Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

An implicit Eulerian-Lagrangian WENO3 scheme for nonlinear conservation laws

Chieh-Sen Huang · Todd Arbogast

Received: May 10, 2018/ Accepted: date

Abstract We present a new, formally third order, implicit Weighted Essentially Non-Oscillatory (iWENO3) finite volume scheme for solving systems of nonlinear conservation laws. We then generalize it to define an implicit Eulerian-Lagrangian WENO (iEL-WENO) scheme. Implicitness comes from the use of an implicit Runge-Kutta (RK) time integrator. A specially chosen two-stage RK method allows us to drastically simplify the computation of the intermediate RK fluxes, leading to a computationally tractable scheme. The iEL-WENO3 scheme has two main steps. The first accounts for particles being transported within a grid element in a Lagrangian sense along the particle paths. Since this particle velocity is unknown (in a nonlinear problem), a fixed trace velocity v is used. The second step of the scheme accounts for the inaccuracy of the trace velocity v by computing the flux of particles crossing the incorrect tracelines. The CFL condition is relaxed when v is chosen to approximate the characteristic velocity. A new Roe solver for the Euler system is developed to account for the Lagrangian tracings, which could be useful even for explicit EL-WENO schemes. Numerical results show that iEL-WENO3 is both less numerically diffusive and can take on the order of about 2 to 3 times longer time steps than standard WENO3 for challenging nonlinear problems. An extension is made to the advection-diffusion equation. When advection dominates, the scheme retains its third order accuracy.

This work was supported in part by the Taiwan Ministry of Science and Technology grant MOST 105-2115-M-110-006-MY2, by the Multidisciplinary and Data Science Research Center of the National Sun Yat-sen University, and by the U.S. National Science Foundation under grant DMS-1418752.

Chieh-Sen Huang
Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan, R.O.C.
E-mail: huangcs@math.nsysu.edu.tw

Todd Arbogast
Department of Mathematics, University of Texas, 2515 Speedway, Stop C1200 Austin, TX 78712-1202 and Institute for Computational Engineering and Sciences, University of Texas, 201 EAST 24th St., Stop C0200 Austin, TX 78712-1229 E-mail: arbogast@ices.utexas.edu

Keywords hyperbolic, Eulerian-Lagrangian, semi-Lagrangian, finite volume, WENO reconstruction, traceline, locally frozen, Roe solver

Mathematics Subject Classification (2000) 65M06, 65M25, 76M20, 76X05

1 Introduction

Consider the scalar conservation law

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

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

with the possibly nonlinear flux $f(u) = f(u;x,t)$. Both *essentially non-oscillatory* (ENO) and *weighted* ENO (WENO) schemes [10,11,16,18,20,23] have proven successful for high-order accurate approximation of this equation, systems of such equations, and multi-dimensional problems. WENO is normally implemented as an explicit scheme. However, many authors have experimented with implicit and semi-implicit WENO schemes, in order to improve the computation of challenging problems (see, e.g., [6,32,5,12,31]).

Apparently, Gottlieb, Mullen, and Ruuth [8,9] were the first to introduce a general implicit WENO scheme. Because the nonlinear weighting required in a WENO scheme is quite involved, they introduced in [9] several schemes in which the flux is included implicitly, but the WENO weighting is done explicitly. They defined the WENO weights using explicit, predictor-corrector, and multistep methods.

We develop a new, formally third order, implicit WENO (iWENO3) finite volume scheme for solving (1)–(2) (and systems of such equations). By a special choice of a two-stage Runge-Kutta time integrator, we can drastically simplify the computation of the intermediate Runge-Kutta fluxes. This leads to a computationally tractable scheme that treats the nonlinear weighting fully implicitly.

However, our real interest in this paper is in developing *Eulerian-Lagrangian* or *semi-Lagrangian* schemes (see, e.g., [22,7,4,2]). Eulerian-Lagrangian schemes have the significant advantage over purely Eulerian schemes in that CFL number restrictions are alleviated. The combination of Eulerian-Lagrangian and WENO ideas appears in many schemes, including those of [24–26,15,14,13]. We develop a formally third order, implicit, Eulerian-Lagrangian (finite volume) WENO scheme (iEL-WENO3) to approximate the scalar conservation law. It is based on the new Eulerian iWENO3 scheme, which has a reasonably simple structure that lends itself to modification by Eulerian-Lagrangian methods.

Similar to other nonlinear Eulerian-Lagrangian WENO schemes [14,13], the iEL-WENO scheme consists of two main steps. The first accounts for particles being transported within a grid element in a Lagrangian sense along the particle paths. Since this particle velocity is unknown (in a nonlinear problem), a fixed trace velocity $v \approx \partial f/\partial u$ must be used, as in the Arbitrary Lagrangian-Eulerian (ALE) schemes. That is, the equation is rewritten as

$$u_t + (vu)_x + \big(f(u) - vu\big)_x = 0 \quad \iff \quad u_t + (vu)_x + \big(g(u)\big)_x = 0,$$

where $g(u) = f(u) - vu$. The particles within a grid element E are traced back in time within the velocity field v to $\check{E}$, where those particles originated from. Integrating the mass leads to explicit Lagrangian mass transport from $\check{E}$ to E. This part of the computation is explicit, and it is essentially exact up to the accuracy of the WENO reconstructions used.

The second step of the scheme, the flux correction, accounts for the inaccuracy of the trace velocity v. We compute the flux of particles crossing the incorrect tracelines bounding the space-time region connecting E to $\check{E}$. Although these tracelines change with time, this computation is similar to what is done in an Eulerian scheme. By our choice of implicit Runge-Kutta method, no higher order derivatives of the flux need be computed. It is thus reasonable to compute the solution by Newton's method, even with nonlinear WENO reconstructions being involved.

Although we omit the details, a Strang splitting in space can be used to extend the one dimensional scheme to higher dimension, as in, e.g., [15].

The method can be extended to systems of equations, but we only present extension to the Euler system for a polytropic gas. Unfortunately, the local characteristic decomposition of the variables must be done explicitly. We develop a new Roe solver to account for the Lagrangian tracings. It could be useful even for explicit EL-WENO schemes.

Because of the Lagrangian tracing, the CFL condition for iEL-WENO is relaxed when v is chosen to approximate the characteristic velocity. Numerical results will show that very long time steps can be taken for linear problems, up to even 15 times the Eulerian CFL limit. For nonlinear problems, compared to standard WENO3, the iEL-WENO3 scheme is both less numerically diffusive and can take longer time steps. We will see that time steps on the order of about 2 to 3 times longer than WENO3 can be used for challenging nonlinear scalar problems and Euler systems.

While it is true that implicit schemes are computationally more demanding, efficiency is not our main concern in this work. Rather, we want the ability to handle challenging nonlinear problems, and to see improved fidelity of the solution. Moreover, we are interested in advection-diffusion problems, which must have an implicit component. Our approach is suited to solving advection-diffusion equations, as long as the problem is advection dominated. In that case, the extra computational work is already required for the diffusive part of the computation, and our scheme becomes computationally reasonable. We present three numerical results for an advection dominated problem, which will show third order accuracy and the ability to capture sharp fronts extremely well.

Our new scheme has three main complications. It is implicit, it uses Lagrangian ideas, and it extends to systems. Before presenting the full scheme, we first describe the iWENO3 scheme in Section 2 for scalar equations completely in Eulerian terms. This Eulerian scheme could be implemented; however, as noted, our interest is to present the Eulerian-Lagrangian version iEL-WENO3, which we do in Section 3. We present numerical results for scalar equations in Section 4, as well as our extension and application to advection-diffusion problems. In Section 5, we extend the iEL-WENO3 scheme to the Euler system, and we present numerical results for it in Section 6. Conclusions appear in the final section of the paper.

2 The implicit finite volume WENO3 scheme

Partition time and space as $0 = t^0 < t^1 < t^2 < \cdots$ and $\cdots < x_{-1} < x_0 < x_1 < \cdots$, respectively, where $\Delta x = x_{i+1} - x_i$ is constant. The grid cell or element is $E_{i+1/2} = [x_i, x_{i+1}]$. We assume that we have as data (approximately) the cell average values of the solution $u(x)$, which are

$$\bar{u}^n_{i+1/2} = \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} u(x,t^n)\,dx, \quad i = \ldots,-1,0,1,\ldots. \tag{3}$$

The standard Eulerian finite volume scheme is derived by integrating (1) over $[x_i,x_{i+1}] \times [t^n,t^{n+1}]$ to obtain

$$\bar{u}^{n+1}_{i+1/2} = \bar{u}^n_{i+1/2} + \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \big(f(u(x_i,t) - f(u(x_{i+1},t))\big)\,dt. \tag{4}$$

For a third order scheme, it is sufficient to replace the time integration by a locally fourth order quadrature rule. We use a two point Gaussian quadrature rule, i.e.,

$$\int_{t^n}^{t^{n+1}} f(u(x_i,t)\,dt \approx \Delta t\Big[\omega^{G_1} f(u(x_i,t^{G_1})) + \omega^{G_2} f(u(x_i,t^{G_2}))\Big], \tag{5}$$

where $\{\omega^{G_1},\omega^{G_2}\}$ are the Gauss weights for the reference interval $[0,1]$ and $\{t^{G_1},t^{G_2}\}$ are the Gauss points for the interval $[t^n,t^{n+1}]$.

In order to accommodate solutions with shocks, we need a numerical flux function $\hat{f}$ that takes into account right and left values of the solution at the grid points. We use the standard Lax-Friedrichs flux, i.e.,

$$\hat{f}(a,b) = \frac{1}{2}[f(a) + f(b) - \alpha(b-a)], \tag{6}$$

where $\alpha = \max\limits_{u,x,t} |\partial f(u)/\partial u|$ is a constant. (We could also define α locally [21], which generally results in sharper resolution of shocks.) Let us denote $u_i(t) := u(x_i,t)$, and denote reconstructed right and left point values by $u_{i+}(t)$ and $u_{i-}(t)$, respectively. Then the scheme takes the general form

$$\begin{aligned}\bar{u}^{n+1}_{i+1/2} = \bar{u}^n_{i+1/2} + \frac{\Delta t}{\Delta x}\sum_{\ell=1}^{2} \omega^{G_\ell}\Big[&\hat{f}\big(u_{i-}(t^{G_\ell}),u_{i+}(t^{G_\ell})\big)\\ &- \hat{f}\big(u_{(i+1)-}(t^{G_\ell}),u_{(i+1)+}(t^{G_\ell})\big)\Big].\end{aligned} \tag{7}$$

In order to find proper values for $u_{i\pm}(t)$ at the Gauss points in time, we fix $x = x_i$ and apply an implicit Runge-Kutta method to the problem

$$\frac{\partial u_{i\pm}(t)}{\partial t} = -f(u_{i\pm}(t))_x := F(t,u_{i\pm}(t)), \tag{8a}$$

$$u_{i\pm}(t^n) = \mathscr{R}_{i\pm}(\bar{u}^n_{i-3/2},\ldots,\bar{u}^n_{i+3/2}) \tag{8b}$$

using right or left biased reconstructed values $\mathscr{R}_{i\pm}$ of $u(x_i,t^n)$ for the initial condition, as described below. A second order Runge-Kutta method is sufficient, since such a scheme is locally third order and the result is used inside a time integral. The solution at the $(n+1)$st time step obtained by a general two-stage Runge-Kutta scheme can be written as

$$u_{i\pm}^{n+1} = u_{i\pm}^{n} + \Delta t\left(b_1 g_{i\pm}^{(1)} + b_2 g_{i\pm}^{(2)}\right), \tag{9}$$

where the Runge-Kutta fluxes are defined by

$$g_{i\pm}^{(1)} = F\big(t^n + c_1\Delta t,\ u_{i\pm}^{n} + \Delta t(a_{11}g_{i\pm}^{(1)} + a_{12}g_{i\pm}^{(2)})\big), \tag{10a}$$

$$g_{i\pm}^{(2)} = F\big(t^n + c_2\Delta t,\ u_{i\pm}^{n} + \Delta t(a_{21}g_{i\pm}^{(1)} + a_{22}g_{i\pm}^{(2)})\big). \tag{10b}$$

The matrix a and the vectors b and c define the Runge-Kutta method, and the method is explicit precisely when a is lower triangular. For consistency, $c_i = \sum_j a_{ij}$.

To avoid solving (8) independently for each Gauss time, it is common to solve it once and use the natural continuous extension (NCE) of Zennaro [33]. The NCE provides a uniformly accurate solution of the differential equation over the entire time interval. In fact, each ν-stage Runge-Kutta method of order p has an NCE. To illustrate the idea, consider $U_t = F(U)$. One simply computes the usual Runge-Kutta fluxes $g^{(\ell)}$, $\ell = 1,\dots,\nu$, which are normally recombined to form the solution at time t^{n+1} as $U(t^{n+1}) \approx U^{n+1} = U^n + \Delta t\sum_{\ell=1}^{\nu} b_\ell g^{(\ell)}$. Zennaro proved that there exist ν polynomials $b_\ell(\theta)$, $\ell = 1,\dots,\nu$, of degree at most p, such that $U(t^n+\theta\Delta t) \approx U^{n+\theta} = U^n + \Delta t\sum_{\ell=1}^{\nu} b_\ell(\theta)g^{(\ell)}$ to order p for any $0 \le \theta \le 1$. In our case, an accurate solution at time $t^n + \theta\Delta t$ can be found as

$$u_{i\pm}(t^n+\theta\Delta t) = u_{i\pm}(t^n) + \Delta t\left(b_1(\theta)g_{i\pm}^{(1)} + b_2(\theta)g_{i\pm}^{(2)}\right), \tag{11}$$

where the polynomials $b_1(\theta)$ and $b_2(\theta)$ depend on the Runge-Kutta method chosen.

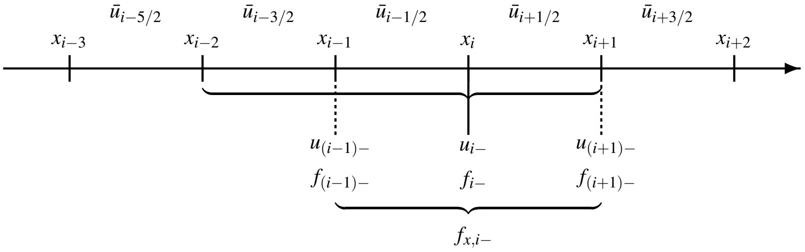


Fig. 1 Illustration of the left reconstructions $\mathscr{R}_{i-} = u_{i-}$ and $f_{x,i-}$ used in the Runge-Kutta step. These computations are explicit at t^n but implicit at t^{n+1}. The average values $\{\bar{u}_{i-3/2}, \bar{u}_{i-1/2}, \bar{u}_{i+1/2}\}$ are combined using standard WENO3 to obtain the point value u_{i-}, and similarly for $u_{(i-1)-}$ and $u_{(i+1)-}$. The values of f can then be evaluated, and further combined using WENO to obtain a reconstruction of the derivative $f_{x,i-}$.

We follow Levy et al. [18,19] to compute the WENO reconstruction of $F(t,u) = -f(u)_x$ in (10). We illustrate the idea in Fig. 1 for the left reconstructions. Average values of the solution $\bar{u}_{j+1/2}^{n}$ or, implicitly, $\bar{u}_{j+1/2}^{n+1}$ are combined using standard WENO3 to obtain right or left biased point values $u_{i\pm}$, respectively. We denote these two operators by $\mathscr{R}_{i\pm}(\bar{u}_{i-3/2},\dots,\bar{u}_{i+3/2}) = u_{i\pm}$, which uses the right stencil $\{\bar{u}_{i-1/2}, \bar{u}_{i+1/2}, \bar{u}_{i+3/2}\}$ or the left stencil $\{u_{i-3/2}, u_{i-1/2}, u_{i+1/2}\}$, respectively. For example,

$$\mathscr{R}_{i-} = w_1\Big(-\frac{\bar{u}_{i-3/2}}{2} + \frac{3\bar{u}_{i-1/2}}{2}\Big) + w_2\Big(\frac{\bar{u}_{i-1/2}}{2} + \frac{\bar{u}_{i+1/2}}{2}\Big),$$

where w_1 and w_2 are the nonlinear weights, i.e., $1/3$ and $2/3$ modified by the smoothness indicators $IS_1 = (\bar{u}_{i-3/2} - \bar{u}_{i-1/2})^2$ and $IS_2 = (\bar{u}_{i-1/2} - \bar{u}_{i+1/2})^2$.

Once we have point values of u, we can then evaluate f. A second WENO reconstruction of three of these values gives the point value of the derivative of f. That is,

$$f_{i\pm,x} = w_1\Big(\frac{f_{i\pm} - f_{i\pm-1}}{\Delta x}\Big) + w_2\Big(\frac{f_{i\pm+1} - f_{i\pm}}{\Delta x}\Big), \tag{12}$$

where w_1 and w_2 are the nonlinear modifications of the linear weights $1/2$ and $1/2$ using $IS_1 = (f_{i\pm-1} - f_{i\pm})^2$ and $IS_2 = (f_{i\pm} - f_{i\pm+1})^2$.

At first glance, it appears that finding the solution of the scheme requires, e.g., Newton's method applied to the equations (7) and (10) for the variables $\bar{u}^{n+1}_{i+1/2}$, $g^{(1)}_{i\pm}$, and $g^{(2)}_{i\pm}$ (using also (11)). However, (9) allows us to remove, e.g., $g^{(2)}_{i\pm}$ in terms of $u^{n+1}_{i\pm}$ and $g^{(1)}_{i\pm}$, which in turn are reconstructions of $\bar{u}^{n+1}_{j+1/2}$; that is, we have that

$$g^{(2)}_{i\pm} = \frac{1}{b_2(1)}\Big(\frac{u^{n+1}_{i\pm} - u^n_{i\pm}}{\Delta t} - b_1(1)\,g^{(1)}_{i\pm}\Big). \tag{13}$$

Computation showed that it is indeed valuable to eliminate $g^{(2)}_{i\pm}$; it is both less costly and allows for longer time steps.

Recalling (9)–(10), perhaps the best way to describe a Runge-Kutta method is by its Butcher tableau $\begin{array}{c|c} c & a \\ \hline & b^T \end{array}$. We chose to investigate two implicit Runge-Kutta schemes. The first is the second-order A-stable scheme [17]

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}, \tag{14}$$

for which $b_1(\theta) = \theta - \theta^2/2$ and $b_2(\theta) = \theta^2/2$. The other is the strongly S-stable diagonally implicit Runge-Kutta (DIRK or RK) scheme [1]

$$\begin{array}{c|cc} 1-\sqrt{2}/2 & 1-\sqrt{2}/2 & 0 \\ 1 & \sqrt{2}/2 & 1-\sqrt{2}/2 \\ \hline & \sqrt{2}/2 & 1-\sqrt{2}/2 \end{array}, \tag{15}$$

where $b_1(\theta) = \sqrt{2}(\theta - \theta^2/2)$ and $b_2(\theta) = (1-\sqrt{2})\theta + \sqrt{2}\theta^2/2$. Computational results showed little difference between the two Runge-Kutta methods (14) and (15). We proceed with the first choice, i.e., the scheme (14). It is simpler than the other, since $g^{(1)}_{i\pm}$ is computed explicitly as

$$g^{(1)}_{i\pm} = F(t^n, u^n_{i\pm}). \tag{16}$$

When (13) is used to remove $g_{i\pm}^{(2)}$, we need only use Newton's method to solve for $\bar{u}_{i+1/2}^{n+1}$. This simplicity is, in fact, needed in the next section to define an Eulerian-Lagrangian version of the scheme.

We now express our implicit scheme using the Runge-Kutta method (14) in somewhat more detail to illuminate its nonlinear structure. First note that, by (11) and (13),

$$\begin{aligned} u_{i\pm}(t^{G_\ell}) &= \Big(1 - \frac{b_2(\theta_\ell)}{b_2(1)}\Big) u_{i\pm}^n + \Delta t \Big(b_1(\theta_\ell) - b_2(\theta_\ell)\frac{b_1(1)}{b_2(1)}\Big) g_{i\pm}^{(1)} + \frac{b_2(\theta_\ell)}{b_2(1)} u_{i\pm}^{n+1} \\ &= (1-\theta_\ell^2) u_{i\pm}^n + \Delta t(\theta_\ell - \theta_\ell^2) g_{i\pm}^{(1)} + \theta_\ell^2 u_{i\pm}^{n+1}, \end{aligned} \tag{17}$$

where $\theta_\ell = \frac{1}{2}(1 + (-1)^\ell/\sqrt{3})$ and $\ell = 1,2$. Thus,

$$u_{i\pm}(t^{G_\ell}) = \theta_\ell^2 \mathscr{R}_{i\pm}(\bar{u}_{i-3/2}^{n+1}, \dots, \bar{u}_{i+3/2}^{n+1}) + r_{i\pm}^\ell, \tag{18}$$

where

$$r_{i\pm}^\ell = (1-\theta_\ell^2)\mathscr{R}_{i\pm}(\bar{u}_{i-3/2}^n, \dots, \bar{u}_{i+3/2}^n) + \Delta t(\theta_\ell - \theta_\ell^2) g_{i\pm}^{(1)} \tag{19}$$

is explicit. Note that, due to our special choice of Runge-Kutta method (14), no *implicit* derivatives of f are needed to compute (18) (only *explicit* derivatives are needed to compute $r_{i\pm}^\ell$, i.e., $g_{i\pm}^{(1)}$ in (19)). Then our scheme (7) with (6) can be expressed as

$$\begin{aligned} \bar{u}_{i+1/2}^{n+1} = \bar{u}_{i+1/2}^n + \frac{\Delta t}{\Delta x} \sum_{\ell=1}^{2} \omega^{G_\ell} \Big[& \hat{f}\big(\theta_\ell^2 \mathscr{R}_{i-}(\bar{u}_{i-3/2}^{n+1}, \bar{u}_{i-1/2}^{n+1}, \bar{u}_{i+1/2}^{n+1}) + r_{i-}^\ell, \\ & \qquad \theta_\ell^2 \mathscr{R}_{i+}(\bar{u}_{i-1/2}^{n+1}, \bar{u}_{i+1/2}^{n+1}, \bar{u}_{i+3/2}^{n+1}) + r_{i+}^\ell\big) \\ & - \hat{f}\big(\theta_\ell^2 \mathscr{R}_{(i+1)-}(\bar{u}_{i-1/2}^{n+1}, \bar{u}_{i+1/2}^{n+1}, \bar{u}_{i+3/2}^{n+1}) + r_{(i+1)-}^\ell, \\ & \qquad \theta_\ell^2 \mathscr{R}_{(i+1)+}(\bar{u}_{i+1/2}^{n+1}, \bar{u}_{i+3/2}^{n+1}, \bar{u}_{i+5/2}^{n+1}) + r_{(i+1)+}^\ell\big)\Big]. \end{aligned} \tag{20}$$

The scheme (20) is implicit since the arguments of $\hat{f}$ contains the five unknowns $\{\bar{u}_{i-3/2}^{n+1}, \dots, \bar{u}_{i+5/2}^{n+1}\}$. It is also a nonlinear scheme, due to the (possible) nonlinearity of $f(u)$, but also due to the nonlinear weighting used within the WENO reconstructions, which appear directly and implicitly in $\mathscr{R}_{i\pm}(\bar{u}_{i-3/2}^{n+1}, \dots, \bar{u}_{i+3/2}^{n+1})$ but also show up explicitly in $r_{i\pm}^\ell$ and $g_{i\pm}^{(1)}$. A nonlinear solver such as Newton's method can be used to solve (20). Terms like

$$\begin{aligned} \frac{\partial f(u_{i\pm}(t^{G_\ell}))}{\partial \bar{u}_{j+1/2}^{n+1}} &= \frac{\partial f(u_{i\pm}(t^{G_\ell}))}{\partial u} \frac{\partial u_{i\pm}(t^{G_\ell})}{\partial \bar{u}_{j+1/2}^{n+1}} \\ &= \frac{\partial f(u_{i\pm}(t^{G_\ell}))}{\partial u} \theta_\ell^2 \frac{\partial \mathscr{R}_{i\pm}(\bar{u}_{i-3/2}^{n+1}, \dots, \bar{u}_{i+3/2}^{n+1})}{\partial \bar{u}_{j+1/2}^{n+1}} \end{aligned}$$

must be computed to obtain the Jacobian matrix. The choice of Runge-Kutta method (14) means that we do not need to compute higher order derivatives of the flux terms.

3 The implicit Eulerian-Lagrangian WENO3 scheme

We present the iEL-WENO3 scheme in this section. A Strang splitting in space can be used to extend the one dimensional scheme to higher dimensions, but we omit the details (see, e.g., [15]).

3.1 The approximate velocity v for Lagrangian tracing

The true Lagrangian trace velocity is unknown in a nonlinear problem. We adopt the point of view of the Arbitrary Lagrangian-Eulerian (ALE) schemes and fix a known velocity $v(x,t)$ that we view as approximating the true velocity over the time interval $[t^n,t^{n+1}]$ (using information computed up to time t^n). Following [13], Lagrangian tracing of a particle at position x at time t^{n+1} through the velocity field $v(x,t)$ gives the position $\check{x}(t) = \check{x}(t;x)$, which we will call the v-trace. It satisfies

$$\frac{d\check{x}(t)}{dt} = v(\check{x}(t),t) \quad \text{and} \quad \check{x}(t^{n+1}) = x. \tag{21}$$

Generally we trace backward to time t^n, and we call $\check{x}^n(x) := \check{x}(t^n;x)$ the v-trace-back point for x. For a grid point x_i, we also denote $\check{x}_i(t) := \check{x}(t;x_i)$ and $\check{x}_i^n := \check{x}(t^n;x_i)$, as illustrated in Fig. 2. As we will see, the scheme itself requires only $\check{x}_i^n$, not the entire curve. We can solve (21) as accurately as we wish, since v is given. However, v is only an approximation, so we do not require $\check{x}_i^n$ to high accuracy. The flux correction step in Section 3.3 will correct any errors made here.

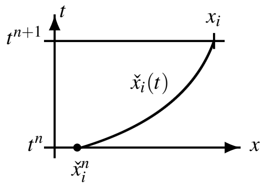


Fig. 2 Illustration of the v-traceline $\check{x}_i(t)$ of a grid point x_i.

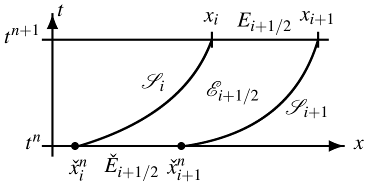


Fig. 3 Illustration of the space-time region $\mathscr{E}_{i+1/2}$ with its top $E_{i+1/2}$, bottom $\check{E}_{i+1/2}$ and sides $\mathscr{S}_i$ and $\mathscr{S}_{i+1}$.

In an Eulerian-Lagrangian method, one traces all the particles within an Eulerian grid element. For the element $E_{i+1/2} = [x_i,x_{i+1}]$, we denote by $\mathscr{E}_{i+1/2}$ the resulting swept space-time region using the velocity v, which is

$$\mathscr{E}_{i+1/2} := \{(\check{x}(t;x),t) : t^n \le t \le t^{n+1} \text{ and } x \in E_{i+1/2}\},$$

as illustrated in Fig. 3. The boundary consists of four parts, $E_{i+1/2}$ on the top when $t = t^{n+1}$ and

$$\begin{aligned}
\check{E}_{i+1/2} &:= \{\check{x}(t^n;x) : x \in E_{i+1/2}\} = [\check{x}_i^n,\check{x}_{i+1}^n] && \text{on the bottom when } t = t^n,\\
\mathscr{S}_i &:= \{(\check{x}_i(t),t) : t^n \le t \le t^{n+1}\} && \text{on the left side,}\\
\mathscr{S}_{i+1} &:= \{(\check{x}_{i+1}(t),t) : t^n \le t \le t^{n+1}\} && \text{on the right side.}
\end{aligned}$$

We call the interval $\check{E}_{i+1/2}$ the v-trace-back of $E_{i+1/2}$.

3.2 Mass conservation over the v-trace-back space-time region

We view the differential equation (1) in divergence form, i.e., as

$$\nabla_{t,x} \cdot \begin{pmatrix} u \\ f(u) \end{pmatrix} = 0. \tag{22}$$

For each grid element $E_{i+1/2}$, we integrate this in space-time over $\mathscr{E}_{i+1/2}$ and apply the divergence theorem. The result is

$$\begin{aligned}
&\int_{E_{i+1/2}} u(x,t^{n+1})\,dx - \int_{\check{E}_{i+1/2}} u(x,t^n)\,dx \\
&\quad = -\int_{\mathscr{S}_i} \begin{pmatrix} u \\ f(u) \end{pmatrix} \cdot \begin{pmatrix} v \\ -1 \end{pmatrix} \frac{d\sigma}{\sqrt{1+v^2}} - \int_{\mathscr{S}_{i+1}} \begin{pmatrix} u \\ f(u) \end{pmatrix} \cdot \begin{pmatrix} -v \\ 1 \end{pmatrix} \frac{d\sigma}{\sqrt{1+v^2}} \\
&\quad = \int_{t^n}^{t_{n+1}} \big(f(u) - vu\big)\big|_{x=\check{x}_i(t)}\,dt - \int_{t^n}^{t_{n+1}} \big(f(u) - vu\big)\big|_{x=\check{x}_{i+1}(t)}\,dt.
\end{aligned} \tag{23}$$

In a strict Eulerian-Lagrangian method, one would trace the particles in space-time along the physically correct velocity, $f(u)/u$. With $v = f(u)/u$, there would be no flux terms on $\mathscr{S}_i$ and $\mathscr{S}_{i+1}$. In our setting, with $f(u)/u$ unknown, we need to account for this flux, as well as the integration of u over the v-trace-back $\check{E}_{i+1/2}$. Therefore, we have two main steps in our scheme: the computation of the mass at time t^n and the flux correction, which is an approximation of the flux along $\mathscr{S}_i$ (and $\mathscr{S}_{i+1}$).

Of course, the integral at time t^{n+1} in (23) is approximated as

$$\int_{E_{i+1/2}} u(x,t^{n+1})\,dx \approx \Delta x\, \bar{u}_{i+1/2}^{n+1}. \tag{24}$$

The integral at time t^n is approximated as

$$\int_{\check{E}_{i+1/2}} u(x,t^n)\,dx = \sum_j \int_{\check{E}_{i+1/2} \cap E_{j+1/2}} u(x,t^n)\,dx \approx \sum_j \int_{\check{E}_{i+1/2} \cap E_{j+1/2}} \check{\mathscr{R}}(x,\bar{u}^n)\,dx, \tag{25}$$

where $\check{\mathscr{R}}(x,\bar{u}^n)$ is the reconstruction operator used at time t^n. We could use a relatively standard WENO3 targeting high order approximation of the integral, which is what is used in the linear transport scheme described by the authors and Qiu in [15]. However, we shall adopt the somewhat simpler approach Levy, Puppo, and Russo [18,19] used to define their CWENO3 scheme, because this approach does not require pre-defined exact linear weights. To avoid confusion, we refer this type of reconstruction as LPR-WENO.

Briefly, $\check{\mathscr{R}}(x,\bar{u})$ is a reconstruction of u using the cell averages $\{\bar{u}_{j-1/2}, \bar{u}_{j+1/2}, \bar{u}_{j+3/2}\}$. The same reconstruction is valid for any point $x \in E_{j+1/2}$. It is a linear combination of the one quadratic polynomial and the two linear polynomials that preserve average mass over each grid element of the interval $[x_{j-1}, x_{j+2}]$, $[x_{j-1}, x_{j+1}]$, and

$[x_j, x_{j+2}]$, respectively. Using the cell centers $x_{k+1/2} = x_k + \Delta x/2$, these polynomials are

$$\begin{aligned} P_0(x) &= \bar{u}_{j+1/2} - \frac{1}{24}\big(\bar{u}_{j+3/2} - 2\bar{u}_{j+1/2} + \bar{u}_{j-1/2}\big) + \frac{\bar{u}_{j+3/2} - \bar{u}_{j-1/2}}{2\Delta x}(x - x_{j+1/2}) \\ &\qquad + \frac{\bar{u}_{j+3/2} - 2\bar{u}_{j+1/2} + \bar{u}_{j-1/2}}{2\Delta x^2}(x - x_{j+1/2})^2, \qquad (26) \\ P_L(x) &= \bar{u}_{j-1/2}\frac{x_{j+1/2} - x}{\Delta x} + \bar{u}_{j+1/2}\frac{x - x_{j-1/2}}{\Delta x}, \qquad (27) \\ P_R(x) &= \bar{u}_{j+1/2}\frac{x_{j+3/2} - x}{\Delta x} + \bar{u}_{j+3/2}\frac{x - x_{j+1/2}}{\Delta x}. \qquad (28) \end{aligned}$$

For some choice of fixed weights $W_1 > 0$ and $W_2 > 0$ such that $2W_1 + W_2 = 1$, the reconstruction is then

$$\check{\mathscr{R}}(x, \bar{u}) = \frac{w_2}{W_2}\big[P_0(x) - W_1\big(P_L(x) + P_R(x)\big)\big] + w_{1,L}P_L(x) + w_{1,R}P_R(x), \qquad (29)$$

where w_2 is the nonlinear weight for W_2 using P_0, and $w_{1,L}$ and $w_{1,R}$ are the nonlinear weights for W_1 using P_L and P_R, respectively. The reconstruction can be integrated over any interval, including the one we need $\check{E}_{i+1/2} \cap E_{j+1/2}$.

3.3 The flux correction: approximation of the flux between space-time regions

It remains to approximate the two integrals of the flux in (23) over the space-time paths $\mathscr{S}_i$ and $\mathscr{S}_{i+1}$. As in the purely Eulerian scheme (20) of the previous section, we need to use a numerical flux for these terms. Again, we use the Lax-Friedrichs flux function, but now this function is taken for

$$g(u) := f(u) - vu. \qquad (30)$$

Technically, we should take $\alpha = \max_{u,x,t} |\partial f(u)/\partial u - v|$. However, for simplicity, we took $\alpha = \max_{u,x,t} |\partial f(u)/\partial u|$ as before, since

$$\max_{u,x,t} |\partial f(u)/\partial u - v| \lessapprox \max_{u} |\max_{u,x,t} |\partial f(u)/\partial u| = \alpha \qquad (31)$$

when v is close to $\partial f/\partial u$. Thus we define

$$\hat{g}(a, b) = \frac{1}{2}[g(a) + g(b) - \alpha(b - a)] = \hat{f}(a, b) - v\frac{a + b}{2}. \qquad (32)$$

Consider the left integral $\mathscr{S}_i$ in (23) (the other being similar). We approximate it as in (5) using the two-point Gauss rule. If we denote $\check{u}_i(t) := u(\check{x}_i(t), t)$, we have

$$\begin{aligned} \int_{t^n}^{t_{n+1}} \big(f(u) - vu\big)\big|_{x=\check{x}_i(t)}\,dt &= \int_{t^n}^{t_{n+1}} g(\check{u}_i(t))\,dt \\ &\approx \int_{t^n}^{t_{n+1}} \hat{g}(\check{u}_{i-}(t), \check{u}_{i+}(t))\,dt \approx \Delta t \sum_{\ell=1}^{2} \omega^{G_\ell}\hat{g}\big(\check{u}_{i-}(t^{G_\ell}), \check{u}_{i+}(t^{G_\ell})\big) \\ &= \Delta t \sum_{\ell=1}^{2} \omega^{G_\ell}\Big[\hat{f}\big(\check{u}_{i-}(t^{G_\ell}), \check{u}_{i+}(t^{G_\ell})\big) - v\big(\check{x}_i(t^{G_\ell}), t^{G_\ell}\big)\frac{\check{u}_{i-}(t^{G_\ell}) + \check{u}_{i+}(t^{G_\ell})}{2}\Big]. \qquad (33) \end{aligned}$$

Combining this with (23), our Eulerian-Lagrangian scheme can be expressed in general terms as

$$\begin{aligned}\bar{u}^{n+1}_{i+1/2} = \frac{1}{\Delta x}\sum_j \int_{\check{E}_{i+1/2}\cap E_{j+1/2}} \check{\mathscr{R}}(x,\bar{u}^n)\,dx \\ + \frac{\Delta t}{\Delta x}\sum_{\ell=1}^{2} \omega^{G_\ell}\Big[\hat{g}\big(\check{u}_{i-}(t^{G_\ell}),\check{u}_{i+}(t^{G_\ell})\big) - \hat{g}\big(\check{u}_{(i+1)-}(t^{G_\ell}),\check{u}_{(i+1)+}(t^{G_\ell})\big)\Big], \quad (34)\end{aligned}$$

which reduces to (7) in the purely Eulerian case, i.e., the choice $v = 0$.

The main task remaining is to find a proper reconstruction of right and left biased values of $\check{u}_i(t) = u(\check{x}_i(t),t)$ at the Gauss points in time along the v-trace. Unlike the case of Eulerian methods, the Lagrangian case has a path of integration that evolves in space, and so some special treatment is required. Along the v-trace where $x = \check{x}_i(t)$, the evolution of $\check{u}_i$ is governed by

$$\begin{aligned}\frac{d\check{u}_i(t)}{dt} &= \frac{du(\check{x}_i(t),t)}{dt} = u_x(\check{x}_i(t),t)\frac{d\check{x}_i}{dt} + u_t(\check{x}_i(t),t) \\ &= u_x(\check{x}_i(t),t)\,v(\check{x}_i(t),t) - (f(u))_x\big|_{(x,t)=(\check{x}_i(t),t)} := \check{F}(t,\check{u}_i(t)), \quad (35)\end{aligned}$$

where we used the definition of the v-trace (21) and the differential equation (1). The solution of this problem by a Runge-Kutta method requires an evaluation of the right-hand side function $\check{F}(t,\check{u}_i)$, and therefore an evaluation of the x-derivatives of f and u, at the intermediate Runge-Kutta times *along the v-trace path.* We use only the second-order A-stable scheme (14), since there are no intermediate slopes, but only $g^{(1)}$ and $g^{(2)}$ at times t^n and t^{n+1}. Recall that $g_i^{(1)}$ is computed explicitly and $g_i^{(2)}$ is eliminated from the system using (13). Moreover, because we use the NCE to handle the Gauss-points in time, we only need to describe how to compute $g_i^{(1)}$, i.e.,

$$g_i^{(1)} = \check{F}(t^n,\check{u}_i^n) = (\check{u}_i^n)_x\, v(\check{x}_i^n,t^n) - \big(f(\check{u}_i^n)\big)_x. \quad (36)$$

The derivatives within $g_i^{(1)}$ require the evaluation of three values of the solution u. To avoid numerical difficulties, we use a locally frozen velocity field, as was used in [13,14]. We illustrate the ideas in Fig. 4. The point x_i is v-traced to $\check{x}_i^n =: \check{x}_{i,0}^n$ at time t^n, tracing out the path $\mathscr{S}_i =: \mathscr{S}_{i,0}$. We then freeze the velocity locally, so that the nearby points $x_{i\pm1}$ are traced on parallel trajectories $\mathscr{S}_{i,\pm1} := \mathscr{S}_i \pm (\Delta x, 0)$. We care only about the points at time t^n, which are $\check{x}_{i,\pm1}^n := \check{x}_i^n \pm \Delta x$. The use of a locally frozen velocity ensures that our three points $\check{x}_{i,\ell}^n$, $\ell = -1,0,1$, are equally spaced. The solution at time t^n is reconstructed using these three points.

At the time t^n, we can explicitly evaluate $\check{u}_{i,\ell}^n \approx u(\check{x}_i^n,t^n)$, $\ell = -1,0,1$, using a WENO reconstruction. Assume for the moment that $\check{x}_{i,0}^n$ is not too close to a grid point, so we can simply take the same value for the right and left reconstructions. In this case, identify the index k such that $\check{x}_{i,0}^n \in E_{k+1/2}$ and reconstruct $\check{u}_{i,\ell}^n$ using $\bar{u}_{k+\ell-1/2}^n$, $\bar{u}_{k+\ell+1/2}^n$, and $\bar{u}_{k+\ell+3/2}^n$. Again to avoid the need for finding exact linear weights, we use the LPR-WENO reconstruction $\check{\mathscr{R}}(x,\bar{u}^n)$ of linear and quadratic polynomials as in [18,19] to find a third order reconstruction of the solution u. Once we have reconstructed $\check{u}_{i,\ell}^n = \check{\mathscr{R}}(\check{x}_{i,\ell}^n,\bar{u}^n)$, $\ell = -1,0,1$, we can then evaluate

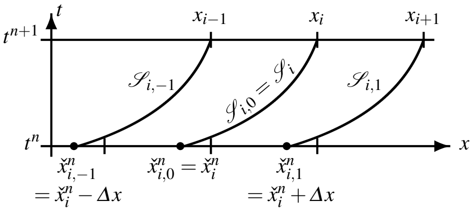

t
t^{n+1}
t^n
x
x_{i-1}
x_i
x_{i+1}
$\mathscr{S}_{i,-1}$
$\mathscr{S}_{i,0} = \mathscr{S}_i$
$\mathscr{S}_{i,1}$
$\check{x}^n_{i,-1}$
$= \check{x}^n_i - \Delta x$
$\check{x}^n_{i,0} = \check{x}^n_i$
$\check{x}^n_{i,1}$
$= \check{x}^n_i + \Delta x$

Fig. 4 A depiction of RK2 used to reconstruct the flux along the v-trace. The point x_i is v-traced to $\check{x}^n_{i,0} := \check{x}^n_i$ at time t^n, tracing out the path $\mathscr{S}_{i,0} := \mathscr{S}_i$. Freezing the velocity locally, the points $x_{i\pm1}$ are traced on parallel trajectories $\mathscr{S}_{i,\pm1} := \mathscr{S}_i \pm (\Delta x, 0)$. So, along the frozen velocity path, the points $x_{i\pm1}$ are traced to $\check{x}^n_{i,\pm1} := \check{x}^n_i \pm \Delta x$. The solution at time t^n is reconstructed using these three points.

$\check{f}^n_{i,\ell} = f(\check{u}^n_{i,\ell})$, as illustrated in Fig. 5. A standard WENO reconstruction (12) gives the derivatives of u and f, denoted $\check{u}^n_{i,x}$ and $\check{f}^n_{i,x}$.

If $\check{x}^n_{i,0}$ is close to a grid point (say, within 1E-10), then we need right and left values of the trace-back quantities, which can be constructed using two values of k. That is, when $\check{x}^n_{i,0} \approx x_j$, take $k = j$ for the right values and $k = j-1$ for the left values to define the reconstructions, which we denote $\check{\mathscr{R}}_\pm(x, \bar{u}^n)$.

We reiterate that in the second order A-stable scheme (14), we only need to compute $g_i^{(1)}$ defined in (36), which is computationally reasonable, since it is explicit. We do *not* need to approximate (35), because Runge-Kutta fluxes are *not* required at intermediate times. Limiting Runge-Kutta fluxes to times t^n and t^{n+1} (where the latter flux is removed using (13)) limits accuracy to third order. It is not clear how one might extend our ideas to computationally efficient implicit Eulerian-Lagrangian schemes of order higher than three, which would require solving (35) to get intermediate Runge-Kutta fluxes.

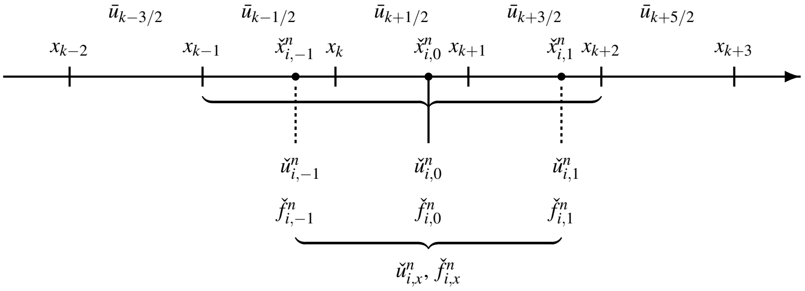


Fig. 5 Illustration of the reconstruction of $\check{u}^n_i = \check{\mathscr{R}}(\bar{u}^n, \check{x}^n_i)$, its derivative, and $\check{f}^n_{x,i}$ used in the evaluation of the Runge-Kutta flux $g_i^{(1)}$. Given the trace-back points $\check{x}^n_{i,\ell} = \check{x}^n_i + \ell\Delta x$, $\ell = -1, 0, 1$ (assuming these are not at grid points), we identify the index k so that $\check{x}^n_i \in [x_k, x_{k+1}]$. We use LPR-WENO to reconstruct the values $\check{u}^n_{i,\ell}$, which can then be substituted into $f(u)$ to obtain $\check{f}^n_{i,\ell}$. Finally, the three are combined to give an approximation to the derivatives.

The nonlinear structure of the scheme (34) is illuminated by expressing $\check{u}_i(t^{G_\ell})$ as in (17)–(18), which is

$$U^\ell_{i\pm}(\bar{u}^{n+1}) := \check{u}_{i\pm}(t^{G_\ell}) = \theta_\ell^2 \mathscr{R}_{i\pm}(\bar{u}^{n+1}) + r^\ell_{i\pm}, \tag{37}$$

where

$$r^\ell_{i\pm} = (1-\theta_\ell^2)\check{\mathscr{R}}_\pm(\check{x}^n_i, \bar{u}^n) + \Delta t(\theta_\ell - \theta_\ell^2) g^{(1)}_{i\pm} \tag{38}$$

and $g^{(1)}_{i\pm}$ is computed from (36). Then we have

$$\begin{aligned}\bar{u}^{n+1}_i = {} & \frac{1}{\Delta x}\sum_j \int_{\check{E}_{i+1/2}\cap E_{j+1/2}} \check{\mathscr{R}}(x,\bar{u}^n)\,dx \\ & + \frac{\Delta t}{\Delta x}\sum_{\ell=1}^{2} \omega^{G_\ell}\Big[\hat{f}\big(U^\ell_{i-}(\bar{u}^{n+1}),\, U^\ell_{i+}(\bar{u}^{n+1})\big) - \hat{f}\big(U^\ell_{(i+1)-}(\bar{u}^{n+1}),\, U^\ell_{(i+1)+}(\bar{u}^{n+1})\big) \\ & \qquad - v\big(\check{x}_i(t^{G_\ell}), t^{G_\ell}\big)\,\frac{U^\ell_{i-}(\bar{u}^{n+1}) + U^\ell_{i+}(\bar{u}^{n+1})}{2} \\ & \qquad + v\big(\check{x}_{i+1}(t^{G_\ell}), t^{G_\ell}\big)\,\frac{U^\ell_{(i+1)-}(\bar{u}^{n+1}) + U^\ell_{(i+1)+}(\bar{u}^{n+1})}{2}\Big].\end{aligned} \tag{39}$$

Solution by Newton's method can be found similarly to the Eulerian scheme.

In our scheme, the main component of the mass is transported in Lagrangian terms in an essentially exact manner. But because the trace velocity v is not exact, the flux correction step must be performed to account for flux crossing the traceline $\mathscr{S}_i$. Therefore, the CFL constraint is based on this step, and the constraint is [13]

$$\Delta t \le \frac{\Delta x}{\max|\partial g/\partial u|} = \frac{\Delta x}{\max|\partial f/\partial u - v|}. \tag{40}$$

So if indeed $v \approx \partial f/\partial u$, then the CFL constraint is significantly relaxed from that of a purely Eulerian scheme for which $v = 0$. Of course, our scheme is both Eulerian-Lagrangian and implicit, so the CFL time step restriction is not necessarily required to maintain stability.

We remark that the domain of dependence of the explicit EL-WENO3 scheme [13] is a trapezoidal region that contains a stencil with five values at time t^n. The implicit scheme only needs a stencil with three or four values at time t^n, depending on whether right and left values are computed. However, the unknowns at time t^{n+1} are fully coupled in the implicit scheme.

4 Some numerical results for scalar equations

We present several examples of our numerical scheme for scalar equations to test its accuracy and performance. In all cases, we define the trace velocity to approximate the characteristic velocity $\partial f/\partial u$. For simplicity, we take a locally constant trace velocity, i.e., for each grid point x_i, we fix the value of the trace velocity as v_i while tracing out the curves $\check{x}_i$. We begin with three test cases for the linear equation

$$u_t + (a(x,t)u)_x = 0, \tag{41}$$

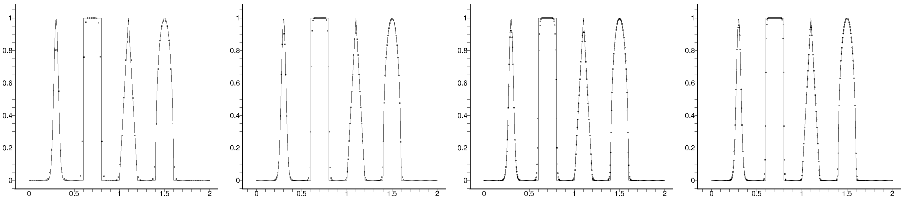
for which we specify only the velocity $a(x,t)$ and take the trace velocity $v_i = a(x_i,t^n)$. We then consider the nonlinear Burgers' and Buckley-Leverett equations, for which the flux $f(u;x,t)$ is a function of u only. In these nonlinear cases, we evaluate $u^n_{i,\text{up}} = \mathscr{R}_{i\pm}(\bar{u}^n_{i-3/2},\dots,\bar{u}^n_{i+3/2})$, where the right or left value is taken to correspond to the upstream value as determined by the Rankine-Hugoniot jump condition. We then take $v_i(x) = f'(u^n_{i,\text{up}})$.

We report errors in the usual discrete $L^1_{\Delta x}$-norm, which is

$$\|u^n_{\text{true}} - u^n_{\text{approx}}\|_{L^1_{\Delta x}} := \sum_i \big|u_{\text{true}}(x_{i+1/2},t^n) - \check{\mathscr{R}}(x_{i+1/2},\bar{u}^n)\big|\,\Delta x. \tag{42}$$

We also report discrete $L^\infty_{\Delta x}$-norm errors, which is the maximum of the absolute cell center point errors $\big|u_{\text{true}}(x_{i+1/2},t^n) - \check{\mathscr{R}}(x_{i+1/2},\bar{u}^n)\big|$.

4.1 Example 1, Shu's linear test with velocity $a = 1$



$m = 100$, CFL = 20.5 $m = 200$, CFL = 20.5 $m = 400$, CFL = 6.5 $m = 400$, CFL = 20.5

Fig. 6 Ex. 1, Shu's linear test. The trace-back points are perturbed randomly by at most $\pm 0.05\Delta x$. Results are shown at time $T = 2$ for various spatial resolutions $\Delta x = 2/m$ and CFL numbers.

We first present the standard *Shu's linear test*, which is simply the linear translate (i.e., velocity $a = 1$) of a complicated initial condition. An Eulerian-Lagrangian scheme gives nearly perfect results for this test (since there is no tracing error). We therefore modify the problem to test the performance of the flux reconstruction by randomly perturbing the trace-back points $\check{x}^n_i$ by an amount at most $\pm 0.05\Delta x$. We see excellent results in Fig. 6 for iEL-WENO3 at the final time $T = 2$, even for CFL numbers well above one. In fact, the scheme shows less error for the higher CFL 20.5 than 6.5 using $m = 400$ grid points. The reduction in error for longer time steps in Eulerian-Lagrangian schemes has been explained in [3].

4.2 Example 2, linear velocity $a(x,t) = \sin(t)$

We next 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 $u^0(x) = 0.5+\sin(\pi x)$. When the grid is sufficiently fine, at least third order convergence is observed in Table 1 for $\Delta t = 5.5\Delta x$. In fact, as we observe, we can increase the time step to $\Delta t = 10.5\Delta x$ and retain third order convergence, while slightly reducing the error.

Table 1 Ex. 2, linear velocity $a = \sin(t)$. Errors and convergence order at $T = 4$.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
$\Delta t = 5.5\Delta x$				
80	1.56029E-03	—-	6.22469E-03	—
160	3.63294E-04	2.10	2.42792E-03	1.36
320	8.39071E-05	2.11	9.15106E-04	1.41
640	1.30059E-05	2.69	2.29379E-04	2.00
1280	8.34722E-07	3.96	2.24449E-05	3.35
2560	4.51594E-08	4.21	8.79687E-07	4.67
$\Delta t = 10.5\Delta x$				
80	7.33851E-04	—	3.58390E-03	—
160	1.86974E-04	1.97	1.53673E-03	1.22
320	4.56080E-05	2.04	6.00667E-04	1.36
640	7.37190E-06	2.63	1.52435E-04	1.98
1280	4.38335E-07	4.07	1.31308E-05	3.54
2560	2.35534E-08	4.22	4.63970E-07	4.82

4.3 Example 3, linear velocity $a(x,t) = \sin(x)$

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

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

We see from Table 2 that the error for iEL-WENO3 as measured in $L^1_{\Delta x}$ converges at the optimal rate. However, when measured in $L^\infty_{\Delta x}$, the error converges to one power less than optimal. Similar results were seen in [14], where a discussion based on [28] explaining the loss of convergence in $L^\infty_{\Delta x}$ is also given.

We also show results for this problem using a fixed number of 10 time steps for various numbers of grid points m in Table 3. In spite of the CFL number increasing with m, we see nearly third order convergence in the $L^1_{\Delta x}$ norm.

4.4 Example 4, Burgers' equation

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

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

A shock will form at time $t = 1/\pi \approx 0.32$. To test convergence, we ran the computation over gradually refined grids up to time $T = 0.25$, before the shock develops. The numerical errors and convergence orders for the scheme are given in Table 4. The solution is only resolved when the grid is fine enough, and so we see relatively poor convergence of about second order when the grids are coarse. Under refinement, however, we pick up at least third order convergence, as expected. The $L^\infty_{\Delta x}$-norm is not as well behaved, but the rates also appear to be optimal in this norm.

Table 2 Ex. 3, linear velocity $a = \sin(x)$. Error and convergence order at $T = 1$.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
$\Delta t = 0.5\Delta x$				
20	4.60463E-02	—	3.93484E-02	—
40	7.49789E-03	2.62	1.43844E-02	1.45
80	1.28815E-03	2.54	4.15480E-03	1.79
160	2.04893E-04	2.65	1.09015E-03	1.93
320	2.96700E-05	2.79	2.75669E-04	1.98
640	4.06187E-06	2.87	6.68901E-05	2.04
$\Delta t = 6.5\Delta x$				
20	4.17349E-02	—	3.60500E-02	—
40	1.12621E-02	1.89	1.11598E-02	1.69
80	1.29803E-03	3.12	3.13010E-03	1.83
160	1.77153E-04	2.87	9.30164E-04	1.75
320	2.30593E-05	2.94	2.51254E-04	1.89
640	3.19292E-06	2.85	6.38763E-05	1.98
$\Delta t = 10.5\Delta x$				
20	4.17349E-02	—	3.60500E-02	—
40	1.12621E-02	1.89	1.11598E-02	1.69
80	5.11521E-03	1.14	3.48220E-03	1.68
160	4.06221E-04	3.65	8.78510E-04	1.99
320	3.75031E-05	3.44	2.42435E-04	1.86
640	3.83848E-06	3.29	6.22530E-05	1.96

Table 3 Ex. 3, linear velocity $a = \sin(x)$. Error and convergence order at $T = 1$ using 10 time steps.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
20	4.75674E-02	—	4.05602E-02	—
40	7.45595E-03	2.67	1.42508E-02	1.51
80	1.23089E-03	2.60	3.99872E-03	1.83
160	1.78506E-04	2.79	1.03140E-03	1.95
320	2.31440E-05	2.95	2.58593E-04	2.00
640	3.44227E-06	2.75	6.25534E-05	2.05

Table 4 Ex. 4, Burgers' equation with initial condition $0.5 + \sin(\pi x)$. Error and convergence order at $T = 0.25$.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
$\Delta t = \Delta x$				
20	1.46752E-02	—	7.80723E-02	—
40	3.20908E-03	2.19	1.12451E-02	2.80
80	9.72271E-04	1.72	3.97263E-03	1.50
160	2.19332E-04	2.15	1.43104E-03	1.47
320	4.77243E-05	2.20	5.63698E-04	1.34
640	7.57870E-06	2.65	1.47194E-04	1.94
1280	4.75282E-07	4.00	1.24983E-05	3.56
$\Delta t = 5\Delta x$				
320	2.54921E-05	—	3.10136E-04	—
640	3.80719E-06	2.74	5.43829E-05	2.51
1280	2.23658E-07	4.09	4.25859E-06	3.67
2560	1.30719E-08	4.10	4.91490E-07	3.12

Fig. 7 shows the solutions at $T = 3/(2\pi) \approx 0.48$ after the shock has formed. There is no numerical oscillation, and the scheme performs well.

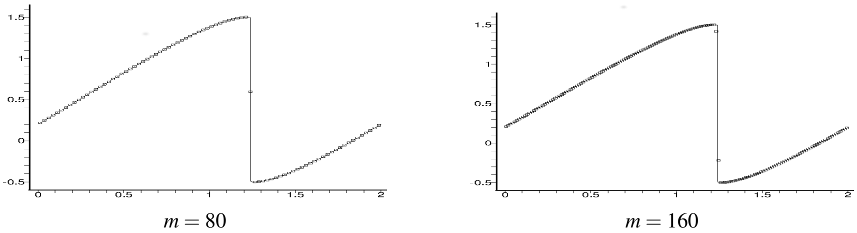


Fig. 7 Ex. 4, Burgers' equation with initial condition $0.5+\sin(\pi x)$. The solution at time $T = 3/(2\pi)$ using $m = 80$ and 160 grid elements.

In Figure 8, we show the propagation of a shock using two different choices for α in the Lax-Friedrichs flux function (6). The circles show the case of using $\alpha = \max\limits_{u,x,t} |\partial f(u)/\partial u|$. This is the value we suggested using above on the right-hand side of (31). The squares use a local value defined using the technically correct value for α given on the left-hand side of (31). It uses the local left and right reconstructed values of the solution, and at grid point x_i, it is given by

$$\alpha_{\text{local}} = \max\left(|(\partial f(\bar{u}^-)/\partial u - v|, |\partial f(\bar{u}^+)/\partial u - v|\right),$$

where $\bar{u}^\pm = \int_{\check{E}_{i\pm1/2}} \check{\mathscr{R}}(x,\bar{u}^n)\,dx/|\check{E}_{i\pm1/2}|$ is computed after tracing back to time t^n (cf., [29]). Both α and α_{local} are defined explicitly. The results demonstrate that the more carefully defined numerical flux indeed reduces the numerical diffusion, as is well known for explicit WENO schemes. However, we did not observe a significant change to the stability of the scheme, presumably because it does not affect the relaxed CFL limit (40).

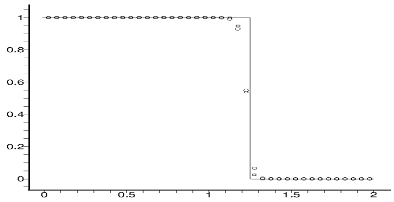


Fig. 8 Ex. 4, Burgers' equation with an initial jump at 0.25. The solution at time $T = 2$ using $m = 40$ grid elements. The circles use $\alpha = \max_{u,x,t} |\partial f(u)/\partial u|$, and the squares use the locally defined value α_{local}.

4.5 Example 5, Buckley-Leverett equation

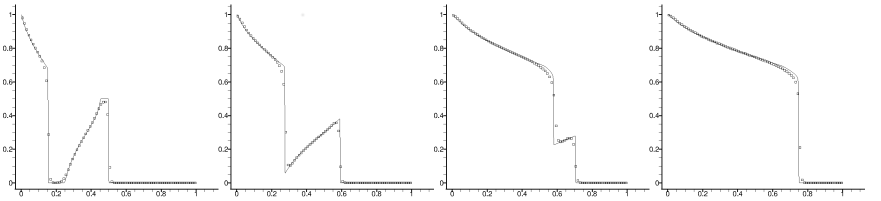
The final scalar example of (1) uses the Buckley-Leverett flux function

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

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

$$u_0(x) = \begin{cases} 1-20x & \text{for } 0 \le x \le 0.05, \\ 0.5 & \text{for } 0.25 \le x \le 0.4, \\ 0 & \text{otherwise,} \end{cases} \tag{43}$$

so the problem has two pulses that merge over time. We use $m = 80$ grid elements. The results are shown in Fig. 9. The schemes handle the merging of the two pulses quite well and reproduces the solution to good accuracy even on this relatively low resolution grid.



$t = 0.1$, shock and rarefaction formation

$t = 0.2$, initial pulse catches advanced pulse

$t = 0.4$, the two pulses merge

$t = 0.5$, the two pulses are fully merged

Fig. 9 Ex. 5, Buckley-Leverett. An interaction of shocks and rarefactions, resulting from the evolution of two initial pulses (43). The solid line is the reference solution, given by CWENO5 with a very small $\Delta x = 1/1280$ and $\Delta t = 1/15360$. The black squares are our iEL-WENO3 results using $m = 80$ and $\Delta t = 0.4\Delta x$.

4.6 Example 6, an advection-diffusion equation

We now consider the following advection-diffusion equation,

$$u_t + \big(f(u)\big)_x - \varepsilon u_{xx} = 0, \quad x \in \mathbb{R},\ t > 0, \tag{44}$$

where ε is assumed to be small, i.e., the equation is advection dominated. We propose to handle the diffusion term by writing the equation in the form

$$u_t + (vu)_x + \big(f(u) - vu - \varepsilon u_x\big)_x = 0 \quad \Longleftrightarrow \quad u_t + (vu)_x + \big(g(u)\big)_x = 0, \tag{45}$$

where

$$g(u) = f(u) - vu - \varepsilon u_x. \tag{46}$$

The computation of the mass integral in (25) would be computed as before. The computation of the flux correction step in Section 3.3 requires comment.

Although we do not recommend it, we could solve the problem as described in the previous sections. We would write (36) with the diffusion term, so we would have the explicit Runge-Kutta flux

$$g_i^{(1)} = \check{F}(t^n, \check{u}_i^n) = (\check{u}_i^n)_x\, v(\check{x}_i^n, t^n) - \big(f(\check{u}_i^n)\big)_x + \varepsilon(\check{u}_i^n)_{xx}, \tag{47}$$

which requires second order derivatives in space. However, this is used to compute an approximation to u, as in (37)–(38), which in turn becomes an argument of $g(u)$ within the scheme (39). That is, $g(u)$ defined by (46) now requires third derivatives in space. In order to maintain the accuracy of the scheme, we are required to find high order WENO reconstructions for third derivatives, which is not desired.

We suggest two ways to circumvent this problem of requiring third derivatives. First, we could simply reduce the temporal order of accuracy of the time quadrature rule, using the trapezoidal rule (i.e., two point Gauss-Lobatto rule) rather than a Gauss rule. This is the same as using the Crank-Nicolson (CN) time stepping procedure for the flux term $g(u)$. Unfortunately, the order of accuracy of the scheme will reduce to $\mathscr{O}(\Delta t^2 + \Delta x^3)$. In this case, however, most of the procedures that we described in Section 2 will not be needed, including using implicit Runge-Kutta, the NCE of Zennaro, and the trick (13).

A second, more accurate approach is to treat the advection as before using a two-point Gauss rule, but now handling the diffusion term with the two point Gauss-Lobatto rule, i.e., Crank-Nicolson time stepping only for diffusion. This scheme is accurate to order $\mathscr{O}(\varepsilon\Delta t^2 + \Delta t^3 + \Delta x^3)$, which remains small when ε is small. In fact, it retains third order accuracy in space and time provided only that $\varepsilon \le \Delta t$ and Δx and Δt are comparable, or if $\Delta x \gg \Delta t$. We verify this statment numerically in the remainder of this section.

No matter what we do, we will require accurate spatial derivatives. The derivative $u^{n+1}_{x,i}$ is needed at the grid point x_i at the advanced time t^{n+1} in (46), so we need to reconstruct implicitly using $u^{n+1}_{i-3/2}$, $u^{n+1}_{i-1/2}$, $u^{n+1}_{i+1/2}$, and $u^{n+1}_{i+3/2}$ to achieve a third order accurate derivative. This reconstruction is symmetric in space, so we do not get right and left values. However, this computation is used to handle the diffusion, so we do not require upstream values. Moreover, the reconstruction of $\breve{u}^n_{x,i,\ell}$, $\ell = -1,0,1$, needs to be at least fourth order accurate to evaluate the second derivative in (47). We actually use a stencil of five elements, i.e., a fifth order LPR-WENO reconstruction, since the fourth order reconstruction is not symmetric in space, which leads to some numerical instability.

4.6.1 Example 6a, linear test with velocity $a = 1$

We first take $f(u) = 1$ and consider the linear advection-diffusion equation

$$\begin{aligned} u_t + u_x - \varepsilon u_{xx} &= 0, & x &\in [0,2],\ t > 0, \\ u_0(x) &= \sin(\pi x), & x &\in [0,2],\ t = 0, \end{aligned}$$

where the exact solution is $u(x,t) = \sin(\pi(x-t))\exp(-\varepsilon\pi^2 t)$. As in Ex. 1, we randomly perturb the trace-back points $\breve{x}^n_i$, but now by an amount at most $\pm 0.5\Delta x^2$.

Using $\varepsilon = 1\text{E-}2$, Table 5 shows a comparison of the the full Crank-Nicolson (CN) scheme to the scheme using only CN for diffusion, i.e., implicit Runge-Kutta (RK) advection and CN diffusion. We see that the schemes attain third order convergence at time $T = 2$, since advection dominates the solution and the error is primarily spatial in our Eulerian-Lagrangian schemes. This is what one should expect from the truncation error estimates $\mathscr{O}(\Delta t^2 + \Delta x^3)$ and $\mathscr{O}(\varepsilon\Delta t^2 + \Delta t^3 + \Delta x^3)$ for the two schemes, since the Δx error term is much larger than the Δt error terms. Moreover, the solution is more accurate if the advection term is handled with the third order in time scheme. We remark that if $\varepsilon = 1\text{E-}1$, so that diffusion dominates and the CN temporal error becomes more pronounced, we see only second order convergence for both schemes, as one should expect.

Table 5 Ex. 6a, advection-diffusion with linear velocity $a = 1$. The trace-back points are perturbed randomly by at most $\pm 0.5\Delta x^2$. Errors and convergence order at $T = 2$ for $\varepsilon = 1\text{E-}2$, using $\Delta t = 5.5\Delta x$ and $18.5\Delta x$. Shown are the schemes using full CN and CN only for diffusion.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
	CN Advection and Diffusion				Implicit RK Advection and CN Diffusion			
				$\Delta t = 5.5\Delta x$				
20	6.51048E-02	—-	8.11891E-02	—-	4.78943E-02	—-	6.08510E-02	—-
40	1.42913E-02	2.19	2.28353E-02	1.83	9.82685E-03	2.29	1.66057E-02	1.87
80	1.98603E-03	2.85	4.75590E-03	2.26	1.34092E-03	2.87	3.32343E-03	2.32
160	2.60817E-04	2.93	8.98874E-04	2.40	1.79057E-04	2.90	6.54588E-04	2.34
320	3.40427E-05	2.94	1.63116E-04	2.46	2.35394E-05	2.93	1.02440E-04	2.68
640	4.23470E-06	3.01	2.36608E-05	2.79	2.92568E-06	3.01	1.46152E-05	2.81
1280	4.99978E-07	3.08	2.18933E-06	3.43	3.41818E-07	3.10	1.40140E-06	3.38
				$\Delta t = 18.5\Delta x$				
20	6.05842E-02	—-	7.54926E-02	—-	4.47388E-02	—-	5.63823E-02	—-
40	1.34068E-02	2.18	2.18070E-02	1.79	8.91021E-03	2.33	1.52693E-02	1.88
80	1.88338E-03	2.83	4.56025E-03	2.26	1.24216E-03	2.84	3.11963E-03	2.29
160	2.46143E-04	2.94	7.83357E-04	2.54	1.65256E-04	2.91	5.44148E-04	2.52
320	3.21541E-05	2.94	1.53646E-04	2.35	2.18219E-05	2.92	1.09517E-04	2.31
640	4.15167E-06	2.95	2.71617E-05	2.50	2.82466E-06	2.95	1.42737E-05	2.94
1280	4.60376E-07	3.17	2.06631E-06	3.72	3.09940E-07	3.19	1.47024E-06	3.28

4.6.2 Example 6b, Burgers' equation with diffusion

We now take $f(u) = u^2/2$ in (44). Exact solutions can be found using the Hopf-Cole transformation, and we take the exact solution

$$u(x,t) = \frac{-2\varepsilon\pi\cos(\pi x)\exp(-\varepsilon\pi^2 t)}{2+\sin(\pi x)\exp(-\varepsilon\pi^2 t)}.$$

We show the results in Table 6 for $\varepsilon = 1\text{E-}2$ and three time steps corresponding to different CFL numbers. The schemes are third order accurate when Δt is small compared to Δx (i.e., the two lower CFL numbers), but they begin to degenerate to second order accuracy when $\Delta t = 18.5\Delta x$ is large, i.e., when the temporal error $\mathscr{O}(\Delta t^2)$ dominates.

If we use $\varepsilon = 1\text{E-}6$, the spacial error dominates even for relatively long time steps. In that case, we see in Table 7 clean third order convergence. We again see that the solution is more accurate if the advection term is handled with the third order in time RK scheme. Moreover, if $\varepsilon = 1\text{E-}1$, so that diffusion is more pronounced, we see the convergence shown in Table 8. The fully CN scheme is pretty much second order throughout the test, as expected from the convergence order estimate $\mathscr{O}(\Delta t^2 + \Delta x^3)$. Not surprisingly, the Implicit RK Advection and CN Diffusion scheme, with its third

Table 6 Ex. 6b, Burgers' with diffusion. Errors and convergence order at $T = 2$ for $\varepsilon = $ 1E-2, using $\Delta t = 3.5\Delta x$, $8.5\Delta x$, and $18.5\Delta x$. Shown are the schemes using full CN and CN only for diffusion.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
	CN Advection and Diffusion				Implicit RK Advection and CN Diffusion			
				$\Delta t = 3.5\Delta x$				
20	2.98875E-03	—-	3.73615E-03	—-	2.01364E-03	—-	2.67277E-03	—-
40	5.46918E-04	2.45	9.53285E-04	1.97	3.77776E-04	2.41	6.86022E-04	1.96
80	7.66088E-05	2.84	1.97031E-04	2.27	5.26212E-05	2.84	1.37566E-04	2.32
160	8.94278E-06	3.10	2.68699E-05	2.87	6.14164E-06	3.10	1.88187E-05	2.87
320	9.27991E-07	3.27	2.39797E-06	3.49	6.28673E-07	3.29	1.65558E-06	3.51
640	8.68270E-08	3.42	1.94562E-07	3.62	5.88615E-08	3.42	1.37304E-07	3.59
1280	7.54444E-09	3.52	1.50364E-08	3.69	5.74102E-09	3.36	1.18598E-08	3.53
				$\Delta t = 8.5\Delta x$				
20	2.90243E-03	—-	3.68226E-03	—-	1.95648E-03	—-	2.63014E-03	—-
40	5.36768E-04	2.43	9.43958E-04	1.96	3.69172E-04	2.41	6.77315E-04	1.96
80	7.52341E-05	2.83	1.95973E-04	2.27	5.12916E-05	2.85	1.36428E-04	2.31
160	8.75831E-06	3.10	2.64808E-05	2.89	5.93501E-06	3.11	1.84876E-05	2.88
320	9.00873E-07	3.28	2.36003E-06	3.49	5.98859E-07	3.31	1.62637E-06	3.51
640	8.43090E-08	3.42	1.90104E-07	3.63	6.29279E-08	3.25	1.36493E-07	3.57
1280	8.82253E-09	3.26	1.52518E-08	3.64	8.09347E-09	2.96	1.42758E-08	3.26
				$\Delta t = 18.5\Delta x$				
20	2.75474E-03	—-	3.63156E-03	—-	1.86443E-03	—-	2.57152E-03	—-
40	5.22841E-04	2.40	9.39808E-04	1.95	3.55307E-04	2.39	6.69409E-04	1.94
80	7.25776E-05	2.85	1.93991E-04	2.28	4.80639E-05	2.89	1.33080E-04	2.33
160	8.23859E-06	3.14	2.57242E-05	2.91	5.42097E-06	3.15	1.75230E-05	2.92
320	8.34483E-07	3.30	2.24859E-06	3.52	6.44055E-07	3.07	1.45210E-06	3.59
640	1.04192E-07	3.00	1.75417E-07	3.68	9.84488E-08	2.71	1.68020E-07	3.11
1280	1.59902E-08	2.70	3.04428E-08	2.53	2.44106E-08	2.01	4.77672E-08	1.81

order in time advection computation $\mathscr{O}(\varepsilon\Delta t^2 + \Delta t^3 + \Delta x^3)$, shows nearly third order convergence for coarser grids, where space error dominates the temporal error.

Table 7 Ex. 6b, Burgers' with diffusion. Errors and convergence order at $T = 2$ for $\varepsilon = $ 1E-6, using $\Delta t = 18.5\Delta x$. Shown are the full CN scheme and the one using CN only for diffusion.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
	CN Advection and Diffusion				Implicit RK Advection and CN Diffusion			
				$\Delta t = 18.5\Delta x$				
20	1.44407E-07	——	2.31669E-07	——	1.03087E-07	——	1.73703E-07	——
40	2.47069E-08	2.55	5.03188E-08	2.20	1.68473E-08	2.61	3.51090E-08	2.31
80	3.36307E-09	2.88	7.51913E-09	2.74	2.25083E-09	2.90	5.06073E-09	2.79
160	4.26284E-10	2.98	9.71268E-10	2.95	2.84418E-10	2.98	6.48351E-10	2.96
320	5.34240E-11	3.00	1.22308E-10	2.99	3.56264E-11	3.00	8.15717E-11	2.99
640	6.68099E-12	3.00	1.53147E-11	3.00	4.45497E-12	3.00	1.02128E-11	3.00
1280	8.36551E-13	3.00	1.91558E-12	3.00	5.57082E-13	3.00	1.27758E-12	3.00

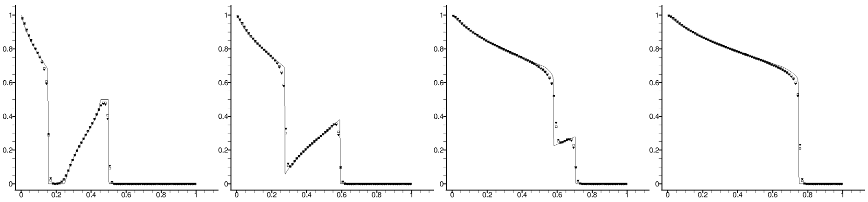
4.6.3 Example 6c, Buckley-Leverett equation with diffusion

In the final advection-diffusion test, we use the Buckley-Leverett flux (4.5) in (44), and the test case used in Ex. 5. We take $\varepsilon = $ 1E-6 and $m = 80$ grid elements. The

Table 8 Ex. 6b, Burgers' with diffusion. Errors and convergence order at $T = 2$ for $\varepsilon = 1\text{E-}1$, using $\Delta t = 3.5\Delta x$ and $8.5\Delta x$. Shown are the full CN scheme and the one using CN only for diffusion.

m	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order	$L^1_{\Delta x}$ error	order	$L^\infty_{\Delta x}$ error	order
	CN Advection and Diffusion				Implicit RK Advection and CN Diffusion			
				$\Delta t = 3.5\Delta x$				
20	3.03290E-03	—	3.85294E-03	—	1.78422E-03	—	3.58386E-03	—
40	6.28277E-04	2.27	7.10545E-04	2.43	3.73373E-04	2.25	4.54328E-04	2.97
80	8.79838E-05	2.83	1.03187E-04	2.78	4.35975E-05	3.09	6.07784E-05	2.90
160	1.65835E-05	2.40	1.83054E-05	2.49	7.17319E-06	2.60	9.84879E-06	2.62
320	3.53054E-06	2.23	3.18086E-06	2.52	1.40835E-06	2.34	1.55569E-06	2.66
640	7.93217E-07	2.15	6.88539E-07	2.20	3.05825E-07	2.20	3.59455E-07	2.11
1280	1.86282E-07	2.09	1.61667E-07	2.09	7.09005E-08	2.10	8.53825E-08	2.07
				$\Delta t = 8.5\Delta x$				
20	6.27758E-03	—-	6.18175E-03	—-	4.33423E-03	—-	6.88311E-03	—-
40	2.20397E-03	1.51	3.51287E-03	0.81	7.18026E-04	2.59	1.95399E-03	1.81
80	2.80481E-04	2.97	2.64041E-04	3.73	1.08048E-04	2.73	1.80663E-04	3.43
160	7.03043E-05	1.99	6.08731E-05	2.11	2.68221E-05	2.01	3.22019E-05	2.48
320	1.68773E-05	2.05	1.46753E-05	2.05	6.25699E-06	2.09	7.43728E-06	2.11
640	4.19442E-06	2.00	3.65180E-06	2.00	1.56984E-06	1.99	1.86472E-06	1.99
1280	1.04308E-06	2.00	9.07631E-07	2.00	3.90454E-07	2.00	4.64028E-07	2.00

results are shown in Fig. 10. The diffusion term is very small, so that the results are similar to Fig. 9. For our two schemes, i.e., CN advection and diffusion and implicit RK advection and CN diffusion, it can be seen that both capture the steep fronts extremely well. Moreover, the latter scheme provides a somewhat sharper front, since the advection is handled by a third order scheme in time.



$t = 0.1$, shock and rarefaction formation

$t = 0.2$, initial pulse catches advanced pulse

$t = 0.4$, the two pulses merge

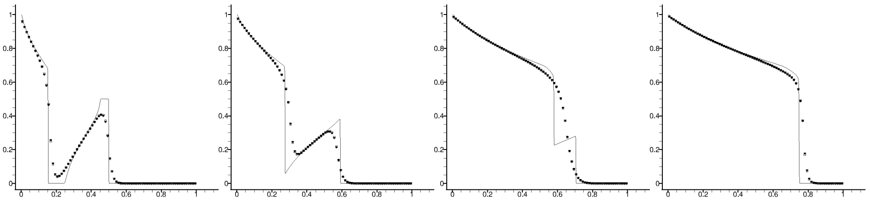
$t = 0.5$, the two pulses are fully merged

Fig. 10 Ex. 6c, Buckley-Leverett with diffusion. An interaction of shocks and rarefactions (43). The solid line is the reference solution without diffusion, given by CWENO5 with a very small $\Delta x = 1/1280$ and $\Delta t = 1/15360$. The solid gradients are the scheme using CN advection and diffusion, and the open squares are implicit RK advection and CN diffusion. Results use $\varepsilon = 1\text{E-}6$, $m = 80$ and $\Delta t = 0.4\Delta x$.

The physical diffusion is quite important in this problem. It smears the front a bit, as can be seen in Fig. 11, where diffusion is increased to $\varepsilon = 1\text{E-}2$.

5 Extension of iEL-WENO3 to the Euler system

The scalar scheme can be generalized to handle systems of nonlinear conservation laws. To simplify the presentation, we consider only the Euler system. For a poly-



$t = 0.1$, shock and rarefaction formation | $t = 0.2$, initial pulse catches advanced pulse | $t = 0.4$, the two pulses merge | $t = 0.5$, the two pulses are fully merged

Fig. 11 Ex. 6c, Buckley-Leverett with diffusion. An interaction of shocks and rarefactions (43). The solid line is the reference solution without diffusion, given by CWENO5 with a very small $\Delta x = 1/1280$ and $\Delta t = 1/15360$. The solid gradients are the scheme using CN advection and diffusion, and the open squares are implicit RK advection and CN diffusion. Results use $\varepsilon =$ 1E-2, $m = 80$ and $\Delta t = 0.4\Delta x$.

tropic gas, the energy is $E = p/(\gamma - 1) + \rho u^2/2$, where p, ρ, and u are the particle pressure, density, and velocity, and the adiabatic index $\gamma = (f+2)/f = 1.4$, where $f = 5$ is the number of degrees of freedom of each gas particle. The one-dimensional dynamics is described by the Euler equations

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}_x = 0 \quad \Longleftrightarrow \quad \mathbf{U}_t + \big(\mathbf{f}(\mathbf{U})\big)_x = 0, \tag{48}$$

where $\mathbf{U} = (\rho, \rho u, E)$.

5.1 Local characteristic decomposition

As is usual, the general WENO reconstruction procedures (but *not* the scheme itself) use a local characteristic decomposition to reduce oscillations. For an implicit scheme, one would expect an implicit local characteristic decomposition. We expand (48) into a system of the form $\mathbf{U}_t + A(\mathbf{U})\mathbf{U}_x = 0$, that is, into

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)u^2 & (3-\gamma)u & \gamma-1 \\ \frac{1}{2}(\gamma-1)u^3 - uh & h-(\gamma-1)u^2 & \gamma u \end{pmatrix} \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_x = 0, \tag{49}$$

where $h = (E+p)/\rho$ is the total specific enthalpy. Assuming that there exist $L(\mathbf{U})$ such that $\Lambda(\mathbf{U}) = L(\mathbf{U})A(\mathbf{U})L^{-1}(\mathbf{U})$, where $\Lambda(\mathbf{U})$ is a diagonal matrix, then

$$L(\mathbf{U})\mathbf{U}_t + \Lambda(\mathbf{U})L(\mathbf{U})\mathbf{U}_x = 0.$$

One would like to set $\mathbf{W} = L(\mathbf{U})\mathbf{U}$ and write this as $\mathbf{W}_t + \Lambda(\mathbf{W})\mathbf{W}_x \overset{?}{=} 0$. However, $\mathbf{W}_t = L_t\mathbf{U} + L\mathbf{U}_t$ and $\mathbf{W}_x = L_x\mathbf{U} + L\mathbf{U}_x$, and so we have

$$\mathbf{W}_t + \Lambda(\mathbf{W})\mathbf{W}_x = L_t\mathbf{U} + \Lambda L_x\mathbf{U} \not\equiv 0.$$

Therefore, we conclude that freezing A is necessary, i.e., an explicit, local characteristic decomposition is inevitable.

Fixing a base state $\mathbf{U}_*$, e.g., the solution at time t^n or its Roe averages (57), the explicit linearized equation can be diagonalized based on its eigenvalues. We have

$$\Lambda_* = L_* A(\mathbf{U}_*) L_*^{-1} = \begin{pmatrix} u_* - c_* & 0 & 0 \\ 0 & u_* & 0 \\ 0 & 0 & u_* + c_* \end{pmatrix}, \tag{50}$$

where $c_* = \sqrt{\gamma p_*/\rho_*}$ is the sound speed. With $\mathbf{W} = L_*\mathbf{U}$, the linearized and diagonalized equations are

$$L_*\mathbf{U}_t + \Lambda_* L_* \mathbf{U}_x = 0 \quad \Longleftrightarrow \quad \mathbf{W}_t + \Lambda_* \mathbf{W}_x = 0. \tag{51}$$

All WENO reconstructions are done in terms of the variables $\mathbf{W} = L_*\mathbf{U}$.

5.2 The iEL-WENO3 Scheme for the Euler System and a Roe Solver

For the scheme itself, we choose one fixed trace velocity v for all three conserved quantities ρ, ρu, and E, as was done in [13]. Then the full system (48), rewritten in terms of the diagonal matrix of trace velocities $v\mathbf{I}$, is

$$\mathbf{U}_t + \big(v\mathbf{U}\big)_x + \big(\mathbf{f}(\mathbf{U}) - v\mathbf{U}\big)_x = 0 \quad \Longleftrightarrow \quad \mathbf{U}_t + \big(v\mathbf{U}\big)_x + \big(\mathbf{g}(\mathbf{U})\big)_x = 0, \tag{52}$$

where $\mathbf{g}(\mathbf{U}) = \mathbf{f}(\mathbf{U}) - v\mathbf{U}$. For the implicit scheme, we solve the Euler system as in Section 3 in two main steps, for the mass integral at time t^n and the flux correction from $\mathbf{g}(\mathbf{U})$. For the mass integral, as in (25), we need a reconstruction operator. We simply use a vector version of $\check{\mathscr{R}}$, which is based on application of $\check{\mathscr{R}}$ to each of the three variables $\bar{\mathbf{W}}^n = L_*\bar{\mathbf{U}}^n$. It is defined by

$$\check{\boldsymbol{\mathscr{R}}}(x, \bar{\mathbf{W}}^n) = \big(\check{\mathscr{R}}(x, \bar{W}_1^n), \check{\mathscr{R}}(x, \bar{W}_2^n), \check{\mathscr{R}}(x, \bar{W}_3^n)\big).$$

Analogous to (34), we have

$$\begin{aligned} \bar{\mathbf{U}}_{i+1/2}^{n+1} &= \frac{1}{\Delta x} \sum_j L_*^{-1} \int_{\check{E}_{i+1/2} \cap E_{j+1/2}} \check{\boldsymbol{\mathscr{R}}}(x, L_*\bar{\mathbf{U}}^n)\, dx \\ &\quad + \frac{\Delta t}{\Delta x} \sum_{\ell=1}^{2} \omega^{G_\ell} \Big[\hat{\mathbf{g}}\big(\check{\mathbf{U}}_{i-}(t^{G_\ell}), \check{\mathbf{U}}_{i+}(t^{G_\ell})\big) - \hat{\mathbf{g}}\big(\check{\mathbf{U}}_{(i+1)-}(t^{G_\ell}), \check{\mathbf{U}}_{(i+1)+}(t^{G_\ell})\big)\Big]. \end{aligned} \tag{53}$$

We elaborate on the last terms above, i.e., on the flux correction step.

As in (34), we require right and left values at a reconstruction point. Rather than the Lax-Friedrichs flux, we use the numerical flux of a Roe solver for the flux correction functions

$$\mathbf{g}(\mathbf{U}) = \begin{pmatrix} \rho u - v\rho \\ \rho u^2 + p - v\rho u \\ u(E+p) - vE \end{pmatrix}. \tag{54}$$

The Roe numerical flux is defined to be

$$\hat{\mathbf{g}}(\mathbf{U}^-, \mathbf{U}^+) = \frac{1}{2}\Big[\mathbf{g}(\mathbf{U}^-) + \mathbf{g}(\mathbf{U}^+) - |A_{v,*}|(\mathbf{U}^- - \mathbf{U}^+)\Big], \tag{55}$$

wherein the arguments $\mathbf{U}^{\pm} = \left(\rho^{\pm}, u^{\pm}, E^{\pm}\right)$, are evolved along the v-traceline using the explicit, local characteristic decomposition (51), as noted above. To be more precise, we follow Section 3.3 and highlight differences. We have $\bar{\mathbf{W}}^n = L_* \bar{\mathbf{U}}^n$ at time t^n (explicitly), which are used in the reconstruction operator $\check{\boldsymbol{\mathscr{R}}}_{\pm}$ (defined analogously to $\check{\boldsymbol{\mathscr{R}}}$) to give point values $\check{\mathbf{W}}^n_{i\pm}$ and, therefore, $\check{\mathbf{U}}^n_{i\pm} = L_*^{-1} \check{\mathbf{W}}^n_{i\pm}$. These are used to define the vector of Runge-Kutta fluxes $\mathbf{g}^{(1)}_{i\pm}$ in (36) by the procedure described in Section 3.3. The Runge-Kutta flux $\mathbf{g}^{(2)}_{i\pm}$ is given implicitly by (13), which requires a vector version of the operator $\mathscr{R}_{i\pm}$ to obtain (implicitly) $\mathbf{U}^{n+1}_{i\pm}$, i.e., $\mathbf{U}^{n+1}_{i\pm} = L_*^{-1} \boldsymbol{\mathscr{R}}_{i\pm}(L_* \mathbf{U}^{n+1})$. Finally, the NCE gives the values of $\check{\mathbf{U}}$ at the Gauss times needed in (53).

The matrix $|A_{v,*}|$ must be defined carefully. Before defining $|A_{v,*}|$, we recall that Roe [27] defined his solver for (48) in the form (49) with the constant matrix $A(\mathbf{U}, \mathbf{V})$ defined as

$$A(\mathbf{U}_*^-, \mathbf{U}_*^+) = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma - 3)\bar{u}_*^2 & (3-\gamma)\bar{u}_* & \gamma - 1 \\ \frac{1}{2}(\gamma - 1)\bar{u}_*^3 - \bar{u}_* \bar{h}_* & \bar{h}_* - (\gamma - 1)\bar{u}_*^2 & \gamma \bar{u}_* \end{pmatrix}, \tag{56}$$

where, for the base state $\mathbf{U}_*^{\pm}$ having right and left values, the Roe averages are

$$\bar{u}_* = \frac{\sqrt{\rho_*^-} u_*^- + \sqrt{\rho_*^+} u_*^+}{\sqrt{\rho_*^-} + \sqrt{\rho_*^+}}, \quad \bar{h}_* = \frac{\sqrt{\rho_*^-} h_*^- + \sqrt{\rho_*^+} h_*^+}{\sqrt{\rho_*^-} + \sqrt{\rho_*^+}}, \tag{57}$$

and

$$h_*^{\pm} = \frac{E_*^{\pm} + p_*^{\pm}}{\rho_*^{\pm}}.$$

The matrix A is designed to find the correct shock speeds, and it satisfies the following Roe's conditions:

1. $\mathbf{f}(\mathbf{U}) - \mathbf{f}(\mathbf{V}) = A(\mathbf{U}, \mathbf{V})(\mathbf{U} - \mathbf{V})$;
2. $A(\mathbf{U}, \mathbf{V}) \to \dfrac{D\mathbf{f}}{D\mathbf{U}}(\mathbf{U})$ as $\mathbf{V} \to \mathbf{U}$;
3. $A(\mathbf{U}, \mathbf{V})$ has only real eigenvalues;
4. $A(\mathbf{U}, \mathbf{V})$ has a complete system of eigenvectors, i.e., it is diagonalizable.

Conditions 3 and 4 allow the decomposition (50) for $A(\mathbf{U}_*^-, \mathbf{U}_*^+)$, i.e.,

$$\Lambda_* := L_* A(\mathbf{U}_*^-, \mathbf{U}_*^+) L_*^{-1} = \begin{pmatrix} \bar{u}_* - \bar{c}_* & 0 & 0 \\ 0 & \bar{u}_* & 0 \\ 0 & 0 & \bar{u}_* + \bar{c}_* \end{pmatrix}, \tag{58}$$

where $\bar{c}_* = \sqrt{(\gamma - 1)(\bar{h}_* - \bar{u}_*^2/2)}$. The matrix $|A_*| = L_*^{-1} |\Lambda_*| L_*$, where $|\Lambda_*|$ is the same as Λ_* except that one takes the absolute value of the entries.

In analogy, we define $|A_{v,*}|$ to be

$$|A_{v,*}| = L_*^{-1} \begin{pmatrix} |\bar{u}_* - \bar{c}_* - v| & 0 & 0 \\ 0 & |\bar{u}_* - v| & 0 \\ 0 & 0 & |\bar{u}_* + \bar{c}_* - v| \end{pmatrix} L_*, \tag{59}$$

and we note that when $v = 0$, $|A_*| = |A_{0,*}|$. In fact, we claim that we have actually defined A_v as

$$A_v(\mathbf{U},\mathbf{V}) = A(\mathbf{U},\mathbf{V}) - v\mathbf{I}, \tag{60}$$

and that it satisfies Roe's conditions. For Roe's first condition,

$$\begin{aligned}\mathbf{g}(\mathbf{U}) - \mathbf{g}(\mathbf{V}) &= (\mathbf{f}(\mathbf{U}) - \mathbf{f}(\mathbf{V})) - V(v)(\mathbf{U}-\mathbf{V}) \\ &= A(\mathbf{U},\mathbf{V})(\mathbf{U}-\mathbf{V}) - v\mathbf{I}(\mathbf{U}-\mathbf{V}) = A_v(\mathbf{U},\mathbf{V})(\mathbf{U}-\mathbf{V}).\end{aligned}$$

For the second condition, we have

$$A_v(\mathbf{U},\mathbf{V}) = A(\mathbf{U},\mathbf{V}) - v\mathbf{I} \to \frac{D\mathbf{f}}{D\mathbf{U}}(\mathbf{U}) - v\mathbf{I} = \frac{D\mathbf{g}}{D\mathbf{U}}(\mathbf{U}), \quad \text{as } \mathbf{V} \to \mathbf{U},$$

since $\dfrac{D\mathbf{g}(\mathbf{U})}{D\mathbf{U}} = \dfrac{\mathbf{f}(\mathbf{U}) - V(v)\mathbf{U}}{D\mathbf{U}} = \dfrac{D\mathbf{f}}{D\mathbf{U}} - v\mathbf{I}$. The third and fourth conditions are satisfied when $v = 0$ and continue to hold for a perturbation by a multiple of the identity, $-v\mathbf{I}$. That is,

$$\begin{aligned}L_* A_{v,*} L_*^{-1} &= L_*(A - vI)L_*^{-1} = \Lambda - vI \\ &= \begin{pmatrix} \bar{u}_* - \bar{c}_* - v & 0 & 0 \\ 0 & \bar{u}_* - v & 0 \\ 0 & 0 & \bar{u}_* + \bar{c}_* - v \end{pmatrix} =: \Lambda_{v,*}.\end{aligned} \tag{61}$$

Thus $A_{v,*}$ as defined in (60) satisfies all four of Roe's conditions, and its absolute value is properly defined in (59). This result shows that A and $A_{v,*}$ can be diagonalized simultaneously. It also suggests that using the same v-trace velocity for each conserved quantity in (52) is the only reasonable choice if we follow Roe's approach [27]; otherwise, we will completely change the eigenstructure of the problem.

We remark that because the matrix $A_{v,*}(\mathbf{U},\mathbf{V})$ is (locally) constant, the computation of the Jacobian matrix is simplified when solving (52) implicitly using Newton's method.

We used $|A|$ in our explicit Eulerian-Lagrangian WENO scheme [13], and it gave good results. However, we observed computationally that using $|A_{v,*}|$ in (55) is crucial, since otherwise a non-entropy solution may be obtained.

We need to make a choice for the trace velocity v. Following [13], we take v to approximate the velocity u, since it is the most neutral single trace velocity v for the entire system. However, this is also the velocity of the mass particles, which must be correctly evolved in a physical sense to have an accurate solution. The iEL-WENO3 scheme should be stable as long as the two untraced characteristic velocities $u \pm c$ do not travel across the space-time trace-back region; that is, the velocity $\pm c$ relative to the trace velocity $v = u$ determines the relaxed CFL constraint, which is about $\Delta t = h/|c|$.

6 Some numerical results for the Euler system

In this section, we consider several standard test problems for the Euler system. In all our tests, we again take a locally constant trace velocity v_i while tracing out the curves $\check{x}_i$ for the grid point x_i. For the grid point x_i, for the local characteristic variables $\bar{\mathbf{W}}^n$ in (51), we reconstruct $\mathbf{W}^n_{i\pm} = \check{\mathscr{R}}_{i\pm}(\bar{\mathbf{W}}^n)$. After transforming back to the variables $\mathbf{U}^n_{i\pm}$, we set $v_i = \bar{u}_{*,i}$, the reference Roe average (57) at x_i.

6.1 Example 7: A smooth problem for the Euler equations

In this example, the initial condition is set to be $\rho(x,0) = 1+0.2\sin(\pi x)$, $u(x,0) = 1$, and $p(x,0)$ is either 1 or 0.01, with 2-periodic boundary conditions. The exact solution is

$$\rho(x,t) = 1+0.2\sin(\pi(x-t)), \quad u=1, \quad p = 1 \text{ or } 0.01.$$

The sound speed is given by

$$c = \sqrt{\frac{1.4p}{\rho}} = \begin{cases} 1.32 & \text{when } p=1, \\ 0.132 & \text{when } p = 0.01, \end{cases}$$

and so the spread of the characteristic speeds is much less in the latter case. Because we use only a single trace velocity $v = u$, it is the spread of the characteristics that determine the limits of the time step. The usual CFL time step is

$$\Delta t_{\text{CFL}} = \frac{h}{u+c} = \begin{cases} 0.43h & \text{when } p=1, \\ 0.88h & \text{when } p = 0.01h, \end{cases}$$

but the relaxed value is

$$\Delta t_{\text{CFL,relaxed}} = \frac{h}{c} = \begin{cases} 0.76h & \text{when } p=1, \\ 7.56h & \text{when } p = 0.01, \end{cases}$$

We compute the solution up to time $T = 2$. The numerical errors and convergence orders for the density are given in Table 9. The scheme achieves third order convergence. When $p = 1$, we use $\Delta t = 0.6h = 1.4\Delta t_{\text{CFL}}$, and when $p = 0.01$, we use $\Delta t = 3.75h = 4.3\Delta t_{\text{CFL}}$.

6.2 Example 8: Riemann problems for the Euler equations

For this series of one-dimensional shock tube tests, we specify a discontinuous initial condition, written in terms of the primitive variables ρ, u, and p. As is typical, we only report the density ρ; the other variables show comparable accuracy. The one-dimensional shock tube test of Sod uses the initial condition

$$\rho,u,p = \begin{cases} \rho_l = 1,\ u_l = 0,\ p_l = 1, & \text{for } x < 1/2, \\ \rho_r = 1/8,\ u_r = 0,\ p_r = 1/10, & \text{for } x > 1/2, \end{cases}$$

Table 9 Ex. 7, Euler equation with $c = 1.32$. Error and convergence order for the density at $T = 2$.

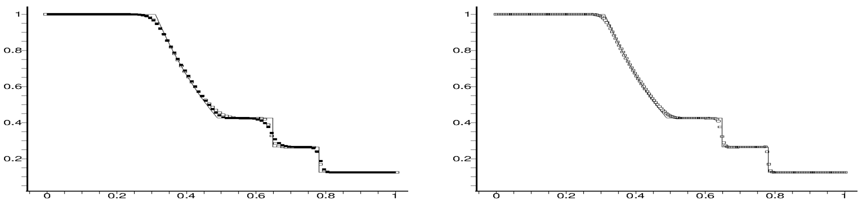
m	L_h^1 error	order	L_h^∞ error	order
$p = 1$, $c = 1.32$, $\Delta t = 0.6h$				
20	2.71096E-02	——	2.65181E-02	——
40	8.26686E-03	1.71	1.11980E-02	1.24
80	1.99465E-03	2.05	4.46671E-03	1.33
160	4.46184E-04	2.16	1.58934E-03	1.49
320	5.99188E-05	2.90	3.71836E-04	2.10
640	4.04820E-06	3.89	3.95759E-05	3.23
$p = 0.01$, $c = 0.132$, $\Delta t = 3.75h$				
20	1.89069E-02	——	3.13287E-02	——
40	3.98620E-03	2.24583	8.36638E-03	1.90481
80	6.30228E-04	2.66107	1.35649E-03	2.62472
160	8.76242E-05	2.84647	3.94738E-04	1.78092
320	9.19777E-06	3.25197	8.91172E-05	2.14712
640	5.25718E-07	4.12892	5.75380E-06	3.95312

and the test of Lax uses the initial condition

$$\rho, u, p = \begin{cases} \rho_l = 0.445,\ u_l = 0.698,\ p_l = 3.528, & \text{for } x < 1/2, \\ \rho_r = 0.5,\ u_r = 0,\ p_r = 0.571, & \text{for } x > 1/2. \end{cases}$$

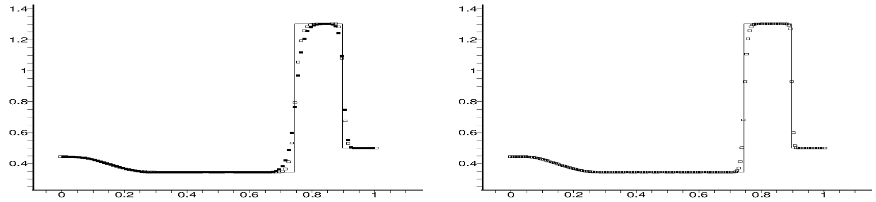
The iEL-WENO3 results in Figs. 12–13 show the effect of grid refinement for these two test problems. Indeed the solution improves when 200 elements are used versus 100. The graphs for 100 elements show the iEL-WENO3 results as open squares and WENO3 results as closed squares. The iEL-WENO results use a time step longer than the CFL limit. For Sod, we use $\Delta t = 2\Delta t_{\mathrm{CFL}}$; for Lax, we use $\Delta t = 2.8\Delta t_{\mathrm{CFL}}$. The WENO3 results use $\Delta t = 0.9\Delta t_{\mathrm{CFL}}$. So the iEL-WENO3 scheme can use 2 to 2.8 times the CFL time step, while also showing less numerical diffusion than WENO3 for both test problems.

Some of the improvement in the time step is due to the use of Lagrangian coordinates and some is due to the implicitness of the scheme. To assess the difference between these two, we ran the examples using trace velocity $v = 0$; that is, we ran a purely Eulerian scheme, so only the implicitness of the scheme manifests itself. Good results were achieved for time steps up to about 1.25 CFL for Sod's test and 2.1 CFL for Lax's test.



$m = 100$ $m = 200$

Fig. 12 Ex. 8, Sod 1-D shock tube test. The density profile is shown as open squares for iEL-WENO3 with $v = u$ at time $T = 0.16$ using $\Delta t = 2\Delta t_{\mathrm{CFL}}$ and $m = 100$ and 200 grid elements. When $m = 100$, we also show WENO3 results using $0.9\Delta t_{\mathrm{CFL}}$ as closed squares.



$m = 100$ $m = 200$

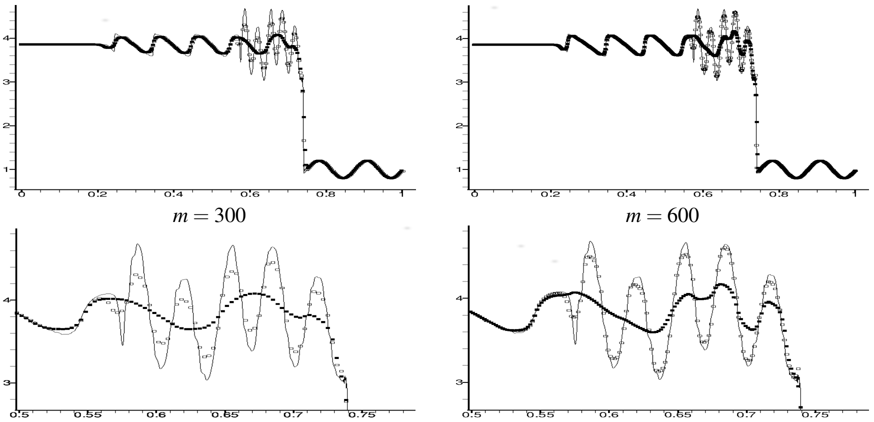
Fig. 13 Ex. 8, Lax 1-D shock tube test. The density profile is shown as open squares for iEL-WENO3 with $v = u$ at time $T = 0.16$ using $\Delta t = 2.8\Delta t_{\text{CFL}}$ and $m = 100$ and 200 grid elements. When $m = 100$, we also show WENO3 results using $0.9\Delta t_{\text{CFL}}$ as closed squares.

6.3 Example 9: Shu and Osher's shock interaction with entropy waves

In our next example, we consider the challenging test case of Shu and Osher [30], in which a Mach 3 shock interacts with entropy sine waves in the density. We scale the problem to the domain $(0,1)$, and the initial conditions are

$$\rho, u, p = \begin{cases} \rho_l = 3.857143,\ u_l = 2.629369,\ p_l = 10.333333, & \text{for } 0 < x < 1/10, \\ \rho_r = 1 + 0.2\sin(5(10x - 5)),\ u_r = 0,\ p_r = 1, & \text{for } 1/10 \le x < 1. \end{cases}$$

We use $\Delta t = 0.4\Delta x$ and compute to the final time $T = 0.18$. Using the grid sizes $m = 300$ and $m = 600$ grid elements, the results appear in Fig. 14. Compared to WENO3, we see that iEL-WENO3 has a much better ability to resolve the sine waves.



$m = 300$ $m = 600$

$m = 300$, zoom $m = 600$, zoom

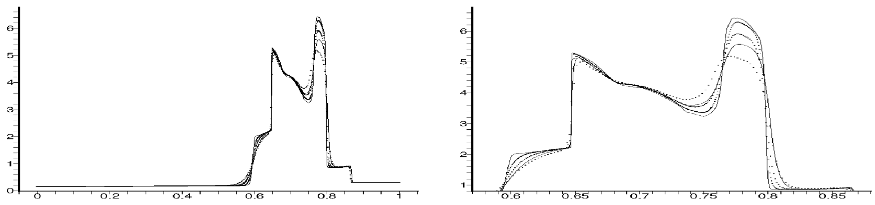
Fig. 14 Ex. 9, Shu and Osher's test with a shock interacting with entropy waves. The density profile at time $T = 0.18$ using $m = 300$ and $m = 600$ grid elements, with a magnified view of the results shown on the bottom. Shown are results for iEL-WENO3 (open squares) and WENO3 (solid squares) and a fine resolution reference solution (solid line).

6.4 Example 10: Woodward and Colella's double blast test

Our final one-dimensional example of the Euler system is the double blast test of Woodward and Colella, which uses the initial condition

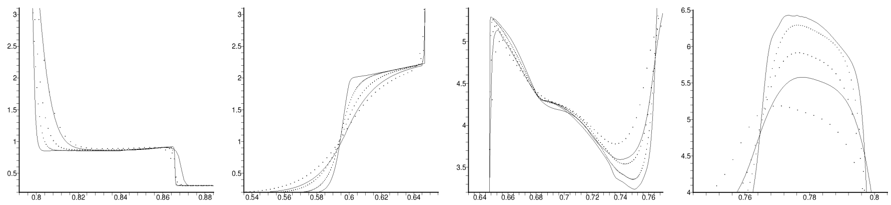
$$\rho, u, p = \begin{cases} \rho_l = 1,\ u_l = 0,\ p_l = 1000, & \text{for } x < 1/10, \\ \rho_m = 1,\ u_m = 0,\ p_m = 1/100, & \text{for } 1/10 < x < 9/10, \\ \rho_r = 1,\ u_r = 0,\ p_r = 100, & \text{for } 9/10 < x. \end{cases}$$

This is a rather challenging problem, because the two blast waves go through each other. Nevertheless, reasonably good results are obtained by our iEL-WENO3 scheme using $m = 400$, 800, and 1600 grid elements, as shown in Fig. 15. The reference solution was computed by a MUSCL scheme with grid size $m = 4000$. The results are compared to a WENO3 scheme. In Fig. 16, we significantly zoom the solution. One can see clearly that iEL-WENO3 shows very accurate resolution over the entire profile, and that it is generally better than WENO3 for the same resolution $m = 800$. Of particular note, iEL-WENO3 matches the solution near $x = 0.69$ in Zoom 2 for all three grids, while the WENO3 solution is too low.



$m = 400, 800, 1600$ $m = 400, 800, 1600$, zoom

Fig. 15 Ex. 10, Woodward and Colella's double blast test. The density profile at time $T = 0.038$ of iEL-WENO3 using $m = 400$, 800, and 1600 grid elements (black squares or dots) and WENO3 using $m = 800$ (diffused, fine line), with a zoomed view of the results shown on the right. The fine resolution reference solution is the sharp, solid line.



Zoom 1 Zoom 2 Zoom 3 Zoom 4

Fig. 16 Ex. 10, Woodward and Colella's double blast test. Magnified views of the density profile at time $T = 0.038$ using $m = 400, 800, 1600$ (iEL-WENO3, black squares or dots) and $m = 800$ (WENO3, diffused, fine line) grid elements. The fine resolution reference solution is the sharp, solid line.

7 Conclusions

We have presented a new, formally third order, implicit Weighted Essentially Non-Oscillatory (iWENO3) finite volume scheme for solving systems of nonlinear conservation laws. The special choice (14) of two-stage, implicit Runge-Kutta time integrator allowed us to drastically simplify the computation of the intermediate Runge-Kutta fluxes, since we could eliminate one using (13) and explicitly compute the other. This led to a computationally reasonable scheme that could be generalized to Lagrangian coordinates.

We generalized our Eulerian scheme to define an implicit, Eulerian-Lagrangian WENO (iEL-WENO) scheme consisting of two main steps. The first accounts for particles being transported within a grid element in a Lagrangian sense along the particle paths. Since this particle velocity is unknown (in a nonlinear problem), a fixed tace velocity v was used, and the equation was rewritten as

$$u_t + \big(f(u)\big)_x = 0 \quad \Longleftrightarrow \quad u_t + (vu)_x + \big(f(u) - vu\big)_x = 0.$$

A space-time integration (23) accounted for mass particles being transported within the velocity field v. The particles within a grid element E were traced back in time to $\check{E}$ where they originated from, and so the transported mass can be quantified. This part of the computation, $u_t + (vu)_x$, is essentially exact up to the accuracy of the WENO reconstructions used.

The second step of the scheme, the flux correction, accounts for the inaccuracy of the trace velocity v by computing the flux of particles crossing the incorrect tracelines bounding the space-time region, due to the term $\big(f(u) - vu\big)_x$. This computation is similar to what is done in an Eulerian scheme, except that the spatial path along which time evolves itself changes with time. By our choice of Runge-Kutta method, no higher order derivatives of the flux need be computed, neither to define the method nor to implement Newton's method. The CFL condition is relaxed when v is chosen to approximate the characteristic velocity.

For the Euler system, a new Roe solver (55) was developed to account for the Lagrangian tracings, which could be useful even for explicit EL-WENO schemes.

Numerical results show that very long time steps can be taken for linear problems, up to even 15 times the Eulerian CFL limit. We also saw good accuracy compared to standard WENO3. The iEL-WENO3 is both less numerically diffusive and can take longer time steps, on the order of about 2 to 3 times longer than WENO3 for challenging nonlinear scalar problems and Euler systems.

Two simple extensions of the scheme to the advection-diffusion equation were given in Section 4.6. When advection dominates, both schemes retain third order accuracy. An advection-diffusion equation must be solved using an implicit method (otherwise the restrictive parabolic CFL constraint $\Delta t \leq c\Delta x^2$ results). Since the main computational cost of solving implicit problems is in the nonlinear solver, the advection-diffusion equation can be solved nearly as efficiently using explicit or implicit WENO schemes. Our new iEL-WENO3 provides an efficient third order scheme that can capture sharp fronts extremely well.

References

1. Alexander, R.: Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. SIAM J. Numer. Anal. **14**(6), 1006–1021 (1977)
2. Arbogast, T., Huang, C.: A fully mass and volume conserving implementation of a characteristic method for transport problems. SIAM J. Sci. Comput. **28**(6), 2001–2022 (2006)
3. Arbogast, T., Wang, W.: Convergence of a fully conservative volume corrected characteristic method for transport problems. SIAM J. Numer. Anal. **48**(3), 797–823 (2010)
4. Arbogast, T., Wheeler, M.F.: A characteristics-mixed finite element method for advection dominated transport problems. SIAM J. Numer. Anal. **32**, 404–424 (1995)
5. Cadiou, A., Tenaud, C.: Implicit WENO shock capturing scheme for unsteady flows. Application to one-dimensional Euler equations. Int. J. Numer. Meth. Fluids **45**, 197–229 (2004)
6. Chen Y Yang S, Y.J.: Implicit weighted essentially non-oscillatory schemes for the incompressible Navier-Stokes equations. Internat. J. Numer. Methods Fluids **31**, 747–765 (1999)
7. Douglas Jr., J., Russell, T.F.: Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. SIAM J. Numer. Anal. **19**, 871–885 (1982)
8. Gottlieb, S., Mullen, J.S.: An implicit weno scheme for steady-state computation of scalar hyperbolic equations. In: K.J. Bathe (ed.) Computational Fluid and Solid Mechanics 2003, vol. 2, pp. 1946–1950. Elsevier (2003)
9. Gottlieb, S., Mullen, J.S., Ruuth, S.J.: A fifth order flux implicit WENO method. J. Sci. Comput. **27**(1–3), 271–287 (2006)
10. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high-order accurate essentially nonoscillatory schemes III. J. Comput. Phys. **71**(2), 231–303 (1987)
11. Harten, A., Osher, S.: Uniformly high-order accurate nonoscillatory schemes I. SIAM J. Numer. Anal. **24**(2), 279–309 (1987)
12. Hsieh, T.J., Wang, C.H., Yang, J.Y.: Simulation of multiple shock-shock interference using implicit anti-diffusive WENO schemes. Internat. J. Numer. Methods Fluids **62**(2), 138–165 (2010)
13. Huang, C.S., Arbogast, T.: An Eulerian-Lagrangian WENO scheme for nonlinear conservation laws. Numer. Meth. Partial Diff. Eqns. **33**(3), 651–680 (2017). DOI 10.1002/num.22091
14. Huang, C.S., Arbogast, T., Hung, C.H.: A semi-Lagrangian finite difference WENO scheme for scalar nonlinear conservation laws. J. Comput. Phys. **322**, 559–585 (2016). DOI 10.1016/j.jcp.2016.06.027
15. Huang, C.S., Arbogast, T., Qiu, J.: An Eulerian-Lagrangian WENO finite volume scheme for advection problems. J. Comput. Phys. **231**(11), 4028–4052 (2012). DOI 10.1016/j.jcp.2012.01.030
16. Jiang, G.S., Shu, C.W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**, 202–228 (1996)
17. Kennedy, C., Carpenter, M.: Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review. Tech. Rep. NASA/TM–2016–219173, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia (2016)
18. Levy, D., Puppo, G., Russo, G.: Central WENO schemes for hyperbolic systems of conservation laws. Math. Model. Numer. Anal. **33**, 547–571 (1999)
19. Levy, D., Puppo, G., Russo, G.: Compact central WENO schemes for multidimensional conservation laws. SIAM J. Sci. Comput. **22**(2), 656–672 (2000)
20. Liu, X.D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. **115**, 200–212 (1994)
21. Osher, S., Shu, C.W.: High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. SIAM. J. Numer. Anal. **28**, 907–922 (1991)
22. Pironneau, O.: On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. Numer. Math. **38**, 309–332 (1981/82)
23. Qiu, J., Shu, C.W.: On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes. J. Comput. Phys. **183**, 187–209 (2002)
24. Qiu, J.M., Christlieb, A.: A conservative high order semi-Lagrangian WENO method for the Vlasov equation. J. Comput. Phys. **229**, 1130–1149 (2010)
25. Qiu, J.M., Shu, C.W.: Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow. J. Comput. Phys. **230**(4), 863–889 (2011)
26. Qiu, J.M., Shu, C.W.: Conservative semi-Lagrangian finite difference WENO formulations with applications to the Vlasov equation. Communications in Comput. Phys. **10**, 979–1000 (2011)

27. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys. **43**, 357–372 (1981)
28. Shu, C.W.: Numerical experiments on the accuracy of ENO and modified ENO schemes. J. Sci. Comput. **5**, 127–149 (1990)
29. Shu, C.W.: High order numerical methods for time dependent Hamilton-Jacobi equations. In: Mathematics and Computation in Imaging Science and Information Processing, *Lecture Notes Series*, vol. 11, chap. 2, pp. 47–91. Inst. for Math. Sci., Nat'l Univ. Singapore (2007)
30. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
31. Črnjarić Žic, N., Crnković, B.: High order accurate semi-implicit WENO schemes for hyperbolic balance laws. Applied Mathematics and Computation **217**, 8611–8629 (2011)
32. Yang, J., Perng, Y., Yen, R.: Implicit weighted essentially non-oscillatory schemes for the compressible Navier-Stokes equations. AIAA Journal **39**(11), 2082–2090 (2001)
33. Zennaro, M.: Natural continuous extensions of Runge-Kutta methods. Math. Comp. **46**, 119–133 (1986)