



On maximum-principle-satisfying high order schemes for scalar conservation laws [☆]

Xiangxiong Zhang ^a, Chi-Wang Shu ^{b,*}

^a Department of Mathematics, Brown University, Providence, RI 02912, United States

^b Division of Applied Mathematics, Brown University, Providence, RI 02912, United States

ARTICLE INFO

Article history:

Received 26 October 2009

Accepted 23 December 2009

Available online 4 January 2010

MSC:

65M06

65M60

65M12

Keywords:

Hyperbolic conservation laws

Finite volume scheme

Discontinuous Galerkin method

Essentially non-oscillatory scheme

Weighted essentially non-oscillatory scheme

Maximum principle

High order accuracy

Strong stability preserving time discretization

Passive convection equation

Incompressible flow

ABSTRACT

We construct uniformly high order accurate schemes satisfying a strict maximum principle for scalar conservation laws. A general framework (for arbitrary order of accuracy) is established to construct a limiter for finite volume schemes (e.g. essentially non-oscillatory (ENO) or weighted ENO (WENO) schemes) or discontinuous Galerkin (DG) method with first order Euler forward time discretization solving one-dimensional scalar conservation laws. Strong stability preserving (SSP) high order time discretizations will keep the maximum principle. It is straightforward to extend the method to two and higher dimensions on rectangular meshes. We also show that the same limiter can preserve the maximum principle for DG or finite volume schemes solving two-dimensional incompressible Euler equations in the vorticity stream-function formulation, or any passive convection equation with an incompressible velocity field. Numerical tests for both the WENO finite volume scheme and the DG method are reported.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

We consider numerical solutions of the scalar conservation law

$$u_t + \nabla \cdot \mathbf{F}(u) = 0, \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad (1.1)$$

where $u_0(\mathbf{x})$ is assumed to be a bounded variation function. The one-dimensional version

$$u_t + f(u)_x = 0, \quad u(x, 0) = u_0(x), \quad (1.2)$$

is used often in this paper to illustrate the main ideas. The main difficulty in solving (1.1) is that the solution may contain discontinuities even if the initial condition is smooth, hence we must consider the physically relevant unique weak solution

[☆] Research supported by AFOSR Grant FA9550-09-1-0126 and NSF Grant DMS-0809086.

* Corresponding author. Tel.: +1 401 863 2549; fax: +1 401 863 1355.

E-mail addresses: zhangxx@dam.brown.edu (X. Zhang), shu@dam.brown.edu (C.-W. Shu).

which is called the entropy solution. An important property of the entropy solution is that it satisfies a strict maximum principle (e.g. [5]), i.e., if

$$M = \max_{\mathbf{x}} u_0(\mathbf{x}), \quad m = \min_{\mathbf{x}} u_0(\mathbf{x}), \quad (1.3)$$

then $u(\mathbf{x}, t) \in [m, M]$ for any \mathbf{x} and t .

Successful high order numerical schemes for solving (1.1) includes, among others, the Runge–Kutta discontinuous Galerkin (RKDG) method with a total variation bounded (TVB) limiter [3], the essentially non-oscillatory (ENO) finite volume and finite difference schemes [9,25], and the weighted ENO (WENO) finite volume and finite difference schemes [17,12]. Although these schemes are nonlinearly stable in numerical experiments and some of them can be proved to be total variation stable, they do not in general satisfy a strict maximum principle. It is very difficult to obtain a uniformly high order accurate scheme satisfying a strict maximum principle in the sense that the numerical solution never goes out of the range $[m, M]$, which is a desired property in some applications, for example, when u is a volume ratio which should not go outside the range of $[0, 1]$.

The total variation diminishing (TVD) schemes [8] satisfy the strict maximum principle, but it is well known that finite difference or finite volume TVD schemes solving (1.2), where the total variation is measured by that of the grid values, necessarily degenerate to first order accuracy at smooth extrema [19], thus TVD schemes are at most second order accurate in the L^1 norm for general smooth and non-monotone solutions. By measuring the total variation of the reconstruction polynomials, Sanders introduced a third order TVD scheme [21] solving (1.2). In [26], we have extended Sanders's scheme to higher order accuracy (up to sixth order), obtaining uniformly high order TVD schemes with the total variation measured by that of the reconstruction polynomials. The schemes in [21,26] satisfy strict maximum principle, however it appears difficult to generalize these schemes to multi-dimensional problems, because the time evolution is exact and is implemented by using the characteristic method, which is realistic only in one space dimension. To get a genuinely high order accurate scheme satisfying the maximum principle, we should measure the maximum of the reconstruction polynomials. In [16], by controlling the number of extrema and the maximum/minimum of the reconstruction polynomials, Liu and Osher developed a third order accurate non-oscillatory scheme in one space dimension, for which they proved a strict local maximum principle and non-oscillatory properties for one-dimensional linear equations. The time evolution in [16] is equivalent to the exact time evolution only for the linear equation. That is why maximum principle of the scheme in [16] can be proved only for the linear equation. In [13], Jiang and Tadmor proved the maximum principle of the second order two multi-dimensional central schemes. However, it appears difficult to generalize the provable maximum principle to uniformly higher order schemes in this class.

In this paper, we develop a genuinely high order accurate maximum-principle-satisfying scheme for one-dimensional and multi-dimensional scalar conservation laws, in the sense that the numerical solution never goes out of the range $[m, M]$ of the initial condition. Our scheme uses the simple Euler forward and the standard strong stability preserving (SSP) Runge–Kutta or multi-step time discretizations [25,23,7], allowing for easy and practical implementation and easy generalization from one to multi-dimensions. The limiter introduced in [16] is used to control the maximum/minimum of the reconstruction polynomials. Simply speaking, this control is achieved by a linear scaling around the cell average. In particular, we demonstrate this procedure by considering a quadratic polynomial $p(x)$ approximating a function $u(x)$ with third order accuracy on a small interval I . Let \bar{p} denote the cell average of $p(x)$ and $M' = \max_{x \in I} p(x)$, $m' = \min_{x \in I} p(x)$, $M_0 = \max_{x \in I} u(x)$, $m_0 = \min_{x \in I} u(x)$, define $\tilde{p}(x)$ by

$$\tilde{p}(x) = \theta(p(x) - \bar{p}) + \bar{p}, \quad \theta = \min \left\{ \frac{M_0 - \bar{p}}{M' - \bar{p}}, \frac{m_0 - \bar{p}}{m' - \bar{p}}, 1 \right\}. \quad (1.4)$$

If $\bar{p} \in [m, M]$, then $\tilde{p}(x)$ is still a third order accurate approximation with the same cell average, and $\tilde{p}(x) \in [m, M]$, for all $x \in I$. This fact is proved in [16]. The major difficulty to use this idea to construct a maximum-principle-satisfying scheme is to maintain the property $\bar{u} \in [m, M]$ during the time evolution, without destroying accuracy. The exact time evolution procedure is such a method, which is used in [21,16,26], however this procedure is very difficult to implement for multi-dimensional nonlinear problems. In this paper, we follow the idea in [20] to show that, under a suitable CFL condition, for a finite volume or a DG scheme, the simple Euler forward or the strong stability preserving (SSP) time discretization [25,23,7] will keep the property $\bar{u} \in [m, M]$ and the validity of the maximum principle if we use the linear scaling (1.4) or a simplified version for the reconstruction polynomials or the DG polynomials, thus maintaining uniform high order accuracy. The simplified limiter is still a linear scaling (1.4) but it replaces the definition of M' and m' by $M' = \max_{x \in S} p(x)$ and $m' = \min_{x \in S} p(x)$ where S is a finite set containing the Legendre Gauss–Lobatto quadrature points on I . Since we avoid evaluating the extrema of polynomials in the simplified limiter, we can easily implement it for polynomials of any degree.

The main conclusion of this paper is as follows: by applying the limiter (1.4) or the simplified version which avoids the evaluation of extrema of polynomials, to a high order accurate finite volume scheme or a discontinuous Galerkin scheme solving one or multi-dimensional scalar conservation laws, with the time evolution by a SSP Runge–Kutta or multi-step method, we obtain a uniformly high order accurate scheme solving (1.1) with the strict maximum principle in the sense that the numerical solution never goes out of the range $[m, M]$, where the m and M are defined in (1.3). The algorithm and conclusion are also valid for two-dimensional incompressible Euler equations in the vorticity stream-function formulation, or for any passive convection equation with an incompressible velocity field.

The paper is organized as follows: we first describe and prove the maximum principle for an arbitrarily high order scheme in one space dimension in Section 2. In Section 3, we provide a straightforward extension to two space dimensions on rectangular meshes. Section 4 is the application of the scheme to two-dimensional incompressible Euler equations in the vorticity stream-function formulation, and to any passive convection equation with an incompressible velocity field. In Sections 5 and 6, numerical tests for the WENO finite volume schemes and for the DG method, respectively, will be shown, including examples from traffic flow problems and two-dimensional incompressible Euler equations. Concluding remarks are given in Section 7.

2. High order schemes satisfying the maximum principle in one dimension

2.1. First order Euler forward in time

In this section, we consider a finite volume method, for example the WENO finite volume method in [17], and the DG method in [3] for solving the one-dimensional scalar conservation laws (1.2). We consider only the first order Euler forward time discretization in this subsection; higher order time discretization will be discussed in the next subsection.

A finite volume scheme or a scheme satisfied by the cell averages of a DG method can be written as

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \lambda [h(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+) - h(u_{j-\frac{1}{2}}^-, u_{j-\frac{1}{2}}^+)], \tag{2.1}$$

where n refers to the time step and j to the spatial cell (we assume uniform mesh size only for simplicity), and $\lambda = \frac{\Delta t}{\Delta x}$ is the ratio of time and space mesh sizes. \bar{u}_j^n is the approximation to the cell averages of $u(x, t)$ in the cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ at time level n , and $u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+$ are the high order approximations of the nodal values $u(x_{j+\frac{1}{2}}, t^n)$ within the cells I_j and I_{j+1} , respectively. These values are either reconstructed from the cell averages \bar{u}_j^n in a finite volume method or read directly from the evolved polynomials in a DG method. We assume that there is a polynomial $p_j(x)$ (either reconstructed in a finite volume method or evolved in a DG method) with degree k , where $k \geq 2$, defined on I_j such that \bar{u}_j^n is the cell average of $p_j(x)$ on I_j , $u_{j-\frac{1}{2}}^+ = p_j(x_{j-\frac{1}{2}})$ and $u_{j+\frac{1}{2}}^- = p_j(x_{j+\frac{1}{2}})$.

To define the scheme (2.1), we need to specify the numerical flux function $h(\cdot, \cdot)$, which is assumed to be an exact or approximate Riemann solver. In particular, $h(\cdot, \cdot)$ is a Lipschitz continuous function of both arguments, and is consistent with the physical flux $f(u)$ in (1.2): $h(u, u) = f(u)$. For stability reasons, we need more restrictions on the flux function. We will use a Lipschitz continuous monotone flux as defined in [4], i.e., $h(\cdot, \cdot)$ is nondecreasing in its first argument and nonincreasing in its second argument. For instance, the global Lax–Friedrichs flux defined by

$$h(u, v) = \frac{1}{2} [f(u) + f(v) - a(v - u)], \quad a = \max |f'(u)|, \tag{2.2}$$

where the maximum is taken over the whole region where u and v vary, is a monotone flux. For a monotone flux $h(\cdot, \cdot)$, it is well known that a first order monotone scheme satisfies the strict maximum principle. We recall this result for the Lax–Friedrichs flux here.

Lemma 2.1. *Under the CFL condition $\lambda a \leq 1$, $a = \max |f'(u)|$, consider the first order Lax–Friedrichs scheme*

$$u_j^{n+1} = u_j^n - \lambda [h(u_j^n, u_{j+1}^n) - h(u_{j-1}^n, u_j^n)]. \tag{2.3}$$

If $u_j^n \in [m, M]$, $\forall j$, then $u_j^{n+1} \in [m, M]$.

Proof. The right hand side of the scheme can be written as

$$\begin{aligned} H(u_{j-1}^n, u_j^n, u_{j+1}^n) &= u_j^n - \lambda [h(u_j^n, u_{j+1}^n) - h(u_{j-1}^n, u_j^n)] \\ &= u_j^n - \lambda \left[\frac{1}{2} (f(u_j^n) + f(u_{j+1}^n) - a(u_{j+1}^n - u_j^n)) - \frac{1}{2} (f(u_{j-1}^n) + f(u_j^n) - a(u_j^n - u_{j-1}^n)) \right] \\ &= (1 - \lambda a) u_j^n + \frac{\lambda}{2} (a u_{j+1}^n - f(u_{j+1}^n)) + \frac{\lambda}{2} (a u_{j-1}^n + f(u_{j-1}^n)). \end{aligned}$$

We clearly have $\frac{\partial}{\partial b} H(b, c, d) = \frac{\lambda}{2} (a + f'(b)) \geq 0$, $\frac{\partial}{\partial c} H(b, c, d) = 1 - \lambda a \geq 0$, and $\frac{\partial}{\partial d} H(b, c, d) = \frac{\lambda}{2} (a - f'(d)) \geq 0$. We also have $H(b, b, b) = b$. Therefore,

$$m = H(m, m, m) \leq u_j^{n+1} \leq H(M, M, M) = M. \quad \square$$

Given a scheme in the form of (2.1), assuming $\bar{u}_j^n \in [m, M]$ for all j , we would like to modify the nodal values $u_{j+\frac{1}{2}}^\pm$ in some way so that $\bar{u}_j^{n+1} \in [m, M]$. We will first show a sufficient condition for (2.1) to satisfy $\bar{u}_j^{n+1} \in [m, M]$, in which we need the Legendre Gauss–Lobatto quadrature points (see Table 2.1 and [11] for more details). Consider the N -point Legendre Gauss–Lobatto quadrature rule on the interval $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, which is exact for the integral of polynomials of degree up to $2N - 3$. We denote these quadrature points on I_j as

Table 2.1

The CFL number (2.6) for $2 \leq k \leq 5$ and the Gauss–Lobatto quadrature points on $[-\frac{1}{2}, \frac{1}{2}]$.

k	CFL	Quadrature points on $[-\frac{1}{2}, \frac{1}{2}]$
2	$\frac{1}{6}$	$\{-\frac{1}{2}, 0, \frac{1}{2}\}$
3	$\frac{1}{6}$	$\{-\frac{1}{2}, 0, \frac{1}{2}\}$
4	$\frac{1}{12}$	$\{-\frac{1}{2}, -\frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}, \frac{1}{2}\}$
5	$\frac{1}{12}$	$\{-\frac{1}{2}, -\frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}, \frac{1}{2}\}$

$$S_j = \left\{ x_{j-\frac{1}{2}} = \hat{x}_j^1, \hat{x}_j^2, \dots, \hat{x}_j^{N-1}, \hat{x}_j^N = x_{j+\frac{1}{2}} \right\}. \tag{2.4}$$

Define $\hat{v}_\alpha = p_j(\hat{x}_j^\alpha)$ for $\alpha = 1, \dots, N$, and let \hat{w}_α be the quadrature weights for the interval $[-\frac{1}{2}, \frac{1}{2}]$ such that $\sum_{\alpha=1}^N \hat{w}_\alpha = 1$. Choose N to be the smallest integer satisfying $2N - 3 \geq k$, then

$$\bar{u}_j^n = \frac{1}{\Delta x} \int_{I_j} p_j(x) dx = \sum_{\alpha=1}^N \hat{w}_\alpha \hat{v}_\alpha. \tag{2.5}$$

Motivated by the approach in [20], our result is

Theorem 2.2. Consider a finite volume scheme or the scheme satisfied by the cell averages of the DG method (2.1) with the Lax–Friedrichs flux (2.2), associated with the approximation polynomials $p_j(x)$ of degree k (either reconstruction or DG polynomials). For convenience, in the interval I_j , we denote $\hat{v}_0 = u_{j-\frac{1}{2}}^-$ and $\hat{v}_{N+1} = u_{j+\frac{1}{2}}^+$. If all the values \hat{v}_α ($\alpha = 0, \dots, N + 1$) and \bar{u}_j^n are in the range $[m, M]$, then $\bar{u}_j^{n+1} \in [m, M]$ under the CFL condition

$$\lambda a \leq \min_x \hat{w}_\alpha. \tag{2.6}$$

Proof. Notice that we have

$$h(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+) - h(u_{j-\frac{1}{2}}^-, u_{j-\frac{1}{2}}^+) = h(\hat{v}_N, \hat{v}_{N+1}) - h(\hat{v}_0, \hat{v}_1) = \sum_{\alpha=1}^N [h(\hat{v}_\alpha, \hat{v}_{\alpha+1}) - h(\hat{v}_{\alpha-1}, \hat{v}_\alpha)].$$

With (2.5), we can rewrite (2.1) as:

$$\begin{aligned} \bar{u}_j^{n+1} &= \bar{u}_j^n - \lambda \left[h(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+) - h(u_{j-\frac{1}{2}}^-, u_{j-\frac{1}{2}}^+) \right] = \sum_{\alpha=1}^N \hat{w}_\alpha \hat{v}_\alpha - \lambda \left[h(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+) - h(u_{j-\frac{1}{2}}^-, u_{j-\frac{1}{2}}^+) \right] \\ &= \sum_{\alpha=1}^N \hat{w}_\alpha \left[\hat{v}_\alpha - \frac{\lambda}{\hat{w}_\alpha} [h(\hat{v}_\alpha, \hat{v}_{\alpha+1}) - h(\hat{v}_{\alpha-1}, \hat{v}_\alpha)] \right] = \sum_{\alpha=1}^N \hat{w}_\alpha H_j^\alpha, \end{aligned}$$

where

$$H_j^\alpha = \hat{v}_\alpha - \frac{\lambda}{\hat{w}_\alpha} [h(\hat{v}_\alpha, \hat{v}_{\alpha+1}) - h(\hat{v}_{\alpha-1}, \hat{v}_\alpha)], \quad \alpha = 1, \dots, N. \tag{2.7}$$

Notice that all the (2.7) are of the type (2.3), with $\frac{\lambda}{\hat{w}_\alpha}$ in the place of λ . Therefore, $H_j^\alpha \in [m, M]$ under the CFL condition (2.6). Hence $\bar{u}_j^{n+1} \in [m, M]$ since it is a convex combination of all the H_j^α . \square

Remark 2.3. It would look that the required condition in the previous theorem is too strong, since it involves restrictions on the interior point values \hat{v}_α ($\alpha = 2, \dots, N - 1$) which do not explicitly appear in the scheme (2.1). It would look natural to require only $\hat{v}_0 = u_{j-\frac{1}{2}}^-$, $\hat{v}_1 = u_{j-\frac{1}{2}}^+$, $\hat{v}_N = u_{j+\frac{1}{2}}^-$, $\hat{v}_{N+1} = u_{j+\frac{1}{2}}^+$ and \bar{u}_j^n are in $[m, M]$, as these are the values which explicitly appear in the scheme (2.1). However, it is very easy to show that these requirements alone will not ensure $\bar{u}_j^{n+1} \in [m, M]$. For example, consider the linear convection equation $u_t + u_x = 0$, for which the Lax–Friedrichs flux becomes $h(u, v) = u$. Let $[0, 1]$ be the desired range, and $\bar{u}_j^n = u_{j-\frac{1}{2}}^- = u_{j-\frac{1}{2}}^+ = u_{j+\frac{1}{2}}^+ = 1$ and $u_{j+\frac{1}{2}}^- = 0.99$, then (2.1) gives $\bar{u}_j^{n+1} = 1 + 0.01\lambda > 1$ for any finite CFL number $\lambda > 0$. The additional restrictions for the interior point values \hat{v}_α ($\alpha = 2, \dots, N - 1$) will however ensure $\bar{u}_j^{n+1} \in [m, M]$ through Theorem 2.2.

This theorem tells us that for the scheme (2.1), we need to modify $p_j(x)$ such that $p_j(x) \in [m, M]$ for all $x \in S_j$ where S_j is set of the Legendre Gauss–Lobatto quadrature points for I_j . For all j , assume $\bar{u}_j^n \in [m, M]$, we use the modified polynomial $\tilde{p}_j(x)$ after the limiter (1.4) instead of $p_j(x)$, i.e.,

$$\tilde{p}_j(x) = \theta(p_j(x) - \bar{u}_j^n) + \bar{u}_j^n, \quad \theta = \min \left\{ \left| \frac{M - \bar{u}_j^n}{M_j - \bar{u}_j^n} \right|, \left| \frac{m - \bar{u}_j^n}{m_j - \bar{u}_j^n} \right|, 1 \right\}, \tag{2.8}$$

with

$$M_j = \max_{x \in I_j} p_j(x), \quad m_j = \min_{x \in I_j} p_j(x). \tag{2.9}$$

Let $\bar{u}_{j-\frac{1}{2}}^+ = \bar{p}_j(x_{j-\frac{1}{2}})$ and $\bar{u}_{j+\frac{1}{2}}^- = \bar{p}_j(x_{j+\frac{1}{2}})$. We get the revised scheme of (2.1):

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \lambda \left[h(\bar{u}_{j+\frac{1}{2}}^-, \bar{u}_{j+\frac{1}{2}}^+) - h(\bar{u}_{j-\frac{1}{2}}^-, \bar{u}_{j-\frac{1}{2}}^+) \right]. \tag{2.10}$$

We will show in the next Lemma that this limiter does not destroy the uniform high order of accuracy.

Lemma 2.4. Assume $\bar{u}_j^n \in [m, M]$, then (2.8) and (2.9) gives a $(k + 1)$ th order accurate limiter.

Proof. We need to show $\bar{p}_j(x) - p_j(x) = O(\Delta x^{k+1})$ for any $x \in I_j$. We only prove the case that $p_j(x)$ is not a constant and $\theta = \left| \frac{M - \bar{u}_j^n}{M_j - \bar{u}_j^n} \right|$, the other cases being similar. Since $\bar{u}_j^n \leq M$ and $\bar{u}_j^n \leq M_j$, we have $\theta = (M - \bar{u}_j^n)/(M_j - \bar{u}_j^n)$. Therefore,

$$\bar{p}_j(x) - p_j(x) = \theta(p_j(x) - \bar{u}_j^n) + \bar{u}_j^n - p_j(x) = (\theta - 1)(p_j(x) - \bar{u}_j^n) = \frac{M - M_j}{M_j - \bar{u}_j^n} (p_j(x) - \bar{u}_j^n) = (M - M_j) \frac{p_j(x) - \bar{u}_j^n}{M_j - \bar{u}_j^n}.$$

By the definition of θ in (2.8), $\theta = \left| \frac{M - \bar{u}_j^n}{M_j - \bar{u}_j^n} \right|$ implies that $\theta = \left| \frac{M - \bar{u}_j^n}{M_j - \bar{u}_j^n} \right| > 1$, i.e. there is overshoot $M_j > M$, and the overshoot $M_j - M = O(\Delta x^{k+1})$ since $p_j(x)$ is an approximation with error Δx^{k+1} . Thus we only need to prove that $\left| \frac{p_j(x) - \bar{u}_j^n}{M_j - \bar{u}_j^n} \right| \leq C_k$, where C_k is a constant depending only on the polynomial degree k . In [16], $C_2 = 3$ is proved. We now prove the existence of C_k for any k . Assume $p_j(x) = a_0 + a_1 \left(\frac{x-x_j}{\Delta x}\right) + \dots + a_k \left(\frac{x-x_j}{\Delta x}\right)^k$ and $p(x) = a_0 + a_1 x + \dots + a_k x^k$, then the cell average of $p(x)$ on $I = [-\frac{1}{2}, \frac{1}{2}]$ is $\bar{p} = \bar{u}_j^n$ and $\max_{x \in I} p(x) = M_j$. So we have

$$\max_{x \in I_j} \left| \frac{p_j(x) - \bar{u}_j^n}{M_j - \bar{u}_j^n} \right| = \max_{x \in I} \left| \frac{p(x) - \bar{p}}{\max_{y \in I} p(y) - \bar{p}} \right|.$$

Notice that it suffices to prove the existence of C_k such that

$$\left| \frac{\min_{x \in I} p(x) - \bar{p}}{\max_{x \in I} p(x) - \bar{p}} \right| \leq C_k.$$

It is easy to check that $|\min_{x \in I} p(x) - \bar{p}|$ and $|\max_{x \in I} p(x) - \bar{p}|$ are both norms on the finite dimensional linear space P^k , which is the set of all polynomials of degree k . Any two norms on this finite dimensional space are equivalent, hence their ratio is bounded by a constant C_k . □

Notice that in (2.9) we need to evaluate the maximum/minimum of a polynomial. We prefer to avoid evaluating the extrema of a polynomial, especially since we will extend the method to two dimensions. Since we only need to control the values at quadrature points, we could replace (2.9) by

$$M_j = \max_{x \in S_j} p_j(x), \quad m_j = \min_{x \in S_j} p_j(x), \tag{2.11}$$

and the limiter (2.8) and (2.11) is sufficient to enforce

$$\bar{p}_j(x) \in [m, M], \quad \forall x \in S_j.$$

As to the accuracy, (2.11) is a less restrictive limiter than (2.9), so the accuracy will not be destroyed. Also, it is a conservative limiter because the it does not change the cell average of the polynomial.

We now have the following theorem.

Theorem 2.5. Assume $p_j(x)$ has degree k , m and M are defined in (1.3). If (2.1) is $(k + 1)$ th order accurate for smooth solutions and $\bar{u}_j^n \in [m, M]$, then the revised scheme (2.8), (2.11) and (2.10) is also $(k + 1)$ th order accurate and satisfies the maximum principle $\bar{u}_j^{n+1} \in [m, M]$, under the CFL condition (2.6). See Table 2.1 for the CFL number for $2 \leq k \leq 5$.

Proof. Let $\bar{v}_\alpha = \bar{p}_j(x_\alpha^z)$ for $\alpha = 1, \dots, N$, $\bar{v}_0 = \bar{p}_{j-1}(x_{j-\frac{1}{2}})$ and $\bar{v}_{N+1} = \bar{p}_{j+1}(x_{j+\frac{1}{2}})$. The limiter (2.8) and (2.11) ensures all the revised values \bar{v}_α are in $[m, M]$. Theorem 2.2 then ensures the maximum principle $\bar{u}_j^{n+1} \in [m, M]$. □

Remark 2.6. We recall that for $k = 2$ the CFL condition for linear stability for the DG scheme [3] is $\lambda a \leq \frac{1}{5}$, which is comparable to our CFL restriction.

Remark 2.7. From the proof of Theorem 2.2, we can see that any type of quadrature rule will work as long as the quadrature points include the two cell ends. It would appear that there is a possibility to achieve a larger CFL number if we can find a better quadrature in the sense that $\min_x \hat{w}_x$ is larger. However, for $k = 2, 3$, we have checked that the Gauss–Lobatto quadrature is the best choice.

Remark 2.8. Although we use Lax–Friedrichs flux in the theorem, any other monotone flux will also work under the corresponding CFL condition.

2.2. Higher order time discretization

We will use strong stability preserving (SSP) high order time discretizations. For more details, see [25,23,7,6]. For example, the third order SSP Runge–Kutta method [25] (with the CFL coefficient $c = 1$) is

$$\begin{aligned} u^{(1)} &= u^n + \Delta t F(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}(u^{(1)} + \Delta t F(u^{(1)})), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}(u^{(2)} + \Delta t F(u^{(2)})), \end{aligned} \tag{2.12}$$

where $F(u)$ is the spatial operator, and the third order SSP multi-step method [23] (with the CFL coefficient $c = \frac{1}{3}$) is

$$u^{n+1} = \frac{16}{27}(u^n + 3\Delta t F(u^n)) + \frac{11}{27}(u^{n-3} + \frac{12}{11}\Delta t F(u^{n-3})). \tag{2.13}$$

Here, the CFL coefficient c for a SSP time discretization refers to the fact that, if we assume the Euler forward time discretization for solving the equation $u_t = F(u)$ is stable in a norm or a semi-norm under a time step restriction $\Delta t \leq \Delta t_0$, then the high order SSP time discretization is also stable in the same norm or semi-norm under the time step restriction $\Delta t \leq c\Delta t_0$.

Since a SSP high order time discretization is a convex combinations of Euler forward, the full scheme with a high order SSP time discretization will still satisfy the maximum principle.

2.3. Implementation for the DG scheme

At time level n , assuming the DG polynomial in cell I_j is $p_j(x)$ with degree k , and the cell average of $p_j(x)$ is $\bar{u}_j^n \in [m, M]$, where m and M are defined in (1.3), then the algorithm flowchart of our algorithm for the Euler forward is

- Evaluate the point values of $p_j(x)$ in S_j to get m_j and M_j in (2.11).
- Compute $\tilde{p}_j(x)$ in (2.8).
- Use $\tilde{p}_j(x)$ instead of $p_j(x)$ in the DG scheme with Euler forward in time under the CFL condition in Theorem 2.5. For the time evolution of the cell average, mathematically it is equivalent to plugging the revised nodal values $\tilde{u}_{j-\frac{1}{2}}^+ = \tilde{p}_j(x_{j-\frac{1}{2}})$ and $\tilde{u}_{j+\frac{1}{2}}^- = \tilde{p}_j(x_{j+\frac{1}{2}})$ in (2.10).

For SSP high order time discretizations, we need to use the limiter in each stage for a Runge–Kutta method or in each step for a multi-step method.

2.4. Implementation for the finite volume WENO scheme

The implementation for a finite volume WENO scheme [17,2,24] is slightly different since there is not a specific reconstruction polynomial in each cell. After the WENO reconstruction, we have the high order accurate nodal values $u_{j-\frac{1}{2}}^+$ and $u_{j+\frac{1}{2}}^-$ on each cell I_j . To have a high order approximation polynomial in order to apply our limiter, we can use the information of the cell averages and the nodal values to construct one.

For example, consider the fifth order accurate finite volume WENO scheme. Assume we already have the nodal values $u_{j-\frac{1}{2}}^+$ and $u_{j+\frac{1}{2}}^-$ from the WENO reconstruction, we will need a polynomial $p_j(x)$ of degree four such that $p_j(x_{j-\frac{1}{2}}) = u_{j-\frac{1}{2}}^+$, $p_j(x_{j+\frac{1}{2}}) = u_{j+\frac{1}{2}}^-$ and the cell average of $p_j(x)$ over I_j is \bar{u}_j^n . Moreover, $p_j(x)$ should be a fifth order accurate approximation to the exact solution on I_j . To this end, we choose to use the Hermite type reconstruction of degree four; i.e., the polynomial $p_j(x)$ should satisfy

$$\frac{1}{\Delta x} \int_{I_i} p_j(x) dx = \bar{u}_i^n, \quad i = j - 1, j, j + 1; \quad \text{and} \quad p_j(x_{j-\frac{1}{2}}) = u_{j-\frac{1}{2}}^+, \quad p_j(x_{j+\frac{1}{2}}) = u_{j+\frac{1}{2}}^-.$$

If the reconstruction is written as

$$p_j(x) = a_4(x - x_j)^4 + a_3(x - x_j)^3 + a_2(x - x_j)^2 + a_1(x - x_j) + a_0,$$

then the coefficients can be given explicitly as

$$\begin{aligned}
 a_0 &= \frac{\bar{u}_{j-1}^n + 298\bar{u}_j^n + \bar{u}_{j+1}^n - 54(u_{j-\frac{1}{2}}^+ + u_{j+\frac{1}{2}}^-)}{192}, \\
 a_1 &= \frac{\bar{u}_{j-1}^n - \bar{u}_{j+1}^n - 10(u_{j-\frac{1}{2}}^+ - u_{j+\frac{1}{2}}^-)}{8\Delta x}, \\
 a_2 &= \frac{-(\bar{u}_{j-1}^n + 58\bar{u}_j^n + \bar{u}_{j+1}^n) + 30(u_{j-\frac{1}{2}}^+ + u_{j+\frac{1}{2}}^-)}{8\Delta x^2}, \\
 a_3 &= \frac{\bar{u}_{j+1}^n - \bar{u}_{j-1}^n + 2(u_{j-\frac{1}{2}}^+ - u_{j+\frac{1}{2}}^-)}{\Delta x^3}, \\
 a_4 &= \frac{5\bar{u}_{j-1}^n + 50\bar{u}_j^n + 5\bar{u}_{j+1}^n - 30(u_{j-\frac{1}{2}}^+ + u_{j+\frac{1}{2}}^-)}{12\Delta x^4}.
 \end{aligned}$$

In summary, the algorithm for the limiter on the fifth order finite volume WENO with the Euler forward time discretization is:

- Construct the Hermite type polynomials $p_i(x)$.
- Evaluate point values of $p_j(x)$ in S_j to get m_j, M_j in (2.11).
- Compute $\tilde{p}_j(x)$ in (2.8).
- Compute the revised nodal values $\tilde{u}_{j-\frac{1}{2}}^+ = \tilde{p}_j(x_{j-\frac{1}{2}})$ and $\tilde{u}_{j+\frac{1}{2}}^- = \tilde{p}_j(x_{j+\frac{1}{2}})$, then plug them in (2.10) with the CFL condition $\lambda a \leq \frac{1}{12}$.

For SSP high order time discretizations, we need to use the limiter for each stage for Runge–Kutta methods or for each step for multi-step methods.

3. High order schemes satisfying the maximum principle in two dimensions

In this section we extend our limiter to finite volume or DG schemes of $(k + 1)$ th order accuracy on rectangular meshes solving two dimensional conservation law

$$u_t + f(u)_x + g(u)_y = 0, \quad u(x, y, 0) = u_0(x, y). \tag{3.1}$$

In this section, m and M refer to $M = \max_{x,y} u_0(x, y)$, $m = \min_{x,y} u_0(x, y)$. As before, we would only need to discuss the Euler forward in time. SSP high order time discretizations will keep the maximum principle.

3.1. Decomposition of the two-dimensional scheme into convex combination of one-dimensional schemes

For simplicity we assume we have a uniform rectangular mesh. At time level n , we have an approximation polynomial $p_{ij}(x, y)$ with the cell average \bar{u}_{ij}^n on the (i, j) cell $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. Let P^k denote the set of two-variable polynomials of degree k and Q^k denote the set of tensor products of single variable polynomials of degree k . We consider $p_{ij}(x, y) \in P^k$ for the DG method and $p_{ij}(x, y) \in Q^k$ for the finite volume ENO and WENO finite volume method [2,24]. Let $u_{i-\frac{1}{2}j}^+(y)$, $u_{i+\frac{1}{2}j}^-(y)$, $u_{ij-\frac{1}{2}}^+(x)$, $u_{ij+\frac{1}{2}}^-(x)$ denote the traces of $p_{ij}(x, y)$ on the four edges, respectively, see Fig. 3.1. All of the traces are single variable polynomials of degree k . A finite volume scheme or the scheme satisfied by the cell averages of a DG method for (3.1) on a rectangular mesh can be written as

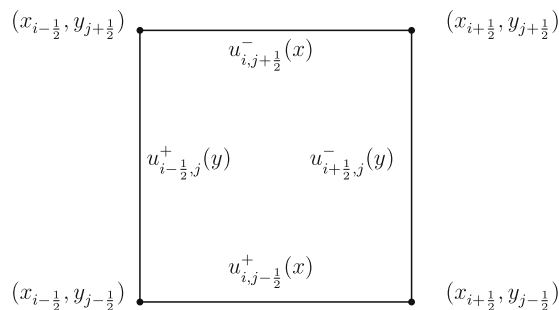


Fig. 3.1. The traces of $p_{ij}(x, y)$.

$$\bar{u}_{ij}^{n+1} = \bar{u}_{ij}^n - \frac{\Delta t}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} h_1 \left[u_{i-\frac{1}{2},j}^-(y), u_{i+\frac{1}{2},j}^+(y) \right] - h_1 \left[u_{i-\frac{1}{2},j}^-(y), u_{i-\frac{1}{2},j}^+(y) \right] dy - \frac{\Delta t}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h_2 \left[u_{i,j+\frac{1}{2}}^-(x), u_{i,j+\frac{1}{2}}^+(x) \right] - h_2 \left[u_{i,j-\frac{1}{2}}^-(x), u_{i,j-\frac{1}{2}}^+(x) \right] dx, \tag{3.2}$$

where $h_1(\cdot, \cdot), h_2(\cdot, \cdot)$ are monotone fluxes. We will use the Lax–Friedrichs flux as an example throughout the paper:

$$h_1(u, v) = \frac{1}{2} [f(u) + f(v) - a_1(v - u)], \quad a_1 = \max |f'(u)|,$$

$$h_2(u, v) = \frac{1}{2} [g(u) + g(v) - a_2(v - u)], \quad a_2 = \max |g'(u)|.$$

The integrals in (3.2) can be approximated by quadratures with sufficient accuracy. Let us assume that we use a Gauss quadrature with L points, which is exact for single variable polynomials of degree k . We assume $S_x^\beta = \{x_i^\beta : \beta = 1, \dots, L\}$ denote the Gauss quadrature points on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, and $S_y^\beta = \{y_j^\beta : \beta = 1, \dots, L\}$ denote the Gauss quadrature points on $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. For instance, $(x_{i-\frac{1}{2}}, y_j^\beta)$ ($\beta = 1, \dots, L$) are the Gauss quadrature points on the left edge of the (i, j) cell. The subscript β will denote the values at the Gauss quadrature points, for instance, $u_{i-\frac{1}{2},j}^+ = u_{i-\frac{1}{2},j}^+(y_j^\beta)$. Also, w_β denotes the corresponding quadrature weight on interval $[-\frac{1}{2}, \frac{1}{2}]$, so that $\sum_{\beta=1}^L w_\beta = 1$. We will still need to use the Gauss–Lobatto quadrature rule, and we distinguish the two quadrature rules by adding hats to the Gauss–Lobatto points, i.e., $\widehat{S}_x^\alpha = \{\hat{x}_i^\alpha : \alpha = 1, \dots, N\}$ will denote the Gauss–Lobatto quadrature points on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, and $\widehat{S}_y^\alpha = \{\hat{y}_j^\alpha : \alpha = 1, \dots, N\}$ will denote the Gauss–Lobatto quadrature points on $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. Subscripts or superscripts β and γ will be used only for Gauss quadrature points and α only for Gauss–Lobatto points.

Then the scheme (3.2) becomes

$$\begin{aligned} \bar{u}_{ij}^{n+1} &= \bar{u}_{ij}^n - \frac{\Delta t}{\Delta x \Delta y} \sum_{\beta=1}^L \left[h_1 \left(u_{i+\frac{1}{2},j}^-, u_{i+\frac{1}{2},j}^+ \right) - h_1 \left(u_{i-\frac{1}{2},j}^-, u_{i-\frac{1}{2},j}^+ \right) \right] w_\beta \Delta y - \frac{\Delta t}{\Delta x \Delta y} \sum_{\beta=1}^L \left[h_2 \left(u_{\beta,j+\frac{1}{2}}^-, u_{\beta,j+\frac{1}{2}}^+ \right) - h_2 \left(u_{\beta,j-\frac{1}{2}}^-, u_{\beta,j-\frac{1}{2}}^+ \right) \right] w_\beta \Delta x \\ &= \bar{u}_{ij}^n - \lambda_1 \sum_{\beta=1}^L w_\beta \left[h_1 \left(u_{i+\frac{1}{2},j}^-, u_{i+\frac{1}{2},j}^+ \right) - h_1 \left(u_{i-\frac{1}{2},j}^-, u_{i-\frac{1}{2},j}^+ \right) \right] - \lambda_2 \sum_{\beta=1}^L w_\beta \left[h_2 \left(u_{\beta,j+\frac{1}{2}}^-, u_{\beta,j+\frac{1}{2}}^+ \right) - h_2 \left(u_{\beta,j-\frac{1}{2}}^-, u_{\beta,j-\frac{1}{2}}^+ \right) \right], \end{aligned} \tag{3.3}$$

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

Let $\bar{u}_{\beta,j}^n$ denote the average of $p_{ij}(x_i^\beta, y)$ over $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ and $\bar{u}_{i,\beta}^n$ denote the average of $p_{ij}(x, y_j^\beta)$ over $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, then the cell average \bar{u}_{ij}^n is

$$\bar{u}_{ij}^n = \frac{1}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_{ij}(x, y) dx dy = \frac{1}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left(\sum_{\beta=1}^L p_{ij}(x_i^\beta, y) w_\beta \Delta x \right) dy = \sum_{\beta=1}^L w_\beta \left(\frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} p_{ij}(x_i^\beta, y) dy \right) = \sum_{\beta=1}^L w_\beta \bar{u}_{\beta,j}^n, \tag{3.4}$$

where we have used the property that the quadrature rules are exact for polynomials of degree k . Similarly we have

$$\bar{u}_{ij}^n = \sum_{\beta=1}^L w_\beta \bar{u}_{i,\beta}^n. \tag{3.5}$$

Plugging (3.4) and (3.5) into (3.3), we get

$$\begin{aligned} \bar{u}_{ij}^{n+1} &= \bar{u}_{ij}^n - \lambda_1 \sum_{\beta=1}^L w_\beta \left[h_1 \left(u_{i+\frac{1}{2},j}^-, u_{i+\frac{1}{2},j}^+ \right) - h_1 \left(u_{i-\frac{1}{2},j}^-, u_{i-\frac{1}{2},j}^+ \right) \right] - \lambda_2 \sum_{\beta=1}^L w_\beta \left[h_2 \left(u_{\beta,j+\frac{1}{2}}^-, u_{\beta,j+\frac{1}{2}}^+ \right) - h_2 \left(u_{\beta,j-\frac{1}{2}}^-, u_{\beta,j-\frac{1}{2}}^+ \right) \right] \\ &= \frac{a_1 \lambda_1}{a_1 \lambda_1 + a_2 \lambda_2} \bar{u}_{ij}^n - \lambda_1 \sum_{\beta=1}^L w_\beta \left[h_1 \left(u_{i+\frac{1}{2},j}^-, u_{i+\frac{1}{2},j}^+ \right) - h_1 \left(u_{i-\frac{1}{2},j}^-, u_{i-\frac{1}{2},j}^+ \right) \right] + \frac{a_2 \lambda_2}{a_1 \lambda_1 + a_2 \lambda_2} \bar{u}_{ij}^n \\ &\quad - \lambda_2 \sum_{\beta=1}^L w_\beta \left[h_2 \left(u_{\beta,j+\frac{1}{2}}^-, u_{\beta,j+\frac{1}{2}}^+ \right) - h_2 \left(u_{\beta,j-\frac{1}{2}}^-, u_{\beta,j-\frac{1}{2}}^+ \right) \right] \\ &= \frac{a_1 \lambda_1}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^L w_\beta \bar{u}_{i,\beta}^n - \lambda_1 \sum_{\beta=1}^L w_\beta \left[h_1 \left(u_{i+\frac{1}{2},j}^-, u_{i+\frac{1}{2},j}^+ \right) - h_1 \left(u_{i-\frac{1}{2},j}^-, u_{i-\frac{1}{2},j}^+ \right) \right] \\ &\quad + \frac{a_2 \lambda_2}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^L w_\beta \bar{u}_{\beta,j}^n - \lambda_2 \sum_{\beta=1}^L w_\beta \left[h_2 \left(u_{\beta,j+\frac{1}{2}}^-, u_{\beta,j+\frac{1}{2}}^+ \right) - h_2 \left(u_{\beta,j-\frac{1}{2}}^-, u_{\beta,j-\frac{1}{2}}^+ \right) \right] \\ &= \frac{a_1 \lambda_1}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^L w_\beta \left[\bar{u}_{i,\beta}^n - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} \left(h_1 \left(u_{i+\frac{1}{2},j}^-, u_{i+\frac{1}{2},j}^+ \right) - h_1 \left(u_{i-\frac{1}{2},j}^-, u_{i-\frac{1}{2},j}^+ \right) \right) \right] \\ &\quad + \frac{a_2 \lambda_2}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^L w_\beta \left[\bar{u}_{\beta,j}^n - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_2} \left(h_2 \left(u_{\beta,j+\frac{1}{2}}^-, u_{\beta,j+\frac{1}{2}}^+ \right) - h_2 \left(u_{\beta,j-\frac{1}{2}}^-, u_{\beta,j-\frac{1}{2}}^+ \right) \right) \right]. \end{aligned}$$

Let us introduce the formal one-dimensional schemes

$$H_x^{i,\beta} = \bar{u}_{i,\beta}^n - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} \left[h_1 \left(u_{i+\frac{1}{2},\beta}^-, u_{i+\frac{1}{2},\beta}^+ \right) - h_1 \left(u_{i-\frac{1}{2},\beta}^-, u_{i-\frac{1}{2},\beta}^+ \right) \right], \tag{3.6}$$

$$H_y^{\beta,j} = \bar{u}_{\beta,j}^n - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_2} \left[h_2 \left(u_{\beta,j+\frac{1}{2}}^-, u_{\beta,j+\frac{1}{2}}^+ \right) - h_2 \left(u_{\beta,j-\frac{1}{2}}^-, u_{\beta,j-\frac{1}{2}}^+ \right) \right], \tag{3.7}$$

then the scheme (3.3) is equivalent to

$$\bar{u}_{ij}^{n+1} = \frac{a_1 \lambda_1}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^L w_\beta H_x^{i,\beta} + \frac{a_2 \lambda_2}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^L w_\beta H_y^{\beta,j},$$

which tells us that \bar{u}_{ij}^{n+1} is a convex combinations of $H_x^{i,\beta}$ and $H_y^{\beta,j}$ ($\beta = 1, \dots, L$). Therefore, to have a maximum principle $\bar{u}_{ij}^{n+1} \in [m, M]$ for (3.3), it suffices to enforce $H_x^{i,\beta}, H_y^{\beta,j} \in [m, M]$. We will only show how to deal with $H_x^{i,\beta}$. The discussion for $H_y^{\beta,j}$ is similar.

Notice that (3.6) is of the type (2.1), associated with $p_{ij}(x, y_j^\beta)$, i.e., $\bar{u}_{i,\beta}^n$ is the cell average and $u_{i+\frac{1}{2},\beta}^-, u_{i-\frac{1}{2},\beta}^+$ are the nodal values of $p_{ij}(x, y_j^\beta)$ on the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. Therefore, it is straightforward to use the limiter in Section 2. If we can enforce $\bar{u}_{i,\beta}^n \in [m, M]$, then we can just use Theorem 2.5 for the formal one-dimensional scheme (3.6). We will show the details of the enforcement of $\bar{u}_{i,\beta}^n \in [m, M]$ in the next subsections. The CFL condition of the method for the two dimensional case is the same as that in one-dimensional case

$$a_1 \lambda_1 + a_2 \lambda_2 \leq \min_{\alpha=1, \dots, N} \hat{w}_\alpha.$$

3.2. Implementation for the DG method

At time level n , assuming the DG polynomial on the (i, j) cell is $p_{ij}(x, y)$ and the cell average of $p_{ij}(x, y)$ is $\bar{u}_{ij}^n \in [m, M]$, we use the following $\tilde{p}_{ij}(x, y)$ to replace $p_{ij}(x, y)$:

$$\tilde{p}_{ij}(x, y) = \theta(p_{ij}(x, y) - \bar{u}_{ij}^n) + \bar{u}_{ij}^n, \quad \theta = \min \left\{ \left| \frac{M - \bar{u}_{ij}^n}{M_{ij} - \bar{u}_{ij}^n} \right|, \left| \frac{m - \bar{u}_{ij}^n}{m_{ij} - \bar{u}_{ij}^n} \right|, 1 \right\}, \tag{3.8}$$

$$M_{ij} = \max_{(x,y) \in S_{ij}} p_{ij}(x, y), \quad m_{ij} = \min_{(x,y) \in S_{ij}} p_{ij}(x, y), \tag{3.9}$$

where S_{ij} is a set of finitely many points inside the (i, j) cell, to be specified later. Then $\tilde{p}_{ij}(x, y) \in [m, M]$ for all $(x, y) \in S_{ij}$. The maintenance of uniform high order accuracy by this limiter can be proved similarly as that in the one-dimensional case. According to the discussion in the previous subsection, we need $\tilde{p}_{ij}(x, y)$ to satisfy the condition that all of the ‘‘one dimensional cell averages’’ $\bar{u}_{i,\beta}^n, \bar{u}_{\beta,j}^n$ and the corresponding point values of the ‘‘one-dimensional polynomials’’ $\tilde{p}_{ij}(x, y_j^\beta), \tilde{p}_{ij}(x_i^\beta, y)$ are in the range $[m, M]$.

By the properties of the quadrature rule,

$$\bar{u}_{i,\beta}^n = \sum_{\gamma=1}^L \tilde{p}_{ij}(x_i^\gamma, y_j^\beta) w_\gamma, \quad \bar{u}_{\beta,j}^n = \sum_{\gamma=1}^L \tilde{p}_{ij}(x_i^\beta, y_j^\gamma) w_\gamma.$$

Thus, to enforce $\bar{u}_{i,\beta}^n, \bar{u}_{\beta,j}^n \in [m, M]$, it suffices to include the tensor product of the Gauss quadrature points (x_i^β, y_j^γ) ($\beta, \gamma = 1, \dots, L$) in S_{ij} .

We use \otimes to denote the tensor product, for instance, $S_i^x \otimes S_j^y = \{(x, y) : x \in S_i^x, y \in S_j^y\}$. Now we can define the set S_{ij} as

$$S_{ij} = (S_i^x \otimes S_j^y) \cup (S_i^x \otimes \hat{S}_j^y) \cup (\hat{S}_i^x \otimes S_j^y). \tag{3.10}$$

For example, for $k = 2$, we need to use three-point Gauss quadrature. On a scaled square $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$, $S^x = S^y = \{-\frac{\sqrt{15}}{10}, 0, \frac{\sqrt{15}}{10}\}$ with weights $(\frac{5}{18}, \frac{4}{9}, \frac{5}{18})$ and $\hat{S}^x = \hat{S}^y = \{-\frac{1}{2}, 0, \frac{1}{2}\}$ with weights $(\frac{1}{6}, \frac{2}{3}, \frac{1}{6})$. S defined in (3.10) contains 21 points.

In summary, the algorithm for our limiter on the DG method solving (3.1) is

- Evaluate the point values of $p_{ij}(x)$ in (3.10) to get m_{ij}, M_{ij} in (3.9).
- Compute $\tilde{p}_{ij}(x)$ in (3.8).
- Use $\tilde{p}_{ij}(x)$ instead of $p_{ij}(x)$ in the DG scheme with the CFL condition $a_1 \lambda_1 + a_2 \lambda_2 \leq \min_{\alpha=1, \dots, N} \hat{w}_\alpha$.

3.3. Implementation for the fifth order finite volume WENO scheme

First, Let us recall the dimension by dimension reconstruction procedure of the finite volume WENO for (3.1) in [22,24]. At time level n , given the cell average \bar{u}_{ij} . Let $\bar{u}_{i-\frac{1}{2}j}$, $\bar{u}_{i+\frac{1}{2}j}$, $\bar{u}_{ij-\frac{1}{2}}$ and $\bar{u}_{ij+\frac{1}{2}}^n$ denote the approximation to the averages of the exact solution along the left, right, bottom and top edges of the (i,j) cell, respectively.

Step 1. We perform two one-dimensional WENO reconstructions using the two-dimensional cell averages to get the four edge averages (see [2,22,24] for more details):

$$\{\bar{u}_{ij}^n\} \rightarrow \{\bar{u}_{i+\frac{1}{2}j}^-, \bar{u}_{i+\frac{1}{2}j}^+\} \quad \text{for fixed } j \tag{3.11}$$

and

$$\{\bar{u}_{ij}^n\} \rightarrow \{\bar{u}_{ij+\frac{1}{2}}^-, \bar{u}_{ij+\frac{1}{2}}^+\} \quad \text{for fixed } i. \tag{3.12}$$

Step 2. We perform one-dimensional WENO reconstructions using the edge averages to get point values at the Gauss quadrature points (see the Appendix for more details):

$$\{\bar{u}_{i+\frac{1}{2}j}^-\} \rightarrow \{u_{i+\frac{1}{2},\beta}^-\}, \quad \{\bar{u}_{i+\frac{1}{2}j}^+\} \rightarrow \{u_{i+\frac{1}{2},\beta}^+\} \quad \text{for fixed } i$$

and

$$\{\bar{u}_{ij+\frac{1}{2}}^-\} \rightarrow \{u_{\beta,j+\frac{1}{2}}^-\}, \quad \{\bar{u}_{ij+\frac{1}{2}}^+\} \rightarrow \{u_{\beta,j+\frac{1}{2}}^+\} \quad \text{for fixed } j.$$

Now, let us consider the implementation of the limiter. The difference from the DG method is that the nodal values $u_{i+\frac{1}{2},\beta}^\pm, u_{\beta,j+\frac{1}{2}}^\pm$ are not the point values of a specific polynomial in the WENO reconstruction. We do not have the “one-dimensional cell averages” $\bar{u}_{i,\beta}^n$ and $\bar{u}_{\beta,j}^n$ either. To this end, we need to construct more polynomials in the (i,j) cell.

Step 3. We construct a single variable Hermite type polynomial $p_1(x)$ in the x -direction first, i.e. the polynomial $p_1(x)$ should satisfy

$$\frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_1(x) dx = \bar{u}_{ij}^n, \quad l = i - 1, i, i + 1;$$

and

$$p_1(x_{i-\frac{1}{2}}) = \bar{u}_{i-\frac{1}{2}j}^+, \quad p_1(x_{i+\frac{1}{2}}) = \bar{u}_{i+\frac{1}{2}j}^-, \quad \text{for fixed } j.$$

Then we construct a single variable Hermite type polynomial $p_2(x)$ in the y -direction, i.e. the polynomial $p_2(x)$ should satisfy

$$\frac{1}{\Delta y} \int_{y_{i-\frac{1}{2}}}^{y_{i+\frac{1}{2}}} p_2(y) dy = \bar{u}_{il}^n, \quad l = j - 1, j, j + 1;$$

and

$$p_2(y_{j-\frac{1}{2}}) = \bar{u}_{i,j-\frac{1}{2}}^+, \quad p_2(y_{j+\frac{1}{2}}) = \bar{u}_{i,j+\frac{1}{2}}^-, \quad \text{for fixed } i.$$

We can get the “one-dimensional cell averages” by

$$\bar{u}_{i,\beta}^n = p_2(y_j^\beta), \quad \bar{u}_{\beta,j}^n = p_1(x_i^\beta). \tag{3.13}$$

The property of quadrature rules implies that $\bar{u}_{i,\beta}^n$ and $\bar{u}_{\beta,j}^n$ above satisfy (3.4) and (3.5).

Step 4. We construct a single variable Hermite type polynomial $p_1^\beta(x)$ in the x -direction along the line $y = y_j^\beta$, i.e. the polynomial $p_1^\beta(x)$ should satisfy

$$\frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_1^\beta(x) dx = \bar{u}_{i,\beta}^n, \quad l = i - 1, i, i + 1;$$

and

$$p_1^\beta(x_{i-\frac{1}{2}}) = u_{i-\frac{1}{2},\beta}^+, \quad p_1^\beta(x_{i+\frac{1}{2}}) = u_{i+\frac{1}{2},\beta}^-, \quad \text{for fixed } \beta \in \{1, \dots, L\}.$$

Likewise, we construct a single variable Hermite type polynomial $p_2^\beta(y)$ in the y -direction along the line $x = x_i^\beta$, i.e. the polynomial $p_2^\beta(y)$ should satisfy

$$\frac{1}{\Delta y} \int_{y_{i-\frac{1}{2}}}^{y_{i+\frac{1}{2}}} p_2^\beta(y) dy = \bar{u}_{\beta,l}^n, \quad l = j - 1, j, j + 1;$$

and

$$p_2^\beta(y_{j-\frac{1}{2}}) = u_{\beta,j-\frac{1}{2}}^+, \quad p_2^\beta(y_{j+\frac{1}{2}}) = u_{\beta,j+\frac{1}{2}}^-, \quad \text{for fixed } \beta \in \{1, \dots, L\}.$$

Now let us consider (3.6), in which $\bar{u}_{i,\beta}^n$ is obtained from (3.13) and $u_{i+\frac{1}{2},\beta}^-, u_{i-\frac{1}{2},\beta}^+$ are obtained from Step 2. Notice that $u_{i+\frac{1}{2},\beta}^-, u_{i-\frac{1}{2},\beta}^+$ are the nodal values and $\bar{u}_{i,\beta}^n$ is the average, respectively, of the single variable polynomial $p_1^\beta(x)$ on the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. Thus, we can use the limiter in Section 2 on $p_1^\beta(x)$ to achieve $H_x^{i,\beta} \in [m, M]$, i.e., use the following $\tilde{p}_1^\beta(x)$ instead of $p_1^\beta(x)$:

$$\tilde{p}_1^\beta(x) = \theta(p_1^\beta(x) - \bar{u}_{i,\beta}^n) + \bar{u}_{i,\beta}^n, \quad \theta = \min \left\{ \left| \frac{M - \bar{u}_{i,\beta}^n}{M' - \bar{u}_{i,\beta}^n} \right|, \left| \frac{m - \bar{u}_{i,\beta}^n}{m' - \bar{u}_{i,\beta}^n} \right|, 1 \right\}, \tag{3.14}$$

$$M' = \max_{x \in S_i^x} p_1^\beta(x), \quad m' = \min_{x \in S_i^x} p_1^\beta(x). \tag{3.15}$$

Similarly, we can define $\tilde{p}_2^\beta(y)$ by:

$$\tilde{p}_2^\beta(y) = \theta(p_2^\beta(y) - \bar{u}_{\beta,j}^n) + \bar{u}_{\beta,j}^n, \quad \theta = \min \left\{ \left| \frac{M - \bar{u}_{\beta,j}^n}{M' - \bar{u}_{\beta,j}^n} \right|, \left| \frac{m - \bar{u}_{\beta,j}^n}{m' - \bar{u}_{\beta,j}^n} \right|, 1 \right\}. \tag{3.16}$$

$$M' = \max_{y \in S_j^y} p_2^\beta(y), \quad m' = \min_{y \in S_j^y} p_2^\beta(y). \tag{3.17}$$

The algorithm for the two-dimensional finite volume WENO scheme with our limiter is summarized as follows:

- Perform Step 1 to Step 4.
- Get $\tilde{p}_1^\beta(x)$ from (3.14) and (3.15).
- Get $\tilde{p}_2^\beta(y)$ from (3.16) and (3.17).
- Compute the revised nodal values

$$\begin{aligned} \tilde{u}_{i+\frac{1}{2},\beta}^- &= \tilde{p}_1^\beta(x_{i+\frac{1}{2}}), & \tilde{u}_{i-\frac{1}{2},\beta}^+ &= \tilde{p}_1^\beta(x_{i-\frac{1}{2}}), \\ \tilde{u}_{\beta,j+\frac{1}{2}}^- &= \tilde{p}_2^\beta(y_{j+\frac{1}{2}}), & \tilde{u}_{\beta,j-\frac{1}{2}}^+ &= \tilde{p}_2^\beta(y_{j-\frac{1}{2}}). \end{aligned}$$

- Plugging them in (3.3), we get the final scheme

$$\bar{u}_{ij}^{n+1} = \bar{u}_{ij}^n - \lambda_1 \sum_{\beta=1}^L w_\beta \left[h_1(\tilde{u}_{i+\frac{1}{2},\beta}^-, \tilde{u}_{i-\frac{1}{2},\beta}^+) - h_1(\tilde{u}_{i-\frac{1}{2},\beta}^-, \tilde{u}_{i+\frac{1}{2},\beta}^+) \right] - \lambda_2 \sum_{\beta=1}^L w_\beta \left[h_2(\tilde{u}_{\beta,j+\frac{1}{2}}^-, \tilde{u}_{\beta,j-\frac{1}{2}}^+) - h_2(\tilde{u}_{\beta,j-\frac{1}{2}}^-, \tilde{u}_{\beta,j+\frac{1}{2}}^+) \right]$$

with the CFL condition

$$a_1 \lambda_1 + a_2 \lambda_2 \leq \min_{\alpha=1, \dots, N} \hat{w}_\alpha.$$

Remark 3.1. For ENO and DG schemes, the cost of the limiter is very small since a polynomial is already available in each cell. The cost to implement this limiter for WENO scheme is significantly larger because no polynomials are obtained after the WENO reconstruction, only point values.

4. Application to two-dimensional incompressible flows

4.1. Preliminaries

We are interested in solving the two-dimensional incompressible Euler equations in the vorticity stream-function formulation:

$$\omega_t + (u\omega)_x + (v\omega)_y = 0, \tag{4.1}$$

$$\Delta\psi = \omega, \quad \langle u, v \rangle = \langle -\psi_y, \psi_x \rangle, \tag{4.2}$$

$$\omega(x, y, 0) = \omega_0(x, y), \quad \langle u, v \rangle \cdot \mathbf{n} = \text{given on } \partial\Omega.$$

The definition of $\langle u, v \rangle$ in (4.2) gives us the divergence-free condition $u_x + v_y = 0$, which implies (4.1) is equivalent to the non-conservative form

$$\omega_t + u\omega_x + v\omega_y = 0. \tag{4.3}$$

The exact solution of (4.3) satisfies the maximum principle $\omega(x, y, t) \in [m, M]$, for all (x, y, t) , where $m = \min_{x,y} \omega_0(x, y)$ and $M = \max_{x,y} \omega_0(x, y)$. For discontinuous solutions or solutions containing sharp gradient regions, it is preferable to solve the

conservative form (4.1) rather than the non-conservative form (4.3). However, without the incompressibility condition $u_x + v_y = 0$, the conservative form (4.1) itself does not imply the maximum principle $\omega(x, y, t) \in [m, M]$ for all (x, y, t) . This is the main difficulty to get a maximum-principle-satisfying scheme solving the conservative form (4.1) directly. In [14], Levy and Tadmor proved the second order central scheme for (4.1) satisfies a strict local maximum principle. Here, we will show the $(k + 1)$ th order accurate (for any k) DG scheme in [15] with the limiter in Section 3.2 under suitable CFL condition satisfies the global maximum principle.

In [15], Liu and Shu introduced a high order discontinuous Galerkin method solving (4.1). We will first recall the method in [15] briefly. First, solve (4.2) by a standard Poisson solver for the stream-function ψ using continuous finite elements, then take $u = -\psi_y, v = \psi_x$. Notice that on the boundary of each cell, $\langle u, v \rangle \cdot \mathbf{n} = \langle -\psi_y, \psi_x \rangle \cdot \mathbf{n} = \frac{\partial \psi}{\partial \tau}$, which is the tangential derivative. Thus $\langle u, v \rangle \cdot \mathbf{n}$ is continuous across the cell boundary since ψ is continuous. Therefore, the DG scheme for (4.1) can be defined as follows: start with a triangulation T_h of the domain Ω , consisting of polygons of maximum size h , and the two approximation spaces

$$V_h^k = \{v : v|_K \in P^k(K), \forall K \in T_h\}, \quad W_{0,h}^k = V_h^k \cap C_0(\Omega),$$

where $P^k(k)$ is the set of all polynomials of degree at most k on the cell K . For given $\psi_h \in W_{0,h}^k$, find the $\omega_h \in V_h^k$ such that

$$\int_K \partial_t \omega_h v dx dy - \int_K \omega_h \mathbf{u}_h \cdot \nabla v dx dy + \sum_{e \in \partial K} \int_e \mathbf{u}_h \cdot \mathbf{n} \widehat{\omega}_h v^- ds = 0, \quad \forall v \in V_h^k, \tag{4.4}$$

where

$$\mathbf{u}_h = \langle u_h, v_h \rangle = \left\langle -\frac{\partial \psi_h}{\partial y}, \frac{\partial \psi_h}{\partial x} \right\rangle.$$

Since the normal velocity $\mathbf{u}_h \cdot \mathbf{n}$ is continuous across any element boundary e , we can define the Lax–Friedrichs upwind biased flux:

$$\mathbf{u}_h \cdot \mathbf{n} \widehat{\omega}_h = h(\omega_h^-, \omega_h^+, \mathbf{u}_h \cdot \mathbf{n}) = \frac{1}{2} [\mathbf{u}_h \cdot \mathbf{n} (\omega_h^+ + \omega_h^-) - a(\omega_h^+ - \omega_h^-)], \tag{4.5}$$

where a is the maximum of $|\mathbf{u}_h \cdot \mathbf{n}|$ either locally or globally.

For convenience, consider the same rectangular mesh as in Section 3. Assume the stream-function ψ is obtained with Q^k elements, where Q^k refers to the space of tensor products of single variable polynomials of degree k , and the DG method uses P^k elements. At time level n , in (i, j) cell, let $\omega_{i-\frac{1}{2},j}^+(y), \omega_{i+\frac{1}{2},j}^-(y), \omega_{i,j-\frac{1}{2}}^+(x)$, and $\omega_{i,j+\frac{1}{2}}^-(x)$ denote the traces of the DG polynomial $\omega_{ij}(x, y)$ on the left, right, bottom and top edges, respectively. On the left, right, bottom and top edges, $\mathbf{u}_h \cdot \mathbf{n}$ is $u_{i-\frac{1}{2},j}^+(y), u_{i+\frac{1}{2},j}^-(y), v_{i,j+\frac{1}{2}}^+(x)$, and $v_{i,j-\frac{1}{2}}^-(x)$, respectively.

4.2. The main result

The cell average scheme with Euler forward in time of the DG method in [15] is

$$\begin{aligned} \bar{\omega}_{ij}^{n+1} &= \bar{\omega}_{ij}^n - \frac{\Delta t}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left[h(\omega_{i+\frac{1}{2},j}^-(y), \omega_{i+\frac{1}{2},j}^+(y), u_{i+\frac{1}{2},j}(y)) - h(\omega_{i-\frac{1}{2},j}^-(y), \omega_{i-\frac{1}{2},j}^+(y), u_{i-\frac{1}{2},j}(y)) \right] dy \\ &\quad - \frac{\Delta t}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[h(\omega_{i,j+\frac{1}{2}}^-(x), \omega_{i,j+\frac{1}{2}}^+(x), v_{i,j+\frac{1}{2}}(x)) - h(\omega_{i,j-\frac{1}{2}}^-(x), \omega_{i,j-\frac{1}{2}}^+(x), v_{i,j-\frac{1}{2}}(x)) \right] dx. \end{aligned} \tag{4.6}$$

The integrals in (4.6) are assumed to be computed exactly. Since all the integrands are single variable polynomials of degree at most $2k - 1$, the integral is equal to the k -point Gauss quadrature. Substituting the integrals by the k -point Gauss quadrature in (4.6), we obtain the mathematically equivalent expression

$$\begin{aligned} \bar{\omega}_{ij}^{n+1} &= \bar{\omega}_{ij}^n - \lambda_1 \sum_{\beta=1}^k w_\beta \left[h(\omega_{i+\frac{1}{2},j}^-, \omega_{i+\frac{1}{2},j}^+, u_{i+\frac{1}{2},j}^\beta) - h(\omega_{i-\frac{1}{2},j}^-, \omega_{i-\frac{1}{2},j}^+, u_{i-\frac{1}{2},j}^\beta) \right] \\ &\quad - \lambda_2 \sum_{\beta=1}^k w_\beta \left[h(\omega_{i,j+\frac{1}{2}}^-, \omega_{i,j+\frac{1}{2}}^+, v_{i,j+\frac{1}{2}}^\beta) - h(\omega_{i,j-\frac{1}{2}}^-, \omega_{i,j-\frac{1}{2}}^+, v_{i,j-\frac{1}{2}}^\beta) \right]. \end{aligned} \tag{4.7}$$

Now let us assume $\bar{\omega}_{ij}^n \in [m, M]$ and the DG polynomial $\omega_{ij}(x, y)$ is already processed by the limiter in Section 3.2 with $L = k$. In particular, $\omega_{ij}(x, y) \in [m, M], \forall (x, y) \in S_{ij}$ with S_{ij} defined in (3.10). Then we will show the scheme (4.7) satisfies the maximum principle $\bar{\omega}_{ij}^{n+1} \in [m, M]$.

If we use $\bar{\omega}_{\beta j}$ to denote the average of $\omega_{ij}(x_i^\beta, y)$ over $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ and $\bar{\omega}_{i,\beta}$ to denote the average of $\omega_{ij}(x, y_j^\beta)$ over $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, then the cell average $\bar{\omega}_{ij}^n$ is

$$\begin{aligned} \bar{\omega}_{ij}^n &= \frac{1}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \omega_{ij}(x, y) dx dy = \frac{1}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left(\sum_{\beta=1}^k \omega_{ij}(x_i^\beta, y) w_\beta \Delta x \right) dy = \sum_{\beta=1}^k w_\beta \left(\frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \omega_{ij}(x_i^\beta, y) dy \right) \\ &= \sum_{\beta=1}^k w_\beta \bar{\omega}_{\beta j}, \end{aligned} \tag{4.8}$$

and similarly we have

$$\bar{\omega}_{ij}^n = \sum_{\beta=1}^k w_\beta \bar{\omega}_{i, \beta}. \tag{4.9}$$

Plugging (4.8) and (4.9) into (4.7), we get

$$\begin{aligned} \bar{\omega}_{ij}^{n+1} &= \bar{\omega}_{ij}^n - \lambda_1 \sum_{\beta=1}^k w_\beta \left[h\left(\omega_{i+\frac{1}{2}, \beta}^-, \omega_{i+\frac{1}{2}, \beta}^+, u_{i+\frac{1}{2}, \beta}\right) - h\left(\omega_{i-\frac{1}{2}, \beta}^-, \omega_{i-\frac{1}{2}, \beta}^+, u_{i-\frac{1}{2}, \beta}\right) \right] \\ &\quad - \lambda_2 \sum_{\beta=1}^k w_\beta \left[h\left(\omega_{\beta j+\frac{1}{2}}^-, \omega_{\beta j+\frac{1}{2}}^+, v_{\beta j+\frac{1}{2}}\right) - h\left(\omega_{\beta j-\frac{1}{2}}^-, \omega_{\beta j-\frac{1}{2}}^+, v_{\beta j-\frac{1}{2}}\right) \right] \\ &= \frac{a_1 \lambda_1}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^k w_\beta \bar{\omega}_{i, \beta} - \lambda_1 \sum_{\beta=1}^k w_\beta \left[h\left(\omega_{i+\frac{1}{2}, \beta}^-, \omega_{i+\frac{1}{2}, \beta}^+, u_{i+\frac{1}{2}, \beta}\right) - h\left(\omega_{i-\frac{1}{2}, \beta}^-, \omega_{i-\frac{1}{2}, \beta}^+, u_{i-\frac{1}{2}, \beta}\right) \right] \\ &\quad + \frac{a_2 \lambda_2}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^k w_\beta \bar{\omega}_{\beta j} - \lambda_2 \sum_{\beta=1}^k w_\beta \left[h\left(\omega_{\beta j+\frac{1}{2}}^-, \omega_{\beta j+\frac{1}{2}}^+, v_{\beta j+\frac{1}{2}}\right) - h\left(\omega_{\beta j-\frac{1}{2}}^-, \omega_{\beta j-\frac{1}{2}}^+, v_{\beta j-\frac{1}{2}}\right) \right] \\ &= \frac{a_1 \lambda_1}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^k w_\beta \left[\bar{\omega}_{i, \beta} - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} \left(h\left(\omega_{i+\frac{1}{2}, \beta}^-, \omega_{i+\frac{1}{2}, \beta}^+, u_{i+\frac{1}{2}, \beta}\right) - h\left(\omega_{i-\frac{1}{2}, \beta}^-, \omega_{i-\frac{1}{2}, \beta}^+, u_{i-\frac{1}{2}, \beta}\right) \right) \right] \\ &\quad + \frac{a_2 \lambda_2}{a_1 \lambda_1 + a_2 \lambda_2} \sum_{\beta=1}^k w_\beta \left[\bar{\omega}_{\beta j} - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_2} \left(h\left(\omega_{\beta j+\frac{1}{2}}^-, \omega_{\beta j+\frac{1}{2}}^+, v_{\beta j+\frac{1}{2}}\right) - h\left(\omega_{\beta j-\frac{1}{2}}^-, \omega_{\beta j-\frac{1}{2}}^+, v_{\beta j-\frac{1}{2}}\right) \right) \right]. \end{aligned} \tag{4.10}$$

For later discussion, we define the following formal one-dimensional schemes:

$$H_x^{i, \beta} = \bar{\omega}_{i, \beta} - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} \left[h\left(\omega_{i+\frac{1}{2}, \beta}^-, \omega_{i+\frac{1}{2}, \beta}^+, u_{i+\frac{1}{2}, \beta}\right) - h\left(\omega_{i-\frac{1}{2}, \beta}^-, \omega_{i-\frac{1}{2}, \beta}^+, u_{i-\frac{1}{2}, \beta}\right) \right], \tag{4.11}$$

$$H_y^{\beta, j} = \bar{\omega}_{\beta j} - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_2} \left[h\left(\omega_{\beta j+\frac{1}{2}}^-, \omega_{\beta j+\frac{1}{2}}^+, v_{\beta j+\frac{1}{2}}\right) - h\left(\omega_{\beta j-\frac{1}{2}}^-, \omega_{\beta j-\frac{1}{2}}^+, v_{\beta j-\frac{1}{2}}\right) \right]. \tag{4.12}$$

To show our main result, we need two lemmas first.

Lemma 4.1. Consider a “one-dimensional first order Lax–Friedrichs” scheme

$$\omega_j^{n+1} = \omega_j - \lambda [h(\omega_j, \omega_{j+1}, u_j) - h(\omega_{j-1}, \omega_j, u_{j-1})], \tag{4.13}$$

where $h(\cdot, \cdot, \cdot)$ is defined in (4.5) and $u_{j-1}, u_j \in [-a, a]$. Assume $\omega_{j-1}, \omega_j, \omega_{j+1} \in [m, M]$, then

$$m - \lambda [h(m, m, u_j) - h(m, m, u_{j-1})] \leq \omega_j^{n+1} \leq M - \lambda [h(M, M, u_j) - h(M, M, u_{j-1})]$$

under the CFL condition

$$a\lambda \leq \frac{1}{2}.$$

Proof. We can rewrite (4.13) as

$$\begin{aligned} \omega_j^{n+1} &= \omega_j - \lambda \left[\frac{1}{2} (u_j (\omega_j + \omega_{j+1}) - a (\omega_{j+1} - \omega_j)) - \frac{1}{2} (u_{j-1} (\omega_{j-1} + \omega_j) - a (\omega_j - \omega_{j-1})) \right] \\ &= \frac{\lambda}{2} (a + u_{j-1}) \omega_{j-1} + \left[1 - \lambda \left(\frac{1}{2} (u_j - u_{j-1}) + a \right) \right] \omega_j + \frac{\lambda}{2} (a - u_j) \omega_{j+1}. \end{aligned}$$

All the three coefficients are nonnegative under the CFL condition. Therefore,

$$\begin{aligned} \omega_j^{n+1} &= \frac{\lambda}{2} (a + u_{j-1}) \omega_{j-1} + \left[1 - \lambda \left(\frac{1}{2} (u_j - u_{j-1}) + a \right) \right] \omega_j + \frac{\lambda}{2} (a - u_j) \omega_{j+1} \\ &\leq \frac{\lambda}{2} (a + u_{j-1}) M + \left[1 - \lambda \left(\frac{1}{2} (u_j - u_{j-1}) + a \right) \right] M + \frac{\lambda}{2} (a - u_j) M = M - \lambda [h(M, M, u_j) - h(M, M, u_{j-1})]. \end{aligned}$$

Similarly, we have $m - \lambda [h(m, m, u_j) - h(m, m, u_{j-1})] \leq \omega_j^{n+1}$. \square

Lemma 4.2. Under the CFL condition $a_1 \lambda_1 + a_2 \lambda_2 \leq \frac{1}{2} \min_{\alpha=1, \dots, N} \hat{w}_\alpha$, we have

$$\left[1 - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} (u_{i+\frac{1}{2}, \beta} - u_{i-\frac{1}{2}, \beta}) \right] m \leq H_x^{i, \beta} \leq \left[1 - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} (u_{i+\frac{1}{2}, \beta} - u_{i-\frac{1}{2}, \beta}) \right] M, \tag{4.14}$$

$$\left[1 - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_2} (v_{\beta, j+\frac{1}{2}} - v_{\beta, j-\frac{1}{2}}) \right] m \leq H_y^{\beta, j} \leq \left[1 - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_2} (v_{\beta, j+\frac{1}{2}} - v_{\beta, j-\frac{1}{2}}) \right] M. \tag{4.15}$$

Proof. We only prove (4.14) here. The proof for (4.15) is similar. We follow the main idea of the proof of Theorem 2.2. Considering the “one-dimensional scheme” (4.11), we would like to rewrite it as a convex combination of several “one-dimensional first order schemes” of the type (4.13). Recall that $\hat{S}_i^x = \{x_{i-\frac{1}{2}} = \hat{x}_i^1, \hat{x}_i^2, \dots, \hat{x}_i^N = x_{i+\frac{1}{2}}\}$ are the Gauss–Lobatto quadrature points on the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. Also \hat{w}_α denotes the weights for the interval $[-\frac{1}{2}, \frac{1}{2}]$ so that $\sum_{\alpha=1}^N \hat{w}_\alpha = 1$. Let $\hat{v}_\alpha^\beta = \omega_{ij}(x_i^\alpha, y_j^\beta)$ for $\alpha = 1, \dots, N$, $\hat{v}_0^\beta = \omega_{i-1, j}(x_{i-\frac{1}{2}}, y_j^\beta)$ and $\hat{v}_{N+1}^\beta = \omega_{i+1, j}(x_{i+\frac{1}{2}}, y_j^\beta)$. Since we assume that all the DG polynomials $\omega_{ij}(x, y)$ are pre-processed by the limiter, we have $\hat{v}_\alpha^\beta \in [m, M]$ for $\alpha = 0, \dots, N + 1$ and $\beta = 1, \dots, k$. By the property of the quadrature rule, we get $\bar{\omega}_{i, \beta} = \sum_{\alpha=1}^N \hat{w}_\alpha \hat{v}_\alpha^\beta$. We also have

$$\begin{aligned} & h(\hat{v}_N^\beta, \hat{v}_{N+1}^\beta, u_{i+\frac{1}{2}, \beta}) - h(\hat{v}_0^\beta, \hat{v}_1^\beta, u_{i-\frac{1}{2}, \beta}) \\ &= h(\hat{v}_N^\beta, \hat{v}_{N+1}^\beta, u_{i+\frac{1}{2}, \beta}) - h(\hat{v}_{N-1}^\beta, \hat{v}_N^\beta, 0) + \sum_{\alpha=2}^{N-1} [h(\hat{v}_\alpha^\beta, \hat{v}_{\alpha+1}^\beta, 0) - h(\hat{v}_{\alpha-1}^\beta, \hat{v}_\alpha^\beta, 0)] + h(\hat{v}_1^\beta, \hat{v}_2^\beta, 0) - h(\hat{v}_0^\beta, \hat{v}_1^\beta, u_{i-\frac{1}{2}, \beta}). \end{aligned}$$

For convenience, let us use λ to denote $\frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1}$ here, then (4.11) becomes:

$$\begin{aligned} H_x^{i, \beta} &= \bar{\omega}_{i, \beta} - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} \left[h(\omega_{i+\frac{1}{2}, \beta}^-, \omega_{i+\frac{1}{2}, \beta}^+, u_{i+\frac{1}{2}, \beta}) - h(\omega_{i-\frac{1}{2}, \beta}^-, \omega_{i-\frac{1}{2}, \beta}^+, u_{i-\frac{1}{2}, \beta}) \right] \\ &= \bar{\omega}_{i, \beta} - \lambda \left[h(\hat{v}_N^\beta, \hat{v}_{N+1}^\beta, u_{i+\frac{1}{2}, \beta}) - h(\hat{v}_0^\beta, \hat{v}_1^\beta, u_{i-\frac{1}{2}, \beta}) \right] \\ &= \sum_{\alpha=1}^N \hat{w}_\alpha \hat{v}_\alpha^\beta - \lambda \left[h(\hat{v}_N^\beta, \hat{v}_{N+1}^\beta, u_{i+\frac{1}{2}, \beta}) - h(\hat{v}_{N-1}^\beta, \hat{v}_N^\beta, 0) + \sum_{\alpha=2}^{N-1} [h(\hat{v}_\alpha^\beta, \hat{v}_{\alpha+1}^\beta, 0) \right. \\ &\quad \left. - h(\hat{v}_{\alpha-1}^\beta, \hat{v}_\alpha^\beta, 0)] + h(\hat{v}_1^\beta, \hat{v}_2^\beta, 0) - h(\hat{v}_0^\beta, \hat{v}_1^\beta, u_{i-\frac{1}{2}, \beta}) \right] \\ &= \hat{w}_N \left[v_N^\beta - \frac{\lambda}{\hat{w}_N} (h(\hat{v}_N^\beta, \hat{v}_{N+1}^\beta, u_{i+\frac{1}{2}, \beta}) - h(\hat{v}_{N-1}^\beta, \hat{v}_N^\beta, 0)) \right] + \sum_{\alpha=2}^{N-1} \hat{w}_\alpha \left[v_\alpha^\beta - \frac{\lambda}{\hat{w}_\alpha} (h(\hat{v}_\alpha^\beta, \hat{v}_{\alpha+1}^\beta, 0) - h(\hat{v}_{\alpha-1}^\beta, \hat{v}_\alpha^\beta, 0)) \right] \\ &\quad + \hat{w}_1 \left[v_1^\beta - \frac{\lambda}{\hat{w}_1} (h(\hat{v}_1^\beta, \hat{v}_2^\beta, 0) - h(\hat{v}_0^\beta, \hat{v}_1^\beta, u_{i-\frac{1}{2}, \beta})) \right]. \end{aligned}$$

By Lemma 4.1, we can get the upper bound

$$\begin{aligned} H_x^{i, \beta} &\leq \hat{w}_N \left[M - \frac{\lambda}{\hat{w}_N} (h(M, M, u_{i+\frac{1}{2}, \beta}) - h(M, M, 0)) \right] + \sum_{\alpha=2}^{N-1} \hat{w}_\alpha \left[M - \frac{\lambda}{\hat{w}_\alpha} (h(M, M, 0) - h(M, M, 0)) \right] \\ &\quad + \hat{w}_1 \left[M - \frac{\lambda}{\hat{w}_1} (h(M, M, 0) - h(M, M, u_{i-\frac{1}{2}, \beta})) \right] \\ &= \sum_{\alpha=1}^N \hat{w}_\alpha M - \lambda \left[h(M, M, u_{i+\frac{1}{2}, \beta}) - h(M, M, 0) + h(M, M, 0) - h(M, M, u_{i-\frac{1}{2}, \beta}) \right] \\ &= M - \lambda [h(M, M, u_{i+\frac{1}{2}, \beta}) - h(M, M, u_{i-\frac{1}{2}, \beta})] = M - \lambda [Mu_{i+\frac{1}{2}, \beta} - Mu_{i-\frac{1}{2}, \beta}] = \left[1 - \frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1} (u_{i+\frac{1}{2}, \beta} - u_{i-\frac{1}{2}, \beta}) \right] M. \end{aligned}$$

Similarly, we can prove the lower bound. \square

Theorem 4.3. Suppose $\bar{\omega}_{ij}^n \in [m, M]$ and the DG polynomial $\omega_{ij}(x, y)$ is already processed by the limiter in Section 3.2 with $L = k$. Thus $\omega_{ij}(x, y) \in [m, M]$ for all $(x, y) \in S_{ij}$ with S_{ij} defined in (3.10). Then the scheme (4.7) satisfies the maximum principle $\bar{\omega}_{ij}^{n+1} \in [m, M]$ under the CFL condition $a_1 \lambda_1 + a_2 \lambda_2 \leq \frac{1}{2} \min_{\alpha=1, \dots, N} \hat{w}_\alpha$.

Proof. For convenience, we use λ to denote $\frac{a_1 \lambda_1 + a_2 \lambda_2}{a_1}$ and μ to denote $\frac{a_1 \lambda_1 + a_2 \lambda_2}{a_2}$ here. Starting with (4.10), (4.11) and (4.12), we have

$$\begin{aligned} \bar{\omega}_{ij}^{n+1} &= \frac{\lambda_1}{\lambda} \sum_{\beta=1}^k w_{\beta} H_x^{i,\beta} + \frac{\lambda_2}{\mu} \sum_{\beta=1}^k w_{\beta} H_y^{\beta,j} \\ \text{(Lemma 4.2)} &\leq \frac{\lambda_1}{\lambda} \sum_{\beta=1}^k w_{\beta} \left[1 - \lambda \left(u_{i+\frac{1}{2},\beta} - u_{i-\frac{1}{2},\beta} \right) \right] M + \frac{\lambda_2}{\mu} \sum_{\beta=1}^k w_{\beta} \left[1 - \mu \left(v_{\beta,j+\frac{1}{2}} - v_{\beta,j-\frac{1}{2}} \right) \right] M \\ \left(\frac{\lambda_1}{\lambda} + \frac{\lambda_2}{\mu} = 1 \right) &= M - M \sum_{\beta=1}^k w_{\beta} \lambda_1 \left(u_{i+\frac{1}{2},\beta} - u_{i-\frac{1}{2},\beta} \right) - M \sum_{\beta=1}^k w_{\beta} \lambda_2 \left(v_{\beta,j+\frac{1}{2}} - v_{\beta,j-\frac{1}{2}} \right) \\ &= M \left[1 - \frac{\Delta t}{\Delta x \Delta y} \left(\sum_{\beta=1}^k \left(u_{i+\frac{1}{2},\beta} - u_{i-\frac{1}{2},\beta} \right) w_{\beta} \Delta y + \sum_{\beta=1}^k \left(v_{\beta,j+\frac{1}{2}} - v_{\beta,j-\frac{1}{2}} \right) w_{\beta} \Delta x \right) \right] \\ \text{(quadrature)} &= M \left[1 - \frac{\Delta t}{\Delta x \Delta y} \left(\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u \left(x_{i+\frac{1}{2}}, y, t \right) - u \left(x_{i-\frac{1}{2}}, y, t \right) dy + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v \left(x, y_{j+\frac{1}{2}}, t \right) - v \left(x, y_{j-\frac{1}{2}}, t \right) dx \right) \right] \\ &= M \left[1 - \frac{\Delta t}{\Delta x \Delta y} \left(\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_x dx dy + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} v_y dy dx \right) \right] \\ &= M \left[1 - \frac{\Delta t}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} (u_x + v_y) dy dx \right] \\ (u_x + v_y = 0) &= M. \end{aligned}$$

Similarly, we can prove $m \leq \bar{\omega}_{ij}^{n+1}$. \square

Remark 4.4. We can prove the same results for the upwind flux defined in [15] following the same lines.

Remark 4.5. The results of Theorem 4.3 holds also for any passive convection linear equations with divergence-free velocity coefficients, namely Eq. (4.1) in which u and v are given functions satisfying $u_x + v_y = 0$, as long as the quadratures are exact for the integrands in the scheme. This can be easily achieved if we pre-process the divergence-free velocity field so that it is piecewise polynomial of the right degree for accuracy, continuous in the normal component across cell boundaries, and divergence-free.

5. Numerical tests for the fifth order finite volume WENO scheme

In this section we provide numerical examples for the finite volume WENO scheme. The scheme for the numerical test is third order SSP Runge–Kutta in time and fifth order spatial finite volume WENO approximation with the limiter, unless otherwise specified. We use the global Lax–Friedrichs flux, although the results hold also for any other monotone flux. The time step is taken as that indicated by the proof of maximum principle in the earlier part of the paper, unless otherwise stated.

5.1. Standard one-dimensional test cases

Example 5.1. We solve the model equation

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

with periodic boundary conditions.

Three smooth initial data $u_0(x)$ on $[0, 1]$ are used to show the accuracy. We list the L^1 and L^∞ errors for the cell averages at time $t = 0.1$ in Table 5.1. In this example we take $\Delta t = O(\Delta x^5)$ for the purpose of showing fifth order accuracy.

For Runge–Kutta, the accuracy degenerates slightly. This is due to the lower order accuracy in the intermediate stages of the Runge–Kutta method. In particular, recall that the limiter (1.4) does not destroy accuracy only if the polynomial $p(x)$ is a $(k + 1)$ th accurate approximation to the exact solution. The reconstruction polynomials $p(x)$ in the intermediate stages of a Runge–Kutta method are in general not $(k + 1)$ th order accurate, therefore the limiter (1.4) may kill the accuracy when it is imposed in the intermediate stages. A similar phenomenon of the Runge–Kutta method in the context of boundary conditions was pointed out in [1]. We emphasize that this accuracy degeneracy usually can only be observed on a very fine mesh.

Table 5.1

Third order SSP time discretization and fifth order finite volume WENO scheme with the limiter, for the linear equation with initial data $u_0(x)$. $\Delta x = \frac{1}{N}$ and $\Delta t = \frac{1}{2} \Delta x^{\frac{5}{3}}$.

N	SSP Runge–Kutta				SSP multi-step			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
$u_0(x) = 0.5 + \sin(2\pi x)$								
20	1.58E–4	–	3.03E–4	–	1.53E–4	–	2.87E–4	–
40	4.65E–6	5.09	1.00E–5	4.91	4.69E–6	5.02	1.03E–5	4.79
80	1.39E–7	5.05	2.79E–7	5.17	1.43E–7	5.03	3.13E–7	5.04
160	4.36E–9	5.00	8.30E–9	5.07	4.40E–9	5.02	9.28E–9	5.08
320	1.42E–10	4.94	7.81E–10	3.41	1.37E–10	5.00	2.52E–10	5.20
640	5.12E–12	4.78	6.82E–11	3.51	4.40E–12	4.96	7.73E–12	5.03
1280	1.75E–13	4.87	5.95E–12	3.52	1.29E–13	5.09	2.21E–13	5.13
$u_0(x) = 0.5 + \sin^4(2\pi x)$								
20	1.07E–2	–	2.60E–2	–	1.22E–2	–	2.37E–2	–
40	1.70E–3	2.66	3.46E–3	2.90	1.85E–3	2.73	3.77E–3	2.65
80	1.01E–4	4.06	3.44E–4	3.33	1.24E–4	3.89	4.90E–4	2.94
160	2.87E–6	5.14	1.59E–5	4.42	2.96E–6	5.39	1.58E–5	4.94
320	7.71E–8	5.21	3.60E–7	5.47	7.71E–8	5.26	3.59E–7	5.46
640	1.81E–9	5.40	5.75E–9	5.97	1.81E–9	5.40	5.75E–9	5.97
1280	4.16E–11	5.44	9.36E–11	5.94	4.17E–11	5.45	9.36E–11	5.94
$u^0(x) = 0.5 + \sin^8(2\pi x)$								
20	3.09E–2	–	9.08E–2	–	2.99E–2	–	8.41E–2	–
40	3.50E–3	3.13	1.26E–2	2.85	3.49E–3	3.10	1.24E–2	2.76
80	2.43E–4	3.84	7.44E–4	4.08	2.46E–4	3.82	7.33E–4	4.07
160	8.52E–6	4.83	3.33E–5	4.48	8.51E–6	4.85	3.34E–5	4.45
320	2.83E–7	4.91	1.12E–6	4.88	2.83E–7	4.90	1.13E–6	4.88
640	8.32E–9	5.09	3.35E–8	5.07	8.32E–9	5.09	3.36E–8	5.07
1280	2.28E–10	5.18	9.41E–10	5.15	2.28E–10	5.18	9.39E–10	5.16
2560	6.45E–12	5.14	3.15E–11	4.89	6.44E–12	5.14	3.09E–11	4.92

Table 5.2

Third order SSP time discretization and fifth order finite volume WENO scheme with the limiter, for the Burgers' equation. $t = 0.15$, $\Delta x = \frac{2}{N}$ and $\Delta t = \frac{1}{2} \Delta x^{\frac{5}{3}}$.

N	SSP Runge–Kutta				SSP multi-step			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
20	1.58E–4	–	3.12E–4	–	1.34E–4	–	2.55E–4	–
40	6.79E–6	4.54	1.27E–5	4.61	6.69E–6	4.32	1.24E–5	4.35
80	2.66E–7	4.67	7.19E–7	4.14	2.62E–7	4.67	7.09E–7	4.13
160	9.45E–9	4.81	5.34E–8	3.75	9.28E–9	4.82	4.95E–8	3.84
320	2.31E–10	5.35	1.24E–9	5.42	2.30E–10	5.32	7.84E–10	5.98
640	5.86E–12	5.30	4.76E–11	4.70	5.56E–12	5.37	1.23E–11	5.99
1280	1.67E–13	5.13	3.75E–12	3.67	1.41E–13	5.29	3.72E–13	5.04

To justify that the limiter itself does not kill accuracy, we also perform accuracy tests for the scheme using the third order SSP multi-step time discretization and the same fifth order finite volume WENO approximation with the limiter. The full accuracy order is observed for the multi-step time discretization. Since accuracy degeneracy is usually only observed on very fine meshes for Runge–Kutta methods, in applications it is often acceptable to use the Runge–Kutta methods, similar to the conclusions in [1].

Example 5.2. We solve the Burgers' equation with periodic boundary conditions

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad -1 \leq x \leq 1,$$

$$u(x, 0) = u_0(x).$$

For the initial data $u_0(x) = 0.25 + 0.5 \sin(\pi x)$, the exact solution is smooth up to $t = \frac{2}{\pi}$, then it develops a moving shock which interacts with a rarefaction wave. In this example we take $\Delta t = O(\Delta x^{\frac{5}{3}})$ for the purpose of showing fifth order accuracy.

At $t = 0.15$ the solution is still smooth. We list the errors in Table 5.2. We can clearly see the designed fifth order accuracy is achieved for the multi-step time discretization, while the accuracy for the Runge–Kutta time discretization is less clean. At $t = \frac{2}{\pi}$ the shock just begins to form; at $t = 2.0$ the interaction between the shock and the rarefaction waves is over, and the solution becomes monotone between the shocks. In Fig. 5.1 we can see that the shock is captured very well.

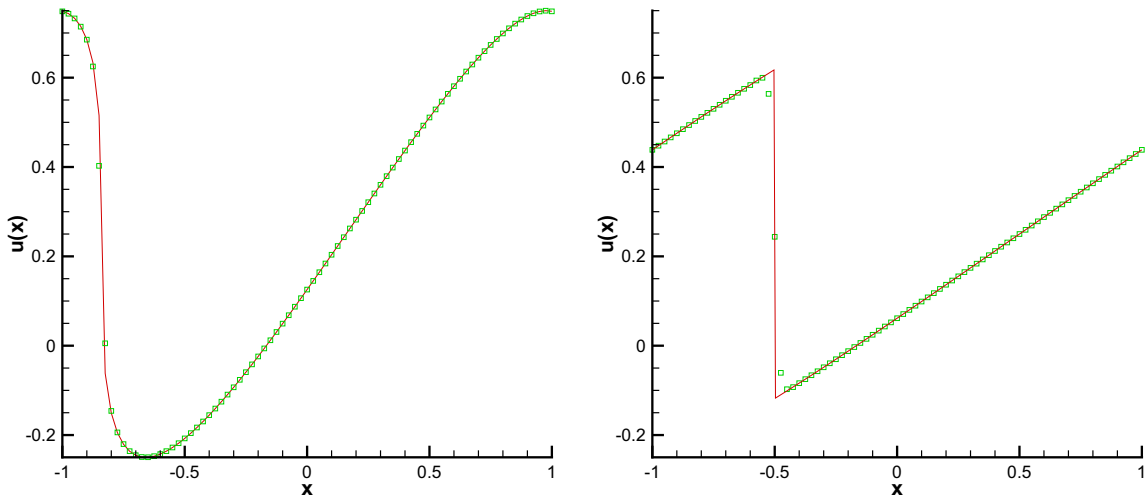


Fig. 5.1. Example 5.2. $N = 80$, $\Delta t = \frac{1}{2} \Delta x^{\frac{5}{3}}$. Solid line: exact solution; symbols: numerical solution (cell averages). Left: $t = 0.6366$; right: $t = 2$.

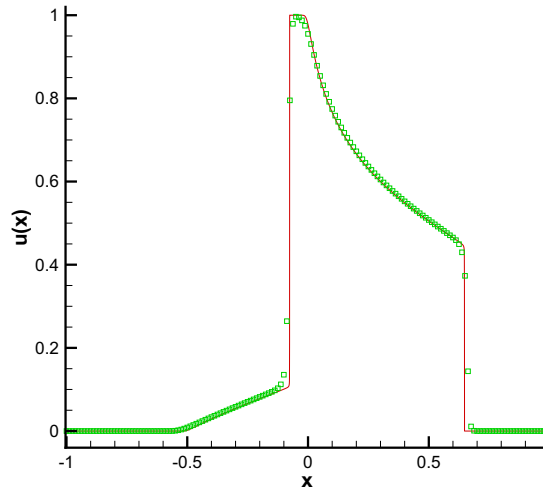


Fig. 5.2. Example 5.3. $t = 0.4$, $N = 160$, $\Delta x = \frac{2}{N}$ and $\frac{\Delta t}{\Delta x} a = \frac{1}{12}$. Solid line: exact solution; symbols: numerical solution (cell averages).

Table 5.3

Third order SSP time discretization and fifth order finite volume WENO scheme with the limiter, for the two-dimensional linear equation with initial data $\sin(2\pi(x + y))$. $t = 0.1$, $\Delta t = \frac{1}{2} \Delta x^{\frac{5}{3}}$ and $\Delta x = \Delta y = \frac{1}{N}$.

$N \times N$	SSP Runge–Kutta				SSP multi-step			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
10×10	7.68E–3	–	1.13E–2	–	6.32E–3	–	9.80E–3	–
20×20	2.88E–4	4.73	5.92E–4	4.26	2.94E–4	4.42	6.00E–4	4.03
40×40	9.03E–6	4.99	1.69E–5	5.12	9.51E–6	4.95	1.97E–5	4.92
80×80	2.86E–7	4.98	8.91E–7	4.25	2.96E–7	5.00	6.17E–7	5.00
160×160	1.05E–8	4.75	8.59E–8	3.37	9.26E–9	5.00	1.90E–8	5.02
320×320	4.44E–10	4.56	7.67E–9	3.49	2.89E–10	4.99	5.28E–10	5.17

Example 5.3. We use the nonconvex Buckley–Leverett flux

$$f(u) = \frac{4u^2}{4u^2 + (1 - u)^2}$$

Table 5.4

Third order SSP time discretization and fifth order finite volume WENO scheme with the limiter, for the two-dimensional Burgers' equation with the initial data $0.5 + \sin(\pi(x + y))$. $t = 0.05$, $\Delta t = \frac{1}{2}\Delta x^3$ and $\Delta x = \Delta y = \frac{2}{N}$.

$N \times N$	SSP Runge–Kutta				SSP multi-step			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
40×40	1.57E–5	–	1.06E–4	–	1.52E–5	–	6.97E–5	–
80×80	7.81E–7	4.33	9.94E–6	3.42	6.15E–7	4.63	4.87E–6	3.84
160×160	3.73E–8	4.38	9.50E–7	3.39	2.52E–8	4.61	3.66E–7	3.74
320×320	1.36E–9	4.77	7.12E–8	3.73	7.59E–10	5.05	1.26E–8	4.89
640×640	5.11E–11	4.73	5.64E–9	3.66	1.90E–11	5.31	1.55E–10	6.34

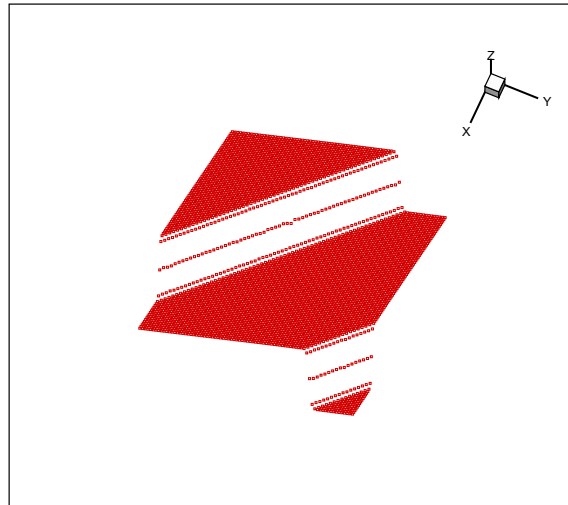


Fig. 5.3. Example 5.5. The numerical solution of the two-dimensional Burgers' equation. $t = 0.6$. $\Delta t = \frac{1}{2}\Delta x^3$ and $\Delta x = \Delta y = \frac{2}{80}$.

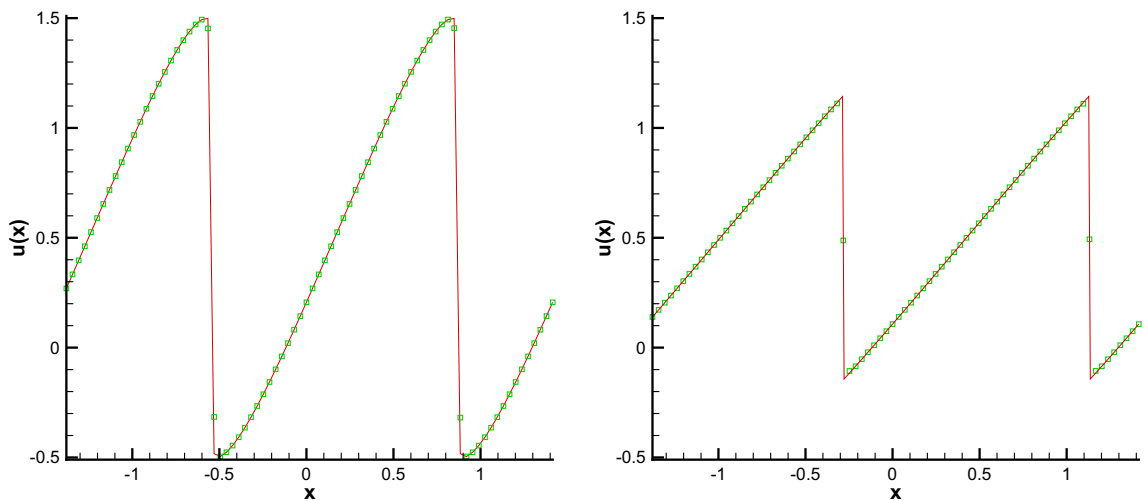


Fig. 5.4. Example 5.5. Cut along the diagonal. Solid line: the exact solution; symbols: numerical solution. Left: $t = 0.23$, right: $t = 0.6$. $\Delta x = \Delta y = \frac{2}{80}$.

to test convergence to the physically correct entropy solutions. The initial condition is taken as $u = 1$ in $[-\frac{1}{2}, 0]$ and $u = 0$ elsewhere. The “exact” solution is obtained from the first order Lax–Friedrichs scheme on a very fine mesh. The computational result is displayed in Fig. 5.2, which is quite satisfactory.

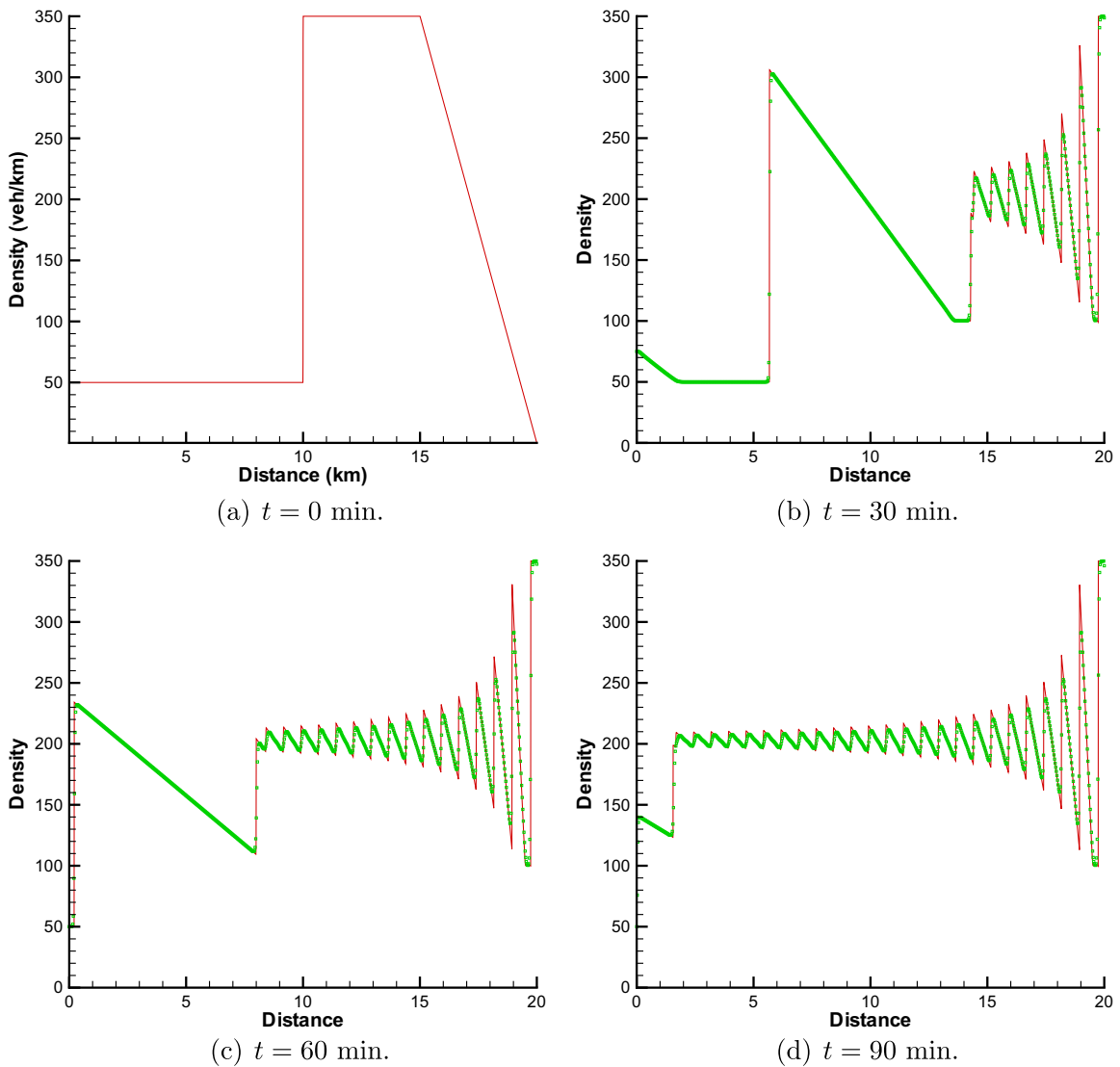


Fig. 5.5. Example 5.6 (traffic flow problem). $N = 800$, $\Delta x = \frac{20}{N}$ and $\frac{\Delta t}{\Delta x} a = \frac{1}{12}$ where $a = 100$. Solid line: exact solution; circles: numerical solution (cell averages).

5.2. Standard two-dimensional test cases

Example 5.4. We solve the linear equation $u_t + u_x + u_y = 0$ on $[0, 1] \times [0, 1]$ with periodic boundary condition. The initial condition is $u_0(x + y) = \sin(2\pi(x + y))$. In this example we take $\Delta t = O(\Delta x^5)$ for the purpose of showing fifth order accuracy. The results are similar to the one-dimensional case, see Table 5.3.

Example 5.5. We solve the Burgers' equation $u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0$ on $[-1, 1] \times [-1, 1]$ with periodic boundary condition. The initial condition is $0.5 + \sin(\pi(x + y))$. In this example we take $\Delta t = O(\Delta x^5)$ for the purpose of showing fifth order accuracy. The results are again similar to the one-dimensional case, see Table 5.4 and Figs. 5.3 and 5.4.

5.3. Test cases from traffic flow models

In the subsection, we test our fifth order finite volume WENO scheme with the limiter on two traffic flow problems. To describe the dynamic characteristics of traffic on a homogeneous and unidirectional highway, the Lighthill–Whitham–Richards (LWR) model is widely used. The governing equation for the LWR model is a scalar conservation law

$$\rho_t + q(\rho)_x = 0$$

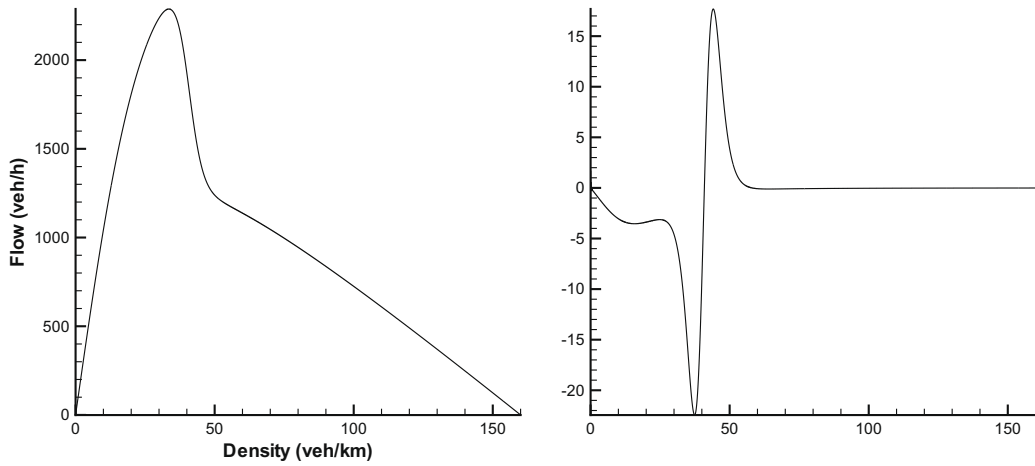


Fig. 5.6. Left: the graph of $q(\rho)$; right: the graph of $q'(\rho)$.

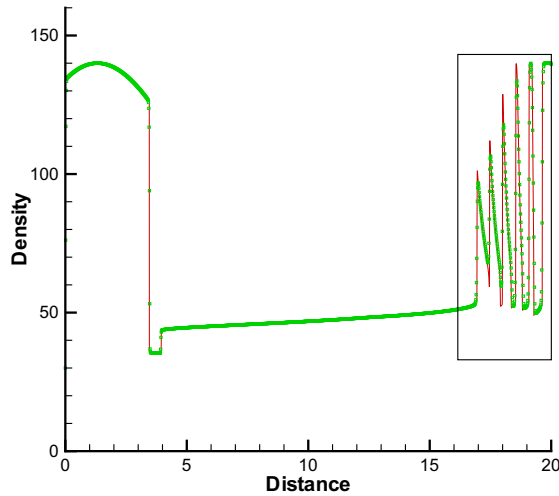


Fig. 5.7. Example 5.7 (traffic flow problem), $t = 18$ min, $N = 1600$, $\Delta x = \frac{20}{N}$ and $\frac{\Delta t}{\Delta x} a = \frac{1}{12}$ where $a \approx 100$. Solid line: exact solution; circles: numerical solution (cell averages). The region inside the rectangle on the right is magnified in Fig. 5.8.

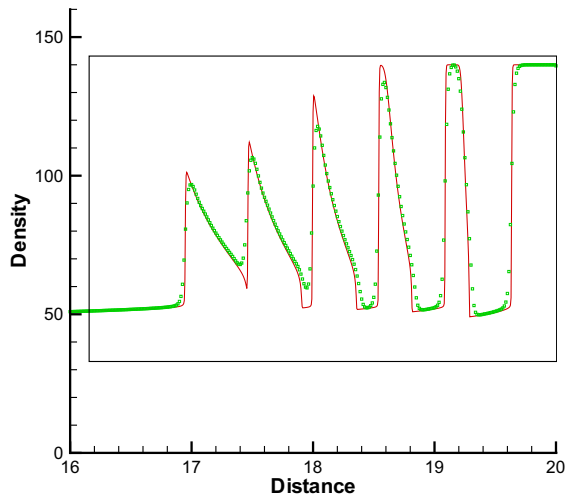


Fig. 5.8. Magnified graph of the region inside the rectangle in Fig. 5.7.

Table 6.1

Third order SSP time discretization and the DG scheme with the limiter, for the linear equation with initial data $\sin(2\pi x)$, $\Delta x = \frac{1}{N}$, $t = 0.1$.

N	SSP Runge–Kutta				SSP multi-step				
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order	
P2	$\Delta t = \frac{1}{24} \Delta x$				$\Delta t = \frac{1}{24} \Delta x$				
	20	4.71E–5	–	7.31E–5	–	4.70E–5	–	7.28E–5	–
	40	2.95E–6	3.99	5.77E–6	3.66	2.97E–6	3.98	4.65E–6	3.97
	80	1.95E–7	3.92	9.48E–7	2.60	1.87E–7	3.98	2.93E–7	3.98
	160	1.86E–8	3.39	2.76E–7	1.78	1.18E–8	3.98	1.85E–8	3.98
	320	2.05E–9	3.18	8.97E–8	1.62	7.55E–10	3.97	1.18E–9	3.97
	640	2.69E–10	2.93	2.84E–8	1.66	4.92E–11	3.93	7.74E–11	3.94
	1280	3.93E–11	2.78	8.59E–9	1.72	3.35E–12	3.87	5.35E–12	3.85
2560	5.65E–12	2.80	2.49E–9	1.78	3.16E–13	3.40	6.73E–13	3.10	
P3	$\Delta t = \frac{1}{15} \Delta x^{\frac{2}{3}}$				$\Delta t = \frac{1}{15} \Delta x^{\frac{2}{3}}$				
	20	1.06E–6	–	1.62E–6	–	3.97E–6	–	6.54E–6	–
	40	7.14E–8	3.90	1.47E–7	3.46	2.13E–7	4.22	4.61E–7	3.82
	80	5.08E–9	3.81	3.45E–8	2.09	7.39E–9	4.85	3.38E–8	3.77
	160	4.48E–10	3.50	8.14E–9	2.08	3.46E–10	4.41	2.01E–9	4.07
	320	4.51E–11	3.31	1.91E–9	2.09	2.17E–11	3.99	1.43E–10	3.81
	640	4.78E–12	3.24	4.15E–10	2.20	1.36E–12	3.99	7.79E–12	4.20
	P4	$\Delta t = \frac{1}{5} \Delta x^{\frac{3}{4}}$				$\Delta t = \frac{1}{5} \Delta x^{\frac{3}{4}}$			
10		1.27E–5	–	3.27E–5	–	5.78E–4	–	1.01E–3	–
20		1.95E–7	6.03	9.45E–7	5.11	3.90E–7	10.5	6.13E–7	10.6
40		1.66E–8	3.55	1.32E–7	2.83	7.38E–9	5.72	1.16E–8	5.72
80		1.29E–9	3.69	1.74E–8	2.95	1.52E–10	5.59	2.39E–10	5.59
160		7.54E–11	4.09	1.88E–9	3.21	3.34E–12	5.51	5.34E–12	5.48
320		4.19E–12	4.16	1.97E–10	3.26	6.95E–14	5.59	1.09E–13	5.61
640		2.37E–13	4.14	2.14E–11	3.20	1.13E–15	5.72	2.07E–15	5.72
P5	$\Delta t = \frac{1}{3} \Delta x^2$				$\Delta t = \frac{1}{3} \Delta x^2$				
	10	5.53E–6	–	1.38E–5	–	3.31E–5	–	5.48E–5	–
	20	4.94E–8	6.81	2.46E–7	5.81	7.48E–9	12.0	1.16E–8	12.0
	40	2.71E–9	4.19	2.11E–8	3.54	1.19E–10	5.97	1.87E–10	5.95
	80	1.57E–10	4.10	1.78E–9	3.56	1.88E–12	5.98	3.02E–12	5.95
	160	6.87E–12	4.52	1.46E–10	3.60	2.92E–14	6.01	4.66E–14	6.02
	320	2.40E–13	4.84	9.92E–12	3.88	4.56E–16	5.99	7.36E–16	5.98

with suitable initial and boundary conditions. Here $\rho \in (0, \rho_{\max})$ is the density, ρ_{\max} is the maximum (jam) density, and $q(\rho)$ is the traffic flow on a homogeneous highway, which is assumed to be a function of the density only in the LWR model. The flow q , the density ρ and the equilibrium speed u are related by

$$q(\rho) = u(\rho)\rho.$$

Example 5.6. The first traffic flow test example is taken from [18]. The flow-density function is given by a concave function

$$q(\rho) = \begin{cases} -0.4\rho^2 + 100\rho, & 0 \leq \rho \leq 50, \\ -0.1\rho^2 + 15\rho + 3500, & 50 \leq \rho \leq 100, \\ -0.024\rho^2 - 5.2\rho + 4760, & 100 \leq \rho \leq 350. \end{cases}$$

Consider a long homogeneous freeway of length 20 km. The entrance density is 50 veh/km. Due to an incident near the downstream end of the freeway, the piecewise linear traffic density profile shown in Fig. 5.5(a) is formed. To release the traffic jam condition downstream, the freeway entrance is blocked for 10 min, after which traffic is released again from the entrance at the capacity density 75 veh/km. After 20 min, the entrance flow returns back to normal with a density 50 veh/h. At the exit boundary, a traffic signal is installed, with a repeated pattern of 2 min green light (zero density) followed by 1 min red light (jam density). The numerical solutions are shown in Fig. 5.5(b)–(d), in which the solid lines are the exact solution and the symbols are the numerical solution obtained by our fifth order finite volume WENO scheme with the limiter using $N = 800$ cells. We can observe that our WENO scheme with the limiter produces very good approximations to the exact solution for this test case. Moreover, the numerical solutions are all in the interval $(0, \rho_{\max})$, which is an important advantage for such applications.

Example 5.7. We consider a similar problem but with a much more complicated flow-density function in [10]. The flow function $q(\rho) = \rho V_e(\rho)$ is given by

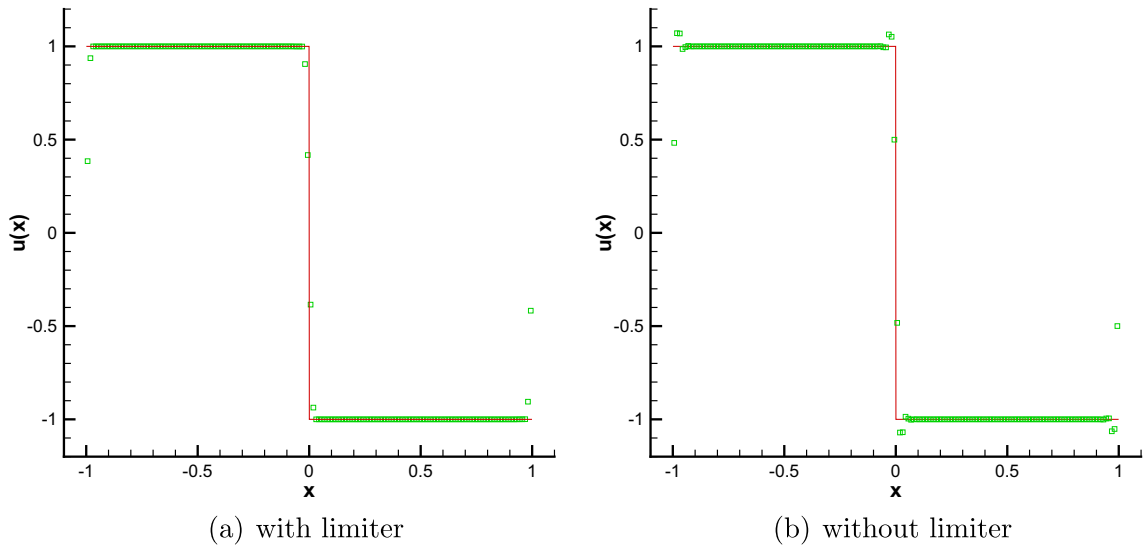


Fig. 6.1. Example 6.1. $N = 160$, $t = 100$, $\Delta x = \frac{2}{N}$ and $\Delta t = \frac{1}{15}\Delta x$. Solid line: exact solution; squares: numerical solution.

Table 6.2

Third order SSP time discretization and the DG scheme with the limiter, for the Burgers' equation with initial data $0.25 + 0.5 \sin(\pi x)$, $\Delta x = \frac{2}{N}$, $t = 0.15$.

N	SSP Runge–Kutta				SSP multi-step				
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order	
P2	$\Delta t = \frac{1}{24}\Delta x$				$\Delta t = \frac{1}{24}\Delta x$				
	20	5.12E-5	–	8.88E-5	–	5.14E-5	–	8.87E-5	–
	40	3.16E-6	4.01	6.78E-6	3.71	3.15E-6	4.03	6.23E-6	3.83
	80	1.98E-7	3.99	9.52E-7	2.83	1.94E-7	4.02	4.97E-7	3.65
	160	1.37E-8	3.85	1.84E-7	2.36	1.19E-8	4.02	3.81E-8	3.70
	320	1.23E-9	3.48	6.70E-8	1.46	7.45E-10	4.00	3.03E-9	3.65
	640	1.31E-10	3.23	2.30E-8	1.54	4.67E-11	3.99	2.53E-10	3.58
P3	$\Delta t = \frac{1}{15}\Delta x^{\frac{2}{3}}$				$\Delta t = \frac{1}{15}\Delta x^{\frac{2}{3}}$				
	20	1.45E-5	–	6.62E-5	–	1.04E-5	–	4.05E-5	–
	40	6.70E-7	4.44	2.79E-6	4.56	5.31E-7	4.29	1.54E-6	4.71
	80	3.54E-8	4.24	3.49E-7	2.99	2.48E-8	4.42	1.11E-7	3.79
	160	1.15E-9	4.94	1.14E-8	4.94	9.91E-10	4.64	4.48E-9	4.63
	320	6.67E-11	4.11	2.29E-9	2.31	4.96E-11	4.32	2.43E-10	4.20
	640	4.39E-12	3.92	5.13E-10	2.16	2.25E-12	4.46	1.62E-11	3.91
P4	$\Delta t = \frac{1}{7}\Delta x^{\frac{3}{4}}$				$\Delta t = \frac{1}{7}\Delta x^{\frac{3}{4}}$				
	10	7.54E-5	–	1.87E-4	–	3.08E-4	–	1.10E-3	–
	20	2.18E-6	5.11	5.19E-6	5.17	2.00E-6	7.26	5.57E-6	7.63
	40	1.00E-7	4.43	1.08E-6	2.26	5.04E-8	5.31	1.92E-7	4.85
	80	6.44E-9	3.96	1.94E-7	2.48	1.39E-9	5.18	5.58E-9	5.11
	160	1.24E-10	5.69	4.49E-9	5.44	3.81E-11	5.19	1.66E-10	5.07
	320	6.09E-12	4.35	4.79E-10	3.23	9.70E-13	5.29	5.11E-12	5.02
P5	$\Delta t = \frac{1}{3}\Delta x^{\frac{5}{6}}$				$\Delta t = \frac{1}{3}\Delta x^{\frac{5}{6}}$				
	10	7.85E-6	–	3.89E-5	–	8.15E-6	–	2.59E-5	–
	20	2.28E-7	5.10	2.26E-6	4.10	5.69E-5	-2.8	2.58E-4	-3.3
	40	1.61E-8	3.83	3.12E-7	2.86	2.11E-9	14.7	8.97E-9	14.8
	80	1.13E-9	3.82	4.36E-8	2.84	3.40E-11	5.95	1.44E-10	5.95
	160	1.10E-11	6.67	4.69E-10	6.54	5.36E-13	5.99	2.26E-12	5.99
	320	4.78E-13	4.53	3.72E-11	3.66	8.40E-15	5.99	3.59E-14	5.98
640	1.67E-14	4.83	2.47E-12	3.91	1.31E-16	5.99	5.63E-16	5.99	

$$V_e(\rho) = \frac{\tilde{V}^2}{2V_0} \left(-1 + \sqrt{1 + \frac{4V_0^2}{\tilde{V}^2}} \right)$$

Table 6.3

Third order SSP time discretization and the DG scheme with the limiter, for the two-dimensional linear equation with initial data $\sin(\pi(x+y))$, $\Delta x = \frac{2}{N}$, $t = 0.1$.

$N \times N$	SSP Runge–Kutta				SSP multi-step				
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order	
P2	$\Delta t = \frac{1}{14} \Delta x$				$\Delta t = \frac{1}{42} \Delta x$				
	10×10	1.69E–2	–	6.38E–2	–	1.79E–2	–	6.72E–2	–
	20×20	2.69E–4	5.97	7.89E–4	6.34	3.28E–4	5.77	1.09E–3	5.95
	40×40	4.53E–5	2.57	1.99E–4	1.99	2.43E–5	3.75	3.98E–5	4.78
	80×80	6.83E–6	2.73	5.33E–5	1.90	2.72E–6	3.16	4.03E–6	3.30
160×160	9.82E–7	2.80	1.40E–5	1.93	3.27E–7	3.06	5.05E–7	2.99	
P3	$\Delta t = \frac{1}{14} \Delta x^{\frac{4}{3}}$				$\Delta t = \frac{1}{42} \Delta x^{\frac{4}{3}}$				
	10×10	7.39E–4	–	1.74E–3	–	8.29E–4	–	1.34E–3	–
	20×20	3.27E–5	4.49	1.63E–4	3.42	1.76E–5	5.56	5.08E–5	4.72
	40×40	3.32E–6	3.30	2.57E–5	2.66	1.31E–6	3.74	3.64E–6	3.80
	80×80	3.10E–7	3.42	4.29E–6	2.58	1.16E–7	3.50	2.63E–7	3.79
160×160	2.76E–8	3.49	7.06E–7	2.36	7.78E–9	3.90	1.53E–8	4.10	

Table 6.4

Third order SSP time discretization and the DG scheme with the limiter, for the two-dimensional Burgers' equation with initial data $\sin(\pi(x+y))$, $\Delta x = \frac{2}{N}$, $t = 0.05$.

$N \times N$	SSP Runge–Kutta				SSP multi-step				
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order	
P2	$\Delta t = \frac{1}{14} \Delta x$				$\Delta t = \frac{1}{42} \Delta x$				
	10×10	3.68E–3	–	1.07E–2	–	3.60E–3	–	1.09E–2	–
	20×20	3.49E–4	3.40	1.11E–3	3.27	2.89E–4	3.64	7.21E–4	3.92
	40×40	5.73E–5	2.61	1.56E–4	2.83	4.57E–5	2.66	1.36E–4	2.41
	80×80	8.87E–6	2.69	4.70E–5	1.73	6.48E–6	2.82	2.42E–5	2.49
160×160	1.39E–6	2.67	1.30E–5	1.85	9.37E–7	2.79	4.51E–6	2.43	
P3	$\Delta t = \frac{1}{14} \Delta x^{\frac{4}{3}}$				$\Delta t = \frac{1}{42} \Delta x^{\frac{4}{3}}$				
	10×10	7.27E–4	–	2.42E–3	–	5.40E–4	–	2.24E–3	–
	20×20	8.76E–5	3.05	2.85E–4	3.08	7.16E–5	2.91	2.51E–4	3.12
	40×40	5.29E–6	4.05	2.08E–5	3.78	3.37E–6	4.41	1.69E–5	3.89
	80×80	4.07E–7	3.70	4.03E–6	2.37	2.04E–7	4.04	9.27E–7	4.19
160×160	3.11E–8	3.71	6.97E–7	2.53	1.25E–8	4.02	5.93E–8	3.97	

with

$$\tilde{V}(\rho) = \frac{1}{T_r} \left(\frac{1}{\rho} - \frac{1}{\rho_{\max}} \right) \sqrt{\frac{\alpha(\rho_{\max})}{\alpha(\rho)}}$$

and

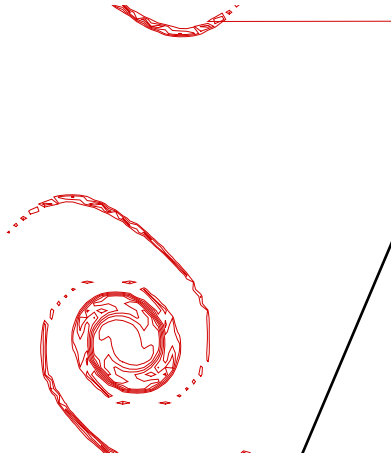
$$\alpha(\rho) = \alpha_0 + \Delta\alpha \left(\tanh\left(\frac{\rho - \rho_c}{\Delta\rho}\right) + 1 \right).$$

Here V_0 , T_r , ρ_{\max} , α_0 , $\Delta\alpha$, ρ_c and $\Delta\rho$ are all constant parameters to be determined by fitting them to the empirical data. The physical meaning of these parameters can be found in [10]. We simply choose some typical values mentioned in [10]: $V_0 = 110$ km/h, $T_r = 1.8$ s, $\rho_{\max} = 160$ veh/km, $\alpha_0 = 0.008$, $\Delta\alpha = 0.02$, $\rho_c = 0.27\rho_{\max}$ and $\Delta\rho = 0.1\rho_{\max}$. With all these parameters, the flow-density function $q(\rho)$ is well-defined, the graphs of this function and its second derivative are plotted in Fig. 5.6. It is clearly neither a globally concave nor a globally convex function.

Consider a long homogeneous freeway of length 20 km. The entrance density is constant 30 veh/km. The initial condition is given as a sine function $\rho_0(x) = \frac{135}{2} \sin(\frac{\pi}{10}x) + \frac{145}{2}$. At the exit boundary, a traffic signal is installed, with a repeated pattern of 1 min green light ($\rho = 10$ veh/km) followed by 2 min red light ($\rho = 140$ veh/km). The numerical solution is shown in Fig. 5.7, with a magnified graph for the boxed region shown in Fig. 5.8, where the solid line is the reference solution obtained by the first order Lax–Friedrichs scheme on a very fine grid ($N = 4,000,000$) and the circles are the numerical solution with $N = 1600$ cells at $t = 18$ min. We again observe very good resolution of our scheme for this nonconvex traffic flow model.

6. Numerical tests for the DG scheme

In this section we provide numerical examples for the DG method. We use the global Lax–Friedrichs flux, although the results hold for any other monotone flux as well. In the first subsection, we mainly test accuracy and compare Runge–Kutta



In this section, the time step is taken as that indicated by the proof of maximum principle in the earlier part of the paper, or that indicated by linear stability [3], whichever is smaller, unless otherwise stated.

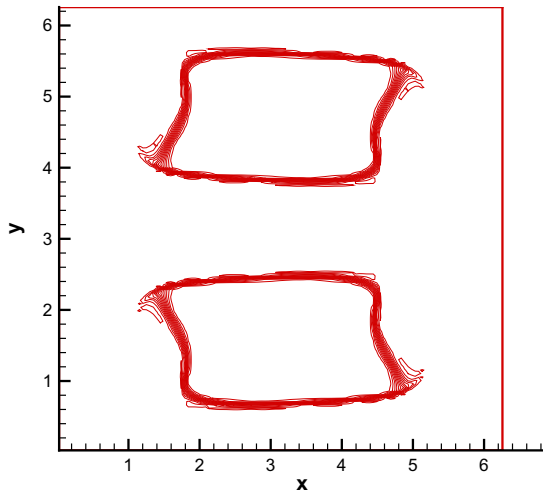
6.1. Standard test cases

Example 6.1. We solve the model equation $u_t + u_x = 0$, with periodic boundary conditions. The initial condition is $u_0(x) = \sin(2\pi x)$ on $[0, 1]$. Time steps are suitably adjusted in order to show a dominant spatial accuracy, see Table 6.1. We clearly observe the designed order of accuracy with a SSP multi-step time discretization, while the accuracy degeneracy phenomenon for the SSP Runge–Kutta time discretization is more prominent than the results in the previous section for the finite volume WENO scheme.

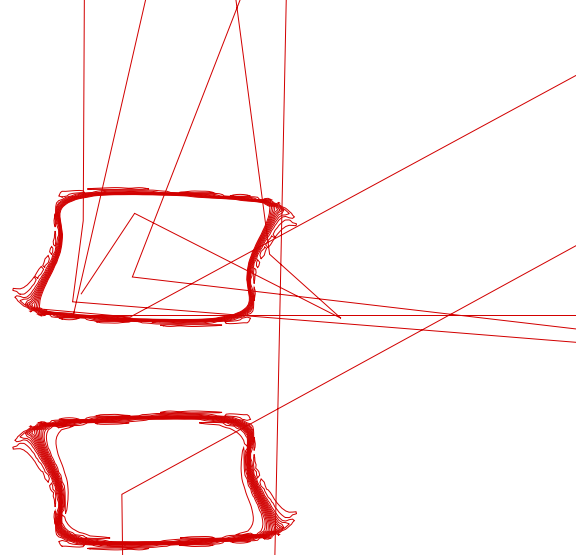
We also test another initial function for the same linear equation

$$u_0(x) = \begin{cases} 1, & -1 \leq x \leq 0, \\ -1, & 0 \leq x \leq 1. \end{cases}$$

The results at $t = 100$ are shown in Fig. 6.1. The scheme is the third order SSP Runge–Kutta method and fifth order DG scheme with the limiter. We can see that numerical solution maintains a strict maximum principle and has relatively good resolution for the discontinuity, for a relatively coarse mesh after a very long time simulation (50 time periods). As a



(a) 128^2 mesh, with limiter



comparison, we also show the result of the same DG scheme without the limiter. The overshoot near the discontinuity is then apparent.

Example 6.2. We solve the Burgers' equation with periodic boundary conditions

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad -1 \leq x \leq 1,$$

$$u(x, 0) = u_0(x).$$

The initial condition is $u_0(x) = 0.25 + 0.5 \sin(\pi x)$. Time steps are again suitably adjusted in order to show a dominant spatial accuracy, see Table 6.2. Again, we can clearly observe the designed order of accuracy with a SSP multi-step time discretization, while the accuracy degeneracy phenomenon for the SSP Runge–Kutta time discretization is more prominent than the results in the previous section for the finite volume WENO scheme.

Example 6.3. We solve the two-dimensional linear equation $u_t + u_x + u_y = 0$ with periodic boundary conditions. The initial condition is $u_0(x + y) = \sin(\pi(x + y))$ on $[-1, 1] \times [0, 1]$. Time steps are suitably adjusted in order to show a dominant spatial accuracy, see Table 6.3. Once more, we can clearly observe the designed order of accuracy with a SSP multi-step time discretization, while the accuracy degeneracy phenomenon for the SSP Runge–Kutta time discretization is more prominent than the results in the previous section for the finite volume WENO scheme.

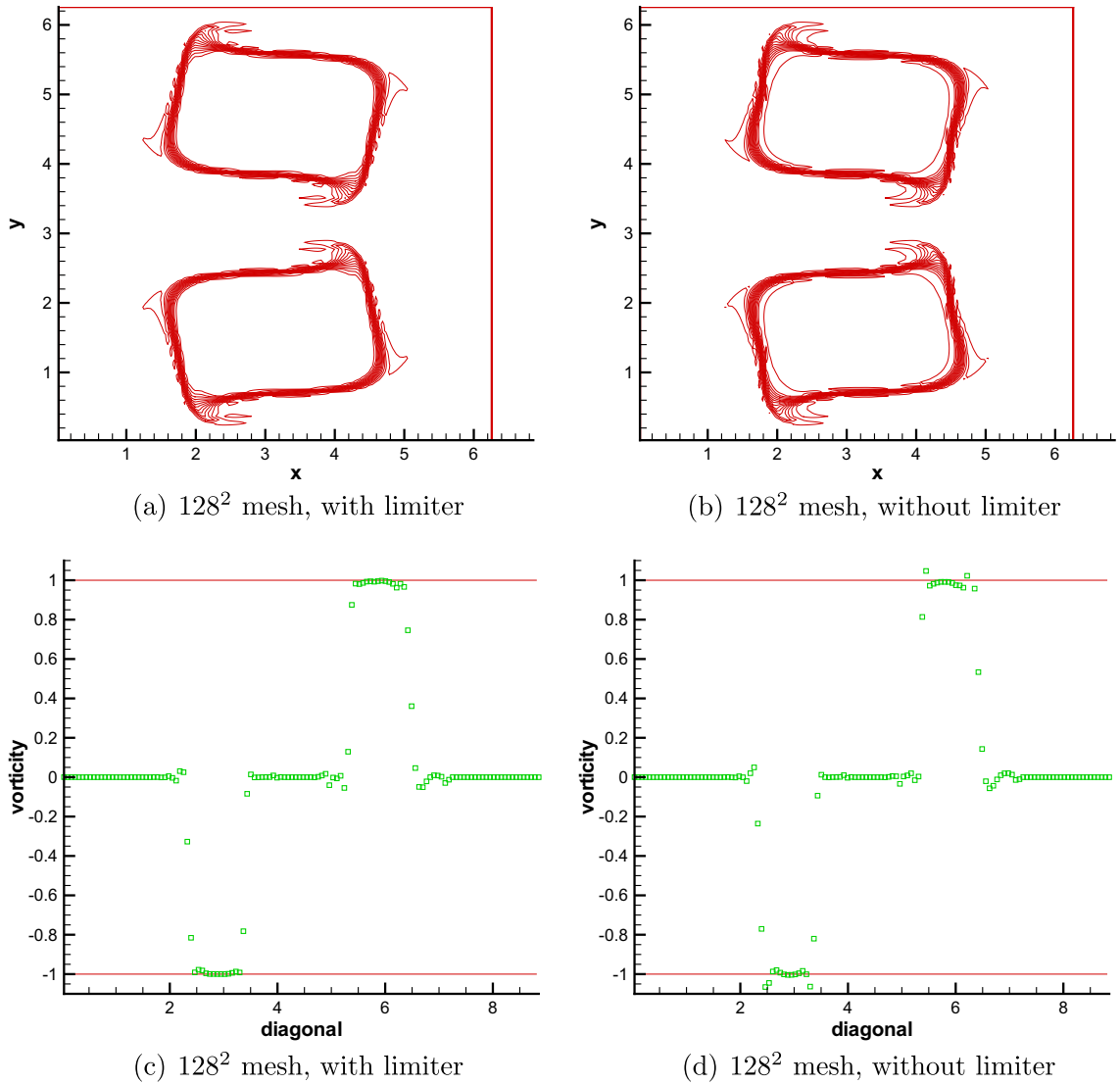


Fig. 6.5. Vorticity at $t = 10$, P^2 . Top: 30 equally spaced contours from -1.1 to 1.1 ; bottom: cut along the diagonal.

Example 6.4. We solve the two-dimensional Burgers' equation $u_t + (\frac{u^2}{2})_x + (\frac{u^2}{2})_y = 0$ with periodic boundary conditions. The initial condition is $\sin(\pi(x+y))$ on $[-1, 1] \times [-1, 1]$. Time steps are suitably adjusted in order to show a dominant spatial accuracy, see Table 6.4. We can clearly observe the designed order of accuracy with a SSP multi-step time discretization, while the accuracy degeneracy phenomenon for the SSP Runge–Kutta time discretization is more prominent than the results in the previous section for the finite volume WENO scheme.

6.2. Two dimensional incompressible flows

In this subsection, we test the DG scheme in [15] with our limiter for the examples in [15]. We take P^2 elements with the third order SSP Runge–Kutta time discretization. We use the Lax–Friedrichs flux, although the upwind flux can do the job as well.

Example 6.5. The example is used to check the accuracy. For the Euler Eqs. (4.1) and (4.2), with periodic boundary condition and initial data $\omega(x, y, 0) = -2 \sin(x) \sin(y)$ on the domain $[0, 2\pi] \times [0, 2\pi]$. The exact solution is $\omega(x, y, t) = -2 \sin(x) \sin(y)$, see Table 6.5. We clearly observe the designed order of accuracy for this special steady state solution.

Example 6.6. For this double shear layer problem, we solve the Euler equations (4.1) in the domain $[0, 2\pi] \times [0, 2\pi]$ with a periodic boundary condition and an initial condition

$$\omega(x, y, 0) = \begin{cases} \delta \cos(x) - \frac{1}{\rho} \operatorname{sech}^2((y - \pi/2)/\rho) & y \leq \pi, \\ \delta \cos(x) + \frac{1}{\rho} \operatorname{sech}^2((3\pi/2 - y)/\rho) & y > \pi, \end{cases}$$

where we take $\rho = \pi/15$ and $\delta = 0.05$.

The solution quickly develops into roll-ups with smaller and smaller scales, so on any fixed grid the full resolution is lost eventually. We use uniform meshes of 64×64 and 128×128 rectangles and perform the computation up to $t = 8$. We plot the contours of the vorticity ω at $t = 6$ and $t = 8$, see Figs. 6.2 and 6.3. Although one can barely see any difference between the results with the limiter and without the limiter from the contour, we point out that the numerical solutions of the scheme with the limiter are in the range $[-\delta - \frac{1}{\rho}, \delta + \frac{1}{\rho}]$.

Example 6.7. The vortex patch problem. We solve the Euler Eqs. (4.1) and (4.2) in $[0, 2\pi] \times [0, 2\pi]$ with the initial condition

$$\omega(x, y, 0) = \begin{cases} -1, & \frac{\pi}{2} \leq x \leq \frac{3\pi}{2}, \frac{\pi}{4} \leq y \leq \frac{3\pi}{4}; \\ 1, & \frac{\pi}{2} \leq x \leq \frac{3\pi}{2}, \frac{5\pi}{4} \leq y \leq \frac{7\pi}{4}; \\ 0, & \text{otherwise} \end{cases}$$

and periodic boundary conditions. The contour plots of the vorticity ω , with 30 equally spaced contour lines between $\omega = -1.1$ and $\omega = 1.1$, are given for $t = 5$ and $t = 10$, see Figs. 6.4 and 6.5. Again, we cannot observe any significant difference between the two results in the contour plots. The cut along the diagonal gives us a clearer view of the advantage in using the limiter. Clearly, the result with the limiter is strictly within the range $[-1, 1]$ while there are apparent overshoots outside $[-1, 1]$ for the result without the limiter.

7. Concluding remarks

In this paper, we have established a general framework to construct arbitrarily high order accurate maximum-principle-satisfying schemes for scalar conservation laws. This is the first time that genuinely high order schemes are obtained which satisfy strict maximum principle especially for multi-dimensional nonlinear problems. The main difficulty is to get the cell averages in the next time step to be within the range $[m, M]$ with a limiter which does not destroy accuracy. The exact time evolution has this property, but it cannot be easily implemented in multi-dimensions with high order accuracy. We prove that SSP time discretization can maintain the cell averages in $[m, M]$ as long as the values of the approximation polynomials at certain quadrature points are in $[m, M]$, which can be enforced by the simplified linear scaling limiter. The implementation is quite easy. With the addition of the limiter in this paper, which involves small additional computational cost, to the DG scheme or the finite volume scheme (e.g. ENO and WENO), then the numerical solutions will satisfy the maximum principle under suitable CFL condition.

We also show that the same result holds for the DG or finite volume schemes solving the two-dimensional incompressible Euler equations in the vorticity stream-function formulation [15]. The result is also valid for any passive convection equation with divergence-free velocity coefficients.

We have tested the fifth order finite volume WENO scheme and the DG scheme with limiter on a variety of examples including those from traffic flow models and two-dimensional incompressible Euler equations. We clearly observe strict maximum principle preserving in all these tests. The SSP multi-step time discretization gives us genuine high order accuracy. The accuracy for the SSP Runge–Kutta time discretization may however degenerate on a fine mesh, similar to the phenomenon discussed in [1] in the context of boundary conditions. However, this effect is mainly of theoretic interest as it can only be observed for very fine meshes.

In this paper we have only shown a straightforward extension of the one dimensional algorithm to two-dimensional finite volume or DG schemes on a rectangular mesh. For triangular meshes, the idea of rewriting the scheme as a convex combination of monotone schemes is still plausible and will be explored in the future. Generalizations to finite difference schemes, and to nonlinear systems in the context of positivity preserving of certain physical quantities such as density and pressure, constitute our ongoing work.

Appendix A. WENO reconstruction for values at the quadrature points

In this Appendix we document the details of the one-dimensional WENO reconstruction problem: assuming a uniform grid, given the cell average \bar{u}_j on each cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, use the WENO reconstruction to obtain the point values at the quadrature points in the interval I_j . We will consider the fifth order accurate WENO procedure here, for which the three-point Gauss quadrature is applicable.

For the three-point Gauss quadrature on $[-\frac{1}{2}, \frac{1}{2}]$, the quadrature points are $x_1 = \frac{\sqrt{15}}{10}$, $x_2 = 0$, $x_3 = -\frac{\sqrt{15}}{10}$ with the weights $w_1 = 5/18$, $w_2 = 4/9$, $w_3 = 5/18$.

For convenience, we will use the rescaled variable $x' = \frac{x-x_j}{\Delta x}$. Assume the fixed stencil reconstruction polynomials on the stencil $\{j+2, j+1, j\}$, $\{j+1, j, j-1\}$, $\{j, j-1, j-2\}$ and $\{j+2, j+1, j, j-1, j-2\}$ are p_1 , p_2 , p_3 and p , respectively. Then

Table A.1
Linear weights.

r_i^β	$\beta = 1$	$\beta = 2$	$\beta = 3$
$i = 1$	$\frac{1008+71\sqrt{15}}{5240}$	$\frac{403}{655}$	$\frac{1008-71\sqrt{15}}{5240}$
$i = 2$	$-\frac{9}{80}$	$\frac{49}{40}$	$-\frac{9}{80}$
$i = 3$	$\frac{1008-71\sqrt{15}}{5240}$	$\frac{403}{655}$	$\frac{1008+71\sqrt{15}}{5240}$

$$\begin{aligned}
 p_1(x') &= \frac{\bar{u}_{j+2} - 2\bar{u}_{j+1} + \bar{u}_j}{2} (x')^2 + \frac{-\bar{u}_{j+2} + 4\bar{u}_{j+1} - 3\bar{u}_j}{2} (x') + \frac{-\bar{u}_{j+2} + 2\bar{u}_{j+1} + 23\bar{u}_j}{24}, \\
 p_2(x') &= \frac{\bar{u}_{j+1} - 2\bar{u}_j + \bar{u}_{j-1}}{2} (x')^2 + \frac{\bar{u}_{j+1} - \bar{u}_{j-1}}{2} (x') + \frac{-\bar{u}_{j+1} + 26\bar{u}_j - \bar{u}_{j-1}}{24}, \\
 p_3(x') &= \frac{\bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j}{2} (x')^2 + \frac{\bar{u}_{j-2} - 4\bar{u}_{j-1} + 3\bar{u}_j}{2} (x') + \frac{-\bar{u}_{j-2} + 2\bar{u}_{j-1} + 23\bar{u}_j}{24}
 \end{aligned}$$

and

$$p(x') = a_4(x')^4 + a_3(x')^3 + a_2(x')^2 + a_1(x') + a_0,$$

with

$$\begin{aligned}
 a_0 &= \frac{9\bar{u}_{j+2} - 116\bar{u}_{j+1} + 2134\bar{u}_j - 116\bar{u}_{j-1} + 9\bar{u}_{j-2}}{1920}, \\
 a_1 &= \frac{-5\bar{u}_{j+2} + 34\bar{u}_{j+1} - 34\bar{u}_{j-1} + 5\bar{u}_{j-2}}{48}, \\
 a_2 &= \frac{-\bar{u}_{j+2} + 12\bar{u}_{j+1} - 22\bar{u}_j + 12\bar{u}_{j-1} - \bar{u}_{j-2}}{16}, \\
 a_3 &= \frac{\bar{u}_{j+2} - 2\bar{u}_{j+1} + 2\bar{u}_{j-1} - \bar{u}_{j-2}}{12}, \\
 a_4 &= \frac{\bar{u}_{j+2} - 4\bar{u}_{j+1} + 6\bar{u}_j - 4\bar{u}_{j-1} + \bar{u}_{j-2}}{24}.
 \end{aligned}$$

Define the linear weights r_i^β by

$$p(x_\beta) = \sum_{i=1}^3 r_i^\beta p_i(x_\beta), \quad \beta = 1, 2, 3.$$

Solving the linear system above gives us the linear weights in Table A.1.

We can use the same smoothness indicators as in [2]:

$$\begin{aligned}
 b_1 &= \frac{13}{12} (\bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2})^2 + \frac{1}{4} (3\bar{u}_j - 4\bar{u}_{j+1} + \bar{u}_{j+2})^2, \\
 b_2 &= \frac{13}{12} (\bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1})^2 + \frac{1}{4} (\bar{u}_{j-1} - \bar{u}_{j+1})^2, \\
 b_3 &= \frac{13}{12} (\bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j)^2 + \frac{1}{4} (\bar{u}_{j-2} - 4\bar{u}_{j-1} + 3\bar{u}_j)^2.
 \end{aligned}$$

Now define the nonlinear weights w_i^β by

$$w_i^\beta = \frac{d_i^\beta}{\sum_{i=1}^3 d_i^\beta}, \quad d_i^\beta = \frac{r_i^\beta}{(\epsilon + b_i)^2},$$

where $\epsilon = 10^{-6}$. The fifth order WENO reconstruction for the values at the three Gauss quadrature points of the cell I_j are:

$$u_j^\beta = \sum_{i=1}^3 w_i^\beta p_i(x_\beta),$$

where

$$\begin{aligned}
 p_1(x_1) &= \frac{2 + 3\sqrt{15}}{60} \bar{u}_{j+2} - \frac{3\sqrt{15} + 1}{15} \bar{u}_{j+1} + \frac{62 + 9\sqrt{15}}{60} \bar{u}_j, \\
 p_2(x_1) &= \frac{2 - 3\sqrt{15}}{60} \bar{u}_{j+1} + \frac{14}{15} \bar{u}_j + \frac{2 + 3\sqrt{15}}{60} \bar{u}_{j-1}, \\
 p_3(x_1) &= \frac{62 - 9\sqrt{15}}{60} \bar{u}_j + \frac{3\sqrt{15} - 1}{15} \bar{u}_{j-1} + \frac{2 - 3\sqrt{15}}{60} \bar{u}_{j-2}, \\
 p_1(x_2) &= -\frac{1}{24} \bar{u}_{j+2} + \frac{1}{12} \bar{u}_{j+1} + \frac{23}{24} \bar{u}_j, \\
 p_2(x_2) &= -\frac{1}{24} \bar{u}_{j+1} + \frac{13}{12} \bar{u}_j - \frac{1}{24} \bar{u}_{j-1}, \\
 p_3(x_2) &= \frac{23}{24} \bar{u}_j + \frac{1}{12} \bar{u}_{j-1} - \frac{1}{24} \bar{u}_{j-2}, \\
 p_1(x_3) &= \frac{2 - 3\sqrt{15}}{60} \bar{u}_{j+2} + \frac{3\sqrt{15} - 1}{15} \bar{u}_{j+1} + \frac{62 - 9\sqrt{15}}{60} \bar{u}_j, \\
 p_2(x_3) &= \frac{2 + 3\sqrt{15}}{60} \bar{u}_{j+1} + \frac{14}{15} \bar{u}_j + \frac{2 - 3\sqrt{15}}{60} \bar{u}_{j-1}, \\
 p_3(x_3) &= \frac{62 + 9\sqrt{15}}{60} \bar{u}_j - \frac{3\sqrt{15} + 1}{15} \bar{u}_{j-1} + \frac{2 + 3\sqrt{15}}{60} \bar{u}_{j-2}.
 \end{aligned}$$

References

- [1] M.H. Carpenter, D. Gottlieb, S. Abarbanel, W.-S. Don, The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error, *SIAM Journal on Scientific Computing* 16 (1995) 1241–1252.
- [2] B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, in: A. Quarteroni (Ed.), *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Lecture Notes in Mathematics, vol. 1697, Springer, 1998.
- [3] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Mathematics of Computation* 52 (1989) 411–435.
- [4] M. Crandall, A. Majda, Monotone difference approximations for scalar conservation laws, *Mathematics of Computation* 34 (1980) 1–21.
- [5] C.M. Dafermos, *Hyperbolic Conservation Laws in Continuum Physics*, Springer-Verlag, Berlin, Heidelberg, New York, 2000.
- [6] S. Gottlieb, D.I. Ketcheson, C.-W. Shu, High order strong stability preserving time discretizations, *Journal of Scientific Computing* 38 (2009) 251–289.
- [7] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high order time discretization methods, *SIAM Review* 43 (2001) 89–112.
- [8] A. Harten, High resolution schemes for hyperbolic conservation laws, *Journal of Computational Physics* 49 (1983) 357–393.
- [9] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes, III, *Journal of Computational Physics* 71 (1987) 231–303.
- [10] D. Helbing, A. Hennecke, V. Shvetsov, M. Treiber, MASTER: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model, *Transportation Research Part B* 35 (2001) 183–211.
- [11] J.S. Hesthaven, S. Gottlieb, D. Gottlieb, *Spectral Methods for Time-dependent Problems*, Cambridge University Press, 2007.
- [12] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *Journal of Computational Physics* 126 (1996) 202–228.
- [13] G.-S. Jiang, E. Tadmor, Nonoscillatory central schemes for multidimensional hyperbolic conservative laws, *SIAM Journal on Scientific Computing* 19 (1998) 1892–1917.
- [14] D. Levy, E. Tadmor, Non-oscillatory central schemes for the incompressible 2D Euler equations, *Mathematical Research Letters* 4 (1997) 321–340.
- [15] J.-G. Liu, C.-W. Shu, A high-order discontinuous Galerkin method for 2D incompressible flows, *Journal of Computational Physics* 160 (2000) 577–596.
- [16] X.-D. Liu, S. Osher, Non-oscillatory high order accurate self similar maximum principle satisfying shock capturing schemes, *SIAM Journal on Numerical Analysis* 33 (1996) 760–779.
- [17] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, *Journal of Computational Physics* 115 (1994) 200–212.
- [18] Y. Lu, S.C. Wong, M. Zhang, C.-W. Shu, W. Chen, Explicit construction of entropy solutions for the Lighthill–Whitham–Richards traffic flow model with a piecewise quadratic flow-density relationship, *Transportation Research Part B* 42 (2008) 355–372.
- [19] S. Osher, S. Chakravarthy, High resolution schemes and the entropy condition, *SIAM Journal on Numerical Analysis* 21 (1984) 955–984.
- [20] B. Perthame, C.-W. Shu, On positivity preserving finite volume schemes for Euler equations, *Numerische Mathematik* 73 (1996) 119–130.
- [21] R. Sanders, A third-order accurate variation nonexpansive difference scheme for single nonlinear conservation law, *Mathematics of Computation* 51 (1988) 535–558.
- [22] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, *Journal of Computational Physics* 175 (2002) 108–127.
- [23] C.-W. Shu, Total-variation-diminishing time discretizations, *SIAM Journal on Scientific and Statistical Computing* 9 (1988) 1073–1084.
- [24] C.-W. Shu, High order weighted essentially non-oscillatory schemes for convection dominated problems, *SIAM Review* 51 (2009) 82–126.
- [25] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *Journal of Computational Physics* 77 (1988) 439–471.
- [26] X. Zhang, C.-W. Shu, A genuinely high order total variation diminishing scheme for one-dimensional scalar conservation laws, *SIAM Journal on Numerical Analysis*, submitted for publication.