IMPLICIT POSITIVITY-PRESERVING HIGH ORDER DISCONTINUOUS GALERKIN METHODS FOR CONSERVATION LAWS*

TONG QIN† AND CHI-WANG SHU‡

Abstract. Positivity-preserving discontinuous Galerkin (DG) methods for solving hyperbolic conservation laws have been extensively studied in the last several years. But nearly all the developed schemes are coupled with explicit time discretizations. Explicit discretizations suffer from the constraint for the Courant-Friedrichs-Levis (CFL) number. This makes explicit methods 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 this paper we develop a high-order positivity-preserving DG method with the backward Euler time discretization for conservation laws. We focus on one spatial dimension, however the result easily generalizes to multidimensional tensor product meshes and polynomial spaces. This work is based on a generalization of the positivity-preserving limiters 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) to implicit time discretizations. Both the analysis and numerical experiments indicate that a lower bound for the CFL number is required to obtain the positivity-preserving property. The proposed scheme 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. Numerical experiments are provided to support these conclusions.

Key words. Positivity-preserving; Discontinuous Galerkin method; Backward Euler

AMS subject classifications. 65M60, 65M12

1. Introduction.

In this paper, we consider the conservation law

$$\begin{aligned} u_t + f(u)_x = 0, &\quad (x,t) \in [0,2\pi] \times [0,+\infty), \\ u(x,0) = u_0(x), &\quad x \in [0,2\pi], \end{aligned} \tag{1.1}$$

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.

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 must satisfy the following positivity-preserving property

$$u_0(x) \ge 0 \implies u(x,t) \ge 0, \quad \forall t \ge 0. \tag{1.2}$$

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

*Submitted to the editors on April 3, 2017.

Funding: Research supported by ARO grant W911NF-15-1-0226 and NSF grants DMS-1418750 and DMS-1719410.

†Division of Applied Mathematics, Brown University, Providence, RI 02912, U.S.A. E-mail: tong_qin@brown.edu

‡Division of Applied Mathematics, Brown University, Providence, RI 02912, U.S.A. E-mail: shu@dam.brown.edu.

the compressible Euler system, is always positive. In this paper, 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 (1.2), 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 [14]. In recent years, the positivity-preserving DG schemes have been actively designed and applied for solving hyperbolic conservation laws [41, 42, 37, 38, 31, 7, 40]. All these methods are coupled with explicit temporal discretizations, such as strong stability preserving (SSP) Runge-Kutta (RK) methods [34, 13] and multi-step methods [33]. Explicit temporal discretizations enjoy many advantages, for example, the easiness in handling the nonlinear terms and boundary conditions, high-order accuracy with SSP properties [13], 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 [41], 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 [29], low Mach numbers [3] or long-time simulations for steady-state calculation [15].

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 [15, 16, 6, 29, 28, 27, 25, 2] and see also the book [12]. Although most of the effort has been made for increasing accuracy of the time discretization and for increasing the efficiency of the nonlinear solver, only a few works exist in the literature concerning the positivity-preserving property of implicit methods. For compressible turbulent flow problems, Batten et al. [4] 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 [26]. In [22, 23, 24], 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 [21], 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 [26]. 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 spatial accuracy. In [39], 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 paper, we extend the general framework for constructing positivity-preserving schemes proposed by Zhang and Shu in [41, 42] to implicit temporal discretizations and develop a positivity-preserving DG method with high-order spatial accuracy for one-dimensional conservation laws. The DG methods were first introduced by Reed and Hill [32] for solving neutron transport equations and were further developed by Cockburn et al. in [10, 9, 8, 11] for solving the hyperbolic conservation laws. The DG method enjoys mathematically provable high-order accuracy and stability. The discontinuous feature of its approximation space makes it a good fit for parallel implementation and for handling unstructured meshes. Moreover, for a class of implicit temporal discretizations, it has been shown in [17] via the cell entropy inequality that the fully discrete scheme for the nonlinear conservation law is unconditionally L^2-stable, which works for arbitrary triangulation and any spatial order of accuracy. We adopt the backward Euler temporal discretization in this paper. 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 [41, 42] 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. The proposed positivity-preserving limiter is inexpensive and easy to implement. It not only preserves the positivity and high-order spatial 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 the paper is as follows. In Section 2 we describe the DG scheme. Then in Section 3 the positivity-preserving technique is introduced for scalar equations. In particular, a CFL condition is derived for linear equations to ensure the positiveness of the scheme. The positivity-preserving DG scheme for the compressible Euler system follows in Section 4. Numerical experiments are presented in Section 5 and concluding remarks are given in Section 6.

2. Implicit DG scheme.

2.1. The DG discretization. In this section, we define the DG scheme for (1.1). First, let us 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 $j = 1, 2, \ldots, N$. 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 and $(\cdot,\cdot)_{\hat{I}}$ the one on $\hat{I}$. Then we define the approximation space

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

where $P_k(I_j)$ denotes the polynomial space on I_j with degree up to k.

The semi-discrete DG scheme is to seek the approximation $u_h(t) \in V_h$, such that in each subinterval I_j,

$$\frac{d}{dt}(u_h(t), v)_j - (f(u_h(t)), v_x)_j + \hat{f}_{j+\frac{1}{2}}(u_h(t))v(x^-_{j+\frac{1}{2}}) - \hat{f}_{j-\frac{1}{2}}(u_h(t))v(x^+_{j-\frac{1}{2}}) = 0 \quad (2.1)$$

holds for any $v \in V_h$, where $v(x^+_{j+\frac{1}{2}})$ and $v(x^-_{j+\frac{1}{2}})$ denote the right and the left limits of the function v at $x_{j+\frac{1}{2}}$. The single valued function $\hat{f}_{j+\frac{1}{2}}(u) = \hat{f}(u(x^-_{j+\frac{1}{2}}), u(x^+_{j+\frac{1}{2}}))$ is the numerical flux, which depends on both the left and right limits of u at $x_{j+\frac{1}{2}}$. In this paper, we consider the global Lax-Friedrichs flux

$$\hat{f}(a, b) = \frac{1}{2}[f(a) + f(b) - \alpha(b - a)], \quad (2.2)$$

where $\alpha = \max_{x\in\Omega} |f'(u_0(x))|$.

2.2. Time discretization. With shorthand notation, the semidiscrete scheme (2.1) can be rewritten as below

$$\frac{d}{dt}(u_h(t), v)_j = L_j(u_h(t), v), \quad \forall v \in V_h, \quad \forall j = 1, \dots, N \tag{2.3}$$

where $L_j(u_h(t), v) = (f(u_h(t)), v_x)_j - \left[\hat{f}_{j+\frac{1}{2}}(u_h(t))v(x^-_{j+\frac{1}{2}}) - \hat{f}_{j-\frac{1}{2}}(u_h(t))v(x^+_{j-\frac{1}{2}})\right]$. We use the backward Euler method to further discretize this ODE system. Then the fully discrete scheme is defined by seeking the approximation at time t^{n+1}, which is denoted by $u_h^{n+1} \in V_h$, such that in each cell I_j, we have

$$(u_h^{n+1}, v)_j - \Delta t L_j(u_h^{n+1}, v) = (u_h^n, v)_j \tag{2.4}$$

for all $v \in V_h$. In the following, we use u_j^n to denote $u_h^n|_{I_j}$, $(u^n_{j+\frac{1}{2}})^{\pm}$ to denote $u_j^n(x^{\pm}_{j+\frac{1}{2}})$ and use $\bar{u}_j^n$ to denote the cell average of u_j^n in the interval I_j.

To further solve the nonlinear system (2.4), there have been many works on how to build efficient solvers, such as the work in [29, 28, 27]. But since our main focus is on the positivity preserving property rather than the efficiency of the nonlinear solver, we use the Newton method [12] 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. Positivity-preserving DG scheme for scalar equations. In this section, we introduce how to add the positivity-preserving property to the scheme (2.4). First, let us give the definition of the positivity-preserving DG scheme for the scalar equation as that in [41].

DEFINITION 3.1. *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 $u_h^{n+1}(x) \geq 0$, $\forall x \in \Omega$.*

This definition will be slightly modified later (requiring positivity on specified quadrature points rather than on all points) in order to obtain a more efficient implementation. Generally, the original high-order DG method is not positivity-preserving. We follow the general approach in [41] and construct high-order positivity-preserving DG methods in the following two steps.

Step 1 First, given $u_j^n(x) \geq 0$ for any $x \in I_j$ and any j, find a sufficient condition such that we have the cell average $\bar{u}_j^{n+1}$ positive for all j.

Step 2 Next, we make the whole polynomial $u_j^{n+1}(x) \geq 0$ by invoking the scaling limiter in [20, 41].

The main difficulty lies in the first step. The implicit DG approximation u_j^{n+1} depends on the approximation at the previous time step u_h^n in a global and implicit way. Effort is needed to represent the cell average $\bar{u}_j^{n+1}$ in terms of u_h^n. In this section, we would first show how to overcome this difficulty for scalar linear equations and then we derive a CFL condition, under which, the step 1 is fulfilled. Then we will introduce the scaling limiter and summarize the algorithm.

3.1. Preliminaries. Let us first recall some definitions and results that will be useful in the following analysis. The first useful tool is the so-called M-matrix. For a thorough introduction, one can refer to [5]. 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 [30], the author listed forty equivalent characterizations for M-matrices. For our purpose, we adopt the following one as the definition.

DEFINITION 3.2. *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 [30].

THEOREM 3.3. *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 $D = diag\{d_1, \dots, d_n\}$ 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 3.4. *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.*

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

$$(n+1)p_{n+1}(x) = (2n+1)xp_n(x) - np_{n-1}(x), \quad p_0(x) = 1, \quad p_1(x) = x, \quad x \in \hat{I}. \tag{3.1}$$

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 [35, Sections 3.2, 4.1, 4.3, 4.7, 7.2].

LEMMA 3.5. *Legendre polynomials defined in* (3.1) *have the following properties*

(i) $p_n(1) = 1$, $p_n(-x) = (-1)^n\, p_n(x)$, $\forall x \in \hat{I}$ *and* $|p_n(x)| < 1$, $\forall x \in (-1, 1)$.

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

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

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

(v) Christoffel-Darboux formula

$$\sum_{n=0}^{k} \alpha_n p_n(x)p_n(y) = \frac{\alpha_k(k+1)}{2k+1}\frac{p_{k+1}(x)p_k(y) - p_{k+1}(y)p_k(x)}{x - y} \tag{3.2}$$

where $\alpha_n > 0$.

3.2. CFL condition for linear equations. We consider the linear equation

$$\begin{aligned} u_t + u_x = 0, &\quad x \in \Omega, \\ u(x, 0) = u_0(x), & \end{aligned} \tag{3.3}$$

with periodic boundary condition. Then the scheme (2.4) becomes

$$(u_h^{n+1}, v)_j - \Delta t(u_h^{n+1}, v_x)_j + \Delta t[(u_{j+\frac{1}{2}}^{n+1})^- v_{j+\frac{1}{2}}^- - (u_{j-\frac{1}{2}}^{n+1})^- v_{j-\frac{1}{2}}^+] = (u_h^n, v)_j \tag{3.4}$$

for all $v \in V_h$. Given $u_h^n(x) \ge 0, \forall x \in \Omega$, we want to derive a CFL condition, under which the cell average $\bar{u}_j^{n+1} \ge 0, \forall j$. In order to do that, 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 (3.4), we have

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

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

$$\Lambda^{-1}\bar{\mathbf{u}}^{n+1} + A(\mathbf{u}^{n+1})^- = \Lambda^{-1}\bar{\mathbf{u}}^n \tag{3.5}$$

where $\bar{\mathbf{u}}^n = (\bar{u}_1^n, \dots, \bar{u}_N^n)^T$ and $(\mathbf{u}^{n+1})^- = ((u_{\frac{3}{2}}^{n+1})^-, \dots, (u_{N+\frac{1}{2}}^{n+1})^-)$. The $N \times N$ matrices Λ and A take 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 $(u_{j+\frac{1}{2}}^{n+1})^-$ in terms of $\bar{u}_j^{n+1}$ and u_h^n. To this end, we need to take other special test functions. Recall that the Dirac delta distribution can be approximated by the following series in the distribution sense [19]

$$\delta(x-y) = \frac{1}{2}\sum_{l=0}^{\infty}(2l+1)p_l(x)p_l(y), \quad x, y \in \hat{I}, \tag{3.6}$$

where $p_l(x)$ is the standard Legendre polynomial defined in (3.1). Then we set $y = 1$, truncate the series (3.6) at the $(k+1)$th term and define

$$\hat{\delta}^k(x) = \frac{1}{2}\sum_{l=0}^{k}(2l+1)p_l(x) = \frac{k+1}{2}\frac{p_{k+1}(x) - p_k(x)}{x-1}, \quad x \in \hat{I}. \tag{3.7}$$

We have employed (3.2) with $\alpha_n = (2n+1)/2$ in the last equality.

The following lemma says that this polynomial is an analogue to the Dirac delta distribution in $P^k(\hat{I})$ at the point $y = 1$.

LEMMA 3.6. *The polynomial $\hat{\delta}^k$ has the following properties*

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

(ii) In the cell I_j, define

$$\delta_j^k(x) = \frac{2}{h_j}\hat{\delta}^k\left(T_j(x)\right), \quad x \in I_j, \tag{3.8}$$

then for any $w \in P^k(I_j)$ we have $(w, \delta_j^k)_j = w(x_{j+\frac{1}{2}})$.

(iii) The mass is concentrated at $x = 1$, in the sense that for $k \geq 1$ and $j = 0, \dots, k-1$, 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 3.5, 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 property (ii) can be shown by a simple change of variable.

For property (iii), by (3.7), we have

$$\begin{aligned}(\hat{\delta}^k)^{(j)}(1) - (\hat{\delta}^k)^{(j)}(x) &= \sum_{l=0}^{k} \frac{2l+1}{2} [p_l^{(j)}(1) - p_l^{(j)}(x)] \\ &= \sum_{l=j+1}^{k} \frac{2l+1}{2} [p_l^{(j)}(1) - p_l^{(j)}(x)]. \end{aligned} \tag{3.9}$$

When $x = -1$. By Lemma 3.5 (i), we have

$$(\hat{\delta}^k)^{(j)}(1) - (\hat{\delta}^k)^{(j)}(-1) = \sum_{l=j+1}^{k} \frac{2l+1}{2} [1 - (-1)^{l+j}] p_l^{(j)}(1). \tag{3.10}$$

We claim that $p_l^{(j)}(1) > 0$ for any $l = 0, \ldots, k$ and any $j = 0, \ldots, l$. By Lemma 3.5 (i), this holds for $j = 0$. For $j \geq 1$, it can be checked that the claim holds for $l = 0, 1$. And for $l \geq 2$, we can use Lemma 3.5 (iii) and show the claim by induction. Then we can conclude that the summation (3.10) is positive, since for fixed $j = 0, \ldots, k-1$, there has to be at least one $l = j+1, \ldots, k$ such that $[1 - (-1)^{l+j}] = 2$.

Next, when $x \in (-1, 1)$, by (3.9), it suffices to show

$$p_l^{(j)}(1) - p_l^{(j)}(x) > 0 \tag{3.11}$$

for $j = 0, \ldots, k-1$ and $l = j+1, \ldots, k$ or equivalently, for $l = 1, \ldots, k$ and $j = 0, \ldots, l-1$. First, by Lemma 3.5(i), (3.11) holds for any l with $j = 0$. Therefore, we only need to consider $l = 2, \ldots, k$ and $j = 1, \ldots, l-1$. It is straightforward to verify that (3.11) holds for $l = 2$. If (3.11) holds for $l \leq m-1$ with $m \geq 3$, then we have when $l = m$, for fixed $j = 1, \ldots, l-1$ and any $x \in (-1, 1)$, by Lemma 3.5 (iii)

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

Then by induction, we have proved (3.11). ∎

With the help of the delta approximation (3.8), we have the following representation of $(u_{j+\frac{1}{2}}^{n+1})^-$.

LEMMA 3.7. *For linear scalar conservation law with $f(u) = u$ discretized by the scheme* (2.4)*, we have*

$$\sigma_j^k (u_{j+\frac{1}{2}}^{n+1})^- = \xi_j^k \bar{u}_j^{n+1} + (\hat{u}_j^n, g_j^k)_{\hat{I}} \tag{3.12}$$

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

$$\alpha_i^k = (\hat{\delta}^k)^{(i)}(1), \quad \beta_i^k = (\hat{\delta}^k)^{(i)}(-1). \tag{3.13}$$

Proof. For fixed $l = 0, 1, \dots, k-1$, take the test function to be $(\delta_j^k)^{(l)}(x) - (\delta_j^k)^{(l)}(x_{j-\frac{1}{2}})$ in the scheme (3.4). By the definition of $\delta_j^k(x)$ and Lemma 3.6, 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 $\phi_j^l(x) = p_l(T_j(x))$, $l = 0, \dots, k$, 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

$$D_l - 2\lambda_j D_{l+1} = C_l, \quad l = 0, \dots, k-1 \tag{3.14}$$

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

$$D_{k-1} = 2\lambda_j D_k + C_{k-1} = 2\lambda_j (\hat{u}_j^{n+1}, (\hat{\delta}^k)^{(k)})_{\hat{I}} + C_{k-1} = 4\lambda_j \beta_k^k \bar{u}_j^{n+1} + C_{k-1}. \tag{3.15}$$

By Lemma 3.6 and by using (3.14), we have the following representation of $(u_{j+\frac{1}{2}}^{n+1})^-$

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

If we continue using (3.14) for another $k-2$ times and by using (3.15), we arrive at

$$\begin{aligned} (u_{j+\frac{1}{2}}^{n+1})^- &= (2\lambda_j)^{k-1} D_{k-1} + \sum_{i=0}^{k-2} (2\lambda_j)^i C_i \\ &= (2\lambda_j)^{k-1} [4\lambda_j \beta_k^k \bar{u}_j^{n+1} + C_{k-1}] + \sum_{i=0}^{k-2} (2\lambda_j)^i C_i \end{aligned}$$

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

Then after plugging in the definition of C_i and some manipulations, we obtain

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

□

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 3.11.

Lemma 3.8. *For any $k \geq 0$, the following results hold*
(i) $\alpha_i^k > \beta_i^k$ for $i = 0, \ldots, k-1$ and hence $\sigma_j^k > 0$ for any j.
(ii) $\beta_{k-2i}^k > 0$, $\beta_{k-2i-1}^k < 0$, for $i = 0, \ldots, \lfloor k/2 \rfloor$.
(iii) $\beta_{k-2i}^k + \beta_{k-2i-1}^k > 0$, for $i = 0, \ldots, \lfloor k/2 \rfloor$.

Proof. The statement (i) is a direct conclusion of Lemma 3.6 (iii).

For the statement (ii), let us first derive an explicit formula for β_i^k. By the Rodrigues' formula in Lemma 3.5 (iv) 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 Leibnitz's rule, we obtain for $i \leq l$,

$$p_l^{(i)}(-1) = \frac{1}{2^l l!} \binom{i+l}{l} l! \,[(x-1)^l]^{(i)} \,|_{x=-1} = \frac{1}{2^l} \frac{(i+l)!}{i! l!} \frac{l!}{(l-i)!} (-2)^{l-i}$$
$$= \frac{(-2)^{-i}}{i!} \frac{(i+l)!}{(l-i)!} (-1)^l$$

and hence we have

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

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

$$C_{k-2j} \beta_{k-2j}^k = (|\gamma_{k-2j}^k| - |\gamma_{k-2j}^{k-1}|) + \cdots + (|\gamma_{k-2j}^{k-2j+2}| - |\gamma_{k-2j}^{k-2j+1}|) + |\gamma_{k-2j}^{k-2j}| > 0. \quad (3.16)$$

For β_{k-2j-1}^k, we have

$$C_{k-2j-1} \beta_{k-2j-1}^k = -[(|\gamma_{k-2j-1}^k| - |\gamma_{k-2j-1}^{k-1}|) + \cdots + (|\gamma_{k-2j-1}^{k-2j}| - |\gamma_{k-2j-1}^{k-2j-1}|)] < 0. \quad (3.17)$$

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

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

If we combine (3.16) and (3.17) 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 3.7 and the equation (3.5), we can obtain the following cell average equation

$$T\bar{\mathbf{u}}^{n+1} = \mathcal{L}(\mathbf{u}^n) \tag{3.18}$$

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 $\bar{u}^{n+1}_j$ positive for any $j = 1, \ldots, N$ are the following

Condition I T is an M-matrix, or by Corollary 3.4 and Lemma 3.8,

$$\xi^k_j \geq 0, \quad \forall j = 1, \ldots, N. \tag{3.19}$$

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

$$(\hat{u}^n_j, 1/(2\lambda_j) - g^k_j/\sigma^k_j)_{\hat{I}} + (\hat{u}^n_{j-1}, g^k_{j-1}/\sigma^k_{j-1})_{\hat{I}} \geq 0, \quad j = 1, \ldots, N. \tag{3.20}$$

By Lemma 3.6 (iii), we obtain $g^k_j(x) < \frac{\sigma^k_j}{2\lambda_j}, \forall x \in \hat{I}$, for any k and j and hence the first term in (3.20) is always positive. Since $\hat{u}^n_j$ and $\hat{u}^n_{j-1}$ can be any independent positive polynomials, the condition (3.20) is further reduced to

$$(v, g^k_j)_{\hat{I}} \geq 0, \quad \forall v \in P^k(\hat{I}) \text{ and } v \geq 0. \tag{3.21}$$

In summary, if we set $F_k(\lambda, x) = \sum_{i=0}^k (2\lambda)^i (\hat{\delta}^k)^{(i)}(x)$ and then $\xi_j^k = 2F_k(\lambda_j, -1)$, $g_j^k = F_k(\lambda_j, x) - F_k(\lambda_j, -1)$, sufficient conditions (3.19) and (3.20) actually require that the CFL numbers λ_j satisfy

$$F_k(\lambda_j, -1) \geq 0, \tag{3.22}$$

$$(F_k(\lambda_j, \cdot) - F_k(\lambda_j, -1), v(\cdot))_{\hat{I}} \geq 0, \quad \forall v \in P^k(\hat{I}) \text{ and } v \geq 0. \tag{3.23}$$

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

THEOREM 3.9. *When λ_j is small, conditions (3.22) and (3.23) 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 (3.22) 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 (3.23) 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 3.8, 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 3.8. 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 (3.23) take $v = 1$ and we obtain

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

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 $y = 0$, by (3.7) and Lemma 3.5 (i) we have

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

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

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

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. ∎

Remark 3.10. This theorem shows that a lower bound for the CFL number is necessary for conditions (3.22) and (3.23), which are sufficient conditions for the positivity of the cell averages at the next time step. It does not imply the necessity of the lower bounds to guarantee the cell averages' positivity. The latter necessity will be confirmed by the numerical evidence in Table 3.2 and Table 3.3.

Theorem 3.9 indicates that unlike the situation for the DG method with Euler forward time discretization in [41], 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, an lower bound may be required. The following analysis and numerical experiments confirm this statement.

If we re-examine the condition (3.23) and note that for fixed λ_j, $F_k(\lambda_j, x) \in P^k(\hat{I})$, the inner product in (3.23) 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 (3.23) can be reduced to require

$$J_\alpha^k(\lambda_j) := F_k(\lambda_j, x^\alpha) - F_k(\lambda_j, -1) \geq 0, \quad \alpha = 1, \ldots, N_q. \tag{3.24}$$

If we define $J_0^k(\lambda_j) = F_k(\lambda_j, -1)$, together with condition (3.22), 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 3.11. *When* $\min_j \lambda_j \geq \frac{1}{2}$, *we have*

$$F_k(\lambda_j, -1) > 0, \tag{3.25}$$

$$F_k(\lambda_j, x) - F_k(\lambda_j, -1) > 0, \quad \forall x \in (-1, 1]. \tag{3.26}$$

Proof. First, let us consider (3.25). 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

$$F_k(\lambda_j, -1) = \\ (2\lambda_j)^{k-1}(2\lambda_j \beta_k^k + \beta_{k-1}^k) + \cdots + \begin{cases} (2\lambda_j \beta_2^k + \beta_1^k) + \beta_0^k, & \text{if } k \text{ is even} \\ (2\lambda_j \beta_1^k + \beta_0^k), & \text{if } k \text{ is odd} \end{cases}. \tag{3.27}$$

When $\lambda_j > 1/2$, i.e., $2\lambda_j > 1$, by Lemma 3.8 (ii) and (iii) we can show that each term in (3.27) is strictly positive and hence $F_k(\lambda_j, -1) > 0$.

For the condition (3.26), 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 3.8 (ii), we have $\frac{\partial^k}{\partial x^k} F_k(\lambda_j, x) = \beta_k^k > 0$, and hence

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

We have used the fact $2\lambda_j \geq 1$ and Lemma 3.8 (ii) and (iii) to derive the last inequality. Then we continue the same procedure till $i = 0$ and obtain

$$F_k(\lambda_j, x) > F_k(\lambda_j, -1), \quad \forall x \in (-1, 1].$$

Therefore, in summary, when $\min_j \lambda_j \geq \frac{1}{2}$, we have conditions (3.25) and (3.26) 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 3.9, 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.,

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

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

$$r_k = \max_{r \in \mathcal{R}_k} r, \tag{3.28}$$

we have the following theorem.

THEOREM 3.12. *Let $\{x_j^\alpha\}_{\alpha=1}^{N_q} \in I_j$ be a set of quadrature points which is exact for polynomials of degree $2k$ and let $u_j^n \in P^k(I_j)$ be the backward Euler DG approximation for the linear equation* (3.3) *in the cell I_j at time t^n. Then given $u_j^n(x_j^\alpha) \geq 0$ for $\alpha = 1, \dots, N_q$ and $j = 1, \dots, N$, we have $\bar{u}_j^{n+1} \geq 0$ for any j under the following CFL condition*

$$\min_j \lambda_j \geq r_k \tag{3.29}$$

where $r_k \in (0, \frac{1}{2})$ is defined as in (3.28).

Proof. When $\lambda_j \geq r_k$ by the definition of $\mathcal{R}_k$, we have (3.24) and (3.22) hold. Therefore T in (3.18) is an M-matrix and $\mathcal{L} \geq 0$. By the definition of M-matrix, we can conclude the result. ∎

Remark 3.13. We only require u_h^n to be positive on 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 3.1.

Remark 3.14. Even though the Theorem 3.12 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 3.15. The results also hold for problems with positive source terms and positive inflow boundary conditions.

Remark 3.16. 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-Lobatto (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 on the LGL points $\{x_j^\alpha\}_{\alpha=1}^{N_q}$ in each cell I_j.

Remark 3.17. The lower bounds in Table 3.1 are sharp for odd k and sufficient for even k. If we start from the following initial condition,

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

Table 3.1
Values of r_k for $k = 1, \ldots, 5$ 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

where $M = \arg\max_j h_j$. After one step we record the minimum cell averages for different odd k and λ 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.

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

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 3.3 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 3.3 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	0.262	1.221 E-28
4	0.120	-1.343 E-03	0.177	5.021 E-46

3.3. 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 following scaling limiter [20, 41].

$$\tilde{u}_j^{n+1} = \theta_j [u_j^{n+1} - \bar{u}_j^{n+1}] + \bar{u}_j^{n+1} \tag{3.30}$$

where

$$\theta_j = \begin{cases} \frac{\bar{u}_j^{n+1}}{\bar{u}_j^{n+1} - \min_{x \in I_j} u_j^{n+1}(x)}, & \text{if } \min_{x \in I_j} u_j^{n+1}(x) < 0 \\ 1, & \text{otherwise} \end{cases}. \tag{3.31}$$

This procedure preserves the original high-order accuracy [41].

LEMMA 3.18. *For the modified polynomial* $\tilde{u}_j^{n+1}(x)$*, we have*

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

where u *is the smooth solution.*

Basically, what this lemma says is that the error we commit in the limiting procedure is bounded by the error of the original approximation up to a constant depending on k. The proof can be found in [40].

Remark 3.19. In order to calculate the scaling parameter θ_j in (3.31) 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 3.12, we only require each polynomial u_j^{n+1} to be positive at the LGL quadrature points $\{x_j^\alpha\}_{\alpha=1}^{N_q}$ and hence we can instead use the following scaling parameter

$$\tilde{\theta}_j = \begin{cases} \frac{\bar{u}_j^{n+1}}{\bar{u}_j^{n+1} - \min_\alpha u_j^{n+1}(x_j^\alpha)}, & \text{if } \min_\alpha u_j^{n+1}(x_j^\alpha) < 0 \\ 1, & \text{otherwise} \end{cases} \tag{3.32}$$

in the limiter. This is similar with the scaling limiter in [41] and the same argument can be conducted here to prove that the modified limiter does not kill the original high-order accuracy either.

3.4. 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 a sufficiently large CFL number, a priori or adaptively enlarge it in each time step until $\bar{u}_j^{n+1} \ge 0$, $\forall j$.
3. Apply the scaling limiter (3.30) with θ_j defined in (3.31) or (3.32) to u_j^{n+1} such that $u_j^{n+1}(x_j^\alpha) \ge \epsilon$, for any $\alpha = 1, \dots, N_q$ and j, where ϵ is a small number to help get rid of the round-off effect. In the numerical examples, we take $\epsilon = 10^{-13}$.

4. Positivity-preserving DG scheme for compressible Euler systems.

We consider the following compressible Euler system for ideal gas

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}, \quad t \ge 0, \quad x \in [0,1], \tag{4.1}$$

with

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

and $m = \rho v$, $E = \frac{1}{2}\rho v^2 + \rho e$, $p = (\gamma - 1)\rho e$, where ρ is the density, v 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 $\zeta_1 = v - c$, $\zeta_2 = v$, $\zeta_3 = v + c$, where $c = \sqrt{\gamma p/\rho}$ is the sound speed.

The physical solution lies in the following admissible set

$$G = \left\{ \mathbf{u} = (\rho, m, E)^T : \rho \geq 0, p = (\gamma - 1)\left(E - \frac{1}{2}\frac{m^2}{\rho}\right) \geq 0 \right\}. \tag{4.2}$$

It can be verified that G is a convex set [42].

The DG scheme for (4.1) is to seek an approximation vector $\mathbf{u}_h(t) \in \mathbf{V}_h = [V_h]^3$ such that in each cell I_j we have

$$\frac{d}{dt}(\mathbf{u}_h(t), \mathbf{v})_j = L_j(\mathbf{u}_h(t), \mathbf{v})_j, \quad \forall \mathbf{v} \in \mathbf{V}_h, \tag{4.3}$$

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

The numerical flux $\hat{\mathbf{f}}(\cdot,\cdot)$ is taken to be the global Lax-Friedrichs flux

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

with $\alpha = \|c + |v|\|_\infty$.

As for the scalar case, the ODE system (4.3) is solved by the backward Euler method. If we use $\mathbf{u}_h^n$ to denote the DG approximation at time level n, then the positivity-preserving DG scheme for the compressible Euler system is defined as below.

DEFINITION 4.1. *A DG scheme for the compressible Euler system 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 $x \in \Omega$.*

We design the positivity-preserving DG scheme by extending the positivity-preserving limiter in [42] and the more robust version in [36] for the explicit time stepping to the backward Euler time stepping. We stress on the applicability of the proposed method rather than its theoretical justification. 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.

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

$$\tilde{\tilde{\mathbf{u}}}_j^{n+1} = \theta_2(\tilde{\mathbf{u}}_j^{n+1} - \bar{\tilde{\mathbf{u}}}_j^{n+1}) + \bar{\tilde{\mathbf{u}}}_j^{n+1},$$

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 $\bar{p}_j^{n+1} = p(\bar{\tilde{\mathbf{u}}}_j^{n+1})$.

5. Numerical experiments.

In this section, we present numerical examples. First, we verify the high-order spatial 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 the compressible Euler system will be presented. In all the examples, the domain is first uniformly decomposed with meshsize h and then each node $x_{j+\frac{1}{2}}$ is randomly perturbed in the range $[x_{j+\frac{1}{2}} - \frac{h}{5}, x_{j+\frac{1}{2}} + \frac{h}{5}]$.

5.1. Accuracy tests.

First, let us test the accuracy of the proposed method. We check the spatial accuracy with the steady-state solution to both of the linear equation and the Burgers' equation. 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 5.1 (Steady-state solution to linear problem). For the linear equation, we consider the following problem

$$u_t + u_x = \sin^4(x), \qquad u(x,0) = \sin^2(x), \quad u(0,t) = 0, \tag{5.1}$$

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 5.1 and Table 5.2 we record the errors, numerical orders of accuracy and the minimum value of the numerical approximation, without and with 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.

Example 5.2 (Steady-state solution to Burgers' problems). For the Burgers' equation, we consider the steady state solution to the following problem

$$u_t + \left(\frac{u^2}{2}\right)_x = \sin\left(\frac{x}{4}\right), \qquad u(x,0) = x, \quad u(0,t) = 0, \tag{5.2}$$

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 Tables 5.3 and 5.4, we present numerical results for the cases where the positivity-preserving limiter is off and on respectively. This example shows the effectiveness of the positivity preserving limiter for the nonlinear scalar problem. With the limiter on, the solution stays positive and the high-order accuracy is preserved.

TABLE 5.1
Error table for Example 5.1, approximation of the steady state solution to the linear problem (5.1)*, without the positivity preserving limiter.*

k	N	L^2 error	order	L^∞ error	order	$\min u_h$
1	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
	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
2	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
	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
3	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
	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
4	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
	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

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

$$u_t + \left(\frac{u^2}{2}\right)_x = \sin^3\left(\frac{x}{4}\right), \qquad u(x,0) = \sin^2\left(\frac{x}{4}\right), \quad u(0,t) = 0, \tag{5.3}$$

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 5.5, we present the error, the 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.

With this example, we also show the advantage of implicit methods over explicit ones in steady-state simulations. In Table 5.6, we record the CPU time for the backward Euler and the TVD-RK3 [34] time discretizations. The space is discretized with P^2-DG element with positivity-preserving limiter on. The error tables for both time discretizations are exactly the same, as shown in Table 5.5. However, since the backward Euler method allows large CFL number, $\lambda = 10$ in this example, it is 20 times faster than the TVD-RK3 method to reach the steady state.

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

TABLE 5.2
Error table for Example 5.1, approximation of the steady state solution to the linear problem (5.1) with the positivity preserving limiter.

k	N	L^2 error	order	L^∞ error	order	$\min u_h$
1	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
	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
2	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
	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
3	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
	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
4	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
	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

TABLE 5.3
Error table for Example 5.2, approximation of the steady state solution to the Burgers' equation (5.2), 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

Example 5.3 (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 5.1, we present the numerical results at $T = 0.2$, with and without the

TABLE 5.4

Error table for Example 5.2, approximation of the steady state solution to the Burgers' equation (5.2)*, 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

TABLE 5.5

Error table for Example 5.2, approximation of the steady-state solution to the Burgers' equation (5.3)*, 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

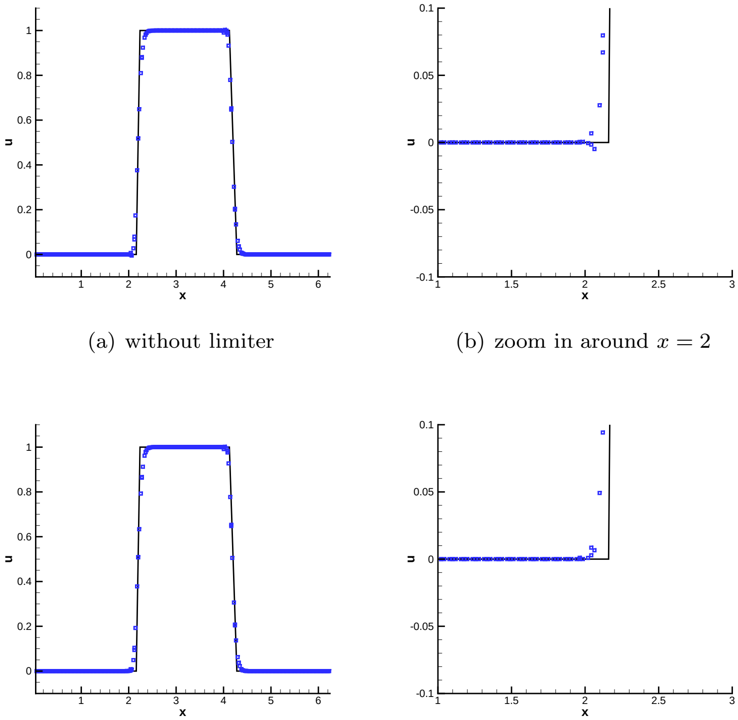
TABLE 5.6

CPU time for Example 5.2, problem (5.3) *solved by* P^2*-DG with backward Euler and TVD-RK3 temporal discretizations. The positivity-preserving limiter is on.*

N	20	40	80	160
Backward Euler	0.31	1.57	9.73	58.12
TVD-RK3	5.16	32.16	199.75	1209.09

positivity-preserving limiter respectively. For the linear equation, we take $\min_j \lambda_j = r_2 = 0.262$ as listed in Table 3.1. As the zoom-in plots show, the limiter helps the solution to stay positive. Without the limiter a negative undershoot appears.

Example 5.4 (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 5.2, 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.



(a) without limiter (b) zoom in around $x = 2$

(c) with limiter (d) zoom in around $x = 2$

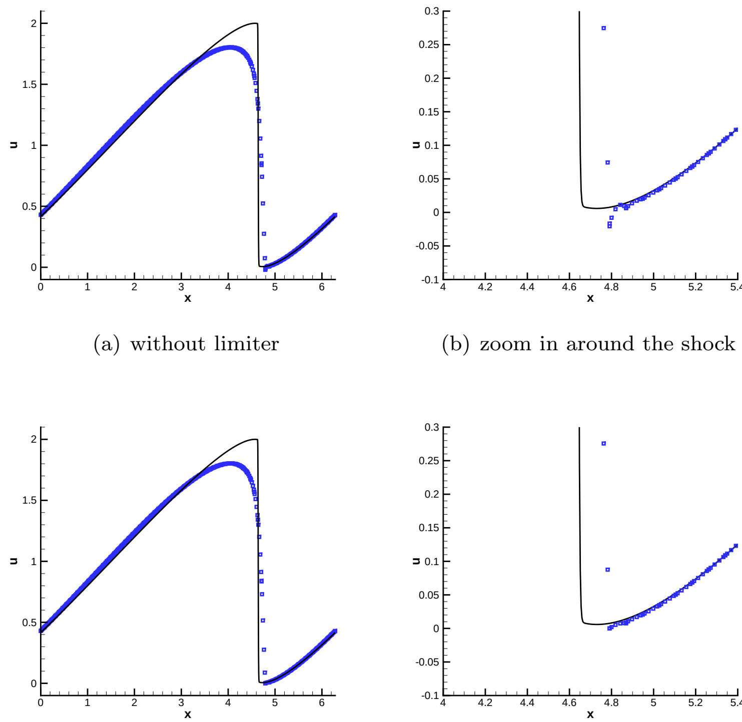
FIG. 5.1. *Example 5.3: at $T = 0.2$, with $\Delta t = 0.266 \max_j h_j$ and $N = 120$. Solid line is the exact solution. Blue squares are point values at the Legendre-Gauss-Lobatto points.*

Example 5.5 (Buckley-Leverett problem). In this example, the Buckley-Leverett problem in $[-1, 1]$ 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 $u_0(x) = I_{[-0.5,1]}$, 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 phenomenon is also observed for the Burgers' 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 an effective positivity-preserving tool when applied to problems with sonic points. In this example, we change the initial condition to

$$u_0(x) = \begin{cases} 0.9, & \text{if } x \in [-0.5, 1] \\ 10^{-3}, & \text{otherwise} \end{cases}.$$

In Figure 5.3, 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.

5.3. Compressible Euler System. At last, we turn to the examination of the applicability of the proposed scheme to the compressible Euler system (4.1). The



(a) without limiter (b) zoom in around the shock

(c) with limiter (d) zoom in around the shock

FIG. 5.2. *Example 5.4 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 $= 2$.*

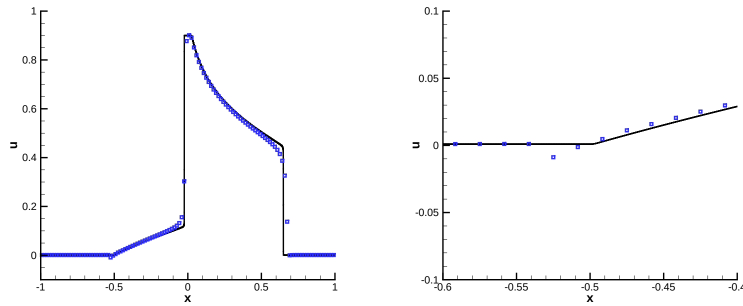
computational domain $\Omega = [0, 1]$ is decomposed into a nonuniform mesh. The generic ratio of specific heats is taken to be $\gamma = 1.4$. For all the examples, quadratic element is employed with the positivity-preserving limiter on. And the time stepping size is set to be $\Delta t = \frac{\text{CFL}}{\|c+|v|\|_\infty} \min_j h_j$, where c is the sound speed and v is the velocity. We take CFL $= 2$ for all the examples.

Example 5.6 (Shock tube). First, let us consider the following shock tube problem

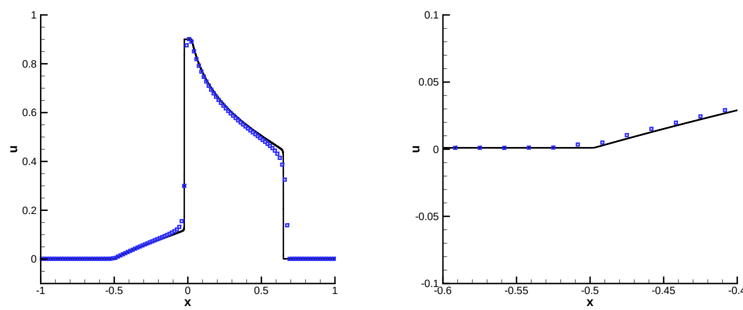
$$\begin{cases} \rho = 1, & \text{if } x < 0.5 \\ v = 0, & \text{if } x < 0.5 \\ p = 1000, & \text{if } x < 0.5 \end{cases} , \qquad \begin{cases} \rho = 1, & \text{if } x \geq 0.5 \\ v = 0, & \text{if } x \geq 0.5 \\ p = 0.01, & \text{if } x \geq 0.5 \end{cases} .$$

The numerical approximation is presented in Figure 5.4. The solutions consists of a strong shock wave, a contact discontinuity and a rarefaction wave. Due to the first-order temporal discretization and the large CFL number, the contact discontinuity and the shock wave in the density are not very well captured. However, with the positivity-preserving limiter on, both the pressure and the density stay positive all the time during the simulation.

Example 5.7 (Double rarefaction). The double rarefaction problem starts from the

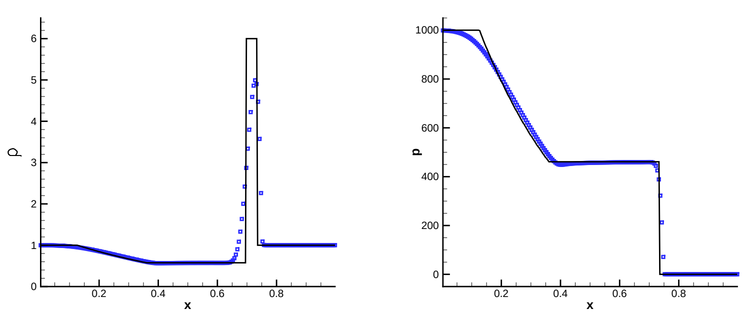


(a) without limiter (b) zoom in around $x = -0.5$



(c) with limiter (d) zoom in around $x = -0.5$

FIG. 5.3. *Example 5.5 at $T = 0.4$. 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 $CFL = 3.5$.*



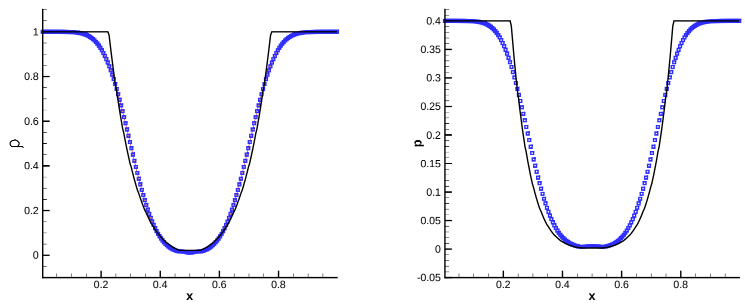
(a) density (b) pressure

FIG. 5.4. *Example 5.6 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.*

following initial condition

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

This problem has a solution consisting of two symmetric rarefaction waves and a 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 5.5, 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.

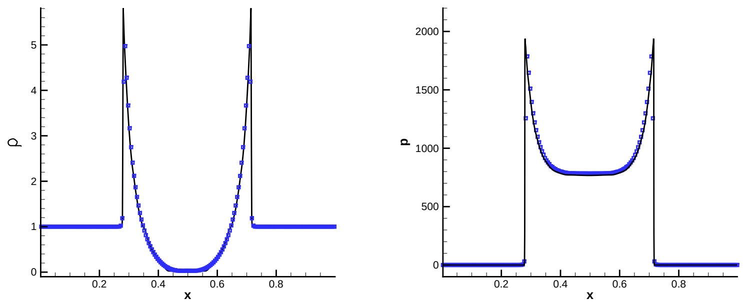


(a) density (b) Pressure

FIG. 5.5. *Example 5.7 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 5.8 (Blast wave). The last example is the Sedov point-blast wave [18]. 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 5.6 we present the numerical approximation and the exact solution [18] 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.

6. Concluding remarks. In this paper, we develop an implicit positivity-preserving DG method with high-order spatial accuracy for one-dimensional conservation laws. This work is an extension of the positivity-preserving limiter in [41, 42] 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 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 result in this paper easily generalizes 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



(a) density (b) Pressure

Fig. 5.6. *Example 5.8 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.*

methods with order greater than one [13], we need to turn to other types of implicit time discretizations, such as BDF methods and fully implicit RK methods.

REFERENCES

[1] H. L. Atkins and C.-W. Shu, *Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations*, AIAA J., 36 (1998), pp. 775–782, `https://doi.org/10.2514/2.436`.

[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*, Comput. Fluids, 118 (2015), pp. 305–320, `https://doi.org/10.1016/j.compfluid.2015.06.007`.

[3] F. Bassi, C. De Bartolo, R. Hartmann, and A. Nigro, *A discontinuous Galerkin method for inviscid low Mach number flows*, J. Comput. Phys., 228 (2009), pp. 3996–4011, `https://doi.org/10.1016/j.jcp.2009.02.021`.

[4] P. Batten, M. A. Leschziner, and U. C. Goldberg, *Average-state Jacobians and implicit methods for compressible viscous and turbulent flows*, J. Comput. Phys., 137 (1997), pp. 38–78, `https://doi.org/10.1006/jcph.1997.5793`.

[5] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Jan. 1994, `https://doi.org/10.1137/1.9781611971262`.

[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*, J. Sci. Comput., 25 (2005), pp. 157–194, `https://doi.org/10.1007/s10915-004-4637-3`.

[7] 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*, J. Comput. Phys., 308 (2016), pp. 198–217, `https://doi.org/10.1016/j.jcp.2015.12.039`.

[8] 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*, Math. Comp., 54 (1990), pp. 545–581, `https://doi.org/10.1090/S0025-5718-1990-1010597-0`.

[9] 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*, J. Comput. Phys., 84 (1989), pp. 90–113, `https://doi.org/10.1016/0021-9991(89)90183-6`.

[10] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework*, Math. Comp., 52 (1989), pp. 411–435, `https://doi.org/10.1090/S0025-5718-1989-0983311-4`.

[11] B. Cockburn and C.-W. Shu, *The Runge–Kutta discontinuous Galerkin method for conser-*

vation laws V: Multidimensional systems, J. Comput. Phys., 141 (1998), pp. 199–224, https://doi.org/10.1006/jcph.1998.5892.

[12] 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.

[13] S. Gottlieb, C. Shu, and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112, https://doi.org/10.1137/S003614450036757X.

[14] Y. Ha, C. L. Gardner, A. Gelb, and C.-W. Shu, *Numerical simulation of high Mach number astrophysical jets with radiative cooling*, J. Sci. Comput., 24 (2005), pp. 29–44, https://doi.org/10.1007/s10915-004-4786-4.

[15] 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, https://doi.org/10.2514/6.1991-1596.

[16] A. Jameson, *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, https://doi.org/10.2514/6.2015-2753.

[17] G. S. Jiang and C.-W. Shu, *On a cell entropy inequality for discontinuous Galerkin methods*, Math. Comp., 62 (1994), pp. 531–538, https://doi.org/10.1090/S0025-5718-1994-1223232-7.

[18] V. Korobeinikov, *Problems of Point Blast Theory*, AIP-Press, 1991.

[19] Y.-T. Li and R. Wong, *Integral and series representations of the Dirac delta function*, Comm. Pure Appl. Math, 7 (2008), pp. 229–247, https://doi.org/10.3934/cpaa.2008.7.229.

[20] X. Liu and S. Osher, *Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes I*, SIAM J. Numer. Anal., 33 (1996), pp. 760–779, https://doi.org/10.1137/0733038.

[21] A. Meister and S. Ortleb, *On unconditionally positive implicit time integration for the DG scheme applied to shallow water flows*, Int. J. Numer. Meth. Fluids, 76 (2014), pp. 69–94, https://doi.org/10.1002/fld.3921.

[22] Y. Moryossef and Y. Levy, *Unconditionally positive implicit procedure for two-equation turbulence models: Application to k–ω turbulence models*, J. Comput. Phys., 220 (2006), pp. 88–108, https://doi.org/10.1016/j.jcp.2006.05.001.

[23] Y. Moryossef and Y. Levy, *Designing a positive second-order implicit time integration procedure for unsteady turbulent flows*, Comput.Methods in Appl.Mech.Eng., 196 (2007), pp. 4196–4206, https://doi.org/10.1016/j.cma.2007.04.001.

[24] Y. Moryossef and Y. Levy, *The unconditionally positive-convergent implicit time integration scheme for two-equation turbulence models: Revisited*, Comput. Fluids, 38 (2009), pp. 1984–1994, https://doi.org/10.1016/j.compfluid.2009.06.005.

[25] A. Nigro, A. Ghidoni, S. Rebay, and F. Bassi, *Modified extended BDF scheme for the discontinuous Galerkin solution of unsteady compressible flows*, Int. J. Numer. Meth. Fluids, 76 (2014), pp. 549–574, https://doi.org/10.1002/fld.3944.

[26] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw Hill, 1980.

[27] W. Pazner and P.-O. Persson, *Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations*, J. Comput. Phys., 335 (2017), pp. 700–717, https://doi.org/10.1016/j.jcp.2017.01.050.

[28] 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, https://doi.org/10.2514/6.2009-606.

[29] 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, https://doi.org/10.2514/6.2006-113.

[30] R. J. Plemmons, *M-matrix characterizations.I—nonsingular M-matrices*, Linear Algebra and its Applications, 18 (1977), pp. 175–188, https://doi.org/10.1016/0024-3795(77)90073-8.

[31] T. Qin, C.-W. Shu, and Y. Yang, *Bound-preserving discontinuous Galerkin methods for relativistic hydrodynamics*, J. Comput. Phys., 315 (2016), pp. 323–347, https://doi.org/10.1016/j.jcp.2016.02.079.

[32] 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.

[33] C.-W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. and Stat. Comput., 9 (1988), pp. 1073–1084, https://doi.org/10.1137/0909073.

[34] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471, https://doi.org/10.1016/0021-9991(88)90177-5.

[35] G. Szegö, *Orthogonal polynomials*, American Mathematical Society, New York, 1939.

[36] C. Wang, X. Zhang, C.-W. Shu, and J. Ning, *Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations*, J. Comput. Phys., 231 (2012), pp. 653–665, https://doi.org/10.1016/j.jcp.2011.10.002.

[37] 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, https://doi.org/10.1016/j.advwatres.2010.08.005.

[38] Y. Yang, D. Wei, and C.-W. Shu, *Discontinuous Galerkin method for Krause's consensus models and pressureless Euler equations*, J. Comput. Phys., 252 (2013), pp. 109–127, https://doi.org/10.1016/j.jcp.2013.06.015.

[39] D. Yuan, J. Cheng, and C.-W. Shu, *High order positivity-preserving discontinuous Galerkin methods for radiative transfer equations*, SIAM J. Sci. Comput., 38 (2016), pp. A2987–A3019, https://doi.org/10.1137/16M1061072.

[40] X. Zhang, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations*, J. Comput. Phys., 328 (2017), pp. 301–343, https://doi.org/10.1016/j.jcp.2016.10.002.

[41] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, J. Comput. Phys., 229 (2010), pp. 3091–3120, https://doi.org/10.1016/j.jcp.2009.12.030.

[42] X. Zhang and C.-W. Shu, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, J. Comput. Phys., 229 (2010), pp. 8918–8934, https://doi.org/10.1016/j.jcp.2010.08.016.