cases. It is usually hard to maintain these physical bounds without sacrificing the numerical accuracy. In the second part, we develop a bound-preserving DG method to solve RHD systems by extending the bound-preserving limiter for the non-relativistic hydrodynamics. The proposed method has the following features. It can theoretically guarantee to preserve the physical bounds for the numerical approximation and maintain its designed high order accuracy. Moreover, it renders L^1-stability to the numerical scheme. The robustness of the scheme is tested on various extreme relativistic cases, including relativistic jets.

Contents

Vita iv

Acknowledgments vi

1 Introduction 1

2 DG Methods and Framework for Bound-Preserving Techniques 5

2.1 DG methods for conservation laws 6

2.1.1 Spacial discretizations 7

2.1.2 Time discretizations 7

2.2 General framework for positivity and bound preserving techniques . . 9

3 Positivity-Preserving DG Methods with Implicit Time Stepping for Conservation Laws 12

3.1 Introduction 13

3.2 Positivity-preserving backward Euler DG scheme for scalar equations 16

3.2.1 Preliminaries 17

3.2.2 CFL condition for P^1-DG methods on uniform meshes 20

3.2.3 CFL condition for arbitrary polynomial degree on general meshes 26

3.2.4 Scaling limiter 41

3.2.5 Algorithm for scalar equations 42

3.3 Positivity-preserving backward Euler DG scheme for compressible Euler systems . . . 43
3.4 Positivity-preserving Crank-Nicolson DG scheme . . . 46
3.4.1 CFL condition for P^1-DG on uniform meshes . . . 47
3.5 Numerical experiments . . . 57
3.5.1 Accuracy tests . . . 57
3.5.2 Moving shocks . . . 63
3.5.3 Compressible Euler system . . . 66
3.6 Concluding remarks . . . 69

4 Positivity and Bound-Preserving DG Methods for Relativistic Hydrodynamics 71
4.1 Introduction . . . 72
4.2 Numerical algorithm in one space dimension . . . 78
4.2.1 The DG scheme . . . 78
4.2.2 Bound-preserving technique . . . 79
4.2.3 High order time discretizations . . . 90
4.3 Numerical algorithm in two space dimensions . . . 91
4.4 Application to relativistic jets . . . 98
4.5 Numerical experiments . . . 102
4.5.1 One-dimensional experiments . . . 103
4.5.2 Two-dimensional experiments . . . 111
4.6 Concluding remarks . . . 119

5 Conclusion 120

List of Tables

3.1 Values of r_k for $k = 1, \cdots, 5$ and for Legendre-Gauss-Lobatto (LGL) and Legendre-Gauss quadrature rules respectively. 40

3.2 Minimum cell average after one time step for odd k. 40

3.3 Minimum cell average after one time step for even k. 41

3.4 Values of $\min_j \bar{u}_j^1$ with initial condition (3.45) for $N = 4, 8, 16, 32$ and different λ. 52

3.5 Values of $\min_j \bar{u}_j^1$ with initial condition (3.46) for $N = 4, 8, 16, 32$ and different λ. 52

3.6 Error table for Example 3.5.1.1 at $T = 1$ with $\lambda = 0.81$, uniform mesh. 58

3.7 Error table for Example 3.5.1.2, approximation of the steady state solution to the linear problem (3.48), without the positivity preserving limiter. 59

3.8 Error table for Example 3.5.1.2, approximation of the steady state solution to the linear problem (3.48) with the positivity preserving limiter. 60

3.9 Error table for Example 3.5.1.3, approximation of the steady state solution to the Burgers equation (3.49), without the positivity preserving limiter. 61

3.10 Error table for Example 3.5.1.3, approximation of the steady state solution to the Burgers equation (3.49), with the positivity preserving limiter. 61

3.11 Error table for Example 3.5.1.3, approximation of the steady-state solution to the Burgers equation (3.50), with and without the positivity preserving limiter. 62

4.1 Example 4.5.1.1: One-dimensional accuracy test at $T = 0.4$ for the second-, third- and fourth-order DG methods with and without the BP limiters. The CFL number for the multistep method is one-third of that for the RK method. k is the degree of polynomial and h is the meshsize. 104

4.2 Example 4.5.1.1: Two-dimensional accuracy test at $T = 0.2$ for the third-order RKDG method with the BP limiter. k is the degree of polynomial and h is the mesh size. 114

List of Figures

3.1 Example 3.5.2.1: at $T = 0.2$, with $\Delta t = 0.266 \max_j h_j$ and $N = 120$. Solid line is the exact solution. Blue squares are point values on the Legendre-Gauss-Lobatto points. 64

3.2 Example 3.5.2.2 at $T = 1.5$. Solid line is the exact solution. Blue squares are point values of the numerical approximation on the Legendre-Gauss-Lobatto points. Mesh with $N = 120$ cells and CFL $= 2$. 65

3.3 Example 3.5.2.3 at $T = 0.6$. Solid line is the exact solution. Blue squares are point values of the numerical approximation on the Legendre-Gauss-Lobatto points. Mesh with $N = 120$ cells and CFL $= 3.5$. 66

3.4 Example 3.5.3.1 at $T = 0.01$ with $N = 200$ and CFL $= 2$. With the positivity-preserving limiter on. Solid line is the exact solution and blue squares are numerical approximations. 67

3.5 Example 3.5.3.2 at $T = 0.1$, with $N = 200$ and CFL $= 2$. With the positivity-preserving limiter on. Solid line is the exact solution and blue squares are the numerical approximations. 68

3.6 Example 3.5.3.3 at $T = 0.003$, with $N = 200$ and CFL $= 2$. With the positivity-preserving limiter on. Solid line is the exact solution and blue squares are the numerical approximations. 69

4.1 Plots for the lower bound of α in (4.19) for the bound-preserving requirement (denoted by α_{bp}) and the spectral radius (4.28) of the Jacobian matrix (denoted by $\alpha_{standard}$). The left panel is to keep $\gamma = 20$ constant and the right one is to keep $h = 20$ constant. 85

4.2 Example 4.5.1.1: Log-log plot of the CPU time against the L^2 error for polynomial degrees $k = 1, 2, 3$ with and without the BP limiter. . 105

4.3 Example 4.5.1.2: Blast wave with initial condition (4.51) at $T = 0.5$ on the mesh of 200 cells, approximated by the third-order RKDG method with the BP limiter. The squares represent the approximate cell averages and the solid line is the exact solution. 106

4.4 Example 4.5.1.3: Strong blast wave with initial condition (4.52) at $T = 0.4$ approximated by the third-order RKDG method with the BP limiter on the mesh with 200 cells. The square represents the approximate cell averages and the solid line is the exact solution. . . . 107

4.5 Example 4.5.1.3: Strong blast wave with initial condition (4.53) at $T = 0.3$ approximated by the third-order RKDG method with the BP limiter. The mesh is decomposed into 400 cells (left column) and 800 cells (right column). The square represents the approximate cell averages and the solid line is the exact solution. 108

4.6 Example 4.5.1.4: One-dimensional shock-heating problem on the mesh of 200 cells, at $T = 1.5$ with $v_{in} = 0.999999$, approximated by the third-order RKDG method with the BP limiter. The squares represent the numerical results and the solid line is the exact solution. 109

4.7 Example 4.5.1.5: One-dimensional Riemann problem with non-zero transverse velocity at $T = 0.6$, approximated by the third-order RKDG method with the BP limiter. Approximations of the density and the transverse velocity on meshes of 400 and 6400 cells. The squares represent the approximate cell averages and the solid lines are the exact solutions. 111

4.8 Example 4.5.1.5: One-dimensional Riemann problem with non-zero transverse velocity at $T = 0.6$, approximated by the third-order RKDG method with the BP limiter. The approximations of w and p on meshes of 400 and 6400 cells are presented respectively. The squares represent the approximate cell averages and the solid lines are the exact solutions. 112

4.9 Example 4.5.1.5: One-dimensional Riemann problem with non-zero transverse velocity $v = 0.999$ at $T = 0.6$, approximated by the third-order RKDG method with the BP limiter. The approximations of the w, v, p and ρ on 3200 cells are presented. The blue solid lines are the numerical results and the black ones are the exact solutions. 113

4.10 Example 4.5.2.2: Blast wave with initial condition (4.56) propagating in $[0, \frac{\sqrt{2}}{2}]^2$. At time $T = 0.3$ on a 120×120 mesh. In the right column, solid lines are exact solution and the blue squares are the approximate ones. 115

4.11 Example 4.5.2.3: 400×400 cells at $T = 0.7$ approximated by the third-order RKDG method with the BP limiter. Thirty equally spaced contours of the logarithm of the proper density are plotted. 116

4.12 Example 4.57: Relativistic jets of model C2 with initial condition (4.57). The left panel is at $T = 20$ and the right is at $T = 35$. The resolution is 10 points per jet radius. 117

4.13 Example 4.5.2.4: Relativistic jet of the model C3 with initial condition (4.58) at $T = 30$, approximated by RKDG with the BP limiter. The resolution is 10 points per jet radius. 118

Chapter One

Introduction

The discontinuous Galerkin (DG) methods were first introduced by Reed and Hill [59] for solving neutron transport equations on triangular meshes. Afterwards, Cockburn et al. developed the Runge-Kutta DG (RKDG) methods in a sequence of papers [12, 11, 10, 13] for solving the hyperbolic conservation laws. After that, DG methods have seen great success in many areas of applications, including the computational fuild dynamics (CFD), semi-conductor device design, computational comsmology and so on. The DG method features mathematically provable high-order accuracy and stability, extremely local data structure, great parallel efficiency and flexibility for dealing with unstructured meshes and h-p adaptivity. One fundamental challenge in desgining high-order DG methods for hyperblic conservation laws is to maintain certain stability without compromising the designed high-order accuracy. This is usually done by limiting the numerical approximation or by modifying the schemes to let the numerical solution respect certain properties of the physical solution, such as the total variation diminishing or bounded (TVD/TVB) properties [12], entropy inequalities [8], maximum principle and so on. One of the recent great achievements in overcoming this difficulty is the genuinely high-order maximum-principle satisfying and positivity preserving DG methods developed by Zhang and Shu in [86] and [87]. In this dissertation, we generalize this work in the following two directions.

1. Generalize the positivity-preserving limiter for the explicit methods to the DG methods with backward Euler time discretizations [54].

2. Generalize the bound-preserving limiter for non-relativistic hydoynmaics to the relativistic one [55].

The first topic is motivated by the practical application of DG methods in the computational fluid dynmaics (CFD) community. When solving the aerospace problems, unstructured and extremely varying meshes are always adopted to resolve

complex geometries or boundary layers and long-time simulations are always needed for the steady-state simulation. The CFL constraint for explicit time discretizations make explicit methods, such as RKDG methods, impractical in these real applictions. To circumvent this difficulty, implicit methods, which have much larger stability regions, are often popular in practice. In particular, for DG methods, in [29] Jiang and Shu have shown by the cell entropy inequality that a class of implicit time dicretizations, including the backward Euler ant the Crank-Nicolson methods, are unconditionally L^2 stable. However, only a few works can be found in the literature dealing with the positivity-preserving property for high-order DG methods coupled with implicit time discretizations. In the first part of this dissertation, we introduce how to construct positivity-preserving DG methods for conservation laws with arbitrary order of spacial accuracy. Since our focus is on the positivity-preserving property of the spacial discretization, the time is discretized with the simple backward Euler methods. The Crank-Nicolson time discretization is also briefly considered for linear element on uniform meshes. High-order implicit time discretization will be explored in future work. We derive a CFL condition for the scalar linear equation under which the cell average is promised to be positive in each time step. Then the positivity of the whole polynomial, at least values at quadrature points, can be obtained by the standard scaling limiter [33, 86].

The second topic is to generlize the bound-preserving limiter for the compressible Euler system with and without source terms in [87, 89] to the RHD system in both of the Cartesian and cylindrical coordinates. One of the challenges in solving relativistic hydrodynamics is how to preserve the physical bounds of the solution. Compared with the non-relativistic one, for the relativistic hydrodynamics, the primitive variables and the conservative variables are related in a nonlinear way due to the Lorentz transformation. In order to recover the positive pressure from the conserva-

tive variables in each time step, a nonlinear equation needs to be solved. However, the existence of the positive root of this nonlinear equation requires the numerical approximation for the conservative variables to fall into certain admissible set. We develop a high-order bound-preserving DG method to solve this robutness problem in both one and two dimensions with both Cartisian and cylindrical coordinates respectively. The method is tested on extreme relativistic cases including the relativistic jets.

This dissertation consists of the following chapters. In Chapter 2, we will do a brief review of the DG methods and the general idea of the positivity and bound-preserving techniques. Then Chapter 3 includes the extension of the positivity-preserving limiter to the implicit DG methods. The extension to the RHD system is introduced in Chapter 4. Concluding remarks will be given in Chapter 5.

Chapter Two

DG Methods and Framework for Bound-Preserving Techniques

In this chapter, we use the following one-dimensional conservation law system

$$\begin{aligned} \mathbf{u}_t + \mathbf{f}(\mathbf{u})_x &= \mathbf{0}, \quad (x,t) \in \Omega \times [0,\infty), \\ \mathbf{u}(x,0) &= \mathbf{u}_0(x), \quad x \in \Omega, \end{aligned} \tag{2.1}$$

with appropriate boundary conditions to illustrate the formulation of the DG method and the general framework for the positivity and bound-preserving technique. Here $\Omega = [0, 2\pi]$, $\mathbf{u} : [0, 2\pi] \times \mathbb{R}^+ \to \mathbb{R}^d$ is a vector of state variables and the function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ is the flux function. When $d = 1$, we use the non-bold face letters u and f to denote the scalar variable and the flux function.

The well posedness of the problem (2.1) requires that for each fixed $\mathbf{u}$, the Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ has real eigenvalues $\{\zeta_j(\mathbf{u})\}_{j=1}^d$, and has a complete set of d linearly independent eigenvectors.

2.1 DG methods for conservation laws

To define the DG scheme for (2.1), let us first fix some notations. We decompose the domain $\Omega = [0, 2\pi]$ into N subintervals, $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, for $j = 1, 2, \cdots, N$. The size of each subinterval is denoted by h_j. Define $\hat{I} = [-1, 1]$ to be the reference cell and define $T_j(x) = 2(x - x_j)/h_j$ to be the affine mapping between the intervals I_j and $\hat{I}$, where $x_j = (x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})/2$ is the midpoint of I_j. Moreover, let $(\cdot, \cdot)_j$ denote the usual L^2 inner product on I_j, $(\cdot, \cdot)_{\hat{I}}$ the one on $\hat{I}$ and let $(\cdot, \cdot)_\Omega$ denote the inner product on Ω. Then we define the approximation space

$$V_h = \{v \in L^2(\Omega) : v|_{I_j} \in P_k(I_j), \forall j = 1, \cdots N\}$$

where $P_k(I_j)$ denotes the polynomial space on I_j with degree up to k and its vector version $\mathbf{V}_h = [V_h]^d$.

2.1.1 Spacial discretizations

For fixed t, the spacial discretization with the DG method for (2.1) is defined by seeking the approximation $\mathbf{u}_h(t) \in \mathbf{V}_h$, such that in each subinterval I_j, we have

$$\frac{d}{dt}(\mathbf{u}_h(t), \mathbf{v})_j - (\mathbf{f}(\mathbf{u}_h(t)), \mathbf{v}_x)_j + \hat{\mathbf{f}}_{j+\frac{1}{2}}(\mathbf{u}_h(t))\mathbf{v}(x^-_{j+\frac{1}{2}}) - \hat{\mathbf{f}}_{j-\frac{1}{2}}(\mathbf{u}_h(t))\mathbf{v}(x^+_{j-\frac{1}{2}}) = \mathbf{0}, \quad (2.2)$$

$\forall \mathbf{v} \in \mathbf{V}_h$, where $\hat{\mathbf{f}}_{j+\frac{1}{2}}(\mathbf{u}) = \hat{\mathbf{f}}(\mathbf{u}(x^-_{j+\frac{1}{2}}), \mathbf{u}(x^+_{j+\frac{1}{2}}))$ and $\hat{\mathbf{f}}(\cdot,\cdot)$ is an exact or approximate Riemann solver [69] for system and a monotone flux [32] for scalar equation. There are many choices for $\hat{\mathbf{f}}$ (see [69, 32]), for example, the global Lax-Friedrichs flux in the following form

$$\hat{\mathbf{f}}(\mathbf{a}, \mathbf{b}) = \frac{1}{2}[\mathbf{f}(\mathbf{a}) + \mathbf{f}(\mathbf{b}) - \alpha(\mathbf{b} - \mathbf{a})], \quad (2.3)$$

where $\alpha = \max_i \|\zeta_i(\mathbf{u}(\cdot, t))\|_\infty$.

2.1.2 Time discretizations

The semi-discrete scheme (2.2) can be written in the following compact form,

$$\frac{d}{dt}(\mathbf{u}_h(t), \mathbf{v})_j = L_j(\mathbf{u}_h(t), \mathbf{v}), \quad \forall \mathbf{v} \in V_h, \quad \forall j = 1, \ldots, N \quad (2.4)$$

with the shorthand notation

$$L_j(\mathbf{u}_h(t), \mathbf{v}) = (\mathbf{f}(\mathbf{u}_h(t)), \mathbf{v}_x)_j - \left[\hat{\mathbf{f}}_{j+\frac{1}{2}}(\mathbf{u}_h(t))\mathbf{v}(x^-_{j+\frac{1}{2}}) - \hat{\mathbf{f}}_{j-\frac{1}{2}}(\mathbf{u}_h(t))\mathbf{v}(x^+_{j-\frac{1}{2}})\right]. \quad (2.5)$$

For the ODE system (2.4), one can either use explicit methods or implicit ones to further discretize it.

For the explicit methods, a common choice is the class of strong stability preserving (SSP) methods [64, 65, 23]. Among them the most popular ones are the following third-order SSP Runge-Kutta method [65],

$$\begin{aligned}
(\mathbf{u}_h^{(1)}, \mathbf{v})_j &= (\mathbf{u}_h^n, \mathbf{v})_j + \Delta t L_j(\mathbf{u}_h^n, \mathbf{v})_j, \quad \forall j, \forall \mathbf{v} \in \mathbf{V}_h, \\
(\mathbf{u}^{(2)}, \mathbf{v})_j &= \frac{3}{4}(\mathbf{u}^n, \mathbf{v})_j + \frac{1}{4}(\mathbf{u}_h^{(1)}, \mathbf{v})_j + \frac{1}{4}\Delta t L_j(\mathbf{u}_h^{(1)}, \mathbf{v}), \quad \forall j, \forall \mathbf{v} \in \mathbf{V}_h, \\
(\mathbf{u}_h^{n+1}, \mathbf{v})_j &= \frac{1}{3}(\mathbf{u}_h^n, \mathbf{v})_j + \frac{2}{3}(\mathbf{u}_h^{(2)}, \mathbf{v})_j + \frac{2}{3}\Delta t L_j(\mathbf{u}_h^{(2)}, \mathbf{v}), \quad \forall j, \forall \mathbf{v} \in \mathbf{V}_h,
\end{aligned} \tag{2.6}$$

and the following third order SSP multistep method [64],

$$\begin{aligned}
(\mathbf{u}_h^{n+1}, \mathbf{v})_j =& \frac{16}{27}\left[(\mathbf{u}_h^n, \mathbf{v})_j + 3\Delta t L_j(\mathbf{u}_h^n, \mathbf{v})\right] \\
&+ \frac{11}{27}\left[(\mathbf{u}_h^{n-3}, \mathbf{v})_j + \frac{12}{11}\Delta t L_j(\mathbf{u}_h^{n-3}, \mathbf{v})\right], \quad \forall j, \forall \mathbf{v} \in \mathbf{V}_h.
\end{aligned} \tag{2.7}$$

where $\mathbf{u}_h^n \in \mathbf{V}_h$ denotes the numerical approximation at time level n.

For the implicit discretizations, many methods have been considered in practice, including the backward difference formula (BDF) methods [27], the implicit modified extended BDF (MEBDF) method [46], linearly implicit Rosenbrock-type RK method [2] and the fully implicit RK methods [28, 49].

In this dissertation, we considered the simplest one, i.e., the backward Euler method

$$(\mathbf{u}_h^{n+1}, \mathbf{v})_j - \Delta t L_j(\mathbf{u}_h^{n+1}, \mathbf{v}) = (\mathbf{u}_h^n, \mathbf{v})_j \tag{2.8}$$

for all $\mathbf{v} \in \mathbf{V}_h$ and $j = 1, \cdots, N$. And the Crank-Nicolson method

$$(\mathbf{u}_h^{n+1}, \mathbf{v})_j - \frac{\Delta t}{2} L_j(\mathbf{u}_h^{n+1}, \mathbf{v}) = (\mathbf{u}_h^n, \mathbf{v})_j + \frac{\Delta t}{2} L_j(\mathbf{u}_h^n, \mathbf{v}) \tag{2.9}$$

for all $\mathbf{v} \in \mathbf{V}_h$ and $j = 1, \cdots, N$.

In the following, we use $\mathbf{u}_j^n$ to denote $\mathbf{u}_h^n|_{I_j}$, $(\mathbf{u}_{j+\frac{1}{2}}^n)^{\pm}$ to denote $\mathbf{u}_j^n(x_{j+\frac{1}{2}}^{\pm})$ and use $\bar{\mathbf{u}}_j^n$ to denote the cell average of $\mathbf{u}_j^n$ in the interval I_j.

2.2 General framework for positivity and bound preserving techniques

The physical admissible solution to (2.1) usually falls into some admissible set G. In most applications, the set G can be verified to be a convex set. For example, for the scalar equation, it is well known that the entropy solution satisfies the following maximum principle

$$\min_{x \in \Omega} u_0(x) \leq u(x,t) \leq \max_{x \in \Omega} u_0(x), \quad \forall t \geq 0. \tag{2.10}$$

In this case the set G is $[m, M]$ with $m = \min_{x \in \Omega} u_0(x)$ and $M = \max_{x \in \Omega} u_0(x)$.

Our goal is to design a high-order DG scheme such that

$$\mathbf{u}_h^n(x) \in G, \forall x \in \Omega \implies \mathbf{u}_h^{n+1}(x) \in G, \forall x \in \Omega \tag{2.11}$$

A general framework for constructing such DG schemes has been proposed in [86, 87], which can be summarized in two steps as below.

Step 1 First, in each cell I_j, given the approximation at time level n is admissible, i.e., $\mathbf{u}_j^n(x) \in G$ for any $x \in I_j$, find a sufficient condition such that we have part of the approximation, i.e., the cell average at the next time level is in G, $\bar{\mathbf{u}}_j^{n+1} \in G$.

Step 2 Next, in each cell I_j, we limit the whole approximation $\mathbf{u}_j^{n+1}(x)$ into the set G by invoking the following scaling limiter [33, 86, 87],

$$\tilde{\mathbf{u}}_j^{n+1} = \theta_j[\mathbf{u}_j^{n+1} - \bar{\mathbf{u}}_j^{n+1}] + \bar{\mathbf{u}}_j^{n+1} \tag{2.12}$$

where $\theta_j \in (0, 1)$ is solved such that $\tilde{\mathbf{u}}_j^{n+1} \in G$.

The main difficulty lies in the first step. The sufficient conditions for DG methods with explicit SSP time discretizations, have been derived in [86, 87, 89] for general scalar conservation laws and the compressible Euler system with and without source terms respectively. In Chapter 4, we will extend this procedure to the RHD system and design a high-order bound preserving explicit DG methods.

For the implicit time discretization, which is popular in practical application, the positivity-preserving DG methods been less explored. In Chapter 3, we mainly investigate the simplest case, i.e., the backward Euler time discretization for the one dimensional conservation laws. And we also briefly consider the Crank-Nicolson time discretization for linear DG methods on uniform meshes. To achieve the goal in Step 1, a set of sufficient conditions are derived for the scalar linear equations. These conditions are further verified by numerical examples for nonlinear scalar equations. The applicability of the proposed method is also tested for the compressible Euler systems.

As for the second step, the main issue is whether this modification will destroy

the original high-order accuracy of the DG method or not. For scalar equations, the complete result for arbitrary order of accuracy is first given in [86] and the proof is further refined in [85]. The result is included in the following lemma.

Lemma 2.2.1. *For scalar equations where $G = [m, M]$, the modified polynomial $\tilde{u}_j^{n+1}(x)$ as defined in (2.12) preserves the original order of accuracy in the following sense,*

$$|\tilde{u}_j^{n+1}(x) - u_j^{n+1}(x)| \leq C_k \max_{x \in I_j} |u_j^{n+1}(x) - u(x)|$$

where u is the smooth solution to (2.1) with $d = 1$.

Basically, what this lemma says is that for the scalar equation the error we commit in the limiting procedure is bounded by the error of the original approximation up to a constant depending on k.

For the system, even though there is no rigorous result as Lemma 2.2.1, both truncation error analysis [88] and numerical results show that the scaling limiter (2.12) preserves the high-order accuracy as well.

Chapter Three

Positivity-Preserving DG Methods with Implicit Time Stepping for Conservation Laws

In this chapter, we consider the following conservation law

$$\begin{aligned} u_t + f(u)_x &= 0, \quad (x,t) \in [0, 2\pi] \times [0, +\infty) \\ u(x,0) &= u_0(x), \quad x \in [0, 2\pi] \end{aligned} \tag{3.1}$$

and its system version with appropriate boundary conditions. We focus on this one-dimensional case, even though the result can be easily generalized to multidimensional tensor product meshes and polynomial spaces.

3.1 Introduction

For scalar conservation laws, it is well known that the entropy solution satisfies the following maximum principle

$$\min_{x\in[0,2\pi]} u_0(x) \le u(x,t) \le \max_{x\in[0,2\pi]} u_0(x), \quad \forall t \ge 0.$$

In particular, if the initial condition is positive, then the entropy solution should satisfy the following positivity-preserving property

$$u_0(x) \ge 0 \implies u(x,t) \ge 0, \quad \forall t \ge 0 \tag{3.2}$$

For systems, even though the entropy solution does not satisfy the maximum principle in general, the physically relevant solution, for example the density and pressure in the compressible Euler system, is always positive. In this chapter, the words "positive" and "positivity" are used actually to mean "nonnegative" and "nonnegativity". We shall use "strictly positive" to mean the usual "positive".

When designing numerical methods, we would like our numerical approximations

to respect this positivity-preserving property (3.2), not only because it makes the numerical approximation physically meaningful, but also it makes the numerical scheme more robust, since negative values sometimes cause ill-posedness of the problem and blow-ups of the numerical algorithm [24]. In recent years, the positivity-preserving DG schemes have been actively designed and applied for solving hyperbolic conservation laws [86, 87, 78, 79, 55, 9, 85]. All these methods are coupled with explicit temporal discretizations, such as strong stability preserving (SSP) Runge-Kutta (RK) methods [65, 23] and multi-step methods [64]. Explicit temporal discretizations enjoy many advantages, for example, the easiness in handling the nonlinear terms and boundary conditions, high-order accuracy with SSP properties [23], low storage requirement and so on. However, they suffer from the CFL constraint. For DG methods, to obtain the linear stability [1] or the maximum-principle stability [86], the CFL constraint becomes more and more severe as we increase the polynomial degree in the approximation space. Such stringent time stepping restriction makes explicit methods impractical in computations involving unstructured and extremely varying meshes, viscous effect [51] or long-time simulations for steady-state calculation [27].

To circumvent the severe CFL constraint of explicit methods, implicit time discretizations, which allow larger CFL numbers especially for stiff problems, are widely used in practice, especially in the CFD community to solve compressible flow problems [27, 28, 6, 51, 50, 49, 46, 2]. Although most of the effort has been made for increasing accuracy of the time discretization and for increasing the efficiency of the nonlinear solver, there are not many works in the literature concerning the positivity-preserving property of implicit methods. For compressible turbulent flow problems, Batten et al. [3] have proposed a positive finite difference scheme by splitting the fluxes into "implicit" and "correction" parts and the source term into positive and negative parts. The "implicit" and negative terms are treated implicitly via the

Patankar trick [48]. In [43, 44, 45], Moryossef and Levy have developed implicit unconditional positive finite volume schemes for unsteady turbulent flows. Their main idea to preserve the positivity is to make the Jacobian matrix in each implicit time step an M-matrix. All these methods mentioned are low-order accurate and are complicated to generalize to high order. For DG methods, in [40], Meister and Ortleb have constructed an unconditional implicit positive DG scheme for solving shallow water equations. The positivity of the numerical approximation is preserved via a modified Patankar trick [48]. The method is shown to be conservative and unconditional positivity-preserving, but only third-order accuracy is proved by a truncation error argument with no rigorous proof for arbitrary high-order spacial accuracy. In [82], Yuan, Cheng and Shu have developed a high-order unconditionally positive implicit DG method for radiative transfer equations. The positivity is preserved by utilizing the particular boundary conditions of the problem and by designing a novel rotational limiter.

In this chapter, we extend the general framework in Chapter 1 to implicit temporal discretizations and develop a positivity-preserving DG method with high-order spacial accuracy for one-dimensional conservation laws. We consider the DG method for the following two reasons. First, the discontinuous feature of its approximation space makes it a good fit for parallel implementation and for handling unstructured and extremely varying meshes. Second, for a class of implicit temporal discretizations, including the backward Euler and Crank-Nicolson methods, Jiang and Shu have shown in [29] via the cell entropy inequality that the fully discrete schemes for the nonlinear conservation laws are unconditionally L^2-stable, which works for arbitrary triangulation and any spacial order of accuracy. In this chapter, we consider the simplest implicit time discretization, i.e., the backward Euler temporal discretization as well as the Crank-Nicolson time discretization. Our focus is on constructing

a spatially high-order positivity-preserving DG scheme. The main conclusion is that in order to generalize the Zhang-Shu positivity-preserving limiter [86, 87] to the backward Euler DG scheme, a lower bound for the CFL number is required. This is proved theoretically for linear scalar equations and numerically verified for nonlinear equations. For the Crank-Nicolson method with P^1 element on uniform mesh, it is proved that both upper and lower bounds for the CFL number are required to have the positivity-preserving limiter take effect. The proposed positivity-preserving limiter is inexpensive and easy to implement. It not only preserves the positivity and high-order spacial accuracy but also makes the numerical scheme more robust, in the sense that it accelerates the convergence towards the steady-state solution and adds robustness to the nonlinear solver for extreme test cases.

The organization of this chapter is as follows. In Section 3.2 the positivity-preserving technique is introduced. In particular, a CFL condition is derived to ensure the positiveness of the scheme. Then the introduction for the scheme for the compressible Euler system is given in Section 3.3. Numerical experiments are presented in Section 3.5 and concluding remarks are given in Section 3.6.

3.2 Positivity-preserving backward Euler DG scheme for scalar equations

The backward Euler DG scheme for solving (3.1) is as introduced in Section 2.1, to seek $u_h^{n+1} \in V_h$ such that in each subinterval I_j, we have

$$(u_h^{n+1}, v)_j - \Delta t L_j(u_h^{n+1}, v) = (u_h^n, v)_j \tag{3.3}$$

for any $v \in V_h$. The spatial operator L_j is as defined in (2.5). The definition of a positivity-preserving DG scheme goes as below.

Definition 3.2.1. *A DG scheme is defined to be positivity preserving if given* $u_h^n(x) \geq 0$, *for any* $x \in \Omega$, *then we have* $u_h^{n+1}(x) \geq 0$, $\forall x \in \Omega$.

In order to obtain more efficient implementations, this definition can be slightly relaxed to require positivity at specified quadrature points rather than at all the points.

Definition 3.2.2 (Relaxed version)**.** *A DG scheme is defined to be positivity preserving if in each subinterval* I_j, *given* $u_j^n(x_j^\alpha) \geq 0$, *then we have* $u_j^{n+1}(x_j^\alpha) \geq 0$, *where* $\{x_j^\alpha\}$ *is a set of selected quadrature points in cell* I_j *for each* $j = 1, \cdots, N$.

In this section, we introduce how to add the positivity-preserving property to the scheme (3.3) by extending the framework in Section 2.2. As mentioned, the main difficulty lies in the first step, i.e., to derive sufficient conditions under which given $u_h^n(x) \geq 0$ for any $x \in \Omega$, we have $\bar{u}_j^{n+1} \in \Omega$ for any j. For implicit DG methods, the polynomial $u_j^{n+1}(x)$ depends on the previous information $u_h^n(x)$ in a global and implicit way. In this section, we would first show how to overcome this difficulty for scalar linear equations and derive a CFL condition. Then we will introduce the scaling limiter and then summarize the algorithm.

3.2.1 Preliminaries

Let us first recall some definitions and results that will be useful in the following analysis.

Block circulant matrices

For linear problems with periodic boundary conditions, when the mesh is uniform, the left hand side of the scheme (3.3) has a block circulant structure. The following block circulant matrices will become handy in the analysis for the uniform mesh cases. For a thorough introduction, one can refer to [15].

Definition 3.2.3. *A matrix A, is defined to be a block circulant matrix bcirc$(A_1, \ldots, A_n)$, if it has the following structure*

$$A = \begin{pmatrix} A_1 & A_2 & \cdots & A_n \\ A_n & A_1 & \cdots & A_{n-1} \\ & & \cdots & \\ A_2 & A_3 & \cdots & A_1 \end{pmatrix}$$

The inverse of a block circulant matrix has the same structure.

Lemma 3.2.1. *The inverse of a block circulant matrix is block circulant.*

M-matrices

Another useful tool is the so-called M-matrix. For a thorough introduction, one can refer to [4]. To define it let us first set

$$\mathcal{Z}^{n\times n} = \{A = (a_{ij}) \in \mathbb{R}^{n\times n} : a_{ij} \leq 0,\ i \neq j\}$$

which is the set of all the $n \times n$ real matrices with nonpositive off-diagonal entries. In [52], the author listed forty equivalent characterizations for M-matrices. For our purpose, we adopt the following one as the definition.

Definition 3.2.4. *A matrix $A \in \mathcal{Z}^{n\times n}$ is called an M-matrix if A is inverse-positive, that is, A^{-1} exists and each entry of A^{-1} is nonnegative.*

M-matrices have the following equivalent characterization [52].

Theorem 3.2.1. *A matrix $A = (a_{ij}) \in \mathcal{Z}^{n\times n}$ is an M-matrix if and only if $a_{ii} > 0$, $1 \le i \le n$, and there exists a positive diagonal matrix $D = diag\{d_1, \cdots, d_n\}$ such that AD is strictly diagonally dominant, that is, $a_{ii}d_i > \sum_{j\neq i} |a_{ij}|d_j$ for $1 \le i \le n$.*

In particular, if D is the identity matrix, we have the following corollary.

Corollary 3.2.1. *A matrix $A = (a_{ij}) \in \mathcal{Z}^{n\times n}$ is an M-matrix if $a_{ii} > 0$ and it is strictly diagonally dominant.*

Legendre polynomials

In the following, we also utilize properties of Legendre polynomials. We consider the standard Legendre polynomials $\{p_n(x)\}$ defined on $\hat{I}$ by the following recursive relationship

$$(n+1)p_{n+1}(x) = (2n+1)xp_n(x) - np_{n-1}(x), \quad p_0(x) = 1, \quad p_1(x) = x, \quad x \in \hat{I} \tag{3.4}$$

In the following lemma, we collect some properties of Legendre polynomials that will be useful in the following analysis. For the proof, one can refer to [5].

Lemma 3.2.2. *Legendre polynomials defined in (3.4) have the following properties*

1. $p_n(1) = 1$, $p_n(-1) = (-1)^n$.
2. $\int_{\hat{I}} p_n(x)\, p_m(x)\, dx = \frac{2}{2n+1}\delta_{nm}$, *where δ_{nm} is the Kronecker delta.*

3. $(2n+1)p_n(x) = \frac{d}{dx}[p_{n+1}(x) - p_{n-1}(x)]$.

4. *Rodrigues' formula* $p_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}[(x^2-1)^n]$.

3.2.2 CFL condition for P^1-DG methods on uniform meshes

We consider the linear equation

$$\begin{aligned} u_t + u_x &= 0, \quad x \in \Omega \\ u(x,0) &= u_0(x), \end{aligned} \tag{3.5}$$

with periodic boundary condition. Then the scheme (3.3) becomes

$$(u_h^{n+1}, v)_j - \Delta t(u_h^{n+1}, v_x)_j + \Delta t[(u_{j+\frac{1}{2}}^{n+1})^- v_{j+\frac{1}{2}}^- - (u_{j-\frac{1}{2}}^{n+1})^- v_{j-\frac{1}{2}}^+] = (u_h^n, v)_j \tag{3.6}$$

Given $u_h^n(x) \geq 0, \forall x \in \Omega$, we want to derive a CFL condition, under which the cell average $\bar{u}_j^{n+1} \geq 0, \forall j$.

For the linear equation (3.5) with linear DG methods ($k = 1$) on uniform meshes, the derivation of the CFL condition is straightforward due to the following reasons.

1. Given a linear function in I_j being positive is equivalent to say the function is positive at the two end points $x_{j\pm\frac{1}{2}}$. And hence we can represent this polynomial with the Lagrange interpolating polynomials at the two endpoints.

2. For uniform mesh, we are able to obtain an exact formula for the inverse operator and hence we can get a CFL condition that is necessary and sufficient for the positivity of the cell averages at the next time level.

If we choose a set of basis functions $\{\phi_l^j\}_{l=0}^k \in P^k(I_j)$, then (3.6) can be rewritten in the following matrix form

$$M^j \mathbf{c}_j^{n+1} - M^j \mathbf{c}_j^n + \Delta t B_1^j \mathbf{c}_j^{n+1} - \Delta t B_2^j \mathbf{c}_{j-1}^{n+1} - \Delta t S^j \mathbf{c}_j^{n+1} = \mathbf{0}$$

which is equivalent to

$$(M^j + \Delta t B_1^j - \Delta t S^j)\mathbf{c}_j^{n+1} - \Delta t B_2^j \mathbf{c}_{j-1}^{n+1} = M^j \mathbf{c}_j^n \tag{3.7}$$

where

$$M_{li}^j = (\phi_l^j, \phi_i^j)_j, \quad S_{li}^j = (\partial_x \phi_l^j, \phi_i^j)_j, \quad (B_1^j)_{li} = \phi_l^j(x_{j+\frac{1}{2}})\phi_i^j(x_{j+\frac{1}{2}}),$$
$$(B_2^j)_{li} = \phi_l^j(x_{j-\frac{1}{2}})\phi_i^{j-1}(x_{j-\frac{1}{2}}).$$

If we transform all the integrations to the reference cell $\hat{I}$ via the affine mapping $T_j(x) = \frac{x - x_j}{h_j/2}$, we will obtain

$$(M + 2\lambda_j B_1 - 2\lambda_j S)\mathbf{c}_j^{n+1} - 2\lambda_j B_2\, \mathbf{c}_{j-1}^{n+1} = M \mathbf{c}_j^n \tag{3.8}$$

where

$$M_{li} = (\hat{\phi}_l, \hat{\phi}_i)_{\hat{I}}, \quad S_{li} = (\partial_x \hat{\phi}_l, \hat{\phi}_i)_{\hat{I}}, \quad B_1 = B_1^j, \quad B_2 = B_2^j$$

The polynomials $\{\hat{\phi}_l\}_{l=0}^k$ are the basis of $P^k(\hat{I})$, with $\phi_j^l = \hat{\phi}_l(T_j(x))$. The coefficient vector $\mathbf{c}_j^{n+1}$ is unchanged.

For uniform mesh $\lambda_j = \lambda, j = 1, \cdots, N$. If we set $A = M + 2\lambda B_1 - 2\lambda S$ and $B = -2\lambda B_2$, the scheme (3.8) can be written in the following compact form

$$R\mathbf{c}^{n+1} = L\mathbf{c}^n \implies \mathbf{c}^{n+1} = R^{-1}L\mathbf{c}^n \tag{3.9}$$

where $R = \text{bcirc}(A, O, \cdots, O, B)$ and $L = \text{Diag}(M)$.

If we set

$$\Theta = \begin{pmatrix} \vec{\Theta}_1 & 0 & \cdots & \cdots & 0 \\ 0 & \vec{\Theta}_2 & 0 & \cdots & 0 \\ & & \cdots & & \\ 0 & \cdots & \cdots & 0 & \vec{\Theta}_N \end{pmatrix}^{N\times(k+1)N}$$

where $\vec{\Theta}_j = (\bar{\phi}_0^j, \cdots, \bar{\phi}_k^j)$.

Then the cell average $\bar{\mathbf{u}}^{n+1}$ can be represented as in the following cell average equation

$$\bar{\mathbf{u}}^{n+1} = \Theta R^{-1} L \mathbf{c}^n \tag{3.10}$$

All we need to do is to work out the inverse matrix R^{-1}. In order to do this, first note that the matrix R a block circulant matrix and hence by Lemma 3.2.1, its inverse is still block circulant. We set $R^{-1} = \text{bcirc}(D_1, D_2, \cdots, D_N)$. Since $RR^{-1} = I$, direct calculation gives

$$\begin{aligned} AD_1 + BD_2 = I, \quad AD_2 + BD_3 = O \\ AD_3 + BD_4 = O, \quad \cdots \\ AD_{N-1} + BD_N = O, \quad AD_N + BD_1 = O \end{aligned}$$

And it is easy to obtain

$$\begin{aligned} D_1 &= [A + (-1)^{N-1} B (A^{-1}B)^{N-1}]^{-1}, \\ D_k &= (-1)^{N+1-k} (A^{-1}B)^{N+1-k} D_1, \quad k = 2, 3, \cdots, N \end{aligned} \tag{3.11}$$

For the linear element ($k = 1$), we have the following result.

Theorem 3.2.2. *For backward Euler linear DG method, at time level n given*

$u_h^n(x) \geq 0$ for any $x \in \Omega$, then we have $\bar{u}_j^{n+1} \geq 0$, for any $j = 1, \cdots, N$ if and only if

$$\lambda \geq \frac{1}{3} \tag{3.12}$$

Proof. We choose the basis $\{\hat{\phi}_0, \hat{\phi}_1\}$ to be Lagrange interpolating polynomials at $x = \pm 1$. Then we have $L = I$ and

$$-A^{-1}B = \begin{pmatrix} \frac{2\lambda(3\lambda-1)}{6\lambda^2+4\lambda+1} & 0 \\ \frac{2\lambda(3\lambda+2)}{6\lambda^2+4\lambda+1} & 0 \end{pmatrix}$$

and if we set $a = \frac{2\lambda(3\lambda-1)}{6\lambda^2+4\lambda+1}$ and $b = \frac{2\lambda(3\lambda+2)}{6\lambda^2+4\lambda+1}$, we furthermore have

$$(-A^{-1}B)^{N-1} = \begin{pmatrix} a^{N-1} & 0 \\ a^{N-2}b & 0 \end{pmatrix}$$

and

$$\begin{aligned} C :&= A + B(-A^{-1}B)^{N-1} \\ &= \begin{pmatrix} \lambda+1 & -3\lambda \\ \lambda & 3\lambda+1 \end{pmatrix} + \begin{pmatrix} 2\lambda a^{N-1} & 0 \\ -4\lambda a^{N-1} & 0 \end{pmatrix} \\ &= \begin{pmatrix} \lambda+1+2\lambda a^{N-1} & -3\lambda \\ \lambda-4\lambda a^{N-1} & 3\lambda+1 \end{pmatrix} \end{aligned}$$

It is easy to obtain

$$\det C = 6\lambda^2 + 4\lambda + 1 - 2\lambda(3\lambda-1)a^{N-1} = (6\lambda^2+4\lambda+1)(1-a^N)$$

Since

$$|a| = \frac{|2\lambda(3\lambda-1)|}{6\lambda^2+4\lambda+1} < 1, \quad \forall \lambda > 0$$

we have $\det C > 0$ and hence C is invertible and we have

$$D_1 = C^{-1} = \frac{1}{\det C}\begin{pmatrix} 3\lambda+1 & 3\lambda \\ -\lambda+4\lambda a^{N-1} & \lambda+1+2\lambda a^{N-1} \end{pmatrix}$$

The local cell average operator $\vec{\theta} := \vec{\Theta}_j = (\frac{1}{2}, \frac{1}{2})$ for any j, and we claim that

$$\vec{\theta} D_1 = \frac{1}{2\det C}(2\lambda+1+4\lambda a^{N-1}, 4\lambda+1+2\lambda a^{N-1})^T > 0, \forall \lambda > 0$$

The second term is always positive since $a \in (-1, 1)$. As for the first term, it is always positive when $a > 0$ or when N is an odd number. When $N = 2m$ is an even number and $a < 0$, we have

$$2\lambda+1+4\lambda a^{2m-1} = 2\lambda+1+4\lambda\, a\, a^{2m-2} \geq 2\lambda+1+4\lambda a = 2\lambda+1+\frac{8\lambda^2(3\lambda-1)}{6\lambda^2+4\lambda+1} > 0$$

for any $\lambda > 0$. Therefore, we have shown the previous statement.

As for $i > 1$, we have

$$\begin{aligned} D_i &= (-A^{-1}B)^{N+1-i} D_1 \\ &= \begin{pmatrix} a^{N+1-i} & 0 \\ a^{N-i}b & 0 \end{pmatrix} D_1 \\ &= \begin{pmatrix} a^{N+1-i}(D_1)_{11} & a^{N+1-i}(D_1)_{12} \\ a^{N-i}b(D_1)_{11} & a^{N-i}b(D_1)_{12} \end{pmatrix} \end{aligned}$$

and then

$$\mathbf{d}_i := \vec{\theta} D_i = \left(\frac{(D_1)_{11}}{2} a^{N-i}(a+b), \frac{(D_1)_{12}}{2} a^{N-i}(a+b) \right)$$

Since

$$(D_1)_{11} = \frac{3\lambda+1}{\det C} > 0, \quad (D_1)_{12} = \frac{3\lambda}{\det C} > 0, \quad a+b = \frac{12\lambda^2+2\lambda}{6\lambda^2+4\lambda+1} > 0, \quad \forall \lambda > 0$$

we have

$$\mathbf{d}_i > 0, \quad \forall i \iff a > 0 \iff \lambda > \frac{1}{3}$$

The cell average $\bar{u}_1^{n+1}$ is given by

$$\bar{u}_1^{n+1} = \sum_{j=1}^{N} \mathbf{d}_j \cdot (\hat{u}_h^n, \vec{\hat{\phi}})_{\hat{I}}$$

Obviously, when $\lambda > \frac{1}{3}$, $\mathbf{d}_j > 0$ for any j and hence we have $\bar{u}_j^{n+1} > 0$.

Otherwise, consider the following example, if we set $\hat{u}_h^n \geq 0$ to be supported only in the cell I_j where $N-j$ is odd. Since moment vector $(\hat{u}_h^n, \vec{\hat{\phi}})_{\hat{I}}$ is always positive, since for linear element the basis functions $\hat{\phi}_l$, $l = 0, 1$ are positive everywhere. When $\lambda < \frac{1}{3}$, the vector $\mathbf{d}_j < 0$, and hence $\bar{u}_j^{n+1} = \mathbf{d}_j \cdot (u_h^n, \vec{\hat{\phi}})_{\hat{I}} < 0$.

Therefore, we have

$$\bar{u}_j^{n+1} \geq 0, \quad \forall j \iff \mathbf{d}_j \geq 0 \iff \lambda \geq \frac{1}{3}$$

□

Remark 3.2.1. *The sharpness of the bound can be verified by considering the following example*

$$u_h^0 = \begin{cases} 1, & \text{if } x \in I_j \\ 0, & \text{otherwise} \end{cases}$$

After one step, we can check the $\min_j \bar{u}_j^1$.

The analysis above, even though gives necessary and sufficient condition, relies on the uniform mesh assumption and works for small k. For non-uniform meshes, the matrix R is no longer a block circulant matrix and hence it is hard to obtain exact formula for R^{-1}. For large k, even for the uniform mesh case, the algebra will become very complicated and it seems impossible to extend to general polynomial degree k.

3.2.3 CFL condition for arbitrary polynomial degree on general meshes

We continue considering the linear equation

$$\begin{aligned} u_t + u_x &= 0, \quad x \in \Omega \\ u(x,0) &= u_0(x), \end{aligned} \tag{3.13}$$

with periodic boundary condition. And still consider the following DG scheme

$$(u_h^{n+1}, v)_j - \Delta t(u_h^{n+1}, v_x)_j + \Delta t[(u_{j+\frac{1}{2}}^{n+1})^- v_{j+\frac{1}{2}}^- - (u_{j-\frac{1}{2}}^{n+1})^- v_{j-\frac{1}{2}}^+] = (u_h^n, v)_j \tag{3.14}$$

Now we consider the general mesh and arbitrary polynomial degree k. In this case, the inverse of the right hand side matrix can not be easily obtained. Therefore, we have to turn to other approaches. We first express $\bar{u}_j^{n+1}$ in terms of u_h^n. The idea is to take $k+1$ different test functions as probes to extract the information out from $u_h^{n+1}(x)$ in terms of $u_h^n(x)$.

First, let us take $v = 1$ in the scheme (3.14), we have

$$\bar{u}_j^{n+1} + \lambda_j[(u_{j+\frac{1}{2}}^{n+1})^- - (u_{j-\frac{1}{2}}^{n+1})^-] = \bar{u}_j^n, \quad j = 1, \cdots, N$$

where $\lambda_j = \frac{\Delta t}{h_j}$. We can rewrite the system above in the matrix form as below

$$\Lambda^{-1}\bar{\mathbf{u}}^{n+1} + A(\mathbf{u}^{n+1})^- = \Lambda^{-1}\bar{\mathbf{u}}^n \tag{3.15}$$

where $\bar{\mathbf{u}}^n = (\bar{u}_1^n, \cdots, \bar{u}_N^n)^T$, $(\mathbf{u}^{n+1})^- = ((u_{\frac{3}{2}}^{n+1})^-, \cdots, (u_{N+\frac{1}{2}}^{n+1})^-)$, Λ and A are $N \times N$ matrices in the following form

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_N \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 0 & \cdots & -1 \\ -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & -1 & 1 \end{pmatrix}.$$

Next, we need to express the cell boundary value $(u_{j+\frac{1}{2}}^{n+1})^-$ by u_h^n. To this end, we need to take other special test functions. Recall that the Dirac delta function has the following expansion

$$\delta(x-y) = \frac{1}{2}\sum_{l=0}^{\infty}(2l+1)p_l(x)p_l(y), \quad x, y \in \hat{I}$$

where $p_l(x)$ is the standard Legendre polynomial defined in (3.4). Then we set $y = 1$, truncate the summation at the $(k+1)$th term and define

$$\hat{\delta}^k(x) = \frac{1}{2}\sum_{l=0}^{k}(2l+1)p_l(x) = \frac{k+1}{2}\frac{p_{k+1}(x) - p_k(x)}{x-1}, \quad x \in \hat{I} \tag{3.16}$$

We have employed the Christoffel-Darboux formula [22] in the last equality.

The following lemma says that this polynomial actually defines the Dirac delta function in $P^k(\hat{I})$ at the point $y = 1$.

Lemma 3.2.3. *The polynomial $\hat{\delta}^k$ has the following properties*

1. *$\hat{\delta}^k(x) \in P^k(\hat{I})$ and for any $w \in P^k(\hat{I})$, we have $(w, \hat{\delta}^k)_{\hat{I}} = w(1)$.*

2. *In the cell I_j, define*

$$\delta_j^k(x) = \frac{2}{h_j}\hat{\delta}^k\left(T_j(x)\right), \quad x \in I_j \tag{3.17}$$

then it is the delta function in $P^k(I_j)$ at $x_{j+\frac{1}{2}}$.

3. *The mass is concentrated at $x = 1$, in the sense that for any k and $j = 0, \cdots, k-1$, we have $(\hat{\delta}^k)^{(j)}(1) - (\hat{\delta}^k)^{(j)}(x) > 0$ for any $x \in [-1, 1)$.*

Proof. It is obvious that $\hat{\delta}^k \in P_k(\hat{I})$. For any polynomial $w \in P_k(\hat{I})$, we can write it as a linear combination of Legendre polynomials as below

$$w(x) = \sum_{l=0}^{k} c_l p_l(x), \quad x \in \hat{I}$$

Then by the definition of $\hat{\delta}^k$ and Lemma 3.2.2, we have

$$(w, \hat{\delta}^k)_{\hat{I}} = \sum_{l=0}^{k} c_l (p_l, \hat{\delta}^k)_{\hat{I}} = \sum_{l=0}^{k} c_l = \sum_{l=0}^{k} c_l p_l(1) = w(1)$$

The second property can be shown by a simple change of variable.

For the third property, by the definition of $\hat{\delta}^k$, it suffices to show

$$p_l^{(i)}(1) - p_l^{(i)}(x) > 0 \tag{3.18}$$

for $l = 0, 1, \cdots, k-1$, $i = 0, \ldots, l-1$ and for any $x \in [-1, 1)$. It is straightforward to verify that (3.18) holds for $l = 0, 1, 2$. For $l > 2$, by Lemma 3.2.2, we have,

$$p_l^{(i)}(x) = (2l-1)p_{l-1}^{(i-1)}(x) + p_{l-2}^{(i)}(x)$$

If (3.18) holds for $l \leq m-1$, then we have when $l = m$, for fixed $i < m$ and any $x \in [-1, 1)$,

$$p_m^{(i)}(1) = (2m-1)p_{m-1}^{(i-1)}(1) + p_{m-2}^{(i)}(1) > (2m-1)p_{m-1}^{(i-1)}(x) + p_{m-2}^{(i)}(x) = p_m^{(i)}(x)$$

Then by induction, we have proved (3.18) and hence the third property in the lemma.

□

With the help of the discrete delta function (3.17), we have the following representation of $(u_{j+\frac{1}{2}}^{n+1})^-$.

Lemma 3.2.4. *For linear scalar conservation law with $f(u) = u$ discretized by the scheme (3.14), we have*

$$\sigma_j^k (u_{j+\frac{1}{2}}^{n+1})^- = \xi_j^k \bar{u}_j^{n+1} + (\hat{u}_j^n, g_j^k)_{\hat{I}} \tag{3.19}$$

where

$$\sigma_j^k = 1 + \sum_{i=0}^{k-1} (2\lambda_j)^{i+1} (\alpha_i^k - \beta_i^k), \quad \xi_j^k = 2 \sum_{i=0}^{k} (2\lambda_j)^i \beta_i^k,$$
$$g_j^k(x) = \sum_{i=0}^{k-1} (2\lambda_j)^i \left[(\hat{\delta}^k)^{(i)}(x) - \beta_i^k \right]$$

and

$$\alpha_i^k = (\hat{\delta}^k)^{(i)}(1), \quad \beta_i^k = (\hat{\delta}^k)^{(i)}(-1) \tag{3.20}$$

Proof. For fixed $l = 0, 1, \cdots, k-1$, take the test function to be $(\delta_j^k)^{(l)}(x) - (\delta_j^k)^{(l)}(x_{j-\frac{1}{2}})$ in the scheme (3.14). By the definition of $\delta_j^k(x)$ and Lemma 3.2.3, we have

$$(u_j^{n+1},(\hat{\delta}^k)^{(l)}(T_j(x)))_j - h_j\beta_l^k\bar{u}_j^{n+1} - 2\lambda_j(u_j^{n+1},(\hat{\delta}^k)^{(l+1)}(T_j(x)))_j$$
$$+ \Delta t(u_{j+\frac{1}{2}}^{n+1})^-(\alpha_l^k - \beta_l^k) = (u_j^n,(\hat{\delta}^k)^{(l)}(T_j(x)))_j - h_j\beta_l^k\bar{u}_j^n$$

If we expand u_j^{n+1} in terms of the basis $\phi_j^l(x) = p_l(T_j(x))$, $l = 0,\ldots,k$, as $u_j^{n+1} = \sum_{l=0}^k (c_j^l)^{n+1}\phi_j^l$ and by a change of variable, we obtain

$$(\hat{u}_j^{n+1},(\hat{\delta}^k)^{(l)})_{\hat{I}} - 2\beta_l^k\bar{u}_j^{n+1} - 2\lambda_j(\hat{u}_j^{n+1},(\hat{\delta}^k)^{(l+1)})_{\hat{I}} + 2\lambda_j(u_{j+\frac{1}{2}}^{n+1})^-(\alpha_l^k - \beta_l^k)$$
$$= (\hat{u}_j^n,(\hat{\delta}^k)^{(l)})_{\hat{I}} - 2\beta_l^k\bar{u}_j^n$$

where $\hat{u}_j^n = \sum_{l=0}^k (c_j^l)^n p_l(x)$. Or equivalently,

$$(\hat{u}_j^{n+1},(\hat{\delta}^k)^{(l)})_{\hat{I}} - 2\lambda_j(\hat{u}_j^{n+1},(\hat{\delta}^k)^{(l+1)})_{\hat{I}}$$
$$= 2\beta_l^k\bar{u}_j^{n+1} - 2\lambda_j(u_{j+\frac{1}{2}}^{n+1})^-(\alpha_l^k - \beta_l^k) + (\hat{u}_j^n,(\hat{\delta}^k)^{(l)})_{\hat{I}} - 2\beta_l^k\bar{u}_j^n$$

If we set

$$D_l = (\hat{u}_j^{n+1},(\hat{\delta}^k)^{(l)})_{\hat{I}}, \quad C_l = 2\beta_l^k\bar{u}_j^{n+1} - 2\lambda_j(u_{j+\frac{1}{2}}^{n+1})^-(\alpha_l^k - \beta_l^k) + (\hat{u}_j^n,(\hat{\delta}^k)^{(l)})_{\hat{I}} - 2\beta_l^k\bar{u}_j^n$$

then we have

$$D_l - 2\lambda_j D_{l+1} = C_l, \quad l = 0,\cdots,k-1$$

and in particular, when $l = k-1$, we have

$$D_{k-1} = 2\lambda_j D_k + C_{k-1} = 2\lambda_j(\hat{u}_j^{n+1},(\hat{\delta}^k)^{(k)})_{\hat{I}} + C_{k-1} = 4\lambda_j\beta_k^k\bar{u}_j^{n+1} + C_{k-1}$$

Then we have the following derivations

$$(u_{j+\frac{1}{2}}^{n+1})^- = (u_j^{n+1},\delta_j^k)_j = (\hat{u}_j^{n+1},\hat{\delta}^k)_{\hat{I}} = D_0 = 2\lambda_j D_1 + C_0$$

$$= (2\lambda_j)^2 D_2 + 2\lambda_j C_1 + C_0 = \cdots = (2\lambda_j)^l D_l + \sum_{i=0}^{l-1} C_i (2\lambda_j)^i = \cdots$$

$$= (2\lambda_j)^{k-1}[4\lambda_j \beta_k^k \bar{u}_j^{n+1} + C_{k-1}] + \sum_{i=0}^{k-2} (2\lambda_j)^i C_i$$

$$= 2(2\lambda_j)^k \beta_k^k \bar{u}_j^{n+1} + \sum_{i=0}^{k-1} (2\lambda_j)^i C_i$$

After plugging in the definition of C_i and after some manipulation, we obtain

$$\left[1 + \sum_{i=0}^{k-1} (2\lambda_j)^{i+1} (\alpha_i^k - \beta_i^k)\right] (u_j^{n+1})^-$$

$$= \left[2 \sum_{i=0}^{k} (2\lambda_j)^i \beta_i^k\right] \bar{u}_j^{n+1} + \sum_{i=0}^{k-1} (2\lambda_j)^i (\hat{u}_j^n, (\hat{\delta}^k)^{(i)} - \beta_i^k)_{\hat{I}}$$

□

For the parameters $\{\beta_j^k, \alpha_j^k\}_{j=1}^N$, we have the following lemma, which will be useful in the proof of Proposition 3.2.1.

Lemma 3.2.5. *For any $k \geq 0$, the following results hold*

1. $\alpha_i^k > \beta_i^k$ *for* $i = 0, \cdots, k-1$ *and hence* $\sigma_j^k > 0$ *for any* j.
2. $\beta_{k-2i}^k > 0$, $\beta_{k-2i-1}^k < 0$, *for* $i = 0, \cdots, \lfloor k/2 \rfloor$.
3. $\beta_{k-2i}^k + \beta_{k-2i-1}^k > 0$, *for* $i = 0, \cdots, \lfloor k/2 \rfloor$.

Proof. The first statement is a direct conclusion of the third property of the discrete delta function in Lemma 3.2.3.

For the second statement, let us first derive an explicit formula for β_i^k. By the

Rodrigues' formula in Lemma 3.2.2 we have

$$p_l(x) = \frac{1}{2^l l!}\frac{d^l}{dx^l}(x^2-1)^l = \frac{1}{2^l l!}\frac{d^l}{dx^l}\left[(x-1)^l(x+1)^l\right]$$

Then by the Leibniz's rule, it is easy to obtain for $i \le l$,

$$\begin{aligned} p_l^{(i)}(-1) &= \frac{1}{2^l l!}\binom{i+l}{l} l!\left[(x-1)^l\right]^{(i)}|_{x=-1} = \frac{1}{2^l}\frac{(i+l)!}{i!l!}\frac{l!}{(l-i)!}(-2)^{l-i} \\ &= \frac{(-2)^{-i}}{i!}\frac{(i+l)!}{(l-i)!}(-1)^l \end{aligned}$$

and hence we have

$$\begin{aligned} \beta_i^k &= \frac{1}{2}\sum_{l=i}^{k}(2l+1)\frac{(-2)^{-i}}{i!}\frac{(l+i)!}{(l-i)!}(-1)^l \\ &= \frac{1}{2^{i+1}i!}\sum_{l=i}^{k}(2l+1)\frac{(l+i)!}{(l-i)!}(-1)^{l-i} = \frac{1}{C_i}\sum_{l=i}^{k}\gamma_i^l \end{aligned}$$

where $C_i = 2^{i+1}i!$ and $\gamma_i^l = (2l+1)\frac{(l+i)!}{(l-i)!}(-1)^{l-i}$.

For γ_i^l, we have

$$|\gamma_i^{l+1}| = (2l+3)\frac{(l+i+1)!}{(l+1-i)!} = \frac{2l+3}{2l+1}\frac{l+i+1}{l-i+1}|\gamma_i^l| > |\gamma_i^l|.$$

Next for $j = 0, \cdots, \lfloor k/2 \rfloor$, let us consider β_{k-2j}^k. If we replace i with $k-2j$, we obtain

$$C_{k-2j}\beta_{k-2j}^k = (|\gamma_{k-2j}^k| - |\gamma_{k-2j}^{k-1}|) + \cdots + (|\gamma_{k-2j}^{k-2j-2}| - |\gamma_{k-2j}^{k-2j-1}) + |\gamma_{k-2j}^{k-2j}| > 0. \quad (3.21)$$

For β_{k-2j-1}^k, we have

$$C_{k-2j-1}\beta_{k-2j-1}^k = -[(|\gamma_{k-2j-1}^k| - |\gamma_{k-2j-1}^{k-1}|) + \cdots + (|\gamma_{k-2j-1}^{k-2j-2}| - |\gamma_{k-2j-1}^{k-2j-1}|)] < 0. \quad (3.22)$$

Therefore, we can conclude that $\beta^k_{k-2j} > 0$ and $\beta^k_{k-2j-1} < 0$.

For the last statement, let us consider the sum $\beta^k_{k-2j} + \beta^k_{k-2j-1}$. To this end, first for general γ^l_j, let us consider the following expression

$$\tau^l_j := \frac{1}{2j}(|\gamma^l_j| - |\gamma^{l-1}_j|) - |\gamma^l_{j-1}| + |\gamma^{l-1}_{j-1}|$$

If we plug the definition of γ^l_j in and after direct calculation, we obtain

$$\tau^l_j = \frac{(l+j-2)!}{j(l-j+1)!}\left[(l^2-j^2)(2j+1) + 2j - 1\right] \geq 0.$$

If we combine (3.21) and (3.22) together we would obtain,

$$C_{k-2j-1}(\beta^k_{k-2j} + \beta^k_{k-2j-1}) = \sum_{i=0}^{j+1} \tau^{k-2i}_{k-2j} + |\gamma^{k-2j}_{k-2j}| > 0$$

which implies the desired conclusion. □

With Lemma 3.2.4 and the equation (3.15), we can obtain the following cell average equation.

$$T\bar{\mathbf{u}}^{n+1} = \mathcal{L}(\mathbf{u}^n) \tag{3.23}$$

where

$$T = \begin{pmatrix} \frac{\xi^k_1}{\sigma^k_1} + \frac{1}{\lambda_1} & 0 & \cdots & -\frac{\xi^k_N}{\sigma^k_N} \\ -\frac{\xi^k_1}{\sigma^k_1} & \frac{\xi^k_2}{\sigma^k_2} + \frac{1}{\lambda_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & -\frac{\xi^k_{N-1}}{\sigma^k_{N-1}} & \frac{\xi^k_N}{\sigma^k_N} + \frac{1}{\lambda_N} \end{pmatrix}$$

and

$$(\mathcal{L}(\mathbf{u}^n))_j = \left(\hat{u}^n_j, \frac{1}{2\lambda_j} - \frac{g^k_j}{\sigma^k_j}\right)_{\hat{I}} + \left(\hat{u}^n_{j-1}, \frac{g^k_{j-1}}{\sigma^k_{j-1}}\right)_{\hat{I}}, \quad j = 1, \ldots, N.$$

A set of sufficient conditions to make the cell average $\bar{u}_j^{n+1}$ positive for any $j = 1, \cdots, N$ are the following

Condition I T is an M-matrix, or by Corollary 3.2.1 and Lemma 3.2.5,

$$\xi_j^k > 0, \quad \forall j = 1, \cdots, N \tag{3.24}$$

Condition II $\mathcal{L}(\mathbf{u}^n)$ is positive, or

$$(\hat{u}_j^n, 1/2\lambda_j - g_j^k/\sigma_j^k)_{\hat{I}} + (\hat{u}_{j-1}^n, g_{j-1}^k/\sigma_{j-1}^k)_{\hat{I}} > 0, \quad j = 1, \cdots, N \tag{3.25}$$

By Lemma 3.2.3, it is easy to obtain $g_j^k(x) < \frac{\sigma_j^k}{2\lambda_j}, \forall x \in \hat{I}$, for any k and j and hence the first term in (3.25) is always positive. Since $\hat{u}_j^n$ and $\hat{u}_{j-1}^n$ can be any independent positive polynomials, the condition (3.25) is further reduced to

$$(v, g_j^k)_{\hat{I}} \geq 0, \quad \forall v \in P^k(\hat{I}) \text{ and } v \geq 0 \tag{3.26}$$

In summary, if we set $F_k(\lambda, x) = \sum_{i=0}^k (2\lambda)^i (\hat{\delta}^k)^{(i)}(x)$ and then $\xi_j^k = F_k(\lambda_j, -1)$, $g_j^k = F_k(\lambda_j, x) - F_k(\lambda_j, -1)$, sufficient conditions (3.24) and (3.25) actually require that the CFL number λ_j satisfy

$$F_k(\lambda_j, -1) \geq 0 \tag{3.27}$$

$$(F_k(\lambda_j, \cdot) - F_k(\lambda_j, -1), v(\cdot))_{\hat{I}} \geq 0, \quad \forall v \in P^k(\hat{I}) \text{ and } v \geq 0. \tag{3.28}$$

The following theorem says that in order to make these two conditions hold simultaneously, the CFL number λ_j can not be arbitrarily small.

Theorem 3.2.3. *When λ_j is small, conditions (3.27) and (3.28) can not hold at the same time. More specifically, we have the following two cases:*

1. *When the polynomial degree k is odd, there exists $\eta_1^k > 0$ such that when $\lambda_j < \eta_1^k$, the condition (3.27) does not hold.*

2. *When $k \geq 2$ is even, there exists $\eta_2^k > 0$ such that when $\lambda_j < \eta_2^k$, the second condition (3.28) does not hold.*

Proof. When k is odd, $F_k(\lambda_j, -1) = \sum_{i=0}^{k} \beta_i^k (2\lambda_j)^i$ is a polynomial of odd degree. By Lemma 3.2.5, we have the leading coefficient $\beta_k^k > 0$ and hence $F_k(\lambda_j, -1) > 0$ when λ_j is large. On the other hand, when $\lambda_j = 0$, $F_k(0, -1) = \beta_0^k < 0$, again by Lemma 3.2.5. Therefore, the polynomial $F_k(\lambda_j, -1)$ must have at least one and at most k positive roots. If we take η_1^k to be the smallest one, we would have the first statement.

When $k \geq 2$ is even, in (3.28) take $v = 1$ and we obtain

$$\begin{aligned}(F_k(\lambda_j, x) - F_k(\lambda_j, -1), 1)_{\hat{I}} \\ = \sum_{i=0}^{k-1} (2\lambda_j)^i \left[(\hat{\delta}^k)^{(i-1)}(1) - (\hat{\delta}^k)^{(i-1)}(-1) - 2(\hat{\delta}^k)^{(i)}(-1) \right] \geq 0\end{aligned}$$

where $(\hat{\delta}^k)^{(-1)}(1) - (\hat{\delta}^k)^{(-1)}(-1) := \int_{\hat{I}} \hat{\delta}^k(x)\, dx$. To simplify the notation, let us set $y = 2\lambda_j$ and set

$$G_k(y) = \sum_{i=0}^{k-1} y^i \left[(\hat{\delta}^k)^{(i-1)}(1) - (\hat{\delta}^k)^{(i-1)}(-1) - 2(\hat{\delta}^k)^{(i)}(-1) \right].$$

Since $k - 1$ is odd, again, we would like to show it has positive real roots. First, when $y = 0$, we have

$$G_k(0) = \int_{\hat{I}} \hat{\delta}^k(x)\, dx - 2\hat{\delta}^k(-1) = 1 - \frac{k+1}{2} = \frac{1-k}{2} \leq -\frac{1}{2}$$

Next, let us check the leading coefficient of $G_k(y)$. Since $(\hat{\delta}^k)^{(k)} = \beta_k^k$, we have $(\hat{\delta}^k)^{(k-2)} = \frac{1}{2}\beta_k^k x^2 + C_1 x + C_2$, where C_1 and C_2 are constants. As a consequence, we have

$$\alpha_{k-2}^k = \frac{1}{2}\beta_k^k + C_1 + C_2, \quad \beta_{k-2}^k = \frac{1}{2}\beta_k^k - C_1 + C_2, \quad \beta_{k-1}^k = -\beta_k^k + C_1$$

and hence the leading coefficient of $G_k(y)$ satisfies

$$\begin{aligned}(\hat{\delta}^k)^{(k-2)}(1) - (\hat{\delta}^k)^{(k-2)}(-1) - 2(\hat{\delta}^k)^{(k-1)}(-1)\\ = \alpha_{k-2}^k - \beta_{k-2}^k - 2\beta_{k-1}^k = 2\beta_k^k > 0\end{aligned}$$

Therefore, the odd-degree polynomial $G_k(y)$ must have at least one and at most $k-1$ positive real roots. If we take η_2^k to be the smallest one, we can conclude the second statement. □

Theorem 3.2.3 indicates that unlike the situation for the DG method with Euler forward time discretization in [86], where an upper bound for the CFL number is sufficient for the cell average at the next time level to be positive, for the DG scheme with the Euler backward time discretization, the CFL number can not be arbitrarily small and an lower bound may be required. The following analysis confirms this statement.

If we re-examine the condition (3.28) and note that for fixed λ_j, $F_k(\lambda_j, x) \in P^k(\hat{I})$, the inner product in (3.28) can actually be approximated exactly by certain quadrature rules, say $\{(x^\alpha, \omega^\alpha)\}_{\alpha=1}^{N_q}$, where $\{x^\alpha\}$ are the abscissas in $\hat{I}$, $\{\omega^\alpha\}$ the weights and N_q is large enough such that the quadrature rule is exact for polynomials of degree $2k$. We denote $\{(x_j^\alpha, \omega_j^\alpha)\}_{\alpha=1}^{N_q}$ to be the transformed quadrature rule in I_j.

Then the condition (3.28) can be reduced to require

$$J_\alpha^k(\lambda_j) := F_k(\lambda_j, x^\alpha) - F_k(\lambda_j, -1) \geq 0, \quad \alpha = 1, \cdots, N_q \tag{3.29}$$

If we define $J_0^k(\lambda_j) = F_k(\lambda_j, -1)$, together with condition (3.27), we require the CFL number to make N_q+1 polynomials $\{J_\alpha^k(\lambda_j)\}_{\alpha=0}^{N_q}$ positive. The following result states that it suffices to require $\lambda_j \geq \frac{1}{2}$.

Proposition 3.2.1. *When* $\min_j \lambda_j \geq \frac{1}{2}$, *we have*

$$F_k(\lambda_j, -1) > 0, \tag{3.30}$$

$$F_k(\lambda_j, x) - F_k(\lambda_j, -1) > 0, \quad \forall x \in \hat{I} \tag{3.31}$$

Proof. First, let us consider (3.30). By the definition we have $F_k(\lambda_j, -1) = \sum_{i=0}^k (2\lambda_j)^k \beta_i^k$, which can be rewritten as

$$F_k(\lambda_j, -1) = (2\lambda_j)^{k-1}(2\lambda_j\beta_k^k + \beta_{k-1}^k) + \cdots + \begin{cases} (2\lambda_j\beta_2^k + \beta_1^k) + \beta_0^k, & \text{if k is even} \\ (2\lambda_j\beta_1^k + \beta_0^k), & \text{if k is odd} \end{cases} \tag{3.32}$$

When $\lambda_j > 1/2$, i.e., $2\lambda_j > 1$, then by Lemma 3.2.5 it is easy to see that each term in (3.32) is strictly positive and hence $F_k(\lambda_j, -1) > 0$.

For the condition (3.31), consider the ith derivative of F_k with respect to x.

$$\frac{\partial^i}{\partial x^i} F_k(\lambda_j, x) = \sum_{l=i}^k (2\lambda_j)^{l-i} (\hat{\delta}^k)^{(l)}(x), \quad i = 0, \ldots, k$$

When $i = k$, by Lemma 3.2.3, we have $\frac{\partial^k}{\partial x^k} F_k(\lambda_j, x) = \beta_k^k > 0$ and hence

$$\frac{\partial^{k-1}}{\partial x^{k-1}} F_k(\lambda_j, x) > \frac{\partial^{k-1}}{\partial x^{k-1}} F_k(\lambda_j, -1) = \beta_{k-1}^k + (2\lambda_j)\beta_k^k > 0, \forall x \in \hat{I}.$$

We have used the facts $2\lambda_j \geq 1$ and Lemma 3.2.5 to derive the last inequality. We can continue this procedure till $i = 0$ and obtain

$$F_k(\lambda_j, x) > F_k(\lambda_j, -1), \forall x \in \hat{I}$$

Therefore, in summary, when $\min_j \lambda_j \geq \frac{1}{2}$, we have conditions (3.30) and (3.31) hold and hence we can draw the conclusion. □

Consequently, when $\lambda_j \geq \frac{1}{2}$ all the polynomials $\{J_\alpha^k\}_{\alpha=0}^k$ are strictly positive. On the other hand by the proof of Theorem 3.2.3, at $\lambda_j = 0$ these polynomials can not be all nonnegative. This implies that if we denote $\mathcal{R}_k$ to be the set of all the positive roots of each polynomial J_α^k, i.e.,

$$\mathcal{R}_k = \{r \in \mathbb{R}^+ : J_\alpha^k(r) = 0 \text{ for some } \alpha = 0, \cdots, N_q\}$$

then we must have $\mathcal{R}_k \neq \emptyset$, $\mathcal{R}_k \in (0, \frac{1}{2})$ and $\mathcal{R}_k$ is finite. Then if we set

$$r_k = \max_{r \in \mathcal{R}_k} r, \tag{3.33}$$

we have the following theorem.

Theorem 3.2.4. *For the backward Euler DG scheme (3.14) with P^k element for linear equation (3.13), at time level n given $u_j^n(x_j^\alpha) \geq 0$ at the quadrature points $\{x_j^\alpha\}_{\alpha=1}^{N_q} \in I_j$ which give exact approximation for polynomials of degree $2k$, we have*

$\bar{u}_j^{n+1} \geq 0$, *for any* j *under the following CFL condition*

$$\min_j \lambda_j \geq r_k \tag{3.34}$$

where $r_k \in (0, \frac{1}{2})$ *is defined as in* (3.33).

Proof. When $\lambda_j \geq r_k$ by the definition of $\mathcal{R}_k$, we have (3.29) and (3.27) hold. Therefore T in (3.23) is an M-matrix and $\mathcal{L} \geq 0$. By the definition of M-matrix, we can conclude the result. □

Remark 3.2.2. *We only require* u_h^n *to be positive at quadrature points* $\{x_j^\alpha\}_{\alpha=1}^{N_q}$, *which is weaker than the condition* $u_h^n(x) \geq 0$, *for any* $x \in \Omega$ *in the Definition 3.2.1.*

Remark 3.2.3. *Even though the Theorem 3.2.4 is only proved for linear equations, numerical experiments suggest that for nonlinear equations, a lower bound for the CFL number is still necessary to make the cell average at the next time level positive.*

Remark 3.2.4. *The results also hold for problems with positive source terms and positive inflow boundary conditions.*

Remark 3.2.5. *The lower bound depends on the polynomial degree* k *as well as the quadrature rule we choose. For each* k *and fixed quadrature rule we can actually obtain the lower bound* r_k *by solving the positive roots of each* J_α^k. *In Table 3.1, we record the lower bounds for Legendre-Gauss-Lobbato (LGL) quadrature rule and the Legendre-Gauss (LG) rule respectively. We see that the LGL rule gives smaller lower bound. In practice we will limit the polynomial* u_j^n *to make it positive at least at the LGL points* $\{x_j^\alpha\}_{\alpha=1}^{N_q}$ *in each cell* I_j.

Remark 3.2.6. *The lower bounds in Table 3.1 are sharp for odd* k *and sufficient*

Table 3.1: Values of r_k for $k = 1, \cdots, 5$ and for Legendre-Gauss-Lobatto (LGL) and Legendre-Gauss quadrature rules respectively.

	LGL rule		LG rule	
k	N_q	r^k	N_q	r^k
1	3	0.333	2	0.333
2	4	0.262	3	0.344
3	5	0.177	4	0.177
4	6	0.177	5	0.212
5	7	0.121	6	0.121

for even k. If we start from the following initial condition,

$$u_0(x) = \begin{cases} 1, & \text{if } x \in I_M \\ 0, & \text{otherwise} \end{cases},$$

where $M = \arg\max_j h_j$. After one step we record the minimum cell averages for different odd k and $\lambda = r_k \pm 0.001$ in Table 3.2. We see that when λ is slightly smaller than the lower bound r_k, after one time step, at least one of the cell averages will become negative. If λ is larger than r_k, the average will be uniformly positive. In

Table 3.2: Minimum cell average after one time step for odd k.

k	$\lambda = r^k - 0.001$	$\min_j \bar{u}_j^1$	$\lambda = r^k + 0.001$	$\min_j \bar{u}_j^1$
1	0.332	-3.498 E-04	0.334	5.284 E-35
3	0.176	-4.177 E-05	0.178	7.941 E-46
5	0.120	-2.135 E-06	0.122	1.980 E-59

the case $k = 1$, the lower bound $r_1 = \frac{1}{3}$ conincides with the lower bound in Theorem 3.2.2 for uniform meshes.

For even k, we consider a different initial condition

$$u_0(x) = \begin{cases} \left(\frac{2(x-x_M)}{h_M} - 0.72\right)^k, & \text{if } x \in I_M \\ 0, & \text{otherwise} \end{cases}$$

The Table 3.1 shows the minimum cell averages for different k and λ. We see that the lower bound for the CFL number is still necessary for the positivity of $\bar{u}_j^1$ and r_k listed in Table 3.1 is sufficient.

Table 3.3: Minimum cell average after one time step for even k.

k	λ	$\min_j \bar{u}_j^1$	$\lambda = r^k$	$\min_j \bar{u}_j^1$
2	0.170	-1.022 E-03	0.266	1.221 E-28
4	0.120	-1.343 E-03	0.177	5.021 E-46

3.2.4 Scaling limiter

Once in each cell I_j, the cell average $\bar{u}_j^{n+1}$ is positive, we limit the whole polynomial $u_j^{n+1}(x)$ towards its cell average by utilizing the scaling limiter (2.12). In this case, we have $G = [0, \infty)$ and $\partial G = \{0\}$. And then, the limiter can be defined in the following way,

$$\tilde{u}_j^{n+1} = \theta_j[u_j^{n+1} - \bar{u}_j^{n+1}] + \bar{u}_j^{n+1}, \tag{3.35}$$

where

$$\theta_j = \begin{cases} \frac{\bar{u}_j^{n+1}}{\bar{u}_j^{n+1} - \min_{x \in I_j} u_j^{n+1}(x)}, & \text{if } \min_{x \in I_j} u_j^{n+1}(x) < 0 \\ 1, & \text{otherwise} \end{cases} \tag{3.36}$$

Remark 3.2.7. *In order to calculate the scaling parameter θ_j in (3.36) we need to calculate the minimum value of u_j^{n+1} in each cell I_j, which can be done efficiently up to $k = 5$ via the root formulas. For larger k, this calculation becomes expensive. But*

recall that in Theorem 3.2.4, we only require each polynomial u_j^{n+1} to be positive on LGL quadrature points $\{x_j^\alpha\}_{\alpha=1}^{N_q}$ and hence we can instead use the following scaling parameter

$$\tilde{\theta}_j = \begin{cases} \frac{\bar{u}_j^{n+1}}{\bar{u}_j^{n+1} - \min_\alpha u_j^{n+1}(x_j^\alpha)}, & \text{if } \min_\alpha u_j^{n+1}(x_j^\alpha) < 0 \\ 1, & \text{otherwise} \end{cases} \tag{3.37}$$

in the limiter. This is similar with the scaling limiter in [86] and the same argument can be conducted here to prove that the modified limiter does not kill the original high-order accuracy either.

3.2.5 Algorithm for scalar equations

Now, we can summarize the positivity-preserving algorithm for scalar equations as below.

1. At time level t^n, given $u_j^n(x)$ being positive at least on the LGL quadrature points $\{x_j^\alpha\}_{\alpha=1}^{N_q}$.

2. Choose CFL number large enough a priori or adaptively enlarge it in each time step until $\bar{u}_j^{n+1} \geq 0$, $\forall j$.

3. Apply the scaling limiter (3.35) with θ_j defined in (3.36) or (3.37) to u_j^{n+1} such that $u_j^{n+1}(x_j^\alpha) \geq \epsilon$, for any $\alpha = 1, \ldots, N_q$ and j, where ϵ is a small number to help get rid of the round-off effect. In the numerical examples, we take $\epsilon = 10^{-13}$.

3.3 Positivity-preserving backward Euler DG scheme for compressible Euler systems

Next, let us consider the following compressible Euler system for ideal gas

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}, \quad t \geq 0, \quad x \in [0,1]. \tag{3.38}$$

with

$$\mathbf{u} = (\rho, m, E)^T, \quad \mathbf{f} = (m, \rho u^2 + p, (E+p)u)^T$$

and

$$m = \rho u, \quad E = \frac{1}{2}\rho u^2 + \rho e, \quad p = (\gamma - 1)\rho e,$$

where ρ is the density, u the velocity, m the momentum, p the pressure, E the total energy and e is the internal energy. The constant $\gamma > 1$ is the ratio of specific heats. The Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ has the following three eigenvalues

$$\zeta_1 = u - c, \quad \zeta_2 = u, \quad \zeta_3 = u + c$$

where $c = \sqrt{\gamma p/\rho}$ is the sound speed.

The backward Euler DG scheme for (3.38) is defined to seek $\mathbf{u}_h^{n+1} \in \mathbf{V}_h$ such that in each cell I_j we have

$$(\mathbf{u}_h^{n+1}, \mathbf{v})_j - \Delta t L_j(\mathbf{u}_h^{n+1}, \mathbf{v}) = (\mathbf{u}_h^n, \mathbf{v})_j, \quad \forall \mathbf{v} \in \mathbf{V}_h \tag{3.39}$$

where L_j is defined as in (2.5) and the parameter in the global Lax-Friedrichs flux is set to be $\alpha = \||c| + |v|\|_\infty$.

The physical solution lies in the following admissible set

$$G = \left\{ \mathbf{u} = (\rho, m, E)^T : \rho \geq 0, p = (\gamma - 1)\left(E - \frac{1}{2}\frac{m^2}{\rho}\right) \geq 0 \right\} \tag{3.40}$$

It can be verified that G is a convex set [87]. If we let $\mathbf{u}_h^n$ to denote the DG approximation at time level n, then the positivity-preserving DG scheme for compressible Euler system is defined as below.

Definition 3.3.1. *A DG scheme for compressible Euler system* (3.38) *is positivity-preserving if at time level n given $\mathbf{u}_h^n(x) \in G$ for all $x \in \Omega$, then at the next time level $(n+1)$, we have $\mathbf{u}_h^{n+1}(x) \in G$ for all $x \in \Omega$.*

Also, for the implementation reason, we have the relaxed version.

Definition 3.3.2 (Relaxed version)**.** *A DG scheme for* (3.38) *is defined to be positivity preserving if in each subinterval I_j, given $\mathbf{u}_j^n(x_j^\alpha) \in G$, then we have $\mathbf{u}_j^{n+1}(x_j^\alpha) \in G$, where $\{x_j^\alpha\}$ is a set of selected quadrature points in cell I_j for each $j = 1, \cdots, N$.*

We design the positivity-preserving DG scheme by extending the general framework in Section 2.2. Unlike the scalar case, for system, we basically can not prove anything. But the analysis for the linear scalar equation suggests that starting from $\mathbf{u}_j^n(x) \in G$, in order to have the cell average $\bar{\mathbf{u}}_j^{n+1} \in G$, a lower bound for the CFL number may be required. Therefore, we formulate the algorithm as follows. We stress on the applicability of the proposed algorithm rather than its theoretical justification.

1. At time level t^n, in each cell I_j, given $\mathbf{u}_j^n(x_j^\alpha) \in G$ at LGL quadrature points $\{x_j^\alpha\}_{\alpha=1}^{N_q}$.

2. Choose a large CFL number a priori or adaptively enlarge it in each time step until $\bar{\mathbf{u}}_j^{n+1} \in G$ for any j.

3. In each cell I_j, apply the following scaling limiter to the first component of $\mathbf{u}_j^{n+1}$ to obtain $\rho_j^{n+1}(x) \geq 0$ at LGL quadrature points $\{x_j^\alpha\}_{\alpha=1}^{N_q}$

$$\tilde{\rho}_j^{n+1} = \theta_1(\rho_j^{n+1} - \bar{\rho}_j^{n+1}) + \bar{\rho}_j^{n+1}$$

where

$$\theta_1 = \begin{cases} \frac{\bar{\rho}_j^{n+1}}{\bar{\rho}_j^{n+1} - \min_\alpha \rho_j^{n+1}(x_j^\alpha)}, & \text{if } \min_\alpha \rho_j^{n+1}(x_j^\alpha) < 0 \\ 1, & \text{otherwise} \end{cases}$$

Denote the modified polynomial by $\tilde{\mathbf{u}}_j^{n+1}$.

4. In each cell I_j, apply the scaling limiter again to the whole modified polynomial $\tilde{\mathbf{u}}_j^{n+1}$ such that $p(x_j^\alpha) \geq 0$ for each α as below.

$$\widetilde{\widetilde{\mathbf{u}}}_j^{n+1} = \theta_2(\widetilde{\mathbf{u}}_j^{n+1} - \overline{\widetilde{\mathbf{u}}}_j^{n+1}) + \overline{\widetilde{\mathbf{u}}}_j^{n+1}$$

where

$$\theta_2 = \begin{cases} \frac{\bar{p}_j^{n+1}}{\bar{p}_j^{n+1} - \min_\alpha \tilde{p}_j^{n+1}(x_j^\alpha)}, & \text{if } \min_\alpha \tilde{p}_j^{n+1}(x_j^\alpha) < 0 \\ 1, & \text{otherwise} \end{cases}$$

and the pressure average is defined by $\bar{p}_j^{n+1} = p(\bar{\tilde{\mathbf{u}}}_j^{n+1})$.

Remark 3.3.1. *The modified polynomial* $\widetilde{\widetilde{\mathbf{u}}}_j^{n+1} \in G$ *implemented in this way in general does not lies on* ∂G*. But this is a more robust version then the original one proposed in [87]. It was first developed in [72] and was further applied for the RHD system in [55]. The preservation of the high-order accuracy was verified via truncation error analysis [72] and numerical examples in [72, 55].*

3.4 Positivity-preserving Crank-Nicolson DG scheme

The Crank-Nicolson P^1-DG scheme for the conservation law (3.1) in each cell I_j, takes the following form

$$\begin{aligned}(w_h^n, v)_j &= (u_h^n, v)_j + \frac{\Delta t}{2} L_j(u_h^n, v), \quad \forall v \in V_h \\ (u_h^{n+1}, v)_j - \frac{\Delta t}{2} L_j(u_h^{n+1}, v) &= (w_h^n, v)_j, \quad \forall v \in V_h\end{aligned} \tag{3.41}$$

In this section we only consider P^1 element. We want to use the framework in Section 2.2 to construct positivity preserving DG scheme. The key is still to derive a CFL condition for the linear problem (3.13) under which given $u_h^n(x) \geq 0$, $\forall x$, we have $\bar{u}_j^{n+1} \geq 0$, for any j.

Note that in (3.41), the first step is a forward Euler scheme and the second step is a backward Euler scheme. A naive thinking is to first use the technique for the positivity-preserving forward Euler scheme to derive a CFL condition, under which $w_h^n \geq 0$ at least at selected quadrature points. Then once $w_h^n(x) \geq 0$, we can use the result for the backward Euler scheme in the last section to obtain $\bar{u}_j^{n+1} \geq 0$. Unfortunately, the following analysis shows the CFL condition for the forward Euler step does not exist even for P^1 element.

Let us consider any $x_j^\alpha \in I_j$ and set $x^\alpha = T_j(x_j^\alpha) \in \hat{I}$. Set the discrete delta function in $P^1(I_j)$ at x_j^α as δ_α^1 and the corresponding one in $P^1(\hat{I})$ as $\hat{\delta}_\alpha^1$, which by definition takes the following form

$$\hat{\delta}_\alpha^1 = \frac{1}{2} + \frac{3x_\alpha}{2} x$$

If we take $v = \delta_\alpha^1$ in the forward Euler scheme in (3.41) and after a change of variable,

we obtain

$$w_\alpha^n = u_\alpha^n + 2\lambda_j(\hat{u}_h^n, \hat{\delta}_\alpha^1))_{\hat{I}} - 2\lambda_j[(u_{j+\frac{1}{2}}^n)^-\hat{\delta}_\alpha^1(1) - (u_{j-\frac{1}{2}}^n)^-\hat{\delta}_\alpha^1(-1)]$$

where $w_\alpha^n = \hat{w}_j^n(x_\alpha) = w_j^n(x_j^\alpha)$ and the same for u_α^n. After plugging the definition of $\hat{\delta}_\alpha^1$ into the scheme, we obtain

$$\begin{aligned} w_\alpha^n &= u_\alpha^n + 6\lambda_j x_\alpha \bar{u}_j^n - \lambda_j u_h^n(x_{j+\frac{1}{2}}^-)(1+3x_\alpha) + \lambda_j u_h^n(x_{j-\frac{1}{2}}^-)(1-3x_\alpha) \\ &= [u_\alpha^n - \lambda_j u_j^n(x_{j+\frac{1}{2}}) + 3\lambda_j x_\alpha u_j^n(x_{j-\frac{1}{2}})] + \lambda_j(1-3x_\alpha)u_{j-1}^n(x_{j-\frac{1}{2}}) \end{aligned}$$

Since u_j^n and u_{j-1}^n are independent positive polynomials, if $x_\alpha > \frac{1}{3}$, the we can set $u_{j-1}^n(x_{j-\frac{1}{2}})$ arbitrarily large and hence there exists no λ_j making w_α^n positive. Recall in Table 3.1, for $k = 1$, we need to require w_j^n to be positive at at least three LGL points and two GL points, which all include abscissas greater than $\frac{1}{3}$. Therefore, we can not combine the results for the positivity-preserving forward and backward Euler methods and extend it to the Crank-Nicolson method.

3.4.1 CFL condition for P^1-DG on uniform meshes

If we further assume the mesh is uniform, by brutal force as for the backward Euler method, we can get some necessary and sufficient results for the Crank-Nicolson method.

Similar with (3.7), the Crank-Nicolson time discretization for the linear DG method for the linear problem (3.13), which, in each cell I_j can be written in the

following matrix form.

$$M^j \frac{\mathbf{c}_j^{n+1} - \mathbf{c}_j^n}{\Delta t} = \frac{1}{2}\left[(S^j \mathbf{c}_j^{n+1} - B_1^j \mathbf{c}_j^{n+1} + B_2^j \mathbf{c}_{j-1}^{n+1}) + (S^j \mathbf{c}_j^n - B_1^j \mathbf{c}_j^n + B_2^j \mathbf{c}_{j-1}^n)\right]$$

which is equivalent to

$$\left(M^j - \frac{\Delta t}{2} S^j + \frac{\Delta t}{2} B_1^j\right) \mathbf{c}_j^{n+1} - \frac{\Delta t}{2} B_2^j \mathbf{c}_{j-1}^{n+1} = M_j \mathbf{c}_j^n + \frac{\Delta t}{2}(S^j \mathbf{c}_j^n - B_1^j \mathbf{c}_j^n + B_2^j \mathbf{c}_{j-1}^n)$$

After rescaling, we have

$$(M - \lambda_j S + \lambda_j B_1)\mathbf{c}_j^{n+1} - \lambda_j B_2 \mathbf{c}_{j-1}^{n+1} = (M + \lambda_j S - \lambda_j B_1)\mathbf{c}_j^n + \lambda_j B_2 \mathbf{c}_{j-1}^n \tag{3.42}$$

or in the compact form

$$P\mathbf{c}^{n+1} = \tilde{P}\mathbf{c}^n$$

where for uniform mesh with $\lambda := \lambda_1 = \cdots = \lambda_N$, we denote $P = \text{bcirc}\{M - \lambda S + \lambda B_1, O, \cdots, O, -\lambda B_2\}$ and $\tilde{P} = \text{bcirc}\{M + \lambda S - \lambda B_1, O, \cdots, O, \lambda B_2\}$. Then we have the following cell average equation

$$\bar{\mathbf{u}}^{n+1} = \Theta P^{-1} \tilde{P} \mathbf{c}^n \tag{3.43}$$

For the linear element, we have the following result.

Theorem 3.4.1. *For the Crank-Nicolson DG method with P^1 element for the linear problem (3.13), given $u_h^n(x) > 0$ for any $x \in \Omega$, we have $\bar{u}_j^{n+1} > 0$ for any j, if and only if*

$$\lambda \in \left(\frac{2}{3}, \lambda_N\right) \tag{3.44}$$

with $\lambda_N \searrow \sqrt{\frac{2}{3}}$ as $N \to \infty$.

Proof. Note that if we replace λ with $\frac{\lambda}{2}$ in the scheme (3.8) the inverse of P takes exactly the same form as R^{-1}, i.e., $P^{-1} = \text{bcirc}\{D_1, D_2, \cdots, D_N\}$, with D_i defined in (3.11). If we set

$$\tilde{A} = M + \lambda(S - B_1), \quad \tilde{B} = \lambda B_2$$

then we have

$$P^{-1}\tilde{P} = \text{bcirc}\{D_1\tilde{A} + D_2\tilde{B}, D_2\tilde{A} + D_3\tilde{B}, \cdots, D_N\tilde{A} + D_1\tilde{B}\}.$$

When $i = 2, \cdots, N-1$, we have

$$\mathbf{d}_i = \vec{\theta}(D_i\tilde{A} + D_{i+1}\tilde{B}) = \left(\frac{\alpha\lambda(9\lambda^2 - 4)}{3\lambda^2 + 4\lambda + 2}, 3\lambda\alpha a^{N-i}(a+b)\right)$$

where $\alpha = \det C$ is always positive.

Then

$$\mathbf{d}_i \geq 0, \forall i = 2, \cdots, N-1 \iff 9\lambda^2 - 4 > 0 \,\&\, a^{N-i} > 0, \forall i \iff \lambda > \frac{2}{3}$$

When $i = N$, similar procedure gives

$$\mathbf{d}_N = \left(\frac{6\lambda(3\lambda+1)(\lambda+4)}{3\lambda^2 + 4\lambda + 2}, 9\alpha\lambda(a+b)\right) > 0, \quad \forall\lambda > 0.$$

At last, let us examine the case when $i = 1$. In this case, direct calculation gives

$$\mathbf{d}_1 = (p_N(\lambda), q_N(\lambda))$$

where

$$p_N(\lambda) = -\frac{3}{2}\lambda^2(1 - a^{N-2}b) + \lambda a^{N-2}(a - b) + 1$$
$$q_N(\lambda) = -\frac{3}{2}\lambda^2(1 - a^{N-1}) + \lambda(2 + a^{N-1}) + 1$$

Let us first examine the first entry $p_N(\lambda)$. We claim it has the following three properties.

(P1) $p_N(\lambda)$ is strictly decreasing and has unique positive root $\lambda_N > 0$.

(P2) For fixed $\lambda > \frac{2}{3}$, we have $p_{N+1}(\lambda) < p_N(\lambda)$.

(P3) For each N, $\lambda_N > \frac{2}{3}$.

The proof for these statements are postponed after the proof of this theorem. With these statements in hand, we want to show $\lambda_N \searrow \sqrt{\frac{2}{3}}$. First of all, since $|a| < 1$, for fixed $\lambda > 0$, it is easy to see that as $N \to \infty$

$$p_N(\lambda) \to p_0(\lambda) = -\frac{3}{2}\lambda^2 + 1$$

which has positive root $\lambda_0 = \sqrt{\frac{2}{3}}$. By (P2), we have for any $M > N$, $p_N(\lambda_0) > p_M(\lambda_0)$. Then by letting $M \to \infty$, we will have $p_N(\lambda_0) > p_0(\lambda_0) = 0$. Then by (P1), we have $\lambda_N > \lambda_0$. Therefore, $\{\lambda_N\}$ is a decreasing sequence with lower bound λ_0 and hence has limit $\tilde{\lambda}_0 = \lim_{N\to\infty} \lambda_N$. Next, we want to show $\tilde{\lambda}_0 = \lambda_0$. Suppose $\tilde{\lambda}_0 > \lambda_0$, then $p_0(\tilde{\lambda}_0) < 0$. Since $p_N(\tilde{\lambda}_0) \to p_0(\tilde{\lambda}_0)$ there must exist N_0 such that $p_{N_0}(\tilde{\lambda}_0) < 0$ and hence $\lambda_{N_0} < \tilde{\lambda}_0$ which is a contradiction. Therefore, $\lambda_N \searrow \lambda_0 = \sqrt{\frac{2}{3}}$.

Then let us look at $q_N(\lambda)$. Similar with p_N, one can show the following results.

(Q1) When $\lambda > \frac{2}{3}$, $q_N(\lambda) > q_{N+1}(\lambda)$.

(Q2) When $\lambda > \frac{2}{3}$ and $N \geq 3$, we have $q_N'(\lambda) < 0$.

(Q3) When $\lambda > \frac{2}{3}$, q_N has unique positive root r_N.

Then by the same token, one can show that $r_N \searrow r_0 = (2+\sqrt{10})/3$.

In summary, for the Crank-Nicolson DG with linear element, the scheme is monotone if and only if

$$\lambda \in \left(\frac{2}{3}, \lambda_N\right)$$

with $\lambda_N \searrow \sqrt{\frac{2}{3}}$. □

Remark 3.4.1. *The sharpness of the bounds can be verified by considering the following two examples. The first one is for testing the upper bound. Consider the following initial condition.*

$$u_h^0 = \begin{cases} (x - x_{j-\frac{1}{2}})/h, & \text{if } x \in I_j \\ 0, & \text{otherwise} \end{cases} \tag{3.45}$$

After one step, we can check the $\min_j \bar{u}_j^1$. *In Table 3.4, we list the minimum cell average for* $\lambda = \sqrt{\frac{2}{3}} - 10^{-9}$ *and* $\lambda = \sqrt{\frac{2}{3}} + 10^{-9}$ *respectively. We see that for small* N, *when* $\lambda_N > \sqrt{\frac{2}{3}}$, *we have the positive minimum cell averages after one time step. As we increase the number of cells* N, *the upper bound converges to* $\sqrt{2/3}$ *and hence the minimum cell average becomes negative. This verifies the dependence of the upper bound on the cell number* N. *But if we take* λ *slightly smaller than* $\sqrt{\frac{2}{3}}$ *the cell average will always be positive.*

Table 3.4: Values of $\min_j \bar{u}_j^1$ with initial condition (3.45) for $N = 4, 8, 16, 32$ and different λ.

	$\min_j \bar{u}_j^1$	
N	$\lambda = \sqrt{\frac{2}{3}} + 10^{-9}$	$\lambda = \sqrt{\frac{2}{3}} - 10^{-9}$
4	6.12 e-5	6.12 e-5
8	6.11 e-11	7.35 e-10
16	-3.37 e-10	3.34 e-19
32	-3.37 e-10	5.99 e-40

For the lower bound, let us consider the following initial condition.

$$u_h^0 = \begin{cases} (-x + x_{j+\frac{1}{2}})/h, & \text{if } x \in I_j \\ 0, & \text{otherwise} \end{cases} \tag{3.46}$$

In Table 3.4, we list the same results for $\lambda = \frac{2}{3} \pm 10^{-9}$. We see that when λ is slightly smaller than the lower bound, the minimal cell average is negative. However, when λ is slightly larger than $\frac{2}{3}$, as expected, the cell average is always positive. And the lower bound is independent of the cell number N.

Table 3.5: Values of $\min_j \bar{u}_j^1$ with initial condition (3.46) for $N = 4, 8, 16, 32$ and different λ.

	$\min_j \bar{u}_j^1$	
N	$\lambda = \frac{2}{3} - 10^{-9}$	$\lambda = \frac{2}{3} + 10^{-9}$
4	-1.48 e-10	8.53 e-18
8	-1.48 e-10	9.51 e-28
16	-1.48 e-10	2.54 e-119
32	-1.48 e-10	9.51 e-28

Remark 3.4.2. *The CFL condition derived in 3.4.1 is very stringent, which makes tthe positivity-preserving Crank-Nicolsono DG methods impractical. In the numerical experiments, we only implement the Crank-Nicolson P^1 DG method for accuracy test purpose, without any other applications.*

Remark 3.4.3. *The conclusion for Crank-Nicolson is surprising at the first look. But if we recall the relationship between C-N and forward and backward Euler methods, this actually is a very natural result. Here is the heuristic argument. The Crank-Nicolson method for* $u_t = L(u)$ *can be written as*

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

which is actually a convex combination of an Euler forward and an Euler backward methods. Then

$$\begin{aligned} u^{(1)} \geq 0 \text{ requires } \lambda \leq r_N \\ u^{n+1} \geq 0 \text{ requires } \frac{\lambda}{2} > \frac{1}{3} \iff \lambda > \frac{2}{3} \end{aligned}$$

and hence we have to require both a lower and a upper bound for the Crank-Nicolson to be positivity-preserving.

Proof of (P1) Let us prove that p_N is a strictly decreasing function for each N. Consider the derivative of it. Direct calculation gives

$$p_N'(\lambda) = \frac{3\lambda\alpha^2}{\beta^N} g_N(\lambda) + 2,$$

where

$$\begin{aligned} \alpha = 3\lambda^2 - 2\lambda, \quad \beta = 3\lambda^2 + 4\lambda, \\ g_N(\lambda) = \alpha^{N-2}\left[9\lambda^4 + (9N+6)\lambda^3 + (24N-20)\lambda^2 + (10N-16)\lambda - 4N\right] - \beta^N. \end{aligned}$$

We want to show $g_N(\lambda) < 0$ for any $\lambda > 0$ and N. First of all, it is very easy to check the result for $N = 1, 2, 3$ and hence we assume $N \geq 4$. And discuss in the following several situations.

First, when $\alpha^{N-2} < 0$, i.e., when $\lambda < \frac{2}{3}$ and N is odd, since in this case, $|\alpha| < 1$, we have

$$g_N \leq 4N|\alpha|^{N-2} - \beta^N \leq 4N - \beta^N \leq 4N - 2^N - 3^N\lambda^{2N} < 0, \tag{3.47}$$

for $N \geq 4$. Next, when $\alpha^{N-2} \in (0, 1)$, in this case we must have $\lambda < 1$ and hence

$$\begin{aligned} g_N &\leq 9\lambda^4 + (9N+6)\lambda^3 + (24N-20)\lambda^2 + (10N-16)\lambda - \beta^N \\ &\leq 9\lambda^4 + (9N+6)\lambda^3 + (24N-20)\lambda^2 - (4\lambda+2)^N \\ &= 9\lambda^4 + (9N+6)\lambda^3 + (24N-20)\lambda^2 + (10N-16)\lambda - \sum_{i=0}^{N} C_n^i (4\lambda)^i 2^{N-i} \\ &=: \sum_{i=0}^{N} c_i \lambda^i \end{aligned}$$

It is obvious that $c_i < 0$ for $i = 0$ and $i > 4$. Let us check for c_i when $i = 1, 2, 3, 4$ and for $N \geq 4$

1. When $i = 1$, $c_1 = 10N - 16 - N - N2^{N+1} < 0$.

2. When $i = 2$, we have

$$\begin{aligned} c_2 &= 24N - 20 - C_N^2 4^2 2^{N-2} = 24N - 20 - 2^{N+3}(N^2 - N) \\ &\leq 24N - 20 - 2^3(N^2 - N) \\ &= -8N^2 + 32N - 20 \\ &< 0, \end{aligned}$$

3. When $i = 3$, we have

$$\begin{aligned} c_3 &= 9N + 6 - C_N^3 2^{N+3} = 9N + 6 - 2^{N+3}(N^3 - 3N^2 + 2N) \\ &= -2^{N+3}(N^3 - 3N^2) - (2^{N+4}N - 9N - 6) \\ &< 0 \end{aligned}$$

4. For c_4, we have $c_4 = 9 - C_N^4 4^4 2^2 < 0$.

Therefore, $g_N < 0$ when $\alpha^{N-2} \in (0, 1)$.

At last, let us consider the case where $\alpha^{N-2} > 1$, i.e., when $\lambda > 1$. First, we have

$$\begin{aligned} g_N &\leq (3\lambda^2)^{N-2}[9\lambda^4 + (9N+6)\lambda^3 + (24N-20)\lambda^2 + (10N-16)\lambda] - (3\lambda^2 + 4\lambda)^N \\ &\leq 3^N\lambda^{2N} + 3^{N-1}(3N+2)\lambda^{2N-1} + 3^{N-2}(24N-20)\lambda^{2N-2} + 3^{N-2}(10N-16)\lambda^{2N-3} \\ &- \sum_{i=0}^{N} C_n^k 3^{N-i} 4^i \lambda^{2N-i} \\ &=: \sum_{i=0}^{N} d_i \lambda^i \end{aligned}$$

Again, we check d_{N-i} for $i = 0, 1, 2, 3$ and for $N \geq 4$.

1. Consider $i = 0$, $d_N = 3^N - 3^N = 0$.

2. For $i = 1$, we have

$$d_{N-1} = 3^{N-1}(3N+2) - 4N3^{N-1} = 3^{N-1}(3N + 2 - 4N) = -3^{N-1}(N-2) < 0.$$

3. For $i = 2$, we have

$$d_{N-2} = 3^{N-2}(-8N^2 + 32N - 20) < 0.$$

4. For $i = 3$, we have

$$\begin{aligned} d_{N-3} &= 3^{N-3}\left[3(10N-16) - \frac{32N(N-1)(N-2)}{3}\right] \\ &= 3^{N-4}[90N - 144 - 32N(N-1)(N-2)] \\ &\leq 3^{N-4}N(-32N^2 + 96N + 26) \\ &< 0 \end{aligned}$$

In conclusion, we have shown $g_N < 0$, which immediately implies $p'_N < 0$, for any $\lambda > 0$.

Proof of (P2) For fixed $\lambda > \frac{2}{3}$, we have

$$\begin{aligned} \frac{1}{a^{N-2}}(p_{N+1}(\lambda) - p_N(\lambda)) &= \frac{3}{2}b\,\lambda^2(a-1) + \lambda(b-1)(a-1) \\ &= (a-1)\left(\frac{3b\lambda^2}{2} + (a-b)\lambda\right) \\ &= (a-1)\frac{3\lambda^2(3\lambda-2)(\lambda+2)}{2\beta} \\ &< 0 \end{aligned}$$

since $|a| < 1$.

Proof of (P3) Because of (P1), it suffices to show $p_N(2/3) > 0$ and $p_N(2) < 0$. This is straightforward as below.

$$\begin{aligned} p_N(2/3) &= -\frac{3}{2}\left(\frac{2}{3}\right)_1^2 > 0 \\ p_N(2) &= -5 + \frac{48}{11}\left(\frac{4}{11}\right)^{N-2} \leq -5 + \frac{48}{11} < 0, \forall N \geq 2. \end{aligned}$$

3.5 Numerical experiments

In this section, we present numerical examples. First, we verify the accuracy of the Crank-Nicolson positivity-preserving P^1-DG method on linear problem. Then for the backward Euler DG method, we verify the high-order spacial accuracy of the proposed method by testing it on both linear and nonlinear steady-state problems. An acceleration of the convergence towards the steady state solution is also observed. Next, we test the methods on moving-shock problems. At last, examples for compressible Euler system will be presented. In all the examples, the domain is first uniformly decomposed and then each node is randomly perturbed up to 20% of the mesh size.

To further solve the nonlinear system (3.3) and (3.39), there have been many works on how to build efficient solvers, such as the work in [51, 50, 49]. But since our main focus is on the positivity preserving property rather than the efficiency of the nonlinear solver, we use the Newton method [16] for the nonlinear system up to accuracy 10^{-13}. For the robustness and accuracy reasons, in each Newton iteration, the Jacobian matrix is solved with the direct solver.

3.5.1 Accuracy tests

First, let us test the accuracy of the proposed method. The accuracy is first tested on the linear problem solved with the Crank-Nicolson P^1-DG method. And then we check the high-order spacial accuracy with the steady-state solution to both of the linear equation and the Burgers equation. For the steady-state simulation, we take $\Delta t = 10 \max_j h_j$ and march in time until $\|u_h^{n+1} - u_h^n\|_2 \leq 10^{-12}$.

Example 3.5.1.1 (Linear problem with Crank-Nicolson P^1-DG)**.** Consider the following problem.

$$u_t + u_x = 0, \quad x \in [0, 2\pi]$$
$$u_0(x) = \sin^4(x)$$

with periodic boundary condition. The exact solution is $u(x,t) = \sin^4(x-t)$. The equation is solved with P^1-DG with Crank-Nicolson time discretization. Uniform mesh is employed. As the Theorem 3.4.1 indicates, the positivity-preserving limiter will only work when $\lambda \in [\frac{2}{3}, \sqrt{\frac{2}{3}}]$. Therefore, we take $\lambda = 0.81$ in this example. In Table 3.6, we record numerical results at $T = 1$. The L^2 and L^∞ errors and the numerical orders of convergence as well as the minimal values at time T are presented. We see that without the positivity-preserving limiter, the minimum value are negative. With the limiter on, the positivity of the numerical solution is preserved and the numerical order of accuracy is maintained.

Table 3.6: Error table for Example 3.5.1.1 at $T = 1$ with $\lambda = 0.81$, uniform mesh.

	N	L^2 error		L^∞ error		$\min u_h$
no limiter	40	4.91 E-2	–	9.11 E-2	–	-3.48 E-2
	80	1.31 E-2	1.91	2.42 E-2	1.91	-7.93 E-3
	160	3.34 E-3	1.97	6.08 E-3	1.99	-1.37 E-3
	320	8.38 E-4	1.99	1.52 E-3	2.00	-2.29 E-4
with limiter	40	4.74 E-2	–	9.11 E-2	–	4.96 E-4
	80	1.34 E-2	1.82	2.63 E-2	1.79	2.10 E-5
	160	3.39 E-3	1.99	6.08 E-3	2.11	1.55 E-6
	320	8.42 E-4	2.01	1.52 E-3	2.00	1.00 E-7

Example 3.5.1.2 (Steady-state solution to linear problem)**.** For the linear equation, we consider the following problem

$$\begin{aligned} u_t + u_x &= \sin^4(x) \\ u(x,0) &= \sin^2(x) \\ u(0,t) &= 0 \end{aligned} \tag{3.48}$$

with the outflow boundary condition at $x = 2\pi$. The exact solution $u(x,t)$ can be derived by the characteristic theory and can be shown to be positive for all $t > 0$. In Table 3.7 and Table 3.8 we record the errors, numerical orders of accuracy and the minimum value of the numerical approximation, with and without the positivity-preserving limiter respectively. We see that without the positivity-preserving limiter the minimum value of the steady-state approximation is negative. When the limiter is put on, the minimum value becomes positive and the high-order accuracy is not destroyed.

Table 3.7: Error table for Example 3.5.1.2, approximation of the steady state solution to the linear problem (3.48), without the positivity preserving limiter.

k	N	L^2 error	order	L^∞ error	order	$\min u_h$
	20	4.555 E-2	–	6.376 E-2	–	-6.037 E-2
	40	1.177 E-2	1.95	1.704 E-2	1.90	-5.075 E-3
1	80	2.967 E-3	1.99	4.347 E-3	1.97	-2.808 E-4
	160	7.434 E-4	2.00	1.092 E-3	1.99	-1.170 E-5
	320	1.859 E-4	2.00	2.733 E-4	2.00	-3.901 E-7
	20	5.749 E-3	–	9.561 E-3	–	-1.667 E-3
	40	7.482 E-4	2.94	1.121 E-3	3.09	-6.550 E-5
2	80	9.449 E-5	2.99	1.543 E-4	2.86	-2.163 E-6
	160	1.184 E-5	3.00	1.975 E-5	2.97	-6.853 E-8
	320	1.481 E-6	3.00	2.484 E-6	2.99	-2.149 E-9
	20	6.987 E-4	–	6.013 E-4	–	-8.652 E-4
	40	4.564 E-5	3.94	4.743 E-5	3.66	-3.909 E-5
3	80	2.885 E-6	3.98	2.986 E-6	3.99	-1.329 E-6
	160	1.808 E-7	4.00	1.887 E-7	3.98	-4.240 E-8
	320	1.131 E-8	4.00	1.178 E-8	4.00	-1.332 E-9
	20	6.094 E-5	–	4.622 E-5	–	-6.545 E-6
	40	1.955 E-5	4.96	1.335 E-6	5.11	-1.329 E-6
4	80	6.149 E-8	4.99	4.597 E-8	4.86	-5.211 E-8
	160	1.925 E-9	5.00	1.471 E-9	4.97	-1.715 E-9
	320	6.018 E-11	5.00	4.623 E-10	4.99	-5.426 E-11

Example 3.5.1.3 (Steady-state solution to Burgers problems)**.** For the Burgers

Table 3.8: Error table for Example 3.5.1.2, approximation of the steady state solution to the linear problem (3.48) with the positivity preserving limiter.

k	N	L^2 error	order	L^∞ error	order	$\min u_h$
	20	4.253 E-2	–	6.376 E-2	–	1.000 E-13
	40	1.173 E-2	1.86	1.704 E-2	1.90	1.000 E-13
1	80	2.966 E-3	1.98	4.347 E-3	1.97	1.000 E-13
	160	7.434 E-4	2.00	1.092 E-3	1.99	1.000 E-13
	320	1.859 E-4	2.00	2.733 E-4	2.00	1.000 E-13
	20	5.762 E-3	–	9.561 E-3	–	1.000 E-13
	40	7.482 E-4	2.95	1.121 E-3	3.09	1.000 E-13
2	80	9.449 E-5	2.99	1.543 E-4	2.86	1.000 E-13
	160	1.184 E-5	3.00	1.975 E-5	2.97	1.000 E-13
	320	1.481 E-6	3.00	2.484 E-6	2.99	1.000 E-13
	20	1.015 E-3	–	2.240 E-3	–	1.000 E-13
	40	5.077 E-5	4.32	9.673 E-5	4.53	1.000 E-13
3	80	2.932 E-6	4.11	3.253 E-6	4.89	1.000 E-13
	160	1.812 E-7	4.02	1.887 E-7	4.11	1.000 E-13
	320	1.131 E-8	4.00	1.178 E-8	4.00	1.000 E-13
	20	6.141 E-5	–	4.622 E-5	–	1.000 E-13
	40	2.230 E-5	4.78	4.356 E-6	3.41	1.000 E-13
4	80	6.830 E-8	5.03	1.700 E-7	4.68	1.000 E-13
	160	2.045 E-9	5.06	5.591 E-9	4.93	1.000 E-13
	320	6.214 E-10	5.04	1.773 E-10	4.98	1.000 E-13

equation, we consider the steady state solution to the following problem

$$\begin{aligned} u_t + \left(\frac{u^2}{2}\right)_x &= \sin\left(\frac{x}{4}\right) \\ u(x,0) &= x \\ u(0,t) &= 0 \end{aligned} \tag{3.49}$$

with the outflow boundary condition at $x = 2\pi$. Again, by the characteristic theory, one can show that the solution to this problem is always positive. In Table 3.9 and 3.10, we present numerical results for the cases where the positivity-preserving limiter is on and off respectively. This example shows the effectiveness of the positivity preserving limiter for nonlinear scalar problem. With the limiter on, the solution

stays positive and the high-order accuracy is preserved.

Table 3.9: Error table for Example 3.5.1.3, approximation of the steady state solution to the Burgers equation (3.49), without the positivity preserving limiter.

k	N	L^2 error	order	L^∞ error	order	$\min u_h$
2	20	2.915 E-6	–	5.085 E-6	–	-8.073 E-6
	40	3.508 E-7	3.05	6.358 E-7	3.00	-1.009 E-6
	80	4.293 E-8	3.03	7.948 E-8	3.00	-1.261 E-7
	160	5.304 E-9	3.02	9.934 E-9	3.00	-1.577 E-8
3	20	2.336 E-9	–	1.648 E-9	–	-3.855 E-10
	40	1.445 E-10	4.02	1.030 E-10	4.00	-1.204 E-11
	80	9.010 E-11	4.00	6.525 E-11	3.98	-3.763 E-13
	160	6.031 E-12	3.90	4.978 E-12	3.71	-1.175 E-14
4	20	3.403 E-10	–	6.312 E-10	–	-1.497 E-9
	40	1.010 E-11	5.07	1.970 E-11	5.00	-4.678 E-11
	80	3.116 E-13	5.02	5.398 E-13	5.19	-1.462 E-12

Table 3.10: Error table for Example 3.5.1.3, approximation of the steady state solution to the Burgers equation (3.49), with the positivity preserving limiter.

k	N	L^2 error	order	L^∞ error	order	$\min u_h$
2	20	3.207 E-6	–	4.408 E-6	–	1.000 E-13
	40	3.696 E-7	3.12	6.010 E-7	2.87	1.000 E-13
	80	4.435 E-8	3.06	8.171 E-8	2.88	1.000 E-13
	160	5.414 E-9	3.03	1.083 E-8	2.92	1.000 E-13
3	20	2.338 E-9	–	1.648 E-9	–	1.000 E-13
	40	1.445 E-10	4.02	1.030 E-10	4.00	1.000 E-13
	80	9.010 E-11	4.00	6.511 E-11	3.98	1.000 E-13
	160	6.031 E-12	3.90	4.987 E-12	3.71	1.051 E-14
4	20	4.792 E-10	–	9.061 E-10	–	1.000 E-13
	40	1.253 E-11	5.26	3.102 E-11	4.87	1.000 E-13
	80	3.653 E-13	5.10	1.086 E-12	4.84	1.000 E-13

Next, let us turn to another steady-state Burgers problem

$$\begin{aligned} u_t + \left(\frac{u^2}{2}\right)_x &= \sin^3\left(\frac{x}{4}\right) \\ u(x,0) &= \sin^2\left(\frac{x}{4}\right) \\ u(0,t) &= 0 \end{aligned} \tag{3.50}$$

with the outflow boundary condition at $x = 2\pi$. This problem is more difficult than the previous one, since the source is much closer to zero around $x = 0$. In Table 3.11, we present the error, numerical convergence rate as well as the number of time steps taken to reach the steady state solution, which is denoted by N_T. Both cases where the positivity preserving limiter is on and off are presented respectively. We use this example to illustrate that the stability added to the scheme by the positivity-preserving limiter helps accelerate the convergence towards the steady-state solution.

Table 3.11: Error table for Example 3.5.1.3, approximation of the steady-state solution to the Burgers equation (3.50), with and without the positivity preserving limiter.

		without limiter			with limiter		
k	N	L^2 error	order	N_T	L^2 error	order	N_T
2	20	1.71 E-5	–	670	1.71 E-5	–	158
	40	2.11 E-6	3.02	1962	2.11 E-6	3.02	500
	80	2.62 E-7	3.01	4973	2.62 E-7	3.01	1560
	160	3.29 E-8	2.99	8352	3.27 E-8	3.01	4560
3	20	9.10 E-8	–	1162	9.10 E-8	–	970
	40	5.36 E-9	4.08	3440	5.36 E-9	4.08	2211
	80	3.44 E-10	3.96	9007	3.35 E-10	4.00	2653

3.5.2 Moving shocks

Next, we test the proposed scheme on problems involving moving shocks. In all the numerical experiments below, quadratic P^2 element and non-uniform mesh are employed.

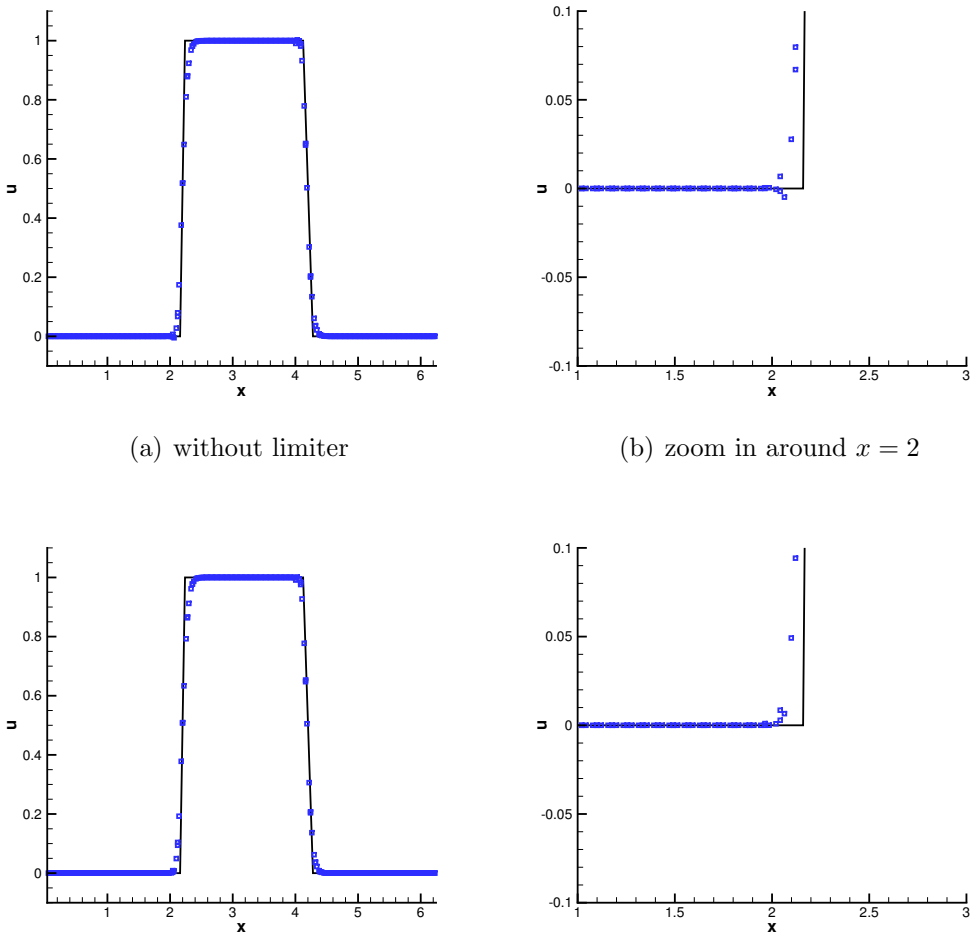
Example 3.5.2.1 (Linear Problem)**.** The first example is the linear equation with the initial data

$$u_0(x) = \begin{cases} 1, & \text{if } x \in [3,4] \\ 0, & \text{otherwise} \end{cases}$$

In Figure 3.1, we present the numerical results at $T = 0.2$, with and without the positivity-preserving limiter respectively. For the linear equation, we take $\min_j \lambda_j = r_2 = 0.266$ as listed in Table 3.1. As the zoom-in plots show, the limiter helps the solution keep positive. Without the limiter a negative undershoot appears.

Example 3.5.2.2 (Burgers problem)**.** Next, let us consider the Burgers equation with the initial condition $u_0(x) = 1 + \sin(x)$ and periodic boundary conditions. The initial condition is positive and takes value zero at $x = \pi$. At $T = 1.5$, a shock is developed. In Figure 3.2, we show the numerical approximation with and without the limiter respectively. We see that even though the profiles are smeared due to the first-order accuracy of the backward Euler time discretization, the effectiveness of the positivity-preserving limiter can still be observed in the zoom-in plots around the shock.

Example 3.5.2.3 (Buckley-Leverett problem)**.** In this example, the Buckley-Leverett problem is considered, with $f(u) = \frac{4u^2}{4u^2+(1-u)^2}$. For this flux function, we have $f'(0) = f'(1) = 0$. If we start with the usual step function $u_0(x) = I_{[3,4]}$, numerical experiments indicate that no matter how large Δt is, we can not make the cell average $\bar{u}_j^1$ positive for all j. The same phenomena is also observed for the Burgers



(a) without limiter

(b) zoom in around $x = 2$

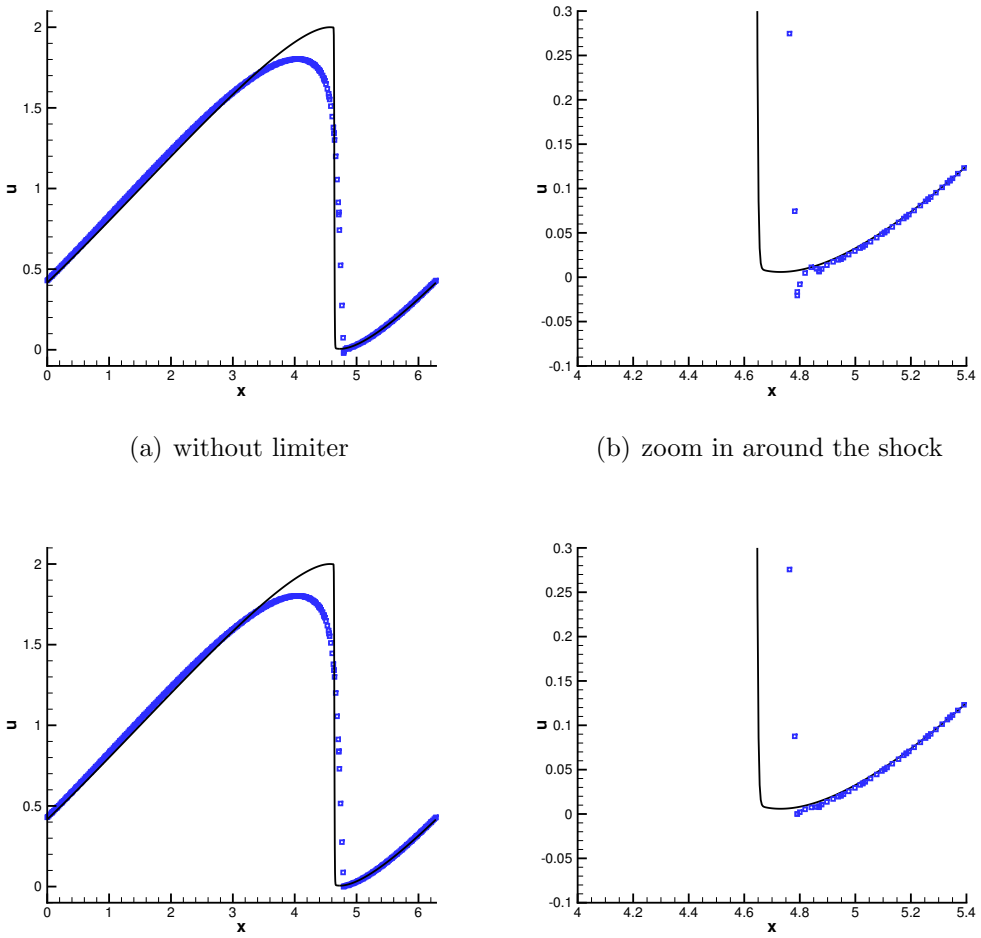
(c) with limiter

(d) zoom in around $x = 2$

Figure 3.1: Example 3.5.2.1: at $T = 0.2$, with $\Delta t = 0.266 \max_j h_j$ and $N = 120$. Solid line is the exact solution. Blue squares are point values on the Legendre-Gauss-Lobatto points.

problem with step function as the initial condition. This indicates that the lower bound for the CFL number is also necessary for the positivity-preserving limiter to work for nonlinear problems. And in general, the limiter may not be applied for problems with sonic points. In this example, we change the initial condition to

$$u_0(x) = \begin{cases} 0.9, & \text{if } x \in [3, 4] \\ 10^{-3}, & \text{otherwise} \end{cases}$$



(a) without limiter

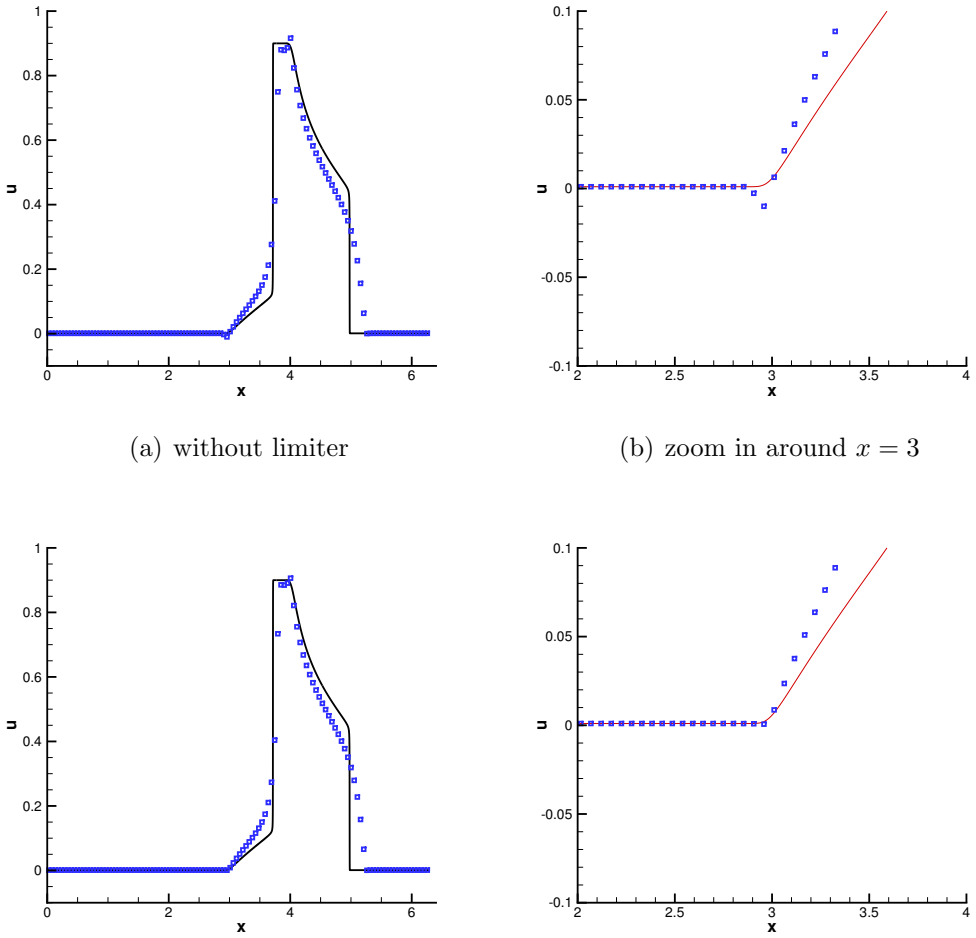
(b) zoom in around the shock

(c) with limiter

(d) zoom in around the shock

Figure 3.2: Example 3.5.2.2 at $T = 1.5$. Solid line is the exact solution. Blue squares are point values of the numerical approximation on the Legendre-Gauss-Lobatto points. Mesh with $N = 120$ cells and CFL $= 2$.

In Figure 3.3, we present the plots with and without limiter. Even though the profile is smeared around the shocks and the rarefaction wave by the first-order time-discretization, the positivity-preserving property is observed in the zoom-in plots.



(a) without limiter

(b) zoom in around $x = 3$

(c) with limiter

(d) zoom in around $x = 3$

Figure 3.3: Example 3.5.2.3 at $T = 0.6$. Solid line is the exact solution. Blue squares are point values of the numerical approximation on the Legendre-Gauss-Lobatto points. Mesh with $N = 120$ cells and CFL $= 3.5$.

3.5.3 Compressible Euler system

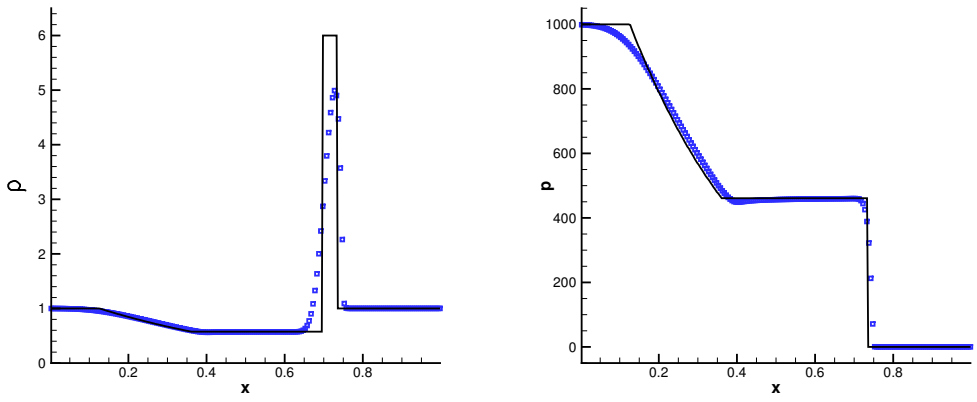
At last, we turn to the examination of the applicability of the proposed scheme to the compressible Euler system (3.38). The computational domain $\Omega = [0, 1]$ is decomposed into nonuniform mesh. The generic ratio of specific heats is taken to be $\gamma = 1.4$. For all the examples, quadratic element is employed with the positivity-preserving limiter on. And the time stepping size is set to be $\Delta t = \frac{\text{CFL}}{\|c+|v|\|_\infty} \min_j h_j$,

where c is the sound speed and v is the velocity. We take CFL $= 2$ for all the examples.

Example 3.5.3.1 (Shock tube)**.** First, let us consider the following shock tube problem.

$$\begin{cases} \rho = 1, & \text{if } x < 0.5 \\ v = 0, & \text{if } x < 0.5 \\ p = 1000, & \text{if } x < 0.5 \end{cases}, \quad \begin{cases} \rho = 1, & \text{if } x \geq 0.5 \\ v = 0, & \text{if } x \geq 0.5 \\ p = 0.01, & \text{if } x \geq 0.5 \end{cases}$$

The numerical approximation is presented in Figure 3.4. The solutions consists of a strong shock wave, a contact discontinuity and a rarefaction wave. All these features are well captured by our scheme. Moreover, with the positivity-preserving limiter on, both the pressure and the density stays positive during the simulation.



(a) density (b) pressure

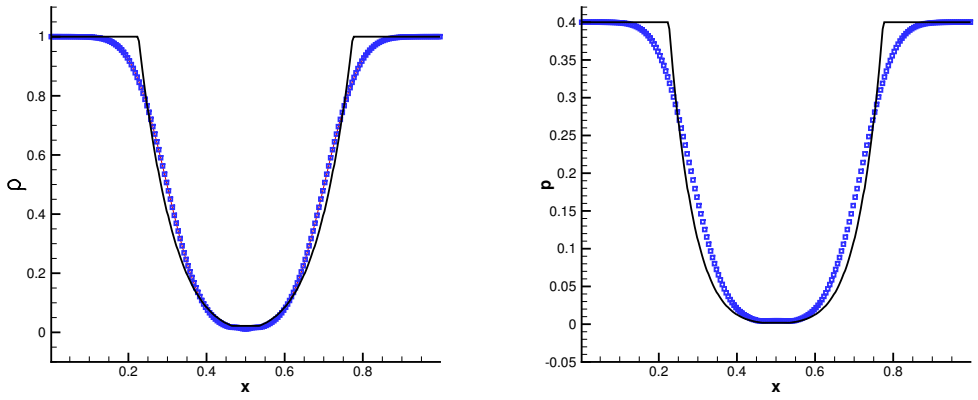
Figure 3.4: Example 3.5.3.1 at $T = 0.01$ with $N = 200$ and CFL $= 2$. With the positivity-preserving limiter on. Solid line is the exact solution and blue squares are numerical approximations.

Example 3.5.3.2 (Double rarefaction)**.** The double rarefaction problem starts from

the following initial condition

$$\begin{cases} \rho = 1, & \text{if } x < 0.5 \\ v = -2, & \text{if } x < 0.5 \\ p = 0.4, & \text{if } x < 0.5 \end{cases}, \quad \begin{cases} \rho = 1, & \text{if } x \geq 0.5 \\ v = 2, & \text{if } x \geq 0.5 \\ p = 0.4, & \text{if } x \geq 0.5 \end{cases}$$

This problem has solution consisting of two symmetric rarefaction waves and trivial contact wave of zero speed. The region between the nonlinear waves around $x = 0.5$ is close to vacuum, which brings difficulty to the simulation. Without the limiter, the nonlinear solver will experience a hard time to converge. In Figure 3.5, we show the plots for the pressure and the density and we see that the near-vacuum region is well resolved with the help of the positivity-preserving limiter.

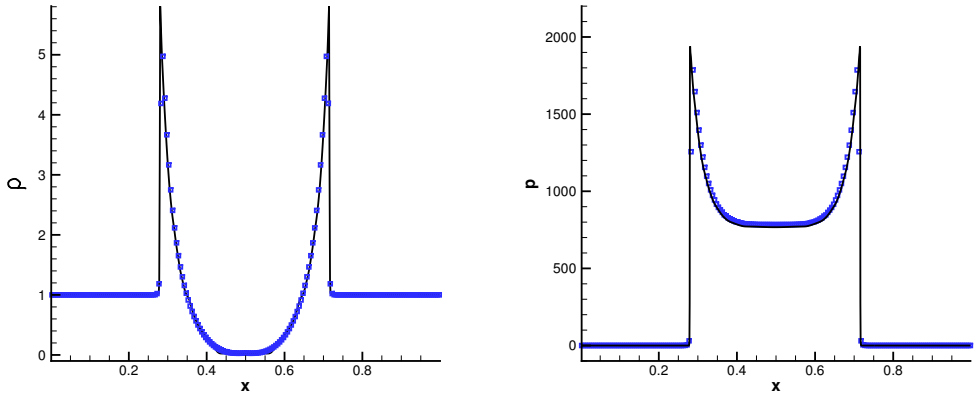


(a) density (b) Pressure

Figure 3.5: Example 3.5.3.2 at $T = 0.1$, with $N = 200$ and CFL $= 2$. With the positivity-preserving limiter on. Solid line is the exact solution and blue squares are the numerical approximations.

Example 3.5.3.3 (Blast wave)**.** The last example is the Sedov point-blast wave [30]. Initially the gas is steady with uniform density one in the whole domain. The pressure is set to be $p = 10^{-9}$, except in the central cell, where the pressure is as high as $p = 10^4$. Then a blast-wave starts to propagate from the central cell with

a shock front. This problem is difficult, since it involves a low density region and strong shocks. In Figure 3.6 we present the numerical approximation and the exact solution [30] for the pressure and density respectively. The positivity-preserving limiter not only helps keep the low-density region positive, but also add robustness to the nonlinear solver. Without it, the code breaks down due to the failure of convergence of the nonlinear solver.



(a) density (b) Pressure

Figure 3.6: Example 3.5.3.3 at $T = 0.003$, with $N = 200$ and CFL $= 2$. With the positivity-preserving limiter on. Solid line is the exact solution and blue squares are the numerical approximations.

3.6 Concluding remarks

In this chapter, we develop an implicit positivity-preserving DG method with high-order spacial accuracy for one-dimensional conservation laws. This work is an extension of the positivity-preserving limiter in [86, 87] for explicit schemes to implicit ones with backward Euler time discretization. To make the scheme positive, a lower bound for the CFL number is necessary. This conclusion is verified via both theoretical analysis and numerical experiments. The Crank-Nicolson time discretization

is also considered and preliminary analysis on uniform mesh indicates that both the upper bound and lower bound are required to apply the limiter. The positivity-preserving limiter not only makes the numerical approximation physically meaningful but also brings robustness to the scheme and accelerates convergence towards the steady-state solution. The scheme also sees its success on the compressible Euler system. In the future, we have the following three directions to further explore. First, even though the results in this chapter trivially generalize to multidimensional tensor product meshes and polynomial spaces, positivity-preserving DG schemes for multi-dimensional domain with unstructured mesh needs to be further developed. Second, we plan to generalize the proposed implicit positivity-preserving DG scheme to other types of equations such as the convection-diffusion equations. Thirdly, high-order implicit temporal discretizations need to be considered. Since there are no implicit SSP methods with order greater than one [23], we need to turn to other types of implicit time discretizations, such as BDF methods and fully implicit RK methods.

Chapter Four

Positivity and Bound-Preserving DG Methods for Relativistic Hydrodynamics

4.1 Introduction

In this chapter, we discuss discontinuous Galerkin (DG) methods to solve the two-dimensional special relativistic hydrodynamics (RHD), which can be written into a system of conservation laws as below

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = 0, \tag{4.1}$$

with

$$\mathbf{u} = \begin{pmatrix} D \\ m \\ n \\ E \end{pmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} Dw \\ mw + p \\ nw \\ m \end{pmatrix}, \quad \mathbf{g}(\mathbf{u}) = \begin{pmatrix} Dv \\ mv \\ nv + p \\ n \end{pmatrix} \tag{4.2}$$

as well as its one dimensional version. The method can be easily extended to three-dimensions, but this is not discussed here. In (4.2), p, D, m, n and E are the thermal pressure, mass density, momentum in the x-direction, momentum in the y-direction, and energy, respectively. (w, v) is the velocity field of the fluid. Moreover, units are normalized such that the speed of light is $c = 1$. If we denote ρ to be the proper rest-mass density, then the conservative variable $\mathbf{u}$ can be written as

$$D = \gamma\rho, \tag{4.3}$$

$$m = Dh\gamma w, \tag{4.4}$$

$$n = Dh\gamma v, \tag{4.5}$$

$$E = Dh\gamma - p, \tag{4.6}$$

where $\gamma = (1 - w^2 - v^2)^{-1/2}$ is the Lorentz factor and h is the specific enthalpy. To close the system, we specify an equation of state $h = h(p, \rho)$. For ideal gas

$$\rho h = \rho + p\Gamma/(\Gamma - 1) \tag{4.7}$$

with Γ being the specific heat ratio, such that $1 < \Gamma \leq 2$ (see for example [66]). Moreover, the sound speed is defined as

$$c_s = \sqrt{\frac{\Gamma p}{\rho h}} = \sqrt{\frac{(\Gamma - 1)(h - 1)}{h}}.$$

Relativistic flows are widely used to model high-energy astrophysical phenomena, such as blast waves of supernova explosions, gravitational collapse and accretion, superluminal jets and gamma-ray bursts. When the speed of the flow is near the speed of the light but there is no strong gravitational field involved, the framework of special relativity is accurate to certain extent to describe the physical phenomena.

Physically, the density D and pressure p are positive, and the velocity field (w, v) satisfies $w^2 + v^2 \leq 1$. Therefore, the admissible set for RHD takes the following form

$$G = \{\mathbf{u} : D > 0, p(\mathbf{u}) > 0, w(\mathbf{u})^2 + v(\mathbf{u})^2 \leq 1\}.$$

By (4.7), it is easy to see that $h > 1$, which further yields $0 < c_s \leq 1$. It is demonstrated in [42, 77] that G is convex and can be represented in terms of the conservative variables as below

$$G = \{\mathbf{u} : D > 0, E > \sqrt{D^2 + m^2 + n^2}\}. \tag{4.8}$$

In order to update the flux in the computation, we further need the inverse of (4.3)-

(4.6) from the conservative vector $\mathbf{u}$ to the primitive vector $\mathbf{w} = \{\rho, w, v, p\}$. Unlike its Newtonian counterpart, in RHD we do not have an explicit formula for this inverse map. By (4.6), (4.7) and the definition of γ, we can derive the nonlinear equation satisfied by the pressure p

$$f(p; \mathbf{u}) := E - \frac{p}{\Gamma - 1} - D\sqrt{1 - \frac{m^2 + n^2}{(E+p)^2}} - \frac{m^2 + n^2}{E + p} = 0, \qquad p \in [0, +\infty) \quad (4.9)$$

By a simple calculation one can show that, if $\mathbf{u} \in G$, then $\frac{\partial f}{\partial p}(p; \mathbf{u}) < 0$ and the equation (4.9) has a unique positive solution [77]. In practice, once the bounds (4.8) are satisfied by the numerical solution, (4.9) can be solved efficiently by standard root finding methods. After the pressure is obtained, other quantities can be calculated sequentially and directly via (4.3)-(4.7). Therefore, an efficient and effective conversion from the conservative vector to the primitive vector crucially depends on guaranteeing the bound-preserving property (4.8) for the numerical solution, which is the main objective of this chapter.

Numerical simulation of RHD has been intensively studied in the last few decades. The first Eulerian method dates back to the early work by Wilson [74, 75], in which the author used explicit finite differencing techniques and a monotonic transport algorithm to discretize the advection terms of the RHD equations. They applied the von Neumann-Richtmyer artificial viscosity method [71, 60] to handle the shock waves. Though this procedure remained standard through the 1980's, it turned out to be unable to resolve the extremely strong shock structures that would appear in the ultra-relativistic regime ($\gamma \geq 2$) [7]. A major breakthrough came in 1994 when Martí et al. [35, 36, 37] reformulated the RHD into the conservation form and for the first time introduced the Godunov-type *high resolution shock capturing* (HRSC) techniques from the classical gas dynamics simulation to that of the RHD. The HRSC techniques produced high-order approximation in the smooth region and were

able to capture shocks and steep transients sharply without spurious oscillations. Since then, various Riemann solvers and modern techniques in gas dynamics were extended to the RHD simulation, for example the relativistic Roe solver by Eulderink et al. [20, 21], the HLLE solver extended by Schneider et al. [61] and the recent relativistic HLLC solver carried out by Mignone et al. [42]. Furthermore, Martí and Müller [37] extended the PPM method [14] to the one dimensional RHD and the multidimensional version was accomplished by Mignone et al. [42]. Donat et al. developed a flux splitting method based on the spectral decomposition of the Jacobian matrix in [18]. Kinetic schemes were developed for RHD in [31, 53].

Besides these, other high order methods were also well studied, for example, Dolezal and Wong introduced the essentially non-oscillatory (ENO) scheme for the RHD in 1995 [17], which was extended by Del Zanna and Bucciantini in [83]. Subsequently, the WENO algorithm was applied to RHD in [67]. The discontinuous Galerkin methods were also applied to general relativistic hydrodynamics by Radice and Rezzolla [58]. More recently, Zhao and Tang applied WENO limiters to the DG schemes in [93] for the special relativistic case. Moreover, the adaptive mesh refinement (AMR) techniques were proved to be powerful and useful in simulating RHD, for example [84, 73, 25] and many software packages have been developed with AMR support and RHD extension, such as the ENZO [47] and RAMSES [68], etc. Additional methods and details can be found in [80, 81, 76, 92, 34], and the references therein, as well as the review paper [38].

Although the methods mentioned above have been working successfully in most cases, many authors have reported difficulties in maintaining the physical bounds for the numerical approximation, see, e.g. [21, 61, 18, 17, 84, 93], especially in extreme relativistic cases such as flows with large Lorentz factor, strong shocks, low density and pressure. The violation of the physical bounds may lead to the crash

of the code. Actually, a slight violation of the bound $E > \sqrt{D^2 + m^2 + n^2}$ could give negative pressure, and a more severe violation with $E < \sqrt{m^2 + n^2}$ will even result in the nonexistence of the solution to (4.9), see [61]. Negative pressure ruins the characteristic decomposition in the HRSC methods or kills the Roe solver (see [21]), which all require the square root of the pressure to calculate the speed of the sound. All these could make the code crash in practice. Several ways have been adopted in the literature to get around this problem. For example, in [37], in order to avoid numerical difficulties, the authors set the initial physical quantities, such as the internal energy and pressure, a small number away from zero. In [84], the authors monitor the physical bounds at every time step and once they are broken the calculation will be repeated under a smaller CFL condition with more diffusive schemes. However, these ad hoc techniques are not guaranteed to cure the problem, especially for higher order schemes, and even if they do, the high order accuracy may no longer be maintained.

As introduced in Chapter 1, physical bounds preserving high order numerical last few years. For the RHD equation, there are few works to mention. Recently, based on the technique in [26], Wu and Tang [77] developed a bound preserving WENO finite difference scheme for the RHD system (4.1). However, their positivity-preserving technique is based on the flux cut-off limiter proposed by Hu, Adams and Shu in [26]. The main drawback of this method is that to preserve the physical bounds while maintaining the high-order accuracy, a rather severe CFL constraint must be assumed.

In this chapter, we restrict ourselves to the DG method and provide a systematic and rigorous way to fix the physical bound violation problem, by extending the bound preserving technique for gas dynamics in [87]. For DG methods, to overcome the oscillations around strong discontinuities, various limiters have been designed,

for example the total variation diminishing or total variation bounded (TVD/TVB) limiters in [63, 12] and the WENO limiter in [56]. But for the RHD system in the extreme relativistic cases, the TVD/TVB or the WENO limiter, when applied alone, will not guarantee to preserve the physical bound and the code could still crash. We would like to emphasize that the BP limiter developed in this chapter is not meant as a substitution to the TVB or the WENO limiter or other non-oscillatory techniques, but is simply a remedy to help maintain the physical bounds without affecting the high order accuracy. The TVB or WENO limiter could still be used and may still be necessary in cases involving strong shocks, since the BP limiter is not designed to remove oscillations around shocks, especially when these oscillations are not happening near the physical bounds. However, since the our main objective is the discussion on the bound-preserving limiter, we will not discuss in length about the DG method itself or the TVB or WENO limiters in controlling spurious oscillations. Finally, let us remark that, even though we only discuss the bound-preserving limiter for DG schemes here, the same limiter can be applied to high order finite volume schemes, such as WENO finite volume schemes as well.

The organization of this chapter is as follows. In section 4.2, we study the one-dimensional problem, including the details of the limiters, the L^1 stability of the method. In sections 4.3 and 4.4, we study the problem in two space dimensions and the implementation of the relativistic axisymmetric jets. Numerical experiments are given in section 4.5. Finally, we will end in section 4.6 with concluding remarks and remarks for future work.

4.2 Numerical algorithm in one space dimension

In this section, we proceed to construct the bound-preserving DG scheme to solve the one-dimensional relativistic hydrodynamics.

4.2.1 The DG scheme

We consider the one-dimensional version of (4.1) on the spatial domain $[0,1]$ and solve

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0, \tag{4.10}$$

where the conservative variable $\mathbf{u} = (D, m, E)^T$ is defined in (4.3), (4.4), (4.6) with $\gamma = (1-w^2)^{-1/2}$. The flux is $\mathbf{f}(\mathbf{u}) = (Dw, mw+p, m)^T$ and the equation of state is given in (4.7). The admissible set is defined to be

$$G_1 = \{(D, m, E)^T : D > 0, E > \sqrt{D^2+m^2}\}.$$

It is easy to see that G_1 is convex.

The DG scheme is defined to seek approximation vector $\mathbf{u}_h(t) \in \mathbf{V}_h$ such that in each cell I_j, for any test function $\mathbf{v} \in \mathbf{V}_h$ we have

$$\frac{d}{dt}\left(\mathbf{u}_h(t), \mathbf{v}\right)_j - \left(\mathbf{f}(\mathbf{u}_h(t)), \mathbf{v}_x\right)_j + \hat{\mathbf{f}}_{j+\frac{1}{2}}\mathbf{v}^-_{j+\frac{1}{2}} - \hat{\mathbf{f}}_{j-\frac{1}{2}}\mathbf{v}^+_{j-\frac{1}{2}} = 0, \tag{4.11}$$

where $\mathbf{v}^-_{j+\frac{1}{2}} = \mathbf{v}(x^-_{j+\frac{1}{2}})$, which denotes the left limit of the vector $\mathbf{v}$ at $x_{j+\frac{1}{2}}$. Likewise for $\mathbf{v}^+_{j+\frac{1}{2}}$.

For the numerical flux $\hat{\mathbf{f}}_{j+\frac{1}{2}} = \hat{\mathbf{f}}(\mathbf{u}_h(x^-_{j+\frac{1}{2}}, t), \mathbf{u}_h(x^+_{j+\frac{1}{2}}, t))$, we choose the local

Lax-Friedrichs flux in the following form

$$\hat{\mathbf{f}}(\mathbf{a},\mathbf{b}) = \frac{1}{2}\left[\mathbf{f}(\mathbf{a}) + \mathbf{f}(\mathbf{b}) - \alpha(\mathbf{b}-\mathbf{a})\right], \tag{4.12}$$

where $\alpha = \alpha(\mathbf{a},\mathbf{b})$ is a positive real number depending only on the two interface values in the definition of the numerical flux. The value of it is to be chosen by the bound-preserving technique. Other numerical fluxes such as the HLLC Riemann solver [70, 41] could of course also be considered, but will not be discussed here.

4.2.2 Bound-preserving technique

For (4.10), direct usage of high order DG methods may result in the appearance of negative density and pressure, and physically irrelevant velocity, leading to ill-posed problems. Moreover, the code may blow up once physically irrelevant quantities appear. Therefore, we would like to apply BP limiters to the scheme. We denote $\mathbf{u}_j^n$ and $\overline{\mathbf{u}}_j^n$ to be the numerical solution and its cell average at time level n in cell I_j. For simplicity, throughout the chapter, if we consider generic numerical solution on the whole computational domain Ω, then the subscript j will be omitted. Suppose the exact solution of equation (4.10) is in G_1, we are interested in constructing numerical solutions which are also in G_1. The whole procedure is given below.

We start from the forward Euler time discretization, and briefly illustrate the idea. The fully discrete forward Euler DG scheme is defined as follows. In each cell I_j, we seek $\mathbf{u}_h^{n+1} \in \mathbf{V}_h$, such that

$$(\mathbf{u}_j^{n+1}, \mathbf{v})_j = (\mathbf{u}_j^n, \mathbf{v})_j + \lambda_j \left[\hat{\mathbf{f}}^n_{j-\frac{1}{2}}\mathbf{v}_{j+\frac{1}{2}} - \hat{\mathbf{f}}^n_{j+\frac{1}{2}}\mathbf{v}_{j-\frac{1}{2}}\right], \quad \forall \mathbf{v} \in \mathbf{V}_h \tag{4.13}$$

where $\lambda_j = \Delta t/h_j$ and $\hat{\mathbf{f}} = \hat{\mathbf{f}}((\mathbf{u}^n_{j+\frac{1}{2}})^-, (\mathbf{u}^n_{j+\frac{1}{2}})^+)$.

To build the bound-preserving scheme, we follow the general framework outlined in Section 2.2. The first step is to derive sufficient conditions under which given $\mathbf{u}^n_h(x) \in G$ for any $x \in \Omega$, we have $\bar{\mathbf{u}}^{n+1}_j \in G$ for any j. To this end, we generalize the procedure in [87] for the compressible Euler system to the RHD system.

First, by taking the test function $\mathbf{v} = \mathbf{1}$ in (4.13) let us write down the equation satisfied by the cell average $\bar{\mathbf{u}}^{n+1}_j$,

$$\bar{\mathbf{u}}^{n+1}_j = \bar{\mathbf{u}}^n_j + \lambda \left[\hat{\mathbf{f}}((\mathbf{u}^n_{j-\frac{1}{2}})^-, (\mathbf{u}^n_{j-\frac{1}{2}})^+) - \hat{\mathbf{f}}((\mathbf{u}^n_{j+\frac{1}{2}})^-, (\mathbf{u}^n_{j+\frac{1}{2}})^+)\right]. \tag{4.14}$$

Note that the right hand side involves both the point values $\mathbf{u}^{\pm}_{j\pm\frac{1}{2}}$ and the cell average $\bar{\mathbf{u}}^n_j$. Then to further get a handle on $\mathbf{u}^n_h$ to adjust $\bar{\mathbf{u}}^{n+1}_j$, we split the cell average $\bar{\mathbf{u}}^n_j$ into summation of point values via Legendre-Gauss-Lobatto rule, which includes the end points $x_{j\pm\frac{1}{2}}$.

Let ω_i be the Legendre Gauss-Lobatto quadrature weights for the interval $[-\frac{1}{2}, \frac{1}{2}]$ such that $\sum_{i=0}^{M} \omega_i = 1$, with $2M - 3 \geq k$, and denote the corresponding Gauss-Lobatto points in cell I_j as $\{x^i_j\}_{i=1}^M$. Because of the special choice $2M - 3 \geq k$ of the Gauss-Lobatto quadrature rule, we have

$$\bar{\mathbf{u}}^n_j = \sum_{i=0}^{M} \omega_i \mathbf{u}^n_j(x^j_i).$$

Clearly, $\mathbf{u}^n_j(x^j_0) = \mathbf{u}^+_{j-\frac{1}{2}}$ and $\mathbf{u}^n_j(x^j_M) = \mathbf{u}^-_{j+\frac{1}{2}}$. Therefore, by considering $\omega_0 = \omega_M$, we have the equation (4.14) can be rewritten into

$$\bar{\mathbf{u}}^{n+1}_j = \sum_{i=0}^{M} \omega_i \mathbf{u}^n_j(x^i_j) + \lambda_j \left(\hat{\mathbf{f}}((\mathbf{u}^n_{j-\frac{1}{2}})^-, (\mathbf{u}^n_{j-\frac{1}{2}})^+) - \hat{\mathbf{f}}((\mathbf{u}^n_{j+\frac{1}{2}})^-, (\mathbf{u}^n_{j+\frac{1}{2}})^+)\right)$$

$$
\begin{aligned}
&= \sum_{i=1}^{M-1} \omega_i \mathbf{u}_j^n(x_j^i) + \omega_0 \left[(\mathbf{u}_{j-\frac{1}{2}}^n)^+ + \frac{\lambda_j}{\omega_0} \left(\hat{\mathbf{f}} \left((\mathbf{u}_{j-\frac{1}{2}}^n)^-, (\mathbf{u}_{j-\frac{1}{2}}^n)^+ \right) - \hat{\mathbf{f}} \left((\mathbf{u}_{j-\frac{1}{2}}^n)^-, (\mathbf{u}_{j+\frac{1}{2}}^n)^+ \right) \right) \right] \\
&+ \omega_0 \left[(\mathbf{u}_{j+\frac{1}{2}}^n)^- + \frac{\lambda_j}{\omega_0} \left(\hat{\mathbf{f}} \left((\mathbf{u}_{j-\frac{1}{2}}^n)^+, (\mathbf{u}_{j+\frac{1}{2}}^n)^- \right) - \hat{\mathbf{f}} \left((\mathbf{u}_{j+\frac{1}{2}}^n)^-, (\mathbf{u}_{j+\frac{1}{2}}^n)^+ \right) \right) \right] \\
&= \sum_{i=1}^{M-1} \omega_i \mathbf{u}_j^n(x_j^i) + \omega_0 \mathbf{H} \left((\mathbf{u}_{j-\frac{1}{2}}^n)^-, (\mathbf{u}_{j-\frac{1}{2}}^n)^+, (\mathbf{u}_{j+\frac{1}{2}}^n)^-, \frac{\lambda_j}{\omega_0} \right) \\
&+ \omega_0 \mathbf{H} \left((\mathbf{u}_{j-\frac{1}{2}}^n)^+, (\mathbf{u}_{j+\frac{1}{2}}^n)^-, (\mathbf{u}_{j+\frac{1}{2}}^n)^+, \frac{\lambda_j}{\omega_0} \right). \qquad (4.15)
\end{aligned}
$$

where

$$
\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \eta) := \mathbf{b} + \eta \left[\hat{\mathbf{f}}(\mathbf{a}, \mathbf{b}) - \hat{\mathbf{f}}(\mathbf{b}, \mathbf{c}) \right]
$$

Since $\mathbf{u}_j^n(x_j^\alpha) \in G_1$ and G_1 is a convex set, in order to have $\bar{\mathbf{u}}_j^{n+1} \in G_1$, we only to require

$$
\mathbf{H} \left((\mathbf{u}_{j-\frac{1}{2}}^n)^-, (\mathbf{u}_{j-\frac{1}{2}}^n)^+, (\mathbf{u}_{j+\frac{1}{2}}^n)^-, \frac{\lambda_j}{\omega_0} \right), \mathbf{H} \left((\mathbf{u}_{j-\frac{1}{2}}^n)^+, (\mathbf{u}_{j+\frac{1}{2}}^n)^-, (\mathbf{u}_{j+\frac{1}{2}}^n)^+, \frac{\lambda_j}{\omega_0} \right) \in G_1. \qquad (4.16)
$$

In general we need to derive condition on η such that given $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c} \in G_1$, then we have

$$
\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \eta) \in G_1. \qquad (4.17)
$$

By plugging in the definition of $\hat{f}$ as in (4.12) into $\mathbf{H}$, we can obtain

$$
\begin{aligned}
\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \eta) =& \mathbf{b} + \frac{\eta}{2} \left[\mathbf{f}(\mathbf{a}) + \mathbf{f}(\mathbf{b}) - \alpha_1 (\mathbf{b} - \mathbf{a}) \right] \\
&- \frac{\eta}{2} \left[\mathbf{f}(\mathbf{b}) + \mathbf{f}(\mathbf{c}) - \alpha_2 (\mathbf{c} - \mathbf{b}) \right] \\
=& \frac{\eta}{2} \left[\alpha_1 \mathbf{a} + \mathbf{f}(\mathbf{a}) \right] + (1 - \frac{\eta}{2}\alpha_1 - \frac{\eta}{2}\alpha_2)\mathbf{b} + \frac{\eta}{2} \left[\alpha_2 \mathbf{c} - \mathbf{f}(\mathbf{c}) \right] \\
=& \frac{\alpha_1 \eta}{2} \mathbf{H}^+(\mathbf{a}, \alpha_1) + (1 - \frac{\eta}{2}\alpha_1 - \frac{\eta}{2}\alpha_2)\mathbf{b} + \frac{\alpha_2 \eta}{2} \mathbf{H}^-(\mathbf{c}, \alpha_2), \qquad (4.18)
\end{aligned}
$$

where $\alpha_1 = \alpha(\mathbf{a}, \mathbf{b})$, $\alpha_2 = \alpha(\mathbf{b}, \mathbf{c})$ and

$$\mathbf{H}^+(\mathbf{a}, \alpha_1) = \mathbf{a} + \frac{1}{\alpha_1}\mathbf{f}(\mathbf{a}), \quad \mathbf{H}^-(\mathbf{c}, \alpha_2) = \mathbf{c} - \frac{1}{\alpha_2}\mathbf{f}(\mathbf{c})$$

Therefore, in order to have (4.17), we need to ensure

Condition I Find condition on α such that $H^{\pm}(\mathbf{u}, \alpha) \in G_1$, for any $\mathbf{u} \in G_1$.

Condition II Find condition on η such that

$$1 - \frac{\eta}{2}\alpha_1 - \frac{\eta}{2}\alpha_2 \geq 0$$

For the first condition, we have the following lemma.

Lemma 4.2.1. *Suppose* $\mathbf{u} \in G_1$ *and the parameter* α *satisfies*

$$\alpha \geq F(\mathbf{u}) = \frac{|w|(h+1-2h\tau)\gamma^2 + \sqrt{\tau^4(h-1)^2 + \tau^2(h-1)(h+1-2h\tau)}}{\gamma^2(h+1-2h\tau) + \tau^2(h-1)}, \quad (4.19)$$

with $\tau = (\Gamma - 1)/\Gamma$, *then* $\mathbf{H}^{\pm}(\mathbf{u}, \alpha) \in G_1$.

Proof. For simplicity, we only study the proof for H^-, as the proof for H^+ can be obtained along the same line. Recall that $\mathbf{u} = (D, m, E)^T \in G_1$ and $\mathbf{f}(\mathbf{u}) = (Du, mu + p, m)^T$. Therefore,

$$\alpha\mathbf{H}^-(\mathbf{u}, \alpha) = \alpha\mathbf{u} - \mathbf{f}(\mathbf{u}) = ((\alpha - u)D, (\alpha - u)m - p, \alpha E - m)^T.$$

It is easy to see that to obtain $\mathbf{H}^- \in G_1$, we need $\alpha\mathbf{H}^- \in G_1$, i.e.

$$\alpha \geq c_1 := u, \quad (4.20)$$

$$\alpha \geq c_2 := \frac{m}{E}, \quad (4.21)$$

$$(\alpha E - m)^2 \geq ((\alpha - w)m - p)^2 + ((\alpha - w)D)^2. \tag{4.22}$$

Define $q(\alpha) = (\alpha E - m)^2 - ((\alpha - w)m - p)^2 - ((\alpha - w)D)^2$. By using (4.3)-(4.6), and define $\beta = \alpha - w$, and $\tau = (\Gamma - 1)/\Gamma$, we have

$$\begin{aligned} q(\alpha) &= \rho^2 \left[(\beta h\gamma^2 - \alpha\tau(h-1))^2 - \beta^2\gamma^2 - (\beta h\gamma^2 w - \tau(h-1))^2\right] \\ &= \rho^2(h-1)\left[\beta^2\gamma^2(h+1) + (\alpha^2-1)\tau^2(h-1) - 2\alpha\beta h\tau\gamma^2 + 2\beta h\tau\gamma^2 w\right] \\ &= \rho^2(h-1)\left[\beta^2\gamma^2(h+1) + (\alpha^2-1)\tau^2(h-1) - 2\beta^2 h\tau\gamma^2\right] \\ &= \rho^2(h-1)\left[A_2\alpha^2 - 2A_1\alpha + A_0\right], \end{aligned}$$

where

$$A_2 = \gamma^2(h+1-2h\tau) + \tau^2(h-1), \tag{4.23}$$

$$A_1 = u\gamma^2(h+1-2h\tau), \tag{4.24}$$

$$A_0 = u^2\gamma^2(h+1-2h\tau) - \tau^2(h-1). \tag{4.25}$$

Since $1 \leq \Gamma \leq 2$, then $0 \leq \tau \leq 1/2$. It is easy to check that

$$A_2 \geq 0, \quad q(1) \geq 0, \quad q(w) \leq 0, \quad q(-1) \geq 0.$$

To obtain (4.22), we can take

$$\alpha \geq c_3 := \frac{A_1 + \sqrt{(A_1)^2 - A_2 A_0}}{A_2}. \tag{4.26}$$

If $u < 0$, then $c_3 \geq 0 \geq c_1 \geq c_2$. If $u \geq 0$, then $c_3 \geq c_2 \geq c_1 \geq 0$. Therefore, we have $H^- \in G$ under (4.26). Similarly, we can define

$$A_2' = \gamma^2(h+1-2h\tau) + \tau^2(h-1) = A_2,$$

$$A_1' = -u\gamma^2(h+1-2h\tau) = -A_1,$$
$$A_0' = u^2\gamma^2(h+1-2h\tau) - \tau^2(h-1) = A_0.$$

to obtain the lower bound of α :

$$\alpha \geq \frac{A_1' + \sqrt{(A_1')^2 - A_2'A_0'}}{A_2'}. \tag{4.27}$$

(4.26) and (4.27) imply

$$\alpha \geq \frac{|A_1| + \sqrt{(A_1)^2 - A_2A_0}}{A_2},$$

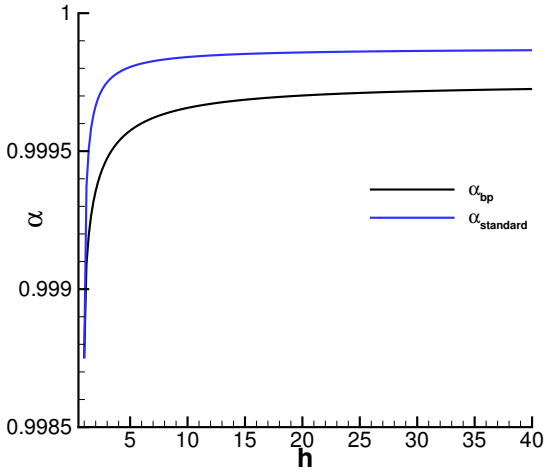
where A_0, A_1 and A_2 are given in (4.23)-(4.25). □

Remark 4.2.1. *The same result also holds for the one-dimensional flow with nonzero transverse velocity. The proof is almost the same and is therefore omitted.*

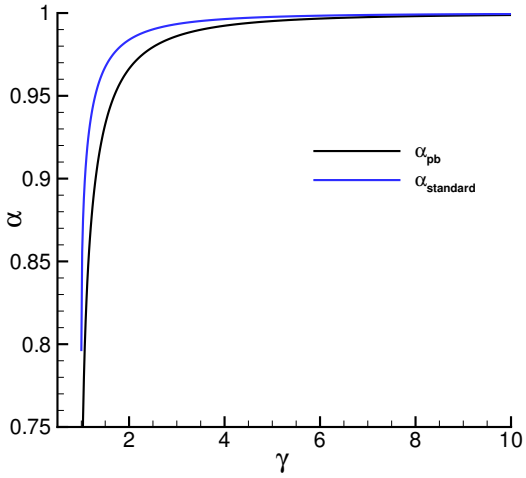
Remark 4.2.2. *In the standard local Lax-Friedrichs flux, the parameter α is chosen to be the upper bound for the absolute value of the eigenvalues of the Jacobian $\partial\mathbf{f}(\mathbf{u})/\partial\mathbf{u}$, which in one dimension is given by (see, e.g. [93])*

$$\tilde{\alpha} = \frac{|w|+c}{1+|w|c} = \frac{\sqrt{1-1/\gamma^2} + \sqrt{(\Gamma-1)(1-1/h)}}{1+\sqrt{1-1/\gamma^2}\sqrt{(\Gamma-1)(1-1/h)}} \tag{4.28}$$

In Figure 4.1, we plot α in (4.19) for the bound-preserving requirement (denoted by α_{bp}) and the spectral radius (4.28) of the Jacobian matrix (denoted by $\alpha_{standard}$), for the two situations with $\gamma = 20$ as a constant or with $h = 20$ as a constant and $\Gamma = 5/3$ in both plots. We can see that the wave speed required for the bound-preserving technique is actually slightly lower (that is, less dissipative!) than the standard wave speed used in the Lax-Friedrichs flux.



(a) γ=20



(b) h=20

Figure 4.1: Plots for the lower bound of α in (4.19) for the bound-preserving requirement (denoted by α_{bp}) and the spectral radius (4.28) of the Jacobian matrix (denoted by $\alpha_{standard}$). The left panel is to keep $\gamma = 20$ constant and the right one is to keep $h = 20$ constant.

Given the condition on α in Lemma 4.19, for the condition II, one sufficient condition is to require

$$\eta \leq \frac{1}{\max\{\alpha_1, \alpha_2\}} \tag{4.29}$$

In summary, in order to have (4.17) we have the following theorem.

Theorem 4.2.1. *Suppose* $\mathbf{a}$, $\mathbf{b}$ *and* $\mathbf{c} \in G_1$, *and the parameter* α *satisfies*

$$\alpha_1 \geq \max\{F(\mathbf{a}),\, F(\mathbf{b})\}, \quad \alpha_2 \geq \max\{F(\mathbf{b}), F(\mathbf{c})\}$$

with F *defined by (4.19), then under the condition* (4.29), *we have*

$$\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \eta) \in G_1$$

Remark 4.2.3. *In the very recent work of Wu and Tang [77], which came to our attention after our original submission of [55], there is a similar result as that in Lemma 4.2.1. However, the authors of [77] assume* $\alpha \geq \tilde{\alpha}$ *as defined in (4.28) and show that this condition is sufficient to ensure* $\mathbf{H}^{\pm}(\mathbf{u}, \alpha) \in G_1$. *Our approach appears to be more constructive and provides a less dissipative solver as shown in Figure 4.1. Moreover, for the first order scheme* $\mathbf{H}$, *the proof in [77] requires a CFL condition* $\eta \leq \frac{1}{2\max\{\alpha_1,\alpha_2\}}$ *to have* $\mathbf{H} \in G_1$. *For our approach, we obtain a more relaxed CFL condition* (4.29).

Now let us come back to (4.15). In each cell I_j, given $\mathbf{u}_j^n(x) \in G_1$ for any $x \in I_j$, or more relaxed condition $\mathbf{u}_j^n(x_j^\alpha) \in G_1$ for any α, the cell average $\bar{\mathbf{u}}_j^{n+1}$ as represented in (4.15) is a convex combination of elements in G_1, since $\sum_{\alpha=0}^{M} \omega_\alpha = 1$ and $\omega_\alpha > 0$. And since G_1 is a convex set, we must have $\bar{\mathbf{u}}_j^{n+1} \in G_1$ if (4.16) holds, which by Lemma 4.19 and Theorem 4.2.1 requires

1. The local Lax-Friedrichs parameter satisfies

$$\alpha \left((\mathbf{u}^n_{j+\frac{1}{2}})^-, (\mathbf{u}^n_{j+\frac{1}{2}})^+ \right) \geq \max\{F((\mathbf{u}^n_{j+\frac{1}{2}})^-, F((\mathbf{u}^n_{j+\frac{1}{2}})^+)\}.$$

2. The following CFL type condition

$$\max_j \lambda_j \leq \frac{\omega_0}{\max_j \alpha((\mathbf{u}^n_{j+\frac{1}{2}})^-, (\mathbf{u}^n_{j+\frac{1}{2}})^+)} \tag{4.30}$$

Scaling limiter

We would like to emphasize that the cell average $\bar{\mathbf{u}}_j^{n+1}$ is shown to be bound-preserving by the original high order DG scheme, before any limiter is applied. Once the cell average is in control, we can modify the numerical solution through a simple scaling limiter

$$\tilde{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^{n+1} + \theta\left(\mathbf{u}_j^{n+1} - \bar{\mathbf{u}}_j^{n+1}\right). \tag{4.31}$$

by taking suitable $\theta \in [0, 1]$, we will have $\tilde{\mathbf{u}}_j^{n+1} \in G_1$ at the Legendre Gauss-Lobatto points, and $\tilde{\mathbf{u}}_j^{n+1}$ is used as the numerical solution at time level $n+1$. For scalar equations, we can prove that this modification does not affect the high order accuracy of the original solution $\mathbf{u}_j^{n+1}$ [88].

Now we give a summary of the complete algorithm. Due to the rounding error, we define

$$G_1^\varepsilon = \left\{ \mathbf{u} = \begin{pmatrix} D \\ m \\ E \end{pmatrix} : D \geq \varepsilon, E \geq \sqrt{m^2 + D^2 + \varepsilon} \right\},$$

$$\partial G_1^\varepsilon = \left\{ \mathbf{u} = \begin{pmatrix} D \\ m \\ E \end{pmatrix} : D = \varepsilon, E = \sqrt{m^2 + D^2 + \varepsilon} \right\}.$$

then the modification of DG solution $\mathbf{u}_j^n$ is given in the following steps.

- Set up a small number $\varepsilon = 10^{-13}$.

- If $\bar{D}_j^n > \varepsilon$, then proceed to the following steps. Otherwise, D_j^n is identified as the approximation to vacuum. Therefore, we take $\widetilde{\mathbf{u}}_j^n = \overline{\mathbf{u}}_j^n$ as the numerical solution and skip the following steps.

- Modify density in each I_j: compute $b_j = \min_\alpha D_j^n(x_j^\alpha)$, where $\{x_j^\alpha\}_{\alpha=0}^M$ are the Gauss-Lobatto points in the cell I_j. If $b_j < \varepsilon$, then take

$$\widetilde{D}_j^n = \overline{D}_j^n + \theta_j^D(D_j^n - \overline{D}_j^n),$$

where

$$\theta_j^D = \frac{\overline{D}_j^n - \varepsilon}{\overline{D}_j^n - b_j}.$$

Then use $\widetilde{D}_j^n$ as the new numerical density D_j^n.

- Enforce $E_j^n \geq \sqrt{(m_j^n)^2 + (\tilde{D}_j^n)^2 + \varepsilon}$ on each Gauss-Lobatto point in each cell I_j: define $\mathbf{q}_j^\alpha = \mathbf{u}_j^n(x_j^\alpha)$ in cell I_j. If $\mathbf{q}_j^\alpha \in G_1^\varepsilon$, then take $\theta_j^\alpha = 1$. Otherwise, take θ_j^α to be the root of

$$\delta_1\left((1-t)\bar{\mathbf{u}}_j^n + t\mathbf{q}_j^\alpha\right) = 0, \quad t \in (0,1) \tag{4.32}$$

where $\delta_1(\mathbf{u}) = E - \sqrt{E^2 + m^2 + \varepsilon}$. Then define $\theta_j = \min_{i=0,\cdots,m} \theta_j^\alpha$, and use

$$\widetilde{\mathbf{u}}_j^n = \overline{\mathbf{u}}_j^n + \theta_j(\mathbf{u}_j^n - \overline{\mathbf{u}}_j^n),$$

as the DG approximation in cell I_j.

Remark 4.2.4. *In cases where wild data is involved, e.g. the shock heating problem, due to the round-off error, the solution to (4.32) may not strictly make* $(1-\theta_i^j)\bar{\mathbf{u}}_j^n + \theta_j^\alpha \mathbf{q}_j^\alpha \in G_1^\varepsilon$. *In [72], when solving the gas detonation propagation problems with bound*

preserving DG methods, the authors provide a more robust way to obtain θ_i^j *without solving any equation. here, we generalize it to the RHD system. It is straightforward to check that for* $t \in (0,1)$,

$$\delta_1((1-t)\bar{\mathbf{u}}_j^n + t\mathbf{q}_j^\alpha) \geq (1-t)\delta_1(\bar{\mathbf{u}}_j^n) + t\delta_1(\mathbf{q}_j^\alpha)$$

given $D_j^n(x_j^\alpha) \geq \varepsilon$, $\bar{\mathbf{u}}_j^n \in G^\varepsilon$ *and* $\delta_1\left(\mathbf{q}_j^\alpha\right) < 0$, *it is sufficient to require*

$$(1-t)\delta_1(\bar{\mathbf{u}}_j^n) + t\delta_1(\mathbf{q}_j^\alpha) = 0, \quad i.e., \quad t = \frac{\delta_1(\bar{\mathbf{u}}_j^n)}{\delta_1(\bar{\mathbf{u}}_j^n) - \delta_1(\mathbf{q}_j^\alpha)}$$

such that $(1-t)\bar{\mathbf{u}}_j^n + t\mathbf{q}_j^\alpha \in G^\varepsilon$. *Note that the* t *obtained in this way is in general smaller than that by solving the equation (4.32), but the high order accuracy can still be shown following the same way as in [87].*

4.2.2.1 L^1 stability

Following [79], we can show the L^1-stability of the numerical scheme, for periodic or compactly supported boundary conditions, with the BP limiter. Since D_h^n is positive, we have, by the conservative property of the DG scheme,

$$\|D_h^n\|_{L^1} = \int_\Omega D_h^n(x)dx = \int_\Omega D_h^0(x)dx = \|D_h^0\|_{L^1},$$

where $\|\cdot\|_{L^1}$ is the standard L^1-norm of w on Ω. Similarly, we can prove $\|E_h^n\|_{L^1} = \|E_h^0\|_{L^1}$. Moreover, it is easy to obtain

$$m = \frac{w\Gamma}{\Gamma - \sigma}E,$$

where $\sigma = \frac{(\Gamma-1)(h-1)}{h\gamma^2} \leq \Gamma - 1$. Hence,

$$\|m\|_{L^1} \leq \frac{|w|\Gamma}{\Gamma - \sigma}\|E\|_{L^1} \leq \Gamma\|E\|_{L^1}.$$

In the last inequality, we use the fact that $|w| \leq 1$. Therefore,

$$\|\mathbf{u}_h^n\|_{L^1} \leq \|D_h^n\|_{L^1} + (\Gamma+1)\|E_h^n\|_{L^1} = \|D_h^0\|_{L^1} + (\Gamma+1)\|E_h^0\|_{L^1} \leq (\Gamma+1)\|\mathbf{u}_h^0\|_{L^1},$$

where $\|\mathbf{u}\|_{L^1} = \|D\|_{L^1} + \|m\|_{L^1} + \|E\|_{L^1}$. This implies the L^1-stability of the scheme.

4.2.3 High order time discretizations

All the previous analyses are based on first-order Euler forward time discretization. We can also use strong stability preserving (SSP) high-order time discretizations to solve the ODE system $\mathbf{u}_t = \mathbf{L}\mathbf{u}$. More details of these time discretizations can be found in [65, 64, 23]. In this chapter, we use the third order SSP Runge-Kutta method [65]

$$\begin{aligned}
\mathbf{u}_h^{(1)} &= \mathbf{u}_h^n + \Delta t\mathbf{L}(\mathbf{u}_h^n), \\
\mathbf{u}_h^{(2)} &= \frac{3}{4}\mathbf{u}_h^n + \frac{1}{4}\left(\mathbf{u}_h^{(1)} + \Delta t\mathbf{L}(\mathbf{u}_h^{(1)})\right), \\
\mathbf{u}_h^{n+1} &= \frac{1}{3}\mathbf{u}_h^n + \frac{2}{3}\left(\mathbf{u}_h^{(2)} + \Delta t\mathbf{L}(\mathbf{u}_h^{(2)})\right),
\end{aligned} \tag{4.33}$$

and the third order SSP multi-step method [65]

$$\mathbf{u}^{n+1} = \frac{16}{27}\left(\mathbf{u}^n + 3\Delta t\mathbf{L}(\mathbf{u}^n)\right) + \frac{11}{27}\left(\mathbf{u}^{n-3} + \frac{12}{11}\Delta t\mathbf{L}(\mathbf{u}^{n-3})\right). \tag{4.34}$$

Since an SSP time discretization is a convex combination of Euler forward, by using the limiter designed in section 4.2.2, the numerical solution obtained from the full scheme is also in G_1.

4.3 Numerical algorithm in two space dimensions

In this section, we extend the bound-preserving discontinuous Galerkin method to relativistic hydrodynamics in two space dimensions. For simplicity, we use Euler forward time discretization and construct high-order bound-preserving DG schemes to solve (4.1). In this section, we construct the numerical solutions to be in G, which is given in (4.8).

For simplicity, we use uniform rectangular meshes. The algorithm including the BP limiter can be easily generalized to unstructured meshes, along the lines in [91]. The cell is defined as $I_{ij} = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times \left[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}\right]$, and the mesh sizes in x and y directions are denoted as Δx and Δy, respectively. At time level n, we approximate the exact solution with a vector of polynomials of degree k, $\mathbf{u}_{ij}^n = (D_{ij}^n, m_{ij}^n, n_{ij}^n, E_{ij}^n)^T$, and define the cell average $\overline{\mathbf{u}}_{ij}^n = (\overline{D}_{ij}^n, \overline{m}_{ij}^n, \overline{n}_{ij}^n, \overline{E}_{ij}^n)^T$. Moreover, we denote $\mathbf{u}^+_{i-\frac{1}{2},j}(y), \mathbf{u}^-_{i+\frac{1}{2},j}(y), \mathbf{u}^+_{i,j-\frac{1}{2}}(x), \mathbf{u}^-_{i,j+\frac{1}{2}}(x)$ as the traces of $\mathbf{u}_{ij}^n$ on the four edges in cell I_{ij}, respectively. More details can be found in [87]. For simplicity, if we consider a generic numerical solution on the whole computational domain at time level n, then the subscript ij will be omitted.

In this section, we only consider high-order schemes, and the one satisfied by the

cell averages can be written as

$$\overline{\mathbf{u}}_{ij}^{n+1} = \overline{\mathbf{u}}_{ij}^{n} + \frac{\Delta t}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \widehat{\mathbf{f}}\left(\mathbf{u}^{-}_{i-\frac{1}{2},j}(y), \mathbf{u}^{+}_{i-\frac{1}{2},j}(y)\right) - \widehat{\mathbf{f}}\left(\mathbf{u}^{-}_{i+\frac{1}{2},j}(y), \mathbf{u}^{+}_{i+\frac{1}{2},j}(y)\right) dy$$
$$+ \frac{\Delta t}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \widehat{\mathbf{g}}\left(\mathbf{u}^{-}_{i,j-\frac{1}{2}}(x), \mathbf{u}^{+}_{i,j-\frac{1}{2}}(x)\right) - \widehat{\mathbf{g}}\left(\mathbf{u}^{-}_{i,j+\frac{1}{2}}(x), \mathbf{u}^{+}_{i,j+\frac{1}{2}}(x)\right) dx, \quad (4.35)$$

where $\widehat{\mathbf{f}}(\cdot,\cdot)$ and $\widehat{\mathbf{g}}(\cdot,\cdot)$ are one-dimensional numerical fluxes. For this problem, we still use the one-dimensional local Lax-Friedrichs flux.

Suppose $(x, y) = (x_{i-\frac{1}{2}}, y_0)$ is a point on the vertical cell interface, at which we have two numerical approximations $\mathbf{u}_\ell = (D_\ell, m_\ell, n_\ell, E_\ell)^T$ and $\mathbf{u}_r = (D_r, m_r, n_r, E_r)^T$ from left and right, respectively. Then the local Lax-Friedrichs flux can be written as

$$\widehat{\mathbf{f}}(\mathbf{u}_\ell, \mathbf{u}_r) = \frac{1}{2}\left(\mathbf{f}\left(\mathbf{u}_\ell\right) + \mathbf{f}\left(\mathbf{u}_r\right) - \alpha_f\left(\mathbf{u}_r - \mathbf{u}_\ell\right)\right),$$

where $\alpha_f \geq \max\{F_1(\mathbf{u}_\ell), F_1(\mathbf{u}_r)\}$ with F_1 is defined by

$$F_1(\mathbf{u}) = \frac{|w|(h+1-2h\tau)\gamma^2 + \sqrt{\tau^4(h-1)^2 + \tau^2(h-1)(h+1-2h\tau)}}{\gamma^2(h+1-2h\tau) + \tau^2(h-1)}, \quad (4.36)$$

and the constant $\tau = \frac{\Gamma-1}{\Gamma}$.

The numerical flux $\widehat{\mathbf{g}}$ can be defined in a similar way with the parameter α_g on the horizontal cell interfaces and the corresponding wave speed defined by

$$F_2(\mathbf{u}) = \frac{|v|(h+1-2h\tau)\gamma^2 + \sqrt{\tau^4(h-1)^2 + \tau^2(h-1)(h+1-2h\tau)}}{\gamma^2(h+1-2h\tau) + \tau^2(h-1)}, \quad (4.37)$$

We extend the definitions of $\mathbf{H}_j$ to two-dimensional problems and define

$$\mathbf{H}_1\left(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \lambda_1\right) = \mathbf{u}_2 + \lambda_1\left(\widehat{\mathbf{f}}\left(\mathbf{u}_1, \mathbf{u}_2\right) - \widehat{\mathbf{f}}\left(\mathbf{u}_2, \mathbf{u}_3\right)\right),$$

$$\mathbf{H}_2\left(\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3,\lambda_2\right)=\mathbf{u}_2+\lambda_2\left(\widehat{\mathbf{g}}\left(\mathbf{u}_1,\mathbf{u}_2\right)-\widehat{\mathbf{g}}\left(\mathbf{u}_2,\mathbf{u}_3\right)\right),$$

where $\lambda_1=\frac{\Delta t}{\Delta x}$ and $\lambda_2=\frac{\Delta t}{\Delta y}$.

Following the same proof of Theorem 4.2.1 with some minor changes, we have the following result

Lemma 4.3.1. *Suppose* $\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3\in G$, *then under a CFL condition* $\max\{\alpha_f^1,\alpha_f^2\}\lambda_1\leq 1$, *where* α_f^1 *is the parameter in the numerical flux* $\widehat{\mathbf{f}}(\mathbf{u}_1,\mathbf{u}_2)$ *and* α_f^2 *is the one for* $\widehat{\mathbf{f}}(\mathbf{u}_2,\mathbf{u}_3)$, *then we have*

$$\mathbf{H}_1(\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3,\lambda_1)\in G.$$

Similarly, under the CFL condition $\max\{\alpha_g^1,\alpha_g^2\}\lambda_2\leq 1$ *with* α_g^1 *and* α_g^2 *parameters for* $\widehat{\mathbf{g}}(\mathbf{u}_1,\mathbf{u}_2)$ *and* $\widehat{\mathbf{g}}(\mathbf{u}_2,\mathbf{u}_3)$, *respectively, then we have*

$$\mathbf{H}_2(\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3,\lambda_2)\in G.$$

To continue, we use L-point Gauss quadratures with $L\geq k+1$ for accuracy to approximate the integrals in (4.35). More details of this requirement can be found in [12]. The Gauss quadrature points on $\left[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}\right]$ and $\left[y_{j-\frac{1}{2}},y_{j+\frac{1}{2}}\right]$ are denoted by

$$p_i^x=\left\{x_i^\beta:\beta=1,\cdots,L\right\}\text{ and }p_j^y=\left\{y_j^\beta:\beta=1,\cdots,L\right\},$$

respectively. Also, we denote w_β as the corresponding weights on the interval $\left[-\frac{1}{2},\frac{1}{2}\right]$. Different from the notations in previous sections, we use

$$\hat{p}_i^x=\{\hat{x}_i^\alpha:\alpha=0,\cdots,M\}\text{ and }\hat{p}_j^y=\left\{\hat{y}_j^\alpha:\alpha=0,\cdots,M\right\}$$

as the Gauss-Lobatto points on $\left[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}\right]$ and $\left[y_{j-\frac{1}{2}},y_{j+\frac{1}{2}}\right]$, respectively. Also,

we denote ν_α as the corresponding weights on the interval $\left[-\frac{1}{2},\frac{1}{2}\right]$.

Then the numerical scheme (4.35) becomes

$$\begin{aligned}\overline{\mathbf{u}}_{ij}^{n+1} = \overline{\mathbf{u}}_{ij}^{n} &+ \lambda_1 \sum_{\beta=1}^{L} w_\beta \left[\widehat{\mathbf{f}}\left(\mathbf{u}^-_{i-\frac{1}{2},\beta}, \mathbf{u}^+_{i-\frac{1}{2},\beta}\right) - \widehat{\mathbf{f}}\left(\mathbf{u}^-_{i+\frac{1}{2},\beta}, \mathbf{u}^+_{i+\frac{1}{2},\beta}\right)\right] \\ &+ \lambda_2 \sum_{\beta=1}^{L} w_\beta \left[\widehat{\mathbf{g}}\left(\mathbf{u}^-_{\beta,j-\frac{1}{2}}, \mathbf{u}^+_{\beta,j-\frac{1}{2}}\right) - \widehat{\mathbf{g}}\left(\mathbf{u}^-_{\beta,j+\frac{1}{2}}, \mathbf{u}^+_{\beta,j+\frac{1}{2}}\right)\right], \end{aligned} \tag{4.38}$$

where $\mathbf{u}^-_{i-\frac{1}{2},\beta} = \mathbf{u}^-_{i-\frac{1}{2},j}(y_j^\beta)$ is a point value in the Gauss quadrature. Likewise for the other point values. As the general treatment, we rewrite the cell average on the right hand side as

$$\overline{\mathbf{u}}_{ij}^{n} = \sum_{\alpha=0}^{M}\sum_{\beta=1}^{L} \nu_\alpha w_\beta \mathbf{u}^1_{\alpha\beta} = \sum_{\alpha=0}^{M}\sum_{\beta=1}^{L} \nu_\alpha w_\beta \mathbf{u}^2_{\beta\alpha},$$

where $\mathbf{u}^1_{\alpha\beta}$ and $\mathbf{u}^2_{\beta\alpha}$ denote $\mathbf{u}^n_{ij}(\hat{x}_i^\alpha, y_j^\beta)$ and $\mathbf{u}^n_{ij}(x_i^\beta, \hat{y}_j^\alpha)$, respectively.

For each vertical edge, use $\alpha_f^{i-\frac{1}{2},\beta}$ to denote the parameter of the numerical flux $\widehat{\mathbf{f}}(\mathbf{u}^-_{i-\frac{1}{2},\beta}, \mathbf{u}^+_{i-\frac{1}{2},\beta})$ and for the horizontal edge use $\alpha_g^{\beta,j-\frac{1}{2}}$ for the parameter of $\widehat{\mathbf{g}}(\mathbf{u}^-_{\beta,j-\frac{1}{2}}, \mathbf{u}^+_{\beta,j-\frac{1}{2}})$. Then define

$$\alpha_f^{i-\frac{1}{2},j} = \max_{\beta=1,\cdots,L} \alpha_f^{i-\frac{1}{2},\beta}, \quad \alpha_g^{i,j-\frac{1}{2}} = \max_{\beta=1,\cdots,L} \alpha_g^{\beta,j-\frac{1}{2}}.$$

Let $\mu = \max_{i,j} \alpha_f^{i-\frac{1}{2},j}\lambda_1 + \max_{i,j} \alpha_g^{i,j-\frac{1}{2}}\lambda_2$, then scheme (4.38) can be written as

$$\begin{aligned}\overline{\mathbf{u}}_{ij}^{n+1} = C_1 \sum_{\beta=1}^{L} w_\beta &\left(\sum_{\alpha=1}^{M-1} \nu_\alpha \mathbf{u}^1_{\alpha\beta} + \nu_0 \mathbf{H}_1\left(\mathbf{u}^-_{i-\frac{1}{2},\beta}, \mathbf{u}^+_{i-\frac{1}{2},\beta}, \mathbf{u}^-_{i+\frac{1}{2},\beta}, \frac{\lambda_1}{\nu_0 C_1}\right)\right. \\ &\left. + \nu_M \mathbf{H}_1\left(\mathbf{u}^+_{i-\frac{1}{2},\beta}, \mathbf{u}^-_{i+\frac{1}{2},\beta}, \mathbf{u}^+_{i+\frac{1}{2},\beta}, \frac{\lambda_1}{\nu_M C_1}\right)\right) \\ + C_2 \sum_{\beta=1}^{L} w_\beta &\left(\sum_{\alpha=1}^{M-1} \nu_\alpha \mathbf{u}^2_{\beta\alpha} + \nu_0 \mathbf{H}_2\left(\mathbf{u}^-_{\beta,j-\frac{1}{2}}, \mathbf{u}^+_{\beta,j-\frac{1}{2}}, \mathbf{u}^-_{\beta,j+\frac{1}{2}}, \frac{\lambda_2}{\nu_0 C_2}\right)\right.\end{aligned}$$

$$+\nu_M \mathbf{H}_2\left(\mathbf{u}^+_{\beta,i-\frac{1}{2}}, \mathbf{u}^-_{\beta,i+\frac{1}{2}}, \mathbf{u}^+_{\beta,j+\frac{1}{2}}, \frac{\lambda_2}{\nu_M C_2}\right)\Bigg), \tag{4.39}$$

where

$$C_1 = \frac{\max_{i,j} \alpha_f^{i-\frac{1}{2},j}\lambda_1}{\mu}, \quad C_2 = \frac{\max_{i,j} \alpha_g^{i,j-\frac{1}{2}}\lambda_2}{\mu}.$$

In (4.39), $\overline{\mathbf{u}}_{ij}^{n+1}$ is the convex combination of $\mathbf{u}$, $\mathbf{H_1}$ and $\mathbf{H_2}$. Therefore, we have the following theorem.

Theorem 4.3.1. *Suppose* $\mathbf{u}^n \in G$ *in scheme* (4.38)*, then* $\overline{\mathbf{u}}^{n+1} \in G$*, under the CFL condition*

$$\frac{\Delta t}{\Delta x}\max_{i,j}\alpha_f^{i-\frac{1}{2},j} + \frac{\Delta t}{\Delta y}\max_{i,j}\alpha_g^{i,j-\frac{1}{2}} \le \nu_0. \tag{4.40}$$

Remark 4.3.1. *It is straightforward to obtain the bound that*

$$F_1(\mathbf{u}) \le 1, \quad F_2(\mathbf{u}) \le 1, \quad \forall \mathbf{u} \in G.$$

In practice, we can replace $\max_{i,j} \alpha_f^{i-\frac{1}{2},j}$ *and* $\max_{i,j} \alpha_g^{i,j-\frac{1}{2}}$ *by* 1 *to obtain the CFL condition.*

Based on the above theorem, the numerical cell average we obtain is in G. Of course, the numerical solution $\mathbf{u}_{ij}^{n+1}$ might still be placed outside. Hence, we have to modify the numerical solution while keeping the cell average untouched. Due to the rounding error, we define

$$G^\varepsilon = \left\{ \mathbf{u} = \begin{pmatrix} D \\ m \\ n \\ E \end{pmatrix} : D \ge \varepsilon, E \ge \sqrt{m^2+n^2+D^2+\varepsilon} \right\},$$

$$\partial G^{\varepsilon} = \left\{ \mathbf{u} = \begin{pmatrix} D \\ m \\ n \\ E \end{pmatrix} : D = \varepsilon, E = \sqrt{D^2 + m^2 + n^2 + \varepsilon} \right\}.$$

Then the modification of $\mathbf{u}_{ij}^n$ is given in the following steps.

- Set up a small number $\varepsilon = 10^{-13}$.
- If $\overline{D}_{ij}^n > \varepsilon$, then proceed to the following steps. Otherwise, D_{ij}^n is identified as the approximation to vacuum. We take $\widetilde{\mathbf{u}}_{ij}^n = \overline{\mathbf{u}}_{ij}^n$ as the numerical solution and skip the following steps.
- Modify the density: Compute $b_{ij} = \min_{\alpha\beta} \left\{ D_{ij}^n(\widehat{x}_i^{\alpha}, y_j^{\beta}), D_{ij}^n(x_i^{\beta}, \widehat{y}_j^{\alpha}) \right\}$. If $b_{ij} < \varepsilon$, then take $\widetilde{D}_{ij}^n$ as

  $$\widetilde{D}_{ij}^n = \overline{D}_{ij}^n + \theta_{ij}^D \left(D_{ij}^n - \overline{D}_{ij}^n \right),$$

  with

  $$\theta_{ij}^D = \frac{\overline{D}_{ij}^n - \varepsilon}{\overline{D}_{ij}^n - b_{ij}},$$

  and use $\widetilde{D}_{ij}^n$ as the new numerical density D_{ij}^n.
- Enforce $E_{ij}^n \geq \sqrt{(m_{ij}^n)^2 + (n_{ij}^n)^2 + (\tilde{D}_{ij}^n)^2 + \epsilon}$ in each cell I_{ij}: Consider $\mathbf{u}_{\alpha\beta}^1$ and $\mathbf{u}_{\beta\alpha}^2$ in the cell I_{ij}, respectively. If $\mathbf{u}_{\alpha\beta}^1 \in G^{\varepsilon}$, then take $\theta_{\alpha\beta}^1 = 1$. Otherwise, take $\theta_{\alpha\beta}^1$ to be the root of the equation

  $$\delta_2 \left((1-t)\overline{\mathbf{u}}_{ij}^n + t\mathbf{u}_{\alpha\beta}^1 \right) = 0, \quad t \in (0,1). \tag{4.41}$$

  where $\delta_2(\mathbf{u}) = E - \sqrt{m^2 + n^2 + D^2 + \varepsilon}$. Alternatively, the t can be obtained in the same way as in Remark 4.2.4. Similarly, we can define $\theta_{\beta\alpha}^2$ in the same

way for $\mathbf{u}^2_{\beta\alpha}$. Finally, we use

$$\widetilde{\mathbf{u}}^n_{ij} = \overline{\mathbf{u}}^n_{ij} + \theta(\mathbf{u}^n_{ij} - \overline{\mathbf{u}}^n_{ij}), \quad \theta = \min_{\alpha,\beta}\left\{\theta^1_{\alpha\beta}, \theta^2_{\beta\alpha}\right\},$$

as the DG approximation in cell I_{ij}.

Now we demonstrate the L^1-stability of the bound-preserving DG scheme. Following the same analysis as in Section 4.2, we have

$$\|D^n_h\|_{L^1} = \|D^0_h\|_{L^1} \quad \text{and} \quad \|E^n_h\|_{L^1} = \|E^0_h\|_{L^1}.$$

It is easy to check that

$$m = \frac{w\Gamma}{\Gamma - \sigma}E \quad \text{and} \quad n = \frac{v\Gamma}{\Gamma - \sigma}E,$$

where $\sigma = \frac{(\gamma-1)(h-1)}{h\gamma^2} \leq \Gamma - 1$. Therefore,

$$\|m\|_{L^1} + \|n\|_{L^1} \leq \frac{(|w| + |v|)\Gamma}{\Gamma - \sigma}\|E\|_{L^1} \leq \sqrt{2}\Gamma\|E\|_{L^1}.$$

In the last inequality, we use the fact that $w^2 + v^2 \leq 1$.

The above analysis yields the L^1-stability of the scheme:

$$\|\mathbf{u}^n_h\|_{L^1} \leq \|D^n_h\|_{L^1} + (\sqrt{2}\Gamma+1)\|E^n_h\|_{L^1} = \|D^0_h\|_{L^1} + (\sqrt{2}\Gamma+1)\|E^0_h\|_{L^1} \leq (\sqrt{2}\Gamma+1)\|\mathbf{u}^0_h\|_{L^1},$$

where $\|\mathbf{u}\|_{L^1} = \|D\|_{L^1} + \|m\|_{L^1} + \|n\|_{L^1} + \|E\|_{L^1}$.

4.4 Application to relativistic jets

In this section, we study the relativistic axisymmetric jets which is described in the two-dimensional cylindrical coordinates (r, z). We adopt the governing system in [39] and multiply both sides of the system by r. Then it reads

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_r + \mathbf{g}(\mathbf{u})_z = \mathbf{s}(r, \mathbf{u}), \tag{4.42}$$

with

$$\mathbf{u} = \begin{pmatrix} rD \\ rm \\ rn \\ rE \end{pmatrix}, \ \mathbf{f}(\mathbf{u}) = \begin{pmatrix} rDu \\ rmu + rp \\ rnu \\ rm \end{pmatrix}, \ \mathbf{g}(\mathbf{u}) = \begin{pmatrix} rDv \\ rmv \\ rnv + rp \\ rn \end{pmatrix}, \ \mathbf{s} = \begin{pmatrix} 0 \\ p \\ 0 \\ 0 \end{pmatrix}. \tag{4.43}$$

We would like to mention that this formulation is not unique. We could also combine the p term in the source and the rp term in the momentum flux to form a non-conservative rp_r term, which is closer to the underlying physics. However, since our method is based on the conservative form of the equation we will stay with (4.42).

Different from what we have discussed in Section 4.3, the source term $\mathbf{s}(r, \mathbf{u})$ is not zero. Therefore, we will demonstrate how to discretize the source terms. The techniques for the discretization of the flux terms and the modification of the numerical approximations follow from the same lines as discussed in Section 4.3, and we will omit them.

In this section, the cell is defined as $I_{ij} = \left[r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}\right] \times \left[z_{j-\frac{1}{2}}, z_{j+\frac{1}{2}}\right]$, with

$1 \le i \le N_r$, $1 \le j \le N_z$ and the mesh sizes in r and z directions are denoted as Δr and Δz, respectively. If not otherwise stated in this section, the notations follow those in Section 4.3. We consider high order schemes only, the one satisfied by the cell averages can be written as

$$\begin{aligned}\overline{\mathbf{u}}_{ij}^{n+1} = & \frac{1}{2}\overline{\mathbf{u}}_{ij}^{n} + \frac{\Delta t}{\Delta r \Delta z}\int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} \widehat{\mathbf{f}}\left(\mathbf{u}^{-}_{i-\frac{1}{2},j}(z), \mathbf{u}^{+}_{i-\frac{1}{2},j}(z)\right) - \widehat{\mathbf{f}}\left(\mathbf{u}^{-}_{i+\frac{1}{2},j}(z), \mathbf{u}^{+}_{i+\frac{1}{2},j}(z)\right) dz \\ & + \frac{\Delta t}{\Delta r \Delta z}\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \widehat{\mathbf{g}}\left(\mathbf{u}^{-}_{i,j-\frac{1}{2}}(r), \mathbf{u}^{+}_{i,j-\frac{1}{2}}(r)\right) - \widehat{\mathbf{g}}\left(\mathbf{u}^{-}_{i,j+\frac{1}{2}}(r), \mathbf{u}^{+}_{i,j+\frac{1}{2}}(r)\right) dr, \\ & + \frac{1}{2}\overline{\mathbf{u}}_{ij}^{n} + \Delta t \overline{\mathbf{s}}_{ij}^{n} \end{aligned} \tag{4.44}$$

where $\overline{\mathbf{s}}_{ij}^{n}$ is the cell average of $\mathbf{s}$ in the cell I_{ij} at time level n. We use L-point Gauss quadratures with $L \ge \frac{k+1}{2}$ to approximate $\overline{\mathbf{s}}_{ij}^{n}$. The Gauss quadrature points on $\left[r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}\right]$ and $\left[z_{j-\frac{1}{2}}, z_{j+\frac{1}{2}}\right]$ are denoted by

$$p_i^r = \left\{r_i^\beta : \beta = 1, \cdots, L\right\} \text{ and } p_j^z = \left\{z_j^\beta : \beta = 1, \cdots, L\right\},$$

respectively. Also, we denote w_β as the corresponding weights on the interval $\left[-\frac{1}{2}, \frac{1}{2}\right]$. Then

$$\frac{1}{2}\overline{\mathbf{u}}_{ij}^{n} + \Delta t \overline{\mathbf{s}}_{ij}^{n} = \sum_{\alpha=1}^{L}\sum_{\beta=1}^{L} w_\alpha w_\beta \mathbf{H}_s(\mathbf{u}_{ij}^{\alpha\beta}, \mathbf{s}_{ij}^{\alpha\beta}, \Delta t),$$

where $\mathbf{u}_{ij}^{\alpha\beta} = \mathbf{u}_{ij}^{n}(r_i^\alpha, z_j^\beta)$, $\mathbf{s}_{ij}^{\alpha\beta} = \mathbf{s}_{ij}^{n}(r_i^\alpha, \mathbf{u}_{ij}^{\alpha\beta})$ and $\mathbf{H}_s(\mathbf{u}, \mathbf{s}, \Delta t) = \frac{1}{2}\mathbf{u} + \Delta t \mathbf{s}$. Since $\mathbf{s}$ is a function of $\mathbf{u}$ and r, $\mathbf{H}_s$ can be written as a function of $\mathbf{u}$, r and Δt, i.e. $\widetilde{\mathbf{H}}_s(\mathbf{u}, r, \Delta t) = \mathbf{H}_s(\mathbf{u}, \mathbf{s}, \Delta t)$. We can choose Δt sufficiently small, such that $\widetilde{\mathbf{H}}_s(\mathbf{u}, r, \Delta t) = \mathbf{H}_s(\mathbf{u}, \mathbf{s}, \Delta t) \in G$, and the result is given in the following lemma.

Lemma 4.4.1. *Suppose $\mathbf{u} \in G$, if the time mesh size Δt satisfies*

$$\Delta t \le \frac{r}{2}\widetilde{\alpha}(\mathbf{u}, r), \tag{4.45}$$

where

$$\widetilde{\alpha}(\mathbf{u},r)=\frac{\sqrt{(w\Gamma h\gamma^2)^2-(\Gamma-1)^2(h-1)^2-\Gamma^2\gamma^2(h-1)^2+2\Gamma h\gamma^2(h-1)}-uh\Gamma\gamma^2}{(\Gamma-1)(h-1)},$$

then $\widetilde{\mathbf{H}}_s(\mathbf{u},r,\Delta t)\in G$.

Proof. It is easy to see that

$$\widetilde{\mathbf{H}}_s(\mathbf{w},r,\Delta t)=(rD,rm+\Delta tp,rn,rE)^T.$$

Define

$$q(\mathbf{w},r,\Delta t)=(rE)^2-(rD)^2-(rm+\Delta tp)^2-(rn)^2.$$

Since $r>0$, we only need to find Δt such that $q(\mathbf{w},r,\Delta t)\geq 0$. It is not difficult to check that

$$E=\left(\frac{\Gamma}{\sigma}-1\right)p \tag{4.46}$$

$$m=\frac{u\Gamma}{\sigma}p \tag{4.47}$$

$$n=\frac{v\Gamma}{\sigma}p \tag{4.48}$$

$$D=\frac{\gamma\Gamma}{(\Gamma-1)(h-1)}p, \tag{4.49}$$

where $\sigma=\frac{(\Gamma-1)(h-1)}{h\gamma^2}$. Therefore, we have

$$\begin{aligned}
q(\mathbf{w},r,\Delta t)&=(rE)^2-(rD)^2-(rm+\Delta tp)^2-(rn)^2\\
&=p^2\left[\left(\frac{r\Gamma}{\sigma}-r\right)^2-\left(\frac{r\gamma\Gamma}{(\Gamma-1)(h-1)}\right)^2-\left(\frac{u\Gamma r}{\sigma}+\Delta t\right)^2-\left(\frac{r\Gamma v}{\sigma}\right)^2\right]\\
&=r^2p^2\left[\frac{\Gamma^2}{\sigma^2\gamma^2}+1-\frac{2\Gamma}{\sigma}-\frac{\Delta t^2}{r^2}-\frac{2u\Gamma}{\sigma}\frac{\Delta t}{r}-\left(\frac{\Gamma\gamma}{(\Gamma-1)(h-1)}\right)^2\right]\\
&=-r^2p^2\left(\frac{\Delta t^2}{r^2}+2A\frac{\Delta t}{r}+B\right),
\end{aligned}$$

where

$$A = \frac{u\Gamma h\gamma^2}{(\Gamma-1)(h-1)}$$

and

$$B = \frac{\Gamma h\gamma^2(\Gamma-2) - \Gamma^2\gamma^2}{(\Gamma-1)^2(h-1)} - 1 < -1$$

Let $q(\mathbf{w}, r, \Delta t) \geq 0$, we need

$$\Delta t \leq (\sqrt{A^2 - B} - A)r. \tag{4.50}$$

□

With the above lemma, we can state the following theorem

Theorem 4.4.1. *Suppose* $\mathbf{u}^n \in G$, *then* $\overline{\mathbf{u}}^{n+1} \in G$ *under the conditions*

$$\Delta t \leq \max_{\substack{1 \leq \alpha, \beta \leq L \\ 1 \leq i \leq N_r \\ 1 \leq j \leq N_z}} \frac{r_i^\alpha}{2} \widetilde{\alpha}(\mathbf{u}_{ij}^{\alpha\beta}, r_i^\alpha),$$

and

$$\frac{\Delta t}{\Delta r} \max_{i,j} \alpha_f^{i-\frac{1}{2},j} + \frac{\Delta t}{\Delta z} \max_{i,j} \alpha_g^{i,j-\frac{1}{2}} \leq \frac{\nu_0}{2}.$$

where $\alpha_f^{i-\frac{1}{2},j}$ *and* $\alpha_g^{i,j-\frac{1}{2}}$ *have the same definition as those in (4.40).*

Now we have finished all the theoretical analysis.

4.5 Numerical experiments

In this section, we present numerical examples in both one and two dimensions to verify the bound-preserving property of the proposed method. In order to demonstrate the effectiveness of the proposed BP limiter, we have deliberately not applied any other non-oscillatory limiters, such as the TVD/TVB or WENO limiters. As expected from the theory, the numerical solution stays in the physical bounds and computations could proceed stably, even though there are spurious oscillations in some test results due to the lack of non-oscillatory limiters. We should emphasize that this bound-preserving limiter, which is extremely local (implemented completely inside each cell without using information from neighboring cells) and inexpensive, is not meant to substitute other non-oscillatory limiters or techniques. It can be used together with other non-oscillatory limiters or techniques, such as TVD/TVB or WENO limiters, or even WENO finite volume schemes, to obtain both bound-preserving and non-oscillatory performance, but we will not pursue such approach here in order to be focused on the bound-preserving property. We would also like to emphasize that many of the examples are chosen to contain solutions particularly challenging in terms of the bound-preserving requirement, such that even with the usual TVD/TVB or WENO limiters added, the code would still fail without the proposed bound-preserving technique, due to the breaking of the physical bound which causes ill-posedness of the problem as well as the nonexistence of the solution to (4.9). Unless otherwise indicated, we consider the third-order RKDG method ($k = 2$) with the local Lax-Friedrichs numerical flux (4.12). The CFL number is set to be 0.15. The specific heat ratio Γ is taken to be 5/3, unless otherwise stated.

4.5.1 One-dimensional experiments

Example 4.5.1.1 (Smooth flow)**.** Consider a one-dimensional flow in the domain $\Omega = [0, 1]$ with the following initial states

$$\rho_0(x) = 1 + 0.9999999 \sin(2\pi x), \quad u_0(x) = 0.9, \quad p_0(x) = 1.0$$

The boundary condition is set to be periodic. The exact solution is

$$\rho(x, t) = 1 + 0.9999999 \sin(2\pi(x - 0.9t)), \quad w(x, t) = 0.9, \quad p(x, t) = 1.0$$

In Table 4.1, we present the numerical results for the proposed method with and without the bound-preserving limiter. For the case $k = 3$, we take $\Delta t = \text{CFL}(\Delta x)^{4/3}$ so that the time error will not dominate. Since this is a high speed smooth flow with its lowest density near zero, the bound-preserving limiter does get turned on. As observed and discussed in [90] and [79], the SSP Runge-Kutta (RK) methods might degenerate the accuracy when the bound-preserving limiter is applied, while for the SSP multi-step (M-S) method, the full order of accuracy is recovered.

Moreover, in Figure 4.2, we plot the CPU time against the L^2 error for polynomial degrees $k = 1, 2, 3$ with and without the BP limiter. The following two observations could be made: (i) As one of the advantages of high order DG methods, for a given error, the computational time spent is diminishing with increasing order; (ii) The additional cost introduced by the BP limiter is very small.

Example 4.5.1.2 (Moderate blast wave)**.** The relativistic blast wave problems are standard tests (see [38]) for a numerical relativistic hydrodynamical code. The computational domain is $\Omega = [0, 1]$. We first consider a moderate case, which has the

		no limiter		with limiter (RK)		with limiter (M-S)	
k	h	L^2 error	order	L^2 error	order	L^2 error	order
1	1/20	3.50 e-2	–	4.85 e-2	–	4.55 e-2	–
	1/40	8.72 e-3	2.00	1.06 e-2	2.11	1.05 e-2	2.12
	1/80	2.17 e-3	2.01	2.55 e-3	2.05	2.55 e-3	2.03
	1/160	5.43 e-4	2.00	6.13 e-4	2.03	6.11 e-4	2.06
	1/320	1.36 e-4	2.00	1.50 e-4	2.04	1.49 e-4	2.03
2	1/20	2.16 e-3	–	3.47 e-3	–	2.40 e-3	–
	1/40	2.77 e-4	2.97	5.05 e-4	2.78	2.79 e-4	3.10
	1/80	3.48 e-5	2.99	9.42 e-5	2.42	3.48 e-5	3.00
	1/160	4.35 e-6	3.00	1.87 e-5	2.33	4.35 e-6	3.00
	1/320	5.44 e-7	3.00	3.68 e-6	2.35	5.44 e-7	3.00
3	1/20	8.99 e-5	–	1.82 e-4	–	1.52 e-4	–
	1/40	5.60 e-6	4.00	2.09 e-5	3.12	5.82 e-6	8.03
	1/80	3.46 e-7	4.02	2.30 e-6	3.18	3.50 e-7	4.05
	1/160	2.16 e-8	4.00	2.70 e-7	3.09	2.19 e-8	4.00
	1/320	1.35 e-9	4.00	1.86 e-8	3.86	1.46 e-9	3.91

Table 4.1: Example 4.5.1.1: One-dimensional accuracy test at $T = 0.4$ for the second-, third- and fourth-order DG methods with and without the BP limiters. The CFL number for the multistep method is one-third of that for the RK method. k is the degree of polynomial and h is the meshsize.

initial states

$$(\rho_0, u_0, p_0) = \begin{cases} (10.0, 0.0, 13.33), & x < 0.5 \\ (1.0, 0.0, 0.0), & x > 0.5 \end{cases} \tag{4.51}$$

The initial discontinuity gives rise to a transonic rarefaction wave propagating left, a shock wave propagating right and a contact discontinuity in between. The numerical approximation in Figure 4.3 shows that our method can resolve these structures quite well. Small oscillations are observed around the contact discontinuity, since we have not applied any non-oscillatory limiters such as the TVD/TVB or WENO limiters. For the same reason, in some of the following examples, we also observe spurious oscillations due to the lack of non-oscillatory limiters.

Example 4.5.1.3 (Strong relativistic blast wave)**.** Next we examine a much more

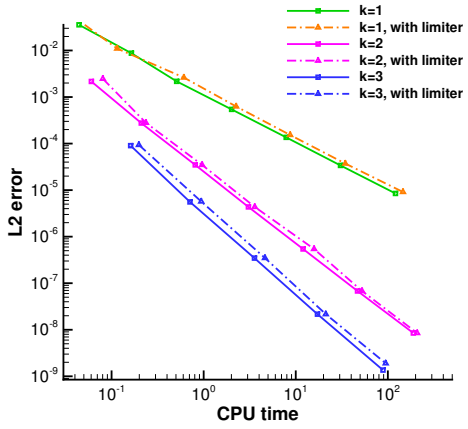


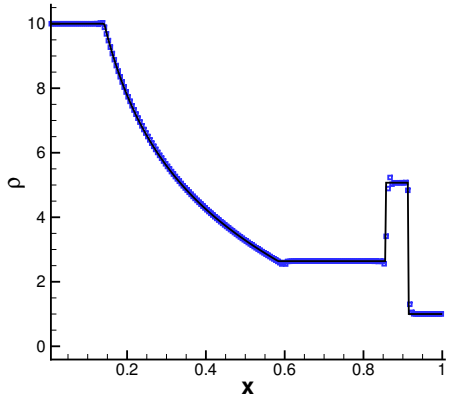
Figure 4.2: Example 4.5.1.1: Log-log plot of the CPU time against the L^2 error for polynomial degrees $k = 1, 2, 3$ with and without the BP limiter.

challenging blast wave example, of which the initial states are

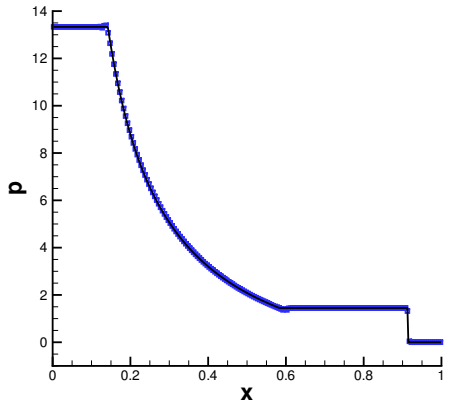
$$(\rho_0, u_0, p_0) = \begin{cases} (1.0, 0.0, 10^3), & x < 0.5 \\ (1.0, 0.0, 0.0), & x > 0.5 \end{cases} \tag{4.52}$$

This example is more relativistic than the previous one. The high relativity is due to the large enthalpy of the left state, which is $h \simeq 2.5 \times 10^3 \gg 1$. This results in a thermodynamically relativistic configuration. The structure of the solution is the same as the moderate case, except for the formation of a very thin dense shell behind the shock in the density and a highly curved profile for the rarefaction fan in the velocity. The relativistic shock propagating at a Lorentz factor $\gamma \simeq 6$ [84]. In Figure 4.4, despite of small oscillations, we can see that our method resolves the curved part in the profile of the velocity very well and captures the thin shell in the density with little smearing.

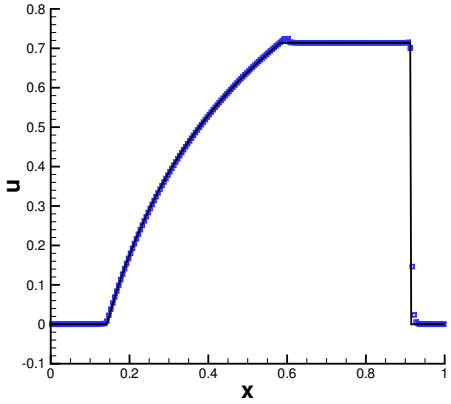
To further test the bound preserving property of our method, we consider a more



(a) density ρ



(b) pressure p



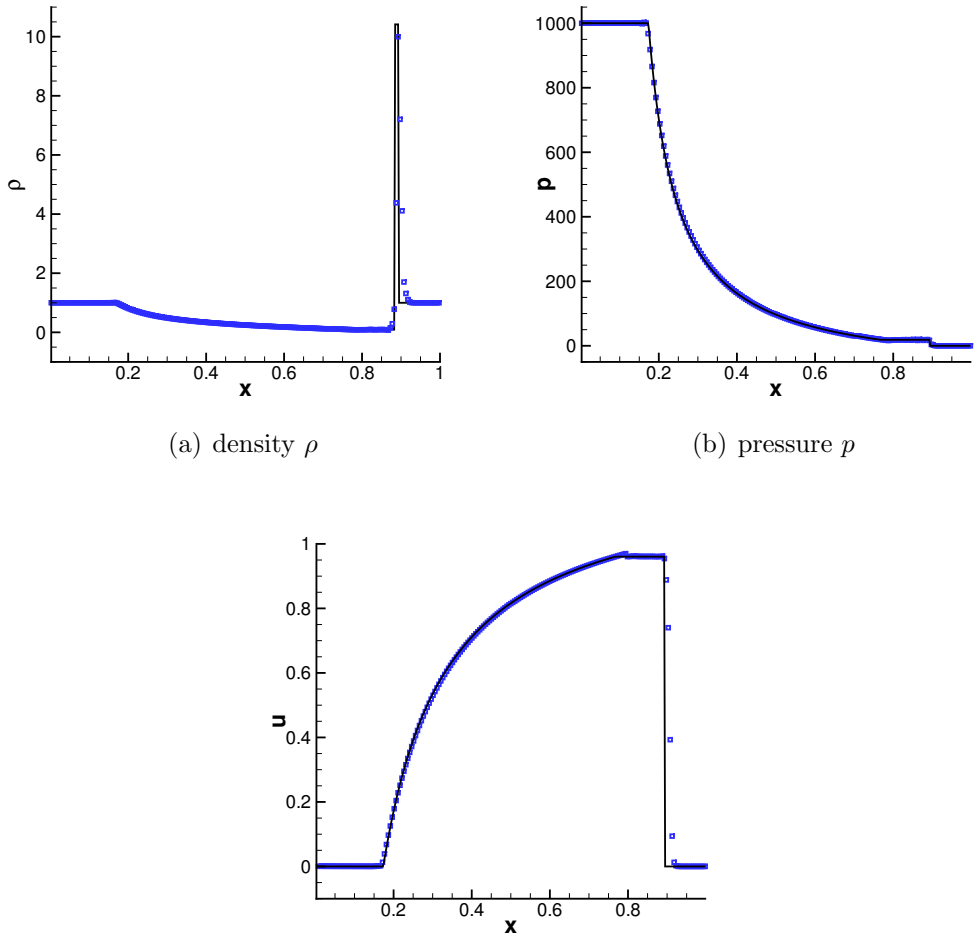
(c) velocity w

Figure 4.3: Example 4.5.1.2: Blast wave with initial condition (4.51) at $T = 0.5$ on the mesh of 200 cells, approximated by the third-order RKDG method with the BP limiter. The squares represent the approximate cell averages and the solid line is the exact solution.

extreme example. The initial condition now is

$$(\rho_0, u_0, p_0) = \begin{cases} (1.0, 0.0, 10^4), & x < 0.5 \\ (1.0, 0.0, 0.0), & x > 0.5 \end{cases} \tag{4.53}$$

with the specific heat ratio $\Gamma = 4/3$. In this case the enthalpy of the left state is $h \simeq 4 \times 10^4$. In the profile of the density, the width of the thin shell is approximately



(a) density ρ

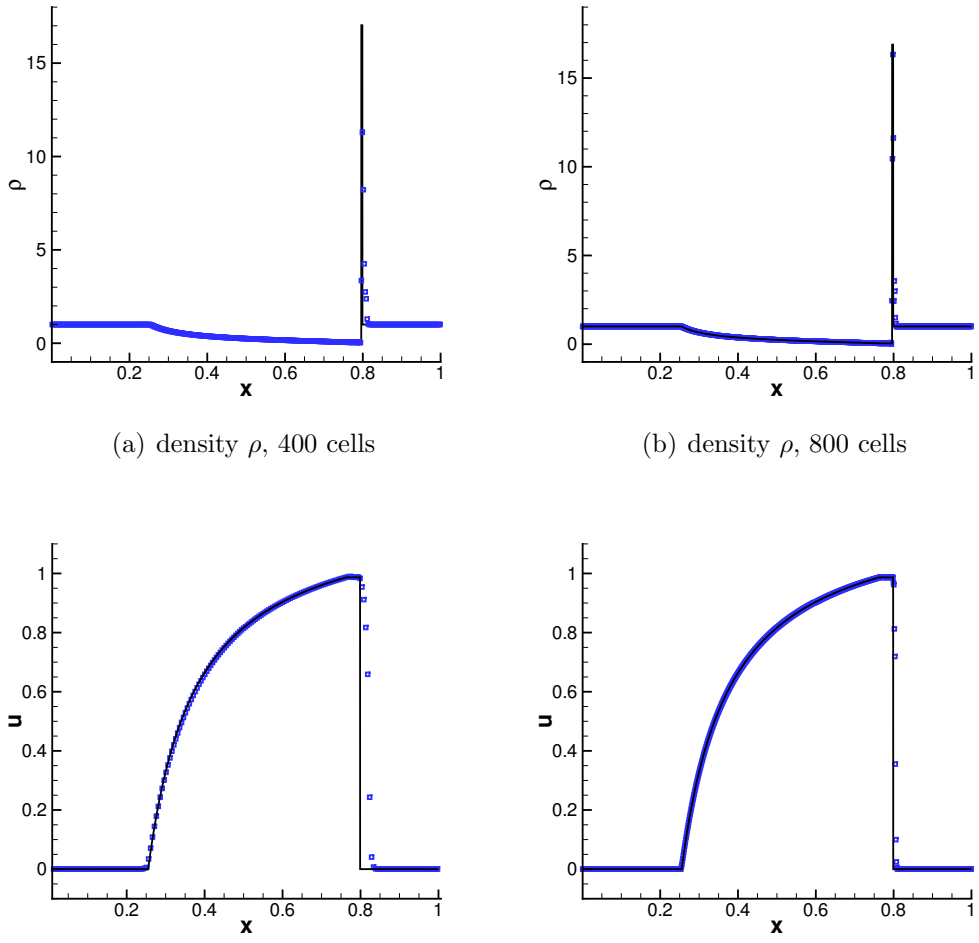
(b) pressure p

(c) velocity w

Figure 4.4: Example 4.5.1.3: Strong blast wave with initial condition (4.52) at $T = 0.4$ approximated by the third-order RKDG method with the BP limiter on the mesh with 200 cells. The square represents the approximate cell averages and the solid line is the exact solution.

2.5×10^{-3} and the rarefaction part in the velocity becomes even more curved. In Figure 4.5, we can see the good performance of our proposed method.

Example 4.5.1.4 (Shock-heating problem)**.** The shock-heating problem (see [83, 84, 37, 38]) is a standard benchmark problem to examine the ability of the method to deal with strong shocks. A warm wall is located at $x = 0$ and cold gas flows in at $x = 1.0$. When the gas gets reflected on the wall, a reverse strong shock forms



(a) density ρ, 400 cells

(b) density ρ, 800 cells

(c) velocity w, 400 cells

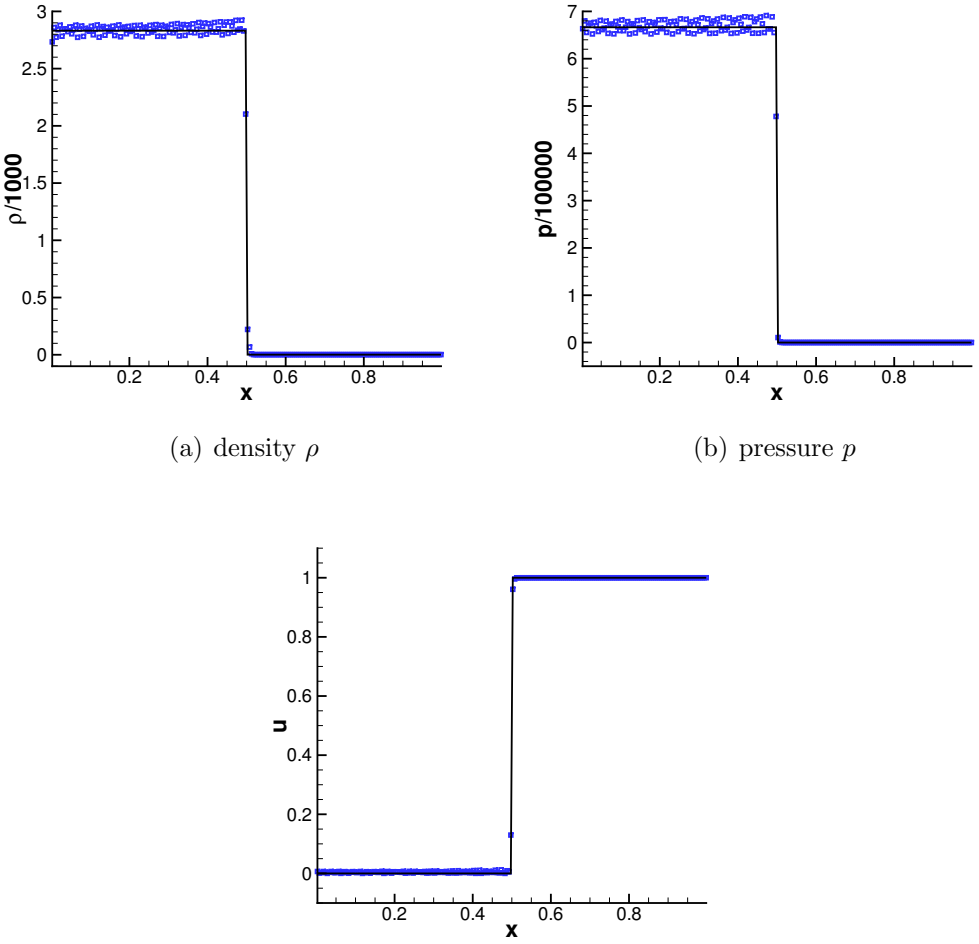
(d) velocity w , 800 cells

Figure 4.5: Example 4.5.1.3: Strong blast wave with initial condition (4.53) at $T = 0.3$ approximated by the third-order RKDG method with the BP limiter. The mesh is decomposed into 400 cells (left column) and 800 cells (right column). The square represents the approximate cell averages and the solid line is the exact solution.

and propagates to the right. In our experiment, the inflow velocity is set to be $v_{in} = 0.999999$, with the corresponding Lorentz factor $\gamma = 707.1$. The initial density is $\rho_0 = 1.0$ and the initial pressure $p_0 = 0.0$. The specific heat ratio in this example is $\Gamma = 4/3$. The analytical solution can be found in [37].

In Figure 4.6, we observe that, due to the BP limiter, near the physical bounds, there is no oscillation and the bounds are well preserved. The shocks are sharply

captured and the oscillations behind the shock may be eliminated with the non-oscillatory techniques such as the TVB or the WENO limiter. There is an undershoot in the left end point of the density, which is the so-called "wall-heating" effect. All these show the effectiveness of the proposed BP limiter in the ultra-relativistic regime with very strong shocks.



(a) density ρ

(b) pressure p

(c) velocity w

Figure 4.6: Example 4.5.1.4: One-dimensional shock-heating problem on the mesh of 200 cells, at $T = 1.5$ with $v_{in} = 0.999999$, approximated by the third-order RKDG method with the BP limiter. The squares represent the numerical results and the solid line is the exact solution.

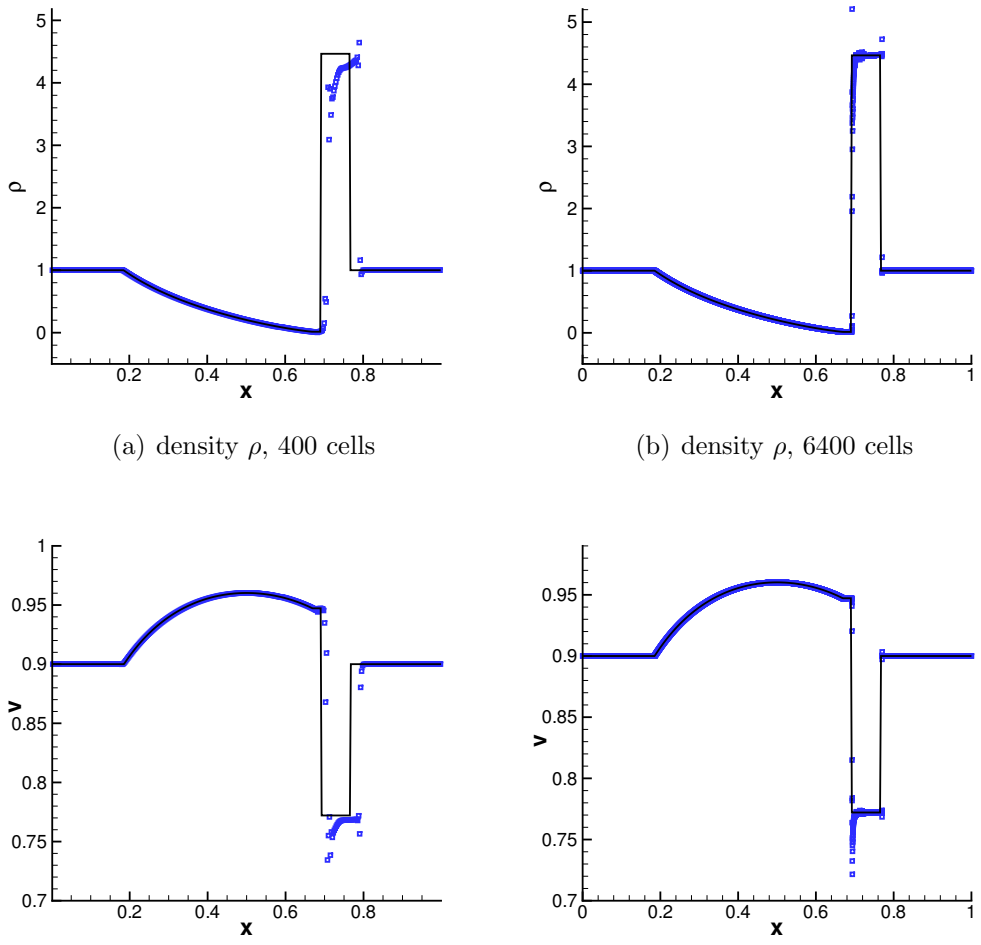
Example 4.5.1.5 (One-dimensional Riemann problem with non-zero transverse velocity)**.** The last one-dimensional test is the Riemann problem with non-zero trans-

verse velocity, see [84] and [42]. The domain is still $\Omega = [0, 1]$ and the initial states are

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (1.0, 0.0, 0.9, 10^3), & x < 0.5 \\ (1.0, 0.0, 0.9, 10^{-2}), & x > 0.5 \end{cases} \tag{4.54}$$

Compared with Example 4.5.1.3, where there is no transverse velocity, the density has a smaller jump and a wider dense shell, however the distance between the tail of the rarefaction and the contact discontinuity is much smaller, which requires very high resolution to resolve the structure. The purpose of this example is to show the robustness of the BP limiter even when the transverse speed is nonzero and close to the speed of light. Admittedly, to correctly restore the right wave speed, it is unavoidable to refine the mesh (see the comparison between 400 cells and 6400 cells in Figures 4.7 and 4.8 and also results in [84] with the adaptive mesh refinement technique), but this goes beyond the out purpose and we will not pursue it here. Our results on meshes of 400 and 6400 cells are presented in Figure 4.7 and Figure 4.8. These results match those in [84] and [42] well.

We further test this problem with a transverse velocity $v = 0.999$. In this case, the dense shell in the density is much thinner and the transverse velocity increases from $v = 0.999$ to $v = 0.99967$ at the rarefaction part, which corresponds to a Lorentz factor $\gamma \simeq 39$. We see in Figure 4.9, our method is still robust in this severe case.



(a) density ρ, 400 cells

(b) density ρ, 6400 cells

(c) velocity v, 400 cells

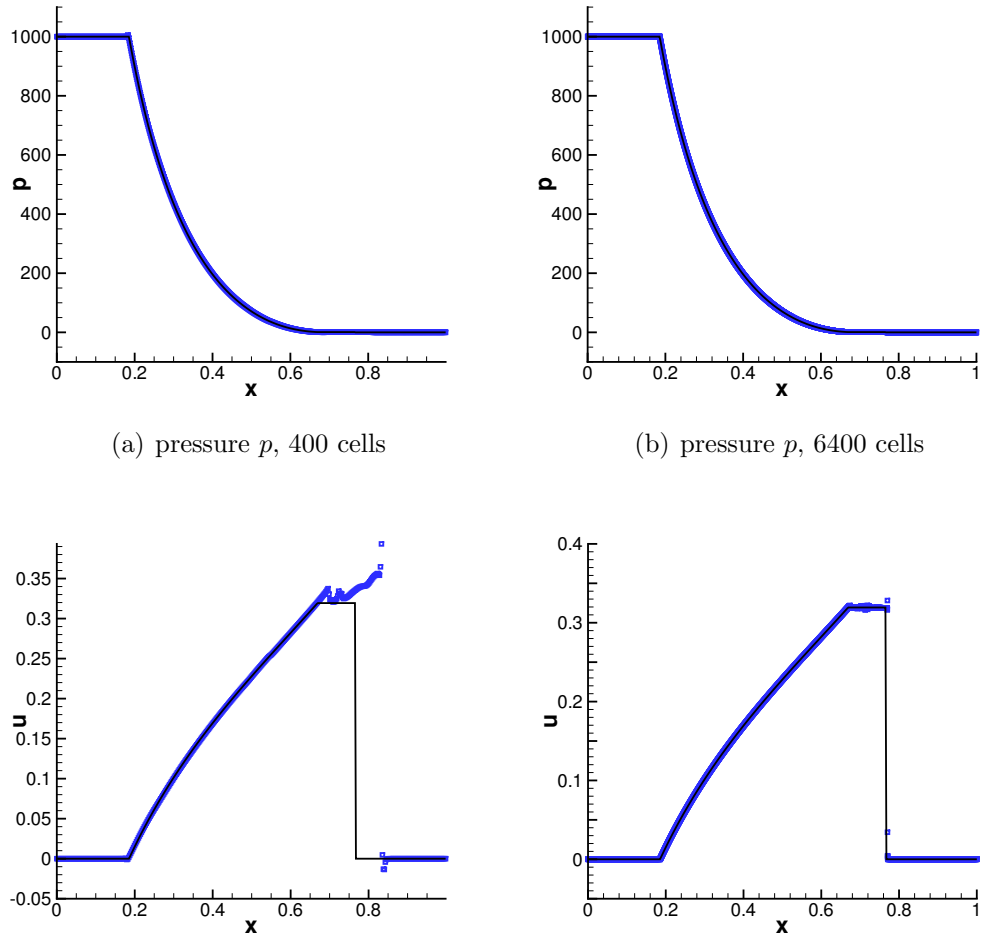
(d) velocity v, 6400 cells

Figure 4.7: Example 4.5.1.5: One-dimensional Riemann problem with non-zero transverse velocity at $T = 0.6$, approximated by the third-order RKDG method with the BP limiter. Approximations of the density and the transverse velocity on meshes of 400 and 6400 cells. The squares represent the approximate cell averages and the solid lines are the exact solutions.

4.5.2 Two-dimensional experiments

Example 4.5.2.1 (Smooth flow)**.** First, the accuracy of the method in 2-D is checked by considering the following smooth flow in the domain $\Omega = [0, \sqrt{2}]^2$.

$$\rho = 1 + 0.999999 \sin[2\pi(\mathbf{r} - \mathbf{v}t) \cdot \mathbf{b}], \quad v_x = 0.9, \quad v_y = 0.2, \quad p = 1.0 \tag{4.55}$$



(a) pressure p, 400 cells

(b) pressure p, 6400 cells

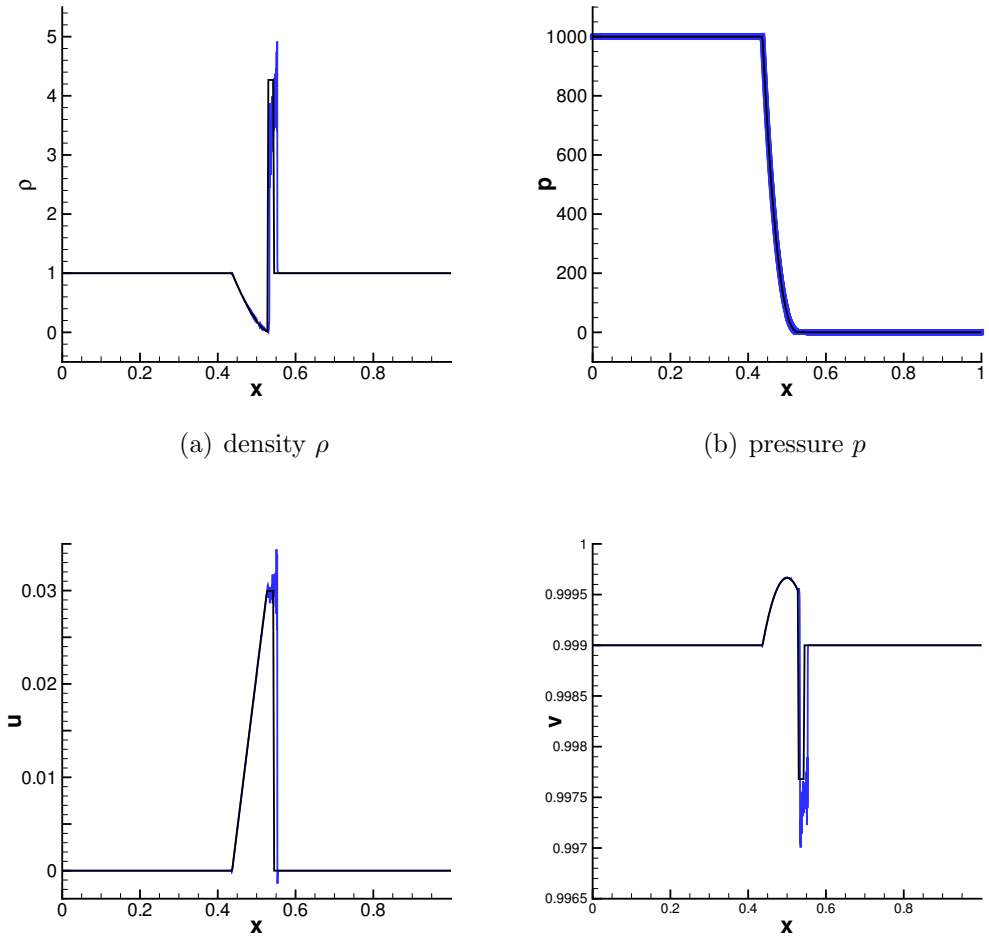
(c) velocity w, 400 cells

(d) velocity w, 6400 cells

Figure 4.8: Example 4.5.1.5: One-dimensional Riemann problem with non-zero transverse velocity at $T = 0.6$, approximated by the third-order RKDG method with the BP limiter. The approximations of w and p on meshes of 400 and 6400 cells are presented respectively. The squares represent the approximate cell averages and the solid lines are the exact solutions.

where $\mathbf{r} = (x, y)$, $\mathbf{v} = (v_x, v_y)$ and $\mathbf{b} = (\cos\alpha, \sin\alpha)$ is the direction vector, along which the wave propagates. In this example we take $\alpha = \pi/4$. In Table 4.2, the numerical errors in L^2 and L^∞ norms of the RKDG method with the BP limiter are listed. We can see that, in this case, the limiter preserves the high-order accuracy.

Example 4.5.2.2 (Oblique shock wave)**.** Next, let us check a two-dimensional oblique 1D shock tube test in the domain $[0, \sqrt{2}/2]^2$. We consider the strong blast



(a) density ρ (b) pressure p

(c) velocity w (d) velocity v

Figure 4.9: Example 4.5.1.5: One-dimensional Riemann problem with non-zero transverse velocity $v = 0.999$ at $T = 0.6$, approximated by the third-order RKDG method with the BP limiter. The approximations of the w, v, p and ρ on 3200 cells are presented. The blue solid lines are the numerical results and the black ones are the exact solutions.

wave in Example 4.5.1.3 propagating along the direction at a 45 degree angle. Initially, the state is divided into two parts by the line $x + y = 1$ and is set to be

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (1.0, 0.0, 0.0, 10^3), & x + y < 1 \\ (1.0, 0.0, 0.0, 0.0), & x + y > 1 \end{cases}. \tag{4.56}$$

The domain is divided into 120×120 cells and the terminal time $T = 0.3$. In Figure

k	h	L^2 error	order	L^∞ error	order
	1/10	4.53 e-2	–	1.98 e-1	–
	1/20	2.61 e-3	4.11	1.64 e-2	3.60
2	1/40	2.97 e-4	3.14	1.65 e-3	3.31
	1/80	3.84 e-5	2.95	2.16 e-4	2.93
	1/160	5.13 e-6	2.91	2.75 e-5	2.97

Table 4.2: Example 4.5.1.1: Two-dimensional accuracy test at $T = 0.2$ for the third-order RKDG method with the BP limiter. k is the degree of polynomial and h is the mesh size.

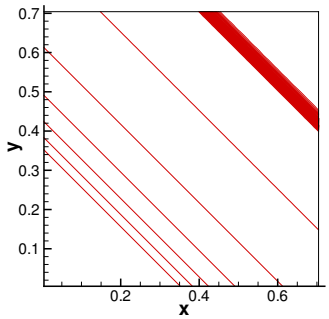
4.10, we present the contours as well as the cuts along the $x = y$ line of the numerical approximation for ρ, p and $\sqrt{w^2 + v^2}$. We see that as for the 1D counterpart, despite of small oscillations, all the structures: the shock, contact continuity and rarefaction, are well resolved and the physical bounds are preserved.

Example 4.5.2.3 (Two-dimensional Riemann problem)**.** The two dimensional Riemann problem involves the interactions of elementary waves, which initially separate four constant states. In [83], the authors extended the 2-D Riemann problem from the classical Newtonian hydrodynamics [62] to the relativistic flows. We take the same initial condition as in [83].

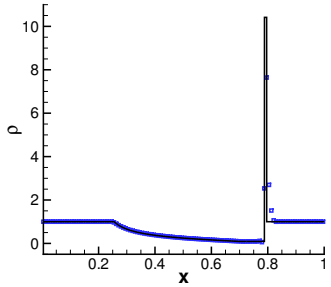
$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (0.1, 0.0, 0.0, 10^{-2}), & x, y > 0 \\ (0.1, 0.99, 0.0, 1.0), & x < 0 < y \\ (0.5, 0.0, 0.0, 1.0), & x, y < 0 \\ (0.1, 0.0, 0.99, 1.0), & y < 0 < x \end{cases}$$

and outflow boundary condition is set everywhere.

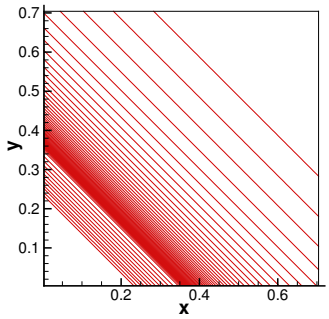
Figure 4.11 shows the numerical approximation at $T = 0.7$ on the 400×400 mesh. Since we only apply the BP limiter without any non-oscillatory technique, there are oscillations around the stationary contact discontinuity and the up-right moving shocks. Despite of this, the structure of the solution matches the results in



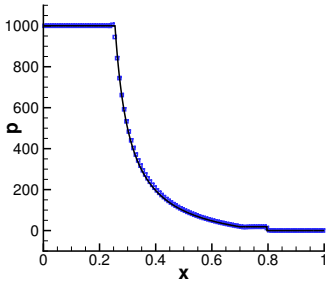
(a) 50 equally spaced contours of ρ from 0 to 8



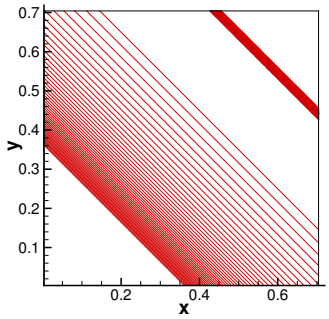
(b) cut along $x = y$



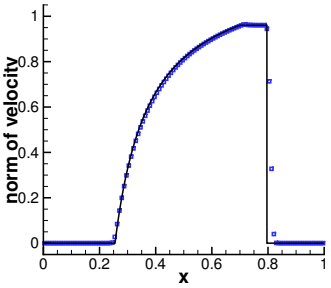
(c) 50 equally spaced contours of p from 0 to 1000



(d) cut along $x = y$



(e) 50 equally spaced contours of $\sqrt{w^2 + v^2}$ from 0 to 1



(f) cut along $x = y$

Figure 4.10: Example 4.5.2.2: Blast wave with initial condition (4.56) propagating in $[0, \frac{\sqrt{2}}{2}]^2$. At time $T = 0.3$ on a 120×120 mesh. In the right column, solid lines are exact solution and the blue squares are the approximate ones.

[84, 83, 42].

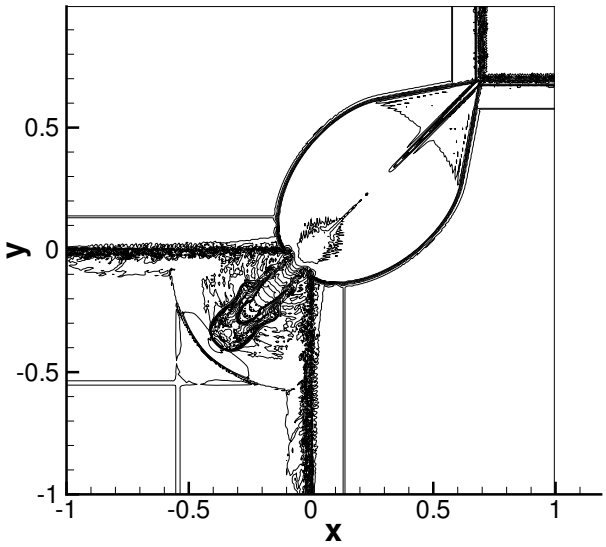
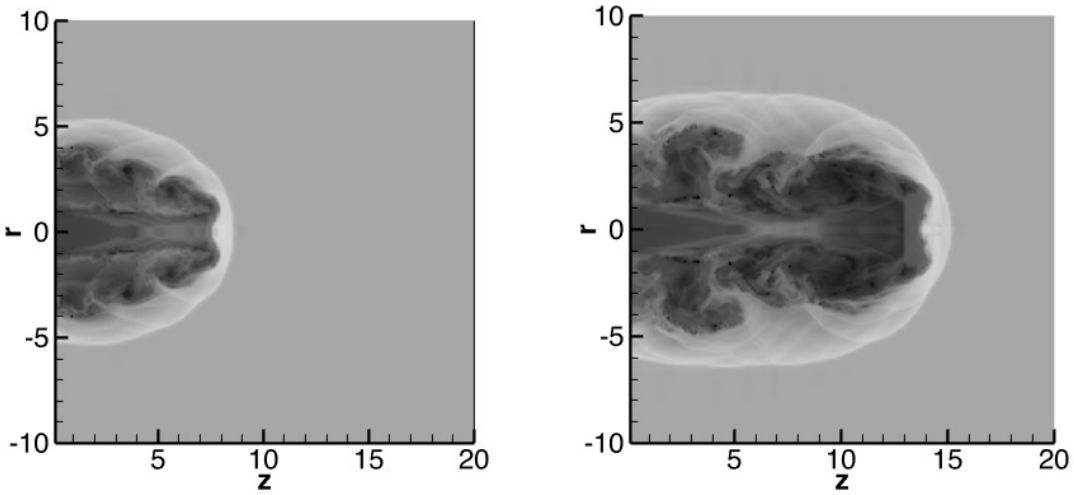


Figure 4.11: Example 4.5.2.3: 400×400 cells at $T = 0.7$ approximated by the third-order RKDG method with the BP limiter. Thirty equally spaced contours of the logarithm of the proper density are plotted.

Example 4.5.2.4 (Axisymmetric relativistic jet)**.** The last example is to apply our bound-preserving DG method to the simulation of the relativistic axisymmetric jet. We consider the equations in the cylindrical coordinates, (4.42) and (4.43). The numerical simulation, morphology and the dynamics of axisymmetric relativistic jets have been well studied in the literature [19, 39]. We first consider the C2 model in [39]. In order to compare the results, we use the same parameters as those in [39]. The initial states are assigned to be

$$(\rho, v_z, v_r, p) = (1.0, 0.0, 0.0, 1.70305 \times 10^{-4}), \quad (r, z) \in [0, 10] \times [0, 20] \tag{4.57}$$

Gas with density $\rho^{in} = 0.01$, pressure $p^{in} = 1.70305 \times 10^{-4}$ and velocity $v_r^{in} = 0.99$ is injected through the inlet part $r < 1$. The classical Mach number of this model is $M_b = 6$ and the corresponding relativistic Mach number is $\mathcal{M}_b = 42$. In Figure 4.12, we show the numerical results at $T = 21$ and $T = 35$ respectively.



(a) T=20 (b) T=35

Figure 4.12: Example 4.57: Relativistic jets of model C2 with initial condition (4.57). The left panel is at $T = 20$ and the right is at $T = 35$. The resolution is 10 points per jet radius.

Next, we consider the C3 model in [39]. The initial states are

$$(\rho, v_z, v_r, p) = (1.0, 0.0, 0.0, 1.7355 \times 10^{-4}), \quad (r, z) \in [0, 15] \times [0, 25] \tag{4.58}$$

The jet material has states $(\rho^{in}, v_z^{in}, v_r^{in}, p^{in}) = (0.01, 0.999, 0.0, 1.7355 \times 10^{-4})$. The relativistic Mach number of this model is around 132. The simulation result at $T = 30$ is presented in Figure 4.13.

The proposed method behaves robustly for both of these highly relativistic jets, especially for the second one. The average speed of the jet head is 0.43 for the C2 model and 0.71 for the C3 model. These match the theoretical estimates in [39], which are 0.42 and 0.70 respectively. Very steady cocoon can be observed in the early stage of the evolution (Figure 4.12, $T = 20$) and it eventually evolves into large

vortices moving backwards. Due to the larger Lorentz factor, the cocoon of the C3 model is less prominent and thinner than that of the C2 one. We believe that, due to the high resolution property of the DG method, the Kelvin-Helmholtz instabilities in the cocoon, which is caused by the discontinuity between the shocked jet material and shocked medium material, are well resolved in both tests. Moreover, there is no "carbuncle" artifact generated. This pathological phenomena was first discussed in [57] for gas dynamics and was discussed and addressed for the RHD problem in [18].

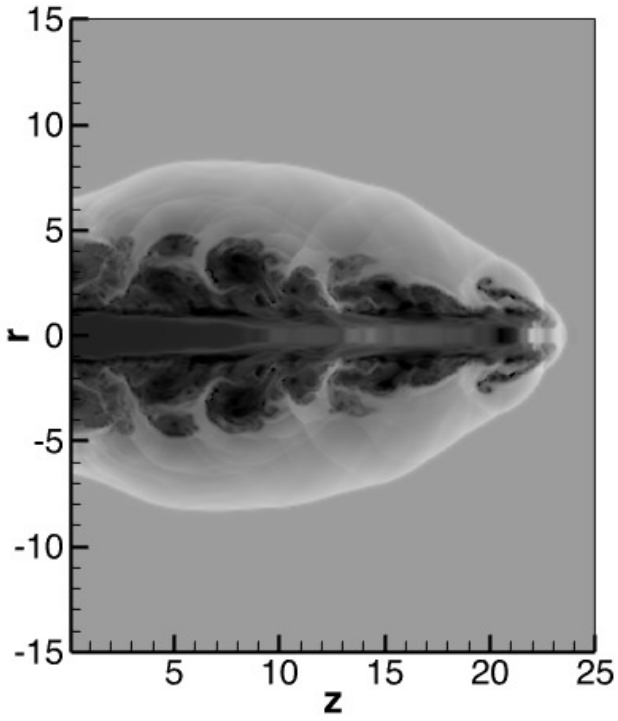


Figure 4.13: Example 4.5.2.4: Relativistic jet of the model C3 with initial condition (4.58) at $T = 30$, approximated by RKDG with the BP limiter. The resolution is 10 points per jet radius.

4.6 Concluding remarks

In this chapter, we have addressed the robustness issues of the discontinuous Galerkin (DG) methods to solve relativistic hydrodynamics (RHD). A special bound-preserving limiter is constructed to obtain physically relevant numerical approximations without compromising the high order accuracy. Moreover, we can prove the L^1-stability of the DG scheme with this limiter. Numerical experiments are given to demonstrate the good performance of the DG scheme. The bound-preserving limiter is very simple and inexpensive, and can be applied to any high order finite volume or DG base schemes.

Chapter Five

Conclusion

This dissertation is based on the work in [55] and [54]. We studied positivity-preserving high order discontinuous Galerkin (DG) methods for solving hyperbolic conservation laws. By extending the general framework for constructing positivity-preserving schemes in [86, 87], we first develop a positivity-preserving DG method with backward Euler time discretization for one-dimensional conservation laws and the method can be easily extended to high-dimensional case with tensor-product meshes and polynomial spaces. The proposed methods work for any meshes including the unstructured ones with extremely varying sizes. It has arbitrarily high order of accuracy for spacial discretizations. The main conclusion is that in order to apply the positivity-preserving limiter, a lower bound for the CFL number is required. This conclusion is theoretically proved for linear scalar problems and numerically verified for nonlinear problems. Applicability of the proposed method is also verified for the compressible Euler system via numerical experiments.

Then we consider the relativistic hydrodynamics (RHD) for ideal gas. For RHD systems, the density and pressure are positive and the velocity is bounded by the speed of light. The violation of these bounds will result in ill-posedness of the problem and blow-up of the code, especially in extreme relativistic cases. It is usually hard to maintain these physical bounds without sacrificing the numerical accuracy. We develop a bound-preserving DG method to solve RHD systems by extending the scheme for non-relativistic hydrodynamics with and without source terms in [87, 89]. Both one and two dimensional models in Cartesian and cylindrical coordinates are considered. The proposed method has the following features. It can theoretically guarantee to preserve the physical bounds for the numerical approximation and maintain its designed high order accuracy. Moreover, it renders L^1-stability to the numerical scheme. All these statements are verified via various extreme numerical test cases including the relativistic jets.

Bibliography

[1] H. L. Atkins and C.-W. Shu, *Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations*, AIAA Journal, 36 (1998), pp. 775–782.

[2] F. Bassi, L. Botti, A. Colombo, A. Ghidoni, and F. Massa, *Linearly implicit Rosenbrock-type Runge–Kutta schemes applied to the discontinuous Galerkin solution of compressible and incompressible unsteady flows*, Computers & Fluids, 118 (2015), pp. 305–320.

[3] P. Batten, M. A. Leschziner, and U. C. Goldberg, *Average-state Jacobians and implicit methods for compressible viscous and turbulent flows*, Journal of Computational Physics, 137 (1997), pp. 38–78.

[4] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Jan. 1994.

[5] P. Borwein and T. Erdélyi, *Polynomials and Polynomial Inequalities*, vol. 161 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1995.

[6] M. H. Carpenter, C. A. Kennedy, H. Bijl, S. A. Viken, and V. N. Vatsa, *Fourth-order Runge–Kutta schemes for fluid mechanics applications*, Journal of Scientific Computing, 25 (2005), pp. 157–194.

[7] J. Centrella and J. R. Wilson, *Planar numerical cosmology. II - The difference equations and numerical tests*, The Astrophysical Journal Supplement Series, 54 (1984), pp. 229–249.

[8] T. Chen and C.-W. Shu, *Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws*, submitted.

[9] Z. Chen, H. Huang, and J. Yan, *Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes*, Journal of Computational Physics, 308 (2016), pp. 198–217.

[10] 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 (1990), pp. 545–581.

[11] 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 (1989), pp. 90–113.

[12] 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 (1989), pp. 411–435.

[13] ——, *The Runge–Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems*, Journal of Computational Physics, 141 (1998), pp. 199–224.

[14] P. Colella and P. R. Woodward, *The Piecewise Parabolic Method (PPM) for gas-dynamical simulations*, Journal of Computational Physics, 54 (1984), pp. 174–201.

[15] P. J. Davis, *Circulant Matrices*, AMS Chelsea Publishing Series, Chelsea, 1994.

[16] V. Dolejší and M. Feistauer, *Discontinuous Galerkin Method - Analysis and Applications to Compressible Flow*, vol. 48 of Springer Series in Computational Mathematics, Springer International Publishing, 2015.

[17] A. Dolezal and S. S. M. Wong, *Relativistic hydrodynamics and essentially non-oscillatory shock capturing schemes*, Journal of Computational Physics, 120 (1995), pp. 266–277.

[18] R. Donat, J. A. Font, J. M. Ibáñez, and A. Marquina, *A flux-split algorithm applied to relativistic flows*, Journal of Computational Physics, 146 (1998), pp. 58–81.

[19] G. C. Duncan and P. A. Hughes, *Simulations of relativistic extragalactic jets*, The Astrophysical Journal, 436 (1994), pp. L119–122.

[20] F. Eulderink, *Numerical Relativistic Hydrodynamics*, phd thesis, Rijksuniversiteit te Leiden, Leiden, Netherlands, 1993.

[21] F. Eulderink and G. Mellema, *Special relativistic jet collimation by inertial confinement*, Astronomy and Astrophysics, 284 (1994), pp. 654–662.

[22] D. Funaro, *Polynomial Approximation of Differential Equations*, vol. 8 of Lecture Notes in Physics, 1992.

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

[24] Y. Ha, C. L. Gardner, A. Gelb, and C.-W. Shu, *Numerical simulation of high Mach number astrophysical jets with radiative cooling*, Journal of Scientific Computing, 24 (2005), pp. 29–44.

[25] P. He and H. Tang, *An adaptive moving mesh method for two-dimensional relativistic hydrodynamics*, Communications in Computational Physics, 11 (2012), pp. 114–146.

[26] X. Y. Hu, N. A. Adams, and C.-W. Shu, *Positivity-preserving method for high-order conservative schemes solving compressible Euler equations*, Journal of Computational Physics, 242 (2013), pp. 169–180.

[27] A. Jameson, *Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings*, in 10th Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, June 1991.

[28] ——, *Application of dual time stepping to fully implicit Runge Kutta schemes for unsteady flow calculations*, in 22nd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum, American Institute of Aeronautics and Astronautics, June 2015.

[29] G. S. Jiang and C.-W. Shu, *On a cell entropy inequality for discontinuous Galerkin methods*, Mathematics of Computation, 62 (1994), pp. 531–538.

[30] V. Korobeinikov, *Problems of Point Blast Theory*, AIP-Press, 1991.

[31] M. Kunik, S. Qamar, and G. Warnecke, *Kinetic schemes for the relativistic gas dynamics*, Numerische Mathematik, 97 (2004), pp. 159–191.

[32] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Lectures in Mathematics, ETH Zürich, Birkhäuser Basel, 1992.

[33] X. Liu and S. Osher, *Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes I*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 760–779.

[34] A. Lucas-Serrano, J. A. Font, J. M. Ibáñez, and J. M. Martí, *Assessment of a high-resolution central scheme for the solution of the relativistic hydrodynamics equations*, Astronomy & Astrophysics, 428 (2004), pp. 703–715.

[35] J. M. Martí, J. M. Ibáñez, and J. A. Miralles, *Numerical relativistic hydrodynamics: Local characteristic approach*, Physical Review D, 43 (1991), pp. 3794–3801.

[36] J. M. Martí and E. Müller, *Analytical solution of the Riemann problem in relativistic hydrodynamics*, Journal of Fluid Mechanics, 258 (1994), pp. 317–333.

[37] J. M. Martiánd E. Müller, *Extension of the piecewise parabolic method to one-dimensional relativistic hydrodynamics*, Journal of Computational Physics, 123 (1996), pp. 1–14.

[38] J. M. Martí and E. Müller, *Numerical hydrodynamics in special relativity*, Living Reviews in Relativity, 6 (2003), p. 7.

[39] J. M. Martí, E. Müller, J. A. Font, J. M. Z. Ibáñez, and A. Marquina, *Morphology and dynamics of relativistic jets*, The Astrophysical Journal, 479 (1997), pp. 151–163.

[40] A. Meister and S. Ortleb, *On unconditionally positive implicit time integration for the DG scheme applied to shallow water flows*, International Journal for Numerical Methods in Fluids, 76 (2014), pp. 69–94.

[41] A. Mignone and G. Bodo, *An HLLC Riemann solver for relativistic flows – II. Magnetohydrodynamics*, Monthly Notices of the Royal Astronomical Society, 368 (2006), pp. 1040–1054.

[42] A. Mignone, T. Plewa, and G. Bodo, *The piecewise parabolic method for multidimensional relativistic fluid dynamics*, The Astrophysical Journal Supplement Series, 160 (2005), p. 199.

[43] Y. Moryossef and Y. Levy, *Unconditionally positive implicit procedure for two-equation turbulence models: Application to k–ω turbulence models*, Journal of Computational Physics, 220 (2006), pp. 88–108.

[44] ——, *Designing a positive second-order implicit time integration procedure for unsteady turbulent flows*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 4196–4206.

[45] ——, *The unconditionally positive-convergent implicit time integration scheme for two-equation turbulence models: Revisited*, Computers & Fluids, 38 (2009), pp. 1984–1994.

[46] A. Nigro, A. Ghidoni, S. Rebay, and F. Bassi, *Modified extended BDF scheme for the discontinuous Galerkin solution of unsteady compressible flows*, International Journal for Numerical Methods in Fluids, 76 (2014), pp. 549–574.

[47] B. O'Shea, G. Bryan, J. Bordner, M. Norman, T. Abel, R. Harkness, and A. Kritsuk, *Introducing Enzo, an AMR cosmology application*, in Springer Lecture Notes in Computational Sciences and Engineering, T. Plewa, T. Linde, and V. Weirs, eds., 2004.

[48] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw Hill, 1980.

[49] W. Pazner and P.-O. Persson, *Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations*, Journal of Computational Physics, 335 (2017), pp. 700–717.

[50] P.-O. Persson, *Scalable parallel Newton-Krylov solvers for discontinuous Galerkin discretizations*, in 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, Jan. 2009.

[51] P.-O. Persson and J. Peraire, *An efficient low memory implicit DG algorithm for time dependent problems*, in 44th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, Jan. 2006.

[52] R. J. Plemmons, *M-matrix characterizations.I—nonsingular M-matrices*, Linear Algebra and its Applications, 18 (1977), pp. 175–188.

[53] S. Qamar and G. Warnecke, *A high-order kinetic flux-splitting method for the relativistic magnetohydrodynamics*, Journal of Computational Physics, 205 (2005), pp. 182–204.

[54] T. Qin and C.-W. Shu, *Implicit high order positivity-preserving discontinuous Galerkin methods for one-dimensional conservation laws*, submitted.

[55] T. Qin, C.-W. Shu, and Y. Yang, *Bound-preserving discontinuous Galerkin methods for relativistic hydrodynamics*, Journal of Computational Physics, 315 (2016), pp. 323–347.

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

[57] J. J. Quirk, *A contribution to the great Riemann solver debate*, International Journal for Numerical Methods in Fluids, 18 (1994), pp. 555–574.

[58] D. Radice and L. Rezzolla, *Discontinuous Galerkin methods for general-relativistic hydrodynamics: Formulation and application to spherically symmetric spacetimes*, Physical Review D, 84 (2011), p. 024010.

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

[60] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems*, Wiley–Interscience, New York, 1967.

[61] V. Schneider, U. Katscher, D. H. Rischke, B. Waldhauser, J. A. Maruhn, and C. D. Munz, *New algorithms for ultra-relativistic numerical hydrodynamics*, Journal of Computational Physics, 105 (1993), pp. 92–107.

[62] C. Schulz-Rinne, J. Collins, and H. Glaz, *Numerical solution of the Riemann problem for two-dimensional gas dynamics*, SIAM Journal on Scientific Computing, 14 (1993), pp. 1394–1414.

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

[64] ——, *Total-variation-diminishing time discretizations*, SIAM Journal on Scientific and Statistical Computing, 9 (1988), pp. 1073–1084.

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

[66] A. H. Taub, *Relativistic rankine-hugoniot equations*, Physical Review, 74 (1948), pp. 328–334.

[67] A. Tchekhovskoy, J. C. McKinney, and R. Narayan, *WHAM: A WENO-based general relativistic numerical scheme–I. Hydrodynamics*, Monthly Notices of the Royal Astronomical Society, 379 (2007), pp. 469–497.

[68] R. Teyssier, *Cosmological hydrodynamics with adaptive mesh refinement - A new high resolution code called RAMSES*, Astronomy & Astrophysics, 385 (2002), pp. 337–364.

[69] E. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer-Verlag Berlin Heidelberg, 2009.

[70] E. F. Toro, M. Spruce, and W. Speares, *Restoration of the contact surface in the HLL-Riemann solver*, Shock Waves, 4 (1994), pp. 25–34.

[71] J. von Neumann and R. D. Richtmyer, *A method for the numerical calculation of hydrodynamic shocks*, Journal of Applied Physics, 21 (1950), pp. 232–237.

[72] 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 (2012), pp. 653–665.

[73] P. Wang, T. Abel, and W. Zhang, *Relativistic hydrodynamic flows using spatial and temporal adaptive structured mesh refinement*, The Astrophysical Journal Supplement Series, 176 (2008), p. 467.

[74] J. R. Wilson, *Numerical study of fluid flow in a Kerr space*, The Astrophysical Journal, 173 (1972), pp. 431–438.

[75] J. R. Wilson, *A numerical method for relativistic hydrodynamics*, 1979, pp. 423–445.

[76] K. Wu and H. Tang, *Finite volume local evolution Galerkin method for two-dimensional relativistic hydrodynamics*, Journal of Computational Physics, 256 (2014), pp. 277–307.

[77] ——, *High-order accurate physical-constraints-preserving finite difference WENO schemes for special relativistic hydrodynamics*, Journal of Computational Physics, 298 (2015), pp. 539–564.

[78] Y. Xing, X. Zhang, and C.-W. Shu, *Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations*, Advances in Water Resources, 33 (2010), pp. 1476–1493.

[79] Y. Yang, D. Wei, and C.-W. Shu, *Discontinuous Galerkin method for Krause's consensus models and pressureless Euler equations*, Journal of Computational Physics, 252 (2013), pp. 109–127.

[80] Z. Yang, P. He, and H. Tang, *A direct Eulerian GRP scheme for relativistic hydrodynamics: One-dimensional case*, Journal of Computational Physics, 230 (2011), pp. 7964–7987.

[81] Z. Yang and H. Tang, *A direct Eulerian GRP scheme for relativistic hydrodynamics: Two-dimensional case*, Journal of Computational Physics, 231 (2012), pp. 2116–2139.

[82] D. Yuan, J. Cheng, and C.-W. Shu, *High order positivity-preserving discontinuous Galerkin methods for radiative transfer equations*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2987–A3019.

[83] L. D. Zanna and N. Bucciantini, *An efficient shock-capturing central-type scheme for multidimensional relativistic flows - I. Hydrodynamics*, Astronomy & Astrophysics, 390 (2002), pp. 1177–1186.

[84] W. Zhang and A. I. MacFadyen, *RAM: A relativistic adaptive mesh refinement hydrodynamics code*, The Astrophysical Journal Supplement Series, 164 (2006), p. 255.

[85] X. Zhang, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations*, Journal of Computational Physics, 328 (2017), pp. 301–343.

[86] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, Journal of Computational Physics, 229 (2010), pp. 3091–3120.

[87] ——, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, Journal of Computational Physics, 229 (2010), pp. 8918–8934.

[88] ——, *Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: Survey and new developments*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 467 (2011), pp. 2752–2776.

[89] ——, *Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms*, Journal of Computational Physics, 230 (2011), pp. 1238–1248.

[90] ——, *Positivity-preserving high order finite difference WENO schemes for compressible Euler equations*, Journal of Computational Physics, 231 (2012), pp. 2245–2258.

[91] X. Zhang, Y. Xia, and C.-W. Shu, *Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes*, Journal of Scientific Computing, 50 (2012), pp. 29–62.

[92] J. Zhao, P. He, and H. Tang, *Steger–Warming flux vector splitting method for special relativistic hydrodynamics*, Mathematical Methods in the Applied Sciences, 37 (2014), pp. 1003–1018.

[93] J. Zhao and H. Tang, *Runge–Kutta discontinuous Galerkin methods with WENO limiter for the special relativistic hydrodynamics*, Journal of Computational Physics, 242 (2013), pp. 138–168.