The Effect of the Sensitivity Parameter in Weighted Essentially Non-oscillatory Methods

Bo Dong, Sigal Gottlieb, Yulia Hristova, Yan Jiang and Haijin Wang

Abstract Weighted essentially non-oscillatory methods (WENO) were developed to capture shocks in the solution of hyperbolic conservation laws while maintaining stability and without smearing the shock profile. WENO methods accomplish this by assigning weights to a number of candidate stencils, according to the smoothness of the solution on the stencil. These weights favor smoother stencils when there is a significant difference while combining all the stencils to attain higher order when the stencils are all smooth. When WENO methods were initially introduced, a small parameter ε was defined to avoid division by zero. Over time, it has become apparent that ε plays the role of the sensitivity parameter in stencil selection. WENO methods allow some oscillations, and it is well known that these oscillations depend on the size of ε. In this work, we show that the value of ε must be below a certain critical threshold ε_c and that this threshold depends on the function used and on the size of the jump discontinuity captured. Next, we analytically and numerically show the size of the oscillations for one time-step and over long time integration when $\varepsilon < \varepsilon_c$

B. Dong · S. Gottlieb (✉)
Department of Mathematics, University of Massachusetts Dartmouth,
Dartmouth, MA, USA
e-mail: sgottlieb@umassd.edu

B. Dong
e-mail: bdong@umassd.edu

Y. Hristova
Department of Mathematics and Statistics, University of Michigan-Dearborn,
Dearborn, MI, USA
e-mail: yuliagh@umich.edu

Y. Jiang
Department of Mathematics, Michigan State University, 619 Red Cedar Rd.,
East Lansing 48824, MI, USA
e-mail: jiangyan@math.msu.edu

H. Wang
College of Science, Nanjing University of Posts and Telecommunications, Nanjing 210023,
Jiangsu Province, People's Republic of China
e-mail: hjwang@njupt.edu.cn

S.C. Brenner (ed.), *Topics in Numerical Partial Differential Equations and Scientific Computing*, The IMA Volumes in Mathematics and its Applications 160, DOI 10.1007/978-1-4939-6399-7_2

and their dependence on the size of ε, the function used, and the size of the jump discontinuity.

1 Introduction: Weighted Essentially Non-oscillatory Methods

When approximating the solution to a conservation law of the form

$$u_t + f(u)_x = 0,$$

we use a conservative finite difference scheme

$$u_t = -\frac{1}{\Delta x}(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}})$$

to obtain a physically relevant solution [9]. The term $\hat{f}_{j+\frac{1}{2}} = \hat{f}(u_{j-k}, \ldots, u_{j+l})$ is the *numerical flux*, and the points $x_{j-k}, \ldots, x_{j+l}$ constitute the *stencil*. To be a reasonable approximation, the numerical flux must be (at least) Lipschitz continuous and consistent with the physical flux f, *i.e.*, $\hat{f}(u, \ldots, u) = f(u)$. Once the spatial derivative is computed by differencing the numerical fluxes, we obtain a system of ODEs which is then evolved to the next time-step using some standard time-stepping method. Different numerical fluxes give rise to different numerical methods. Any differences between the properties of such methods are a result of differences in the numerical flux.

A major issue with the use of finite difference methods for computations with shocks is that oscillations arise when we take points on opposite sides of the shock to evaluate the derivative at a given point. These oscillations at the shock location propagate to the smooth regions, destroying the stability of the solution. To avoid oscillations and instabilities that arise from using a finite-difference stencil that takes information from both sides of the shock, ENO ([3], [4], [12]) schemes search for the locally smoothest stencil and use that stencil to calculate the numerical fluxes. The idea behind ENO schemes is stencil switching in order to eliminate oscillations. The ENO scheme evaluates the smoothness of several stencils near the point of interest and picks the smoothest stencil for evaluating the derivative at that point. This means that the method should avoid picking stencils that cross the shock, so that the stencil is chosen only from a smooth region (to the left or right of the shock) in which linear stability is enough to ensure nonlinear stability.

Liu, Osher, and Chan [10] improved upon the ENO method by assigning each stencil a weight which depends on its smoothness and summing the approximations from all the candidate stencils, each with its corresponding weight. The weights are chosen so that in smooth regions, we obtain higher-order accuracy, whereas near discontinuities, the method imitates the ENO scheme by assigning near-zero weights to the stencils that contain discontinuities. This approach is called the weighted ENO (or WENO) method. An rth order ENO scheme considers a total of $2r - 1$ points to

evaluate the flux. The WENO scheme uses all the candidate stencils and therefore $2r-1$ points, so that a clever choice of weights [7] results in a WENO scheme which is of order $2r-1$ in smooth regions [13].

In the following subsections, we describe three variants of the WENO procedure. We focus on the choice $r=3$ which gives fifth-order methods for smooth problems and third order in non-smooth regions. The three methods we consider are the WENO method described by Jiang and Shu in [7], the mapped WENO procedure given in [5], and the improved method presented in [1]. We refer to these methods as WENO-JS, WENO-M, and WENO-Z, respectively. In all of the following, we assume that we have a flux $f(u)$ such that $\frac{df}{du} \geq 0$. If this is not the case, then we split the flux into the positive and negative parts

$$f(u) = f^+(u) + f^-(u),$$

such that $\frac{df^+}{du} \geq 0$ and $\frac{df^-}{du} \leq 0$. Then, we handle each part separately, using differently biased stencils for the negative flux. We will not describe this in detail here, but the interested reader can consult [13].

1.1 WENO-JS

In this section, we present the $r=3$ method of Jiang and Shu [7]. To calculate the numerical flux $\hat{f}_{j+\frac{1}{2}}$ at any given point x_j, we begin by calculating the smoothness measurements to determine whether a shock lies within the stencil. For the fifth-order scheme, these are as follows:

$$\begin{aligned}
IS_0 &= \frac{13}{12}\left(f_{j-2} - 2f_{j-1} + f_j\right)^2 + \frac{1}{4}\left(f_{j-2} - 4f_{j-1} + 3f_j\right)^2 \\
IS_1 &= \frac{13}{12}\left(f_{j-1} - 2f_j + f_{j+1}\right)^2 + \frac{1}{4}\left(f_{j-1} - f_{j+1}\right)^2 \\
IS_2 &= \frac{13}{12}\left(f_j - 2f_{j+1} + f_{j+2}\right)^2 + \frac{1}{4}\left(3f_j - 4f_{j+1} + f_{j+2}\right)^2
\end{aligned}$$

(Note that the factor of $\frac{1}{12}$ can be ignored due to the normalization later). Next, we use the smoothness measurements to calculate the stencil weights

$$\alpha_0^{(JS)} = \omega_0 \left(\frac{1}{\varepsilon + IS_0}\right)^p \quad \alpha_1^{(JS)} = \omega_1 \left(\frac{1}{\varepsilon + IS_1}\right)^p \quad \alpha_2^{(JS)} = \omega_2 \left(\frac{1}{\varepsilon + IS_2}\right)^p$$

where $\omega_0 = \frac{1}{10}$, $\omega_1 = \frac{6}{10}$, and $\omega_2 = \frac{3}{10}$ are the centered difference weights, and p is the power of the weights, typically chosen to be $p=2$.

Note that the parameter ε is added to the denominator to prevent division by zero. To avoid division by machine zero, we must pick ε larger than the square root of the smallest positive number seen as nonzero by the computer. For single precision, this is $\varepsilon > 10^{-18}$, while for double precision, we need $\varepsilon > 10^{-153}$.

These weights are then normalized

$$\omega_0^{(JS)} = \frac{\alpha_0^{(JS)}}{\sum_{i=1}^{3} \alpha_i^{(JS)}} \qquad \omega_1^{(JS)} = \frac{\alpha_1^{(JS)}}{\sum_{i=1}^{3} \alpha_i^{(JS)}} \qquad \omega_2^{(JS)} = \frac{\alpha_2^{(JS)}}{\sum_{i=1}^{3} \alpha_i^{(JS)}},$$

and the normalized weights are used to compute the numerical fluxes

$$\hat{f}_{j+\frac{1}{2}} = \omega_0^{(JS)} \left(\frac{2}{6} f_{j-2} - \frac{7}{6} f_{j-1} + \frac{11}{6} f_j \right) + \omega_1^{(JS)} \left(-\frac{1}{6} f_{j-1} + \frac{5}{6} f_j + \frac{2}{6} f_{j+1} \right) + \omega_2^{(JS)} \left(\frac{2}{6} f_j + \frac{5}{6} f_{j+1} - \frac{1}{6} f_{j+2} \right).$$

Finally, the derivative is computed by taking a difference of the fluxes

$$f(u)_x \approx \frac{1}{\Delta x} \left(\hat{f}^+_{j+\frac{1}{2}} - \hat{f}^+_{j-\frac{1}{2}} \right).$$

This process leads to a system of ordinary differential equations, which can then be evolved by standard time-stepping methods such as Runge–Kutta schemes.

1.2 Mapped WENO (WENO-M)

The mapped WENO method [5] was developed to overcome the loss of order of convergence near critical points of f that is experienced by WENO-JS. Rather than creating a new smoothness measure, the mapped WENO algorithm uses the ideal weights ω_k and the WENO-JS weights $\omega_k^{(JS)}$ to create new mapped weights, $\omega_k^{(M)}$ given by

$$\omega_k^{(M)} = \frac{\alpha_k^{(M)}}{\sum_{i=0}^{2} \alpha_i^{(M)}}$$

where

$$\alpha_k^{(M)} = \frac{\omega_k^{(JS)} \left(\omega_k + \omega_k^2 - 3\omega_k \omega_k^{(JS)} + (\omega_k^{(JS)})^2 \right)}{\omega_k^2 + \omega_k^{(JS)} (1 - 2\omega_k)}.$$

These weights are then used for the computation of the numerical fluxes

$$\hat{f}_{j+\frac{1}{2}} = \omega_0^{(M)} \left(\frac{2}{6} f_{j-2} - \frac{7}{6} f_{j-1} + \frac{11}{6} f_j \right) + \omega_1^{(M)} \left(-\frac{1}{6} f_{j-1} + \frac{5}{6} f_j + \frac{2}{6} f_{j+1} \right) + \omega_2^{(M)} \left(\frac{2}{6} f_j + \frac{5}{6} f_{j+1} - \frac{1}{6} f_{j+2} \right).$$

The convergence of this WENO-M method near critical points was studied in [5] with the value $\varepsilon = 10^{-40}$ and verified that this method converges at design order even near critical points.

1.3 WENO-Z

The mapped WENO approach above is much slower than the original WENO-JS. An improvement of both these methods was introduced in [1] where the smoothness measurements are modified. This method defines a value $\tau_5 = |IS_0 - IS_2|$ and computes the new weights

$$\alpha_k^{(z)} = \omega_k \left(1 + \left(\frac{\tau_5}{IS_k + \varepsilon}\right)^p\right).$$

These weights are then normalized

$$\omega_0^{(z)} = \frac{\alpha_0^{(z)}}{\sum_{i=1}^3 \alpha_i^{(z)}} \qquad \omega_1^{(z)} = \frac{\alpha_1^{(z)}}{\sum_{i=1}^3 \alpha_i^{(z)}} \qquad \omega_2^{(z)} = \frac{\alpha_2^{(z)}}{\sum_{i=1}^3 \alpha_i^{(z)}},$$

and the normalized weights are used to compute the numerical fluxes

$$\begin{aligned}\hat{f}_{j+\frac{1}{2}} = {} & \omega_0^{(z)} \left(\frac{2}{6} f_{j-2} - \frac{7}{6} f_{j-1} + \frac{11}{6} f_j\right) + \omega_1^{(z)} \left(-\frac{1}{6} f_{j-1} + \frac{5}{6} f_j + \frac{2}{6} f_{j+1}\right) \\ & + \omega_2^{(z)} \left(\frac{2}{6} f_j + \frac{5}{6} f_{j+1} - \frac{1}{6} f_{j+2}\right).\end{aligned}$$

The fifth-order WENO-Z scheme was shown to have less dissipation and higher resolution than the WENO-JS scheme, generates solutions that are as sharp as those in WENO-M, and does not suffer as much as WENO-JS from reduced convergence rates at critical points. In addition, the WENO-Z method has a computational cost about the same as WENO-JS and significantly smaller than WENO-M.

In [1], the authors consistently use a small value of ε to avoid this parameter dominating over the smoothness measurements. Later, in [6], the authors consider the appropriate values of ε and suggest the value $\varepsilon = \Delta x^{r-1}$ as a compromise that avoids the loss of stencil sensitivity when ε dominates over the smoothness measurement, while still serving its original role of preventing division by zero.

2 The Importance of Choosing a Small Enough Sensitivity Parameter

When the parameter ε was initially introduced, its sole function was to prevent division by zero. The parameter value chosen initially was $\varepsilon = 10^{-6}$ [7]. In fact, the parameter ε serves a role as a sensitivity parameter that determines the difference in size between smoothness measurements that induce stencil switching. If the smoothness indicators are all below the level of ε, all the candidate stencils are seen as equally smooth and the centered difference method is attained. However, if one of the smoothness indicators is larger than ε, then the weights will no longer be equal and the chosen stencils will be biased away from the shock.

Over time, users of the WENO method have observed that the value of ε had important implications on the presence of oscillations. In studying the order of convergence of WENO-JS, Henrick et al. [5] noted that the parameter ε plays a critical though unintended role. They noted that the expansion of the nonnormalized weights yields $\alpha_k = \frac{\omega_k}{\varepsilon + \Delta x^2 f'^2 + O(\Delta x^4)}$ so that as $\Delta x \to 0$, the parameter ε will eventually dominate over the smoothness indicators, and the WENO-JS method approaches the behavior of the central difference scheme. At the same time, they also note that oscillations of order ε^2 can be seen near discontinuities (an observation that we will study rigorously later in this paper). Both these observations suggest that smaller values of ε are preferable. The first question we ask is what is that value of ε_c, the critical value that ε must be below so that its effect is easily controlled. In this section, we perform a few numerical studies that show the dependence of ε_c on the power of the function and the size of the jump.

To understand what is happening, we start with a simple example: consider the function $f(u) = cu^m$ where

$$u = \begin{cases} 0 & \text{if } n \le j \\ M & \text{if } n > j \end{cases}$$

In this case, m is the order of the function, and M is the size of the jump. The smoothness measurements for this function are mostly zero, except near the shock. At that point, we get

$$\begin{aligned} IS_0 &= \frac{13}{12}\left(f_{j-2} - 2f_{j-1} + f_j\right)^2 + \frac{1}{4}\left(f_{j-2} - 4f_{j-1} + 3f_j\right)^2 = 0 \\ IS_1 &= \frac{13}{12}\left(f_{j-1} - 2f_j + f_{j+1}\right)^2 + \frac{1}{4}\left(f_{j-1} - f_{j+1}\right)^2 = \frac{4}{3}c^2(M)^{2m} \\ IS_2 &= \frac{13}{12}\left(f_j - 2f_{j+1} + f_{j+2}\right)^2 + \frac{1}{4}\left(3f_j - 4f_{j+1} + f_{j+2}\right)^2 = \frac{10}{3}c^2(M)^{2m}. \end{aligned}$$

The weights then become (with the choice of $p = 2$)

$$\alpha_0 = \frac{1}{10}\left(\frac{1}{\varepsilon}\right)^2, \qquad \alpha_1 = \frac{6}{10}\left(\frac{1}{\varepsilon + \frac{4}{3}c^2(M)^{2m}}\right)^2, \qquad \alpha_2 = \frac{3}{10}\left(\frac{1}{\varepsilon + \frac{10}{3}c^2(M)^{2m}}\right)^2.$$

Notice that if $\varepsilon << c^2M^{2m}$, the smooth stencil will get the large weights and the other stencils get very small weights. However, if $\varepsilon \geq \varepsilon_c \approx c^2M^{2m}$, the wrong stencils will be chosen, producing oscillations. The purpose of this section is to explore this critical value of ε and its effects in a variety of numerical settings.

In the following examples, we demonstrate the effect of choosing a value of ε that is too large and examine the values of ε that are small enough to prevent unwanted behavior of the method. First, we examine the size of the overshoot/undershoot in a simple linear advection equation.

Example 1: Consider linear advection equation

$$u_t + u_x = 0 \quad \text{with} \quad u(x,0) = \begin{cases} -1 \text{ if } x < 0 \\ 1 \quad \text{otherwise} \end{cases}$$

We discretize the spatial grid with $N = 100$ points between $[-1, 1]$. We use WENO-JS to evaluate the derivatives and use the third-order SSP Runge–Kutta method with $\Delta t = \frac{1}{2}\Delta x$ to advect the solution over fifty time-steps. If $|u| > 1$, we have an undershoot or overshoot. The table below lists the maximal over/undershoot for the first 50 time-steps for each value of ε. We observe that for this example, we require $\varepsilon \leq 10^{-24}$ for the under- or overshoot to be within roundoff error and even smaller ε for the under- or overshoot to be within machine precision.

ε	10^{-6}	10^{-9}	10^{-13}	10^{-16}	10^{-19}	10^{-24}	10^{-29}	10^{-39}
$max\|u\| - 1$	5.66e-5	1.62e-6	1.42e-8	4.16e-10	1.18e-11	3.26e-14	2.22e-16	2.22e-16

This simple demonstration indicates that the choice of ε strongly affects the size of the oscillations. There is a critical value ε_c below which the oscillations are no longer significant. In the next section, we explore the dependence of this critical value of ε on the size of the jump.

2.1 Dependence of the Critical Value of ε on the Size of the Jump Discontinuity

In this section, we study numerically the behavior of several problems, with several jump sizes, using the WENO-JS method.

Example 1a We revisit the linear advection equation in **Example 1**, this time with step function initial conditions of size M

$$u(x,0) = \begin{cases} M \text{ if } -1/8 < x < 1/8 \\ 0 \quad \text{otherwise} \end{cases}$$

If we use a relatively large $\varepsilon = M = 10^{-2}$, we get very bad looking results after 50 time-steps, as seen in Figure 1.

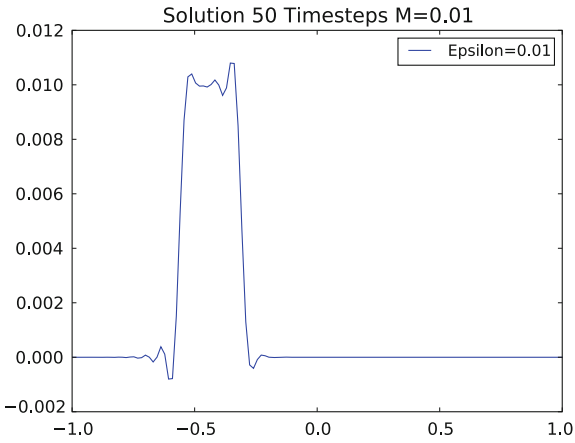

Fig. 1 Linear advection with $\varepsilon = M = 10^{-2}$, WENO-JS.

But even if the results look nice, ε can have a bad effect on the total variation (TV) of the numerical solution as seen in Figure 2 on the left, with $M = 1$. Another observation is that the smaller M is, the smaller we require ε to avoid a significant TV increase (Figure 2 with $M = 10^{-1}$ and $M = 10^{-4}$). Although the TV of this problem increases generally, whatever the value of ε, an additional rise that is attributable to the value of ε is concerning. In these simulations, we observe that the critical values of ε needed to prevent a spurious rise in total variation are $\varepsilon_c \approx 10^{-8}M^2$.

Example 2 We consider Burgers' equation

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

with initial condition

$$u(x, 0) = M \sin(\pi x).$$

This problem develops a stationary shock. We discretize this grid with 100 points in space and set $\Delta t = \frac{CFL}{M}\Delta x$ with $CFL = 0.5$. Choosing $M = 0.01$ and $\varepsilon = 10^{-5}$, we step this forward for 50 time-steps. The results show an oscillatory profile, as seen in Figure 3. If ε is chosen small enough, there is no rise in TV. The values of ε_c for each M, required for TV norm to settle, are seen in the table below.

M	10^{-1}	10^{-3}	10^{-5}	10^{-6}	$10^{-6.5}$	10^{-7}	10^{-9}
ε_c	10^{-5}	10^{-13}	10^{-21}	10^{-25}	10^{-27}	10^{-29}	10^{-37}

$\longrightarrow \quad \varepsilon_c \approx 10^{-1}M^4.$

Example 3 To see whether the pattern of dependence on m continues, we test the problem

$$u_t + \left(\frac{1}{3}u^3\right)_x = 0 \quad \text{with} \quad u(x, 0) = \begin{cases} M \text{ if } -1/8 < x < 1/8 \\ 0 \quad \text{otherwise} \end{cases}$$

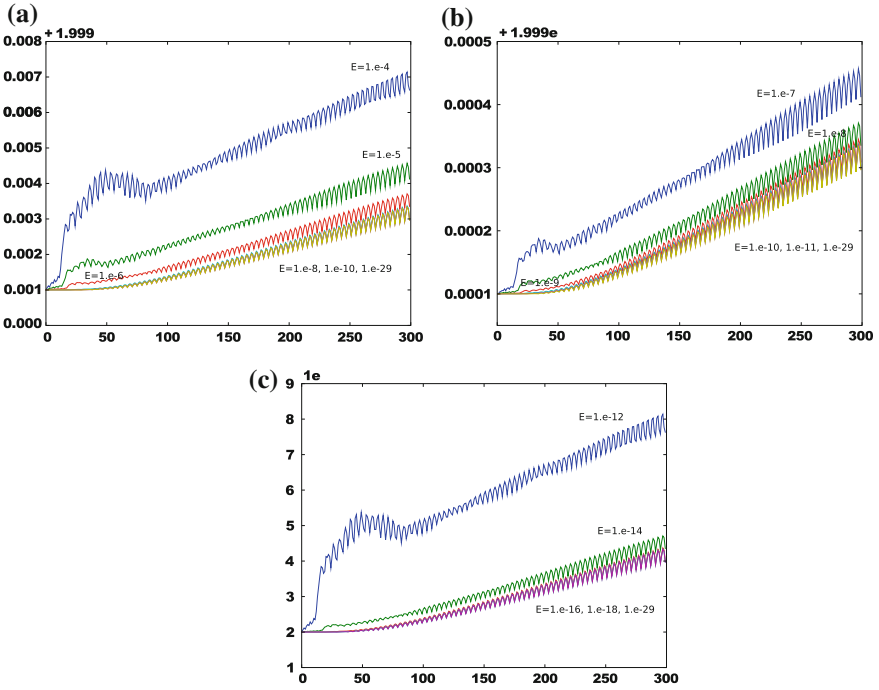

Fig. 2 Linear advection with different values of M. The smaller the M is, the smaller ε needs to be to avoid a significant TV increase. (a) $M = 1$. (b) $M = 10^{-1}$. (c) $M = 10^{-4}$.

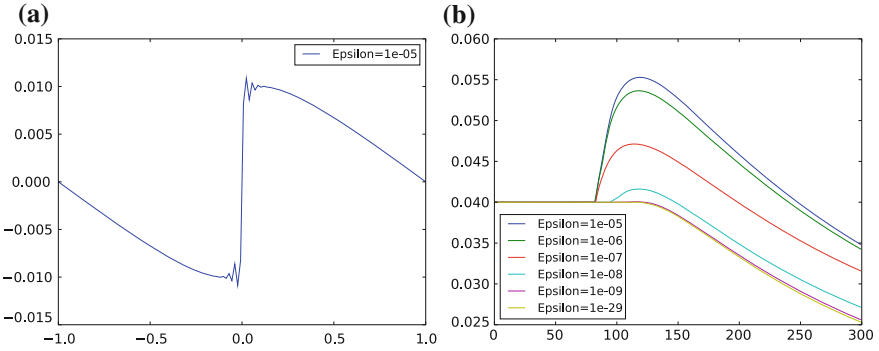

Fig. 3 Burgers' example with $M = 10^{-2}$. Left: numerical solution for $\varepsilon = 10^{-5}$, after 50 time-steps. Right: TV of the solution over 300 time-steps, for different values of ε. (a) Solution Profile with $\varepsilon = 10^{-5}$. (b) Total Variation.

The relationship between the value of M and the ε required to avoid a TV "bump" is shown in the table below.

M	10^{-0}	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
ε_c	10^{-3}	10^{-9}	10^{-15}	10^{-21}	10^{-27}	10^{-33}	10^{-39}

$$\longrightarrow \quad \varepsilon_c \approx 10^{-3} M^6.$$

Example 4 Consider the Euler system

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \begin{pmatrix} \rho u \\ P + \rho u^2 \\ u(P+E) \end{pmatrix} = 0$$

on $0 \le x \le 9$, and $t \ge 0$. Here, $\rho(x,t)$ is the density, $\rho u(x,t)$ the momentum, $E(x,t)$ the energy, $P(x,t)$ is the pressure, and $c(x,t)$ is the soundspeed. They are related by $P(x,t) = (\gamma - 1)\left(E - \frac{1}{2}\rho v^2\right)$, and $c(x,t) = \sqrt{\frac{\gamma P}{\rho}}$.

We impose boundary conditions

$$\rho_L = M \quad u_L = 0 \quad P_L = M * (\gamma - 1) * 0.1$$

$$\rho_R = 0.001M \quad u_R = 0 \quad P_R = M * (\gamma - 1) * 10^{-7}$$

and step function initial conditions

$$\rho(x,t) = \begin{cases} \rho_L & \text{if } x \le 3 \\ \rho_R & \text{if } x > 3 \end{cases} \quad u(x,t) = \begin{cases} u_L & \text{if } x \le 3 \\ u_R & \text{if } x > 3 \end{cases} \quad P(x,t) = \begin{cases} P_L & \text{if } x \le 3 \\ P_R & \text{if } x > 3. \end{cases}$$

For the Euler system, it is important that the pressure or density does not become negative, even at intermediate stages. The value of ε has an effect on this, as well. The table below shows the values of ε_c such that when $\varepsilon > \varepsilon_c$, we get negative pressure or density and the code stops working.

M	1	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
ε_c	10^{-5}	10^{-9}	10^{-13}	10^{-17}	10^{-21}	10^{-25}	10^{-29}

$$\longrightarrow \quad \varepsilon_c \approx 10^{-5} M^2.$$

This section provided numerical evidence that we must choose $\varepsilon \le \varepsilon_c \approx M^{2m}$ to avoid a variety of problems is $\varepsilon_c \approx M^{2m}$. The important observation here is that the size of ε varies with the size of the jump. In the next section, we consider the behavior of the method when $\varepsilon << \varepsilon_c$ and observe the size of the oscillations that occur in that range and their dependence on ε.

3 The Effect of the Sensitivity Parameter for One Time-step

In the previous section, we showed that ε functions as a stencil sensitivity parameter and that the value of ε needs to be chosen appropriately to avoid or mitigate the effect of undershoots or overshoots. The presence of an undershoot or overshoot even in a single time-step can have a profound effect on the computation. In the

Euler system example above, we saw that case where pressure or density becomes negative, and the code will no longer work. Setting these quantities to zero when they become negative resolves this problem, but may introduce significant errors into the computation. Thus, the aim is to avoid undershoots and overshoots as much as possible. In this section, we will see that any nonzero ε values may produce oscillations and discuss the dependence of these oscillations on ε as well as the size of the jump M. We will also examine the effect of the ratio of the time-step to the spatial grid size plays a role in the size of the oscillations.

In the following, we consider the prototype problem

$$u_t + f(u)_x = 0 \tag{1}$$

with step-function initial conditions

$$u(x,0) = \begin{cases} M, & x \in [-\frac{1}{8}, \frac{1}{8}] \\ 0, & otherwise \end{cases} \tag{2}$$

where $f(u) = \frac{1}{m}u^m$ for $m = 1, 2, 3, 4$. This problem is a representative of all Riemann problems, which are the building blocks of methods for hyperbolic PDEs.

In the previous section, we looked at what happens if ε is too large for a particular stencil. In this section, we study small ε sizes and determine the effect of the size of ε on the presence of oscillations. We begin with a mathematical analysis of the behavior of the undershoot for the example above with WENO-JS for one time-step using forward Euler (FE) and the two-stage second-order SSP Runge–Kutta method (RK2) and verify this analysis with numerical simulations. We then proceed to show numerically the behavior of the oscillations for one time-step of the three-stage third-order SSP Runge–Kutta method (RK3,3) and for the ten-stage fourth-order SSP Runge–Kutta method (RK10,4). We repeat these numerical explorations for WENO-M and for WENO-Z and observe similar behaviors. Our conclusion in this section is that size of the oscillation after one time-step scales as

$$\text{oscillations} \approx \frac{\varepsilon^2}{M^{4m-1}},$$

where this scaling depends on the CFL, the particular WENO method, and the value of m. It is important to note that smaller jumps M result in larger oscillations, all else being equal.

3.1 One Step of WENO-JS with Forward Euler Timestepping

FE is the first stage of many higher-order Runge–Kutta methods. A thorough study of WENO-JS with FE will help us have a better understanding of the behavior of WENO methods with other Runge–Kutta methods. In this section, we first estimate the undershoot/overshoot error of WENO-JS with FE for the linear case $f(u) = u$

and then extend the analysis to the nonlinear case $f(u) = \frac{1}{m}u^m$ where $m = 2, 3, 4$. We also show numerical results, which are consistent with our theoretical estimates.

3.1.1 Linear Case

First, we consider the behavior of WENO-JS on problem (1) above with $m = 1$ and initial conditions (2). We assume that $\varepsilon << M$ and examine the values of the numerical flux $\hat{f}_{j+1/2}$ for each possible j in a case-by-case sense. Note that in the formulas for the smoothness indicators, we ignore the factor of $\frac{1}{12}$ multiplying every term:

Stencil values	numerical flux
$u_{j-2}, \ldots, u_{j+2} = 0$	$\Rightarrow \quad \hat{f}_{j+1/2} = 0$
$u_{j-2}, \ldots, u_{j+1} = 0$ and $u_{j+2} = M$	$\Rightarrow \quad \hat{f}_{j+1/2} = \frac{\frac{3}{(16M^2+\varepsilon)^2}}{\frac{7}{\varepsilon^2}+\frac{3}{(16M^2+\varepsilon)^2}}(-\frac{1}{6}M)$
	$\approx -\frac{1}{3584}(\frac{\varepsilon}{M^2})^2 M$
$u_{j-2}, \ldots, u_j = 0$ and $u_{j+1}, u_{j+2} = M$	$\Rightarrow \quad \hat{f}_{j+1/2} = \frac{\frac{6}{(16M^2+\varepsilon)^2}(\frac{2}{6}M)+\frac{3}{(40M^2+\varepsilon)^2}(\frac{4}{6}M)}{\frac{1}{\varepsilon^2}+\frac{6}{(16M^2+\varepsilon)^2}+\frac{3}{(40M^2+\varepsilon)^2}}$
	$\approx \frac{29}{3200}(\frac{\varepsilon}{M^2})^2 M$
$u_{j-2}, u_{j-1} = 0$ and $u_j, \ldots, u_{j+2} = M$	$\Rightarrow \quad \hat{f}_{j+1/2} = \frac{\frac{1}{(40M^2+\varepsilon)^2}(\frac{11}{6}M)+\frac{6}{(16M^2+\varepsilon)^2}(\frac{7}{6}M)+\frac{3}{\varepsilon^2}M}{\frac{1}{(40M^2+\varepsilon)^2}+\frac{6}{(16M^2+\varepsilon)^2}+\frac{3}{\varepsilon^2}}$
	$\approx (1+\frac{17}{11520}(\frac{\varepsilon}{M^2})^2)M$
$u_{j-2} = 0$ and $u_{j-1}, \ldots, u_{j+2} = M$	$\Rightarrow \quad \hat{f}_{j+1/2} = \frac{\frac{1}{(16M^2+\varepsilon)^2}(\frac{4}{6}M)+\frac{6}{\varepsilon^2}M+\frac{3}{\varepsilon^2}M}{\frac{1}{(16M^2+\varepsilon)^2}+\frac{6}{\varepsilon^2}+\frac{3}{\varepsilon^2}}$
	$\approx (1-\frac{1}{6912}(\frac{\varepsilon}{M^2})^2)M$
$u_{j-2}, \ldots, u_{j+2} = M$	$\Rightarrow \quad \hat{f}_{j+1/2} = M$

Now, we consider what happens near the first discontinuity, the jump from 0 to M, when using the WENO-JS method with one step of the first-order forward Euler (FE) method in time:

$$u_j^1 = u_j^0 - \lambda(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}),$$

where $\lambda = \frac{\Delta t}{\Delta x}$.

1. If $u_{j-3}^0, \ldots, u_{j+2}^0 = 0$, then $u_j^1 = 0$
2. If $u_{j-3}^0, \ldots, u_{j+1}^0 = 0$ and $u_{j+2}^0 = M$, then $u_j^1 \approx -\lambda\{-\frac{1}{3584}(\frac{\varepsilon}{M^2})^2 M\} = \frac{1}{3584}\lambda(\frac{\varepsilon}{M^2})^2 M$
3. If $u_{j-3}^0, \ldots, u_j^0 = 0$ and $u_{j+1}^0, u_{j+2}^0 = M$, then $u_j^1 \approx -\lambda\{\frac{29}{3200}(\frac{\varepsilon}{M^2})^2 M + \frac{1}{3584}(\frac{\varepsilon}{M^2})^2 M\} = -\frac{837}{89600}\lambda(\frac{\varepsilon}{M^2})^2 M$ and there will be undershoot.
4. If $u_{j-3}^0, \ldots, u_{j-1}^0 = 0$ and $u_j^0, \ldots, u_{j+2}^0 = M$, then $u_j^1 \approx M - \lambda\{(1+\frac{17}{11520}(\frac{\varepsilon}{M^2})^2)M - \frac{29}{3200}(\frac{\varepsilon}{M^2})^2 M\} = \{1-\lambda(1-\frac{437}{57600}(\frac{\varepsilon}{M^2})^2)\}M$
5. If $u_{j-3}^0, u_{j-2}^0 = 0$ and $u_{j-1}^0, \ldots, u_{j+2}^0 = M$, then $u_j^1 \approx M - \lambda\{(1-\frac{1}{6912}(\frac{\varepsilon}{M^2})^2)M - (1+\frac{17}{11520}(\frac{\varepsilon}{M^2})^2)M\} = \{1+\frac{7}{4320}\lambda(\frac{\varepsilon}{M^2})^2)\}M$ and there will be overshoot.

6. If $u^0_{j-3} = 0$ and $u^0_{j-2}, \ldots, u^0_{j+2} = M$, then $u^1_j \approx M - \lambda\{M - (1 - \frac{1}{6912}(\frac{\varepsilon}{M^2})^2)M\} = \{1 - \frac{1}{6912}\lambda(\frac{\varepsilon}{M^2})^2\}M$
7. If $u^0_{j-3}, \ldots, u^0_{j+2} = M$, then $u^1_j = M$.

In the same way, we can also compute u^1_j when u^0_j jump from M to 0:

1. If $u^0_{j-3}, \ldots, u^0_{j+1} = M$ and $u^0_{j+2} = 0$, then $u^1_j \approx M - \lambda\{(1 + \frac{1}{3584}(\frac{\varepsilon}{M^2})^2)M - M\} = \{1 - \frac{1}{3584}\lambda(\frac{\varepsilon}{M^2})^2\}M$.
2. If $u^0_{j-3}, \ldots, u^0_j = M$ and $u^0_{j+1}, u^0_{j+2} = 0$, then $u^1_j \approx M - \lambda\{(1 - \frac{29}{3200}(\frac{\varepsilon}{M^2})^2)M - (1 + \frac{1}{3584}(\frac{\varepsilon}{M^2})^2)M\} = \{1 + \frac{837}{89600}\lambda(\frac{\varepsilon}{M^2})^2\}M$ and there will be overshoot.
3. If $u^0_{j-3}, \ldots, u^0_{j-1} = M$ and $u^0_j, \ldots, u^0_{j+2} = 0$, then $u^1_j \approx -\lambda\{-\frac{17}{11520}(\frac{\varepsilon}{M^2})^2M - (1 - \frac{29}{3200}(\frac{\varepsilon}{M^2})^2)M\} = \lambda\{1 - \frac{437}{57600}(\frac{\varepsilon}{M^2})^2\}M$
4. If $u^0_{j-3}, u^0_{j-2} = M$ and $u^0_{j-1}, \ldots, u^0_{j+2} = 0$, then $u^1_j \approx -\lambda\{\frac{1}{6912}(\frac{\varepsilon}{M^2})^2M + \frac{17}{11520}(\frac{\varepsilon}{M^2})^2M\} = -\frac{7}{4320}\lambda(\frac{\varepsilon}{M^2})^2M$ and there will be undershoot.
5. If $u^0_{j-3} = M$ and $u^0_{j-2}, \ldots, u^0_{j+2} = 0$, then $u^1_j \approx -\lambda\{0 - \frac{1}{6912}(\frac{\varepsilon}{M^2})^2M\} = \frac{1}{6912}\lambda(\frac{\varepsilon}{M^2})^2M$.

In summary, after one forward Euler time-step of WENO-JS, the maximum of overshoot is $\lambda\frac{837}{89600}(\frac{\varepsilon}{M^2})^2M = \lambda\frac{837}{89600}\frac{\varepsilon^2}{M^3}$, and the maximum of undershoot is $-\lambda\frac{837}{89600}\frac{\varepsilon^2}{M^3}$. In Table 1, we show numerically that the actual undershoot error of WENO-JS after one forward Euler time-step with $\lambda = 0.25$ is $2.3354 \times 10^{-3}\frac{\varepsilon^2}{M^3}$,

Table 1 The undershoot error of WENO-JS after one time-step with FE, using step function initial conditions, with $nx = 200$ points in space, and $CFL = 0.25$.

		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	ε = 1e-10	2.3354e-23	2.3354e-20	2.3354e-17	2.3354e-14	$2.3354 \times 10^{-3}\frac{\varepsilon^2}{M^3}$
	ε = 1e-11	2.3354e-25	2.3354e-22	2.3354e-19	2.3354e-16	
	ε = 1e-12	2.3354e-27	2.3354e-24	2.3354e-21	2.3354e-18	
$f(u) = u^2/2$	ε = 1e-18	2.5761e-38	2.5761e-31	2.5761e-24	2.5761e-17	$2.5761 \times 10^{-2}\frac{\varepsilon^2}{M^7}$
	ε = 1e-19	2.5761e-40	2.5761e-33	2.5761e-26	2.5761e-19	
	ε = 1e-20	2.5761e-42	2.5761e-35	2.5761e-28	2.5761e-21	
$f(u) = u^3/3$	ε = 1e-23	1.0702e-48	1.0702e-37	1.0702e-26	1.0702e-15	$1.0702 \times 10^{-2}\frac{\varepsilon^2}{M^{11}}$
	ε = 1e-24	1.0702e-50	1.0702e-39	1.0702e-28	1.0702e-17	
	ε = 1e-25	1.0702e-52	1.0702e-41	1.0702e-30	1.0702e-19	
$f(u) = u^4/4$	ε = 1e-29	7.3960e-61	7.3960e-46	7.3960e-31	7.3960e-16	$7.3960 \times 10^{-3}\frac{\varepsilon^2}{M^{15}}$
	ε = 1e-30	7.3960e-63	7.3960e-48	7.3960e-33	7.3960e-18	
	ε = 1e-31	7.3960e-65	7.3960e-50	7.3960e-35	7.3960e-20	

Table 2 The undershoot error of WENO-JS after one time-step with FE using step function initial conditions, $nx = 200, M = 1, \varepsilon = 1e-10$. This table shows the linear increase with the CFL number.

	$CFL = 0.125$	$CFL = 0.25$	$CFL = 0.5$
$f(u) = u$	1.1677e-23	2.3354e-23	4.6708e-23
$f(u) = u^2/2$	1.2880e-22	2.5761e-22	5.1521e-22
$f(u) = u^3/3$	5.3510e-23	1.0702e-22	2.1404e-22
$f(u) = u^4/4$	3.6980e-23	7.3960e-23	1.4792e-22

which is *exactly* the same as the undershoot/overshoot value

$$\lambda \frac{837}{89600} \frac{\varepsilon^2}{M^3} \tag{3}$$

derived above. This shows that the analysis above and the resulting estimate of the undershoot error for the case where ε is small enough is *sharp*. The numerical results in Table 2 confirm that the undershoot error is proportional to the CFL number, which is equal to λ in the linear case.

3.1.2 Nonlinear Cases

The analysis of the undershoot and overshoot in the case where f is nonlinear ($m > 1$) can be estimated in the same way as that for the linear case in the section above. However, for the nonlinear case, calculations are long and tedious, so we omit the details and only summarize the results from our analysis. We use the value CFL, where $\Delta t = \frac{CFL}{\max |f'(u)|} \Delta x = \frac{CFL}{M^{m-1}} \Delta x$. Clearly, $\lambda = \frac{CFL}{M^{m-1}}$ so in the linear case ($m = 1$), the two are identical. However, in the nonlinear case, the two are different, and while the term λ is more useful for analysis of a given problem, the CFL is useful in comparing different problems and different values of M. For this reason, in this section, we will use both.

Assuming that $\varepsilon \ll M \leq 1$, we can show analytically[1] that the maximal overshoot/undershoot is of the form

$$\alpha_m \times CFL \times M \times \left(\frac{\varepsilon}{M^{2m}}\right)^2, \tag{4}$$

where

$$\alpha_2 = \frac{779}{7560} \qquad \alpha_3 = \frac{6137}{143360} \qquad \alpha_4 = \frac{18871}{637875}.$$

[1] Details are omitted for space considerations.

The numerical results in Tables 1 and 2 confirm the formula for the undershoot error of WENO-JS (4) and the values of α_m. Once again, we stress that the above estimates of undershoot error are valid only if ε is small compared to M. When ε is not small enough, we will see different behavior. For this reason, we used the values of ε for which the constants in the overshoot/undershoot values converged. In the tables in this section, the values of ε shown are the largest ones that give the undershoot error accurate up to five digits for listed M.

Remark Note that the exponent 2 outside of the final parentheses in the relation above comes from the fact that we use the power $p = 2$ in the WENO method. In the remainder of this section, we assume that $p = 2$ and we will see this power appears in the oscillations for all methods. Using a different power p in the WENO method results in a corresponding exponent in the dependence of the oscillation on ε and M^{2m}.

3.2 One Step of WENO-JS with Two-Stage Second-Order Runge–Kutta

In this section, we provide an analysis and numerical confirmation for the behavior of the overshoot/undershoot when one time-step is taken using the strong stability preserving two-stage second-order Runge–Kutta (RK2) method [2, 11]

$$\begin{aligned} u^{(1)} &= u^n + \Delta t F(u^n) \\ u^{n+1} &= u^n + \frac{1}{2}\Delta t F(u^n) + \frac{1}{2}\Delta t F(u^{(1)}). \end{aligned}$$

This analysis shows the emerging complexity as higher-order methods are used. As before, we begin with a complete analysis of the linear case and omit the full details.

3.2.1 Linear Case

As before, we assume that $\varepsilon \ll M$. We can show that the maximal overshoot is of the form $P_1(\lambda)\frac{\varepsilon^2}{M^3}$, where P_1 is given by the maximum of two rational functions $P_1(\lambda) = \max\left(p_1(\lambda), p_2(\lambda)\right)$ where

$$\begin{aligned} p_1 =& (14400 - 136800\lambda + 504900\lambda^2 - 860760\lambda^3 + 598228\lambda^4 - 116306\lambda^5 \\ &+328413\lambda^6 - 467600\lambda^7 + 270625\lambda^8)/(1244160\lambda^2(4 - 19\lambda + 25\lambda^2)^2) \\ p_2 =& -\ (\lambda(2343600 - 19628460\lambda + 75692628\lambda^2 - 177350059\lambda^3 + 276036659\lambda^4 \\ &- 287679978\lambda^5 + 190949760\lambda^6 - 72004875\lambda^7 + 11803125\lambda^8))/(2508800 \\ &(-1 + \lambda)^3(10 - 31\lambda + 25\lambda^2)^2). \end{aligned}$$

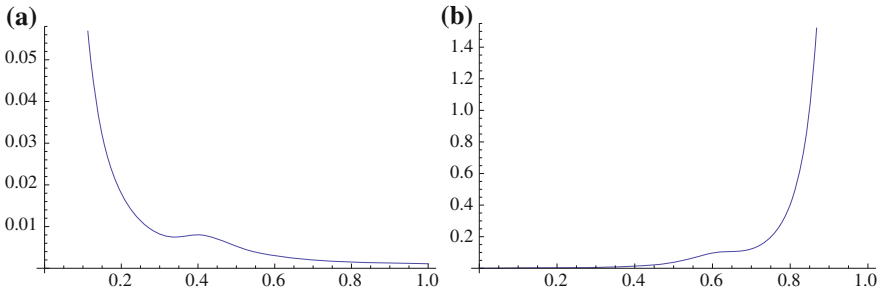

Fig. 4 The functions $p_1(\lambda)$ and $p_2(\lambda)$. (a) $p_1(\lambda)$. (b) $p_2(\lambda)$.

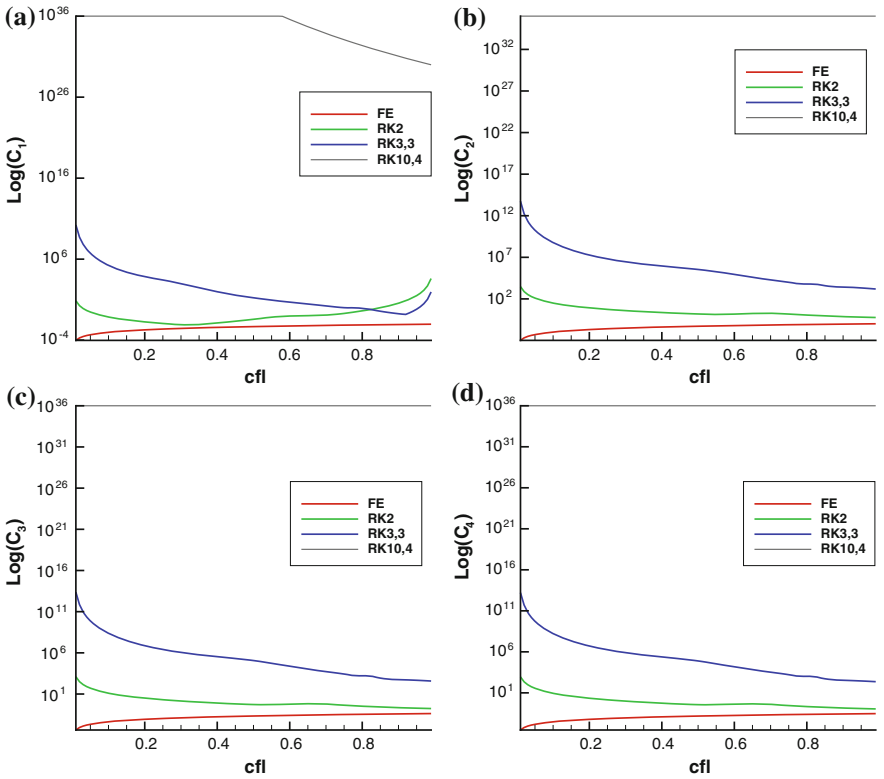

Fig. 5 The coefficient $C_m(CFL)$ of the undershoot of WENO-JS after one time-step. The CFL is between $[0.01, 0.99]$ with an increment of 0.01. (a) WENO-JS, $f(u) = u$. (b) WENO-JS, $f(u) = u^2/2$. (c) WENO-JS, $f(u) = u^3/3$. (d) WENO-JS, $f(u) = u^4/4$.

The undershoot is given by $Q_1(\lambda)\frac{\varepsilon^2}{M^3}$, where $Q(\lambda) = \max\left(q_1(\lambda), q_2(\lambda)\right)$. In this case, $p_1 = q_1$ and $p_2 = q_2$, so the maximal overshoot and undershoot are the same.

Let $C_1 = P_1$ and notice that C_1 is a function of λ that depends on the functions $p_1(\lambda)$ and $p_2(\lambda)$. Using the expressions of P and Q, and the graphs in Figure 4, we can see that when fixing ε small enough and decreasing λ from 1 to 0, the overshoot/undershoot will decrease at first and then increase. This is also reflected in Figure 5(a) in the green line.

3.2.2 Nonlinear Cases

Once again, we must assume that ε is small enough. For our purposes, we require $\varepsilon \ll M^{2m}$. If we take $\Delta t = CFL\Delta x/M^{m-1}$, then after one RK2 time-step, we get the following behavior for the overshoot/undershoot:

$$C_m(CFL)\frac{\varepsilon^2}{M^{4m-1}}, \tag{5}$$

where the value $C_m(CFL)$ depends only on the CFL number and m and on whether we are looking for an overshoot or undershoot. Figure 5 shows the graphs of C_m as a function of the CFL number for $m = 1, 2, 3, 4$. The numerical results for the undershoot error of WENO-JS with RK2 are given in Table 3, for $CFL = \frac{1}{4}$. These values verify that the undershoot error is proportional to $\frac{\varepsilon^2}{M^{4m-1}}$ and give us the values of $C_m(\frac{1}{4})$ in Table 4.

3.3 *One Step of WENO-JS with Higher-Order Time-stepping Schemes*

In practice, the explicit strong stability preserving three-stage third-order Runge–Kutta method of Shu and Osher (RK3,3) [2, 11] (given in the Appendix) is frequently used to evolve the solution forward. Another excellent alternative is Ketcheson's explicit strong stability preserving ten-stage fourth-order Runge–Kutta (RK10,4) method [2, 8], given in the Appendix. It is interesting to observe the effect of the higher-order time-stepping method on the overshoots and undershoots produced by WENO-JS. We experimented with these time-stepping methods, and our numerical results for the undershoots with different ε and M suggest that the undershoot error is of the form

$$C_m(CFL)\frac{\varepsilon^2}{M^{4m-1}}, \tag{6}$$

Table 3 The undershoot error of WENO-JS after one time-step with RK2 using step function initial conditions, $nx = 200$, $CFL = 0.25$.

		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	$\varepsilon = 1e{-}11$	$1.1413e{-}24$	$1.1413e{-}21$	$1.1413e{-}18$	$1.1413e{-}15$	$1.1413 \times 10^{-2} \frac{\varepsilon^2}{M^3}$
	$\varepsilon = 1e{-}12$	$1.1413e{-}26$	$1.1413e{-}23$	$1.1413e{-}20$	$1.1413e{-}17$	
	$\varepsilon = 1e{-}13$	$1.1413e{-}28$	$1.1413e{-}25$	$1.1413e{-}22$	$1.1413e{-}19$	
$f(u) = u^2/2$	$\varepsilon = 1e{-}20$	$5.4665e{-}40$	$5.4665e{-}33$	$5.4665e{-}26$	$5.4665e{-}19$	$5.4665 \frac{\varepsilon^2}{M^7}$
	$\varepsilon = 1e{-}21$	$5.4665e{-}42$	$5.4665e{-}35$	$5.4665e{-}28$	$5.4665e{-}21$	
	$\varepsilon = 1e{-}22$	$5.4665e{-}44$	$5.4665e{-}37$	$5.4665e{-}30$	$5.4665e{-}23$	
$f(u) = u^3/3$	$\varepsilon = 1e{-}26$	$2.0345e{-}52$	$2.0345e{-}41$	$2.0345e{-}30$	$2.0345e{-}19$	$2.0345 \frac{\varepsilon^2}{M^{11}}$
	$\varepsilon = 1e{-}27$	$2.0345e{-}54$	$2.0345e{-}43$	$2.0345e{-}32$	$2.0345e{-}21$	
	$\varepsilon = 1e{-}28$	$2.0345e{-}56$	$2.0345e{-}45$	$2.0345e{-}34$	$2.0345e{-}23$	
$f(u) = u^4/4$	$\varepsilon = 1e{-}31$	$1.4155e{-}62$	$1.4155e{-}47$	$1.4155e{-}32$	$1.4155e{-}17$	$1.4155 \frac{\varepsilon^2}{M^{15}}$
	$\varepsilon = 1e{-}32$	$1.4155e{-}64$	$1.4155e{-}49$	$1.4155e{-}34$	$1.4155e{-}19$	
	$\varepsilon = 1e{-}33$	$1.4155e{-}66$	$1.4155e{-}51$	$1.4155e{-}36$	$1.4155e{-}21$	

Table 4 The undershoot error of the WENO method after one time-step of the time-stepping method. Initial condition is the using step function initial conditions. The spatial number of points is $nx = 200$, with $CFL = 0.25$.

Time	Space	$C_1(0.25)$	$C_2(0.25)$	$C_3(0.25)$	$C_4(0.25)$
FE	WENO-JS	2.3354×10^{-3}	2.5761×10^{-2}	1.0702×10^{-2}	7.3960×10^{-2}
	WENO-M	6.9447×10^{-3}	1.1345×10^{-1}	4.6954×10^{-2}	3.2318×10^{-2}
	WENO-Z	3.0301×10^{-3}	3.4021×10^{-2}	1.4021×10^{-2}	9.6069×10^{-3}
*RK*2	WENO-JS	1.1413×10^{-2}	5.4665	2.0345	1.4155
	WENO-M	3.9962×10^{-2}	1.7779×10^{1}	6.4421	4.4725
	WENO-Z	1.3381×10^{-2}	5.0724	1.8059	1.2517
*RK*3, 3	WENO-JS	2.6639×10^{3}	6.6192×10^{6}	2.5725×10^{6}	1.7984×10^{6}
	WENO-M	2.4487×10^{4}	2.4851×10^{7}	9.8591×10^{6}	6.9289×10^{6}
	WENO-Z	3.2231×10^{3}	5.9477×10^{6}	2.3146×10^{6}	1.6191×10^{6}
*RK*10, 4	WENO-JS	2.9481×10^{45}	2.0732×10^{55}	8.6113×10^{54}	6.0468×10^{54}
	WENO-M	1.0722×10^{46}	6.5880×10^{55}	2.7364×10^{55}	1.9215×10^{55}
	WENO-Z	3.2443×10^{45}	1.8348×10^{55}	7.6210×10^{54}	5.3515×10^{54}

where C_m depends on the CFL number and m. However, the values of the coefficient C_m increase with the order of the time-stepping method, as shown in Figure 5. The values of $C_m(\frac{1}{4})$ for both RK3,3 and RK10,4 are given in Table 4. Clearly, the coefficients for the RK10,4 are much larger, indicating larger oscillations. Of course, higher-order time-stepping methods allow us to use larger time-steps and therefore larger values of *CFL*, for both accuracy and stability. Nevertheless, the sensitivity of the overshoot/undershoot to the order of the time-stepping method exceeds this benefit.

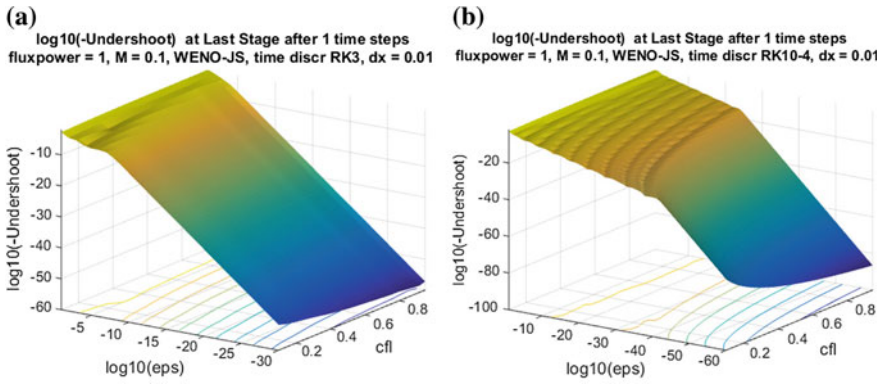

Fig. 6 Undershoot of WENO-JS after one time-step. $f(u) = u$, $M = 0.1$. Left: RK3,3, right: RK10,4. (a) WENO-JS with RK3,3, ε=1e-1 to 1e-30. (b) WENO-JS with RK10,4, ε=1e-1 to 1e-30.

Note that we have not determined exactly how ε_c, the critical value of ε at which the undershoot error starts the asymptotic behavior, depends on M, CFL number, time schemes, and the power m in $f(u)$. However, numerical tests show that higher-order time-stepping methods require smaller ε to reach the asymptotic region. In Figure 6, we consider the linear case $f(u) = u$ using $M = 0.1$. We see that RK10,4 requires much smaller ε than RK3,3 to reach the asymptotic region.

3.4 Other WENO Methods

We ran the same numerical tests using the WENO-M and WENO-Z methods. We observed that WENO-M and WENO-Z have a similar behavior to WENO-JS, in the sense that the undershoot error is of the form

$$C_m(CFL) \times M \times \left(\frac{\varepsilon}{M^{2m}}\right)^2.$$

The difference between methods results in different values of C_m. The values of C_m as a function of the CFL number for WENO-M and WENO-Z are given in Figure 7. In these Figures, the CFL number is between $[0.01, 0.99]$ with an increment 0.01. Again, we see that the higher-order the time scheme we use, the larger the coefficient C_m we have for the undershoot error, but the different WENO methods have similar coefficients. A comparison of the values of the $C_m(0.25)$ is given in Table 4, where we observe that the values are larger for WENO-M, but generally of the same order.

To study this behavior on a smoother function, with only one discontinuity, we consider the case where the initial condition u_0 is not a step-function:

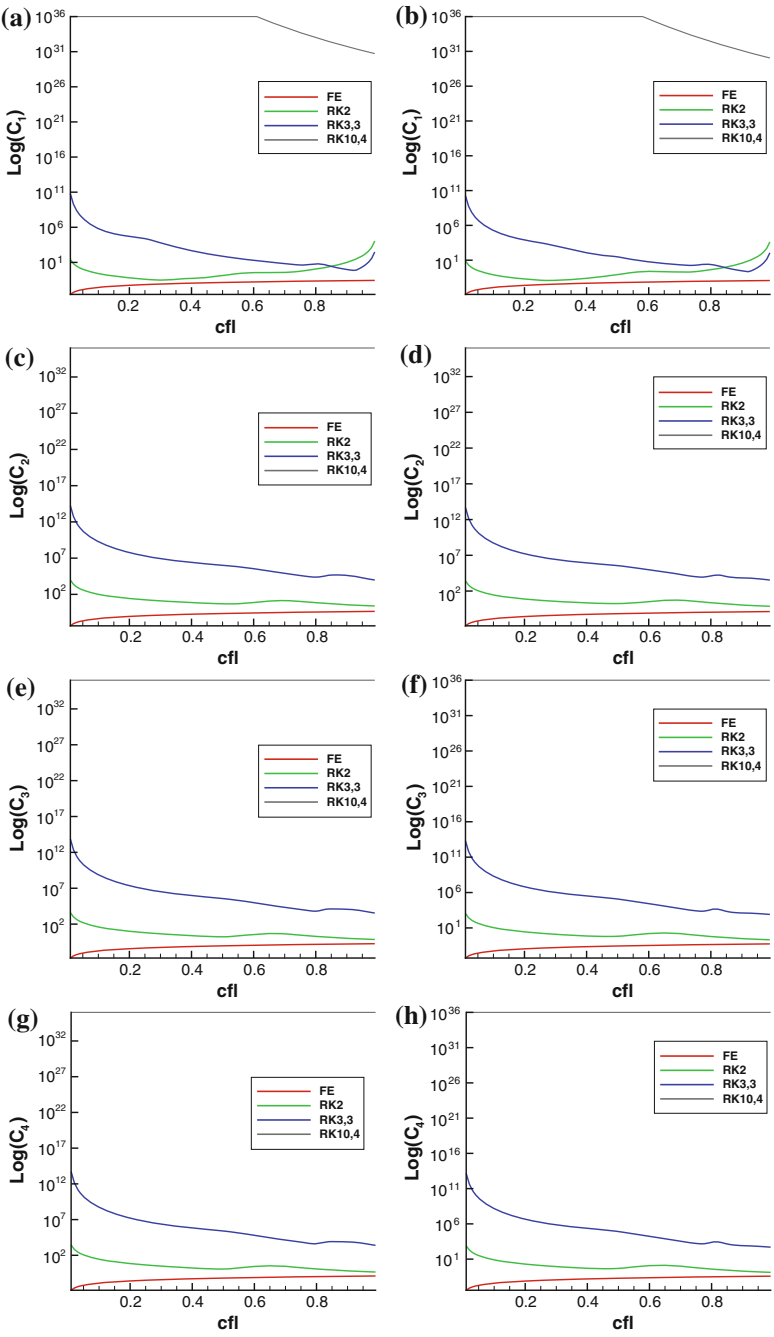

Fig. 7 The coefficient $C_m(CFL)$ of the undershoot of WENO-M and WENO-Z after one time-step. (a) WENO-M, $f(u) = u$. (b) WENO-Z, $f(u) = u$. (c) WENO-M, $f(u) = u^2/2$. (d) WENO-Z, $f(u) = u^2/2$. (e) WENO-M, $f(u) = u^3/3$. (f) WENO-Z, $f(u) = u^3/3$. (g) WENO-M, $f(u) = u^4/4$. (h) WENO-Z, $f(u) = u^4/4$.

Table 5 The undershoot error of the WENO method after one time-step of the time-stepping method with the nonstep function initial condition (7). The spatial number of points is $nx = 200$, with $CFL = 0.25$.

Time	Space	$C_1(0.25)$	$C_2(0.25)$	$C_3(0.25)$	$C_4(0.25)$
FE	WENO-JS	4.1831×10^{12}	4.8296×10^{12}	4.8295×10^{12}	4.8295×10^{12}
	WENO-M	1.1561×10^{13}	1.5339×10^{13}	1.5338×10^{13}	1.5338×10^{13}
	WENO-Z	4.3273×10^{12}	4.2740×10^{12}	4.2739×10^{12}	4.2739×10^{12}
*RK*2	WENO-JS	1.4404×10^{13}	3.3623×10^{16}	3.3866×10^{16}	3.4110×10^{16}
	WENO-M	3.9712×10^{13}	1.0678×10^{17}	2.0755×10^{17}	1.0833×10^{17}
	WENO-Z	1.4864×10^{13}	2.9755×10^{16}	2.9970×10^{16}	3.0186×10^{16}
*RK*3, 3	WENO-JS	7.8146×10^{12}	3.9911×10^{22}	4.0461×10^{22}	4.1034×10^{22}
	WENO-M	2.1346×10^{13}	1.2676×10^{23}	1.2850×10^{23}	1.3032×10^{23}
	WENO-Z	7.9869×10^{12}	3.5320×10^{22}	3.5806×10^{22}	3.6313×10^{22}
*RK*10, 4	WENO-JS	2.9505×10^{45}	1.0158×10^{71}	1.5943×10^{71}	1.6888×10^{71}
	WENO-M	1.0731×10^{46}	4.7812×10^{71}	5.0610×10^{71}	5.3607×10^{71}
	WENO-Z	3.2469×10^{45}	1.3328×10^{71}	1.4108×10^{71}	1.4945×10^{71}

$$u_0(x) = \begin{cases} 0, & x \in [-1, -1/8] \\ Me \cdot e^{\frac{1}{16(x-1/8)^2-1}}, & x \in (-1/8, 1/8] \\ 0, & x \in (1/8, 1] \end{cases} \tag{7}$$

We refer to this as the "shark" function due to its appearance which resembles a shark fin. Our numerical results show that the undershoots of WENO methods have similar forms to those in the previous subsection where we use step function initial conditions. The values of C_m shown in Table 5 are much larger than for the step function initial conditions, and the ε needed for the behavior to settle is much smaller, but the qualitative behavior is the same.

4 The Effect of the Sensitivity Parameter Over Time Integration

In the previous section, we studied the effect of the value of ε and the size of the jump M on one step. In this section, we discuss how the undershoots and overshoots propagate over time and their dependence on both ε and M. We observe that the behavior of oscillations for all three WENO methods, WENO-JS, WENO-M, and WENO-Z, is similar and that the choice of time integration also does not affect the long time behavior of the oscillations.

We ran the examples in Section 4 up to time $T = 2$, for WENO-JS, WENO-M, and WENO-Z, with time-stepping methods RK3,3 and RK10,4. Figure 8 shows that for $\varepsilon = 1.e - 6$, the undershoot varies over time integration. In all cases, the

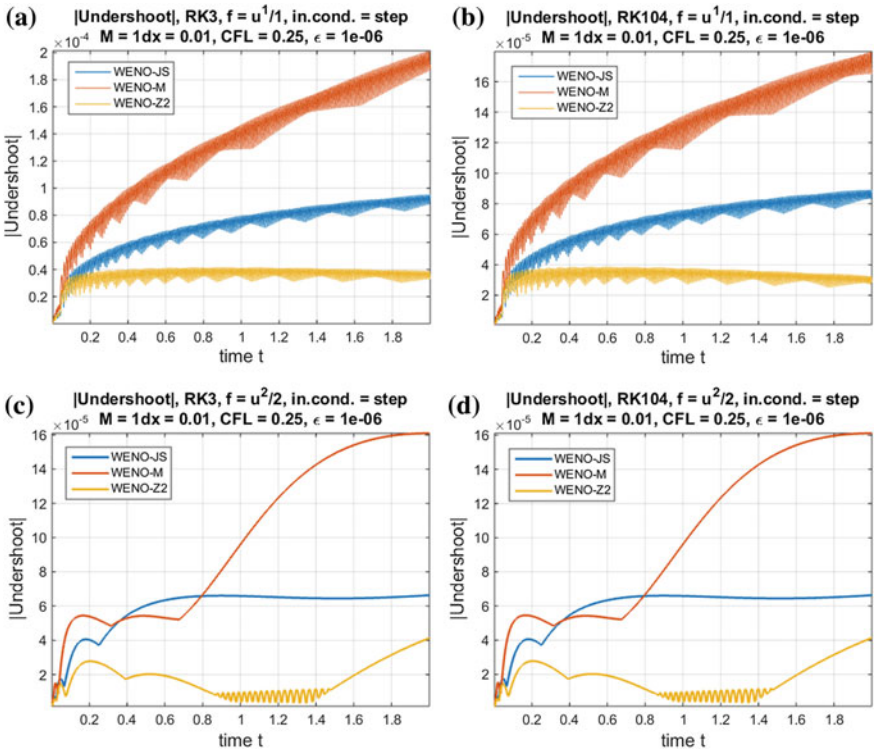

Fig. 8 Behavior of the undershoots for different WENO methods. (a) WENO with RK3,3, $m = 1$. (b) WENO with RK10,4, $m = 1$. (c) WENO with RK3,3, $m = 2$. (d) WENO with RK10,4, $m = 2$.

undershoot is larger for RK3,3 than for RK10,4, and the undershoot for WENO-M is the largest, followed by WENO-JS, and WENO-Z has the smallest undershoots. However, we also see that these overshoots are all of the same order of magnitude: approximately 10^{-5} for $\varepsilon = 1.e-6$. Numerous experiments with different values of ε, shown in Figure 9, confirm that the value of ε is the major determinant of the size of the oscillation. The major observation from these figures is that the long time behavior of the oscillations scales with $\sqrt{\varepsilon}$. This pattern is confirmed and further strengthened in our study of the maximal undershoot observed over 800 time-steps. The clear pattern that emerges from the numerical results in Tables 6 and 7 is that the undershoot is

$$\mathcal{O}(\sqrt{\varepsilon}/M^{m-1}). \tag{8}$$

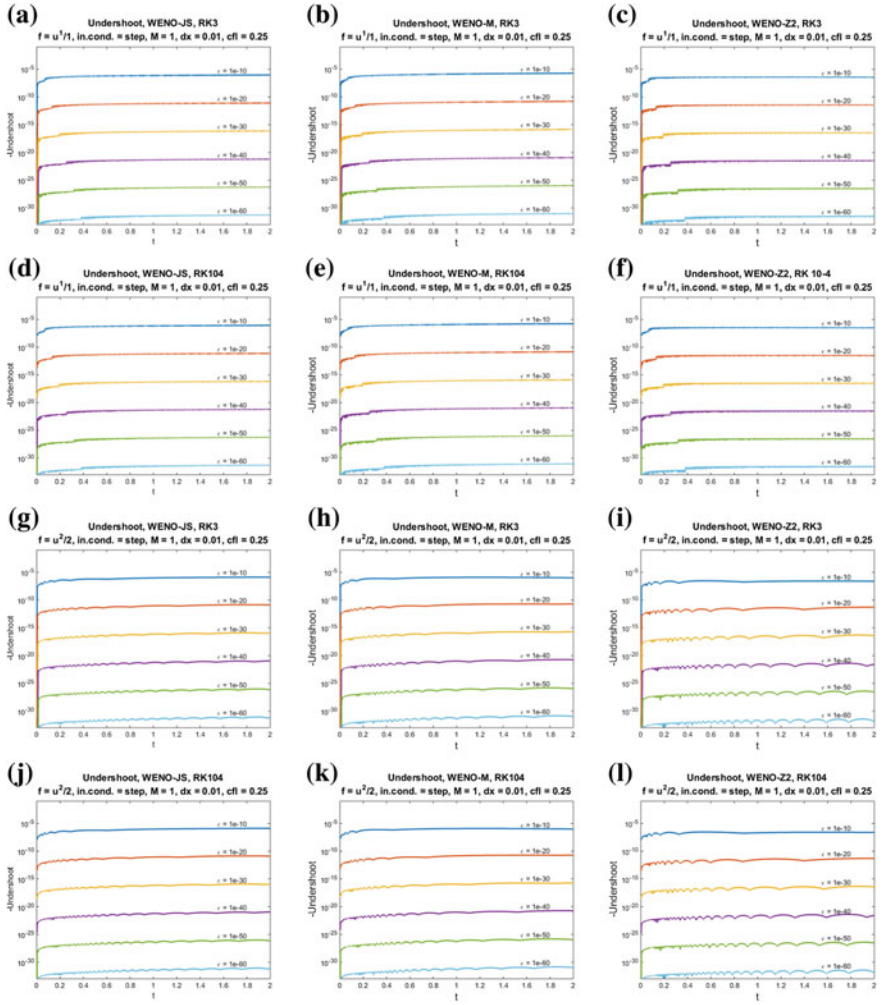

Fig. 9 Undershoot for WENO-JS with unit step initial condition, $CFL = 0.25$. Figures (a)–(f) have $m = 1$, while Figures (g)–(l) have $m = 2$. (a) WENO-JS with RK3,3. (b) WENO-M with RK3,3. (c) WENO-Z with RK3,3. (d) WENO-JS with RK10,4. (e) WENO-M with RK10,4. (f) WENO-Z with RK10,4. (g) WENO-JS with RK3,3. (h) WENO-M with RK3,3. (i) WENO-Z with RK3,3. (j) WENO-JS with RK10,4. (k) WENO-M with RK10,4. (l) WENO-Z with RK10,4.

Table 6 Maximum undershoot values for different WENO methods over 800 time-steps with RK3,3. $nx = 200$, $CFL = 0.25$.

WENO-JS with RK3,3						
		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	ε = 1e-12	9.3143E-08	9.4454E-08	9.6084E-08	9.5517E-08	$0.09\sqrt{\varepsilon}$
	ε = 1e-14	9.1005E-09	9.3143E-09	9.4454E-09	9.6084E-09	
	ε = 1e-16	8.8992E-10	9.1005E-10	9.3143E-10	9.4454E-10	
$f(u) = u^2/2$	ε = 1e-22	2.0375E-12	2.1317E-11	1.9802E-10	1.3910E-09	$0.2\sqrt{\varepsilon}/M$
	ε = 1e-24	2.0936E-13	2.0530E-12	2.0619E-11	1.9183E-10	
	ε = 1e-26	1.8374E-14	2.0375E-13	2.1317E-12	1.9802E-11	
$f(u) = u^3/3$	ε = 1e-32	3.8517E-17	4.3820E-15	4.2972E-13	4.3048E-11	$0.4\sqrt{\varepsilon}/M^2$
	ε = 1e-34	4.0889E-18	4.6507E-16	4.8072E-14	4.3279E-12	
	ε = 1e-36	4.2655E-19	4.2884E-17	4.2528E-15	4.6742E-13	
$f(u) = u^4/4$	ε = 1e-42	7.7903E-22	7.1192E-19	7.1688E-16	8.3674E-13	$0.8\sqrt{\varepsilon}/M^3$
	ε = 1e-44	7.5143E-23	7.3278E-20	8.2754E-17	7.4515E-14	
	ε = 1e-46	7.2119E-24	7.7690E-21	8.6189E-18	8.4379E-15	
WENO-JS with RK3,3						
		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	ε = 1e-12	1.9108E-07	1.9576E-07	1.9905E-07	2.0141E-07	$0.2\sqrt{\varepsilon}$
	ε = 1e-14	1.8409E-08	1.9108E-08	1.9576E-08	1.9905E-08	
	ε = 1e-16	1.8016E-09	1.8409E-09	1.9108E-09	1.9576E-09	
$f(u) = u^2/2$	ε = 1e-22	3.4297E-12	3.2013E-11	2.1447E-10	1.2370E-09	$0.3\sqrt{\varepsilon}/M$
	ε = 1e-24	3.3031E-13	3.4222E-12	2.7275E-11	1.8059E-10	
	ε = 1e-26	3.1131E-14	3.4297E-13	3.2013E-12	2.1447E-11	
$f(u) = u^3/3$	ε = 1e-32	6.2104E-17	5.9566E-15	7.1510E-13	5.0514E-11	$0.6\sqrt{\varepsilon}/M^2$
	ε = 1e-34	6.1962E-18	5.9036E-16	6.8629E-14	6.4428E-12	
	ε = 1e-36	6.0855E-19	6.1169E-17	6.4301E-15	7.1007E-13	
$f(u) = u^4/4$	ε = 1e-42	1.2019E-21	1.1666E-18	1.2908E-15	1.1876E-12	$1.2\sqrt{\varepsilon}/M^3$
	ε = 1e-44	1.2258E-22	1.1601E-19	1.2513E-16	1.2989E-13	
	ε = 1e-46	1.2459E-23	1.1654E-20	1.2143E-17	1.3348E-14	
WENO-Z with RK3,3						
		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	ε = 1e-12	9.3141E-08	9.4217E-08	9.6086E-08	9.5518E-08	$0.09\sqrt{\varepsilon}$
	ε = 1e-14	9.0855E-09	9.3146E-09	9.4450E-09	9.6084E-09	
	ε = 1e-16	8.8874E-10	9.1067E-10	9.3144E-10	9.4454E-10	
$f(u) = u^2/2$	ε = 1e-22	2.0086E-12	2.1317E-11	1.9802E-10	1.3910E-09	$0.2\sqrt{\varepsilon}/M$
	ε = 1e-24	2.1053E-13	2.0530E-12	2.0619E-11	1.9183E-10	
	ε=1e-26	1.8560E-14	2.0375E-13	2.1317E-12	1.9802E-11	
$f(u) = u^3/3$	ε = 1e-32	3.9087E-17	4.3820E-15	4.2972E-13	4.3048E-11	$0.4\sqrt{\varepsilon}/M^2$
	ε = 1e-34	3.9973E-18	4.6507E-16	4.8072E-14	4.3279E-12	
	ε = 1e-36	4.2212E-19	4.2884E-17	4.2528E-15	4.6742E-13	
$f(u) = u^4/4$	ε = 1e-42	7.7883E-22	7.1192E-19	7.1688E-16	8.3674E-13	$0.8\sqrt{\varepsilon}/M^3$
	ε = 1e-44	7.5875E-23	7.3278E-20	8.2754E-17	7.4515E-14	
	ε = 1e-46	7.3054E-24	7.7690E-21	8.6189E-18	8.4379E-15	

Table 7 Maximum undershoot values for different WENO methods over 800 time-steps with RK10,4. $nx = 200$, $cfl = 0.25$.

WENO-JS with RK10,4						
		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	ε = 1e-12	8.7682E-08	8.9456E-08	9.0061E-08	8.9096E-08	$0.09\sqrt{\varepsilon}$
	ε = 1e-14	8.5621E-09	8.7682E-09	8.9456E-09	9.0061E-09	
	ε = 1e-16	8.3900E-10	8.5621E-10	8.7682E-10	8.9456E-10	
$f(u) = u^2/2$	ε = 1e-22	2.0375E-12	2.1317E-11	1.9801E-10	1.3908E-09	$0.2\sqrt{\varepsilon}/M$
	ε = 1e-24	2.0938E-13	2.0533E-12	2.0622E-11	1.9184E-10	
	ε = 1e-26	1.8376E-14	2.0375E-13	2.1317E-12	1.9801E-11	
$f(u) = u^3/3$	ε = 1e-32	3.8523E-17	4.3814E-15	4.2966E-13	4.3052E-11	$0.4\sqrt{\varepsilon}/M^2$
	ε = 1e-34	4.0884E-18	4.6508E-16	4.8073E-14	4.3274E-12	
	ε = 1e-36	4.2653E-19	4.2891E-17	4.2537E-15	4.6747E-13	
$f(u) = u^4/4$	ε = 1e-42	7.7905E-22	7.1203E-19	7.1704E-16	8.3673E-13	$0.8\sqrt{\varepsilon}/M^3$
	ε = 1e-44	7.5150E-23	7.3265E-20	8.2744E-17	7.4536E-14	
	ε = 1e-46	7.2126E-24	7.7681E-21	8.6190E-18	8.4371E-15	
WENO-M with RK10,4						
		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	ε = 1e-12	1.7391E-07	1.7691E-07	1.7946E-07	1.7970E-07	$0.17\sqrt{\varepsilon}$
	ε = 1e-14	1.7005E-08	1.7391E-08	1.7691E-08	1.7946E-08	
	ε = 1e-16	1.6484E-09	1.7005E-09	1.7391E-09	1.7691E-09	
$f(u) = u^2/2$	ε = 1e-22	3.4304E-12	3.2001E-11	2.1465E-10	1.2365E-09	$0.3\sqrt{\varepsilon}/M$
	ε = 1e-24	3.3045E-13	3.4220E-12	2.7258E-11	1.8063E-10	
	ε = 1e-26	3.1149E-14	3.4304E-13	3.2001E-12	2.1465E-11	
$f(u) = u^3/3$	ε = 1e-32	6.2056E-17	5.9619E-15	7.1537E-13	5.0485E-11	$0.6\sqrt{\varepsilon}/M^2$
	ε = 1e-34	6.1915E-18	5.8986E-16	6.8670E-14	6.4411E-12	
	ε = 1e-36	6.0807E-19	6.1120E-17	6.4351E-15	7.1012E-13	
$f(u) = u^4/4$	ε = 1e-42	1.2024E-21	1.1672E-18	1.2913E-15	1.1874E-12	$1.2\sqrt{\varepsilon}/M^3$
	ε = 1e-44	1.2262E-22	1.1607E-19	1.2518E-16	1.2989E-13	
	ε = 1e-46	1.2462E-23	1.1660E-20	1.2149E-17	1.3350E-14	
WENO-Z with RK10,4						
		$M = 1$	$M = 0.1$	$M = 0.01$	$M = 0.001$	pattern
$f(u) = u$	ε = 1e-12	8.7426E-08	8.9384E-08	9.0065E-08	8.9096E-08	$0.09\sqrt{\varepsilon}$
	ε = 1e-14	8.5971E-09	8.7851E-09	8.9456E-09	9.0061E-09	
	ε = 1e-16	8.4214E-10	8.5788E-10	8.7687E-10	8.9456E-10	
$f(u) = u^2/2$	ε = 1e-22	2.0084E-12	2.1316E-11	1.9801E-10	1.3908E-09	$0.2\sqrt{\varepsilon}/M$
	ε = 1e-24	2.1055E-13	2.0533E-12	2.0622E-11	1.9184E-10	
	ε = 1e-26	1.8562E-14	2.0375E-13	2.1317E-12	1.9801E-11	
$f(u) = u^3/3$	ε = 1e-32	3.9095E-17	4.3814E-15	4.2966E-13	4.3052E-11	$0.4\sqrt{\varepsilon}/M^2$
	ε = 1e-34	3.9963E-18	4.6508E-16	4.8073E-14	4.3274E-12	
	ε = 1e-36	4.2208E-19	4.2891E-17	4.2537E-15	4.6747E-13	
$f(u) = u^4/4$	ε = 1e-42	7.7883E-22	7.1203E-19	7.1704E-16	8.3673E-13	$0.8\sqrt{\varepsilon}/M^3$
	ε = 1e-44	7.5877E-23	7.3265E-20	8.2744E-17	7.4536E-14	
	ε = 1e-46	7.3057E-24	7.7681E-21	8.6190E-18	8.4371E-15	

Table 8 Critical CFL stability limits for the linear and nonlinear equations with $\varepsilon = 1e-50$.

Time-stepping method	WENO scheme	Linear Equation		Burgers' Equation	
		CFL_{shark}	CFL_{step}	CFL_{shark}	CFL_{step}
RK3,3	JS	0.82	0.78	1.48	1.44
	M	0.79	0.73	1.66	1.48
	Z	0.78	0.73	1.58	1.44
RK10,4	JS	3.15	3.20	6.64	5.37
	M	2.91	2.94	6.96	5.79
	Z	3.07	3.03	6.98	5.82

4.1 Stability Limits

The long time behavior of the solution led us to investigate the stability limits of the numerical methods used. We find that nonlinear instability occurs at certain CFL numbers that depend most strongly on the time-stepping algorithm and to some extent on the WENO method. Small variations depending on the initial condition are observed as well when we tested the initial functions (2) and (7). When ε is reasonably small, its precise value does not seem to play a significant role. The largest stable value of CFL for the different methods is given in Table 8.

In the linear case, with a given reasonably small ε, the differences between the allowable time-step for the different WENO methods are not significant. However, the choice of time-stepping method makes a difference. The CFL for the ten-stage fourth-order RK10,4 is significantly larger than that of three-stage third-order RK3,3. Even when we take into account that RK10,4 requires ten stages and RK3,3 only three, we can conclude that RK10,4 has an advantage, because it allows for higher *effective* CFL numbers when taking into account the number of stages. For example, for the shark profile RK3,3 with WENO-Z requires at most 0.78 CFL (0.78/3 = 0.26), while RK10,4 for the same WENO method has a limitation of 3.07 CFL (3.07/10 = 0.307).

The linear and nonlinear cases exhibit remarkably different behaviors in terms of the allowable CFL, but the nonlinear cases all behaved similarly for $m = 2, 3, 4$. For this reason, the results for Burgers' equation serve as a representative for the nonlinear problems. For Burgers' equation, we observed that WENO-JS had the smallest allowable time-step, but that all WENO methods behaved generally the same. The time-stepping methods result in very different stability limits, and once again, we observe that RK10,4 has a significant advantage over RK3,3 even when normalizing for the number of function evaluations.

Acknowledgments This project is part of an IMA-funded WhAM! workshop and ongoing program. We thank the IMA for its support. Our thanks to Daniel Higgs who produced the graphs in Section 3.

Appendix A

Strong Stability Preserving Runge–Kutta Time Evolution Methods

To preserve the designed nonlinear stability properties of the WENO methods, it is advisable to use strong stability preserving Runge–Kutta methods [2]. In this work, we use

The three-stage third order SSP method by Shu and Osher [11, 12] **(RK3,3)**

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

The ten-stage fourth order SSP method by Ketcheson [8] **(RK10,4)**

$$\begin{aligned} u^{(1)} &= u^n + \frac{1}{6}\Delta t F(u^n) \\ u^{(k+1)} &= u^{(k)} + \frac{1}{6}\Delta t F(u^{(k)}) \quad \text{for } k = 1, 2, 3 \\ u^{(5)} &= \frac{3}{5}u^n + \frac{2}{5}u^{(4)} + \frac{1}{15}\Delta t F(u^{(4)}) \\ u^{(k+1)} &= u^{(k)} + \frac{1}{6}\Delta t F(u^{(k)}) \quad \text{for } k = 5, 6, 7, 8 \\ u^{n+1} &= \frac{1}{25}u^n + \frac{9}{25}u^{(4)} + \frac{3}{5}u^{(9)} + \frac{3}{50}\Delta t F(u^{(4)}) + \frac{1}{10}\Delta t F(u^{(9)}) \,. \end{aligned}$$

References

1. R. Borges, M. Carmona, B. Costa, and W. S. Don, *An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws*, Journal of Computational Physics, 227 (2008), pp. 3191 – 3211.
2. S. Gottlieb, D. I. Ketcheson, and C.- W. Shu, *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations*, World Scientific Press, 2011.

3. A. HARTEN, B. ENGQUIST, S. OSHER AND S. CHAKRAVARTHY, *Uniformly High Order Essentially Non-Oscillatory Schemes I*, SIAM Journal on Numerical Analysis, **vol. 24**, (1987), pp.279-309.
4. A. HARTEN, B. ENGQUIST, S. OSHER AND S. CHAKRAVARTHY, *Uniformly High Order Essentially Non-Oscillatory Schemes III*, Journal of Computational Physics, **vol. 71**, (1987), pp.231-303.
5. A. K. HENRICK, T. D. ASLAM, AND J. M. POWERS, *Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points*, Journal of Computational Physics, 207 (2005), pp. 542 – 567.
6. G. B. JACOBS AND W.- S. DON, *A high-order WENO-Z finite difference based particle-source-in-cell method for computation of particle-laden flows with shocks*, Journal of Computational Physics, 228 (2009), pp. 1365 – 1379.
7. G.- S. JIANG AND C.- W. SHU, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1996), pp. 202 – 228.
8. D. I. KETCHESON, *Highly efficient strong stability preserving Runge–Kutta methods with low-storage implementations*, SIAM Journal on Scientific Computing, 30 (2008), pp. 2113–2136.
9. R.J. LEVEQUE, *Numerical Methods for Conservation Laws*, (Lectures in Mathematics), Birkhauser, Basel; Boston; Berlin; 1992.
10. X.- D. LIU, S. OSHER AND T. CHAN, *Weighted Essentially Non-Oscillatory Schemes*, Journal of Computational Physics, **v. 115**, (1994), p.200.
11. C.- W. SHU, *Total-Variation-Diminishing Time Discretizations*, SIAM J. Sci. Stat. Comput. **v. 9**, (1988), pp.1073-1084.
12. C.- W. SHU AND S. OSHER, *Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes*, J. Comput. Phys. **v. 77**, (1988), pp.439-471.
13. C.- W. SHU, *High order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD*, International Journal of Computational Fluid Dynamics, v17 (2003), pp.107-118.

http://www.springer.com/978-1-4939-6398-0

Topics in Numerical Partial Differential Equations and Scientific Computing

Brenner, S.C. (Ed.)
2016, X, 176 p. 88 illus., Hardcover
ISBN: 978-1-4939-6398-0