Introduction to WENO Methods

WENO Reconstruction for Scalar Conservation Laws

Tommaso Buvoli
Krithika Manohar

February 2015

1 Introduction

Finite-difference methods have proven effective for solving a wide variety of partial differential equations (PDEs). For PDEs with smooth solutions, it is beneficial to consider high-order finite-differences (spectral methods) which can help preserve solution features and improve computational efficiency. The situation is markedly different when considering discontinous initial conditions or equations that can lead to shock formation. Essentially non-oscillatory or ENO methods have historically proven successful at capturing shock discontinuities by choosing the smoothest interpolating polynomial at several neighboring stencils. Weighted ENO, or WENO improves this methodology further by considering a convex combination of lower order interpolating polynomials, hence preserving high-resolution for shocks and high-order accuracy for smooth data.

The goals of this paper are two-fold. First we seek to summarize the WENO methodology for interpolation and reconstruction as investigated in [1, 2, 5]. Secondly, we implement and analyze the performance of WENO finite-volume schemes on one-dimensional hyperbolic equations such as the advection equation and Burgers' equation. We compare our results to those obtained using `clawpack`, and briefly investigate the benefits of using total variation diminishing (TVD) Runge-Kutta schemes versus traditional non-TVD integrators.

2 WENO Interpolation and Reconstruction

Consider the problem of constructing the function $L(x)$ that passes though a set of data $\{f_j\}_{j=0}^{N}$ at distinct node locations $\{\mathbf{x}_j\}_{j=0}^{N}$. Polynomial interpolation, rational interpolation, and radial basis functions perform excellent when the data f_j originates from a sufficiently smooth function $f(x)$ such that $f_j = f(x_j)$. However, these traditional methods often introduce unwanted oscillations when applied to discontinuous data even when Chebyshev or Gauss–Legendre nodes are used. WENO methods can be used to overcome this difficulty and provide high-order accuracy in smooth regions, as well as "oscillation free" approximations near jumps or higher-order discontinuities. We describe the WENO methodology for interpolation in the following section, and later generalize this approach for function reconstruction from cell averages.

2.1 Interpolation

We seek an approximation to the function $f(x)$, given $2r-1$ equispaced data points $\{(x_j, f_j)\}_{j=0}^{2r-1}$ where $\Delta x = x_{i+1} - x_i$. Consider the set of r interpolating polynomials $\{P_k(x)\}_{k=1}^{r}$, where the kth polynomial passes through the points $\{(x_j, f_j)\}_{j=k}^{k+r-1}$ (Figure 1). The polynomials $P_k(x)$ each provide an $\mathrm{O}(\Delta x^r)$ approximation to $f(x)$ on the interval $[x_1, x_{2r-1}]$. In cases where the data f_j is discontinuous, we should select the polynomial which introduces the fewest oscillations. Conversely, when the data f_j is smooth, we should instead consider the interpolating polynomial $L(x)$ which passes through all $2r-1$ points and provides $\mathrm{O}(\Delta x^{2r-1})$ accuracy.

We begin our discussion of WENO by noting that the polynomial $L(x)$ can be written as a linear combination of the polynomials $P_k(x)$ such that

$$L(x) = \sum_{i=0}^{r} C_k^r(x) P_k(x)$$

where $C_k^r(x)$ are rth order polynomials in x (see appendix A for obtaining $C_k^r(x)$). The WENO methodology which we present in this paper is only valid for problems where the weights $C_k^r(x)$ are convex, which is often the case. A modification to the WENO procedure has been proposed to overcome negative weights [4] but we do not consider it here. The classical WENO methodology approximates $f(x)$ via a convex combination of the polynomials $P_j(x)$ so that

$$f(x) \approx \sum_{k=1}^{r} w_k^r(x) P_k(x).$$

The weights $w_k(x)$ should be chosen so that when $f(x)$ is smooth everywhere, $w_k(x) \approx C_k^r(x)$, and when $f(x)$ is discontinuous inside the interval $[x_k, x_k + r]$, $w(x) \approx 0$. In order to determine $w(x)$, we must define a smoothness measure for each of the polynomials $P_k(x)$. As first proposed in [1], we consider the smoothness measure

$$S_k(x) = \sum_{i=1}^{r} (\Delta x)^{2i-1} \int_{x-\Delta x/2}^{x+\Delta x/2} \left(\frac{d^i P_k(x)}{dx^i} \right)^2 dx \tag{1}$$

which is a sum over the two-norms of the derivatives, independent of the grid spacing Δx. We then use the constants S_j to determine the nonlinear WENO weights

$$w_k(x) = \frac{\alpha_k}{\sum_{i=1}^{r} \alpha_i}, \qquad \text{and} \qquad \alpha_j(x) = \frac{C_k^r(x)}{(\epsilon + S_j(x))^p} \tag{2}$$

where ϵ is commonly taken to be 10^{-6} and $p = 2$. Larger ϵ and smaller $p > 0$ lead to WENO weights that more often approximate the optimal values $C_k^r(x)$. From Taylor analysis, it follows that $S_j(x) = \alpha(x) + O(h^{r-1})$ (where α does not depend on k) garantees that the WENO approximation will be $2r-1$ accurate in smooth regions. Though we do not demostrate this here, this condition has been shown to hold for Eq. (1) in [1]. The weights and smoothness indicators are not unique, and other choices may be beneficial for certain applications [5]. We provide coefficients C_k and measures $S_k(x)$ for $r = 2$ and $r = 3$ in Appendix A.

2.2 Conservation Laws and Reconstruction

2.2.1 Solving Conservation Laws

We seek to numerically integrate the conservation law

$$q_t + f(q)_x = 0.$$

Integrating once in time, once in space, and appropriately exchanging the order of integration leads to

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \left[q(x, t^{n+1}) - q(x, t^n) \right] dt = - \int_{t^n}^{t^{n+1}} \left[f(q(x_{i+1/2}, t)) - f(q(x_{i-1/2}, t)) \right] dt.$$

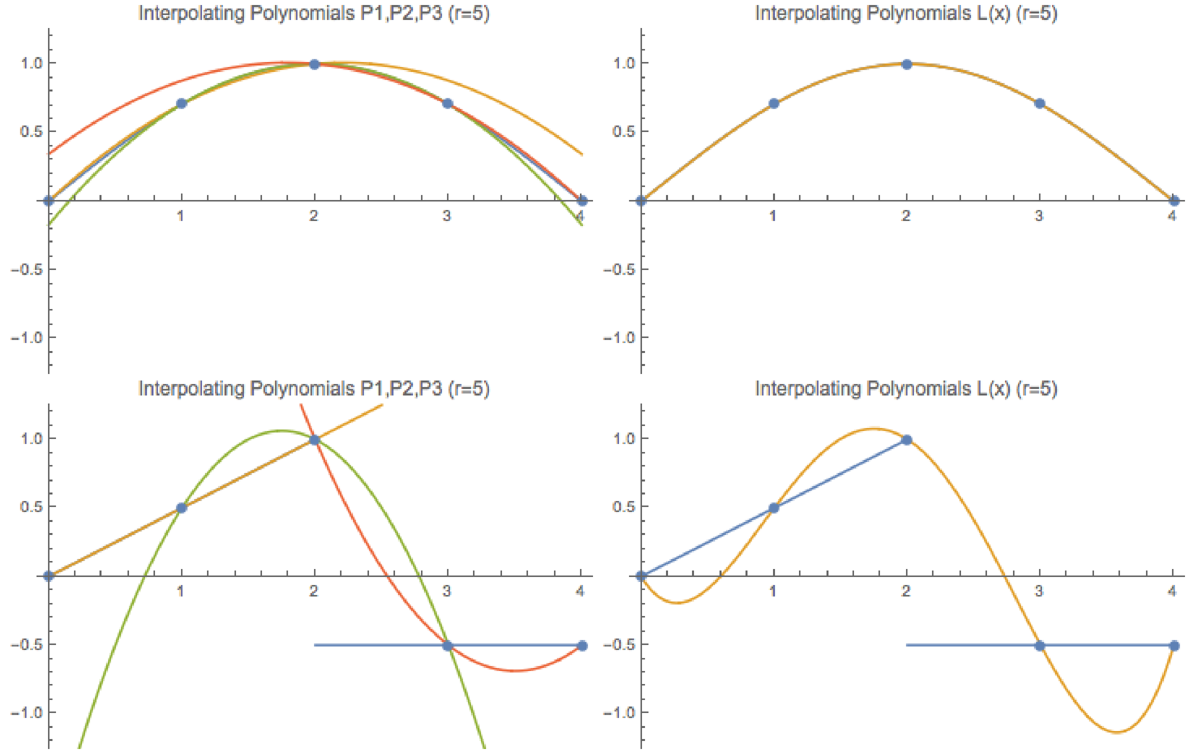


Figure 1: We consider a smooth function $f_s(x) = \sin(\pi x/4)$ (blue curve in top 2 figures) and the discontinuous function

$$f_d(x) = \begin{cases} \frac{x}{2} & x \leq 2 \\ -\frac{1}{2} & x > 1 \end{cases}$$

(blue curve in bottom two figures). We take $r = 3$ and $\Delta x = 1$ and compare the interpolating polynomials $S_1(x)$, $S_2(x)$, $S_3(x)$ (respectively colored in yellow, green, red) to the interpolant $L(x)$ which passes through all five points. Notice that for the smooth function, the interpolant $L(x)$ provides a visibly better approximation, whereas $S_1(x)$, $S_2(x)$ and $S_3(x)$ provide better approximations in the appropriate interval.

Dividing through by $\Delta x = (x_i - x_{i-1})$, and letting Q_i^n denote the ith cell average of $q(x, t^n)$ and $q_{i-1/2}(t)$ denote $q(x_{i-1/2}, t)$, this becomes

$$Q_i^{n+1} = Q_{i-1}^n + \frac{1}{\Delta x} \int_{t^n}^{t^{n+1}} \left[f(q_{i-1/2}(t)) - f(q_{i+1/2}(t)) \right] dt$$

We can define $\mathbf{Q} = [Q_1, \ldots Q_n]^T$, and rewrite this system compactly as

$$\begin{cases} \frac{d}{dt}\mathbf{Q} = L(\mathbf{Q}) \\ \mathbf{Q}(x, t = 0) = Q_0 \end{cases} \qquad \text{where,} \qquad L(Q)_i = f(q_{i-1}) - f(q_i). \tag{3}$$

In order to integrate Eq. (3), we require a method for estimating $f(q_{i-1/2})$ and $f(q_{i+1/2})$ from the cell averages Q_i. In the following section, we describe WENO polynomial reconstruction for solving this problem.

2.2.2 Polynomial Reconstruction

In the reconstruction problem, we are given function averages a_j at cell centers x_j. The function averages are taken between cells such that

$$a_j = \int_{x_{j-1/2}}^{x_{j+1/2}} f(x)dx$$

We can define the reconstruction polynomial $L(x)$, commonly known as the primitive function, by imposing the conditions

$$a_j = \int_{x_{j-1/2}}^{x_{j+1/2}} L(x)dx \qquad j = 1, \ldots, N.$$

Taking $L(x) = \sum_{i=1}^{N} c_i x^i$, then we have the system $\mathbf{Ac} = \mathbf{b}$ where

$$\mathbf{A}_{ij} = \frac{x^j_{i+1/2} - x^j_{i-1/2}}{j}, \qquad \mathbf{b}_j = a_j. \tag{4}$$

The reconstruction polynomial $L(x)$ can then be evaluated at any point $x = x_0$.

2.2.3 WENO for Reconstruction

We seek an approximation to the function $q(x)$, given $2r-1$ cell averages $\{(x_j, Q_j)\}_{j=1}^{2r-1}$. Let $\{R_k^r(x)\}_{k=1}^{r}$ be the set of rth order reconstruction polynomials, where

$$Q_j = \int_{x_{j-1/2}}^{x_{j+1/2}} R_k^r(x)dx \qquad j = k, \ldots, k + r - 1.$$

Next, let $L(x)$ be the $(2r-1)$th order reconstruction polynomial which satisfies

$$Q_j = \int_{x_{j-1/2}}^{x_{j+1/2}} L(x)dx \qquad j = 1, \ldots, 2r - 1.$$

The reconstruction polynomial $L(x)$ can again be expressed as a linear combination of the polynomials $R_k^r(x)$ so that

$$L(x) = \sum_{k=0}^{r} C_k^r(x) R_k(x).$$

The WENO approximation to $q(x)$ is the convex combination of the polynomial $R_j(x)$ so that

$$q(x) \approx \sum_{k=1}^{r} w_k^r(x) R_k(x).$$

The weights $w_k^r(x)$ are determined from Eq. (2), and the smoothness measures S_j are chosen by replacing $P_j(x)$ with $R_j(x)$ in Eq. (1).

When solving conservation laws, we are solely interested in evaluating $q(x)$ at the cell midpoints $x = x_{j+1/2}$. We reconstruct $q(x)$ using the $2r - 1$ cell averages $Q_{j-r}, \ldots, Q_{j+r-1}$ and evaluating $q(x)$ at $x_{j+r+1/2}$. With these assumptions, the WENO formulas become

$$q_{j+1/2} \approx \sum_{k=1}^{r} w_k^r R_k(x_{j+r+1/2}) \qquad \text{where} \qquad R_k(x_{j+r+1/2}) = \sum_{l=1}^{r} a_{k,l}^r Q_{j-r+k+l-1}$$

and the coefficients $a_{k,l}^r$ are obtained from solving Eq. (4). We provide a table of coefficients C_k^r, and $a_{k,l}^r$ for $r = 2$, and $r = 3$ in Appendix B.

3 Implementation & Results

3.1 WENO Implementation in MATLAB

Our implementation uses WENO weights based on the smoothness measure (1) introduced by Jiang & Shu in [1] that improves WENO spatial accuracy for $r = 3$ from fourth order accurate to fifth order accurate.

Our program only considers one-dimensional scalar problems with periodic boundary conditions and therefore utilizes a standard time-stepping loop. The time-stepper calls the spatial discretization operator $L(\mathbf{Q})$ three or four times depending on whether we use Runge-Kutta 3 or Runge-Kutta 4. At each Q_i, the spatial discretization operator L calls the WENO reconstruction subroutine for the relevant cell boundary fluxes

$$L(Q_i) = \frac{1}{\Delta x}(\hat{f}_{i-1/2} - \hat{f}_{i+1/2}).$$

Although the code loops through all cell boundaries, constructing and storing the $\hat{f}_{j+1/2}$ to be referenced later, the performance suffers from reliance on subroutines and from calculating the smoothness measure (1) for each cell boundary (the C_k, ϵ, p parameters are all fixed, see Appendix B).

It is interesting to note that WENO has been shown to run faster than ENO on some vector machines in which logical statements (for selecting the smoothest stencil) are significantly more computationally expensive. Further runtime comparisons on supercomputers can be found in [1]. In addition, the method can be generalized to hyperbolic systems of equations with wave propagation methods [2].

3.1.1 A Note about Time-Stepping

Equation (3) is essentially an initial value problem across the cell averages Q_i that requires high accuracy reconstructions at the cell boundaries $q_{i\pm1/2}$. Therefore we require high-order time-steppers to preserve that spatial accuracy. To observe fifth-order spatial convergence, using the fourth order accurate Runge-Kutta time-stepping scheme requires taking

$$\Delta t \sim \Delta x^{5/4} \Rightarrow O(\Delta t^4) = O(\Delta x^5)$$

Likewise, with the third-order TVD or Total Variation Diminishing Runge-Kutta time-stepping scheme requires timestep

$$\Delta t \sim \Delta x^{5/3} \Rightarrow O(\Delta t^3) = O(\Delta x^5)$$

In practice the TVD Runge-Kutta-3 scheme is used due to its strong-stability preserving property [1] that introduces no oscillations in conjunction with a TVD spatial discretization. Although WENO-5 is not TVD, it is essentially non-oscillatory due to its use of a convex combination of quadratic reconstruction polynomials.

Consequently we expected oscillations with large timesteps under the non-TVD RK4 scheme, but found that not to be the case with the equations we tested. For example, with the Burgers equation we noticed that RK4 error was consistently lower than the TVD RK3 error as the timestep was increased:

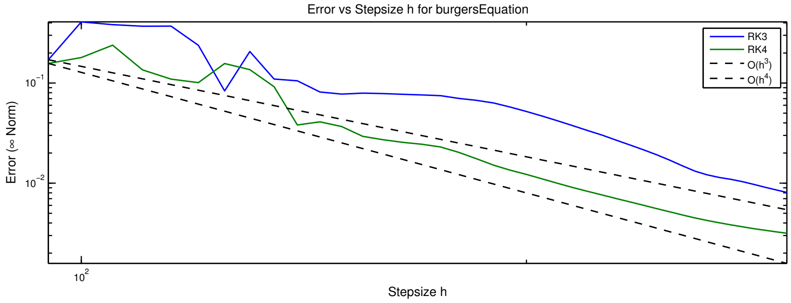


Figure 2: RK4 convergence is not affected at large timesteps

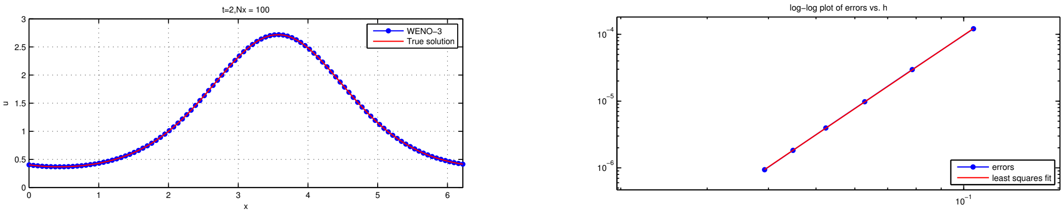
We conclude that although oscillations do not corrupt the RK4 behavior for the Burgers equation with smooth solution, the use of the RK4 scheme may introduce oscillations at shock discontinuities near which the desired order of convergence will not be observed.

3.2 Convergence

3.2.1 Smooth Solutions

In our WENO implementation, we observe the desired fifth-order spatial accuracy in the L^2 norm for the advection equation with the smooth solution $u(x,t) = e^{\sin(x-t)}$:

$$u_t + u_x = 0, \quad u(x,0) = e^{\sin(x)}$$



(a) Solution at $t = 2$

(b) WENO5-RK4 $E(h) = O(h^{4.95856})$

Figure 3: WENO5-RK4 for smooth solutions

WENO5-RK4 stands for fifth-order WENO ($r = 3$) and RK4 denotes fourth-order Runge-Kutta time integration. Note we refined with $\Delta t \approx (\Delta x)^{5/4}$ so that the time-stepping is essentially fifth-order [1]. We calculate the least squares fit through the error on a log scale to observe the expected accuracy for both our WENO implementation and `clawpack`'s second-order accurate Lax-Wendroff scheme. Note that we observe fifth-order accuracy in as few as 75 grid points.

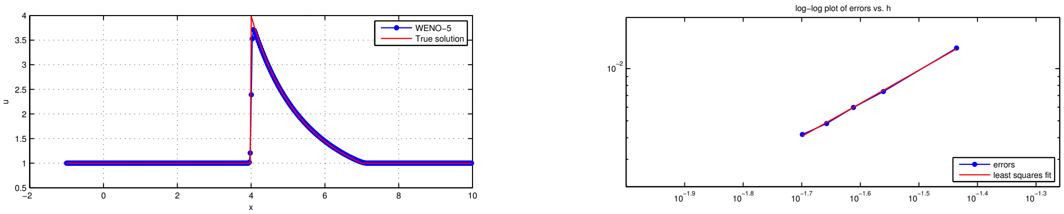
We briefly restate a result from [1]: in general, sufficiently smooth solutions using a WENO method of r candidate polynomials and a nth-order Runge-Kutta time integrator such that $n \geq \max(r, 3)$ will have accuracy $O(\Delta x^r)$. However, in the cases $r = 2, 3$ we do better, or specifically, $O(\Delta x^{2r-1})$ with the new smoothness measure 1.

3.2.2 Discontinuous Solutions

We now consider the following equation

$$u_t + (\sqrt{u})_x = 0, \quad u(x,0) = \begin{cases} 4 & \text{if } 0 < x < 1 \\ 1 & \text{otherwise} \end{cases}$$

For discontinuities we can only observe convergence in the typical sense in the continuous regions of the solution, as demonstrated below. The error from the WENO reconstruction at the discontinuity contaminates the fifth-order accuracy we would expect in the continuous regions to only second order accuracy in the following scenario. Note that in the region $6.5 < x < 7.5$ the true solution is continuous but not smooth.



(a) WENO5-RK5 approximation

(b) WENO5-RK4 error $E(h) = O(h^{1.90852})$

Figure 4: Error: WENO5-RK4 for shock

We observe that ordinary WENO-5 reconstruction requires an extremely fine grid to even obtain second-order accuracy away from the shock. In the following section we experiment with sharpening methods that resolve discontinuities by incorporating an augmented smoothness measure.

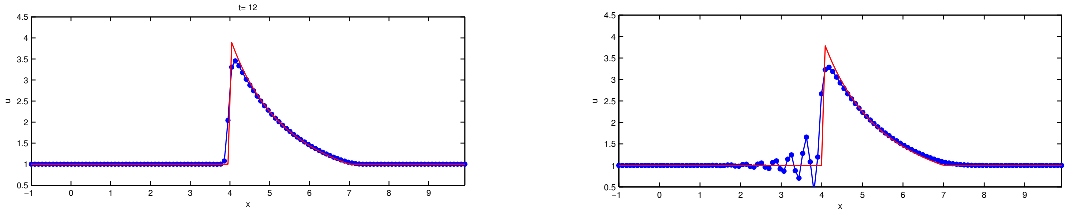
3.3 Sharpening Shocks

To obtain better resolution of contact discontinuities we follow the suggestion in [1] and implement Yang's artificial compression method [6]. The flux for the advection equation $\hat{f}_{j+1/2}$ is replaced by a "sharpened" flux $\hat{f}^A_{j+1/2} = \hat{f}_{j+1/2} + c_{j+1/2}$ such that

$$c_{j+1/2} = \text{minmod}\left[\frac{\alpha_j}{2}\,\text{minmod}(\hat{f}^R_{j+1/2} - \hat{f}_{j+1/2}, \hat{f}^R_{j-1/2} - \hat{f}_{j-1/2}), f_{j+1} - \hat{f}_{j+1/2}, \hat{f}^R_{j-1/2} - f_{j-1}\right]$$

$$\alpha_j = 33\left(\frac{|f_{j+1} - 2f_j + f_{j-1}|}{|f_{j+1} - f_j| + |f_j - f_{j-1}|}\right)^2$$

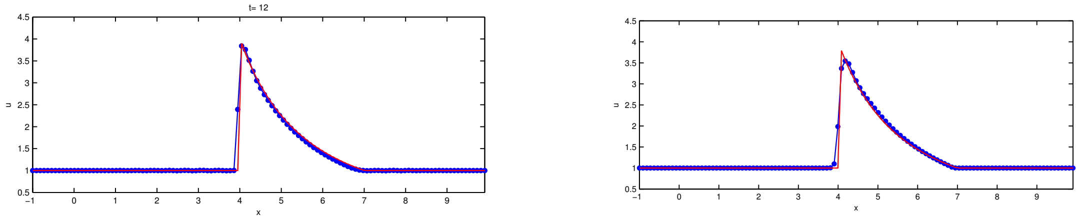
We had previously considered a left-leaning stencil of cell averages (three cell averages to the left, two to the right) for the advection speed $a > 0$. The $f^R_{j\pm 1/2}$ terms represent a right-leaning stencil (pretending the advection speed $a < 0$. The use of minmod has the effect of choosing the smoothest slope between the various test differences from the right and left to choose the correct quantity $c_{j+1/2}$ to add to the old flux. The new approximation has the effect of "drifting" the flux towards its smoothest nearby region from the previous timestep, thereby *sharpening* the solution near discontinuities. This method has the additional benefit of retaining the original WENO accuracy because the magnitudes of the added terms are of the same order of the truncation error.



(a) Standard WENO5-RK4

(b) CLAWPACK's Lax-Wendroff, no limiter

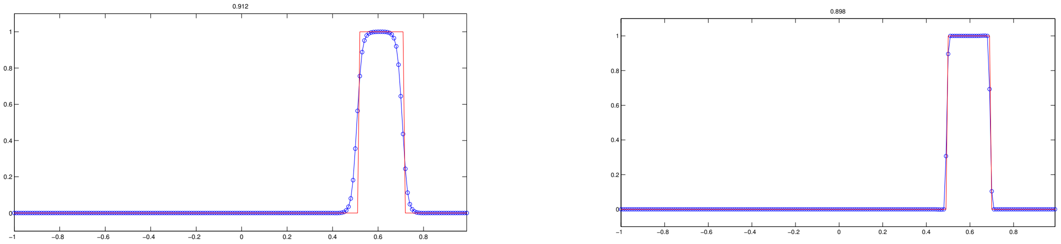
Figure 5: Standard WENO5 without sharpening produces a satisfactory result while Lax-Wendroff without limiters introduces spurious oscillations at the discontinuity. Both were run with 120 grid points and 150 timesteps to time $t = 12$.



(a) WENO5-RK4-A (with sharpening)

(b) CLAWPACK's Lax-Wendroff, Superbee limiter

Figure 6: With the same spatial and time resolution as Fig. 5 WENO5 sharpened gives better resolution of the shock discontinuity while preserving accuracy over the smooth rarefaction region of the solution, while Lax-Wendroff with the Superbee limiter fails to capture the sharp peak.



(a) WENO5-RK4

(b) WENO5-RK4-A sharpened

Figure 7: Another example of a sharpened traveling wave

3.4 WENO Weight Parameters ϵ, p

Recall the formulas for the WENO weights w_k

$$w_k = \frac{\alpha_k}{\sum_{j=1}^{r} \alpha_j}$$

where

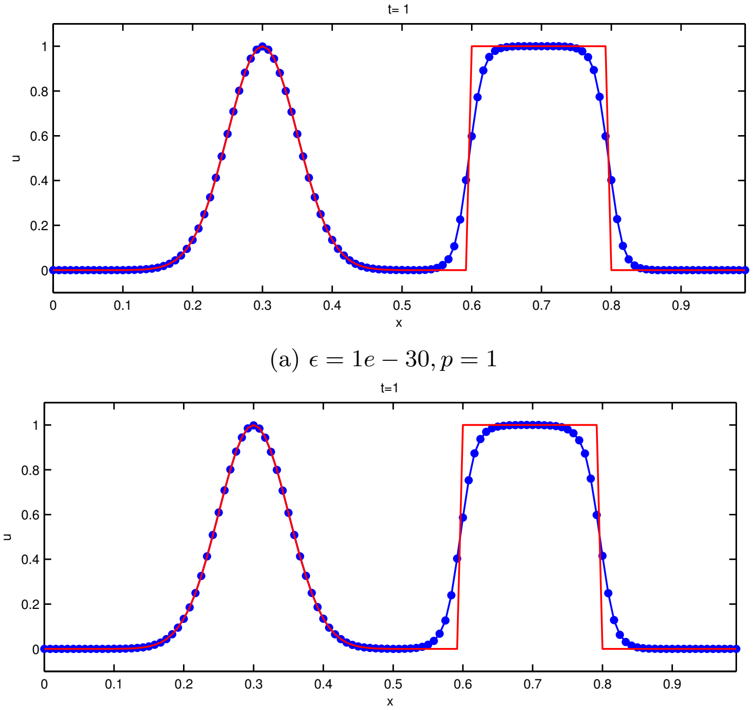
$$\alpha_k = \frac{C_k}{(\epsilon + S_k(x))^p}.$$

Jiang et. al. [1] use $\epsilon = 1e - 5, p = 2$ for their numerical tests, but we discovered that sharpness at discontinuities is sensitive to different values of these parameters. The suggestion to explore smaller values of ϵ such as $\epsilon = 1e-30$ had no clearly visible effect. After taking $\epsilon = 1e-30$ we varied the value of the exponent p in the denominator which essentially exaggerates the desired effect:

$$\alpha_k = \frac{C_k}{(\epsilon + S_k(x))^p} \approx \frac{C_k}{S_K^p(x)}$$

However the coefficients are more subject to numerical instability since we essentially divide by very small numbers when the solution is nearly constant and stencils containing smooth constant data are weighed much more heavily. This (increasing the value of p and decreasing ϵ) has the slightly visible effect of “sharpening” the solution to drift faster to the smooth areas of the solution below.

$$\begin{aligned} u_t + u_x &= 0 \\ u(x,0) &= \begin{cases} e^{-200(x-0.3)^2} & \text{if } |x-0.7| > 1 \\ 1 & \text{otherwise} \end{cases}. \end{aligned}$$



(a) $\epsilon = 1e-30, p = 1$

(b) $\epsilon = 1e-30, p = 3$

Figure 8: Higher values of p push the WENO weights closer to the optimal weights designed for smooth solution and therefore perform relatively poorly around discontinuities. Note that the gaussian’s resolution is unaffected because the WENO weights are designed to replicate the optimal weights in smooth regions.

4 Conclusion

High-resolution WENO methods outperform standard ENO reconstruction particularly for equations with smooth solutions with rich structure. They can be easily modified with sharpening to yield good resolution at jump discontinuities without losing their high spatial accuracy in smooth regions of the solution, a feature that makes WENO methods more attractive than standard second-order accurate wave-propagation solvers in practice.

A Interpolation Weights

We consider the interval $[1, 2r-1]$ with interpolation data $\{(x_j, f_j)\}_{j=1}^{2r-1}$ where $x_j = j$. We seek the the weights C_k^r such that

$$L(x) = \sum_{i=1}^{r} C_k^r(x) P_k^r(x)$$

where $L(x)$ is the $2r-1$th order Polynomial passing through all points $\{(x_j, f_j)\}_{j=1}^{2r-1}$ and $P_r^k(x)$ denotes the rth order polynomial passing through points $\{(x_j, f_j\}_k^{k+r-1}$. To solve for $C_k^r(x)$, we assume

$$C_k^r(x) = \sum_{i=1}^{r} \alpha_{k,i} x^{i-1}$$

which leads to the r^2 unknowns α_k, i. To obtain the α_k, i, we match the coefficients $[f_j \cdot x^k]$ for $j, k = 1, \ldots r$. For certain values of x, the weights $C_k^r(x)$ form a convex combination, i.e. $\sum_{i=1}^{r} C_k^r(x) = 1$ and $C_k^r(x) > 0$. For example, if we take $r = 2$, we find that

$$C_1^2(x) = 1 - \tfrac{x}{2} \qquad P_1^2(x) = f_2 + (-f_2 + f_3)(-1 + x)$$

$$C_2^2(x) = \tfrac{x}{2} \qquad P_2^2(x) = f_1 - (f_1 + f_2)x$$

$$L(x) = f_1 + (-f_1 + f_2 + 1/2(f_1 - 2f_2 + f_3)(-1 + x))x$$

The smoothness measures are given by

$$S_1(x) = 4f_1^2 - 6f_1f_2 + 2f_2^2 - 2(-f_1 + f_2)^2 - 2f_1^2x + 4f_1f_2x - 2f_2^2x$$
$$S_2(x) = 4f_2^2 - 6f_2f_3 + 2f_3^2 - 2(-f_2 + f_3)^2 - 2f_2^2x + 4f_2f_3x - 2f_3^2x$$

B Reconstruction Weights

The polynomial coefficients for a general WENO method that uses r candidate polynomials P_k to approximate the flux, where

$$P_k^r(f_0, \dots, f_{r-1}) = \sum_{l=0}^{r-1} a_{k,l}^r f_l,$$

are given in the following table

Table 1: Coefficients $a_{k,l}^r$

r	k	$l = 0$	$l = 1$	$l = 2$
2	0	-1/2	3/2	
	1	1/2	1/2	
3	0	1/3	-7/6	11/6
	1	-1/6	5/6	1/3
	2	1/3	5/6	-1/6

These polynomial coefficients were derived by solving a linear system for the coefficients a_i of $P(x) = a_0 + a_1 x + a_2 x^2$. For example, to obtain coefficients for the leftmost stencil $k = 0$ when $r = 3$, we solve

$$\int_{x_{j-5/2}}^{x_{j-3/2}} P(x)dx = u_{j-2}$$
$$\int_{x_{j-3/2}}^{x_{j-1/2}} P(x)dx = u_{j-1}$$
$$\int_{x_{j-1/2}}^{x_{j+1/2}} P(x)dx = u_j,$$

thus obtaining the coefficients a_i in terms of u_j, the values of u at the grid points. Evaluating $P(x)$ at $x_{j+1/2}$ yields

$$P(x_{j+1/2}) = \frac{1}{3}u_{j-2} - \frac{7}{6}u_{j-1} + \frac{11}{6}u_j,$$

which are exactly the coefficients given in row 4 of Table 1.

Table 2: Weights C_k^r

C_r^k	$k = 0$	$l = 1$	$l = 2$
$r = 2$	1/2	2/3	
$r = 3$	1/10	6/10	3/10

References

[1] Jiang, G.-S., and Shu, C.-W. Efficient implementation of weighted eno schemes. Tech. rep., DTIC Document, 1995.

[2] Ketcheson, D. I., Parsani, M., and LeVeque, R. J. High-order wave propagation algorithms for hyperbolic systems. *SIAM Journal on Scientific Computing 35*, 1 (2013), A351–A377.

[3] LeVeque, R. J. *Finite volume methods for hyperbolic problems*, vol. 31. Cambridge University Press, 2002.

[4] Shi, J., Hu, C., and Shu, C.-W. A technique of treating negative weights in weno schemes. *Journal of Computational Physics 175*, 1 (2002), 108–127.

[5] Shu, C.-W. High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM review 51*, 1 (2009), 82–126.

[6] Yang, H. An artificial compression method for eno schemes: the slope modification method. *Journal of Computational Physics 89*, 1 (1990), 125–160.