A re-averaged WENO reconstruction and a third order CWENO scheme for hyperbolic conservation laws

Chieh-Sen Huang[a,1], Todd Arbogast[b,2], Chen-Hui Hung[c,3]

[a] *Department of Applied Mathematics and National Center for Theoretical Sciences, National Sun Yat-sen University, Kaohsiung 804, Taiwan, R.O.C.*

[b] *University of Texas at Austin; Institute for Computational Engineering and Sciences; 201 EAST 24th St., Stop C0200; Austin, TX 78712–1229 and Mathematics Department, RLM 8.100; 2515 Speedway, Stop C1200; Austin, TX 78712–1202; U.S.A.*

[c] *Department of Mathematic and Physical Sciences, Air Force Academy, No. Sisou 1, Jieshou W. Rd., Gangshan Dist., Kaohsiung City 82047, Taiwan, R.O.C.*

Abstract

A WENO re-averaging (or re-mapping) technique is developed that converts function averages on one grid to another grid to high order. Nonlinear weighting gives the essentially non-oscillatory property to the re-averaged function values. The new *reconstruction* grid is used to obtain a standard high order WENO reconstruction of the function averages at a select point. By choosing the reconstruction grid to include the point of interest, a high order function value can be reconstructed using only positive linear weights. The re-averaging technique is applied to define two variants of a classic CWENO3 scheme that combines two linear polynomials to obtain formal third order accuracy. Such a scheme cannot otherwise be defined, due to the nonexistence of linear weights for third order reconstruction at the center of a grid element. The new scheme uses a compact stencil of three solution averages, and only positive linear weights are used. The scheme extends easily to problems in higher space dimensions, essentially as a tensor product of the one-dimensional scheme. The scheme maintains formal third order accuracy in higher dimensions. Numerical results show that this CWENO3 scheme is third order accurate for smooth problems and gives good results for non-smooth problems, including those with shocks.

Keywords: re-average, re-map, Central WENO, CWENO3, WENO reconstruction, compact stencil, positive linear weights, Euler equations
2000 MSC: 65M06, 65M08, 76M20, 76M12

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

[1] Supported in part under Taiwan National Science Council grant NSC 99-2115-M-110-006-MY3.

[2] Supported as part of the Center for Frontiers of Subsurface Energy Security, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award Number DE-SC0001114.

[3] Supported in part under Taiwan National Science Council grant NSC 101-2115-M-013-001.

1. Introduction

Since their introduction in the late 1980's, *essentially non-oscillatory* (ENO) and *weighted essentially non-oscillatory* (WENO) schemes [1, 2, 3, 4, 5, 6] have become two of the standard methods for approximating the solution to hyperbolic equations. They have been shown to be quite successful in a variety of applications. ENO and WENO schemes approximate the solution by a constant on each grid element, and advance in time explicitly by using a higher order polynomial reconstruction of the current solution. The ENO scheme computes several high order reconstructions, each on a different stencil, but uses only the stencil on which the solution is the smoothest (presumably the stencil does not cross a shock). The WENO schemes exploit the fact that often a convex linear combination of all the reconstructions can achieve even higher order accuracy than ENO at a specific point when the solution is smooth. A nonlinear weighting of the polynomial reconstructions, based on the solution smoothness, is actually used to avoid stencils that cross shocks in the solution.

Because reconstruction stencils need to avoid shocks, ENO and WENO are essentially upwind methods, and some form of flux-splitting is necessary when the system of equations is subject to flow in both directions. The central methods avoid flux splitting, and central WENO (CWENO) schemes were developed later in the 1990's [4, 6]. These methods use staggered grids, offset by one-half grid element, and so require higher order WENO reconstructions at the midpoint of a grid element.

Formally fifth and even ninth order CWENO (CWENO5 and CWENO9) schemes have appeared that use the classic WENO methodology to define high order reconstructions from low order ones [4, 6]. However, there is no classic CWENO3 scheme in the literature, due to the need for high order reconstruction at the center of the grid element. The linear weights required do not exist in the third (and seventh) order case. Indeed, quoting from Qiu and Shu [6, p. 194], "We remark that it is easy to verify that, when r is an even number ($r = 2, 4, \ldots$), there are no linear weights that satisfy [the appropriate] condition ... . Hence there are no third-, seventh-, eleventh-, etc., order central WENO reconstructions for the point values with these choices of stencils." A classic CWENO3 scheme would combine two linear approximations, each requiring the solution on two grid elements, for an overall stencil of three grid elements. However, the two linear polynomials agree at the center point, and so no convex combination can be higher than second order in a classic WENO reconstruction.

Third order CWENO schemes have been proposed in the literature by Levy, Puppo, and Russo [7, 8]. We will call these schemes "augmented," because they use a convex combination of the two linear approximations augmented directly by a higher order polynomial (a quadratic) in the case of one space dimension. Thus there is no weighting of low order polynomials to achieve higher order accuracy, as in classic WENO. Rather, high order accuracy comes directly from the quadratic reconstruction. One might view these methods as some kind of adaptive ENO schemes, because they choose the linear weights somewhat arbitrarily and use the WENO smoothness indicator to bias the stencil combination from the quadratic in smooth regions to one of the linears in non-smooth regions. To put it another way, classic WENO methodology uses lower order stencils, and weights them to *increase* accuracy in regions where the solution is smooth, as opposed to the augmented CWENO3 schemes, which include the higher-order stencil with the low order stencils

and weights them to *reduce* accuracy near discontinuities. Clearly both approaches have a great deal of merit.

In this paper, we present a classic CWENO3 scheme that indeed combines only two linear polynomials. It also uses a compact stencil of three grid element values in the reconstruction. The key to this work is to develop a technique that allows us to reconstruct average solution values on a grid different from the original. In the case of CWENO3, two natural *reconstruction grids* are defined, one given by shifting the computational grid by one-half grid element and resizing by a factor of one-quarter locally, and the other is nonuniform and described later. Re-averaging (or re-mapping) the solution from the original computational grid to the new reconstruction grid is then accomplished using high order integration, as described in a recent paper [9] (see also [10, 11]). Once this re-averaging to the new grid is complete, we need a third order reconstruction of the solution at the endpoints of a reconstruction grid element, rather than at the center of a computational grid element. The linear weights always exist for the endpoints of a grid element, and so we can proceed to define a classic CWENO scheme that is formally third order accurate. A formally seventh order, and in fact any odd order, scheme could be devised with the same technique, although we do not present the details here (see, e.g., [12, 13] for very high order schemes).

When linear weights exist in a classic WENO reconstruction, they may still be problematic if some of them are negative [11]. Fortunately, Shi, Hu, and Shu provided a technique to handle this case [14]. Our new re-averaging technique provides an alternate way to deal with negative weights, by avoiding them altogether. One can re-average to a shifted and perhaps locally rescaled reconstruction grid that includes the targeted point as an endpoint, since the linear weights for the endpoint case are known to be positive.

Standard WENO reconstructions cannot be used for arbitrary points, since we may not have the required linear weights. However, in some cases one may wish to obtain formally high order approximate values at arbitrary points, say in a postprocessing step. Our re-averaging technique can be used to overcome this limitation. This is done in [15], wherein an Eulerian-Lagrangian (or semi-Lagrangian) scheme is devised that requires high-order reconstructions at arbitrary points.

We also present in this paper an extension of the scheme to problems in more than one space dimension. It turns out to be relatively straightforward to do so, in the sense that much of the computation remains the same as in the one dimensional case; that is, the scheme is developed in an essentially tensor product form, iterating the one space dimensional scheme. We therefore only present the case of two space dimensions. Since it is built on one-dimensional computations, the scheme is relatively straightforward to implement, and it is computationally efficient, as opposed to the multi-dimensional approach of Levy, Puppo, and Russo [7, 8]. We also prove that the scheme is indeed formally third order accurate.

The re-averaging WENO reconstruction technique is presented in the next section. Section 3 presents the new CWENO3 scheme for a scalar conservation equation in one space dimension. The scheme is extended to problems posed in two space dimensions in Section 4. Numerical results in one space dimension are presented in Section 5, and the technique is applied to the Euler system in Section 6. Numerical results in two space dimensions are presented in Section 7, and the paper ends with conclusions in the final section.

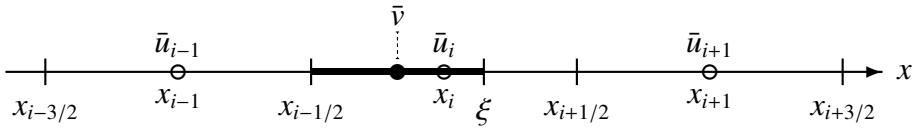


Figure 1: The re-averaging process. The three values $\bar{u}_{i-1}$, $\bar{u}_i$, and $\bar{u}_{i+1}$ are re-averaged to $\bar{v}$, which is a third order approximation to the average of u on the subinterval $[x_{i-1/2}, \xi]$.

2. The re-averaging technique

Let us fix the notation for the computational grid by choosing the spacing $h > 0$ and setting the grid points to be $x_{i-1/2} = (i-1)h$ and the central points to be $x_i = (i-1/2)h$. The ith grid element is then $[(i-1)h, ih] = [x_{i-1/2}, x_{i+1/2}]$, and its center is x_i. The function $u(x)$ is approximated on the grid by its element averages

$$\bar{u}_i \approx \frac{1}{h} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x)\, dx. \tag{1}$$

Our re-averaging WENO reconstruction technique is a two stage process. In the first stage, we define the reconstruction grid and re-average the computational grid element averages $\bar{u}_i$ to our new, reconstruction grid. We will take a reconstruction grid that is a shift and local rescaling of the computational grid. The re-averaging is done according to the high-order integration as described in [9]. The second stage of the process is to apply any standard WENO reconstruction for which the linear weights exist for the new reconstruction grid.

In this section we discuss the first stage of the process. The second stage is determined by the application, and is described in the next section when we define our new CWENO3 schemes. More details of the re-averaging can be found in [9, 10, 11, 15], but we present here only the two natural cases needed for a CWENO3 scheme. To fix ideas, suppose that we need to reconstruct, say, the function value at x_i, the center of grid element i, to third order accuracy using only the values $\bar{u}_{i-1}$, $\bar{u}_i$, and $\bar{u}_{i+1}$. Again, there is no standard third-order WENO reconstruction in this case, since the linear weights do not exist for the center point.

2.1. The re-averaging process

The key to the overall WENO reconstruction process is to produce a high order reconstruction of the function averages on a new grid. Such a high order reconstruction was described for fifth order reconstructions in a recent paper [9]. We present the details for third order reconstruction here.

As illustrated in Fig. 1, consider a point ξ such that $x_{i-1/2} \le \xi \le x_{i+1/2}$, which gives a subinterval $[x_{i-1/2}, \xi]$ of the central element $[x_{1-1/2}, x_{i+1/2}]$. Our goal is to reconstruct u using only $\bar{u}_{i-1}$, $\bar{u}_i$, and $\bar{u}_{i+1}$ so that we obtain a high order approximation to the integral

$$U_i(x) := \int_{x_{i-1/2}}^{x} u(y)\, dy.$$

The ENO approximations are linear functions [3, 11], one on a left stencil using $\bar{u}_{i-1}$ and $\bar{u}_i$, giving

$$P_L(x) := \bar{u}_{i-1} \frac{x_i - x}{h} + \bar{u}_i \frac{x - x_{i-1}}{h},$$

and one on a right stencil using $\bar{u}_i$ and $\bar{u}_{i+1}$, giving

$$P_R(x) := \bar{u}_i \frac{x_{i+1} - x}{h} + \bar{u}_{i+1} \frac{x - x_i}{h}.$$

We need to integrate these to determine the linear weights. However, it is simpler to determine the integrals by approximating directly the primitive function $U_i(x)$. The integrals of P_L and P_R are simply the two quadratic polynomials interpolating U_i using the proper masses. The left polynomial approximates $-h\bar{u}_{i-1}$ at $x_{i-3/2}$, 0 at $x_{i-1/2}$, and $h\bar{u}_i$ at $x_{i+1/2}$, and it is given by

$$Q_L(x) := -\bar{u}_{i-1} \frac{(x - x_{i-1/2})(x - x_{i+1/2})}{2h} + \bar{u}_i \frac{(x - x_{i-3/2})(x - x_{i-1/2})}{2h} = \int_{x_{i-1/2}}^{x} P_L(y)\, dy.$$

Similarly, the right polynomial approximates 0 at $x_{i-1/2}$, $h\bar{u}_i$ at $x_{i+1/2}$, and $h(\bar{u}_i + \bar{u}_{i+1})$ at $x_{i+3/2}$. It is

$$Q_R(x) := -\bar{u}_i \frac{(x - x_{i-1/2})(x - x_{i+3/2})}{h} + (\bar{u}_i + \bar{u}_{i+1}) \frac{(x - x_{i-1/2})(x - x_{i+1/2})}{2h} = \int_{x_{i-1/2}}^{x} P_R(y)\, dy.$$

These two quadratics are combined using the linear weights γ_L and $\gamma_R = 1 - \gamma_L$ into the polynomial

$$Q(x) := \gamma_L Q_L(x) + \gamma_R Q_R(x).$$

To set γ_L, we compare $Q(x)$ to the higher order cubic polynomial $C(x)$ that interpolates all the data, i.e., it is $-h\bar{u}_{i-1}$ at $x_{i-3/2}$, 0 at $x_{i-1/2}$, $h\bar{u}_i$ at $x_{i+1/2}$, and $h(\bar{u}_i + \bar{u}_{i+1})$ at $x_{i+3/2}$. Thus

$$\begin{aligned} C(x) := \bar{u}_{i-1} \frac{(x - x_{i-1/2})(x - x_{i+1/2})(x - x_{i+3/2})}{6h^2} - \bar{u}_i \frac{(x - x_{i-3/2})(x - x_{i-1/2})(x - x_{i+3/2})}{2h^2} \\ + (\bar{u}_i + \bar{u}_{i+1}) \frac{(x - x_{i-3/2})(x - x_{i-1/2})(x - x_{i+1/2})}{6h^2}. \end{aligned}$$

If we wanted to approximate the primitive function to high order at $x = \xi$, we would choose γ_L so that $Q(\xi) = C(\xi)$, no matter the values of $\bar{u}_{i-1}$, $\bar{u}_i$, and $\bar{u}_{i+1}$. This overdetermined system has a solution, and the resulting linear weights are positive, and given by

$$\gamma_L(\xi) = \frac{2 - \rho(\xi)}{3} > 0, \quad \gamma_R(\xi) = 1 - \gamma_L(\xi) = \frac{1 + \rho(\xi)}{3} > 0, \quad \rho(\xi) := \frac{\xi - x_{i-1/2}}{h} \in [0, 1]. \tag{2}$$

These linear weights ensure that the formal order of accuracy of $Q(\xi)$ to $U_i(\xi)$ is $O(h^4)$. That is, using the ratio of distance across the ith grid element $\rho = \rho(\xi)$, we have

$$U_i(\xi) \approx \gamma_L(\xi) \left[\frac{\rho(1 - \rho)}{2} \bar{u}_{i-1} + \frac{\rho(1 + \rho)}{2} \bar{u}_i \right] h + \gamma_R(\xi) \left[\frac{\rho(3 - \rho)}{2} \bar{u}_i - \frac{\rho(1 - \rho)}{2} \bar{u}_{i+1} \right] h. \tag{3}$$

In WENO applications, the linear weights $\gamma_L(\xi) = (2 - \rho)/3$ and $\gamma_R(\xi) = (1 + \rho)/3$ would be modified by the smoothness indicator, as described below.

However, we want to approximate an average of u between, say, $\xi_1 = x_{i-1/2} + \rho_1 h$ and $\xi_2 = x_{i-1/2} + \rho_2 h$, which is

$$\bar{v}(\rho_1, \rho_2) \approx \frac{1}{\xi_2 - \xi_1} \int_{\xi_1}^{\xi_2} u(x)\, dx = \frac{1}{\xi_2 - \xi_1} (U_i(\xi_2) - U_i(\xi_1)). \tag{4}$$

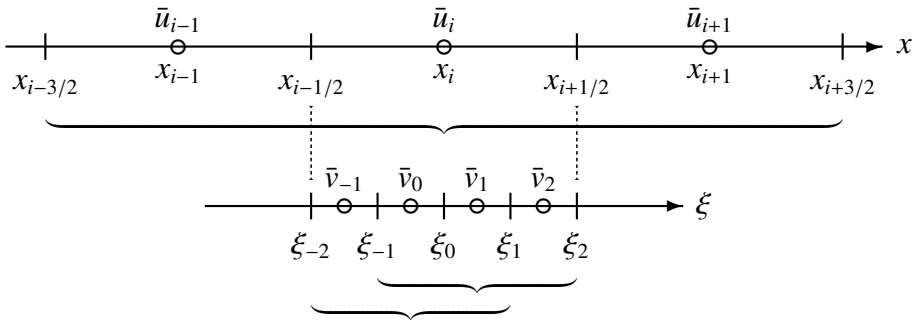


Figure 2: The uniform reconstruction grid. The computational grid is labeled by x, while the reconstruction grid is typeset below it and labeled in ξ. The three values $\bar{u}_{i-1}$, $\bar{u}_i$, and $\bar{u}_{i+1}$ are re-averaged to $\bar{v}_{-1}$, $\bar{v}_0$, $\bar{v}_1$, and $\bar{v}_2$, which in turn form two sets of three values for third order WENO reconstruction at $x_i = \xi_0$.

This involves differences of the primitive function, and the linear weights do not always exist in this case [9]. However, for our CWENO3 scheme, we need averages of u only over certain subintervals of the ith grid element for which the linear weights exist, and, in fact, they are positive. We present the two natural cases of reconstruction grids in the next two subsections.

2.2. Case 1, a uniform reconstruction grid

We take in the first case of a new CWENO3 scheme the reconstruction grid shifted by one-half grid element and resized by a factor of one-quarter locally. That is, as depicted in Fig. 2, the new grid near the target point x_i covers the interval $[x_{i-1/2}, x_{i+1/2}]$, the ith element, and is given by $\xi_{-2} = x_{i-1/2}$, $\xi_{-1} = x_{i-1/2} + h/4$, $\xi_0 = x_i = x_{i-1/2} + h/2$, $\xi_1 = x_{i+1/2} + 3h/4$, and $\xi_2 = x_{i+1/2}$.

We need to re-average the solution on the computational grid to the reconstruction grid $\{\xi_{-2}, \xi_{-1}, \xi_0, \xi_1, \xi_2\}$. For example, to compute a higher order approximation to the function average over the second quarter of the ith grid element, i.e., to get $\bar{v}_0 = \bar{v}(1/4, 1/2)$, we would set γ_L and γ_R so that

$$\begin{aligned}\frac{h}{4}\bar{v}_0 &\approx \gamma_L[Q_L(x_{i-1/2} + h/2) - Q_L(x_{i-1/2} + h/4)] + \gamma_R[Q_R(x_{i-1/2} + h/2) - Q_R(x_{i-1/2} + h/4)] \\ &= C(x_{i-1/2} + h/2) - C(x_{i-1/2} + h/4).\end{aligned}$$

The result is that for the quarter subintervals,

$$\bar{v}(0, 1/4) \approx \bar{v}_{-1} := \gamma_L\left[\frac{3}{8}\bar{u}_{i-1} + \frac{5}{8}\bar{u}_i\right] + \gamma_R\left[\frac{11}{8}\bar{u}_i - \frac{3}{8}\bar{u}_{i+1}\right] \quad \text{with } \gamma_L = \frac{7}{12}, \gamma_R = \frac{5}{12}, \tag{5}$$

$$\bar{v}(1/4, 1/2) \approx \bar{v}_0 := \gamma_L\left[\frac{1}{8}\bar{u}_{i-1} + \frac{7}{8}\bar{u}_i\right] + \gamma_R\left[\frac{9}{8}\bar{u}_i - \frac{1}{8}\bar{u}_{i+1}\right] \quad \text{with } \gamma_L = \frac{1}{4}, \ \gamma_R = \frac{3}{4}, \tag{6}$$

$$\bar{v}(1/2, 3/4) \approx \bar{v}_1 := \gamma_L\left[-\frac{1}{8}\bar{u}_{i-1} + \frac{9}{8}\bar{u}_i\right] + \gamma_R\left[\frac{7}{8}\bar{u}_i + \frac{1}{8}\bar{u}_{i+1}\right] \quad \text{with } \gamma_L = \frac{3}{4}, \ \gamma_R = \frac{1}{4}, \tag{7}$$

$$\bar{v}(3/4, 1) \approx \bar{v}_2 := \gamma_L\left[-\frac{3}{8}\bar{u}_{i-1} + \frac{11}{8}\bar{u}_i\right] + \gamma_R\left[\frac{5}{8}\bar{u}_i + \frac{3}{8}\bar{u}_{i+1}\right] \quad \text{with } \gamma_L = \frac{5}{12}, \gamma_R = \frac{7}{12}. \tag{8}$$

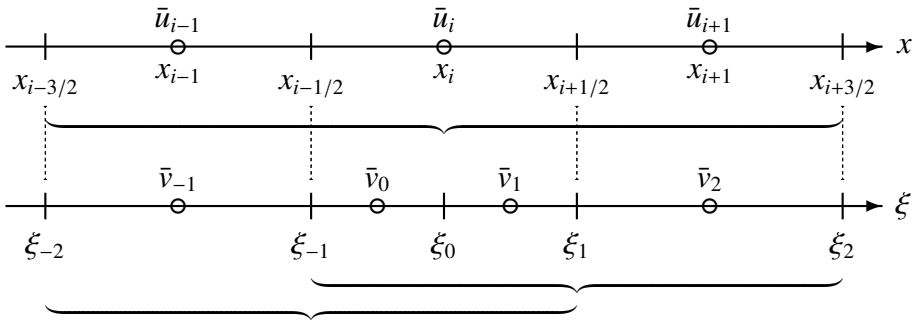


Figure 3: The nonuniform reconstruction grid. The computational grid is labeled by x, while the reconstruction grid is typeset below it and labeled in ξ. The three values $\bar{u}_{i-1}$, $\bar{u}_i$, and $\bar{u}_{i+1}$ are re-averaged to $\bar{v}_{-1} = \bar{u}_{i-1}$, $\bar{v}_0$, $\bar{v}_1$, and $\bar{v}_2 = \bar{u}_{i+1}$, which in turn form two sets of three values for third order WENO reconstruction at $x_i = \xi_0$.

2.3. Case 2, a nonuniform reconstruction grid

In this second case, we select a reconstruction grid that is nonuniform. As depicted in Fig. 3, we cover the interval $[x_{i-3/2}, x_{i+3/2}]$ with the new grid $\xi_{-2} = x_{i-3/2}$, $\xi_{-1} = x_{i-1/2}$, $\xi_0 = x_i$, $\xi_1 = x_{i+1/2}$ and $\xi_2 = x_{i+3/2}$; that is, we simply divide the center element in half. In this case, formula (3) applies directly, and leads to

$$\bar{v}_{-1} := \bar{u}_{i-1}, \tag{9}$$

$$\bar{v}(0,1/2) \approx \bar{v}_0 := \gamma_L\Big[\frac{1}{4}\bar{u}_{i-1} + \frac{3}{4}\bar{u}_i\Big] + \gamma_R\Big[\frac{5}{4}\bar{u}_i - \frac{1}{4}\bar{u}_{i+1}\Big] \quad \text{with } \gamma_L = \frac{1}{2},\ \gamma_R = \frac{1}{2}, \tag{10}$$

$$\bar{v}(1/2,1) \approx \bar{v}_1 := \gamma_L\Big[-\frac{1}{4}\bar{u}_{i-1} + \frac{5}{4}\bar{u}_i\Big] + \gamma_R\Big[\frac{3}{4}\bar{u}_i + \frac{1}{4}\bar{u}_{i+1}\Big] \quad \text{with } \gamma_L = \frac{1}{2},\ \gamma_R = \frac{1}{2}, \tag{11}$$

$$\bar{v}_2 := \bar{u}_{i+1}. \tag{12}$$

2.4. Modification of the linear weights for non-smoothness

Of course to avoid discontinuities within the WENO framework, we modify these linear weights with the usual smoothness indicator

$$IS_k^i := \sum_{\ell=1}^{2} \int_{x_{i-1/2}}^{x_{i+1/2}} h^{2\ell-1}\Big[\frac{\partial^\ell P_k(x)}{\partial x^\ell}\Big]^2 dx, \quad k = L, R, \tag{13}$$

using the original linear polynomials. For a uniform grid, an explicit integration yields

$$IS_L^i(\bar{u}) = (\bar{u}_i - \bar{u}_{i-1})^2 \quad \text{and} \quad IS_R^i(\bar{u}) = (\bar{u}_{i+1} - \bar{u}_i)^2. \tag{14}$$

The nonlinear weights are then given by

$$\tilde{\gamma}_L^i(\gamma_L, \bar{u}) := \frac{\dfrac{\gamma_L}{(\epsilon + IS_L^i(\bar{u}))^2}}{\dfrac{\gamma_L}{(\epsilon + IS_L^i(\bar{u}))^2} + \dfrac{1-\gamma_L}{(\epsilon + IS_R^i(\bar{u}))^2}} \quad \text{and} \quad \tilde{\gamma}_R^i(\gamma_R, \bar{u}) := 1 - \tilde{\gamma}_L^i(1-\gamma_R, \bar{u}). \tag{15}$$

where $\epsilon > 0$ is some small parameter. That is, in (5)–(12), the linear weights in the formulas are replaced by the nonlinear weights $\tilde{\gamma}_L^i(\gamma_L, \bar{u})$ and $\tilde{\gamma}_R^i(\gamma_R, \bar{u})$, using the original linear weights γ_L and γ_R as given on the right-hand side.

In (5)–(11), we have the re-averaged values of $\bar{v}$, defined in (4), for the subintervals that we will need later, accurate to $O(h^3)$ when the function $u(x)$ is smooth, since $\xi_2 - \xi_1$ is $O(h)$. Note that the re-averaging uses only the linear polynomials $P_L(x)$ and $P_R(x)$ (in integrated form $Q_L(x)$ and $Q_R(x)$), and that the stencil is compact, using only the grid element averages $\bar{u}_{i-1}$, $\bar{u}_i$, and $\bar{u}_{i+1}$. It is also essentially non-oscillatory near shocks, since it drops back to only using one linear stencil by the nonlinear smoothness indicator correction to the linear weights.

Remark. For the quarter subintervals, the re-averaged function values (5)–(8) do not preserve mass when the nonlinear weights are used, since the linear weights differ between subintervals. A mass conservative re-averaging is possible [15] if one computes the integrals always from the left endpoint $x_{i-1/2}$. That is, as

$$\begin{aligned}
\bar{v}(0,1/4) &\approx \frac{4}{h} Q(x_{i-1/2} + h/4) \\
&= \bar{v}_{-1} := \tilde{\gamma}_L^i(7/12, \bar{u})\left[\frac{3}{8}\bar{u}_{i-1} + \frac{5}{8}\bar{u}_i\right] + \tilde{\gamma}_R^i(5/12, \bar{u})\left[\frac{11}{8}\bar{u}_i - \frac{3}{8}\bar{u}_{i+1}\right], && (16) \\
\bar{v}(1/4,1/2) &\approx \frac{4}{h} Q(x_{i-1/2} + h/2) - \bar{v}_{-1} \\
&= \bar{v}_0 := \tilde{\gamma}_L^i(1/2, \bar{u})\left[\frac{1}{2}\bar{u}_{i-1} + \frac{3}{2}\bar{u}_i\right] + \tilde{\gamma}_R^i(1/2, \bar{u})\left[\frac{5}{2}\bar{u}_i - \frac{1}{2}\bar{u}_{i+1}\right] - \bar{v}_{-1}, && (17) \\
\bar{v}(1/2,3/4) &\approx \frac{4}{h} Q(x_{i-1/2} + 3h/4) - \bar{v}_{-1} - \bar{v}_0 \\
&= \bar{v}_1 := \tilde{\gamma}_L^i(5/12, \bar{u})\left[\frac{3}{8}\bar{u}_{i-1} + \frac{21}{8}\bar{u}_i\right] + \tilde{\gamma}_R^i(7/12, \bar{u})\left[\frac{27}{8}\bar{u}_i - \frac{3}{8}\bar{u}_{i+1}\right] - \bar{v}_{-1} - \bar{v}_0, && (18) \\
\bar{v}(3/4,1) &= \bar{v}_2 := 4\bar{u}_i - \bar{v}_{-1} - \bar{v}_0 - \bar{v}_1, && (19)
\end{aligned}$$

wherein the linear weights have been converted to the nonlinear weights $\tilde{\gamma}_L^i$, $\tilde{\gamma}_R^i$ using (14)–(15). Since the nonlinear weight corrections are done consistently for each subinterval, mass is conserved. However, mass conservation at this stage is not required for our CWENO3 scheme, since the re-averaged function values are used merely to obtain high-order approximations to the function at a point. Of course, the overall scheme itself does conserve mass. In computations, we did not see any noticeable difference between using (5)–(8) and (16)–(19). Of course, the half subinterval formulas use the same weights, and so conserve mass at this and all stages of the scheme.

Remark. Both of our two re-averaging procedures exhibit a superconvergence effect for the higher order reconstruction of a point value. To be precise, if one re-averages $\{\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}\}$ to the uniform or nonuniform reconstruction grid to obtain $\{\bar{v}_{-1}, \bar{v}_0, \bar{v}_1, \bar{v}_2\}$ and then reconstructs $u(x_i)$ in the standard way either from the left stencil $\{\bar{v}_{-1}, \bar{v}_0, \bar{v}_1\}$ or the right stencil $\{\bar{v}_0, \bar{v}_1, \bar{v}_2\}$ using the linear weights directly without nonlinear modification in both stages of the reconstruction, then one obtains the accuracy $O(h^4)$. To prove this fact, one needs to show that the procedure is exact when the underlying function $u(x)$ is a cubic polynomial. In fact, it is enough to show that when $u(x) = x^3$,

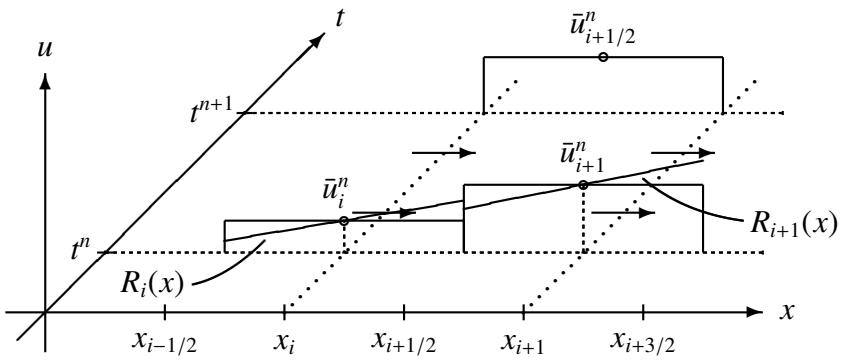


Figure 4: An illustration of a time step in a CWENO scheme. The shifted grid element average $\bar{u}^{n+1}_{i+1/2}$ at the advanced time t^{n+1} is computed from the reconstructed solution $R_i(x)$ and $R_{i+1}(x)$ at time t^n, which gives the initial mass of the solution in place over the region $[x_i, x_{i+1}]$, and the flux values over time at the grid locations x_i and x_{i+1}.

the reconstruction value is zero. Conceptually, when the linear weights are used, the reconstruction value is the same as what arises from the quadratic approximation of $u(x)$ evaluated at $x = 0$ in both stages of the reconstruction. For the re-averaging stage, since the cubic is an odd function and the reconstruction grid is symmetric, the quadratic approximation of x^3 must also be odd, i.e., a multiple of x. The second stage will reconstruct the value zero for x, and the proof is complete. Of course, one can simply verify the result for $u(x) = x^3$ by applying the formulas (5)–(8) and (26)–(27) for the uniform reconstruction grid and (9)–(12) and (28)–(29) for the nonuniform reconstruction grid. This superconvergence result will break down when nonlinear weights are used. Nevertheless, one can check the accuracy of a computer code implementation of the algorithm by checking for this superconvergence.

3. CWENO3 for a scalar conservation law in one space dimension

Given $f(u; x, t)$, consider the initial value problem for a hyperbolic conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \qquad x \in \mathbb{R},\ t > 0, \tag{20}$$

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}. \tag{21}$$

A description of the details of a Central WENO scheme can be found in [4, 6]. We present herein a summary of the entire scheme for completeness, although the only new feature is the way we handle the WENO reconstruction at the center of the grid element by using our re-averaging procedure.

Let $t^0 = 0 < t^1 < t^2 < \cdots$ define the time levels. In a CWENO scheme, the grid shifts one-half grid element each time step. We describe a step from time t^n to t^{n+1} using the grid indexing we have been using, i.e., x_i is the center of an element of the grid at time t^n. CWENO is based on integrating (20) over a shifted grid element $[x_i, x_{i+1}]$ in the space variable x and $[t^n, t^{n+1}]$ in time t. That is, as illustrated in Fig. 4, the solution average on the shifted element $[x_i, x_{i+1}]$ at time t^{n+1} is

$\bar{u}^{n+1}_{i+1/2}$, and it is set from the approximation

$$\bar{u}^{n+1}_{i+1/2} \approx \frac{1}{h}\Big\{ \int_{x_i}^{x_{i+1}} u(x,t^n)\,dx - \int_{t^n}^{t^{n+1}} f(u)\Big|_{x_{i+1}}\,dt + \int_{t^n}^{t^{n+1}} f(u)\Big|_{x_i}\,dt\Big\}. \tag{22}$$

The first integral in (22) uses the WENO reconstruction function $R_u(x,t^n)$ of $\{\bar{u}_{i-1},\bar{u}_i,\bar{u}_{i+1},\bar{u}_{i+2}\}$. To be more precise, $R_u(x,t^n)$ is defined by $R_i(x)$ on $[x_{i-1/2},x_{i+1/2}]$, which involves $\{\bar{u}_{i-1},\bar{u}_i,\bar{u}_{i+1}\}$, and $R_{i+1}(x)$ on $[x_{i+1/2},x_{i+3/2}]$, which involves $\{\bar{u}_i,\bar{u}_{i+1},\bar{u}_{i+2}\}$. The required integrals were actually given above in (11) and (10), and they are

$$\begin{aligned}\frac{1}{h}\int_{x_i}^{x_{i+1}} R_u(x,t^n)\,dx &= \frac{1}{h}\int_{x_i}^{x_{i+1/2}} R_i(x)\,dx + \frac{1}{h}\int_{x_{i+1/2}}^{x_{i+1}} R_{i+1}(x)\,dx\\ &= \tilde{\gamma}^i_L(1/2,\bar{u}^n)\Big[-\frac{1}{8}\bar{u}^n_{i-1}+\frac{5}{8}\bar{u}^n_i\Big] + \tilde{\gamma}^i_R(1/2,\bar{u}^n)\Big[\frac{3}{8}\bar{u}^n_i+\frac{1}{8}\bar{u}^n_{i+1}\Big]\\ &\quad + \tilde{\gamma}^{i+1}_L(1/2,\bar{u}^n)\Big[\frac{1}{8}\bar{u}^n_i+\frac{3}{8}\bar{u}^n_{i+1}\Big] + \tilde{\gamma}^{i+1}_R(1/2,\bar{u}^n)\Big[\frac{5}{8}\bar{u}^n_{i+1}-\frac{1}{8}\bar{u}^n_{i+2}\Big],\end{aligned} \tag{23}$$

wherein the linear weights happen to all be $1/2$, but the two halves of the integral use a different position for the smoothness indicator to define the nonlinear weights.

For the two flux integrations, i.e., the second and third terms in (22), we use a Gaussian quadrature rule for the integration of the flux in time over $[t^n,t^{n+1}]$, and the two-point, third order rule is appropriate here. Let $\Delta t^n = t^{n+1}-t^n$ and

$$\theta^{\pm}_G := \frac{1}{2} \pm \frac{1}{2\sqrt{3}}. \tag{24}$$

For $j = i, i+1$, the quadrature rule is

$$\int_{t^n}^{t^{n+1}} f(u)\Big|_{x_j}\,dt \approx \frac{\Delta t^n}{2}\Big[f(u(x_j,t^n+\Delta t^n\theta^-_G)) + f(u(x_j,t^n+\Delta t^n\theta^+_G))\Big]. \tag{25}$$

Therefore, the main task remaining is to find properly reconstructed values for $u(x_j,t^n+\Delta t^n\theta^{\pm}_G)$, i.e., at the centers of the computational grid elements in space and at the Gauss points in time.

3.1. Flux evaluation in space using re-averaging

Evaluation of (25) in space uses our new re-averaging technique. To obtain high order accuracy for $u^n_i \approx u(x_i,t^n)$, we can use either the uniform or nonuniform grid from Sections 2.2 and 2.3. We re-average the solution $\{\bar{u}^n_{i-1},\bar{u}^n_i,\bar{u}^n_{i+1}\}$ to the chosen reconstruction grid to obtain $\{\bar{v}_{-1},\bar{v}_0,\bar{v}_1,\bar{v}_2\}$. As illustrated in Figures 2 and 3, there are actually two possible stencils of three re-averaged quantities for standard WENO3 reconstruction to the point x_i. That is, we can define two linear WENO reconstruction polynomials, one for each of the stencils $\{\bar{v}_{-1},\bar{v}_0,\bar{v}_1\}$ and $\{\bar{v}_0,\bar{v}_1,\bar{v}_2\}$.

For the uniform reconstruction grid, we have the left and right stencil reconstructions

$$u^n_{i,L} := \tilde{\gamma}^0_L(1/3,\bar{v})\Big[-\frac{1}{2}\bar{v}_{-1}+\frac{3}{2}\bar{v}_0\Big] + \tilde{\gamma}^0_R(2/3,\bar{v})\Big[\frac{1}{2}\bar{v}_0+\frac{1}{2}\bar{v}_1\Big], \tag{26}$$

$$u^n_{i,R} := \tilde{\gamma}^1_L(2/3,\bar{v})\Big[\frac{1}{2}\bar{v}_0+\frac{1}{2}\bar{v}_1\Big] \quad + \tilde{\gamma}^1_R(1/3,\bar{v})\Big[\frac{3}{2}\bar{v}_1-\frac{1}{2}\bar{v}_2\Big], \tag{27}$$

using the smoothness indicator defined at the appropriate index for differences of values of $\bar{v}$. For the nonuniform reconstruction grid, it is easy to derive the linear weights, and we have

$$u^n_{i,L} := \tilde{\gamma}^0_L(1/4, \bar{v})\Big[-\frac{1}{3}\bar{v}_{-1} + \frac{4}{3}\bar{v}_0\Big] + \tilde{\gamma}^0_R(3/4, \bar{v})\Big[\frac{1}{2}\bar{v}_0 + \frac{1}{2}\bar{v}_1\Big], \tag{28}$$

$$u^n_{i,R} := \tilde{\gamma}^1_L(3/4, \bar{v})\Big[\frac{1}{2}\bar{v}_0 + \frac{1}{2}\bar{v}_1\Big] \qquad + \tilde{\gamma}^1_R(1/4, \bar{v})\Big[\frac{4}{3}\bar{v}_1 - \frac{1}{3}\bar{v}_2\Big]. \tag{29}$$

In this latter case, it should be noted that the smoothness indicator must be computed from (13) using our nonuniform grid. The left stencil has

$$IS^i_L(\bar{v}) = \left(\frac{2}{3}(v_0 - v_{-1})\right)^2 \quad \text{and} \quad IS^i_R(\bar{v}) = (v_1 - v_0)^2, \tag{30}$$

and for the right stencil,

$$IS^i_L(\bar{v}) = (v_1 - v_0)^2 \quad \text{and} \quad IS^i_R(\bar{v}) = \left(\frac{2}{3}(v_2 - v_1)\right)^2. \tag{31}$$

To produce a central scheme, we average the results of these two stencils and define the formally third order reconstructions

$$u(x_i, t^n) \approx u^n_i := \frac{u^n_{i,L} + u^n_{i,R}}{2} \quad \text{and} \quad f(u(x_i, t^n); x_i, t^n) \approx f^n_i := f(u^n_i; x_i, t^n). \tag{32}$$

(As an alternative, one could introduce flux-splitting and take the upstream weighted stencil, but we present only the central scheme herein.)

3.2. Flux evaluation in time using Runge-Kutta

Evaluation of (25) in time is handled in the usual way, in our case, by using a two-stage Runge-Kutta method combined with the natural continuous extension of Zennaro [16] applied for fixed $x = x_i$ to the initial value problem

$$\frac{\partial u}{\partial t} = -\frac{\partial f(u)}{\partial x} \quad \text{and} \quad u(x_i, t^n) = u^n_i.$$

One needs to approximate the space derivative of $f(u; x, t)$ using the reconstructed point values f^n_i.

The first stage of the Runge-Kutta method is to define

$$g^{(1)} := -\tilde{\gamma}^i_L(1/2, f^n)\frac{f^n_i - f^n_{i-1}}{h} - \tilde{\gamma}^i_R(1/2, f^n)\frac{f^n_{i+1} - f^n_i}{h} \approx -\frac{\partial f(u)}{\partial x}\bigg|_{x_i, t^n} \tag{33}$$

(so that the overall scheme has a compact stencil of five points) and then set the intermediate values

$$w_i := u^n_i + \Delta t^n\, g^{(1)}. \tag{34}$$

The second stage is then to define $f^{n+1^-}_i = f(w_i; x_i, t^{n+1})$ and

$$g^{(2)} := -\tilde{\gamma}^i_L(1/2, f^{n+1^-})\frac{f^{n+1^-}_i - f^{n+1^-}_{i-1}}{h} - \tilde{\gamma}^i_R(1/2, f^{n+1^-})\frac{f^{n+1^-}_{i+1} - f^{n+1^-}_i}{h} \approx -\frac{\partial f(u)}{\partial x}\bigg|_{x_i, t^{n+1}}. \tag{35}$$

The Runge-Kutta method would be completed by defining $u_i^{n+1^-} := u_i^n + \frac{\Delta t^n}{2}(g^{(1)} + g^{(2)})$. However, we do not need u at $t = t^{n+1^-}$, but rather at the Gauss points in time. The natural continuous extension of the Runge-Kutta values over the time interval gives us

$$u(x_i, t^n + \Delta t^n \theta_G^\pm) \approx u_i^{n+\theta_G^\pm} := u_i^n + \frac{\Delta t^n}{2}[(2\theta_G^\pm - (\theta_G^\pm)^2)\, g^{(1)} + (\theta_G^\pm)^2\, g^{(2)}]. \tag{36}$$

3.3. Summary of the scheme

Combining the steps above, (22) with (23) and (25) simplifies to

$$\begin{aligned}
\bar{u}_{i+1/2}^{n+1} := {} & \tilde{\gamma}_L^i(1/2, \bar{u}^n)\Big[-\frac{1}{8}\bar{u}_{i-1}^n + \frac{5}{8}\bar{u}_i^n\Big] + \tilde{\gamma}_R^i(1/2, \bar{u}^n)\Big[\frac{3}{8}\bar{u}_i^n + \frac{1}{8}\bar{u}_{i+1}^n\Big] \\
& + \tilde{\gamma}_L^{i+1}(1/2, \bar{u}^n)\Big[\frac{1}{8}\bar{u}_i^n + \frac{3}{8}\bar{u}_{i+1}^n\Big] + \tilde{\gamma}_R^{i+1}(1/2, \bar{u}^n)\Big[\frac{5}{8}\bar{u}_{i+1}^n - \frac{1}{8}\bar{u}_{i+2}^n\Big] \\
& - \frac{\Delta t^n}{2h}\Big[f(u_{i+1}^{n+\theta_G^-}; x_{i+1}, t^n + \Delta t^n\theta_G^-) + f(u_{i+1}^{n+\theta_G^+}; x_{i+1}, t^n + \Delta t^n\theta_G^+)\Big] \\
& + \frac{\Delta t^n}{2h}\Big[f(u_i^{n+\theta_G^-}; x_i, t^n + \Delta t^n\theta_G^-) + f(u_i^{n+\theta_G^+}; x_i, t^n + \Delta t^n\theta_G^+)\Big],
\end{aligned} \tag{37}$$

where $u_j^{n+\theta_G^\pm}$ are defined in (33)–(36), which requires u_j^n and f_j^n defined in (26)–(32) and $\bar{v}$ defined either in (5)–(8) for a uniform reconstruction grid and in (9)–(12) for a nonuniform reconstruction grid. One also needs $\theta_G^\pm$ defined in (24) and $\tilde{\gamma}_L^j$ and $\tilde{\gamma}_R^j$ defined in (14)–(15).

Remark. The nonuniform reconstruction grid is somewhat more computationally efficient. However, the numerical results (see Example 5.2) perhaps indicate that the uniform reconstruction grid is the better choice.

4. CWENO3 for a scalar conservation law in higher space dimensions

We consider now extension to problems with more than one space dimension. Extension to three and even higher dimensions is straightforward from the case of two space dimensions, so we present only this case, which is

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = 0, \qquad (x, y) \in \mathbb{R}^2,\ t > 0, \tag{38}$$

$$u(x, y, 0) = u_0(x, y), \quad (x, y) \in \mathbb{R}^2, \tag{39}$$

for the given $f(u; x, y, t)$ and $g(u; x, y, t)$.

The computational grid is a tensor product of uniform grids. The spacing in x is denoted $h_x > 0$ and the spacing in y is $h_y > 0$, and we tacitly assume that the ratio h_x/h_y is bounded above and below as these parameters are refined. Again, for a step from t^n to t^{n+1}, the center points of the grid elements are $(x_i, y_j) = (ih_x, jh_y)$ and the current approximation to the element averages are denoted

$$\bar{u}_{i,j}^n \approx \frac{1}{h_x h_y}\int_{y_{j-1/2}}^{y_{j+1/2}}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x, y, t^n)\, dx\, dy. \tag{40}$$

Integration of the differential equation leads us to the basis of our approximation scheme, to wit

$$\begin{aligned}\bar{u}^{n+1}_{i+1/2,j+1/2} \approx \frac{1}{h_x h_y}\Big\{ &\int_{y_j}^{y_{j+1}}\int_{x_i}^{x_{i+1}} u(x,y,t^n)\,dx\,dy \\ &- \int_{t^n}^{t^{n+1}}\int_{y_j}^{y_{j+1}} f(u)\Big|_{x=x_{i+1}}\,dy\,dt + \int_{t^n}^{t^{n+1}}\int_{y_j}^{y_{j+1}} f(u)\Big|_{x=x_i}\,dy\,dt \\ &- \int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} g(u)\Big|_{y=y_{j+1}}\,dx\,dt + \int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} g(u)\Big|_{y=y_j}\,dx\,dt\Big\}. \end{aligned} \tag{41}$$

4.1. High order integration in two-dimensions

The first double integral in (41) is approximated to high order using a two-dimensional analogue of (3). To derive this analogue, let $x_{i-1/2} \le \xi \le x_{i+1/2}$ and $y_{j-1/2} \le \eta \le y_{j+1/2}$. Apply (3) to a given function $u(x,y)$ in the variable x keeping y fixed, and then apply it again in y. In terms of the scaled values

$$\rho(\xi) := \frac{\xi - x_{i-1/2}}{h_x} \in [0,1] \quad \text{and} \quad \sigma(\eta) := \frac{\eta - y_{j-1/2}}{h_y} \in [0,1], \tag{42}$$

the result is

$$\begin{aligned} U_{i,j}(\xi,\eta) &:= \int_{y_{j-1/2}}^{\eta}\int_{x_{i-1/2}}^{\xi} u(x,y)\,dx\,dy \\ &\approx \Big\{ \gamma_{L,L}(\xi,\eta)\Big[\frac{\rho(1-\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i-1,j-1} + \frac{\rho(1-\rho)}{2}\frac{\sigma(1+\sigma)}{2}\bar{u}_{i-1,j} \\ &\qquad + \frac{\rho(1+\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i,j-1} + \frac{\rho(1+\rho)}{2}\frac{\sigma(1+\sigma)}{2}\bar{u}_{i,j}\Big] \\ &\quad + \gamma_{L,R}(\xi,\eta)\Big[\frac{\rho(1-\rho)}{2}\frac{\sigma(3-\sigma)}{2}\bar{u}_{i-1,j} - \frac{\rho(1-\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i-1,j+1} \\ &\qquad + \frac{\rho(1+\rho)}{2}\frac{\sigma(3-\sigma)}{2}\bar{u}_{i,j} - \frac{\rho(1+\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i,j+1}\Big] \\ &\quad + \gamma_{R,L}(\xi,\eta)\Big[\frac{\rho(3-\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i,j-1} + \frac{\rho(3-\rho)}{2}\frac{\sigma(1+\sigma)}{2}\bar{u}_{i,j} \\ &\qquad - \frac{\rho(1-\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i+1,j-1} - \frac{\rho(1-\rho)}{2}\frac{\sigma(1+\sigma)}{2}\bar{u}_{i+1,j}\Big] \\ &\quad + \gamma_{R,R}(\xi,\eta)\Big[\frac{\rho(3-\rho)}{2}\frac{\sigma(3-\sigma)}{2}\bar{u}_{i,j} - \frac{\rho(3-\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i,j+1} \\ &\qquad - \frac{\rho(1-\rho)}{2}\frac{\sigma(3-\sigma)}{2}\bar{u}_{i+1,j} + \frac{\rho(1-\rho)}{2}\frac{\sigma(1-\sigma)}{2}\bar{u}_{i+1,j+1}\Big]\Big\} h_x h_y, \end{aligned} \tag{43}$$

wherein the linear weights are given by (2) and

$$\gamma_{k,\ell}(\xi,\eta) := \gamma_k(\xi)\,\gamma_\ell(\eta), \quad k,\ell = L,R. \tag{44}$$

We claim that the accuracy of the approximation in (43) is $O(h^5)$, where $h = \max(h_x, h_y)$. Let us write (3) in terms of the linear approximation operator A^z and the error $E^z = O(h^4)$ when considering an integral in $z = x$ or y, e.g., for x as

$$\int_{x_{i-1/2}}^{\xi} u(x)\,dx = A^x(\bar{u}) + E^x(\bar{u})\,.$$

Let $\bar{u}_i(y) = \frac{1}{h_x}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x,y)\,dx$ and $\tilde{u}_j(x) = \frac{1}{h_y}\int_{y_{j-1/2}}^{y_{j+1/2}} u(x,y)\,dy$. Then

$$\begin{aligned} U_{i,j}(\xi,\eta) &= \int_{y_{j-1/2}}^{\eta}\int_{x_{i-1/2}}^{\xi} u(x,y)\,dx\,dy \\ &= \int_{y_{j-1/2}}^{\eta} \left[A^x(\bar{u}(y)) + E^x(\bar{u}(y))\right] dy \\ &= A^x\Big(\int_{y_{j-1/2}}^{\eta} \bar{u}(y)\,dy\Big) + O(h^5) \\ &= A^x\big(A^y(\bar{u}) + E^y(\bar{u})\big) + O(h^5) \\ &= A^xA^y(\bar{u}) + A^x(E^y(\bar{u})) + O(h^5), \end{aligned}$$

and

$$A^x(E^y(\bar{u})) = \int_{x_{i-1/2}}^{\xi} E^y(\tilde{u}(x))\,dx - E^x(E^y(\tilde{u}(x)) = O(h^5),$$

since the error term is smooth when u itself is smooth, and so the claim is established.

The linear weights should be modified by the two-dimensional smoothness indicator, which can be written in terms of the multi-index α as

$$IS_{k,\ell}^{i,j} := \sum_{|\alpha|=1,2} \int_{y_{j-1/2}}^{y_{j+1/2}}\int_{x_{i-1/2}}^{x_{i+1/2}} h_x^{2\alpha_1-1}\, h_y^{2\alpha_2-1} [D^\alpha P_{k,\ell}(x,y)]^2\,dx\,dy, \quad k,\ell = L,R, \tag{45}$$

using the tensor product linear, i.e., bilinear polynomials $P_{k,\ell}(x,y)$ that match the data. For example,

$$\begin{aligned} P_{L,L}(x,y) :&= \bar{u}_{i-1,j-1}\frac{(x_i - x)(y_j - y)}{h_x h_y} + \bar{u}_{i-1,j}\frac{(x_i - x)(y - y_{j-1})}{h_x h_y} \\ &\quad + \bar{u}_{i,j-1}\frac{(x - x_{i-1})(y_j - y)}{h_x h_y} + \bar{u}_{i,j}\frac{(x - x_{i-1})(y - y_{j-1})}{h_x h_y}. \end{aligned}$$

An explicit integration yields

$$IS_{L,L}^{i,j}(\bar{u}) = (\bar{u}_{i,j} - \bar{u}_{i-1,j})^2 + (\bar{u}_{i,j} - \bar{u}_{i,j-1})^2 + \frac{7}{6}(\bar{u}_{i,j} - \bar{u}_{i-1,j} - \bar{u}_{i,j-1} + \bar{u}_{i-1,j-1})^2, \tag{46}$$

and the other smoothness indicators $IS_{LR}^{i,j}(\bar{u})$, $IS_{RL}^{i,j}(\bar{u})$, $IS_{RR}^{i,j}(\bar{u})$ can be computed by symmetry. The nonlinear weights are then given by rescaling as in (15).

We approximate the first integral in (41) according to the decomposition into four grid elements

$$\begin{aligned} h_x h_y \, M^n_{i+1/2,j+1/2} &\approx \int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} u(x,y,t^n)\,dx\,dy \\ &= \int_{y_{j+1/2}}^{y_{j+1}} \int_{x_{i+1/2}}^{x_{i+1}} u(x,y,t^n)\,dx\,dy + \int_{y_j}^{y_{j+1/2}} \int_{x_{i+1/2}}^{x_{i+1}} u(x,y,t^n)\,dx\,dy \\ &\quad + \int_{y_{j+1/2}}^{y_{j+1}} \int_{x_i}^{x_{i+1/2}} u(x,y,t^n)\,dx\,dy + \int_{y_j}^{y_{j+1/2}} \int_{x_i}^{x_{i+1/2}} u(x,y,t^n)\,dx\,dy \end{aligned} \tag{47}$$

by applying (43) with $\rho = \sigma = 1/2$, or its symmetric equivalent, to these four integrals individually. In terms of the generic approximate integral operator (divided by $h_x h_y$)

$$\begin{aligned} &U(i,j,\bar{u}\,;\,i_1,i_2,i_3\,;\,j_1,j_2,j_3) := \\ &\quad \tilde{\gamma}^{i,j}_{L,L}(1/4,\bar{u})\Big[\frac{1}{64}\bar{u}_{i_1,j_1} + \frac{3}{64}\bar{u}_{i_1,j_2} + \frac{3}{64}\bar{u}_{i_2,j_1} + \frac{9}{64}\bar{u}_{i_2,j_2}\Big] \\ &+ \tilde{\gamma}^{i,j}_{L,R}(1/4,\bar{u})\Big[\frac{5}{64}\bar{u}_{i_1,j_2} - \frac{1}{64}\bar{u}_{i_1,j_3} + \frac{15}{64}\bar{u}_{i_2,j_2} - \frac{3}{64}\bar{u}_{i_2,j_3}\Big] \\ &+ \tilde{\gamma}^{i,j}_{R,L}(1/4,\bar{u})\Big[\frac{5}{64}\bar{u}_{i_2,j_1} + \frac{15}{64}\bar{u}_{i_2,j_2} - \frac{1}{64}\bar{u}_{i_3,j_1} - \frac{3}{64}\bar{u}_{i_3,j_2}\Big] \\ &+ \tilde{\gamma}^{i,j}_{R,R}(1/4,\bar{u})\Big[\frac{25}{64}\bar{u}_{i_2,j_2} - \frac{5}{64}\bar{u}_{i_2,j_3} - \frac{5}{64}\bar{u}_{i_3,j_2} + \frac{1}{64}\bar{u}_{i_3,j_3}\Big], \end{aligned} \tag{48}$$

the result is

$$\begin{aligned} M^n_{i+1/2,j+1/2} := {} & U(i+1,j+1,\bar{u}^n\,;\,i,i+1,i+2\,;\,j,j+1,j+2) \\ & + U(i+1,j,\bar{u}^n\,;\,i,i+1,i+2\,;\,j+1,j,j-1) \\ & + U(i,j+1,\bar{u}^n\,;\,i+1,i,i-1\,;\,j,j+1,j,j+2) \\ & + U(i,j,\bar{u}^n\,;\,i+1,i,i-1\,;\,j+1,j,j-1). \end{aligned} \tag{49}$$

4.2. Reconstruction of the solution at the current time t^n at the center points

In anticipation of our need to evaluate the flux terms in (41) in space and time, we first reconstruct to high order the solution at the center points of the grid elements at the current time. We do this by first re-averaging and reconstructing in y and then doing the same in x. Of course, we could do this the other way around, and we could even compute using both approaches and average the results to preserve symmetry. However, for computational efficiency, we simply use the stated approach.

For each index i, re-averaging in y produces the four values $\{\tilde{v}_{i,-1}, \tilde{v}_{i,0}, \tilde{v}_{i,1}, \tilde{v}_{i,2}\}$, using either (5)–(8) for the uniform reconstruction grid or (9)–(12) for the nonuniform reconstruction grid, but of course, using the nonlinear weights. These re-averaged values are then reconstructed using (26)–(27) or (28)–(29) on two stencils to produce the two values $\bar{v}_{i,j,L}$ and $\bar{v}_{i,j,R}$, which are averaged to obtain $\bar{v}_{i,j}$. These reconstructed values are high order accurate approximations of the averages in x, i.e., of $\frac{1}{h_x}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x,y_j,t^n)\,dx$ (because the reconstruction is a linear operation which commutes with integration in the other variable, see [9] for a proof).

Now that we have high order accurate averages in x for $y = y_j$, we can repeat the procedure in x, that is, re-average $\{\bar{v}_{i-1,j}, \bar{v}_{i,j}, \bar{v}_{i+1,j}\}$ in x using (5)–(8) or (9)–(12), reconstruct using (26)–(27) or (28)–(29), and average the results from the two stencils to obtain the high-order accurate point value

$$u^n_{i,j} \approx u(x_i, y_j, t^n).$$

Moreover, we define

$$f^n_{i,j} := f(u^n_{i,j}; x_i, y_j, t^n) \quad \text{and} \quad g^n_{i,j} := g(u^n_{i,j}; x_i, y_j, t^n).$$

We remark that a genuine two-dimensional reconstruction of the center values could be developed. It would require, say, sixteen two-dimensional re-averagings to a uniform 4×4 reconstruction grid within the (i, j)-grid element, and then a two-dimensional reconstruction on four stencils. This would entail much more work than using the approach we have presented, which involves eight one-dimensional re-averagings and four one-dimensional reconstructions (and twice this for a symmetric implementation). To be fair, the computation of (47) would be trivial in the former case, and requires four two-dimensional re-averagings in the latter case, but the work is still less using our iterated one-dimensional approach.

4.3. Evaluation of the solution in time at the center points

To handle time integration of the flux terms in (41), we again use a two-point Gauss rule, which requires approximations of the solution $u(x_i, y_j, t)$ at the Gauss times $t = t^n + \Delta t^n \theta^{\pm}_G$. Following a two-dimensional analogue of the usual Runge-Kutta procedure with the natural continuous extension outlined in Section 3.2, i.e., with an additional term involving the derivative of g with respect to y, we obtain our approximations

$$u^{n+\theta^{\pm}_G}_{i,j} \approx u(x_i, y_j, t^n + \Delta t^n \theta^{\pm}_G),$$

and set

$$f^{n+\theta^{\pm}_G}_{i,j} := f(u^{n+\theta^{\pm}_G}_{i,j}; x_i, y_j, t^{n+\theta^{\pm}_G}) \quad \text{and} \quad g^{n+\theta^{\pm}_G}_{i,j} := g(u^{n+\theta^{\pm}_G}_{i,j}; x_i, y_j, t^{n+\theta^{\pm}_G}).$$

4.4. Evaluation of the flux integrals over space and time

The evaluation of the flux integrals in (41) is accomplished by using a two-point Gauss rule in time and a WENO reconstruction of the space integral. For example, the last flux integral in (41) is

$$\frac{1}{h_x h_y} \int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} g(u)\Big|_{y=y_j} dx\, dt \approx \frac{\Delta t^n}{2 h_x h_y} \Big[\int_{x_i}^{x_{i+1}} g(u)\Big|_{y=y_j, t=t^n+\Delta t^n \theta^-_G} dx + \int_{x_i}^{x_{i+1}} g(u)\Big|_{y=y_j, t=t^n+\Delta t^n \theta^+_G} dx \Big].$$

We would like to use (3) for the approximation of the space integral, but we are using high order point values rather than element averages at this stage of the computation. Instead, we use a standard WENO reconstruction. The space integral is approximated by breaking it into two pieces at $x = x_{i+1/2}$. For example, the left piece over $[x_i, x_{i+1/2}]$ is reconstructed using $\{g^{n+\theta^{\pm}_G}_{i-1,j}, g^{n+\theta^{\pm}_G}_{i,j}, g^{n+\theta^{\pm}_G}_{i+1,j}\}$ by matching γ_L times the integral of the left linear interpolant with γ_R times the integral of the right linear interpolant to the integral of the quadratic interpolant. The linear weights turn out to

be are 1/3 and 2/3. Thus, using the nonlinear weights, we have the approximate values for each time level as $\Phi^y_{i+1/2,j}(g^{n+\theta^{\pm}_G})$, where

$$\phi^y_{i+1/2,j}(g) = \frac{1}{h_y}\Big\{\tilde\gamma^i_L(1/3, g_{(\cdot),j})\Big[-\frac{1}{8}g_{i-1,j}+\frac{5}{8}g_{i,j}\Big] + \tilde\gamma^i_R(2/3, g_{(\cdot),j})\Big[\frac{3}{8}g_{i,j}+\frac{1}{8}g_{i+1,j}\Big] + \tilde\gamma^{i+1}_L(2/3, g_{(\cdot),j})\Big[\frac{1}{8}g_{i,j}+\frac{3}{8}g_{i+1,j}\Big] + \tilde\gamma^{i+1}_R(1/3, g_{(\cdot),j})\Big[\frac{5}{8}g_{i+1,j}-\frac{1}{8}g_{i+2,j}\Big]\Big\}. \tag{50}$$

Similarly, define

$$\phi^x_{i,j+1/2}(f) = \frac{1}{h_x}\Big\{\tilde\gamma^j_L(1/3, f_{i,(\cdot)})\Big[-\frac{1}{8}f_{i,j-1}+\frac{5}{8}f_{i,j}\Big] + \tilde\gamma^j_R(2/3, f_{i,(\cdot)})\Big[\frac{3}{8}f_{i,j}+\frac{1}{8}f_{i,j+1}\Big] + \tilde\gamma^{j+1}_L(2/3, f_{i,(\cdot)})\Big[\frac{1}{8}f_{i,j}+\frac{3}{8}f_{i,j+1}\Big] + \tilde\gamma^{j+1}_R(1/3, f_{i,(\cdot)})\Big[\frac{5}{8}f_{i,j+1}-\frac{1}{8}f_{i,j+2}\Big]\Big\}. \tag{51}$$

Then the sum of the four flux integrals in (41) is

$$\Phi^n_{i+1.2,j+1/2} = \frac{\Delta t^n}{2}\Big\{-\Phi^x_{i+1,j+1/2}(f^{n+\theta^-_G}) - \Phi^x_{i+1,j+1/2}(f^{n+\theta^+_G}) + \Phi^x_{i,j+1/2}(f^{n+\theta^-_G}) + \Phi^x_{i,j+1/2}(f^{n+\theta^+_G}) - \Phi^y_{i+1/2,j+1}(g^{n+\theta^-_G}) - \Phi^y_{i+1/2,j+1}(g^{n+\theta^+_G}) + \Phi^y_{i+1/2,j}(g^{n+\theta^-_G}) + \Phi^y_{i+1/2,j}(g^{n+\theta^+_G})\Big\}. \tag{52}$$

4.5. Summary of the scheme in two space dimensions

We finally summarize the scheme based on (41). We first use (49) and (48) to compute the mass $M^n_{i+1/2,j+1/2}$ at time t^n. We then define the center point values $u^n_{i,j}$, $f^n_{i,j}$, and $g^n_{i,j}$ using the re-averaging and reconstruction procedure of Section 4.2, and then produce $u^{n+\theta^{\pm}_G}_{i,j}$, $f^{n+\theta^{\pm}_G}_{i,j}$, and $g^{n+\theta^{\pm}_G}_{i,j}$ using the Runge-Kutta with natural continuous extension procedure of Section 4.3. Finally, we compute the flux $\Phi^n_{i+1.2,j+1/2}$ using (50)–(52), and set

$$\bar u^{n+1}_{i+1/2,j+1/2} := M^n_{i+1/2,j+1/2} + \Phi^n_{i+1/2,j+1/2}. \tag{53}$$

5. Numerical results for problems in one space dimension

In most works, the nonlinear weights (15) use the parameter ϵ = 1E-6 [3]. However, in [8], it was found that augmented CWENO3 works better when using larger values of ϵ (say, ϵ = 1E-4 or ϵ = 1E-2). The larger the value of ϵ, the more the nonlinear weights resemble the linear ones. Herein we sometimes vary ϵ from its usual value. In the second example, we will also experiment with changing the strength of the measure of smoothness.

All our test results use periodic boundary conditions.

5.1. Example 1, constant linear transport

We first test our scheme in the simple case of constant linear transport as applied to

$$u_t + u_x = 0, \quad x \in (0, 2), \quad u_0(x) = 0.75 + 0.25\sin(\pi x).$$

Table 1: Ex. 1, linear transport. Error and convergence order at $T = 10$ with $\Delta t = 0.2h$ and $h = 2/m$. We use $\epsilon =$ 1E-2 in the nonlinear weights and the uniform reconstruction grid.

m	L_h^1 error	order	L_h^∞ error	order
10	2.31287E-01	——	1.78861E-01	——
20	4.67840E-02	2.30560	4.17714E-02	2.09825
40	4.16134E-03	3.49090	4.72872E-03	3.14299
80	4.89365E-04	3.08806	4.57875E-04	3.36843
160	6.10702E-05	3.00237	5.04137E-05	3.18307
320	7.63440E-06	2.99988	6.07320E-06	3.05329
640	9.54460E-07	2.99976	7.52046E-07	3.01356
1280	1.19299E-07	3.00010	9.37727E-08	3.00358

Table 2: Ex. 1, linear transport. Error and convergence order at $T = 10$ with $\Delta t = 0.2h$ and $h = 2/m$. We use $\epsilon =$ 1E-2 in the nonlinear weights and the nonuniform reconstruction grid.

m	L_h^1 error	order	L_h^∞ error	order
10	2.26608E-01	——	1.75287E-01	——
20	4.41577E-02	2.35946	3.92850E-02	2.15767
40	4.11885E-03	3.42235	4.49674E-03	3.12703
80	4.89164E-04	3.07385	4.47272E-04	3.32966
160	6.10693E-05	3.00180	5.00641E-05	3.15930
320	7.63440E-06	2.99986	6.06220E-06	3.04586
640	9.54460E-07	2.99976	7.51702E-07	3.01161
1280	1.19299E-07	3.00010	9.37620E-08	3.00309

Table 3: Ex. 1, linear transport. Error and convergence order at $T = 10$ with $\Delta t = 0.2h$ and $h = 2/m$. We use $\epsilon =$ 1E-4 in the nonlinear weights and the uniform reconstruction grid.

m	L_h^1 error	order	L_h^∞ error	order
10	2.92394E-01	——	2.25889E-01	——
20	1.64297E-01	0.83161	1.41641E-01	0.67338
40	3.13130E-02	2.39147	3.56153E-02	1.99167
80	3.99845E-03	2.96925	5.04667E-03	2.81909
160	1.93416E-04	4.36966	2.84825E-04	4.14718
320	9.51802E-06	4.34490	1.37065E-05	4.37714
640	9.54468E-07	3.31789	9.91066E-07	3.78974
1280	1.19299E-07	3.00011	1.01243E-07	3.29116

For this smooth problem, we can simply use the linear weights (i.e., we can make no nonlinear smoothness correction to the linear weights). In this case, we observe a clean third order rate of convergence for the scheme, as expected. However, the scheme will normally be applied to problems that have non-smooth behavior, and so we need to test the performance of the nonlinear weights.

The numerical results using the nonlinear weights are not as clean. We show L_h^1 and L_h^∞ errors for the nonlinear weights with a large ϵ = 1E-2 in Tables 1–2, comparing the uniform and nonuniform reconstruction grids. In this case, we see a fairly clean third order of convergence. We also note that there is little difference between the results for uniform and nonuniform reconstruction grids. In Table 3 we show the same comparison but using a smaller ϵ = 1E-4 for the uniform reconstruction grid (results for the nonuniform grid are very similar). The convergence is not clean, but it appears to be third order. Moreover, the errors are comparable to the previous case of a larger ϵ.

Table 4: Ex. 1, linear transport with the alternate initial condition $u_0(x) = \sin^4(\pi x)$. Error and convergence order at $T = 1$ with $\Delta t = 0.2h$ and $h = 2/m$. We use ϵ = 1E-2 in the nonlinear weights and the uniform reconstruction grid.

m	L_h^1 error	order	L_h^∞ error	order
10	6.51455E-01	——	4.19944E-01	——
20	3.19967E-01	1.02574	3.64131E-01	0.20574
40	8.84790E-02	1.85452	1.32501E-01	1.45845
80	1.07544E-02	3.04041	2.11501E-02	2.64727
160	9.36958E-04	3.52080	1.67371E-03	3.65955
320	1.01509E-04	3.20638	1.27499E-04	3.71449
640	1.24687E-05	3.02522	1.29981E-05	3.29411
1280	1.55225E-06	3.00588	1.52999E-06	3.08671

We also test our scheme with the initial condition $u_0(x) = \sin^4(\pi x)$. The results are shown in Table 4 (for the uniform reconstruction grid and ϵ = 1E-2). The scheme gives a good approximation in this case, which has an initial discontinuity in the derivative at $x = 1$ and $x = 2$.

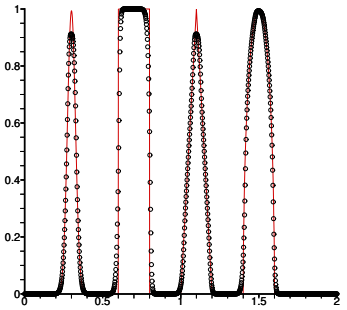
Finally, we remark that the augmented CWENO3 scheme of Levy, Puppo, and Russo [7, 8] show results similar to those obtained by our CWENO3 scheme on these tests.

5.2. Example 2, Shu's linear test

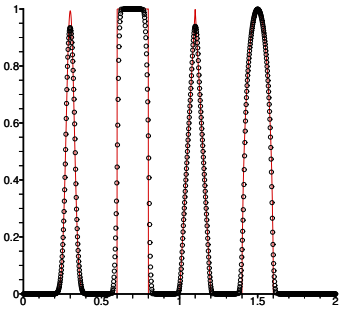
We take a standard test problem, called *Shu's linear test,* defined for $x \in [0, 2]$, and solve it for various choices of parameters. The main results use a mesh of $m = 1000$ grid points, which is reasonable for comparison to other papers using CWENO5 with about 200 points. Our CWENO3 scheme drops to second order near the discontinuities, for an error $O((2/1000)^2) = O(4 \times 10^{-6})$ versus CWENO5, which is third order, for an accuracy of $O((2/200)^3) = O(10^{-6})$. Results are shown after linear transport by speed one at time $T = 2.0$ using $\Delta t = 0.4h$.

As we saw in the previous example, a larger value of ϵ in the nonlinear weights (15) can produce a cleaner convergence rate on a smooth problem, since this biases the nonlinear weights toward being the linear weights. However, a larger ϵ does *not* work well for non smooth examples, such as Shu's linear test, since the use of the unmodified linear weights leads to oscillations in the solution. As an alternative, we can increase the strength of the measure of smoothness.

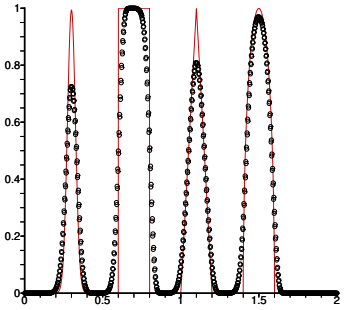
Instead of using the L^2-measure of smoothness as given above in (13), we use the L^p-measure



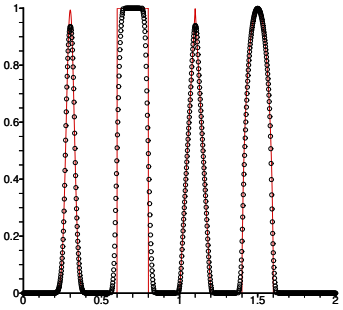
$m = 1000$, $p = 2$, uniform



$m = 1000$, $p = 4$, uniform



$m = 1000$, $p = 2$, nonuniform



$m = 1000$, $p = 4$, nonuniform

Figure 5: Ex. 2, Shu's Linear test. The solution using $m = 1000$ grid points ($h = 0.002$) and $\Delta t = 0.4h$ at time $T = 2.0$. The top row of graphs show results using the uniform reconstruction grid, and the bottom row uses the nonuniform reconstruction grid. The left column uses $p = 2$ in the nonlinear weights, and the right column uses $p = 4$. The results suggest that the uniform reconstruction grid and the greater p is more accurate.

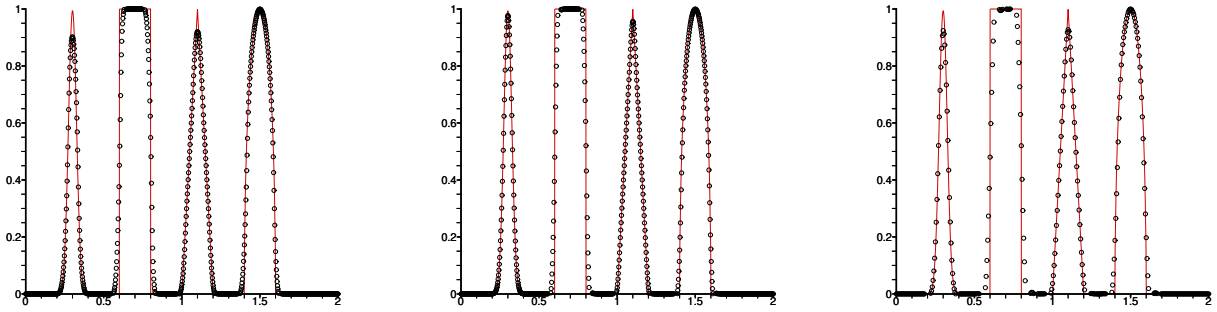
for $p > 2$. That is, we define

$$IS_k^{p,i} := \sum_{\ell=1}^{2} \int_{x_{i-1/p}}^{x_{i+1/p}} h^{p\ell-1} \left[\frac{\partial^\ell P_k(x)}{\partial x^\ell} \right]^p dx, \quad k = L, R. \tag{54}$$

In our case of linear polynomials, the new measure of smoothness is given simply by replacing the exponent 2 by p in (14) for the uniform grid and in (30) and (31) for the nonuniform grid. The nonlinear weights are still defined by (15), using expressions like $(\epsilon + IS_k^{p,i})^2$, $k = L, R$, as opposed to the more standard way of increasing the strength, which involves the expressions $(\epsilon + IS_k^{2,i})^p$ (see [3]). The reason for using this alternate way of increasing the strength is that, in our case, increasing p does not change the strength of the parameter ϵ.

Our main results compare the choice of p in the nonlinear weights (using a fixed $\epsilon =$ 1E-8) and the choice of reconstruction grid. As shown in Fig. 5, at least for this test problem, we see better accuracy when $p = 4$ as opposed to $p = 2$. We also see better accuracy using the uniform, versus the nonuniform, reconstruction grid.

We also varied m to show that accurate results can be obtained for this example using a coarse grid, as long as p in the nonlinear weights is taken to be large enough ($p = 6$ or $p = 8$). The results appear in Fig. 6. It seems to be an open question as to the best choice of nonlinear weight parameters. The case $p = 2$ is generally considered to be adequate [3], but it is not the best



$m = 800$, $p = 4$ $m = 800$, $p = 6$ $m = 400$, $p = 8$

Figure 6: Ex 2, Shu's Linear test. We obtain good results on relatively coarse grids provided a larger p is used in the nonlinear weights. Here the uniform reconstruction grid is used.

choice in this example. However, the Buckley-Leverett test of Section 5.4 showed no significant difference between using $p = 2$ versus $p = 4$, so our results are far from conclusive in the general case.

5.3. Example 3, Burgers' equation

In this example we test Burgers' equation with a simple initial condition,

$$u_t + (u^2/2)_x = 0, \quad x \in (0, 2), \quad u_0(x) = 0.75 + 0.25\sin(\pi x).$$

With ϵ = 1E-4 in the nonlinear weights, we see third order convergence in Table 5.

Table 5: Ex. 3, Burgers' equation. Error and convergence order at $T = 1$ with $\Delta t = 0.2h$ and $h = 2/m$. We use ϵ = 1E-4 in the nonlinear weights and the uniform reconstruction grid.

m	L_h^1 error	order	L_h^∞ error	order
10	9.30494E-02	——	1.22347E-01	——
20	2.48902E-02	1.90242	6.09936E-02	1.00425
40	5.09583E-03	2.28819	2.82353E-02	1.11116
80	9.46210E-04	2.42909	8.57124E-03	1.71992
160	1.56259E-04	2.59822	2.00149E-03	2.09843
320	2.18843E-05	2.83597	3.53081E-04	2.50300
640	2.82913E-06	2.95146	4.90373E-05	2.84805
1280	3.52475E-07	3.00477	6.19008E-06	2.98585

We now study Burgers' equation with a combination of a shock and a rarefaction. The initial condition is changed to

$$u_0(x) = \begin{cases} 0.5, & \text{if } x \le 0.3 \text{ and } x > 0.75, \\ 1.0, & \text{if } 0.3 < x \le 0.75. \end{cases}$$

With only $m = 80$ grid points, we see good accuracy in Fig. 7 compared to the augmented CWENO3 [7, 8].

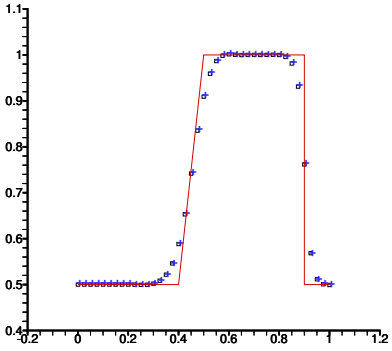
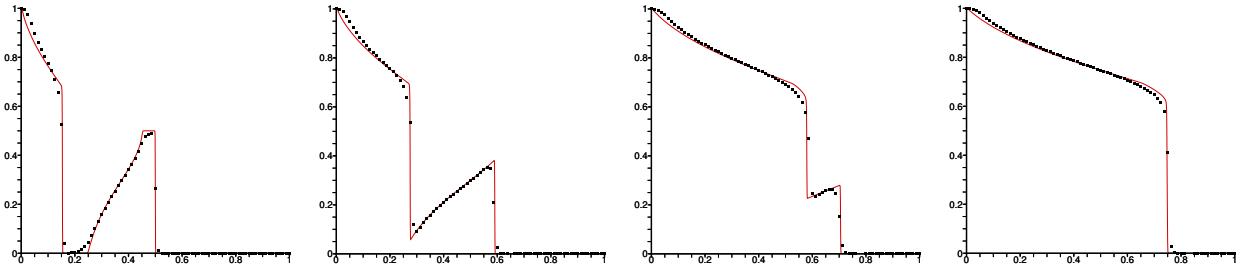


Figure 7: Ex. 3, Burgers' equation with a shock and rarefaction. The solution using $m = 80$ for our CWENO3 scheme, shown in black squares, and the augmented CWENO3 scheme, shown in blue crosses. (We use the uniform reconstruction grid.) The two schemes are comparable in accuracy.



$t = 0.1$, shock and rarefaction formation

$t = 0.2$, the initial pulse catches the advanced pulse

$t = 0.4$, the two pulses merge

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

Figure 8: Ex. 4, Buckley-Leverett equation. An interaction of shocks and rarefactions, resulting from the evolution of the initial condition of two pulses given in (55). The red line is the reference solution, given by CWENO5 with a very small $h = 1/1280$ and $\Delta t = 1/15360$. The black squares are our CWENO3 results using $m = 80$ and $\Delta t = 0.2h$.

5.4. Example 4, Buckley-Leverett equation

This last example involves (20) with the Buckley-Leverett flux function

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

and an interaction of shocks and rarefactions. The initial condition is

$$u_0(x) = \begin{cases} 1 - 20x & \text{for } 0 \le x \le 0.05, \\ 0.5 & \text{for } 0.25 \le x \le 0.4, \\ 0 & \text{otherwise,} \end{cases} \tag{55}$$

and has two pulses that merge in time. We use $m = 80$ grid points and $p = 2$ in the nonlinear weights (no major difference was observed using $p = 4$). The results are shown in Fig. 8, and the scheme handles the merging of the two pulses quite well for using only 80 grid points.

6. Application to the Euler system

For a polytropic 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 γ is 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

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}_x = 0. \tag{56}$$

To respect the characteristic structure of the system, it is usual to reconstruct the needed information using the linearized system. We expand (56) into a system of the form $U_t + A(U)U_x = 0$, that is, into

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)u^2 & (3-\gamma)u & \gamma-1 \\ \frac{1}{2}(\gamma-1)u^3 - \dfrac{u(E+p)}{\rho} & \dfrac{(E+p)}{\rho} - (\gamma-1)u^2 & \gamma u \end{pmatrix} \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_x = 0.$$

The linearized equation can be diagonalized as $\Lambda^0 = L^0 A(U^0) (L^0)^{-1}$, giving

$$L^0 \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} u^0 - c^0 & 0 & 0 \\ 0 & u^0 & 0 \\ 0 & 0 & u^0 - c^0 \end{pmatrix} L^0 \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_x = 0,$$

where $c = \sqrt{(\gamma p)/\rho}$ is the sound speed. The re-averaging of Section 2 and the reconstructions needed for the integral over the previous time level, i.e., (23), are performed on the diagonalized variables $L^0(\rho, \rho u, E)$. We then transform back to the original variables $(\rho, \rho u, E)$ by applying $(L^0)^{-1}$, and the time evolution of the scheme continues for the nonlinear system (56), as described above in Section 3.

6.1. Example 5, Riemann problems for the Euler equations

For this series of 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.

Sod's 1-D shock tube test. The one-dimensional shock tube test of Sod uses the initial condition

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

The results are shown in Fig. 9. Good results are obtained using $m = 200$ and $m = 400$ grid points.

Lax's 1-D shock tube test. The one-dimensional shock tube test of Lax uses the initial condition

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

Reasonably good results are shown in Fig. 10, using $m = 200$ and $m = 400$ grid points.

Figure 9: Ex. 5, Sod's 1-D shock tube test. The density profile at time $T = 0.16$ using $\Delta t = 0.1h$ and $m = 200$ (left) and $m = 400$ (right) grid points.

Figure 10: Ex. 5, Lax's 1-D shock tube test. The density profile at time $T = 0.16$ using $\Delta t = 0.05h$ and $m = 200$ (left) and $m = 400$ (right) grid points.

Figure 11: Ex. 5, Arora and Roe's Mach 3 shock tube test. The density profile at time $T = 0.09$ using $\Delta t = 0.05h$ and $m = 200$ (left) and $m = 400$ (right) grid points.

Arora and Roe's Mach 3 shock tube test. The mach 3 shock tube test of Arora and Roe has the initial condition

$$\rho, u, p = \begin{cases} \rho_l = 3.857, u_l = 0.92, p_l = 10.333, & \text{for } x < 1/2, \\ \rho_r = 1, u_r = 3.55, p_r = 1, & \text{for } x > 1/2. \end{cases}$$

Good results are shown in Fig. 11, using $m = 200$ and $m = 400$ grid points.

Figure 12: Ex. 6, Woodward and Colella's double blast test. The density profile at time $T = 0.038$ using $m = 800$ (left) and $m = 1200$ (right) grid points. The fine resolution reference solution is shown in red.

6.2. Example 6, Woodward and Colella's double blast test

The double blast test of Woodward and Colella 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 difficult problem. Nevertheless, reasonably good results are obtained by our CWENO3 scheme using $m = 800$ and $m = 1200$ grid points, as shown in Fig. 12. The reference solution was computed by a MUSCL scheme with grid size $m = 4000$.

7. Numerical results for problems in two space dimensions

We now discuss some numerical examples in two space dimensions.

7.1. Example 7, Rigid body rotation

We first test the convergence rate of the scheme as applied to a smooth problem involving two dimensional rigid body rotation (cf. [9]). The problem is

$$u_t - ((y-1)u)_x + ((x-1)u)_y = 0, \quad x \in [0,2],\ y \in [0,2],\ t > 0, \tag{57}$$

with a smooth, radial bump function as the initial condition, defined as

$$\begin{aligned} u(x,y,0) &= \tfrac{2}{5}[\psi(1 + r(x,y))\psi(1 - r(x,y)) + 1], \\ r(x,y) &= \sqrt{(x-1)^2 + (y-1)^2}, \end{aligned}$$

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

We solve the problem using a time step that corresponds to a fixed CFL number of 0.1, which is $\Delta t = 0.1h$, for a given $h = h_x = h_y = 2/m$ grid elements. In Table 6 we see the L_h^1 and L_h^∞ norms of the errors and the corresponding orders of convergence as m increases. Third order convergence is observed, as expected.

Table 6: Ex. 7, smooth rigid body rotation. Errors and convergence order at $t = 2\pi$, using $\Delta t = 0.1h$ (CFL = 0.1), where $h = h_x = h_y = 2/m$.

m	L_h^1 error	order	L_h^∞ error	order
20	1.62287E-02	———	3.32619E-02	———
40	6.36774E-03	1.34970	1.58469E-02	1.06967
80	1.04839E-03	2.60260	2.56955E-03	2.62461
160	1.45170E-04	2.85236	1.83471E-04	3.80790
320	1.80685E-05	3.00620	2.25254E-05	3.02592
640	2.24239E-06	3.01036	2.72756E-06	3.04587

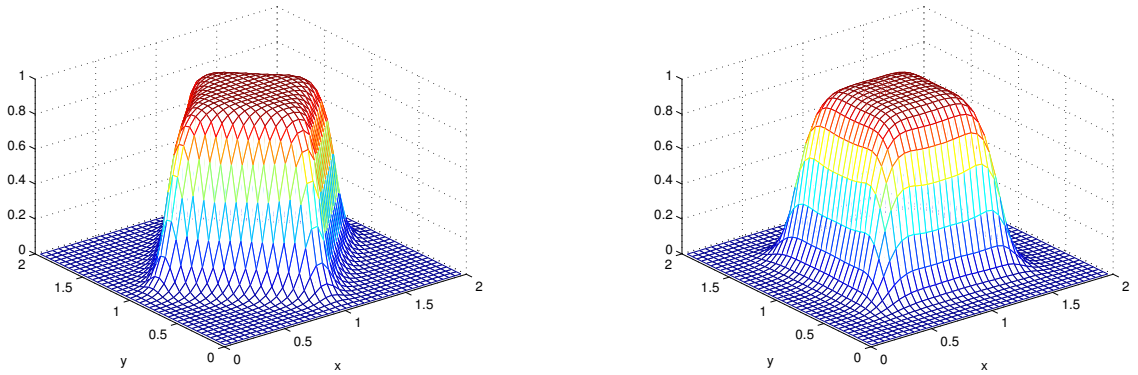


Figure 13: Ex. 7, rigid body rotation of a square. One eighth and one quarter rotation of a square using 40 grid elements in each coordinate direction and Δt is 0.425 of the CFL limit.

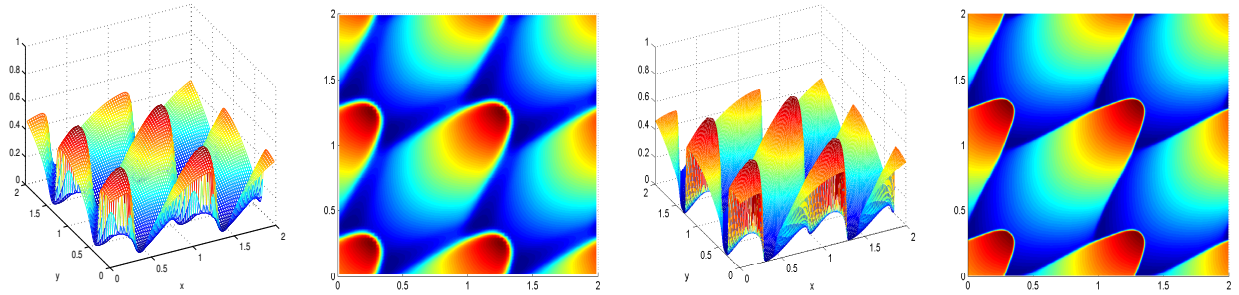


Figure 14: Ex. 8, a Burgers' problem in two-dimensions. The solution at time $T = 1.5$ is shown in profile and as contours using $h = 1/80$ for the two plots on the left and $h = 1/320$ on the right, with Δt being 0.425 of the CFL step.

We next show the solution for a rigid body rotation of a square (cf. [7, 8]), to see how well the scheme handles discontinuous solutions. The domain is $[0, 2]^2$, and the solution vanishes outside the smaller square $[1/2, 3/2]^2$. Using 40 grid elements in each coordinate direction and with Δt being 0.425 of the CFL limit, the solution is shown in Fig. 13 after an eight and a quarter revolution. The discontinuity is captured relatively well, with no spurious oscillations, even though the method is only formally third order accurate and the mesh is relatively coarse (the results are comparable to those in [8]).

7.2. Example 8, A Burgers' problem in two-dimensions

The last example is the nonlinear, two-dimensional Burgers equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) + \frac{\partial}{\partial y}\left(\frac{1}{2}u^2\right) = 0, \quad x \in [0,2],\ y \in [0,2],\ t > 0, \tag{58}$$

with the initial condition $u(x, y, 0) = \sin^2(\pi x)\sin^2(\pi y)$, using periodic boundary conditions. In Fig. 14 we present the solution obtained at time $T = 1.5$ with two different mesh spacings, 80 and 320 elements in each coordinate direction. The time step Δt is chosen to correspond to a CFL number of 0.425. One can easily see that, although shocks develop in the solution, they are well resolved and exhibit no spurious oscillations. The result is comparable to [8, Fig. 4.5], although in that paper it appears that the solution has been plotted in reverse in both directions (and they show only one-quarter of what we plot here).

8. Conclusions

We have presented a WENO re-averaging (or re-mapping) technique that converts function averages to a new grid and to high order. Nonlinear weighting ensures that we maintain the essentially non-oscillatory property of the re-averaged function values. We call the new grid the reconstruction grid, since we use it to obtain other, standard high order WENO reconstructions of the function averages, such as high order point values. By choosing the reconstruction grid to include a point of interest, we can reconstruct a high order function value using only positive weights.

We applied the re-averaging technique to define a classic CWENO3 scheme in one space dimension that combines two linear polynomials to obtain formal third order accuracy. Such a scheme cannot otherwise be defined, due to the nonexistence of linear weights for high order reconstruction at the center of a grid element. Our new scheme uses a compact stencil of three solution averages in the reconstrutions (and five for the overall scheme, including the Runge-Kutta step). Stencils are combined using only positive linear weights in both the re-averaging step and in the step for function reconstruction at the center of the computational grid element (which is now at a grid point of the reconstruction grid). There are actually two variants, depending on whether one chooses to use a uniform or a nonuniform reconstruction grid.

The new CWENO3 scheme is different from what we call the augmented CWENO3 scheme of Levy, Puppo, and Russo [7, 8], since their scheme uses directly the quadratic approximation in addition to the linear ones.

There are classic CWENO schemes of order five, nine, etc., so we do not discuss these orders. Prior to this work, there were are no classic CWENO schemes of order three, seven, eleven, etc. due to the nonexistence of the linear weights for reconstruction at the grid element center points for these orders of accuracy. The CWENO3 scheme was given in detail, and the novel re-averaging and reconstruction at center points idea that we presented allows one to define the other order schemes in a straightforward way. The key is to be able to re-average the function to a reconstruction grid to any order of accuracy. A proof that this can indeed be done, in fact on nonuniform grids, is presented in [15, 10].

We extended the scheme to approximate problems with two space dimensions. The extension was developed easily in an essentially tensor product form, iterating the one space dimensional scheme and maintaining the compactness of the stencil. The extension is therefore relatively straightforward to implement, and it is computationally efficient. We proved that the scheme maintains formal third order accuracy. Due to the way that the two-dimensional scheme is defined, extension to problems in higher space dimensions is straightforward to implement as well.

The numerical results show that our CWENO3 scheme behaves as expected. It is third order accurate for smooth problems. It also gives good results for non-smooth problems, including Riemann problems for the Burgers, Buckley-Leverett, and Euler equations.

Acknowledgment

We thank Professor Feng Xiao of the Department of Energy Science, Tokyo Institute of Technology, for providing his MUSCL scheme code, which we used to obtain the reference solution for Woodward and Colella's double blast test.

References

[1] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy, Uniformly high-order accurate essentially nonoscillatory schemes III, J. Comput. Phys. 71 (2) (1987) 231–303.

[2] A. Harten, S. Osher, Uniformly high-order accurate nonoscillatory schemes I, SIAM J. Numer. Anal. 24 (2) (1987) 279–309.

[3] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.

[4] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, Math. Model. Numer. Anal. 33 (1999) 547–571.

[5] X. D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.

[6] J. Qiu, C.-W. Shu, On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes, J. Comput. Phys. 183 (2002) 187–209.

[7] D. Levy, G. Puppo, G. Russo, A third order central WENO scheme for 2D conservation laws, Appl. Numer. Math. 33 (2000) 415–421.

[8] D. Levy, G. Puppo, G. Russo, Compact central WENO schemes for multidimensional conservation laws, SIAM J. Sci. Comput. 22 (2) (2000) 656–672.

[9] C.-S. Huang, T. Arbogast, J. Qiu, An Eulerian-Lagrangian WENO finite volume scheme for advection problems, J. Comput. Phys. 231 (11) (2012) 4028–4052, DOI 10.1016/j.jcp.2012.01.030.

[10] E. Carlini, R. Ferretti, and G. Russo, A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations, SIAM J. Sci. Comput. 27 (2005) 1071–1091.

[11] Y.-Y. Liu, C.-W. Shu, M.-P. Zhang, On the positivity of linear weights in WENO approximations, Acta Mathematicae Applicatae Sinica 25 (2009) 503–538.

[12] D. S. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, J. Comput. Phys. 160 (2000) 405–452.

[13] G. A. Gerolymos, D. Séneéchal, I. Vallet, Very-high-order WENO schemes, J. Comput. Phys. 228 (2009) 8481–8524.

[14] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (1) (2002) 108–127.

[15] C.-S. Huang, T. Arbogast, An Eulerian-Lagrangian WENO scheme for nonlinear conservation laws, submitted, 2013.

[16] M. Zennaro, Natural continuous extensions of Runge-Kutta methods, Math. Comp. 46 (1986) 119–133.