

# A semi-Lagrangian finite difference WENO scheme for scalar nonlinear conservation laws

Chieh-Sen Huang<sup>a,1</sup>, Todd Arbogast<sup>b,2</sup>, Chen-Hui Hung<sup>c</sup>

<sup>a</sup> Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan, R.O.C.

<sup>b</sup> University of Texas at Austin; Institute for Computational Engineering and Sciences; 201 EAST 24th St., Stop C0200; Austin, TX 78712-1229; U.S.A.

<sup>c</sup> Department of Mathematic and Physical Sciences, Air Force Academy, No. Sisou 1, Jieshou W. Rd., Gangshan Dist., Kaohsiung City 82047, Taiwan, R.O.C.

---

## Abstract

For a nonlinear scalar conservation law in one-space dimension, we develop a locally conservative semi-Lagrangian finite difference scheme based on weighted essentially non-oscillatory reconstructions (SL-WENO). This scheme has the advantages of both WENO and semi-Lagrangian schemes. It is a locally mass conservative finite difference scheme, it is formally high-order accurate in space, it has small time truncation error, and it is essentially non-oscillatory. The scheme is nearly free of a CFL time step stability restriction for linear problems, and it has a relaxed CFL condition for nonlinear problems. The scheme can be considered as an extension of the SL-WENO scheme of Qiu and Shu [J. Comput. Phys., 2011] developed for linear problems. The new scheme is based on a standard sliding average formulation with the flux function defined using WENO reconstructions of (semi-Lagrangian) characteristic tracings of grid points. To handle nonlinear problems, we use an approximate, locally frozen trace velocity and a flux correction step. A special two-stage WENO reconstruction procedure is developed that is biased to the upstream direction. A Strang splitting algorithm is used for higher-dimensional problems. Numerical results are provided to illustrate the performance of the scheme and verify its formal accuracy. Included are applications to the Vlasov-Poisson and guiding-center models of plasma flow.

*Keywords:* hyperbolic, semi-Lagrangian, WENO reconstruction, Eulerian-Lagrangian, traceline, locally frozen  
*2000 MSC:* 65M06, 65M25, 76M20, 76X05

---

## 1. Introduction

We develop a semi-Lagrangian finite difference WENO (SL-WENO) scheme to approximate the scalar conservation law

$$u_t + (f(u))_x = 0, \quad x \in \mathbb{R}, t > 0, \quad (1)$$

$$u(x, 0) = u^0(x), \quad x \in \mathbb{R}, \quad (2)$$

with the possibly nonlinear flux  $f(u) = f(u; x, t)$ . Moreover, multiple space dimensions will be treated using operator splitting.

The semi-Lagrangian computational scheme uses a mixture of Eulerian and Lagrangian reference frames in the sense that it has a fixed (Eulerian) numerical grid but advances each time step by evolving the partial differential equations by propagating information along (Lagrangian) characteristic curves. A semi-Lagrangian scheme does not

---

*Email addresses:* huangcs@math.nsysu.edu.tw (Chieh-Sen Huang), arbogast@ices.utexas.edu (Todd Arbogast), hungch@math.nsysu.edu.tw (Chen-Hui Hung)

<sup>1</sup>Supported in part by Taiwan National Science Council grant NSC 102-2115-M-110-010-MY3.

<sup>2</sup>Supported in part by U.S. National Science Foundation grant DMS-1418752.

inherently suffer the CFL time step restriction of an Eulerian approach. In consequence, longer time step evolution can be achieved, potentially resulting in less computational effort and less numerical diffusion.

In 2011, Qiu and Shu [1] (see also [2]) developed a semi-Lagrangian finite difference WENO method for *linear* scalar conservation laws for which  $f(u) = au$  for some advection velocity  $a(x, t)$ . In fact, they could develop the linear weights needed in the WENO reconstruction [3, 4] only when the velocity  $a$  is constant. For variable advection velocity, they developed weights based on heuristic empirical criteria.

In this paper we develop a locally mass conservative SL-WENO scheme for *nonlinear* scalar conservation laws. In the linear case, it is an extension of the scheme of Qiu and Shu [1]. It reduces to their scheme if the advection velocity  $a$  is constant and linear WENO weights are used; however, our scheme has exact WENO linear weights in the case of variable advection problems. Our scheme is new in the case of nonlinear problems, and it can be viewed as an extension of related work on finite volume Eulerian-Lagrangian WENO schemes [5, 6], in which characteristic tracing is incorporated only in an approximate way. For simplicity of exposition, we will describe only the third order SL-WENO scheme (SL-WENO3), but in principle any order scheme can be implemented.

Our finite difference scheme is based on the standard sliding average formulation of Shu and Osher [7, 8], but the flux function is defined using WENO reconstructions of (semi-Lagrangian) characteristic tracings of grid points [1]. To handle nonlinear problems for which the trace velocity is unknown, we use a fixed, approximate trace velocity (as is done in, e.g., [9, 6] and in arbitrary Lagrangian-Eulerian (ALE) schemes [9]). This approximate trace velocity is locally frozen to avoid numerical difficulties [6]. The introduction of an approximate trace velocity means that we do not trace the exact characteristic curves, and necessitates the use of a flux correction step [6], which is implemented using the approach of Levy, Puppo, and Russo [10] developed for CWENO schemes.

Unlike the direct WENO reconstructions used by Qiu and Shu [1], we develop a two-stage WENO reconstruction procedure. We first reconstruct cell averages of the numerical flux function, and then we reconstruct the flux function at the needed point. In our procedure, the linear WENO weights always exist. We nonlinearly weight at each stage using a smoothness indicator that is based on the underlying smoothness of the solution  $u$ ; that is, rather than using the smoothness of the cell average fluxes in the second stage, we use the smoothness indicator for  $u$  in both stages.

For nonlinear problems, the scheme requires a special upstream bias in the computation of the flux function between two grid points, since nonsmooth shocks may develop. The Rankine-Hugoniot shock speed is computed assuming a shock based on the solution  $u$  between the two grid points (i.e., the Roe speed [11]). The direction of this assumed shock determines the wave direction and consequent upstream biasing in our scheme.

The flux correction step requires a relaxed CFL constraint, given later in (42) [12, 6]. The constraint is based on the speed of the difference between the true value of  $\partial f(u)/\partial u$  and the fixed approximate trace velocity. If these are approximately equal, the constraint is very mild.

For multi-dimensional problems, we use a standard, usually second order Strang splitting which decouples the differential equation into a sequence of one-dimensional problems (see (53)). The advantage of a finite difference scheme is that the splitting error manifests itself as temporal error, because the splitting does not induce a shearing of the variable coefficients, and so the scheme maintains its formal high order spatial accuracy [2].

We present a detailed description of our formally third order SL-WENO3 scheme in the next section. Sections 3 and 4 are devoted to numerical results of SL-WENO3 and SL-WENO5 in one and two space dimensions, respectively. In Section 5 we apply our new scheme to two models of plasma transport: (1) the Vlasov-Poisson system, which has a constant advection within the Strang split scheme, and (2) the guiding-center model [13, 14], which has a nonconstant advection. We conclude briefly in the last section.

## 2. The semi-Lagrangian finite difference WENO scheme

The development of our semi-Lagrangian finite difference WENO (SL-WENO) scheme for nonlinear problems parallels and extends the development of the finite difference scheme given by Qiu and Shu [1] for linear problems. For simplicity of exposition, we describe the third order SL-WENO scheme.

Partition time and space as  $t^0 < t^1 < t^2 < \dots$  and  $\dots < x_{-1} < x_0 < x_1 < \dots$ , respectively, where  $\Delta x = x_i - x_{i-1}$  is constant. We approximate  $u(x_i, t^n)$  as  $u_i^n$ . Integrate (1) over  $[t^n, t^{n+1}]$  and evaluate at a grid point  $x_i$  to obtain

$$u_i^{n+1} = u_i^n - \left( \int_{t^n}^{t^{n+1}} f(u) dt \right) \Big|_{x=x_i} = u_i^n - \mathcal{F}'(x_i), \quad (3)$$

where

$$\mathcal{F}(x) = \int_{t^n}^{t^{n+1}} f(u) dt. \quad (4)$$

We introduce a function  $\mathcal{H}(x)$  whose sliding average is  $\mathcal{F}(x)$  [7, 8]; that is,

$$\mathcal{F}(x) = \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} \mathcal{H}(\xi) d\xi. \quad (5)$$

Taking the derivative in  $x$  gives

$$\mathcal{F}'(x) = \frac{1}{\Delta x} [\mathcal{H}(x + \Delta x/2) - \mathcal{H}(x - \Delta x/2)], \quad (6)$$

and thus (3) can be written in conservative form as

$$u_i^{n+1} = u_i^n - \frac{1}{\Delta x} [\mathcal{H}(x_{i+1/2}) - \mathcal{H}(x_{i-1/2})], \quad (7)$$

where  $x_{i+1/2} = x_i + \Delta x/2$ . We call  $\mathcal{H}(x_{i+1/2})$  the *flux function* at  $x = x_{i+1/2}$ .

### 2.1. Evaluation of the cell averages $\bar{\mathcal{H}}_i$ of the flux function

As in a finite difference WENO scheme,  $\mathcal{H}(x_{i+1/2})$  is reconstructed from its neighboring cell averages. For most of this section, consider that the index  $i$  is fixed. Then the cell average of  $\mathcal{H}$  at  $x_i$  is

$$\bar{\mathcal{H}}_i = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathcal{H}(x) dx = \mathcal{F}(x_i), \quad (8)$$

due to (5). We use Lagrangian tracing to define  $\bar{\mathcal{H}}_i$  [1], but to handle the complexity of nonconstant and possibly nonlinear velocities, we introduce a known, fixed trace velocity  $v_{i+1/2}(x, t)$  (fixed for each index  $i$ ) [6] that approximates the true characteristic velocity. In Section 2.5 we will discuss how  $v_{i+1/2}$  should be chosen. The  $i$ th traceline from  $(x_i, t^{n+1})$  backward in time is defined by

$$\frac{d\check{x}_i}{dt} = v_{i+1/2}(\check{x}_i(t), t) \quad \text{and} \quad \check{x}_i(t^{n+1}) = x_i. \quad (9)$$

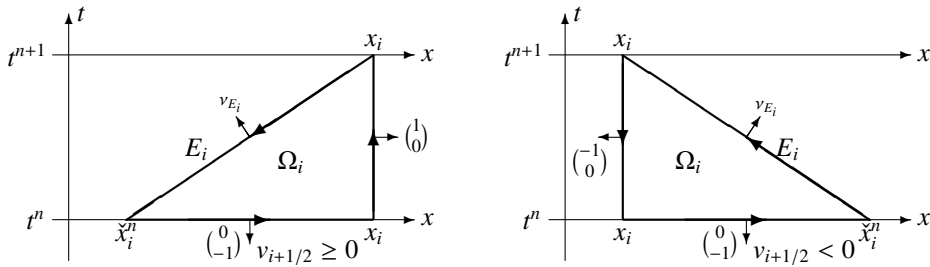


Figure 1: Representation of the “triangular” space-time region  $\Omega_i$ , depending on the sign of  $v_{i+1/2}$ , which is bounded by the lines  $t = t^n$  and  $x = x_i$  and the curve  $E_i$  (shown as a straight line). The counterclockwise orientation is shown in each case.

From grid point  $(x_i, t^{n+1})$ , we follow the traceline to define the curve

$$E_i = \{\check{x}_i(t) : t^n \leq t \leq t^{n+1}\} \subset \mathbb{R} \times [t^n, t^{n+1}].$$

The end of this curve is located at  $(\check{x}_i^n, t^n)$ , where the traceback point  $\check{x}_i^n$  is  $\check{x}_i(t^n)$ . We denote by  $\Omega_i \subset \mathbb{R} \times [t^n, t^{n+1}]$  the space-time “triangular” region bounded by  $E_i$  and the two lines joining  $(\check{x}_i^n, t^n)$  to  $(x_i, t^n)$  and back to  $(x_i, t^{n+1})$ , as shown in Fig. 1. We integrate (1) over the region  $\Omega_i$  and apply the Divergence Theorem. The sign of  $v_{i+1/2}$  determines

whether  $\Omega_i$  slopes to the left or the right. Letting the top and bottom signs in a stacked pair refer to the cases  $v_{i+1/2} \geq 0$  and  $v_{i+1/2} < 0$ , respectively (more precisely, the cases are  $\check{x}_i^n \leq x_i$  and  $\check{x}_i^n > x_i$ ), we obtain that

$$\begin{aligned} 0 &= \int_{\Omega_i} [u_t + (f(u))_x] dx dt = \int_{\Omega_i} \nabla_{x,t} \cdot (f(u), u) dx dt = \int_{\partial\Omega_i} (f(u), u) \cdot \nu_{E_i} ds \\ &= \mp \int_{\check{x}_i^n}^{x_i} u(x, t^n) dx \pm \int_{t^n}^{t^{n+1}} f(u(x_i, t)) dt + \int_{E_i} (f(u), u) \cdot \nu_{E_i} ds, \end{aligned} \quad (10)$$

where  $\nu_{E_i}$  is the outer unit normal to  $\partial\Omega_i$  along  $E_i$ . This direction is orthogonal to  $(v_{i+1/2} u, u)$ , and so we can replace

$$\int_{E_i} (f(u), u) \cdot \nu_{E_i} ds = \int_{E_i} (f(u) - v_{i+1/2} u) \nu_{E_{i,x}} ds = \int_{E_i} g_i(u) \nu_{E_{i,x}} ds,$$

where  $g_i(u) = f(u) - v_{i+1/2} u$ . Thus  $\bar{\mathcal{H}}_i = \mathcal{F}(x_i) = \int_{t^n}^{t^{n+1}} f(u(x_i, t)) dt$  is

$$\bar{\mathcal{H}}_i = \int_{\check{x}_i^n}^{x_i} u(x, t^n) dx \mp \int_{E_i} g_i(u) \nu_{E_{i,x}} ds. \quad (11)$$

Note that the last term above would vanish if  $f(u) = au$  has a constant linear velocity and one chose  $v_{i+1/2} = a$ . The finite difference schemes in [1] are based on this assumption. Here, we allow  $a(x, t)$  to be non-constant, and we even treat nonlinear problems. The price we pay is that we must compute the last term involving  $g_i(u)$  by some WENO reconstruction to the desired accuracy.

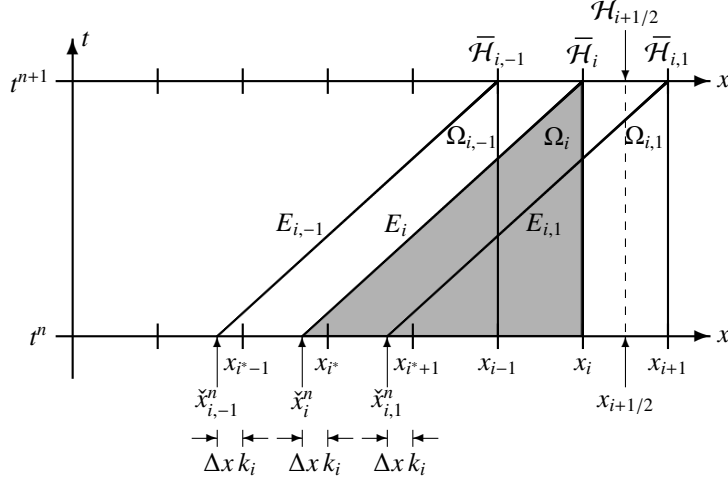


Figure 2: Representation of the “triangular” space-time region  $\Omega_i = \Omega_{i,0}$  and its two translates  $\Omega_{i,\ell}$  for  $\ell = -1, 1$ , assuming  $v_{i+1/2} > 0$ . The region  $\Omega_i$  is highlighted, and each  $\Omega_{i,\ell}$  is bounded by the lines  $t = t^n$  and  $x = x_{i+\ell}$  and the curve  $E_{i,\ell}$  (shown as a straight line), where  $E_i = E_{i,0}$ . These regions are used to define  $\bar{\mathcal{H}}_{i,\ell}$ ,  $\ell = -1, 0, 1$  ( $\bar{\mathcal{H}}_i = \bar{\mathcal{H}}_{i,0}$ ), which in turn are used to WENO reconstruct  $\mathcal{H}_{i+1/2} \approx \mathcal{H}(x_{i+1/2})$ .

Later we will reconstruct  $\mathcal{H}_{i+1/2} \approx \mathcal{H}(x_{i+1/2})$  from values of  $\bar{\mathcal{H}}$  near  $x_i$ . We have defined  $\bar{\mathcal{H}}_i$  in (11). We define neighboring values in a similar way, but we use a locally frozen trace velocity [6] to insure that no numerical complications arise from merging or diverging trachelines (e.g., all the traceback points will be regularly spaced so that linear WENO weights will be known to exist). For fixed  $i$ , the parallel tracheline for a neighboring point  $x_{i+\ell}$  is defined by translation, that is, we let

$$\check{x}_{i,\ell}(t) = \check{x}_i(t) + \ell \Delta x \quad (12)$$

be the  $i$ th tracheline offset by index  $\ell$ . Then  $\bar{\mathcal{H}}_{i,\ell}$  is defined by integrating over  $\Omega_{i,\ell} = \Omega_i + (\ell \Delta x, 0) \subset \mathbb{R} \times [t^n, t^{n+1}]$ , which results in

$$\bar{\mathcal{H}}_{i,\ell} = \int_{\check{x}_{i,\ell}^n}^{x_{i+\ell}} u(x, t^n) dx \mp \int_{E_{i,\ell}} g_i(u) \nu_{E_{i,\ell,x}} ds, \quad (13)$$

where  $\check{x}_{i,\ell}^n = \check{x}_{i,\ell}(t^n)$  and  $E_{i,\ell} = E_i + (\ell\Delta x, 0)$ , as depicted in Fig. 2.

We define  $x_{i^*}$  to be the closest grid point to  $\check{x}_i^n$  that is between  $x_i$  and  $\check{x}_i^n$ . That is, if  $v_{i+1/2} \geq 0$ , then  $x_{i^*}$  is the closest grid point greater than or equal to  $\check{x}_i^n$ , and if  $v_{i+1/2} < 0$ , then  $x_{i^*}$  is the closest grid point less than or equal to  $\check{x}_i^n$ . Since we keep the same tracing speed  $v_{i+1/2}$  for all indices, we have equal spacing ratio  $k_i < 1$  as defined in

$$|\check{x}_{i,\ell} - x_{i^*+\ell}| = \Delta x k_i < \Delta x.$$

Expand  $\bar{\mathcal{H}}_{i,\ell}$  in (13) into three terms, respectively, as

$$\begin{aligned} \bar{\mathcal{H}}_{i,\ell} &= \begin{cases} \int_{x_{i,\ell}^n}^{x_{i^*+\ell}} u(x, t^n) dx + \int_{x_{i^*+\ell}}^{x_{i^*+\ell}} u(x, t^n) dx - \int_{E_{i,\ell}} g_i(u) v_{E_{i,\ell},x} ds, & \text{if } v_{i+1/2} \geq 0, \\ - \int_{x_{i^*+\ell}}^{\check{x}_{i,\ell}^n} u(x, t^n) dx - \int_{x_{i^*+\ell}}^{x_{i^*+\ell}} u(x, t^n) dx + \int_{E_{i,\ell}} g_i(u) v_{E_{i,\ell},x} ds, & \text{if } v_{i+1/2} < 0, \end{cases} \\ &= T_{i,\ell}^1 + T_{i,\ell}^2 + T_{i,\ell}^3, \quad \ell = -1, 0, 1, 2. \end{aligned} \quad (14)$$

We next discuss approximation of the terms  $T_{i,\ell}^1$  and  $T_{i,\ell}^3$ .

## 2.2. Evaluation and reconstruction of $T_{i,\ell}^1$

Unlike as is done in [1], we use a simple WENO process to reconstruct  $T_{i,\ell}^1$  for the first term in (14), i.e.,

$$T_{i,\ell}^1 = \pm \int_{\check{x}_{i,\ell}^n}^{x_{i^*+\ell}} u(x, t^n) dx.$$

Three values of  $T_{i,\ell}^1$  will be needed later when we reconstruct  $\bar{\mathcal{H}}_{i+1/2}$  in (44). When  $v_{i+1/2} \geq 0$ , these are  $\ell = -1, 0, 1$ , and we reconstruct from  $u_{i^*-1}^n, u_{i^*}^n$ , and  $u_{i^*+1}^n$ . When  $v_{i+1/2} < 0$ , these are  $\ell = 0, 1, 2$ , and we reconstruct from  $u_{i^*}^n, u_{i^*+1}^n$ , and  $u_{i^*+2}^n$ .

To continue the exposition, we assume that  $v_{i+1/2} \geq 0$ . Let  $p_{i,L}(x)$  be the linear polynomial that interpolates the pointwise data  $\{u_{i^*-1}^n, u_{i^*}^n\}$  at  $\{x_{i^*-1}, x_{i^*}\}$ , respectively. Similarly,  $p_{i,R}(x)$  is the linear polynomial interpolating  $\{u_{i^*}^n, u_{i^*+1}^n\}$  at  $\{x_{i^*}, x_{i^*+1}\}$ . For each  $\ell = -1, 0, 1$ , we seek the linear weights  $\gamma_{i,\ell,L}$  and  $\gamma_{i,\ell,R}$  so that

$$\gamma_{i,\ell,L} \int_{\check{x}_{i,\ell}^n}^{x_{i^*+\ell}} p_{i,L}(x) dx + \gamma_{i,\ell,R} \int_{\check{x}_{i,\ell}^n}^{x_{i^*+\ell}} p_{i,R}(x) dx = \int_{\check{x}_{i,\ell}^n}^{x_{i^*+\ell}} Q_i(x) dx \approx T_{i,\ell}^1, \quad (15)$$

where  $Q_i(x)$  is the quadratic polynomial interpolating  $\{u_{i^*-1}^n, u_{i^*}^n, u_{i^*+1}^n\}$  at  $\{x_{i^*-1}, x_{i^*}, x_{i^*+1}\}$ . We see below that these linear weights exist, because we use a locally frozen trace velocity  $v_{i+1/2}$ . This means that the traceback points remain a distance  $\Delta x$  apart from each other, and no confusion arises due to tracelines tracking to irregular positions that lie in the same grid cell or that skip a cell. In fact, this way we have that the offset ratio  $k_i = (x_{i^*+\ell} - \check{x}_{i,\ell}^n)/\Delta x$  is constant for each fixed index  $i$ .

We compute that when  $\ell = -1$ ,

$$\int_{\check{x}_{i,-1}^n}^{x_{i^*-1}} p_{i,L}(x) dx = \frac{\Delta x k_i}{2} [(2 + k_i)u_{i^*-1}^n - k_i u_{i^*}^n], \quad (16)$$

$$\int_{\check{x}_{i,-1}^n}^{x_{i^*-1}} p_{i,R}(x) dx = \frac{\Delta x k_i}{2} [(4 + k_i)u_{i^*}^n - (2 + k_i)u_{i^*+1}^n], \quad (17)$$

$$\int_{\check{x}_{i,-1}^n}^{x_{i^*-1}} Q_i(x) dx = \frac{\Delta x k_i}{12} [(12 + 9k_i + 2k_i^2)u_{i^*-1}^n - (12k_i + 4k_i^2)u_{i^*}^n + (3k_i + 2k_i^2)u_{i^*+1}^n].$$

Therefore, we see that the linear weights exist and are in fact

$$\gamma_{i,-1,L}^+ = \frac{12 + 9k_i + 2k_i^2}{12 + 6k_i} \quad \text{and} \quad \gamma_{i,-1,R}^+ = -\frac{3k_i + 2k_i^2}{12 + 6k_i}. \quad (18)$$

When  $\ell = 0$ , we compute

$$\int_{\tilde{x}_i^n}^{x_i^{*s}} p_{i,L}(x) dx = \frac{\Delta x k_i}{2} [k_i u_{i-1}^n + (2 - k_i) u_i^n], \quad (19)$$

$$\int_{\tilde{x}_i^n}^{x_i^{*s}} p_{i,R}(x) dx = \frac{\Delta x k_i}{2} [(2 + k_i) u_i^n - k_i u_{i+1}^n], \quad (20)$$

$$\int_{\tilde{x}_i^n}^{x_i^{*s}} Q_{i^*}(x) dx = \frac{\Delta x k_i}{12} [(3k_i + 2k_i^2) u_{i-1}^n + (12 - 4k_i^2) u_i^n + (-3k_i + 2k_i^2) u_{i+1}^n],$$

and determine that

$$\gamma_{i,0,L}^+ = \frac{3 + 2k_i}{6} \quad \text{and} \quad \gamma_{i,0,R}^+ = \frac{3 - 2k_i}{6}. \quad (21)$$

Finally, when  $\ell = 1$ ,

$$\int_{\tilde{x}_{i,1}^n}^{x_{i+1}^{*s+1}} p_{i,L}(x) dx = \frac{\Delta x k_i}{2} [(k_i - 2) u_{i-1}^n + (4 - k_i) u_i^n], \quad (22)$$

$$\int_{\tilde{x}_{i,1}^n}^{x_{i+1}^{*s+1}} p_{i,R}(x) dx = \frac{\Delta x k_i}{2} [k_i u_i^n + (2 - k_i) u_{i+1}^n], \quad (23)$$

$$\int_{\tilde{x}_{i,1}^n}^{x_{i+1}^{*s+1}} Q_{i^*}(x) dx = \frac{\Delta x k_i}{12} [(-3k_i + 2k_i^2) u_{i-1}^n + (12k_i - 4k_i^2) u_i^n + (12 - 9k_i + 2k_i^2) u_{i+1}^n],$$

and so

$$\gamma_{i,1,L}^+ = \frac{3k_i - 2k_i^2}{12 - 6k_i} \quad \text{and} \quad \gamma_{i,1,R}^+ = \frac{12 - 9k_i + 2k_i^2}{12 - 6k_i}. \quad (24)$$

We define  $T_{i,\ell}^1$  by (15) using (16)–(24), but replacing the linear weights by their nonlinear counterparts defined using the usual smoothness indicator. For a uniform grid, this indicator is defined when  $v_{i+1/2} \geq 0$  for the values  $\{u_{i-1}^n, u_i^n, u_{i+1}^n\}$  and when  $v_{i+1/2} < 0$  for the values  $\{u_i^n, u_{i+1}^n, u_{i+2}^n\}$  as

$$\begin{aligned} IS_{i,L}^+(u^n) &= (u_i^n - u_{i-1}^n)^2, & IS_{i,R}^+(u^n) &= IS_{i,L}^-(u^n) = (u_{i+1}^n - u_i^n)^2, \\ IS_{i,R}^-(u^n) &= (u_{i+2}^n - u_{i+1}^n)^2, \end{aligned} \quad (25)$$

and then nonlinear weights are defined for either sign of  $v_{i+1/2}$  as

$$\tilde{\gamma}_{i,\ell,L}^\pm(\gamma_{i,\ell,L}^\pm, u^n) = \frac{\gamma_{i,\ell,L}^\pm}{(\epsilon + IS_{i,L}^\pm(u^n))^2} \quad \text{and} \quad \tilde{\gamma}_{i,\ell,R}^\pm = 1 - \tilde{\gamma}_{i,\ell,L}^\pm, \quad (26)$$

$$\frac{\gamma_{i,\ell,L}^\pm}{(\epsilon + IS_{i,L}^\pm(u^n))^2} + \frac{\gamma_{i,\ell,R}^\pm}{(\epsilon + IS_{i,R}^\pm(u^n))^2}$$

where  $\epsilon > 0$  is some small parameter. Finally, for  $v_{i+1/2} \geq 0$ , we have

$$T_{i-1}^1 = \frac{\Delta x k_i}{2} \{\tilde{\gamma}_{i-1,L}^+ [(2 + k_i) u_{i-1}^n - k_i u_i^n] + \tilde{\gamma}_{i-1,R}^+ [(4 + k_i) u_i^n - (2 + k_i) u_{i+1}^n]\}, \quad (27)$$

$$T_{i,0}^1 = \frac{\Delta x k_i}{2} \{\tilde{\gamma}_{i,0,L}^+ [k_i u_{i-1}^n + (2 - k_i) u_i^n] + \tilde{\gamma}_{i,0,R}^+ [(2 + k_i) u_i^n - k_i u_{i+1}^n]\}, \quad (28)$$

$$T_{i,1}^1 = \frac{\Delta x k_i}{2} \{\tilde{\gamma}_{i,1,L}^+ [(k_i - 2) u_{i-1}^n + (4 - k_i) u_i^n] + \tilde{\gamma}_{i,1,R}^+ [k_i u_i^n + (2 - k_i) u_{i+1}^n]\}. \quad (29)$$

Each reconstruction involves two linear weights which could be negative for some values of  $k_i$ . In this case, we use the usual techniques for treating negative weights in WENO schemes [15].

When  $v_{i+1/2} < 0$ , we need to reconstruct  $T_{i,\ell}^1$  for  $\ell = 0, 1, 2$  from  $u_{i^*}^n$ ,  $u_{i^*+1}^n$ , and  $u_{i^*+2}^n$ . These can be determined by symmetry from the case when  $v_{i+1/2} \geq 0$ . We need to account for the change in sign, exchange  $L$  and  $R$ , and we need to identify indices  $i-1$ ,  $i$ , and  $i+1$  with  $i+2$ ,  $i+1$ , and  $i$ , respectively. The linear weights are

$$\gamma_{i,0,L}^- = \frac{12 - 9k_i + 2k_i^2}{12 - 6k_i} \quad \text{and} \quad \gamma_{i,0,R}^- = \frac{3k_i - 2k_i^2}{12 - 6k_i}, \quad (30)$$

$$\gamma_{i,1,L}^- = \frac{3 - 2k_i}{6} \quad \text{and} \quad \gamma_{i,1,R}^- = \frac{3 + 2k_i}{6}, \quad (31)$$

$$\gamma_{i,2,L}^- = -\frac{3k_i + 2k_i^2}{12 + 6k_i} \quad \text{and} \quad \gamma_{i,2,R}^- = \frac{12 + 9k_i + 2k_i^2}{12 + 6k_i}, \quad (32)$$

and the values of  $T_{i,\ell}^1$ , for  $v_{i+1/2} < 0$ , are

$$T_{i,0}^1 = \frac{\Delta x k_i}{2} \{ \tilde{\gamma}_{i,0,L}^- [(k_i - 2)u_{i^*}^n - k_i u_{i^*+1}^n] + \tilde{\gamma}_{i,0,R}^- [(k_i - 4)u_{i^*+1}^n + (2 - k_i)u_{i^*+2}^n] \}, \quad (33)$$

$$T_{i,1}^1 = \frac{\Delta x k_i}{2} \{ \tilde{\gamma}_{i,1,L}^- [k_i u_{i^*}^n - (2 + k_i)u_{i^*+1}^n] + \tilde{\gamma}_{i,1,R}^- [(k_i - 2)u_{i^*+1}^n - k_i u_{i^*+2}^n] \}, \quad (34)$$

$$T_{i,2}^1 = \frac{\Delta x k_i}{2} \{ \tilde{\gamma}_{i,2,L}^- [(2 + k_i)u_{i^*}^n - (4 + k_i)u_{i^*+1}^n] + \tilde{\gamma}_{i,2,R}^- [k_i u_{i^*+1}^n - (2 + k_i)u_{i^*+2}^n] \}, \quad (35)$$

using (25)–(26).

### 2.3. Evaluation of $T_{i,\ell}^3$ , the flux correction step

We now turn our attention to computing the third term of (14),

$$T_{i,\ell}^3 = \mp \int_{E_{i,\ell}} g_i(u) v_{E_{i,\ell},x} ds, \quad \ell = -1, 0, 1, 2,$$

which is the term in our scheme that corrects the mass flux when  $g_i(u) = f(u) - v_{i+1/2} u \neq 0$ . After changing variables (see Fig. 1), we have

$$T_{i,\ell}^3 = \int_{t^n}^{t^{n+1}} g_i(u(\tilde{x}_{i,\ell}(t), t)) dt. \quad (36)$$

Applying a two-point Gauss quadrature rule gives

$$T_{i,\ell}^3 = \Delta t [w_{G,L} g_i(u(\tilde{x}_{i,\ell}(t_{G,L}), t_{G,L})) + w_{G,R} g_i(u(\tilde{x}_{i,\ell}(t_{G,R}), t_{G,R}))], \quad (37)$$

where  $w_{G,k}$  are the weights and  $t_{G,k}$  are the Gauss points,  $k = L, R$ . The main task is to find values for  $u(\tilde{x}_{i,\ell}(t), t)$  at the Gauss points along the traceline. We adapt an algorithm given originally by Levy, Puppo, and Russo [10] for Eulerian CWENO methods and modified for the Lagrangian setting by two of the authors in [6]. The algorithm uses a Runge-Kutta method combined with its natural continuous extension for time evolution of the differential equation along the traceline curve. We describe the algorithm below; more details can be found in [10, 6]. We remark that a more traditional flux correction procedure could be used to compute  $T_{i,\ell}^3$ , but this would require flux splitting. Our choice avoids the flux splitting, since the semi-Lagrangian nature of our overall scheme is already upstream biased.

In general, given the Cauchy problem

$$u' = F(t, u(t)) \quad \text{and} \quad u(t_0) = u_0, \quad (38)$$

the solution at time step  $n+1$  can be obtained by a  $\nu$ -stage Runge-Kutta method

$$u^{n+1} = u^n + \Delta t \sum_{i=1}^{\nu} b_i g^{(i)},$$

where each  $g^{(i)}$  is a Runge-Kutta flux

$$g^{(i)} = F\left(t^n + \Delta t c_i, u^n + \Delta t \sum_{j=1}^{\nu} a_{ij} g^{(j)}\right)$$

and  $c_i = \sum_j a_{ij}$ . The method is completely determined by the vector  $b$  and the lower triangular matrix  $a$ , which has a zero diagonal for explicit methods. For our third order SL-WENO scheme, we use the two-stage method

$$a = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}.$$

After all  $g^{(i)}$  are obtained, the Natural Continuous Extension (NCE) [16] of the Runge-Kutta method provides a uniform accuracy of the solution in the time interval  $[t^n, t^{n+1}]$ . Each  $\nu$ -stage Runge-Kutta method of order  $p$  has an NCE of degree  $d \leq p$  in the sense that there exist  $\nu$  polynomials  $b_i(\theta)$ ,  $i = 1, \dots, \nu$ , of degree at most  $d$ , such that

$$u(t^n + \theta \Delta t) \approx u^{n+\theta} = u^n + \Delta t \sum_{i=1}^{\nu} b_i(\theta) g^{(i)}, \quad 0 \leq \theta \leq 1, \quad (39)$$

satisfies

$$\max_{t^n \leq t \leq t^n + \Delta t} \left| \frac{d^k}{dt^k} [u^{n+(t-t^n)/\Delta t} - u(t)] \right| = \mathcal{O}((\Delta t)^{d+1-k}), \quad 0 \leq k \leq d.$$

In our case,  $b_1(\theta) = (b_1 - 1)\theta^2 + \theta$  and  $b_2(\theta) = b_2\theta^2$ . In particular, we can evaluate the solution at the Gauss points  $t_{G,k}$  using  $\theta_{G,k} = (t_{G,k} - t^n)/\Delta t$ .

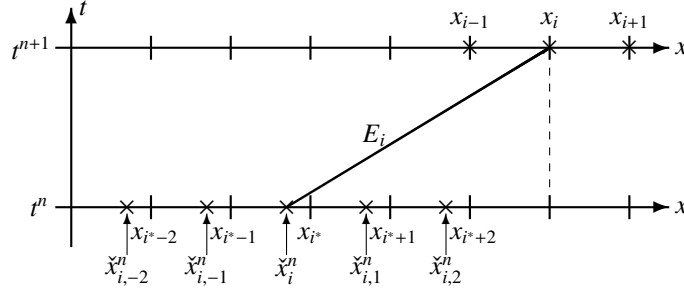


Figure 3: An illustration of the evaluation of the Runge-Kutta fluxes  $g^{(1)}$  and  $g^{(2)}$  from evolution of (41) when  $v_{i+1/2} \geq 0$ . To reconstruct the derivatives appearing in  $g^{(2)}$  at  $(x_i, t^{n+1})$ , we need function values at  $x_{i-1}$ ,  $x_i$ , and  $x_{i+1}$  at time  $t^{n+1}$ . These function values are computed using three values of  $g^{(1)}$ , which requires us to reconstruct derivatives at the points  $\check{x}_{i-1}^n$ ,  $\check{x}_i^n$ , and  $\check{x}_{i+1}^n$  at time  $t^n$ , and so we need to reconstruct point values at the five points  $\check{x}_{i,\ell}^n$ ,  $\ell = -2, -1, 0, 1, 2$ . Moreover, we in fact need to reconstruct  $g^{(2)}$  at the three points  $x_{i-1}$ ,  $x_i$ , and  $x_{i+1}$ , and so ultimately we need to reconstruct  $g^{(1)}$  at the five points  $\check{x}_{i,\ell}^n$ ,  $\ell = -2, -1, 0, 1, 2$ , and therefore we require point values reconstructed at the seven points  $\check{x}_{i,\ell}^n$ ,  $\ell = -3, -2, \dots, 3$ .

In our setting, that is, to evaluate (37), we need to obtain values of the function  $g_i$  along the traceline at the Gauss points. We solve Cauchy problems based on the original equation (1), but modified to evolve along the traceline  $\check{x}_i(t)$  defined by  $v_{i+1/2}$  in (9). That is, at gridpoint  $x_i$ , we have that

$$u_t + v_{i+1/2} u_x = v_{i+1/2} u_x - (f(u))_x. \quad (40)$$

Let  $\check{u}_i(t) = u(\check{x}_i(t), t)$ . Then along the traceline, this is simply

$$\frac{d\check{u}_i}{dt} = v_{i+1/2} \check{u}_{i,x} - (f(\check{u}_i))_x = F(\check{u}_i). \quad (41)$$

We remark that the right-hand side reduces to  $F(\check{u}_i) = -(g_i(\check{u}_i(t)))_x$  if  $v_{i+1/2}$  is taken to be a constant in space. Thus the computation of the  $i$ th Runge-Kutta flux  $g^{(i)}$  requires the evaluation of the  $x$ -derivatives of  $f(\check{u}_i)$  and  $\check{u}_i$  (or just of



$g_i(\check{u}_i(t))$  when  $v_{i+1/2}$  is constant). We are using a two-stage method, so we need the two Runge-Kutta fluxes  $g^{(1)}$  and  $g^{(2)}$ .

To continue, suppose that  $v_{i+1/2} \geq 0$ , so that  $\ell = -1, 0, 1$  rather than  $\ell = 0, 1, 2$ . Let  $\check{u}_{i,\ell}(t) = u(\check{x}_{i,\ell}(t), t)$ . We use standard WENO schemes to reconstruct values and derivatives of  $\check{u}_{i,\ell}^n$  and  $f(\check{u}_{i,\ell}^n)$  at time  $t^n$  to the appropriate order, where  $\ell = -2, -1, 0, 1, 2$ . As illustrated in Fig. 3, to obtain  $g^{(2)}$  at  $x_i$  requires  $g^{(1)}$  at  $x_{i-1}$ ,  $x_i$ , and  $x_{i+1}$ , since we need derivatives. The single value  $g^{(1)}$  at  $x_{i+\ell}$  requires reconstruction at  $\check{x}_{i,\ell-1}$ ,  $\check{x}_{i,\ell}$ , and  $\check{x}_{i,\ell+1}$ , again because we need derivatives.

In fact, for this WENO3 scheme, we have to compute  $T_{i,\ell}^3$ ,  $\ell = -1, 0, 1$ . Because we use a locally frozen velocity, we can reuse our reconstructions for  $\ell = 0$ . Thus we only require reconstructions of  $\check{u}_{i,\ell}^n$  and  $f(\check{u}_{i,\ell}^n)$  at time  $t^n$  to the appropriate order for  $\ell = -3, -2, \dots, 3$ . In the end, we evolve (41) three times to obtain the needed values of  $\check{u}_{i,\ell}(t_{G,k})$  in (37) to compute  $T_{i,\ell}^3$ ,  $\ell = -1, 0, 1$ .

For each traceback point, we could in fact use standard WENO to reconstruct two values for  $T_{i,\ell}^3$  from stencils biased either to the left or to the right of the point. The numerical results showed only an insignificant difference between the left and right choices for our SL-WENO3 scheme. The SL-WENO5 numerical results that we present later simply use the right reconstructed value.

The requirement of a flux correction step imposes a relaxed CFL constraint on the time step. The constraint was identified by Stockie, Mackenzie, and Russell [12, (2.7)] and also in [6]. The relaxed CFL condition is

$$\Delta t \leq \Delta t_{\text{CFL}} := \frac{\Delta x}{\max \left| \frac{\partial f(u)}{\partial u} - v_{i+1/2} \right|}. \quad (42)$$

Our choice of  $v_{i+1/2}$  later in (48) renders this constraint relatively mild compared to a purely Eulerian scheme that has  $v_{i+1/2} = 0$ .

#### 2.4. Reconstruction of the flux function $\mathcal{H}_{i+1/2}$

We determine  $\mathcal{H}_{i+1/2}$  using a third order WENO reconstruction of  $\bar{\mathcal{H}}_{i-1}$ ,  $\bar{\mathcal{H}}_i = \bar{\mathcal{H}}_{i,0}$ , and  $\bar{\mathcal{H}}_{i,1}$  when  $v_{i+1/2} \geq 0$  and  $\bar{\mathcal{H}}_i$ ,  $\bar{\mathcal{H}}_{i,1}$ , and  $\bar{\mathcal{H}}_{i,2}$  when  $v_{i+1/2} < 0$ . If  $\mathcal{R}_{i+1/2}$  represents this linear reconstruction operator for index  $i$ , then

$$\begin{aligned} \mathcal{H}_{i+1/2} &= \mathcal{R}_{i+1/2}(\bar{\mathcal{H}}_{i-1}, \bar{\mathcal{H}}_i, \bar{\mathcal{H}}_{i,1}, \bar{\mathcal{H}}_{i,2}) = \mathcal{R}_{i+1/2}(\bar{\mathcal{H}}_{i,\ell}) \\ &= \mathcal{R}_{i+1/2}(T_{i,\ell}^1 + T_{i,\ell}^2 + T_{i,\ell}^3) = \mathcal{R}_{i+1/2}(T_{i,\ell}^1 + T_{i,\ell}^3) + \mathcal{R}_{i+1/2}(T_{i,\ell}^2), \end{aligned} \quad (43)$$

abusing the notation here and below by suppressing the argument list that should include arguments for indices  $\ell = -1, 0, 1$ , and 2 (possibly omitting  $-1$  or 2). Following the standard WENO reconstruction for average values,  $\mathcal{R}_{i+1/2}(T_{i,\ell})$  is defined with linear weights as

$$\mathcal{R}_{i+1/2}(T_{i,\ell}) = \begin{cases} r_L^+ \frac{-T_{i,-1} + 3T_{i,0}}{2} + r_R^+ \frac{T_{i,0} + T_{i,1}}{2}, & \text{if } v_{i+1/2} \geq 0, \\ r_L^- \frac{T_{i,0} + T_{i,1}}{2} + r_R^- \frac{3T_{i,1} - T_{i,2}}{2}, & \text{if } v_{i+1/2} < 0, \end{cases} \quad (44)$$

where the linear weights are  $r_L^+ = r_R^+ = 1/3$  and  $r_R^- = r_L^- = 2/3$ .

From (14), we see that the second term

$$T_{i,\ell}^2 = \int_{x_{i^*+\ell}}^{x_{i+\ell}} u(x, t^n) dx, \quad \ell = -1, 0, 1, 2,$$

is an integral over a complete set of grid cells. From [1, Proposition 3.7 and the discussion in §3.3.3], the reconstruction is simply

$$\mathcal{R}_{i+1/2}(T_{i,\ell}^2) = \begin{cases} \sum_{k=i^*+1}^i u_k^n \Delta x, & v_{i+1/2} \geq 0, \\ -\sum_{k=i+1}^{i^*-1} u_k^n \Delta x, & v_{i+1/2} < 0. \end{cases} \quad (45)$$

Unlike as is done in [1], we use a two-stage WENO process. Stage one gave us  $T_{i,\ell}^1$  and  $T_{i,\ell}^3$ , and now stage two is  $\mathcal{R}(T_{i,\ell})$ ,  $T_{i,\ell} = T_{i,\ell}^1 + T_{i,\ell}^3$ . Of course, to avoid discontinuities within the WENO framework, we modify the linear weights  $r_L^\pm$  and  $r_R^\pm$  in (44) with a smoothness indicator. Normally, one would base the smoothness indicator on  $T_{i,\ell}$ . However, we computed  $T_{i,\ell}^1$  from a WENO reconstruction that was already biased to the smoother stencil, so the  $T_{i,\ell}^1$  do not reflect the discontinuities of the underlying function  $u(x)$ . Therefore, we base the smoothness indicator directly on the values of  $u_{i\pm\ell}^n$  and use the previously defined indicator (25). So we define  $\mathcal{R}_{i+1/2}$  by replacing the linear weights in (44) by the nonlinear weights, analogous to (26).

Finally, then, the finite difference scheme for (1), or (7), is

$$u_i^{n+1} = u_i^n - \frac{1}{\Delta x} [\mathcal{H}_{i+1/2}^n - \mathcal{H}_{i-1/2}^n], \quad (46)$$

with

$$\mathcal{H}_{i+1/2}^n = \mathcal{R}(T_{i,\ell}^1 + T_{i,\ell}^3) + \mathcal{R}_{i+1/2}(T_{i,\ell}^2),$$

and  $T_{i,\ell}^1$  as defined in (27)–(29) or (33)–(35),  $T_{i,\ell}^3$  as evaluated in (37) as described in Section 2.3, and  $\mathcal{R}_{i+1/2}(T_{i,\ell}^2)$  as evaluated in (45).

A fifth order semi-Lagrangian finite difference WENO (SL-WENO5) scheme can be defined analogously to our third order SL-WENO3 scheme. We need  $\bar{\mathcal{H}}_{i,\ell}$ ,  $\ell = -2, -1, 0, 1, 2$ . A similar Runge-Kutta-NCE computation is needed. Such a fifth order computation is described in [6], and we refer the reader to that paper for more details. We include SL-WENO5 computations in the two sections on numerical results.

### 2.5. Determination of the upstream direction and selection of $v_{i+1/2}$

For the linear problem, where  $f(u) = a(x, t)u$ , no special treatment of the trace velocity  $v_{i+1/2}$  is required. It may be evaluated simply as  $v_{i+1/2}(x, t) = a(x, t)$ , possibly fixing  $t = t^n$  or  $t^{n+1}$  and  $x = x_i, x_{i+1}$ , or  $x_{i+1/2}$ . Assuming that  $a$  is smooth, all choices give essentially equivalent results. We choose  $v_{i+1/2} = a(x_i, t^n)$  for most of our linear problems.

We have observed numerically that the nonlinear problem requires a careful selection of  $v_{i+1/2}$  in regions where the characteristic wave speed changes direction and the velocity is not smooth, i.e., near shocks. Otherwise the scheme may exhibit instability or give an incorrect solution. The nonlinear weighting of the Eulerian WENO scheme provides good stabilization near shocks by biasing the finite difference stencil to one side of the shock where the solution remains smooth. To maintain this property in our semi-Lagrangian scheme, we need to know which direction the shock is moving, which is given by the Rankine-Hugoniot jump condition. In our case, we take the shock speed as computed using  $u_i^n$  and  $u_{i+1}^n$  (this speed is also called the Roe speed [11]), and so

$$s_{i+1/2} = \frac{f(u_i^n) - f(u_{i+1}^n)}{u_i^n - u_{i+1}^n}. \quad (47)$$

We then set

$$v_{i+1/2} = \begin{cases} \frac{\partial f}{\partial u}(u_i^n; x_i, t^n) & \text{if } s_{i+1/2} \geq 0, \\ \frac{\partial f}{\partial u}(u_{i+1}^n; x_{i+1}, t^n) & \text{if } s_{i+1/2} < 0. \end{cases} \quad (48)$$

The entropy condition implies that  $f'(u_i^n) \geq s_{i+1/2} \geq f'(u_{i+1}^n)$ , so the shock speed and the trace velocity have the same sign.

In regions where the solution is smooth, or a rarefaction forms, we use the same criterion, since it seems to matter little. We may as well also apply this criterion to linear problems instead of taking the simple condition  $v_{i+1/2} = a(x_i, t^n)$ .

We remark that we tested the scheme using the trace velocity taken as the approximate particle velocity  $v_{i+1/2} = f(u_i^n)/u_i^n$  or  $f(u_{i+1}^n)/u_{i+1}^n$ . The results are disappointing compared to using an approximate characteristic velocity.

The condition (48) is problematic when the Roe speed  $s_{i+1/2} = 0$ . We arbitrarily took one of the two cases, but in truth we should not bias to either direction. We therefore modify the scheme for some small  $\epsilon > 0$ , when  $|s_{i+1/2}| < \epsilon$ , by replacing the construction of the numerical flux  $\mathcal{H}_{i+1/2}$  in the scheme as follows. Compute  $\mathcal{H}_{i+1/2}^{n,-}$  and  $\mathcal{H}_{i+1/2}^{n,+}$  using  $v_{i+1/2}$  defined in (48) assuming  $s_{i+1/2} > 0$  and  $s_{i+1/2} < 0$ , respectively. Also compute  $u_{i+1/2}^-$  and  $u_{i+1/2}^+$  using standard

WENO to reconstruct the two values of  $u_{i+1/2}$  from stencils biased to the left and to the right of the point  $x_{i+1/2}$ . Then define  $\mathcal{H}_{i+1/2}^n$  using a type of Lax-Friedrichs flux, which in this case is

$$\mathcal{H}_{i+1/2}^n = \frac{1}{2}[\mathcal{H}_{i+1/2}^{n,-} + \mathcal{H}_{i+1/2}^{n,+} - \Delta t \alpha (u_{i+1/2}^+ - u_{i+1/2}^-)], \quad (49)$$

where  $\alpha = \max |\partial f / \partial u|$ . Note the presence of  $\Delta t$  above, which arises because  $\mathcal{H}_{i+1/2}^{n,\pm}$  is, more or less, an approximation to  $\int_{t^n}^{t^{n+1}} f(u) dt$ . We remark that a similar but more complex construction is used in the standard finite difference WENO scheme (WENO-FD) [17, pp. 24–25]; however, we cannot use the same construction here because  $f^\pm$  does *not* satisfy the hyperbolic equation  $u_t + f^\pm(u)_x = 0$ , where  $f(u) = f^+(u) + f^-(u)$ .

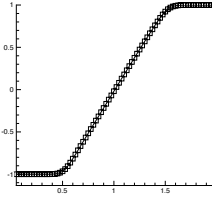


Figure 4: A Riemann test for the modified scheme using (49). The solution at time  $T = 0.5$  to Burgers' equation using the initial condition  $u^0(x) = u_L = -1$  when  $x < 1$  and  $u^0(x) = u_R = 1$  when  $x > 1$ . We use 80 grid cells in  $[0, 2]$  ( $\Delta x = 1/40$ ) and  $\Delta t = 0.3\Delta x$ . The entropy solution, a rarefaction, is obtained.

In fact, we require this modification only in the very rare cases for which  $|s_{i+1/2}| < \epsilon$ . For most problems we do not even bother to check for this condition. However, it can be important in some cases, such as the famous example of a Riemann problem for Burgers' equation using the initial condition  $u^0(x) = u_L = -1$  when  $x < 1$  and  $u^0(x) = u_R = 1$  when  $x > 1$ . The entropy solution is a rarefaction wave. The unmodified scheme does show slow convergence to a rarefaction wave, since the scheme is based on semi-Lagrangian principles. However, the modified scheme provides the correct solution even on coarser meshes, as shown in Fig. 4. In fact, the modification comes into play only for the first time step, since the wave direction becomes clear after a single step.

### 3. Some numerical results in one space dimension

We present several examples of our numerical scheme to test its accuracy and performance. Some of the examples are for the linear equation

$$u_t + (a(x, t)u)_x = 0, \quad (50)$$

for which we specify only the velocity  $a(x, t)$ . In that case, unless stated otherwise, we use the simple trace velocity  $v_{i+1/2} = a(x_i, t^n)$ .

#### 3.1. Example 1, linear velocity $a = 1$

We begin with the simple test case of linear translation with velocity  $a = 1$  and the initial condition  $u_0(x) = 0.75 + 0.25 \sin(\pi x)$  over  $[0, 2]$ . Since the traceback points could be found exactly, we perturb them randomly by an amount at most  $\pm 0.05\Delta x$ . We see from Table 1 that the solution converges with a clean third or fifth order for SL-WENO3 and SL-WENO5, respectively, when the error is measured in either the  $L^1$  or  $L^\infty$  norm.

The CPU time to compute  $T^3$  is also given in Table 1. Since this is a linear problem, we can apply the scheme of Qiu and Shu [1]. Because their scheme and ours are computationally similar, the CPU time we need to compute  $T^1$  and  $T^2$  is essentially the time needed to compute the scheme of Qiu and Shu. We also report the percentage of extra CPU time needed to compute  $T^3$  versus computing  $T^1$ ,  $T^2$ , and  $T^3$ . For SL-WENO3,  $T^3$  takes around 50% of the time, so our scheme is about twice as costly as the scheme of Qiu and Shu for the same  $\Delta x$  and  $\Delta t$ . For SL-WENO5, around 70% more time is needed for computing  $T^3$ , making the scheme about 3.3 times more costly. However, with the flux correction term  $T^3$ , our scheme may potentially allow a longer time step to be used for some problems, which results in a less numerically diffuse solution, so it is difficult to make a fair comparison of the computational effort. Moreover, our scheme handles nonlinear problems.

Table 1: Ex. 1, linear velocity  $a = 1$ . SL-WENO error and convergence order at  $T = 10$  with  $\Delta t = 10.5\Delta x$ . The traceback points are perturbed randomly by an amount at most  $\pm 0.05\Delta x$ . The CPU time needed to compute the term  $T^3$ , and the percentage of time spent computing  $T^3$  versus computing all three terms.

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order	CPU for $T^3$ (sec.)	CPU for $T^3$ (%)
SL-WENO3						
20	7.01366E-04	—	5.45853E-04	—	1.56E-4	31.6
40	8.79353E-05	2.99565	6.89518E-05	2.98485	4.23E-4	44.7
80	1.04822E-05	3.06849	8.22295E-06	3.06786	1.60E-3	44.3
160	1.30657E-06	3.00409	1.02901E-06	2.99840	5.83E-3	48.2
320	1.63175E-07	3.00130	1.28445E-07	3.00203	2.23E-2	48.4
640	2.04083E-08	2.99920	1.60473E-08	3.00075	8.80E-2	48.3
SL-WENO5						
20	1.11926E-04	—	1.10994E-04	—	1.21E-3	76.4
40	1.57164E-06	6.15413	1.43448E-06	6.27381	4.15E-3	78.9
80	3.43443E-08	5.51605	5.61897E-08	4.67408	1.53E-2	78.0
160	1.16414E-09	4.88274	1.25944E-09	5.47945	6.14E-2	75.0
320	3.41564E-11	5.09096	3.13533E-11	5.32803	2.46E-1	73.7
640	9.38473E-13	5.18570	9.00169E-13	5.12228	1.05E+0	68.8

The standard finite difference WENO scheme (WENO-FD) can handle nonlinear problems, and it is based more or less on a flux correction like ours (41). However, for each grid point  $x_i$ , WENO-FD requires two solves due to flux splitting, and we need to compute our approximation three times for SL-WENO3 and five times for SL-WENO5. Therefore, the computational cost for WENO3-FD is roughly  $(2/3)0.5$  or 33% of our SL-WENO3, and WENO5-FD is roughly  $(2/5)0.7$  or 28% of our SL-WENO5. To put it more simply, we need about three to three and a half times more CPU time than a WENO-FD scheme. But again, it is difficult to make a fair comparison of the computational effort versus the accuracy achieved.

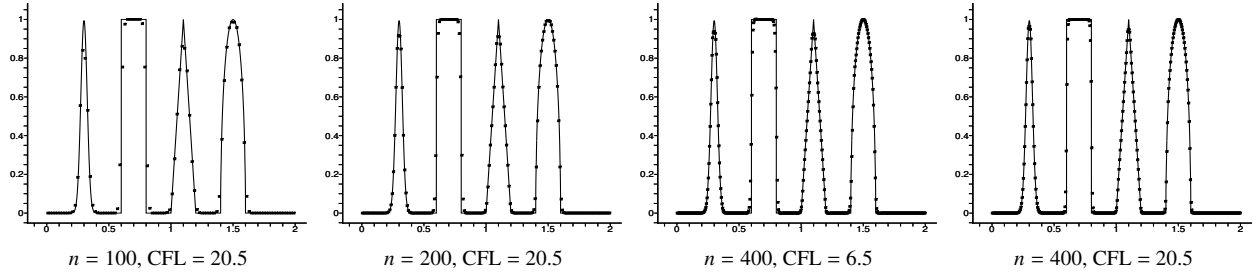


Figure 5: Ex. 1, Shu's linear test. The traceback points are perturbed randomly by an amount at most  $\pm 0.05\Delta x$ . Results are shown at time  $T = 2$  for SL-WENO3 for various spatial resolutions  $\Delta x = 2/n$  and CFL numbers. Results for SL-WENO5 are similar.

We also present the standard test case called *Shu's linear test*, which is simply the linear translate (with velocity  $a = 1$ ) of a complicated initial condition. Again, the traceback points are perturbed randomly by an amount at most  $\pm 0.05\Delta x$ . We see excellent results in Fig. 5 for SL-WENO3 at the final time  $T = 2$ . Results for SL-WENO5 are similar.

### 3.2. Example 2, linear velocity $a(x, t) = \sin(t)$

In the next example, we consider linear transport with  $a(x, t) = \sin(t)$  on  $[0, 2]$ , for which the exact solution is  $u(x, t) = u_0(x + 1 + \cos(t))$  and the initial condition is chosen to be  $u_0(x) = 0.75 + 0.25 \sin(\pi x)$ . Clean third order convergence is observed in Table 2 for  $\Delta t = 5.5\Delta x$ . If we increase the time step to  $\Delta t = 10.5\Delta x$ , then clean fourth order convergence is observed. Similar results are observed for SL-WENO5, as given in Table 3.

Table 2: Ex. 2, linear velocity  $a(x, t) = \sin(t)$ . SL-WENO3 errors and convergence order at  $T = 4$  with  $\Delta t = 5.5\Delta x$  and  $\Delta t = 10.5\Delta x$ , using RK4 for the trace-back points.

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = 5.5\Delta x$				
20	2.93569E-03	—	2.28050E-03	—
40	1.04654E-04	4.81000	8.21084E-05	4.79568
80	5.28225E-06	4.30833	4.14897E-06	4.30671
160	3.96333E-07	3.73637	3.11253E-07	3.73659
320	3.85977E-08	3.36012	3.03127E-08	3.36010
640	4.21385E-09	3.19530	3.30955E-09	3.19521
$\Delta t = 10.5\Delta x$				
20	4.08665E-02	—	3.20055E-02	—
40	2.00674E-03	4.34799	1.57052E-03	4.34901
80	8.23635E-05	4.60671	6.47211E-05	4.60086
160	3.08929E-06	4.73666	2.42631E-06	4.73740
320	1.53627E-07	4.32977	1.20657E-07	4.32977
640	9.51076E-09	4.01373	7.46968E-09	4.01373

Table 3: Ex. 2, linear velocity  $a(x, t) = \sin(t)$ . SL-WENO5 errors and convergence order at  $T = 4$  with  $\Delta t = 5.5\Delta x$  and  $\Delta t = 10.5\Delta x$ , using RK4 for the trace-back points.

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = 5.5\Delta x$				
20	6.90696E-05	—	6.38140E-05	—
40	1.32375E-06	5.70535	1.23738E-06	5.68851
80	2.64693E-08	5.64417	3.70303E-08	5.06244
160	1.17479E-09	4.49385	1.19085E-09	4.95864
320	3.25780E-11	5.17236	2.91145E-11	5.35411
640	9.30040E-13	5.13046	9.08273E-13	5.00247
$\Delta t = 10.5\Delta x$				
20	1.36122E-03	—	1.06236E-03	—
40	1.87142E-05	6.18462	1.47547E-05	6.16996
80	2.41995E-07	6.27302	1.89562E-07	6.28236
160	3.92645E-09	5.94561	3.10189E-09	5.93338
320	5.89586E-11	6.05738	4.75878E-11	6.02641
640	9.65577E-13	5.93217	9.76441E-13	5.60692

### 3.3. Example 3, linear velocity $a(x, t) = \sin(x)$

We test our scheme using  $a(x, t) = \sin(x)$  over  $[0, 2\pi]$  up to the final time  $T = 1$ . The exact solution is

$$u(x, t) = \frac{\sin(2 \arctan(e^{-t} \tan(x/2)))}{\sin(x)}.$$

We see from Table 4 that the error for SL-WENO3 as measured in  $L_{\Delta x}^1$  converges at the optimal rate. However, when measured in  $L_{\Delta x}^\infty$ , the error converges to one power less than optimal. Similar results are seen in Table 5 for SL-WENO5. If we apply the more complex choice of  $v_{i+1/2}$  using the Rankine-Hugoniot jump condition given in (48), we see very little change in the errors in Tables 6–7.

Our results indicate a degeneracy in the  $L^\infty$ -norm error. In a 1990 paper [18], Shu showed that an ENO scheme may experience a loss of one order of accuracy in the  $L^\infty$ -norm error when the stencil used by the scheme varies from one point to another. In particular, there is a problem when the stencil jumps over, and therefore omits, a point of the grid. The stencil jump is problematic due to a failure to cancel error in the conservative form (7). There appears to be no simple resolution of this problem.

Table 4: Ex. 3, linear velocity  $a(x, t) = \sin(x)$ . SL-WENO3 error and convergence order at  $T = 1$  using RK4 tracing.

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = 0.5\Delta x$				
20	3.24579E-01	—	2.92166E-01	—
40	8.55810E-02	1.92321	8.35633E-02	1.80585
80	8.11433E-03	3.39875	1.40641E-02	2.57085
160	4.04675E-04	4.32564	2.76850E-03	2.34484
320	2.96076E-05	3.77272	5.55736E-04	2.31663
640	3.72088E-06	2.99225	1.38315E-04	2.00644
$\Delta t = 6.5\Delta x$				
20	3.15283E-01	—	3.01127E-01	—
40	9.46767E-02	1.73557	9.15660E-02	1.71749
80	2.52199E-02	1.90845	5.89933E-02	0.63426
160	1.37485E-03	4.19722	3.68352E-03	4.00140
320	9.03128E-05	3.92820	6.01375E-04	2.61475
640	6.85537E-06	3.71962	1.38941E-04	2.11379
$\Delta t = 10.5\Delta x$				
20	3.15283E-01	—	3.01127E-01	—
40	9.46767E-02	1.73557	9.15660E-02	1.71749
80	2.51429E-02	1.91286	2.66881E-02	1.77861
160	1.63479E-03	3.94297	2.95283E-03	3.17603
320	1.07356E-04	3.92863	6.19259E-04	2.25348
640	8.43736E-06	3.66947	1.41818E-04	2.12651

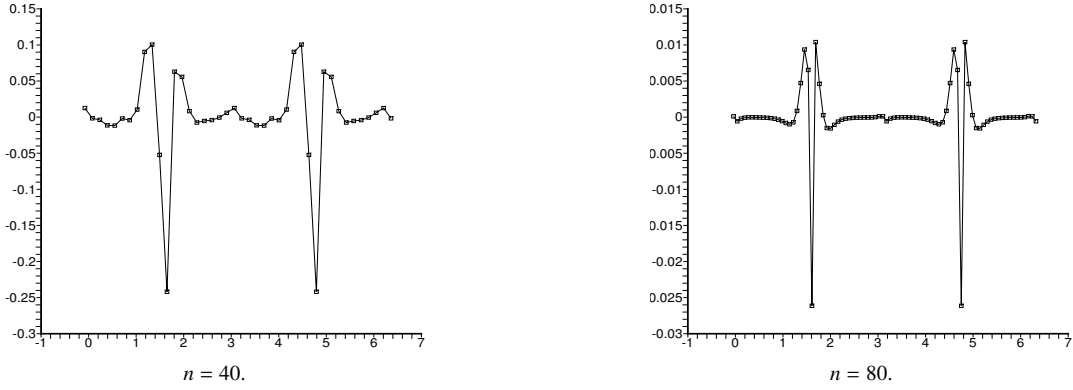


Figure 6: Ex. 3, linear velocity  $a(x, t) = \sin(2x)$ . The pointwise  $L_{\Delta x}^\infty$  error for two resolutions  $n = 40$  and  $n = 80$  at the time  $T = 1/2$ . Note that the error is clustered near  $x = \pi/2$  and  $x = 3\pi/2$  where the sign of the velocity  $a = \sin(2x)$  changes from positive to negative.

Our scheme may experience a stencil jump in this problem near  $x = \pi$  where the velocity changes from positive to negative. The positive stencil at  $x_i$  is  $\{u_{i-1}^*, u_i^*, u_{i+1}^*\}$  and the negative stencil at  $x_{i+1}$  is  $\{u_{(i+1)^*}, u_{(i+1)^*+1}, u_{(i+1)^*+2}\}$ . If  $x_{i^*+1} + \Delta x < x_{(i+1)^*}$ , then we omit data at  $x_{i^*+2}$ , for example.

To verify that stencil switching is the problem, we rescale the velocity to  $a(x) = \sin(2x)$  over  $[0, 2\pi]$  so that we cover two complete periods and experience a change in stencil at  $x = 0, \pi/2, \pi$ , and  $3\pi/2$ . Fig. 6 shows that the pointwise errors are indeed relatively large at the two points  $x = \pi/2$  and  $3\pi/2$  where the sign of  $a(x)$  changes from positive to negative. Moreover, if we use a fixed stencil, say the one for positive velocities  $\{u_{i-1}^*, u_i^*, u_{i+1}^*\}$  regardless the sign of  $a(x)$ , and we rely on the WENO nonlinear weights to pick up the correct stencil, we recover the lost order of accuracy in the  $L^\infty$ -norm. However, this modification breaks down in two space dimensional problems, and it cannot be expected to work for nonlinear problems.

Table 5: Ex. 3, linear velocity  $a(x, t) = \sin(x)$ . SL-WENO5 error and convergence order at  $T = 1$  using RK4 tracing.

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = 0.5\Delta x$				
20	4.41248E-02	—	5.45453E-02	—
40	3.60168E-03	3.61485	4.76705E-03	3.51629
80	1.23245E-04	4.86907	5.19132E-04	3.19892
160	2.98001E-06	5.37007	3.21900E-05	4.01142
320	6.04899E-08	5.62248	1.12337E-06	4.84071
640	1.59594E-09	5.24422	5.50376E-08	4.35127
1280	4.97604E-11	5.00326	3.40876E-09	4.01310
$\Delta t = 6.5\Delta x$				
20	4.41513E-02	—	4.98883E-02	—
40	5.19001E-03	3.08865	5.57751E-03	3.16101
80	1.96211E-04	4.72526	5.65309E-04	3.30251
160	6.41004E-06	4.93593	4.03784E-05	3.80738
320	2.00220E-07	5.00067	2.52046E-06	4.00182
640	5.63142E-09	5.15195	1.22039E-07	4.36827
1280	1.50732E-10	5.22344	4.15521E-09	4.87628
$\Delta t = 10.5\Delta x$				
20	4.41513E-02	—	4.98883E-02	—
40	5.19001E-03	3.08865	5.57751E-03	3.16101
80	1.05895E-03	2.29310	8.58960E-04	2.69896
160	2.87648E-05	5.20219	5.33115E-05	4.01007
320	9.68361E-07	4.89262	3.06933E-06	4.11845
640	3.04895E-08	4.98916	1.58211E-07	4.27801
1280	9.65502E-10	4.98089	5.53593E-09	4.83688

### 3.4. Example 4, Burgers' equation

In the next example we test Burgers' equation with a simple initial condition to evaluate the convergence rates of the schemes for a nonlinear problem; that is, for

$$u_t + (u^2/2)_x = 0 \quad \text{and} \quad u_0(x) = 0.5 + \sin(\pi x) \quad \text{for} \quad x \in (0, 2).$$

Shocks will form at time  $t = 1/\pi \approx 0.32$ . To test convergence, we ran the computation over gradually refined meshes up to time  $T = 0.25$ , before the shocks are fully developed. The numerical errors and convergence orders for the scheme are given in Tables 8–9. We see third and fifth order convergence in the  $L^1$ -norm for SL-WENO3 and SL-WENO5, respectively, as the mesh becomes fine enough for this nonlinear problem, as expected. The  $L^\infty$ -norm is not as well behaved, but the rates also appear to be optimal in this norm. We note that our scheme, despite being based on exact time integration, nevertheless has some time discretization error in the approximation of the flux correction term  $T^3$ . Thus, the error from  $T^3$  will eventually increase as  $\Delta t$  increases, as one sees in Tables 8–9, especially for the case  $\Delta t = 15\Delta x$ , although third and fifth order convergence is maintained.

Fig. 7 shows the solutions at  $T = 3/(2\pi) \approx 0.48$  after the shocks have formed. There is no numerical oscillation, and both schemes perform satisfactorily.

### 3.5. Example 5, Buckley-Leverett equation

The next example for scalar conservation law (1) uses the Buckley-Leverett flux function

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

Table 6: Ex. 3, linear velocity  $a(x, t) = \sin(x)$ . SL-WENO3 error and convergence order at  $T = 1$  using RK4 tracing. Here we use the more complex choice of  $v_{i+1/2}$  using the Rankine-Hugoniot jump condition given in (48).

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = 0.5\Delta x$				
20	1.24992E-01	—	8.68310E-02	—
40	2.28395E-02	2.45223	3.09329E-02	1.48907
80	3.90462E-03	2.54828	8.64846E-03	1.83863
160	5.61515E-04	2.79779	2.09878E-03	2.04289
320	7.70512E-05	2.86543	3.96888E-04	2.40275
640	9.47769E-06	3.02321	7.55154E-05	2.39389
$\Delta t = 6.5\Delta x$				
20	9.66143E-02	—	8.22429E-02	—
40	5.21595E-02	0.88931	4.41931E-02	0.89607
80	5.02481E-03	3.37579	8.69606E-03	2.34539
160	6.53692E-04	2.94239	2.41547E-03	1.84806
320	7.63623E-05	3.09768	4.27393E-04	2.49867
640	8.88052E-06	3.10414	6.86186E-05	2.63889
$\Delta t = 10.5\Delta x$				
20	9.66143E-02	—	8.22429E-02	—
40	5.21595E-02	0.88931	4.41931E-02	0.89607
80	1.50708E-02	1.79117	1.25268E-02	1.81881
160	1.01809E-03	3.88782	3.24624E-03	1.94818
320	8.95787E-05	3.50656	5.68740E-04	2.51293
640	9.23741E-06	3.27760	5.75340E-05	3.30528

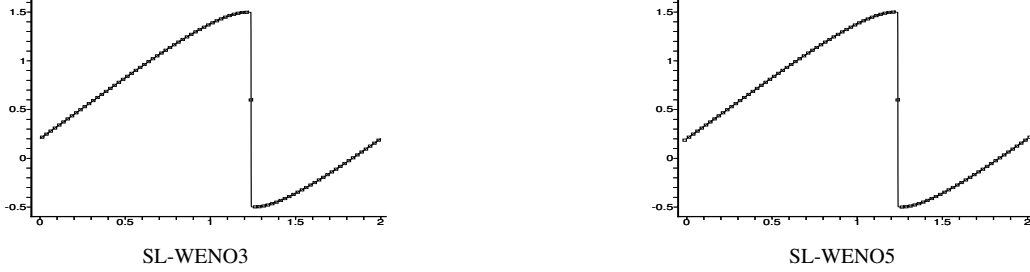


Figure 7: Ex. 4, Burgers' equation with a shock. The solution at time  $T = 3/(2\pi)$  of the SL-WENO schemes using  $m = 80$  grid elements and  $\Delta t = 2\Delta x = 4/m$ .

and involves the interaction of shocks and rarefactions. The initial condition is

$$u_0(x) = \begin{cases} 1 - 20x & \text{for } 0 \leq x \leq 0.05, \\ 0.5 & \text{for } 0.25 \leq x \leq 0.4, \\ 0 & \text{otherwise,} \end{cases} \quad (51)$$

so the problem has two pulses that merge over time. We use  $m = 80$  grid elements. The results for the third and fifth order schemes are shown in Fig. 8 and Fig. 9, respectively. The schemes handle the merging of the two pulses quite well and reproduce the solution to adequate accuracy even on a relatively low resolution grid. In fact, the SL-WENO results are somewhat better than those of the Eulerian finite difference WENO scheme (WENO-FD). In particular, WENO5-FD shows more numerical diffusion than SL-WENO in this test.



Table 7: Ex. 3, linear velocity  $a(x, t) = \sin(x)$ . SL-WENO5 error and convergence order at  $T = 1$  using RK4 tracing. Here we use the more complex choice of  $v_{i+1/2}$  using the Rankine-Hugoniot jump condition given in (48).

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = 0.5\Delta x$				
20	1.07696E-02	—	8.27402E-03	—
40	2.25629E-03	2.25493	4.35033E-03	0.92746
80	1.09633E-04	4.36320	4.99510E-04	3.12254
160	4.33732E-06	4.65973	3.15522E-05	3.98470
320	1.52751E-07	4.82755	1.49442E-06	4.40009
640	4.84909E-09	4.97732	4.69414E-08	4.99258
1280	1.37993E-10	5.13505	1.91373E-09	4.61640
$\Delta t = 6.5\Delta x$				
20	2.89233E-02	—	2.72218E-02	—
40	4.50924E-03	2.68127	3.86699E-03	2.81548
80	1.84267E-04	4.61301	3.02948E-04	3.67407
160	7.64841E-06	4.59049	2.07271E-05	3.86948
320	2.87272E-07	4.73467	9.93562E-07	4.38276
640	1.01709E-08	4.81989	7.85410E-08	3.66109
1280	3.51343E-10	4.85543	5.94584E-09	3.72349
$\Delta t = 10.5\Delta x$				
20	2.89233E-02	—	2.72218E-02	—
40	4.50924E-03	2.68127	3.86699E-03	2.81548
80	1.10704E-03	2.02618	8.14590E-04	2.24707
160	4.25166E-05	4.70254	2.18153E-04	1.90073
320	1.62855E-06	4.70636	1.28226E-05	4.08858
640	5.76343E-08	4.82052	7.08507E-07	4.17776
1280	2.02261E-09	4.83263	4.92683E-08	3.84605

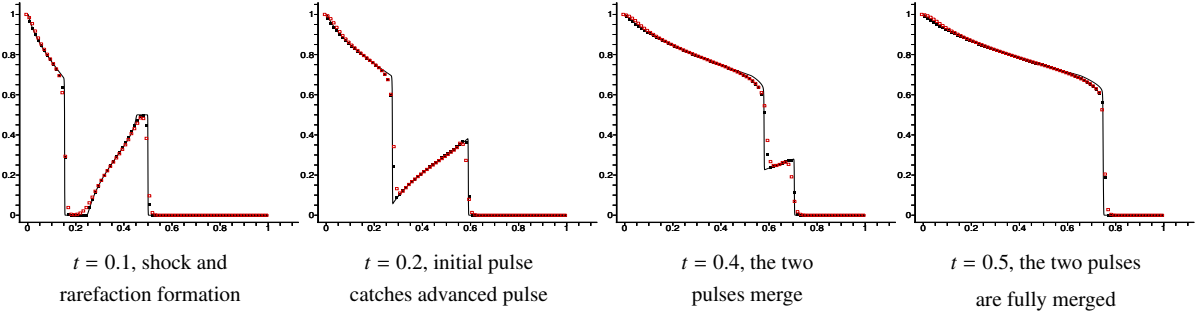


Figure 8: Ex. 5, Buckley-Leverett, WENO3. An interaction of shocks and rarefactions, resulting from the evolution of the initial condition of two pulses given in (51). The solid line is the reference solution, given by CWENO5 with a very small  $\Delta x = 1/1280$  and  $\Delta t = 1/15360$ . The filled black squares are our SL-WENO3 results, and the open red squares are WENO3-FD results, both using  $m = 80$  and  $\Delta t = 0.4\Delta x$ .

#### 4. Some numerical results in two space dimensions

There are standard ways to apply our finite difference scheme to two-space dimensional problems. For

$$u_t + (f(u))_x + (g(u))_y = 0, \quad x, y \in \mathbb{R}, \quad t > 0, \quad (52)$$

we might simply compute the analogue of (7), which involves a sliding average function in the  $y$ -variable  $\mathcal{K}$ . The conservative scheme is based on

$$u_{ij}^{n+1} = u_{ij}^n - \frac{1}{\Delta x} [\mathcal{H}(x_{i+1/2, j}) - \mathcal{H}(x_{i-1/2, j})] - \frac{1}{\Delta y} [\mathcal{K}(x_{i, j+1/2}) - \mathcal{K}(x_{i, j-1/2})]. \quad (53)$$

Table 8: Ex. 4, Burgers' equation. Error and convergence order at  $T = 0.25$  for the SL-WENO3 scheme.

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = \Delta x$				
40	5.20606E-03	—	1.24030E-02	—
80	1.29974E-03	2.00197	4.07879E-03	1.60448
160	3.25383E-04	1.99801	3.20622E-03	0.34727
320	4.09994E-05	2.98847	7.20030E-04	2.15474
640	4.23083E-06	3.27659	2.31519E-04	1.63693
1280	4.43564E-07	3.25373	5.25378E-05	2.13970
$\Delta t = 5\Delta x$				
40	3.62204E-02	—	1.78528E-01	—
80	5.75650E-03	2.65354	5.10251E-02	1.80687
160	8.59292E-04	2.74397	1.08849E-02	2.22888
320	1.06154E-04	3.01699	1.69780E-03	2.68059
640	1.10888E-05	3.25899	2.16030E-04	2.97436
1280	7.74855E-07	3.83903	4.75593E-05	2.18343
$\Delta t = 15\Delta x$				
40	3.62204E-02	—	1.78528E-01	—
80	3.08074E-02	0.23353	1.47193E-01	0.27844
160	3.93878E-03	2.96745	2.94998E-02	2.31893
320	5.05804E-04	2.96110	6.10904E-03	2.27169
640	5.42105E-05	3.22194	8.66999E-04	2.81684
1280	4.30414E-06	3.65477	9.24604E-05	3.22912

Table 9: Ex. 4, Burgers' equation. Error and convergence order at  $T = 0.25$  for the SL-WENO5 scheme.

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
$\Delta t = \Delta x$				
40	2.16066E-03	—	1.82590E-02	—
80	3.72800E-04	2.53500	7.02089E-03	1.37888
160	1.96631E-05	4.24484	5.89346E-04	3.57447
320	8.77459E-07	4.48601	2.58235E-05	4.51236
640	2.76269E-08	4.98919	8.69127E-07	4.89298
1280	8.74263E-10	4.98186	2.80420E-08	4.95390
$\Delta t = 5\Delta x$				
40	9.16230E-03	—	4.94176E-02	—
80	6.88724E-04	3.73371	7.05318E-03	2.80868
160	3.06176E-05	4.49149	5.81517E-04	3.60038
320	1.27523E-06	4.58554	2.20463E-05	4.72121
640	3.70836E-08	5.10383	5.87805E-07	5.22906
1280	7.39448E-10	5.64819	1.79596E-08	5.03251
$\Delta t = 15\Delta x$				
40	9.16230E-03	—	4.94176E-02	—
80	1.21819E-02	-0.41096	6.96252E-02	-0.49458
160	1.55674E-03	2.96814	8.98972E-03	2.95326
320	7.86725E-05	4.30653	1.23385E-03	2.86510
640	5.54985E-06	3.82534	1.08509E-04	3.50728
1280	1.81762E-07	4.93233	4.26829E-06	4.66802

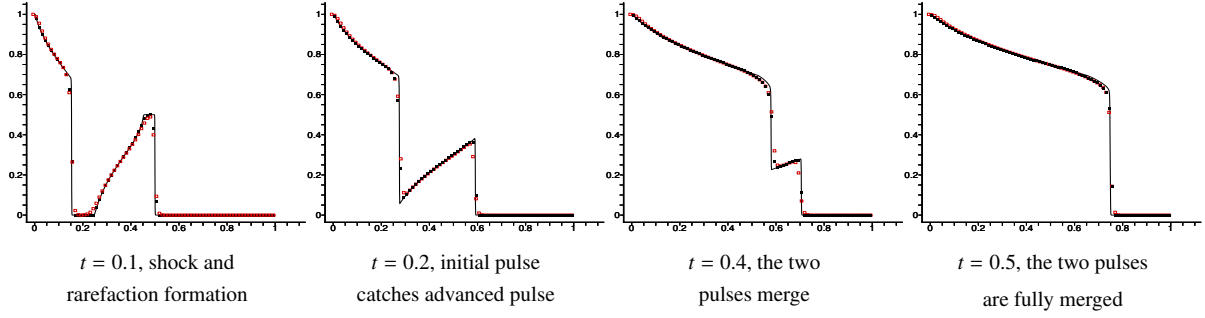


Figure 9: Ex. 5, Buckley-Leverett, WENO5. An interaction of shocks and rarefactions, resulting from the evolution of the initial condition of two pulses given in (51). The solid line is the reference solution, given by CWENO5 with a very small  $\Delta x = 1/1280$  and  $\Delta t = 1/15360$ . The filled black squares are our SL-WENO5 results, and the open red squares are WENO5-FD results, both using  $m = 80$  and  $\Delta t = 0.4\Delta x$ .

However, we prefer to view the treatment of space as a Strang splitting. For a first order Strang splitting, we compute the two-stage scheme

$$\tilde{u}_{ij} = u_{ij}^n - \frac{1}{\Delta x} [\mathcal{H}(x_{i+1/2,j}) - \mathcal{H}(x_{i-1/2,j})], \quad (54)$$

$$u_{ij}^{n+1} = \tilde{u}_{ij} - \frac{1}{\Delta y} [\tilde{\mathcal{K}}(x_{i,j+1/2}) - \tilde{\mathcal{K}}(x_{i,j-1/2})], \quad (55)$$

wherein  $\tilde{\mathcal{K}}$  is computed using the values  $\tilde{u}_{ij}$  rather than  $u_{ij}^n$ . For a second order Strang splitting, (54) is computed for the time interval  $[t^n, t^{n+1/2} = t^n + \Delta t/2]$ , (55) is applied to produce  $\tilde{\tilde{u}}_{ij}$ , and (54) is applied to these values over the rest of the time interval  $[t^{n+1/2}, t^{n+1}]$ .

#### 4.1. Example 6, rigid body rotation

The first example in two space dimensions is rigid body rotation. The governing equation is

$$u_t - ((y-1)u)_x + ((x-1)u)_y = 0, \quad x \in [0, 2], \quad y \in [0, 2], \quad t > 0. \quad (56)$$

We begin with a test using an initial condition defined as the smooth, radial bump function

$$u(x, y, 0) = \frac{2}{5} [\psi(1 + r(x, y))\psi(1 - r(x, y)) + 1], \quad r(x, y) = \sqrt{(x-1)^2 + (y-1)^2},$$

where  $\psi(s) = e^{-1/s^2}$  for  $s > 0$  and  $\psi(s) = 0$  otherwise.

Table 10 gives the  $L_{\Delta x}^1$  and  $L_{\Delta x}^\infty$  errors and the corresponding orders of convergence, using  $\Delta t = 2.5h$  (so the CFL number is  $\text{CFL}_{\Delta t} = 2.5$ ). Third and fifth order convergence is observed for SL-WENO3 and SL-WENO5, respectively, as expected. We used a first order in time Strang splitting for the third order results and a second order in time Strang splitting for the fifth order results.

We next use an initial condition that includes a slotted disk, a cone, and a “smooth” hump, similar to that used by LeVeque [19]. The initial condition is shown in Fig. 10(a).

The numerical solution after six full revolutions of the scheme using  $\Delta t = 3\Delta x$  ( $\text{CFL}_{\Delta t} = 3$ ) are shown in Fig. 10(b)–(c). One-dimensional cross-sections of the solution are given in Fig. 11, benchmarked against the exact solution. There is a little oscillation for the WENO5 results using  $N = 100$  grid elements. A refined result removes the oscillation.

#### 4.2. Example 7, swirling flow

A more severe test than rigid rotation is obtained by using a swirling and deforming flow field. Following [19], we take the velocity in the form

$$a_1(x, y) = \sin^2\left(\frac{\pi x}{2}\right) \sin(\pi y) g(t), \quad a_2(x, y) = -\sin^2\left(\frac{\pi y}{2}\right) \sin(\pi x) g(t). \quad (57)$$

Table 10: Ex. 6, rigid body rotation with smooth initial condition. SL-WENO errors and convergence order at  $t = 2\pi$ , using  $\Delta t = 2.5h$  ( $CFL_{\Delta t} = 2.5$ ).

$m$	$L_{\Delta x}^1$ error	order	$L_{\Delta x}^\infty$ error	order
SL-WENO3, first order Strang splitting				
20	1.38421E-03	—	4.27331E-03	—
40	4.27301E-04	1.69574	5.55972E-04	2.94227
80	6.29175E-05	2.76372	1.13849E-04	2.28789
160	4.61648E-06	3.76860	9.43786E-06	3.59252
320	5.38700E-07	3.09924	1.13354E-06	3.05762
SL-WENO5, second order Strang splitting				
20	8.61188E-04	—	7.75444E-04	—
40	4.47859E-05	4.26521	5.57405E-05	3.79822
80	1.66327E-06	4.75095	2.38691E-06	4.54551
160	5.15057E-08	5.01315	8.43301E-08	4.82295
320	2.57651E-09	4.32124	3.30820E-09	4.67193

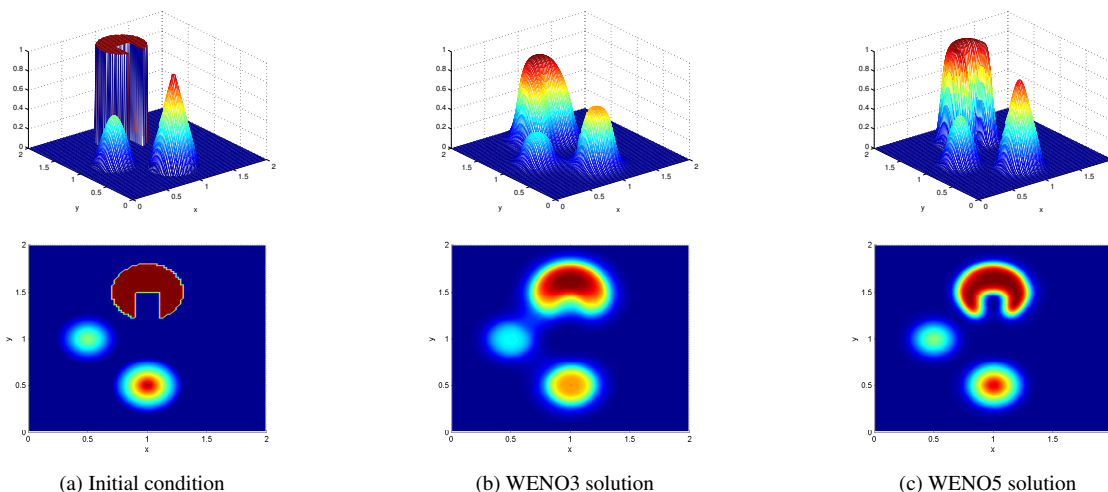


Figure 10: Ex. 6, rigid body rotation with a nonsmooth initial condition. (a) The initial condition and (b) SL-WENO3 and (c) SL-WENO5 solutions at  $t = 12\pi$ , using  $\Delta t = 3h$  ( $CFL_{\Delta t} = 3$ ), shown using point values (top) and a graphically smoothed contour plot (bottom).

This flow satisfies  $a_1 = a_2 = 0$  on the boundaries of the domain  $(0, 2) \times (0, 2)$ .

The function  $g(t)$  is used to introduce time dependence in the flow field, and in our first test we use

$$g(t) = 2 \cos(\pi t/T)$$

on the time interval  $0 \leq t \leq T$ . The flow swirls in the clockwise direction, slows down, and reverses direction in such a way that the initial data is recovered at time  $T$ . We use  $T = 1.5$  and the nonsmooth initial condition Fig. 10(a) from in the previous example. We remark that in [19], LeVeque worked over the unit square domain. Compared to his work, we have rescaled both the velocity field and  $g(t)$ . Moreover, Qiu and Shu [1] used a similar scaling over the domain  $(-\pi, \pi)^2$ .

The second order Strang splitting results are shown in Fig. 12 for time step,  $\Delta t = 3\Delta x$ . We also report cross-sectional slices in Fig. 13, benchmarked against the exact solution. The initial data becomes quite deformed by time  $T/2$ . The solution is very similar to that given in [19, 1]. We remark that very similar results are obtained if one uses the Rankine-Hugoniot jump condition to define the trace velocity (48).

As a second test using the swirling flow (57), we take  $g(t) \equiv 1$ . We also rescale the problem to the unit square and

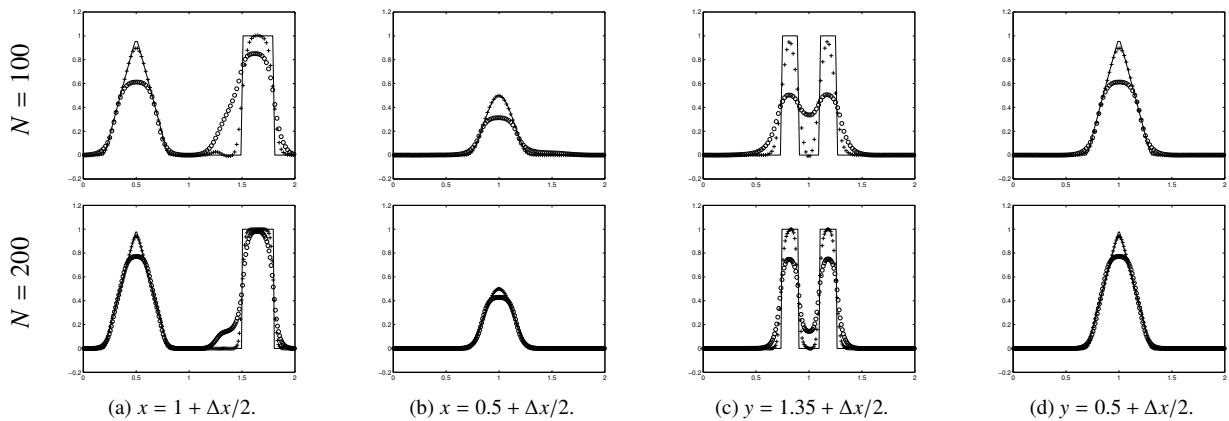


Figure 11: Ex. 6, rigid body rotation with a nonsmooth initial condition. Cross-sections of the numerical solution at (a)  $x = 1 + \Delta x/2$ , (b)  $x = 0.5 + \Delta x/2$ , (c)  $y = 1.35$ , and (d)  $y = 0.5 + \Delta x/2$ . The true solution is also shown as a solid line, circles are results of SL-WENO3, and crosses are results of SL-WENO5. The top row uses  $N = 100$  grid elements and the bottom uses  $N = 200$ .

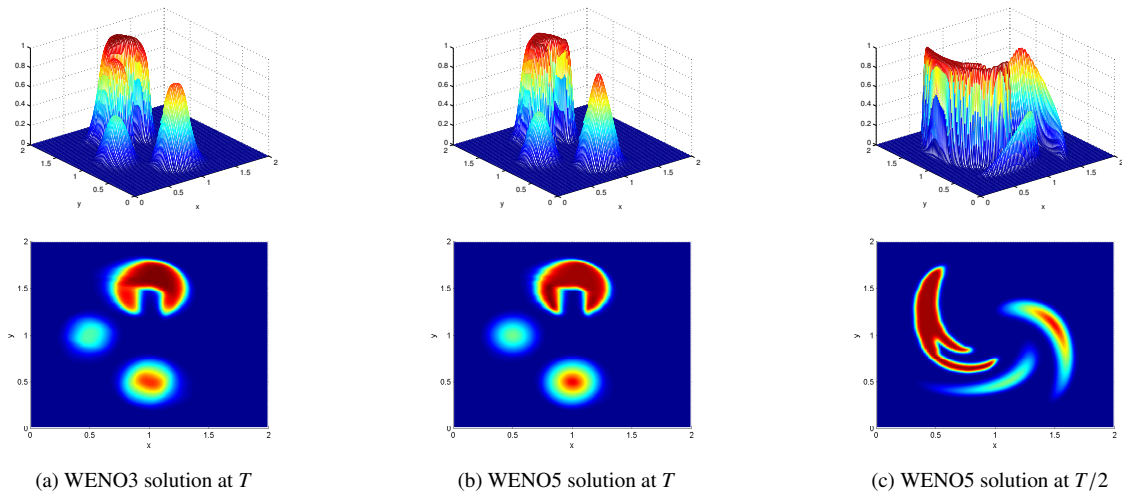


Figure 12: Ex. 7, swirling flow. Solutions for (a) SL-WENO3 at  $T = 1.5$ , (b) SL-WENO5 at  $T$ , and (c) SL-WENO5 at  $T/2$  using second order Strang splitting and  $\Delta t = 3\Delta x$  ( $\text{CFL}_{\Delta t} = 3$ ), shown using point values (top) and a graphically smoothed contour plot (bottom).

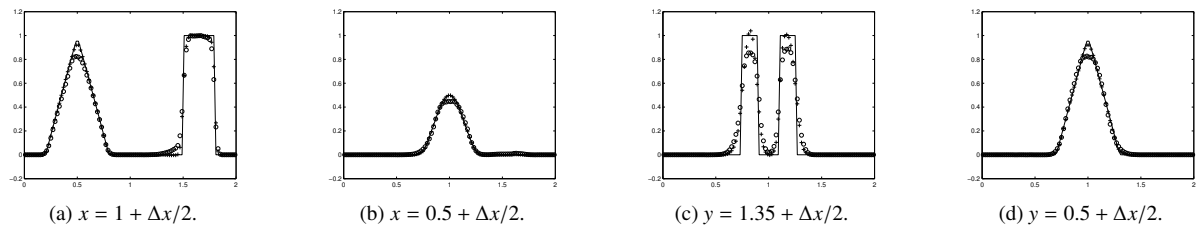


Figure 13: Ex. 7, swirling flow. Cross-sections of the numerical solution at (a)  $x = 1 + \Delta x/2$ , (b)  $x = 0.5 + \Delta x/2$ , (c)  $y = 1.35$ , and (d)  $y = 0.5 + \Delta x/2$ . The true solution is also shown as a solid line. Circles are results of WENO3, crosses are results of WENO5

apply the initial condition

$$u(x, y, 0) = \begin{cases} 1 & \text{if } (x - 1/2)^2 + (y - 1/2)^2 < 0.4^2, \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 14 shows the computed solution at time  $t = 2.5$  using  $\Delta t = 3\Delta x$ , so the CFL number is  $\text{CFL}_{\Delta t} = 3$ . The level of

numerical diffusion is very low on this  $200 \times 200$  grid.

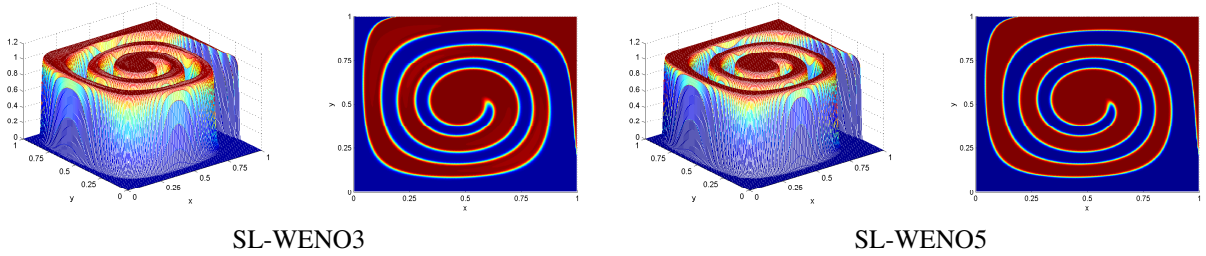


Figure 14: Ex. 7, swirling flow ( $g(t) = 1$ ). SL-WENO3 and SL-WENO5 solutions at time  $t = 2.5$  using  $\Delta t = 3\Delta x$  ( $\text{CFL}_{\Delta t} = 3$ ) and  $\Delta x = 1/200$ . Note the low level of numerical diffusion.

#### 4.3. Example 8, 2-D Burgers' equation

The next example is a simple two dimensional Burgers' equation

$$u_t + (u^2/2)_x + (u^2/2)_y = 0, \quad x \in [0, 2], y \in [0, 2], t > 0, \quad (58)$$

with the initial condition

$$u(x, y, 0) = \sin^2(\pi x) \sin^2(\pi y)$$

and periodic boundary conditions. The problem is solved using a second order Strang splitting in space. In Fig. 15 we present the SL-WENO5 solution obtained at time  $T = 1.5$  with two different mesh spacings, 80 and 320 elements in each coordinate direction. The time step  $\Delta t$  is chosen to correspond to a CFL number of 2. One can easily see that, although shocks develop in the solution, they are well resolved and exhibit no spurious oscillations. The result is comparable to [20, Fig. 4.5], although in that paper it appears that the solution has been plotted in reverse in both directions (and they show only one-quarter of what we plot here).

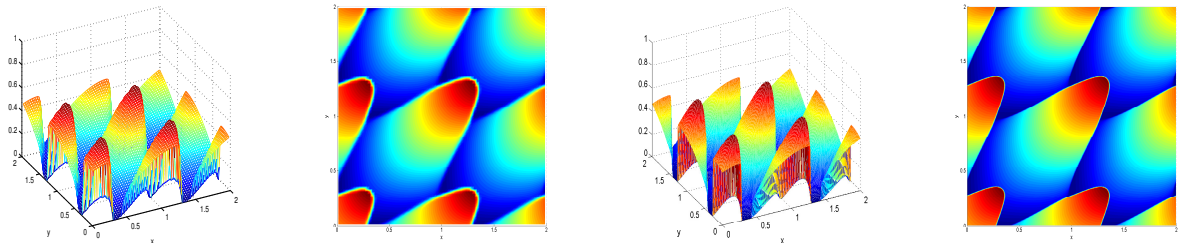


Figure 15: Ex. 8, a periodic 2-D Burgers' equation. The solution at time  $T = 1.5$  using  $\Delta t$  twice the CFL step is shown in profile and as contours using  $h = 1/80$  and  $h = 1/320$  for the two plots on the left and right, respectively.

We now impose the more challenging initial condition used by Jiang and Tadmor [21] involving the “oblique” data

$$u(x, y, 0) = \begin{cases} 0.5, & x < 0.5, y < 0.5, \\ 0.8, & x > 0.5, y < 0.5, \\ -0.2, & x < 0.5, y > 0.5, \\ -1.0, & x > 0.5, y > 0.5. \end{cases}$$

We use no-flux boundary conditions. In Fig. 16 we present the solution obtained at time  $T = 0.5$  with mesh spacing 80 elements in each coordinate direction. The time step  $\Delta t = 0.9\Delta x$  is chosen to correspond to a CFL number of 1.27. The figure shows the control of spurious oscillations in this problem involving shock interaction.

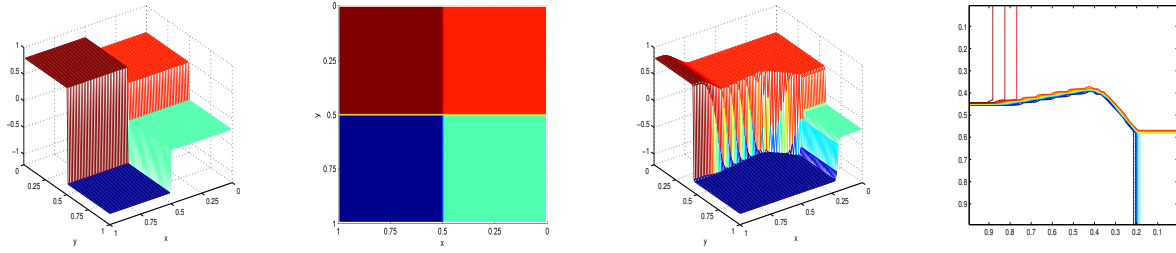


Figure 16: Ex. 8, a Burgers' problem subject to "oblique" initial data. The initial condition is similar to an example from [21], and it is shown in profile and as contours using  $h = 1/80$  for the two plots on the left. The solution at time  $T = 0.2$  is on the right, with  $\Delta t$  being 1.27 of the CFL step. The contour plot has 21 level lines.

#### 4.4. Example 9, 2-D Buckley-Leverett equation

The final example for scalar conservation law (1) uses the Buckley-Leverett flux function in each direction,

$$f_i(u) = \frac{u^2}{u^2 + (1-u)^2}, \quad i = 1, 2,$$

and involves the interaction of shocks and rarefactions. The initial condition is similar to (51) used in the one-dimensional Example 5. It is

$$u_0(x) = \begin{cases} \max(1 - 20x, 1 - 20y) & \text{for } 0 \leq x \leq 0.05 \text{ or } 0 \leq y \leq 0.05, \\ 0.5 & \text{for } 0.25 \leq x, y \leq 0.4, \\ 0 & \text{otherwise.} \end{cases} \quad (59)$$

We use  $m = 100$  grid elements and take a small nonzero constant in place of the strict minimal value zero. As one can observe in Fig. 17, the solution evolves cleanly.

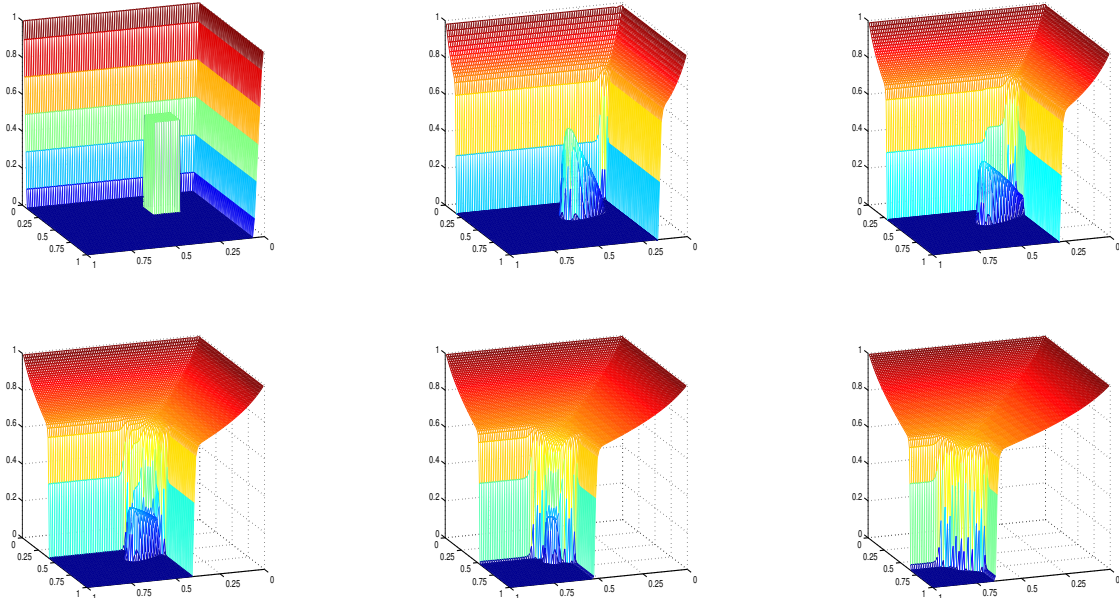


Figure 17: Ex. 9, 2-D Buckley-Leverett. An interaction of shocks and rarefactions, resulting from the evolution of the initial condition of two pulses given as the tensor product of (51). The SL-WENO5 results are shown at times 0, 0.1, 0.2, 0.3, 0.4, and 0.5.

## 5. Some numerical results related to plasma flow

### 5.1. The Vlasov-Poisson (VP) system in one space dimension

The well-known nondimensionalized Vlasov-Poisson (VP) system in one space dimension is

$$f_t + v f_x + E(t, x) f_v = 0, \quad (60)$$

$$E(t, x) = -\phi_x, \quad -\phi_{xx}(t, x) = \rho(t, x), \quad (61)$$

where  $x$  and  $v$  are the coordinates in the phase space  $(x, v) \in \mathbb{R} \times \mathbb{R}$ ,  $E$  is the electric field,  $\phi$  is the self-consistent electrostatic potential, and  $f(t, x, v)$  is the probability distribution function describing the probability of finding a particle with velocity  $v$  at position  $x$  at time  $t$ . The probability distribution function couples to long ranges fields via the relative charge density  $\rho$ . For a periodic problem in  $x \in [0, L_x]$ ,

$$\rho(t, x) = \int_{\mathbb{R}} f(t, x, v) dv - \rho_0, \quad \rho_0 = \frac{1}{L_x} \int_0^{L_x} \int_{\mathbb{R}} f(0, x, v) dv dx, \quad (62)$$

where we take the limit of uniformly distributed, infinitely massive ions in the background. We recall three classical conserved quantities:

(a) Preservation of the  $L^p$  norm, for  $1 \leq p < \infty$ ,  $\frac{d}{dt} \iint f(t, x, v)^p dx dv = 0$ ;

(b) Preservation of the energy,  $\frac{d}{dt} \left( \iint f(t, x, v) v^2 dx dv + \int E(t, x)^2 dx \right) = 0$ ;

(c) Preservation of the entropy,  $\frac{d}{dt} \iint f(t, x, v) \log(f(t, x, v)) dx dv = 0$ .

We solve a periodic VP system by taking  $f$  periodic in  $x$  over a period  $[0, L_x]$ . We also restrict the velocity to  $[-V_{\max}, V_{\max}]$  for some  $V_{\max} > 0$ , so the phase space domain is  $[0, L_x] \times [-V_{\max}, V_{\max}]$ . As is commonly done, we use Strang splitting to solve the problem (for details, see, e.g., [2, 13]), which requires solution of

$$f_t + (vf)_x = 0 \quad \text{and} \quad f_t + (Ef)_v = 0,$$

both of which have constant advection. Recall that in this case, our scheme is very similar to the scheme of Qiu and Shu [1] (only the nonlinear weighting technique differs). We report results for our fifth order SL-WENO5 scheme. A fast Fourier transform is used to solve the Poisson equation since  $f(t, x, v)$  is periodic in space.

#### 5.1.1. Example 10, Landau damping

Consider the example of Landau damping for the VP system with the initial condition

$$f(t = 0, x, v) = \frac{1}{\sqrt{2\pi}} (1 + \alpha \cos(kx)) \exp\left(-\frac{v^2}{2}\right). \quad (63)$$

For weak damping, take  $\alpha = 0.01$  and  $k = 0.5$ . We solve the system using  $L_x = 2\pi/k = 4\pi$  and  $V_{\max} = 4\pi$ . We use a rectangular mesh of  $N_x = 256$  and  $N_v = 512$  elements, and we use the time step  $\Delta t = \Delta x = L_x/N_x$ .

The correct rate of damping of the electric field is observed in Fig. 18, benchmarked with the theoretical slope  $\gamma = 0.1533$ . The largest variation in the conserved quantities, the  $L^1$ -norm,  $L^2$ -norm, energy, and entropy are  $2.0 \times 10^{-13}$ ,  $5.7 \times 10^{-7}$ ,  $4.5 \times 10^{-6}$ , and  $2.0 \times 10^{-6}$ , respectively.

For strong Landau damping of the VP system, take the initial condition (63) now with  $\alpha = 0.5$  and  $k = 0.5$ . We use  $N_x = 128$ ,  $N_v = 256$ ,  $L_x = 2\pi/k = 4\pi$ ,  $V_{\max} = 2\pi$ , and  $\Delta t = \Delta x$ . Results appear in Fig. 19. We also provide the time evolution of the first three Fourier modes of the electric field in Fig. 20. Results show damping rates of the electric field that agree with published values. All of our results are comparable to those in, e.g., [22, 23, 24].



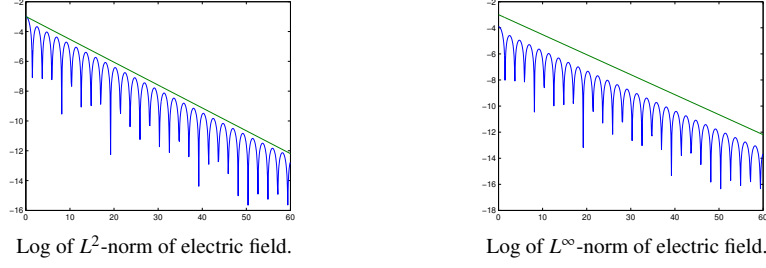


Figure 18: Ex. 10, Weak Landau damping: time evolution of the electric field in  $L^2$  (left) and  $L^\infty$  (right).

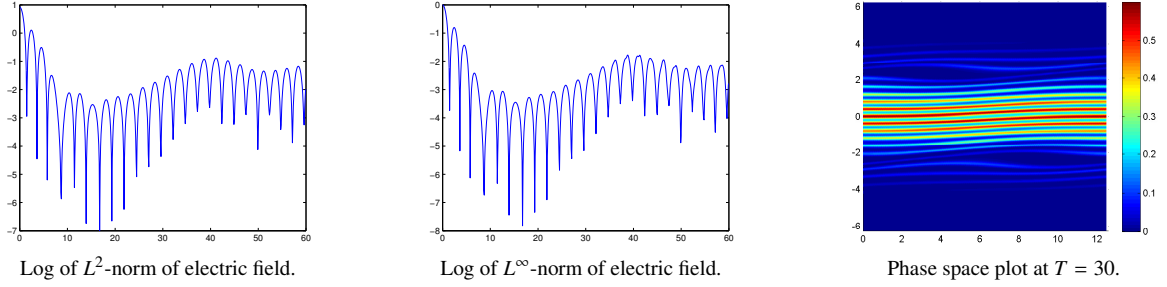


Figure 19: Ex. 10, Strong Landau damping: time evolution of the electric field in  $L^2$  (left),  $L^\infty$  (middle) and phase space plot (right at time  $T = 30$ ).

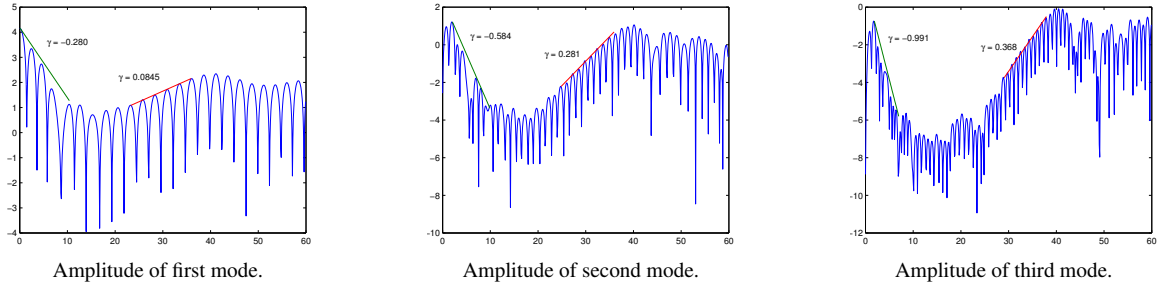


Figure 20: Ex. 10, Strong Landau damping: time evolution of the amplitude of the first three modes of the electric field.

### 5.1.2. Example 11, Two stream instabilities

Consider the first symmetric two stream instability with the unstable initial condition

$$f(t=0, x, v) = \frac{2}{7\sqrt{2\pi}} \left[ 1 + \alpha \left( \frac{\cos(2kx) + \cos(3kx)}{1.2} + \cos(kx) \right) \right] (1 + 5v^2) \exp\left(-\frac{v^2}{2}\right), \quad (64)$$

where  $\alpha = 0.01$ , and  $k = 0.5$ . We use  $L_x = 4\pi$ ,  $N_x = 256$ ,  $N_v = 256$ ,  $V_{\max} = 5$ , and  $\Delta t = \Delta x$ . The phase plots at times  $t = 30$ ,  $t = 40$  (as in [24]), and  $t = 53$  (as in [2]) are shown in Fig. 21. The results appear to show less numerical diffusion than the results in [24, 2].

Now consider the second symmetric two stream instability with initial condition

$$f(t=0, x, v) = \frac{1}{2v_{\text{th}}\sqrt{2\pi}} (1 + 0.05 \cos(kx)) \left[ \exp\left(-\frac{(v-u)^2}{2v_{\text{th}}^2}\right) + \exp\left(-\frac{(v+u)^2}{2v_{\text{th}}^2}\right) \right], \quad (65)$$

where  $u = 0.99$ ,  $v_{\text{th}} = 0.3$  and  $k = 2/13$ . We use  $L_x = 13\pi$ ,  $N_x = 1024$ ,  $N_v = 512$ ,  $V_{\max} = 5$ , and  $\Delta t = \Delta x$ . The phase plot is shown in Fig. 22. The result compares favorably with that in [2].

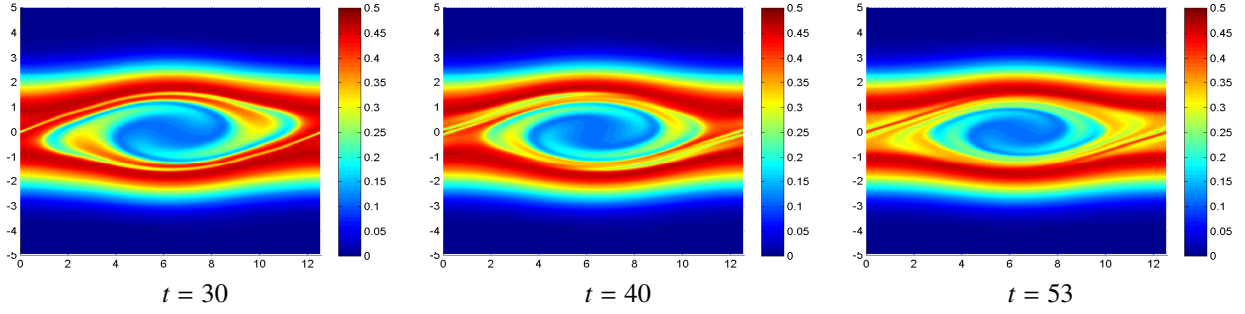


Figure 21: Ex. 11, Phase space plots of the first two stream instability using initial condition (64) at  $t = 30$ ,  $t = 40$ , and  $t = 53$ .

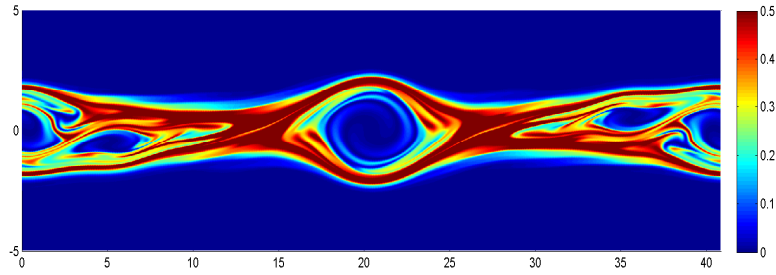


Figure 22: Ex. 11, Phase space plot of the second two stream instability using initial condition (65) at time  $t = 70$ .

## 5.2. Example 12, The guiding-center (GC) model

A computationally more challenging Vlasov-type model of plasmas is given by the guiding-center (GC) model [13, 14, 24], because the advection is not constant within the Strang splitting algorithm. The model is

$$\rho_t + \mathbf{E}^\perp \cdot \nabla \rho = 0, \quad (66)$$

$$\mathbf{E} = -\nabla \phi, \quad -\Delta \phi = \rho, \quad (67)$$

where  $\rho(t, x, y)$  is the particle density,  $\mathbf{E}(t, x, y) = (E_1, E_2)$  is the electric field,  $\phi(t, x, y)$  is the electrostatic potential, and  $\mathbf{E}(t, x, y)^\perp = (E_2, -E_1)$ . This model preserves the  $L^1$ - and  $L^2$ -norms of  $\rho$  as well as the energy, the  $L^2$ -norm of  $\mathbf{E}$ .

We pose a periodic problem on the domain  $[0, 4\pi] \times [0, 2\pi]$  by using the periodic initial condition

$$\rho(t = 0, x, y) = \sin(y) + \epsilon \cos(kx),$$

where  $\epsilon = 0.015$  and  $k = 0.5$ . We solve the transport problem using both SL-WENO3 and SL-WENO5 on various meshes with  $\Delta t = 0.1$ , and we solve for the electric field using the fast Fourier transform on a  $64 \times 64$  grid.

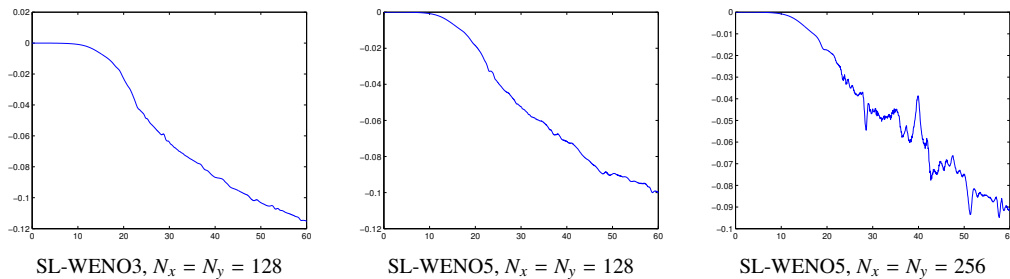


Figure 23: Ex. 12, Guiding center model: time evolution of the (log of the) energy  $\|\mathbf{E}\|_{L^2}$  for two resolutions using SL-WENO3 or SL-WENO5. The electric field  $\mathbf{E}$  is computed only on a  $64 \times 64$  grid, and so some oscillation appears in the SL-WENO5 results.

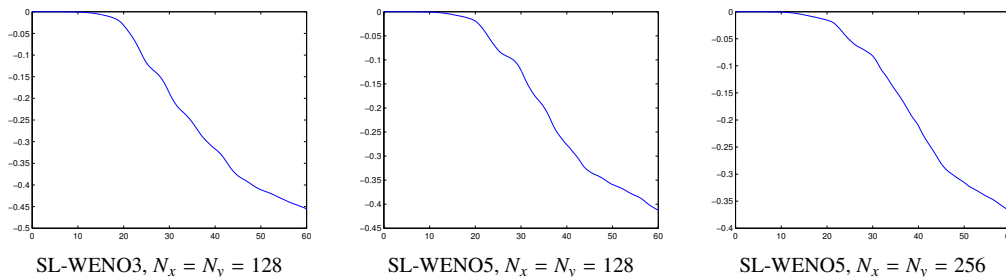


Figure 24: Ex. 12, Guiding center model: time evolution of the enstrophy  $\|\rho\|_{L^2}$  for two resolutions using SL-WENO3 or SL-WENO5.

Our scheme is mass conservative, so the  $L^1$ -norm of  $\rho$  is conserved. In Fig. 23, we show that the energy ( $L^2$ -norm of  $\mathbf{E}$ ) is conserved well to about time 20 or 30, as is typical for these problems, since the scale of the solution's features is unresolved on our grids at about this time [13]. The electric field  $\mathbf{E}$  is computed only on a  $64 \times 64$  grid, and so some oscillation appears in the SL-WENO5 results. Similar results are achieved for the enstrophy ( $L^2$ -norm of  $\rho$ ), as seen in Fig. 24.

Because we use a Strang splitting algorithm, we check the effect of grid orientation on the solution by solving the periodic problem as rotated by  $45^\circ$ . The domain is not square, so we double it before rotation. The results at time  $t = 20$  and 30 using SL-WENO3 and a grid  $N_x = N_y = 128$  are shown in Fig. 25. The results show almost no grid-orientation effect at time  $t = 20$ . There is some discrepancy at time  $t = 30$ , when the overall solution is starting to break down because of the lack of resolution, as mentioned above. Of course, the splitting error could be reduced by using the integral deferred correction method (IDC), as is done in [24].

The particle density  $\rho$  is shown in Fig. 26 at time  $t = 30$ . The results show a level of detail comparable to that in [24]. Compared to [13], we see less numerical diffusion and better preservation of the maximal densities.

## 6. Conclusions

We developed a new semi-Lagrangian finite difference WENO scheme for nonlinear scalar conservation laws in one or more space dimensions. The scheme is locally mass conservative, formally high-order accurate in space, has small time truncation error, and is essentially non-oscillatory. It is subject to a relaxed CFL time step stability condition (42). The scheme can be considered as an extension of the SL-WENO scheme of Qiu and Shu [1] for linear problems, and as a finite difference version of related finite volume schemes [5, 6].

Numerical results verified that the scheme works well, achieves its formal spatial accuracy on smooth problems, and is computationally efficient for nonlinear problems. The results showed low numerical diffusion and the ability of the scheme to capture shocks and interactions between shocks and rarefactions in one and two space dimensions. Application to the Vlasov-Poisson system gave good results, as well as to the guiding-center model. The latter requires a scheme such as ours to handle the nonconstant advection, and ours appears to be the first scheme to use the proper WENO linear weights as opposed to user defined empirical weights.

## References

- [1] J.-M. Qiu, C.-W. Shu, Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow, *J. Comput. Phys.* 230 (4) (2011) 863–889.
- [2] J.-M. Qiu, C.-W. Shu, Conservative semi-Lagrangian finite difference WENO formulations with applications to the Vlasov equation, *Communications in Comput. Phys.* 10 (2011) 979–1000.
- [3] X. D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* 115 (1994) 200–212.
- [4] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, A. Quarteroni (Eds.), *Advanced numerical approximation of nonlinear hyperbolic equations*, Vol. 1697 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, 1998, Ch. 4, pp. 325–432.
- [5] C.-S. Huang, T. Arbogast, J. Qiu, An Eulerian-Lagrangian WENO finite volume scheme for advection problems, *J. Comput. Phys.* 231 (11) (2012) 4028–4052, DOI 10.1016/j.jcp.2012.01.030.
- [6] C.-S. Huang, T. Arbogast, An Eulerian-Lagrangian WENO scheme for nonlinear conservation laws, Submitted.
- [7] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, *J. Comput. Phys.* 77 (1988) 439–471.

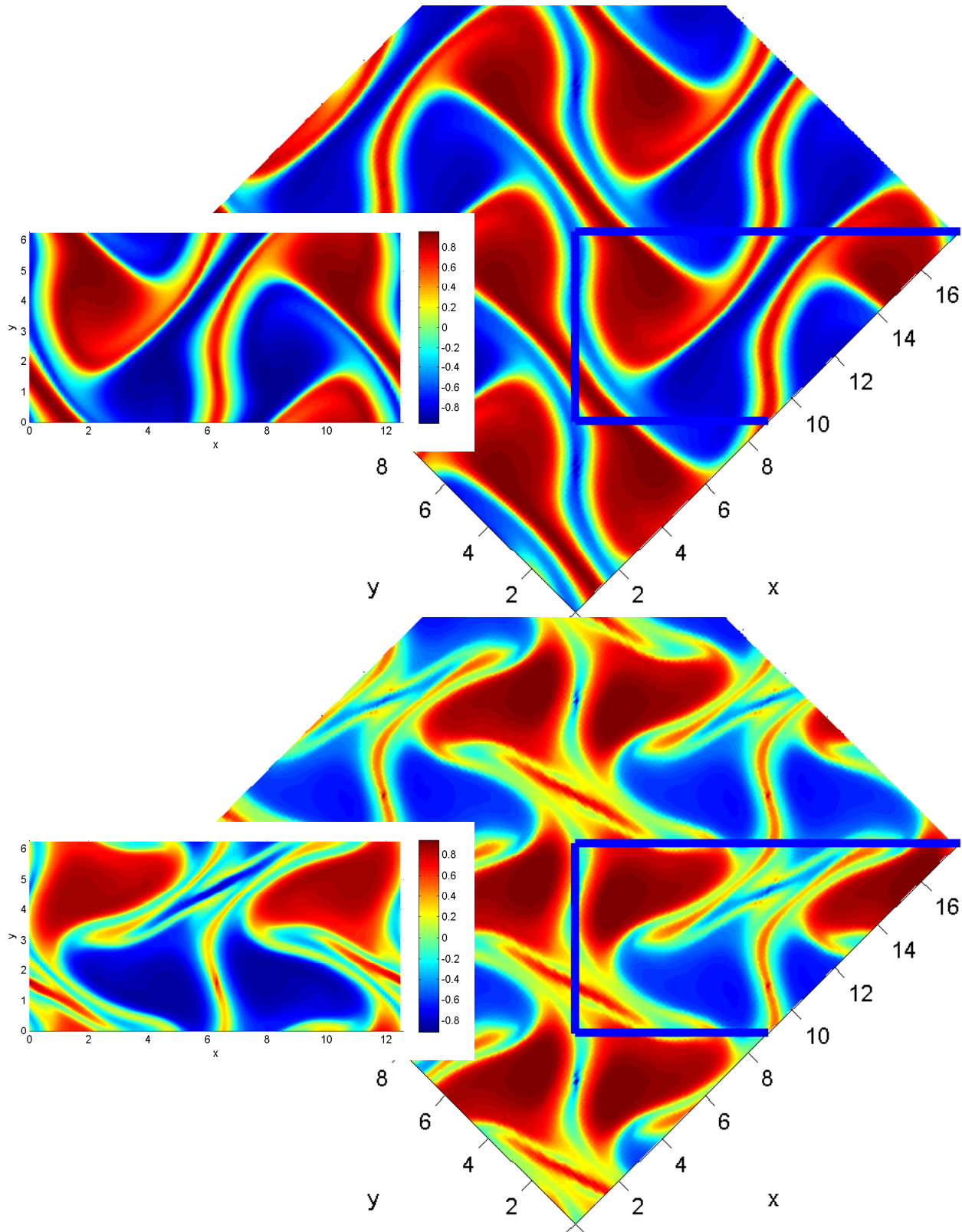


Figure 25: Ex. 12, Guiding center model: particle density  $\rho$  using SL-WENO3 and  $N_x = N_y = 128$  at times  $t = 20$  (top) and  $30$  (bottom), computed normally and rotated by  $45^\circ$ .

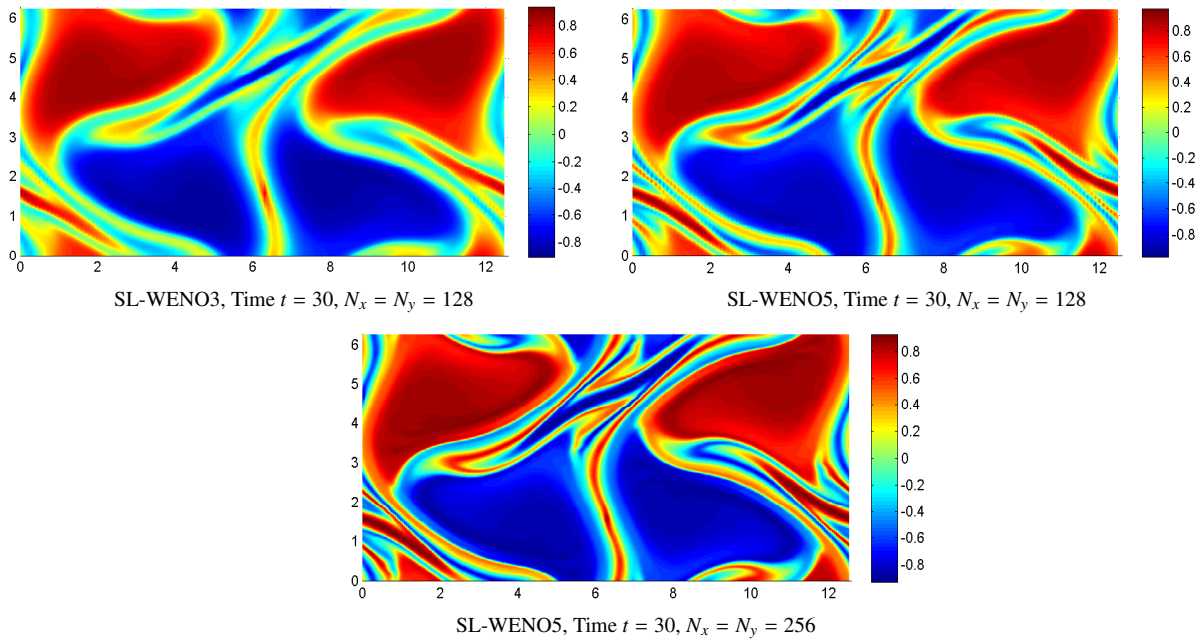


Figure 26: Ex. 12, Guiding center model: particle density  $\rho$  using SL-WENO3 (left) or SL-WENO5 (right and center) and  $N_x = N_y = 128$  (top) or  $N_x = N_y = 256$  (bottom) at time  $t = 30$ .

- [8] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202–228.
- [9] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian-Eulerian methods, in: E. Stein, R. de Borst, T. J. R. Hughes (Eds.), *Encyclopedia of Computational Mechanics*, Vol. 1: Fundamentals, John Wiley & Sons, Ltd., 2004, Ch. 14, pp. 413–437.
- [10] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, *Math. Model. Numer. Anal.* 33 (1999) 547–571.
- [11] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* 43 (1981) 357–372.
- [12] J. M. Stockie, J. A. Mackenzie, R. D. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws, *SIAM J. Sci. Comput.* 22 (5) (2001) 1791–1813.
- [13] N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, Conservative semi-Lagrangian schemes for Vlasov equations, *J. Comput. Phys.* 229 (2010) 1927–1953.
- [14] J.-P. Braeunig, N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, Guiding-center simulations on curvilinear meshes, *Discrete and Continuous Dynamical Systems Series S* 5 (2) (2012) 271–282. doi:doi:10.3934/dcdss.2012.5.271.
- [15] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, *J. Comput. Phys.* 175 (1) (2002) 108–127.
- [16] M. Zennaro, Natural continuous extensions of Runge-Kutta methods, *Math. Comp.* 46 (1986) 119–133.
- [17] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, Tech. Rep. ICASE Report no. 97–65, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia (November 1997).
- [18] C.-W. Shu, Numerical experiments on the accuracy of ENO and modified ENO schemes, *J. Sci. Comput.* 5 (1990) 127–149.
- [19] R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, *SIAM J. Numer. Anal.* 33 (2) (1996) 627–665.
- [20] D. Levy, G. Puppo, G. Russo, Compact central WENO schemes for multidimensional conservation laws, *SIAM J. Sci. Comput.* 22 (2) (2000) 656–672.
- [21] G. S. Jiang, E. Tadmor, Nonoscillatory central schemes for multidimensional hyperbolic conservation laws, *SIAM J. Sci. Comput.* 19 (1998) 1892–1917.
- [22] C. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, *J. Comput. Phys.* 22 (1976) 330–351.
- [23] T. Nakamura, T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space, *Computer Phys. Comm.* 120 (2) (1999) 122–154.
- [24] A. Christlieb, W. Guo, M. Morton, J.-M. Qiu, A high order time splitting method based on integral deferred correction for semi-Lagrangian Vlasov simulations, *J. Comput. Phys.* 267 (2014) 7–27.