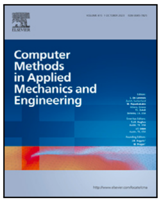
Computer Methods in Applied Mechanics and Engineering 421 (2024) 116818

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma



A finite volume multilevel WENO scheme for multidimensional scalar conservation laws

Todd Arbogast[a,b,1], Chieh-Sen Huang[c,*,2], Chenyu Tian[b]

[a] *Department of Mathematics, C1200; University of Texas at Austin, Austin, TX 78712, USA*
[b] *Oden Institute for Computational Engineering and Sciences, C0200; University of Texas at Austin, Austin, TX 78712, USA*
[c] *Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan, ROC*

ARTICLE INFO

MSC:
65D15
65M08
76M12

Keywords:
Stencil polynomial
Reconstruction
ML-WENO
Hyperbolic equation

ABSTRACT

We develop a general framework for solving scalar conservation laws using finite volume weighted essentially non oscillatory (WENO) techniques on general computational meshes in multiple space dimensions. We address two fundamental issues. First, polynomial approximations on general stencils of mesh cells can be of poor quality, even for what appear to be geometrically nice stencils. We present a robust and efficient procedure for producing accurate stencil polynomial approximations. Bad stencils are identified by considering the condition number of the linear system used to define the stencil polynomial. Second, we develop a novel and efficient finite volume, multilevel WENO (ML-WENO) reconstruction that is flexible enough to be applied effectively in a variety of settings and with essentially any reasonable set of stencils. It combines stencil polynomial approximations of various degrees with a nonlinear weighting biasing the reconstruction away from both inaccurate oscillatory polynomials of high degree (i.e., those crossing a shock or steep front) and smooth polynomials of low degree, thereby selecting the smooth polynomial(s) of maximal degree of approximation. We conduct numerical tests showing poor quality mesh stencils, the behavior of the reconstruction for both smooth and discontinuous functions, and applications to scalar conservation laws.

1. Introduction

We consider numerical schemes for solving scalar hyperbolic conservation laws

$$u_t(\mathbf{x},t) + \nabla\cdot\mathbf{f}(u;\mathbf{x},t) = 0, \quad t>0,\ \mathbf{x}\in\Omega, \tag{1}$$

where the domain $\Omega\subset\mathbb{R}^d$, $d\geq 1$ is an integer, $u:\Omega\times(0,\infty)\to\mathbb{R}$, and $\mathbf{f}:\mathbb{R}\times\Omega\times(0,\infty)\to\mathbb{R}^d$. We are interested in the case of multiple space dimensions, $d\geq 2$.

Weighted essentially non-oscillatory (WENO) schemes for solving (1) in one space dimension [1–3] have proven quite effective in practice; moreover, they work well on rectangular meshes for $d\geq 2$. Finite difference WENO schemes are particularly problematic in multiple space dimensions, so we concentrate on finite volume schemes on possibly unstructured meshes. In this paper, we address two challenging issues that make finite volume WENO schemes difficult to implement effectively on unstructured meshes.

[*] Corresponding author.
E-mail addresses: arbogast@oden.utexas.edu (T. Arbogast), huangcs@math.nsysu.edu.tw (C.-S. Huang), chenyu@ices.utexas.edu (C. Tian).

[1] Funded in part by the U.S. National Science Foundation grant DMS-1912735.

[2] Funded by the Taiwan Ministry of Science and Technology grant MOST 109-2115-M-110-003-MY3, the Taiwan National Science and Technology Council grant NSTC 112-2115-M-110-005-MY2, and the National Center for Theoretical Sciences, Taiwan.

https://doi.org/10.1016/j.cma.2024.116818
Received 17 October 2023; Received in revised form 27 January 2024; Accepted 27 January 2024

First, it is a challenge to define good mesh stencils in multiple space dimensions (see, e.g., [4–9]). Given a stencil of K cells, the cell averages of the stencil polynomial P should agree with the cell averages of the solution u. If P has degree $r-1$, it has $N_r = \binom{r-1+d}{d}$ coefficients, which may not match K. This can be resolved by using a least squares fitting technique [4]. It is known that the approximation error is of order r in the mesh spacing h when the solution is smooth [10]; that is, that

$$|P(\mathbf{x}) - u(\mathbf{x})| \le Ch^r, \tag{2}$$

for some constant C. Even though it is well defined, the stencil polynomial may not give a good approximation of the solution u, since C can be large.

In this paper, we do *not* discuss the best strategy for defining stencils; rather, given a stencil, we determine the degree of the polynomial that should be used on the stencil. The strategy we propose here is to attempt to use a polynomial of maximal degree $r-1$ on the stencil, so $N_r \le K < N_{r+1}$. We will present a theoretical analysis resulting in a simple criterion (see (10) below) involving the condition number of the least squares fitting technique to determine whether the polynomial provides a good approximation of the solution u. If not, we reduce r by 1 and try again. At worst, the strategy will terminate with a constant polynomial.

A second challenge is to reconstruct the solution in a flexible way. We would like to be able to use any number and arrangement of stencils, and to allow the stencil polynomials to be of any degree, including constant polynomials. For example, constants would be especially useful in solving (1) near the boundary of the domain, $\partial\Omega$, when a shock or steep front is entering *into* the domain, and only a single layer of mesh cells provides accurate approximation of the solution (see, e.g., [11]). Conversely, near the outflow boundaries it would be useful to include additional high order stencil polynomials.

Classic WENO reconstructions [12] require special linear weights; consequently, they are too inflexible to be used on unstructured meshes. WENO reconstructions with adaptive order (WENO-AO), also called central or compact WENO schemes (CWENO) [13–16] were developed to allow arbitrary weighting, and these naturally extend to multiple space dimensions. When one combines polynomials of exactly two different degrees, the reconstructions are considered to be two-level. WENO-AO reconstructions are naturally two-level, and in fact require a single large stencil supporting a high degree polynomial and many small stencils supporting polynomials of the same low degree. Extension to multiple levels (i.e., reconstructions using polynomials of various degrees) has been defined [14,15], but in a recursive manner that requires a hierarchical set of stencils.

There is by now an extensive literature on multi-resolution WENO techniques. These seem to have originated in the work of Harten [17–19], and they were later adapted to WENO schemes (see, e.g., [20–26]). Like WENO-AO, these methods require a hierarchical set of stencils that can be described as a nested set of central spatial stencils. As stated in [24], "Generally speaking, the main objective of using the multi-resolution technique is to focus the computational effort mainly in the small regions containing strong discontinuities". Our objective is quite different. We desire to define a flexible reconstruction so that it can be applied effectively in a variety of settings and with essentially any reasonable set of stencils.

In recent work of Semplice and Visconti [27], an adaptive order CWENO-type reconstruction is defined in one space dimension. In their approach, the stencils do not need to be hierarchically arranged. They envision a three level reconstruction with a large "central" stencil supporting a high degree polynomial, many stencil polynomials of intermediate degree, and some low degree polynomials, which may simply be the constant polynomial. The low order polynomials use linear weights that the authors call *infinitesimally small*, meaning that they are proportional to the mesh spacing to an appropriate power.

In this paper, we develop novel multilevel WENO (ML-WENO) reconstructions suitable for use in multiple space dimensions using unstructured computational meshes. Our multilevel reconstruction seamlessly weights the stencil polynomial approximations of various degrees (including constant polynomials) to bias away from oscillatory polynomials and smooth polynomials of low degree, thereby selecting the smooth polynomial(s) of maximal degree of approximation.

The paper is organized as follows. In the next section, we discuss the selection of stencil polynomials and their approximation properties. In Section 3, we define our ML-WENO reconstruction and explore its theoretical properties. Section 4 is devoted to a computational investigation of the ML-WENO reconstruction in isolation. Section 5 shows the performance of the stencil selection procedure and the ML-WENO reconstructions when applied to solving scalar conservation laws with either periodic or nonperiodic boundary conditions. Finally, a summary of our results and conclusions appear in the last section, as well as suggestions for future work.

2. Selection of the stencil polynomials

We begin by making precise some of the terms and notation we use in the multidimensional setting. A *cell* or *element* $E \subset \mathbb{R}^d$ is a bounded, closed subset that is homeomorphic to a closed ball (i.e., the closure of its interior is the cell and the interior is simply connected), such as a polytope. Let the domain $\Omega \subset \mathbb{R}^d$ and $h > 0$ be given. Let $\mathcal{T}_h$ be a quasiuniform mesh of cells that tesselate Ω such that $h_E = \operatorname{diam}(E) \le h = \max_{E' \in \mathcal{T}_h} h_{E'}$ for each cell $E \in \mathcal{T}_h$.

By saying that the cells tesselate the domain, we mean that their union is the closure of Ω and they intersect only on their boundaries. Recall that as $h \to 0^+$, we write

$$f(h) = \Theta(h) \quad \Longleftrightarrow \quad c_1 h \le |f(h)| \le c_2 h, \tag{3}$$

for positive constants c_1 and c_2 when $h > 0$ is sufficiently small. Quasiuniformity of the mesh means that $h_E = \Theta(h)$ uniformly as $h \to 0^+$.

A mesh *stencil* $S = \{E_1, E_2, \dots, E_K\}$ is a subset of cells of $\mathcal{T}_h$ such that their union $\Omega_S = \cup_{k=1}^{K} E_k$ is a closed, bounded subset that is homeomorphic to a closed ball. The cell averages of $u : \Omega \to \mathbb{R}$, the *solution*, are

$$\bar{u}_E = \frac{1}{|E|} \int_E u(\mathbf{x})\, d\mathbf{x} \quad \forall E \in \mathcal{T}_h. \tag{4}$$

We denote the space of polynomials of degree $r \geq 0$ in d variables by $\mathbb{P}_r$.

2.1. Stencil polynomials

Given a stencil S of the mesh and an index $r \geq 1$, our goal is to define an associated *stencil polynomial* $P \in \mathbb{P}_{r-1}$. To increase the numerical stability of the computations, we need to represent the polynomial in a way that is independent of the mesh spacing h. Several authors address this issue to handle mesh anisotropy (see, e.g., [9,10,28,29]). However, we have assumed quasiuniform meshes here, so we use a simple scaling and represent P as

$$P(\mathbf{x}) = \sum_{|\alpha|<r} c_\alpha \left(\frac{\mathbf{x} - \mathbf{x}_S}{h_S} \right)^{\alpha}, \tag{5}$$

where $\mathbf{x}_S$ is the center of $\Omega_S = \bigcup_{E \in S} E \subset \mathbb{R}^d$, $h_S = \max\{h_E : E \in S\}$, and α is a d-dimensional multi-index.

In the finite volume context, we require that the "mass" of u be preserved for each $E \in S$,

$$\frac{1}{|E|} \int_E P(\mathbf{x})\, d\mathbf{x} = \bar{u}_E, \tag{6}$$

which leads directly to the linear system

$$A\mathbf{c} = \mathbf{u} \tag{7}$$

for the vector of polynomial coefficients $\mathbf{c} = (c_\alpha)$ given the vector of solution averages $\mathbf{u} = (\bar{u}_E)$ on the stencil, i.e., only for $E \in S$.

The matrix A is $K \times N$ where $N = \begin{pmatrix} r+d-1 \\ d \end{pmatrix}$ is the number of coefficients and $K \geq N$ is the number of cells in the stencil. The linear system is generally overdetermined, i.e., $K > N$, so we need to find the least squares solution by solving the normal equations $A^T A\mathbf{c} = A^T \mathbf{u}$. We have just defined a linear projection operator π_S taking u to $\pi_S u = P$ as given in (5).

However, it is not enough to merely define the polynomial P. We also require that it approximates the solution accurately. To be precise, each stencil under consideration $S = S^{(r)}$ must have an index $r \geq 1$ such that its associated stencil polynomial $P \in \mathbb{P}_{r-1}$ satisfies

$$\begin{aligned} &|\mathcal{D}^\alpha u(\mathbf{x}) - \mathcal{D}^\alpha P(\mathbf{x})| \\ &\leq \begin{cases} C\, h^{r-|\alpha|} & \text{if } u \text{ is smooth on the stencil,} \\ C & \text{otherwise,} \end{cases} \qquad \forall \mathbf{x} \in E \in S^{(r)}, \end{aligned} \tag{8}$$

for some constant $C > 0$ independent of the mesh as $h \to 0^+$, where $\mathcal{D}^\alpha$ is partial differentiation of order α and $|\alpha| \leq r$.

The Bramble–Hilbert Lemma [30,31] tells us that P will approximate u and its derivatives provided that two conditions hold. First, π_S must preserve polynomials, which it clearly does. Second, π_S must be a bounded operator, which it will be as long as the matrix $A^T A$ is well conditioned and the mesh is quasiuniform.

We propose finding the compact singular value decomposition (SVD) and the singular values $s_1 \geq s_2 \geq \cdots \geq s_M > 0$ of A, i.e.,

$$A = U \Sigma V^T, \quad \Sigma = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_M \end{bmatrix}. \tag{9}$$

The condition number of the least squares matrix $A^T A$ is infinite if $M < N$ (i.e., if the system is singular and cannot be solved) and $(s_1/s_N)^2$ otherwise (i.e., $M = N$). In double precision floating point arithmetic, to maintain accuracy we would require that

$$(s_1/s_N)^2 < \kappa_{\max}, \quad \text{with perhaps } \kappa_{\max} = 1\text{e+8}. \tag{10}$$

If this condition fails, we should reject the approximation P.

In the case of a rejected P, one could add cells to the stencil, but it is unclear how one might do this efficiently, reliably, and automatically in more than one space dimension. Instead, we propose to reduce the order of the approximation (r is replaced by $r-1$) and try again. Note that if we drop order to $r = 1$ (so $P \in \mathbb{P}_0$ is a constant), we will have a 1×1 nonsingular matrix $A^T A$ with condition number 1, and so our algorithm will terminate.

In practice, one generally evaluates the stencil polynomial on some *target cell* $E_0 \in S$. In this case, it is important to add a constraint to the least squares solution [4]. We require that P satisfy the "mass" constraint exactly on $E_0 \in S$. We can write this as

$$\frac{1}{|E_0|} \int_{E_0} P(\mathbf{x})\, d\mathbf{x} = \bar{u}_{E_0} \quad \Longleftrightarrow \quad \beta \cdot \mathbf{c} = \bar{u}_{E_0}, \tag{11}$$

where β^T is the row of A corresponding to cell E_0. Let $\mathbf{e}_0$ denote the standard unit vector that is one for cell E_0 and zero otherwise. Then the constraint is

$$\beta \cdot \mathbf{c} = \bar{u}_{E_0} = \mathbf{e}_0 \cdot \mathbf{u}, \quad \beta = A^T \mathbf{e}_0. \tag{12}$$

The constrained problem is

$$\begin{bmatrix} A^T A & \beta \\ \beta^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{c} \\ \lambda \end{pmatrix} = \begin{pmatrix} A^T \mathbf{u} \\ \bar{u}_{E_0} \end{pmatrix}. \tag{13}$$

This is easily solved in terms of the unconstrained least squares solution of (7),

$$\hat{\mathbf{c}} = B\mathbf{u}, \quad B = (A^T A)^{-1} A^T = V \Sigma^{-2} V^T A^T = V \Sigma^{-1} U^T, \tag{14}$$

and the vector

$$\alpha = (A^T A)^{-1} \beta = B\mathbf{e}_0. \tag{15}$$

The result is

$$\mathbf{c} = \hat{\mathbf{c}} + \frac{1}{\beta \cdot \alpha} \big(\bar{u}_{E_0} - \beta \cdot \hat{\mathbf{c}}\big) \alpha. \tag{16}$$

This can be rearranged into

$$\mathbf{c} = D_0 \mathbf{u}, \quad D_0 = B + \frac{\alpha}{\beta \cdot \alpha} \big(\mathbf{e}_0^T - \beta^T B\big). \tag{17}$$

2.2. Details of determining the stencil polynomials

We recap the procedure of the previous section. There is a stencil polynomial $P = P^r_{S,E_0}$ for each stencil S and each target cell E_0 that will be used, and each is defined by its degree $r-1$ and coefficients $\mathbf{c}$. Our goal is to find the set of all stencil polynomials defined over the entire mesh in the context of a time-stepping loop. We separate the steps that are linked only to the (fixed) computational mesh from those that also involve the time dependent solution data. We describe the algorithm that we use in our computer code and simply note that other choices could be made in the implementation.

Before beginning the time-stepping loop, we need to precompute quantities involving only the computational mesh. These quantities will not change during the time-stepping (unless the mesh changes within the loop, in which case these steps need to be repeated). We first compute the mass preservation matrix $\mathcal{A}$ for the entire mesh. That is, for cell E and monomial $\mathbf{x}^\alpha$, we compute

$$\mathcal{A}_{E,\alpha} = \frac{1}{|E|} \int_E \mathbf{x}^\alpha \, d\mathbf{x}. \tag{18}$$

Normally E is a polytope and can be divided into simplices. The integral can then be computed exactly using quadrature over each simplex.

For each mesh stencil S that we use, we fix a desired polynomial degree $r-1$. The mass preservation matrix $A = A^r_S$ in (7) is extracted from $\mathcal{A}$ according to the cells in the stencil S and degree $r-1$. This is not a simple process because the polynomial is expanded as in (5) for the given degree $r-1$. The Binomial Theorem needs to be employed to evaluate A^r_S, i.e.,

$$\begin{aligned} (A^r_S)_{E,\alpha} &= \frac{1}{|E|} \int_E \left(\frac{\mathbf{x} - \mathbf{x}_S}{h_S} \right)^\alpha d\mathbf{x} \\ &= \frac{1}{|E| h_S^{|\alpha|}} \int_E \sum_{\beta \le \alpha} \binom{\alpha}{\beta} \mathbf{x}_S^{\alpha-\beta} \mathbf{x}^\beta \, d\mathbf{x} \\ &= \frac{1}{h_S^{|\alpha|}} \sum_{\beta \le \alpha} \binom{\alpha}{\beta} \mathbf{x}_S^{\alpha-\beta} \mathcal{A}_{E,\beta}. \end{aligned} \tag{19}$$

Once A^r_S is determined, its compact SVD, $A^r_S = U \Sigma V^T$ is computed. If the normal matrix $(A^r_S)^T (A^r_S)$ is singular or badly conditioned, we reject the polynomial, replace r by $r-1$, and try again, starting from extracting a new A^r_S from $\mathcal{A}$ (using the reduced r). The process will terminate with $r \geq 1$, which is the index representing the highest possible polynomial degree that provides accurate numerical approximation on the stencil.

For each stencil S, we store in computer memory A^r_S and the matrix $B^r_S = V \Sigma^{-2} V^T (A^r_S)^T = V \Sigma^{-1} U^T$. These matrices depend only on the stencil S and r, but not on whatever target cell we choose later. Note that the vector β in (12) is merely a row of A^r_S and α in (15) is a column of B^r_S.

Depending on what strategy we choose later, at this stage we can also compute and store the matrix $D_0 = D^r_{S,E_0}$ in (17) for all the required $\mathbf{e}_0$, i.e., for all the target cells E_0 that will be used with the stencil S. At this point, we could clear the memory associated with $\mathcal{A}$.

We now enter the time-stepping loop and compute quantities that depend on the solution data, i.e., on $\mathbf{u}$. We can choose one of two strategies.

The first strategy is appropriate only for explicit time stepping. It reduces the amount of computer memory required when multiple target cells E_0 are used for each stencil, because we do not precompute the matrices D^r_{S,E_0} above. Within the time stepping

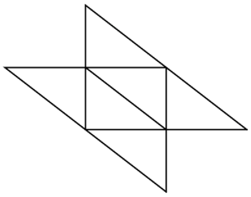


Fig. 1. A stencil of 6 cells that leads to a singular matrix when $r = 3$.

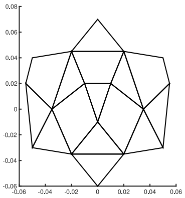


Fig. 2. A stencil of 15 cells that leads to a badly conditioned matrix when $r = 5$.

Table 1
Approximation of $2(1 + \cos(2\pi x))\exp(xy - y)$ on refined stencils congruent to the one depicted in Fig. 2 using $r = 5$ and $r = 4$. The target cell is the center one, and the average L^1-error is computed on this cell.

Refinement	$r = 5, \mathbb{P}_4$		$r = 4, \mathbb{P}_3$	
level	L^1-error	Rate	L^1-error	Rate
0	4.124e+07	–	5.697e–06	–
1	1.291e+06	5.00	3.550e–07	4.00
2	4.036e+04	5.00	2.218e–08	4.00
3	1.261e+03	5.00	1.386e–09	4.00
4	3.947e+01	5.00	8.658e–11	4.00

loop and for the given stencil S, we use the precomputed B^r_S to compute the vector $\hat{\mathbf{c}} = B^r_S \mathbf{u}_S$, where $\mathbf{u}_S$ is the restriction of the solution averages in $\mathbf{u}$ to the cells of S. For each target cell, then, (16) gives the vector of coefficients of the polynomial P^r_{S,E_0} on S for target cell E_0.

The second strategy is appropriate for either explicit or implicit time stepping, and it uses the precomputed matrices D^r_{S,E_0}. Within the time stepping loop and for the given stencil S and target cell E_0, we compute the vector of coefficients of the polynomial P^r_{S,E_0} using (17) (i.e., $\mathbf{c} = D^r_{S,E_0} \mathbf{u}_S$). Note that D^r_{S,E_0} is the derivative of the stencil polynomial with respect to the cell averages. This derivative is needed in Newton's method to solve the nonlinear problem that arises in implicit methods.

2.3. Examples of bad stencils

It is not difficult to find poor stencils for a given value of r. We depict in Fig. 1 a stencil of six cells that leads to a singular matrix A when $r = 3$ (note that $\dim \mathbb{P}_2 = 6$). The 4×4 square stencil of 16 cells with side length 0.25 does not support accurate approximation by a polynomial of degree 4 ($r = 5$), despite the fact that $\dim \mathbb{P}_4 = 15$. The extreme singular values are $s_1 = 2.656$ and $s_{16} = 4.061\text{e}{-17}$, and the condition number $(s_1/s_{15})^2$ is 3.292e+33. In Fig. 2, we depict a stencil of 15 cells that also has a large condition number when $r = 5$. The extreme singular values are in fact $s_1 = 4.174$ and $s_{15} = 3.391\text{e}{-17}$, so the condition number $(s_1/s_{15})^2$ is 1.515e+34. From the geometric layout of the stencils, it is not obvious to the authors that any of them would perform so poorly.

When we drop the index of the stencil in Fig. 2 to $r = 4$, the extreme singular values are $s_1 = 4.106$ and $s_{10} = 0.114$, so the condition number is only $(s_1/s_{10})^2 = 1303.97$. We report in Table 1 a study of the accuracy and convergence rate of the approximations using $r = 5$ and $r = 4$. The diameter of the initial stencil is about 0.14, and each refinement level shrinks the mesh by a factor of 2. The average L^1-error is computed on the target cell, which is the center one. We see that the 4th degree polynomial is unacceptably inaccurate, even though it shows 5th order convergence (as the theory predicts [10]). The 3rd order polynomial shows good accuracy and 4th order convergence.

Consider now $r = 3$ and the stencil of 6 square cells given by the meshpoints $(0.1i, 0.1j)$ for $i = 0, 1, 2, 3$ and $j = 0, 1, 2$. Clearly the matrix $A^T A$ is singular. However, if we perturb the point $(0.2, 0.1)$ to $(0.2, 0.1 + \theta)$ for small $\theta > 0$, the problem is no longer singular. To the naked eye, however, there is little difference in the stencils for $\theta = 1\text{e}{-2}$, $1\text{e}{-4}$, and $1\text{e}{-8}$, so we would not expect any of these stencils to perform acceptably.

Table 2
Approximation of $2(1+\cos(2\pi x))\exp(xy-y)$ for $r=3$ on refined stencils given by the initial meshpoints $(0.1i, 0.1j)$ for $i=0,1,2,3$ and $j=0,1,2$, $(i,j)\neq(2,1)$ and $(0.2, 0.1+\theta)$. The condition number κ is given, as well as the average L^1-error computed on the lower middle (target) cell centered at $(0.15, 0.05)$.

Refinement	θ = 1e−2 κ = 3.18312e+07		θ = 1e−4 κ = 3.13172e+11		θ = 1e−8 κ = 3.13121e+19	
level	L^1-error	Rate	L^1-error	Rate	L^1-error	Rate
0	1.455e−00	–	1.452e+02	–	1.452e+06	–
1	4.803e−01	1.60	4.799e+01	1.60	4.799e+05	1.60
2	7.745e−02	2.63	7.740e+00	2.63	7.743e+04	2.63
3	1.064e−02	2.86	1.064e+00	2.86	1.065e+04	2.86
4	1.385e−03	2.94	1.384e−01	2.94	1.378e+03	2.95

We show in Table 2 a study of the condition number and approximation rates of these stencils for the aforementioned θ. We see third order convergence for all stencils, but only the stencil using $\theta = 0.01$ could be considered to give an accurate stencil polynomial. This stencil is also the only one with condition number meeting our requirement $\kappa <$ 1e+8.

2.4. Stencil polynomial smoothness indicators

The measure of smoothness of $u(\mathbf{x})$ on the stencil $S^{(r)} = \{E_1, E_2 \dots, E_K\}$ with respect to the target cell $E_0 \in S^{(r)}$ is defined by a *smoothness indicator* σ. The classic way to define σ is due to Jiang and Shu [6,12]. We use this smoothness indicator, for which constant polynomials are naturally assigned $\sigma = 0$.

Any smoothness indicator must satisfy that for some $D \geq 0$, as $h \to 0^+$,

$$\sigma = \begin{cases} Dh^2 + \mathcal{O}(h^3) & \text{if } u \text{ is smooth on the stencil,} \\ \Theta(1) & \text{if } u \text{ has a jump discontinuity on the stencil.} \end{cases} \tag{20}$$

Unfortunately, any reasonable definition of σ would give a value that is not necessarily bounded below when there is a jump discontinuity in u, i.e., we have merely $\sigma = \mathcal{O}(1)$ in this case. To obtain (20) rigorously, we would need to make an assumption, for example, that the discontinuity is bounded away from the boundaries of the cells [15]. Fortunately, this issue does not cause problems when solving conservation laws.

3. ML-WENO reconstructions

Our proposed selection of accurate stencil polynomials leads to a possible reduction in the polynomial degree. This would be unacceptable for standard WENO and WENO-AO reconstructions, both of which require exactly two levels of approximation. We now develop a new multilevel WENO (ML-WENO) reconstruction which will be able to combine stencil polynomials of any order, including constant polynomials.

3.1. The reconstruction

Given a target cell $E_0 \in \mathcal{T}_h$, select a nonempty collection of stencils $S_\ell^{(r_\ell)} \ni E_0$, $\ell = 0, 1, \dots, L$. Each stencil has its polynomial $P_\ell \in \mathbb{P}_{r_\ell - 1}$ approximating u, as in (8). Let the linear weights $\omega_\ell > 0$ be chosen, one for each stencil. These weights can be chosen arbitrarily, and they need not sum to one. The reconstruction of u on E_0 for the collection of stencils is then

$$R(\mathbf{x}) = \sum_\ell \tilde{\omega}_\ell \, P_\ell(\mathbf{x}), \quad \mathbf{x} \in E_0, \tag{21}$$

where $\tilde{\omega}_\ell$ is the ℓ-th nonlinear weight, which we now define (these nonlinear weights will be positive and sum to one).

We begin by choosing parameters $\epsilon > 0$ and $s \geq 1/2$ (for example, take ϵ =1e−2 or 1e−4 and $s = 1$). We then scale each linear weight to define

$$\hat{\omega}_\ell = \frac{\omega_\ell}{(\sigma_\ell + \epsilon h_0^2)^{s r_\ell + \eta_\ell}}, \tag{22}$$

where $\eta_\ell = \eta(r_\ell)$ must be chosen as a nonnegative, nondecreasing function of r_ℓ. We normally take $\eta_\ell = 0$ for all ℓ; however in some cases, nonvanishing η_ℓ is useful to prevent overly biasing to the stencils for constant and linear polynomials when the solution is smooth, so, e.g., one might take

$$\eta(r_\ell) = \eta_\ell = \begin{cases} 1, & \text{if } r_\ell = 1, \\ 3, & \text{if } r_\ell = 2, \\ 4, & \text{if } r_\ell \geq 3. \end{cases} \tag{23}$$

We finally define the nonlinear weights through normalization as

$$\tilde{\omega}_\ell = \frac{\hat{\omega}_\ell}{\sum_k \hat{\omega}_k}. \tag{24}$$

We reiterate that it is often sufficient to take $\eta_\ell = 0$ rather than using (23) in the weighting procedure. When we need to make the distinction clear, we will denote the reconstruction as ML1-WENO when $\eta_\ell = 0$, and as ML2-WENO when we use (23). The effect of using nonzero η_ℓ can be understood as a preprocessing step, in which we scale each linear weight according to the formula

$$\check{\omega}_\ell = \frac{\omega_\ell}{(\sigma_\ell + \epsilon h_0^2)^{\eta_\ell}} = \begin{cases} \Theta(h^{-2\eta_\ell}) & \text{if } u \text{ is smooth on the stencil,} \\ \Theta(1) & \text{if } u \text{ has a jump on the stencil.} \end{cases} \tag{25}$$

That is, we de-emphasize the constant and linear weights by a power of h when the solution is smooth. Our ML2-WENO procedure is somewhat like what Semplice and Visconti do in their adaptive order reconstruction [27] when they directly scale the low order polynomial weight(s) by a power of h.

Although not necessary, we could normalize these weights $\check{\omega}_\ell$ to define the intermediate nonlinear weights as

$$\bar{\omega}_\ell = \frac{\check{\omega}_\ell}{\sum_k \check{\omega}_k}. \tag{26}$$

This is essentially equivalent to the usual WENO-AO weighting procedure, and results in biasing the weights to avoid stencils containing discontinuities, without overly biasing to the stencils for constant and linear polynomials. We would then finish the reconstruction by applying ML1-WENO (i.e., with $\eta_\ell = 0\ \forall \ell$) starting from the intermediate nonlinear weights (26) in place of the linear ones. The result is exactly the same as (24) described above.

We remark that instead of using ML2-WENO, it is often simpler to prescribe smaller linear weights to the constant and linear stencil polynomials. In fact, we do this in applications to scalar conservation laws (see Section 5).

3.2. The ML-WENO reconstruction error

In this section we consider the accuracy of the ML-WENO reconstruction under the assumption that either u is smooth or that u has a jump discontinuity on some, but not all, of the stencils. Among other things, this means that we view u as being smooth on the target cell E_0. Because we constrain the local averages to be correct on E_0, $P_\ell = \bar{u}_{E_0}$ when $r_\ell = 1$ regardless of the behavior of u on the rest of its stencil. We therefore view the stencil as $\{E_0\}$ when $r_\ell = 1$.

To analyze the error in the reconstruction (21), we need to consider the many possibilities of where u might be discontinuous. We find it convenient to divide the stencil indices into disjoint subsets as follows. For $r \geq 2$, let

$$\text{Smooth}(r) = \{\ell : r_\ell = r \text{ and } u \text{ is smooth on } S_\ell^{(r_\ell)}\}, \tag{27}$$

$$\text{Jump}(r) = \{\ell : r_\ell = r \text{ and } u \text{ has a jump discontinuity on } S_\ell^{(r_\ell)}\}, \tag{28}$$

and let Smooth(1) $= \{\ell : r_\ell = 1\}$, Jump(1) $= \emptyset$. We tacitly assume that there are no other possibilities, so

$$\bigcup_r \big(\text{Smooth}(r) \cup \text{Jump}(r)\big) = \{0, 1, \dots, L\}.$$

We also tacitly assume that $\bigcup_r \text{Smooth}(r)$ is not empty.

We shall prove that the reconstruction is accurate on the target cell E_0 to order

$$r_{\max} = \max\big\{r : \text{Smooth}(r) \neq \emptyset\big\} \geq 1. \tag{29}$$

This is the index of the best (maximal r) stencil approximation that avoids the discontinuities in u. We further define the sets of indices

$$\text{SM} = \text{Smooth}(r_{\max}), \quad \text{SL} = \bigcup_{r<r_{\max}} \text{Smooth}(r), \quad \text{Jmp} = \bigcup_r \text{Jump}(r). \tag{30}$$

3.2.1. The effect of the scaling

We begin by estimating the effect of the scaling in (22).

Lemma 3.1. *If $h_0 = \Theta(h)$, (20) holds, and for some power $t > 0$,*

$$\hat{\omega}_\ell = \frac{\omega_\ell}{(\sigma_\ell + \epsilon h_0^2)^t}, \tag{31}$$

then

$$\hat{\omega}_\ell = \begin{cases} \dfrac{\omega_\ell}{(D+\epsilon)^t} h_0^{-2t}\big(1 + \mathcal{O}(h)\big) & \text{if } \ell \in SM \cup SL, \\ \Theta(1) & \text{if } \ell \in Jmp. \end{cases} \tag{32}$$

We interject that in the proofs of this section, we will make use of well-known results from asymptotic theory (see, e.g., [32]). Specifically, when $r \ge t$, $a > 0$, and $b > 0$,

$$\frac{a}{bh^t + \mathcal{O}(h^r)} = \frac{ah^{-t}}{b}\big(1 + \mathcal{O}(h^{r-t})\big), \tag{33}$$

$$\frac{a}{bh^{-r} + \mathcal{O}(h^{-t})} = \frac{ah^{r}}{b}\big(1 + \mathcal{O}(h^{r-t})\big). \tag{34}$$

Proof. If $\ell \in \mathrm{SM} \cup \mathrm{SL}$, then using (20) and (33), we see that

$$\begin{aligned}\hat{\omega}_\ell &= \frac{\omega_\ell}{(\sigma_\ell + \epsilon h_0^2)^t} = \frac{\omega_\ell}{((D+\epsilon)h_0^2 + \mathcal{O}(h^3))^t}\\ &= \omega_\ell\left(\frac{h_0^{-2}}{D+\epsilon}\big(1+\mathcal{O}(h)\big)\right)^t = \frac{\omega_\ell}{(D+\epsilon)^t}h_0^{-2t}\big(1+\mathcal{O}(h)\big)\end{aligned}$$

as claimed. If $\ell \in \mathrm{Jmp}$, then $r_\ell > 1$ and $\sigma_\ell = \Theta(1)$, so $\hat{\omega}_\ell = \frac{\omega_\ell}{(\sigma_\ell + \epsilon h_0^2)^t} = \Theta(1)$, and the proof is complete. □

3.2.2. The effect on the nonlinear weights

Lemma 3.2. *Let $r_{\max}$ be defined by (29), and let $\eta_{\max} = \eta(r_{\max})$. If $h_0 = \Theta(h)$ and (20) holds, then*

$$\tilde{\omega}_\ell = \begin{cases} \dfrac{\omega_\ell}{\sum\limits_{k\in SM}\omega_k}\big(1+\mathcal{O}(h)\big) & \text{if } \ell \in SM,\\ \Theta(h^{2[s(r_{\max}-r_\ell)+(\eta_{\max}-\eta_\ell)]}) & \text{if } \ell \in SL,\\ \Theta(h^{2(sr_{\max}+\eta_{\max})}) & \text{if } \ell \in Jmp.\end{cases} \tag{35}$$

We remark that for $\ell \in \mathrm{SM}$, the relative proportion of the linear weights is maintained by the nonlinear weights.

Proof. Using Lemma 3.1 with $t = sr_\ell + \eta_\ell$, the denominator of (24) is

$$\begin{aligned}\sum_\ell \hat{\omega}_\ell &= \sum_{\ell\in\mathrm{SM}}\hat{\omega}_\ell + \sum_{\ell\in\mathrm{SL}}\hat{\omega}_\ell + \sum_{\ell\in\mathrm{Jmp}}\hat{\omega}_\ell\\ &= \sum_{\ell\in\mathrm{SM}}\frac{\omega_\ell}{(D+\epsilon)^{sr_{\max}+\eta_{\max}}}h_0^{-2(sr_{\max}+\eta_{\max})}\big(1+\mathcal{O}(h)\big)\\ &\quad + \sum_{\ell\in\mathrm{SL}}\Theta(h^{-2(sr_\ell+\eta_\ell)}) + \sum_{\ell\in\mathrm{Jmp}}\Theta(1)\\ &= \sum_{\ell\in\mathrm{SM}}\frac{\omega_\ell}{(D+\epsilon)^{sr_{\max}+\eta_{\max}}}h_0^{-2(sr_{\max}+\eta_{\max})} + \mathcal{O}(h^{1-2(sr_{\max}+\eta_{\max})}),\end{aligned}$$

since when $\ell \in \mathrm{SL}$, $r_\ell \le r_{\max} - 1$, $\eta_\ell \le \eta_{\max}$, and $s \ge 1/2$. Using this result, (34) implies that

$$\tilde{\omega}_\ell = \frac{\hat{\omega}_\ell}{\sum\limits_k \hat{\omega}_k} = \frac{\hat{\omega}_\ell}{\sum\limits_{k\in\mathrm{SM}}\dfrac{\omega_k}{(D+\epsilon)^{sr_{\max}+\eta_{\max}}}}h_0^{2(sr_{\max}+\eta_{\max})}\big(1+\mathcal{O}(h)\big).$$

For any case of ℓ, Lemma 3.1 gives the order of $\hat{\omega}_\ell$, and the result follows. □

3.2.3. A bound for the overall reconstruction error

Theorem 3.3. *Let $R(\mathbf{x})$ be the reconstruction (21). If $h_0 = \Theta(h)$ and (8) and (20) hold, then there is a constant $C > 0$ such that for any $\mathbf{x} \in E_0$,*

$$|u(\mathbf{x}) - R(\mathbf{x})| \le Ch^{r_{\max}}, \tag{36}$$

where $r_{\max}$ is defined by (29).

Proof. Since the nonlinear weights are positive and sum to 1, we have for $\mathbf{x} \in E_0$ that the reconstruction error is

$$\begin{aligned}|u(\mathbf{x}) - R(\mathbf{x})| &= \left|\sum_\ell \tilde{\omega}_\ell\big(u(\mathbf{x}) - P_\ell(\mathbf{x})\big)\right|\\ &\le \sum_{\ell\in\mathrm{SM}}\tilde{\omega}_\ell|u(\mathbf{x}) - P_\ell(\mathbf{x})| + \sum_{\ell\in\mathrm{SL}}\tilde{\omega}_\ell|u(\mathbf{x}) - P_\ell(\mathbf{x})| + \sum_{\ell\in\mathrm{Jmp}}\tilde{\omega}_\ell|u(\mathbf{x}) - P_\ell(\mathbf{x})|.\end{aligned} \tag{37}$$

Combining this with Lemma 3.2 and the polynomial approximation error (8), we obtain

$$|u(\mathbf{x}) - R(\mathbf{x})| \le \sum_{\ell\in\mathrm{SM}}\Theta(1)\,\mathcal{O}(h^{r_{\max}}) + \sum_{\ell\in\mathrm{SL}}\Theta(h^{2[s(r_{\max}-r_\ell)+(\eta_{\max}-\eta_\ell)]})\,\mathcal{O}(h^{r_\ell})$$

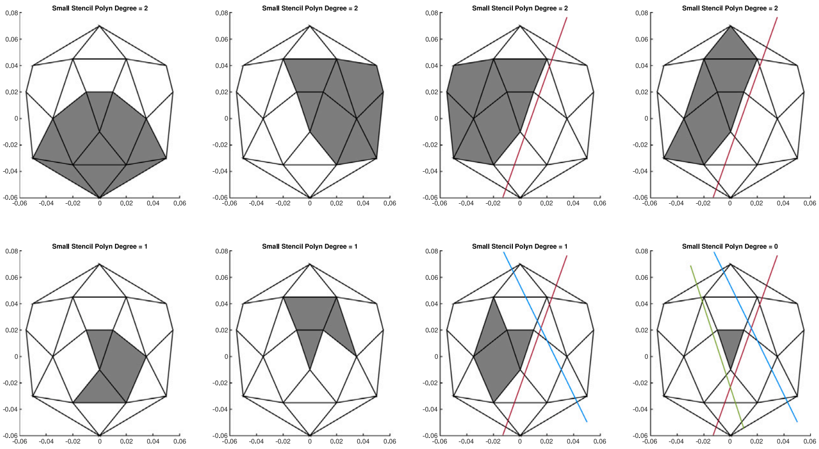

Small Stencil Polyn Degree = 2
Small Stencil Polyn Degree = 2
Small Stencil Polyn Degree = 2
Small Stencil Polyn Degree = 2
Small Stencil Polyn Degree = 1
Small Stencil Polyn Degree = 1
Small Stencil Polyn Degree = 1
Small Stencil Polyn Degree = 0
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.06 -0.04 -0.02 0 0.02 0.04 0.06

Fig. 3. Stencils used in the tests of the reconstructions. The large stencil has 19 cells, and the top row shows the four small stencils used in the two-level tests. The multilevel tests include also the three tiny stencils shown in the bottom row as well as the constant (single cell) stencil. The central cell is the target cell. The colored lines show where the test function may jump, but we only show the lines when they do *not* intersect the stencil.

$$
\begin{aligned}
&\qquad + \sum_{\ell \in \mathrm{Jmp}} \Theta(h^{2(sr_{\max}+\eta_{\max})})\, \mathcal{O}(1) \\
&= \mathcal{O}(h^{r_{\max}}) + \sum_{\ell \in \mathrm{SL}} \mathcal{O}(h^{r_{\max}+(2s-1)(r_{\max}-r_\ell)+2(\eta_{\max}-\eta_\ell)}) + \mathcal{O}(h^{2(sr_{\max}+\eta_{\max})}) \\
&= \mathcal{O}(h^{r_{\max}}),
\end{aligned}
$$

which is the desired result. □

The theoretical optimal choice of s would be $1/2$, but this would require in general the computation of square roots. In practice, $s = 1, 2, 3, \dots$ is more efficient; moreover, it gives additional bias in R to the best approximations P_ℓ for $\ell \in \mathrm{SM}$. We generally use $s = 1$.

3.3. Some remarks on alternative linear weight scalings

As the above theory shows, the key is to obtain nonlinearly scaled weights that satisfy (32), at least up to the value of the constant. There are many alternate scalings that satisfy this property. One might define

$$
\hat{\omega}_\ell = \frac{\omega_\ell}{\sigma_\ell^{sr_\ell+\eta_\ell} + \epsilon h_0^{2(sr_\ell+\eta_\ell)}}, \tag{38}
$$

which is very similar to the scaling we used.

For some constant $\bar{D} > 0$, one might define

$$
\hat{\omega}_\ell = \omega_\ell \left(\frac{\sigma_\ell + 2\bar{D}h_0}{\sigma_\ell + \bar{D}h_0^2} \right)^{sr_\ell+\eta_\ell}. \tag{39}
$$

For this form of the scaling, the theoretical optimal choice of s is $s = 1$ (rather than $s = 1/2$), and additional bias in R to the best approximations P_ℓ for $\ell \in \mathrm{SM}$ is accomplished by taking a large $\bar{D}$ (say 10). We do not find this choice to work well when solving conservation laws.

4. Numerical tests of the ML-WENO reconstruction

In this section, we perform numerical tests of the performance of ML-WENO reconstructions. The test function to be reconstructed is $u(x, y) = 2(1 + \cos(2\pi x))e^{xy-y}$, with possibly the addition of a unit jump term. Two sets of tests are conducted. The first set uses only two-levels of reconstructions (combining one degree 4 and several quadratic polynomials), so we can compare the results to the usual WENO-AO (5,3) reconstruction. The second set uses four levels. In all the tests, we take $\epsilon = 1\mathrm{e}{-2}$ and $s = 1$.

An initial large stencil of 19 cells is chosen along with several smaller stencils, as depicted in Fig. 3. The cells are fairly coarse with a diameter of about $h = 0.04$. The colored lines show where the test function may jump. The stencils are refined by contraction about the origin (by a factor of two for each successive level). The position of a jump is similarly scaled, so that the jump position appears fixed as we refine.

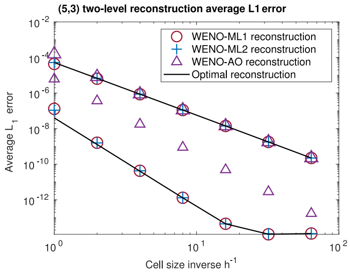


Fig. 4. Two-level reconstruction convergence results for the smooth (bottom set of data) and discontinuous (top set of data) test function. The discrete L^1-error versus 2 to the power of the refinement level is given on a log–log plot.

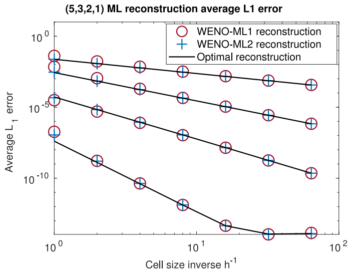


Fig. 5. Multilevel reconstruction convergence results in the four scenarios. The discrete L^1-error versus 2 to the power of the refinement level is given on a log–log plot.

4.1. Two-level reconstructions

We begin with a test of traditional two-level reconstruction comparing all three methods: ML1-WENO(5,3), ML2-WENO(5,3), and WENO-AO(5,3) using equal linear weights for each stencil. As shown in the top row of Fig. 3, there is one large and four small stencils that support accurate approximation by polynomials of degree 4 (i.e., $r_{\text{large}} = 5$) and 2 (i.e., $r_{\text{small}} = 3$). The number of cells in the five stencils are 19, 9, 7, 7, and 6, respectively. A single unit jump may be added to the solution, as depicted in the upper two right stencils, which do *not* see the jump. The jump cuts the other three stencils.

We display the convergence results for the reconstructions in Fig. 4. We see fifth order convergence when function u is smooth (bottom set of data), at least until rounding error degrades the results. The solid line shows the (optimal) convergence order of the large stencil polynomial. Clearly the two ML-WENO methods bias almost completely to this stencil polynomial, at least on fine meshes, while the WENO-AO does so only approximately.

When the jump is added to u, we see third order convergence in Fig. 4 (top set of data). Here, the solid line shows the (essentially optimal) convergence order of the average of the two stencil polynomials not crossing the jump. All reconstructions bias almost perfectly to these two stencil polynomials.

4.2. Multilevel reconstruction

We now turn to the second set of tests which uses all four levels of stencil polynomials. As shown in Fig. 3, to the large and small stencils, we have added three tiny stencils of three cells each that support accurate approximation by polynomials of degree 1 (i.e., $r_{\text{tiny}} = 2$) and we add the single target cell stencil for approximation of degree 0 (i.e., $r_{\text{constant}} = 1$). We also use linear weights equal to about 0.125 for each stencil except the constant, which has linear weight of about 0.125e−4.

We may add unit jumps to the solution, as depicted in the four right stencils of Fig. 3. In the following scenarios, we would expect to see the convergence rates:

- 1 with the blue and green jumps (only the constant stencil);
- 2 with the blue jump (one tiny stencil, but avoid the constant stencil);
- 3 with the red jump (two small stencils, but avoid one tiny and constant stencil);
- 5 with no jump (avoid all the small, tiny, and constant stencils).

Except for the first case, all the other cases have suitable stencils giving good but nonoptimal convergence. We would denote our multilevel reconstructions as ML1-WENO(5,3,2,1) and ML2-WENO(5,3,2,1).

In Fig. 5, we present our results. The solid line shows the (essentially optimal) convergence order of (the average of) the stencil polynomials not crossing the jump. We indeed see the expected convergence rates, at least until rounding error degrades the convergence. It is worth noting that the coarser meshes show a bit more error than fine meshes.

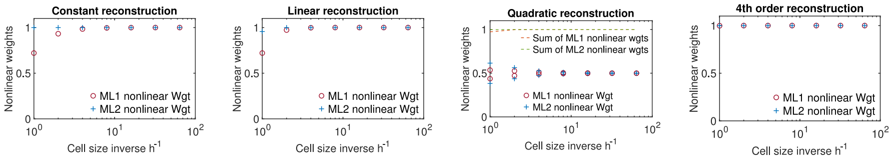


Fig. 6. Nonlinear weights of the dominant stencil(s) in the four scenarios versus the base 10 log of 2 to the power of the refinement level. The linear weights are all about 0.125, except for the constant stencil, which has linear about weight 0.125e−4.

In Fig. 6, we show nonlinear weights for the dominant stencil polynomial(s) in the four scenarios. The nonlinear weights are close to one, except for the quadratic case, which has two dominant stencil polynomials with nonlinear weights summing almost to one. Clearly, the correct stencils are being selected by the nonlinear weights, although the coarser meshes do so only approximately compared to the finer meshes. In the first and second scenarios with low order dominant stencil polynomials, ML2-WENO shows a bit better behavior than ML1-WENO on coarse meshes. The final two scenarios for which the dominant stencil polynomial is a quadratic or greater, ML2-WENO does not improve the overall accuracy much, since ML1-WENO is already sufficiently accurate.

5. Application to scalar conservation laws

We apply the ML-WENO reconstruction to solve the conservation law

$$u_t(\mathbf{x},t) + \nabla\cdot\mathbf{f}(u;x,t) = 0, \quad \mathbf{x}\in\Omega,\ t>0, \tag{40}$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \qquad \mathbf{x}\in\Omega, \tag{41}$$

where the vector flux function $\mathbf{f}(u;x,t)$ and initial condition $u_0(\mathbf{x})$ are given. Let ν denote the outer unit normal vector to $\partial\Omega$. We can specify an inflow Dirichlet boundary condition

$$u(\mathbf{x},t) = u_D(\mathbf{x},t), \quad \mathbf{x}\in\partial\Omega,\ t>0,\ \text{when } \mathbf{f}(u)\cdot\nu<0, \tag{42}$$

for some specified $u_D(\mathbf{x},t)$, or an inflow flux boundary condition

$$\mathbf{f}(u;\mathbf{x},t)\cdot\nu = f_B(\mathbf{x},t), \quad \mathbf{x}\in\partial\Omega,\ t>0,\ \text{when } \mathbf{f}(u)\cdot\nu<0, \tag{43}$$

for some specified flux $f_B(\mathbf{x},t)$, or a periodic boundary condition (if Ω is suitable).

The semidiscrete finite volume approximation is

$$\bar{u}_{E,t} + \frac{1}{|E|}\int_{\partial E}\hat{f}(u^-,u^+)\,ds(\mathbf{x}) = 0, \tag{44}$$

where u^- and u^+ are left and right limits of the approximate solution at the interface ∂E, given by our ML-WENO reconstruction of the cell averages $\bar{u}_E$. We incorporated a numerical flux $\hat{f}$, such as the Lax–Friedrichs one,

$$\hat{f}(u^-,u^+;\mathbf{x},t) = \frac{1}{2}\left[(\mathbf{f}(u^-;\mathbf{x},t)+\mathbf{f}(u^+;\mathbf{x},t))\cdot\nu_E - \alpha_{\mathrm{LF}}(u^+-u^-)\right], \tag{45}$$

where ν_E is the outer unit normal vector to ∂E and

$$\alpha_{\mathrm{LF}} = \max_u \left\|\partial\mathbf{f}/\partial u\right\| \tag{46}$$

is the maximum wave speed.

Time is discretized at the time levels $0 = t^0 < t^1 < t^2 < \cdots$ and a Runge–Kutta time integrator is employed. In the sequel, we compute $\bar{u}_E^n \approx \bar{u}_E(t^n)$ for each $n > 0$.

5.1. Features of the numerical tests

In the rest of Section 5, we present some numerical results in two space dimensions to illustrate the performance of ML-WENO in solving conservation laws. For simplicity, we write $\mathbf{x} = (x_1, x_2) = (x, y)$. In all cases, unless otherwise noted, the reconstructions (21) use $\epsilon = 0.01$, $s = 1$, and the (unnormalized) linear weights 1 for all stencils, except 1e−4 for constant stencils.

There are many ways to define mesh stencils. In this paper, we take stencils defined by rings (see [5,7,9,10]). Given a cell E, $S_0 = \{E\}$ is its ring 0 stencil. If to S_0 we add every cell sharing an edge with S_0, we have $S_{1/2}$, its 1/2-ring stencil (which is E plus its von Neumann neighborhood). If we add to S_0 every cell that shares at least one vertex with S_0, we have S_1, the full ring 1 stencil (E plus its Moore neighborhood). We may continue this process from S_1 to define the ring 3/2 and 2 stencils, and so on to define rings greater than 2. As an illustration, on a rectangular mesh with rectangular indexing, these stencils are

$$S_0 = \{E_{i,j}\}, \qquad S_{1/2} = \{E_{i,j}, E_{i-1,j}, E_{i+1,j}, E_{i,j-1}, E_{i,j+1}\},$$
$$S_1 = S_{1/2} \cup \{E_{i-1,j-1}, E_{i+1,j-1}, E_{i-1,j+1}, E_{i+1,j+1}\},$$

Table 3
Example 5.2.1, the smooth linear convergence test. ML1-WENO errors and convergence orders (using σ_{JS}) and weights 1, 1, 1e−4.

		L^1-norm		L^∞-norm	
Scheme	Mesh	Error	Order	Error	Order
ML1-WENO(5)	10 × 10	7.35e−03	–	2.07e−02	–
	20 × 20	2.80e−04	4.71	7.90e−04	4.71
	40 × 40	9.09e−06	4.95	2.58e−05	4.94
	80 × 80	2.87e−07	4.99	8.13e−07	4.99
	160 × 160	8.99e−09	5.00	2.55e−08	4.99
ML1-WENO(5,3)	10 × 10	2.00e−02	–	5.51e−02	–
	20 × 20	2.48e−04	6.33	6.59e−04	6.39
	40 × 40	8.77e−06	4.82	2.57e−05	4.68
	80 × 80	2.84e−07	4.95	8.13e−07	4.98
	160 × 160	8.96e−09	4.99	2.55e−08	4.99
ML1-WENO(5,3,1)	10 × 10	1.98e−02	–	5.27e−02	–
	20 × 20	2.48e−04	6.32	6.59e−04	6.32
	40 × 40	8.77e−06	4.82	2.57e−05	4.68
	80 × 80	2.84e−07	4.95	8.13e−07	4.98
	160 × 160	8.96e−09	4.99	2.55e−08	4.99
ML1-WENO(5,3,2)	10 × 10	1.98e−02	–	5.06e−02	–
	20 × 20	2.48e−04	6.32	6.59e−04	6.26
	40 × 40	8.77e−06	4.82	2.57e−05	4.68
	80 × 80	2.84e−07	4.95	8.13e−07	4.98
	160 × 160	8.96e−09	4.99	2.55e−08	4.99
ML1-WENO(5,2)	10 × 10	7.53e−02	–	3.01e−01	–
	20 × 20	2.26e−03	5.06	5.39e−02	2.48
	40 × 40	9.09e−06	7.96	2.63e−05	11.00
	80 × 80	2.87e−07	4.99	8.13e−07	5.02
	160 × 160	8.99e−09	5.00	2.55e−08	4.99
ML1-WENO(5,2,1)	10 × 10	7.54e−02	–	3.02e−01	–
	20 × 20	2.26e−03	5.06	5.39e−02	2.49
	40 × 40	9.09e−06	7.96	2.63e−05	11.00
	80 × 80	2.87e−07	4.99	8.13e−07	5.02
	160 × 160	8.99e−09	5.00	2.55e−08	4.99

with 1, 5, and 9 cells, respectively. Note that S_1 is a 3×3 array of cells, and that $S_{3/2}$ would have 21 cells.

We can generalize this cell-based construction by taking a different starting point. An edge-based stencil about edge $e = E_- \cap E_+$ would use $S_0 = \{E_-, E_+\}$. The vertex-based stencil about vertex $\mathbf{x}$ would start with $S_0 = \{E \in \mathcal{T}_h : \mathbf{x} \text{ is a vertex of } E\}$. In particular, the vertex-based stencil on a rectangular mesh about vertex $\mathbf{x}_{i+1/2,j+1/2}$ would be $S_0 = \{E_{i,j}, E_{i+1,j}, E_{i,j+1}, E_{i+1,j+1}\}$, which is a 2×2 array of cells suitable for linear polynomial approximation.

Unless otherwise stated, in all our computational tests we use the following collections of stencils. The collection of *large stencils* is the set of cell-based ring stencils about each $E \in \mathcal{T}_h$. We also use one or more collections of *small stencils*. Each such collection will consist of either cell-, edge-, or vertex-based ring stencils, one for each cell, edge, or vertex in the mesh. For a given target cell $E_0 \in \mathcal{T}_h$, the reconstruction uses the *single* large stencil about E_0, as well as the *union of all* small stencils that contain E_0. By reusing the small stencils, we reduce the overall workload, since the SVD factorization (9) is computed only for the stencil, and not for the target cell.

5.2. Numerical results for the linear equation

Let $\mathbf{f}(u) = \mathbf{a}u$ for some vector $\mathbf{a}(\mathbf{x})$ and $\Omega = (0,1)^2$. We impose periodic boundary conditions for these tests.

5.2.1. Smooth convergence rate

Take $\mathbf{a} = (1,1)$ and apply the initial condition $u_0(\mathbf{x}) = \sin^2(\pi x)\,\sin^2(\pi y)$, which is periodic. The true solution is $u(\mathbf{x},t) = u_0(\mathbf{x} - \mathbf{a}t)$. We use uniform square meshes for these tests and various stencils. A fifth order Runge–Kutta method due to Ruuth and Spiteri [33] is used, and the timestep about CFL = 0.7.

We show the results of our convergence study in Table 3. We show the errors measured in both the discrete L^1- and L^∞-norms. (The discrete norms compare the cell averages.) In all cases the large stencil is the ring 1.5 stencil of the target cell. It has 21 cells, and supports approximation to order $r = 5$ (degree 4). The ML1-WENO(5) results can be considered as the base case. They use only the large stencil polynomials centered around each cell and *not* a weighted reconstruction. The results show a clean fifth order convergence rate, as expected.

The multilevel results use small stencils of degree 2 ($r = 3$) defined on stencils of 3×3 cells (cell-based ring 1 stencils), degree 1 ($r = 2$) defined on stencils of 2×2 cells (vertex-based ring 0 stencils), and degree 0 ($r = 1$) defined on a single cell (cell-based ring

0 stencils). The reconstructions are either two-level, and use equal linear weights for every stencil, or three-level, in which case the lowest level stencil polynomials have a relative linear weight of 1e−4 compared to the other stencils relative weights being 1.

The errors of ML1-WENO(5,3), ML1-WENO(5,3,1), and ML1-WENO(5,3,2) are nearly identical, and very similar to those of ML1-WENO(5). The multilevel reconstructions properly select the highest order, large stencil polynomials. The inclusion of quadratic stencil polynomials actually reduces the error very slightly on finer meshes. The three level results show no degradation from the two-level results. The lowest level, whether it is $r = 1$ or $r = 2$, does not pollute the results because its relative linear weight is 1e−4.

The errors of ML1-WENO(5,2) and ML1-WENO(5,2,1) are nearly the same, and similar to those of ML1-WENO(5) on finer meshes. The constant stencil polynomials have a relative linear weight of 1e−4, and so do not pollute the results for ML1-WENO(5,2,1). On the coarse meshes ($h = 1/10$ and $h = 1/20$), the inclusion of linear polynomials results in some degradation of the errors, because the smoothness indicator has some difficulty separating the fourth degree polynomials from the linear ones. As a consequence, the ML2-WENO reconstruction does not help matters here. However, we can compensate by taking a smaller relative linear weight for the linear stencil polynomials. If 1e−4 is used for both linear and constant polynomials, ML1-WENO(5,2,1) recovers the good accuracy of ML1-WENO(5) on all the meshes. If we take a more moderate relative weight of 1e−2 for the linear polynomials, we see respectable L^1 errors of 4.48e−02 for the 10 × 10 meshes and 2.80e−04 for 20 × 20 meshes, and L^∞-errors of 2.12e−01 and 1.18e−03, respectively.

To summarize, including linear and constant polynomials in the reconstruction on coarse meshes is somewhat delicate and must be compensated for by adjusting the linear weights. If that is done, the ML1-WENO reconstruction is fully capable of biasing the approximation to the large stencil polynomials and away from all the low order small stencil polynomials.

5.2.2. *Handling discontinuities*

For this and some of the other examples below, we use a polygonal mesh generated by a modification of the code PolyMesher [34] with two smoothing iterations (the modification was made to create periodic meshes). Appearing in Fig. 7, it is a Voronoi mesh with 18,858 vertices and 10,000 cells, with cells having up to 10 edges.

For each (target) mesh cell, we used the cell-based ring 1.5 large stencil and all overlapping small cell-based stencils of size 1 ring. The actual number of cells in each stencil varied depending on the local mesh. To give the reader a rough idea of the stencils used, a uniform mesh of hexagons would have a large ring 1.5 stencil of 19 cells (and it would agree with the ring 2 stencil). The small ring 1 stencils would each have 7 cells. The reconstructions would then use 1 large stencil polynomial with the target cell in the center, and 7 small stencil polynomials that overlap the target cell.

There are 10,000 large stencils, each with a polynomial having desired degree 4. Two of these stencils produced condition numbers in excess of 1e+8, and 85 other stencils were too small to support the desired polynomial degree 4, so our computer code automatically reduced these stencil polynomials to degree 3. There are also 10,000 small stencils with desired polynomial degree 2, but 196 of these stencils were too small and used polynomial degree 1. Therefore, the overall reconstruction technique is of type ML1-WENO(5,4,3,2), although most often over the domain it is ML1-WENO(5,3).

Following [35], we take $\mathbf{a}(x, y) = (0.5 - y, x - 0.5)$ (a rotating flow) and an initial condition consisting of a slotted disk, a cone, and a smooth hump:

$$\begin{aligned} r_0 = 0.15, \quad r_H &= \sqrt{(x-0.25)^2 + (y-0.50)^2}, \\ r_C = \sqrt{(x-0.50)^2 + (y-0.25)^2}, \quad r_D &= \sqrt{(x-0.50)^2 + (y-0.75)^2} \\ u_0(x, y) = (r_D <= r_0)\big[1 - (y < 0.75)(0.45 < x)(x < 0.55)\big] & \\ + (r_C <= r_0)(1 - r_C/r_0) + (r_H <= r_0)0.25(1 + \cos(\pi r_H/r_0)), & \end{aligned} \tag{47}$$

wherein we used the convention that a logical comparison that is true is 1 and false is 0. The projection of the initial condition onto the polygonal mesh appears in Fig. 8.

We computed the solution to the problem to time 2π using 1000 timesteps and the standard SSP3 Runge–Kutta time integrator. For comparison, results were also obtained for two WENO with Adaptive Order schemes [13,15,16], WENO-AO and WENOZ-AO, which use classic and Z-weighting [36], respectively. The test is mildly unfair to these schemes, because the stencils do not organize into two levels (i.e., they are not strictly WENO-AO(5,3) and WENOZ-AO(5,3)). Nevertheless, we can attempt to solve the problem with these schemes, since the stencils usually give (5,3)-type reconstructions.

Overall, the three schemes handle the contact discontinuities and steep fronts of this problem very well, as shown in Fig. 9. On the top row, we plot the computed solutions $\bar{u}$, which are the average values of the computed solution on each cell. On the bottom row, we plot the piecewise discontinuous reconstructed computed solution $R(\mathbf{x})$, which for $\mathbf{x} \in E \in \mathcal{T}_h$ is the reconstruction for target cell E.

Perhaps the reader can see that the sharpest solution was obtained using ML1-WENO. The L^1 and L^∞ discrete norms of the error are given in the second and third columns of

Table 4, which quantifies and confirms our observation. However, ML1-WENO exhibits significant undershoot and overshoot. We rectify this using flux corrected transport [37,38]. Implementation of this remarkable algorithm is independent of the particulars of the underlying numerical scheme, so there are no new ideas needed to combine flux correction with our scheme (or any other scheme). All that is required is the flux from a low order scheme that satisfies the maximum principle. We start from the trivially maximum principle preserving no-flux scheme (i.e., zero low order flux) and apply the flux correction algorithm to correct the “high order” SATh flux so that the scheme satisfies the maximum principle. We then iterate the process a few times by correcting

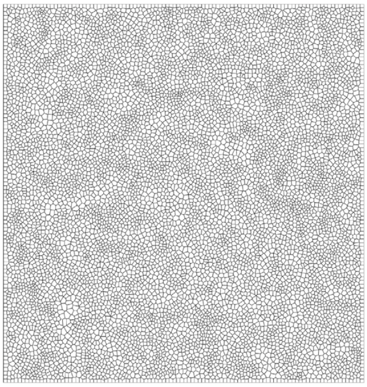


Fig. 7. A polygonal mesh of 10,000 cells.

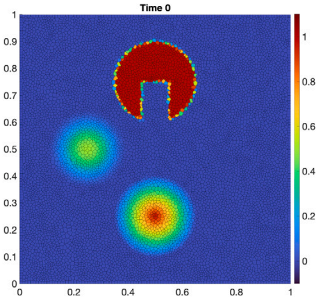
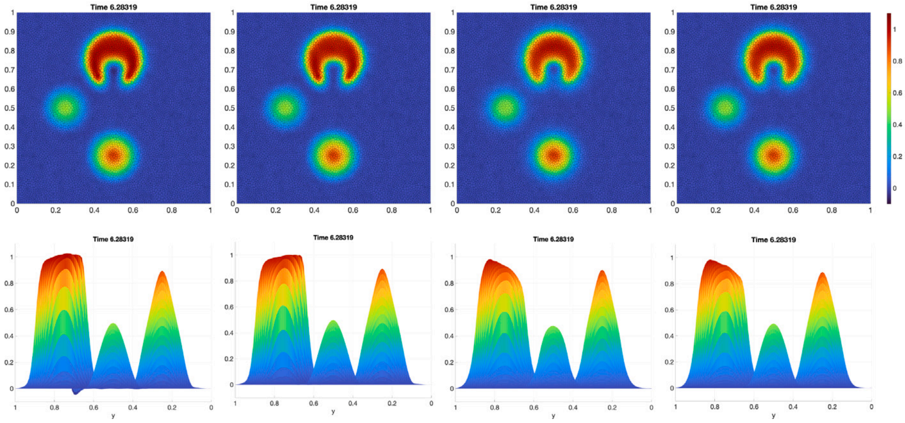
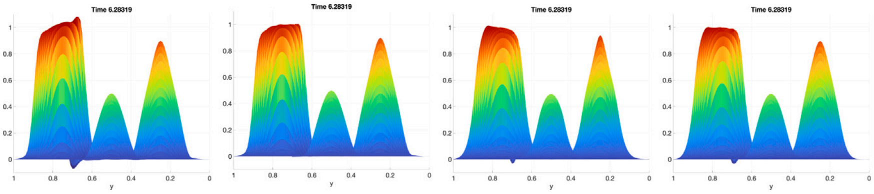


Fig. 8. Section 5.2.2. Initial condition projected onto the polygonal mesh of Fig. 7.



ML1-WENO ML1-WENO (FC) WENO-AO WENOZ-AO

Fig. 9. Section 5.2.2. The finite volume solution (top row) and the reconstructed solution (bottom row) at time 2π (one revolution) using the mesh from Fig. 7. From left to right are results for ML1-WENO, ML1-WENO (flux corrected), WENO-AO, and WENOZ-AO.



ML1-WENO ML1-WENO (FC) WENO-AO WENOZ-AO

Fig. 10. Section 5.2.2. The reconstructed solution at time 2π (one revolution) using a 100×100 square mesh. From left to right are results for ML1-WENO, ML1-WENO (flux corrected), WENO-AO, and WENOZ-AO.

Table 4
Example 5.2.2. L^1 and L^∞ errors at the final time of the various methods on two types of meshes. The last row reports the flux corrected results of ML1-WENO on the polygonal mesh.

	Polygonal Mesh		Square Mesh		Flux Corrected on Square Mesh	
	L^1-err	L^∞-err	L^1-err	L^∞-err	L^1-err	L^∞-err
ML1-WENO	1.82e−2	6.40e−1	1.86e−2	7.09e−1	1.89e−2	7.05e−1
WENO-AO	2.68e−2	7.18e−1	2.68e−2	7.54e−1	2.75e−2	7.53e−1
WENOZ-AO	2.37e−2	6.89e−1	2.39e−2	7.27e−1	2.46e−2	7.27e−1
ML1-WENO (FC)	1.88e−2	6.39e−1	×	×	×	×

Table 5
Section 5.3.1. L^1 and L^∞ errors at the final time using ML1-WENO(5,3) on randomly perturbed rectangular meshes.

Mesh	L^1-norm		L^∞-norm	
	Error	Order	Error	Order
30 × 10	3.01e−2	—	3.07e−1	—
60 × 20	1.48e−2	1.024	2.96e−1	0.053
120 × 40	7.68e−3	0.946	3.05e−1	−0.043
240 × 80	3.83e−3	1.004	2.91e−1	0.068

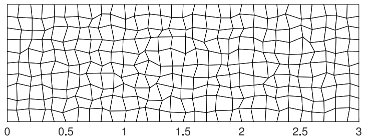


Fig. 11. Section 5.3.1. The 30 × 10 mesh.

again the SATh flux using the previously corrected flux each iteration (see, e.g., [39]). The overall L^1 error increases slightly, but now the solution satisfies the global maximum principle.

For a completely fair test of the two WENO-AO schemes, we also show the result of this test on a 100 × 100 uniform square mesh, which has the same number of cells as the polygonal mesh. All schemes approximate the solution using the same large and small stencil polynomials of degree 4 and 2, respectively, so they are all two-level reconstructions of (5,3)-type. The results appear in Fig. 10 and Table 4, where we see behavior similar to that observed using the polygonal mesh. In this test, all three schemes would benefit from flux corrected transport.

5.3. Numerical results for nonlinear equations

We continue with a few tests of nonlinear fluxes, concentrating on the ability of the scheme to handle shocks and rarefactions.

5.3.1. Simple Riemann shock and rarefaction

Let $\mathbf{f}(u) = (u^2/2, u^2/2)$ be the Burgers flux function. We test a Riemann shock and rarefaction in this example by taking the initial condition $u_0(x, y)$ equal to 1 in the strip $0.5 < x < 1.5$ and 0 elsewhere within the domain $\Omega = (0, 3) \times (0, 1)$. The true solution is

$$u(x,y,t) = \begin{cases} 0, & x < 0.5,\ x \geq 0.5t + 1.5, \\ (x-0.5)/t, & 0.5 \leq x < t + 0.5, \\ 1, & t + 0.5 \leq x < 0.5t + 1.5, \end{cases} \tag{48}$$

valid up to time $t = 2.0$, when the trailing rarefaction reaches the leading shock.

We use logically rectangular meshes, each given by randomly perturbing the vertices by a factor of 0.28 times the unperturbed mesh spacing h. The 30 × 10 mesh is depicted in Fig. 11. The reconstruction for each target mesh cell used the ring 1.5 large stencil and all cell-based small stencils of size 1 ring that overlap the target cell.

We computed the ML1-WENO(5,3) solution to time 2 using the standard SSP3 Runge–Kutta time integrator with time step $\Delta t = 0.25\,h$. The L^1 and L^∞ discrete norms of the error at the final time are given in Table 5. As expected, we see order 1 convergence in L^1 (and no L^∞ convergence) due to the presence of the shock. The solution appears nonoscillatory in Fig. 12.

5.3.2. Shock formation from a smooth initial condition

We use the Burgers flux $\mathbf{f}(u) = (u^2/2, u^2/2)$ on $(0,1)^2$, the smooth initial condition $u_0(x, y) = \sin^2(2\pi x)\sin^2(2\pi y)$, and periodic boundary conditions. The mesh depicted in Fig. 7 is used, and $\Delta t = 0.001$ (about CFL 0.66). The reconstructions are the same as those used in Section 5.2.2, and so are of type ML-WENO(5,3) over most of the domain.

The results are shown in Fig. 13. We see a clean shock formation, and essentially no undershoot.

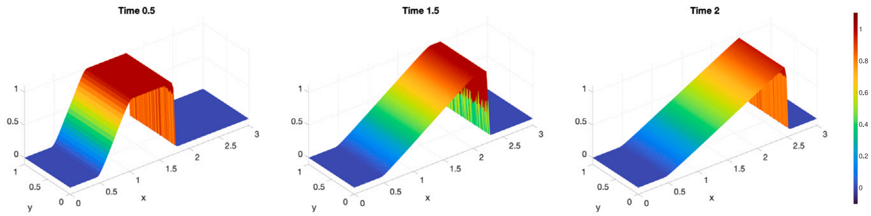


Fig. 12. Section 5.3.1. The reconstructed solution on the 240 × 80 perturbed mesh at times 0.5, 1.5, and 2.0. (The rarefaction and shock meet at time 2.0.)

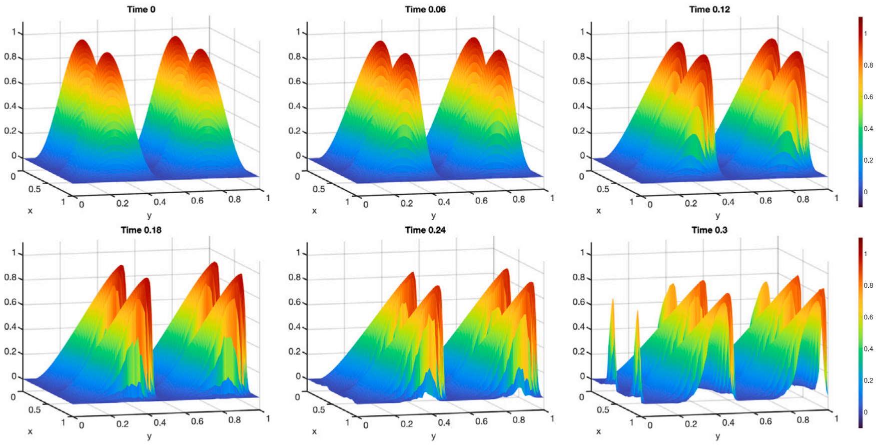


Fig. 13. Section 5.3.2, shock formation. The (nearly) ML-WENO(5,3) reconstructed solution on the mesh from Fig. 7 using $\Delta t = 0.001$ (about CFL 0.66).

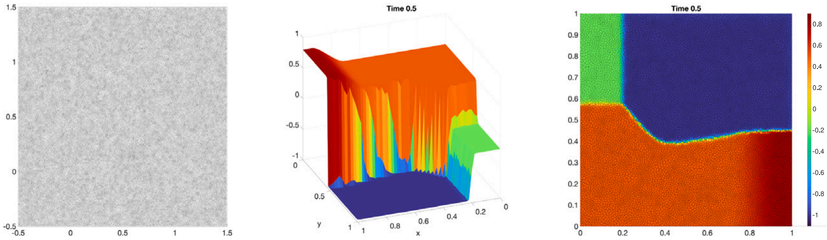


Fig. 14. Section 5.3.3, Burgers. From left to right we show the full mesh, the reconstructed solution, and the finite volume solution at time 0.5.

5.3.3. Complex interaction of Riemann shocks and rarefactions

We present two test cases involving the complex interaction of shocks and rarefactions. Both cases use the standard SSP3 Runge–Kutta time integrator. The reconstructions use the ring 1.5 large stencils and the cell-based ring 1 small stencils as before (targeting reconstructions of the desired type WENO(5,3)).

For the first case, again take the Burgers flux $\mathbf{f}(u) = (u^2/2, u^2/2)$, and impose the Riemann problem defined in [40]:

$$u_0(x,y) = \begin{cases} 0.5, & x < 0.5, y < 0.5, \\ 0.8, & x > 0.5, y < 0.5, \\ -0.2, & x < 0.5, y > 0.5, \\ -1.0, & x > 0.5, y > 0.5. \end{cases} \tag{49}$$

We use periodic boundary conditions on the domain $[-0.5, 1.5]^2$, but we only report results on the region on interest $[0,1]^2$. Polymesher [34] with two smoothing iterations gave the 200^2 cell and 75,921 vertex polygonal mesh used in the simulations (the mesh has about 10^4 cells in the region of interest). Only one stencil polynomial (which happened to be posed on a large stencil) had its degree lowered due to a condition number being greater that 1e+8, and some stencils ended up being smaller than desired, so overall the reconstruction is of type WENO(5,4,3,2). We used a time step $\Delta t = 0.002$, which is about 2/3 of the CFL limit (0.00315). The full mesh and solution at time 0.5 appear in Fig. 14.

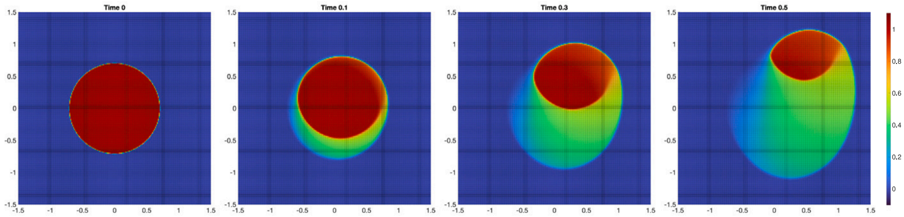


Fig. 15. Section 5.3.3, Buckley–Leverett with gravity. The reconstructed solution at times 0, 0.1, 0.3, and 0.5, computed using a 200 × 200 cell square mesh.

For the second test case, we consider the nonconvex Buckley–Leverett flux, modified to incorporate gravity, so $\mathbf{f}(u) = \dfrac{u^2}{u^2+(1-u)^2}\begin{pmatrix}1\\1-5(1-u)^2\end{pmatrix}$. In this test [41–43], the initial condition is zero over the domain $[-1.5, 1.5]^2$ except that it is one in the circle $x^2 + y^2 \leq 0.5$. We used a 200 × 200 cell square mesh, giving reconstructions of (5,3)-type, and a time step $\Delta t = 0.002$, which is about half the CFL limit (0.00410). The results at times 0, 0.1, 0.3, and 0.5 appear in Fig. 15.

For both tests, the results are of similar quality to what one sees in the literature; moreover, they are nearly identical to results we obtained using the WENO-AO and WENOZ-AO reconstructions.

5.4. Handling boundary conditions

High order finite volume and finite difference methods require wide stencils, which can be difficult to manage near the boundary of the domain while maintaining high order accuracy. Various approaches have been developed to address this challenge. Many involve using ghost cells or points outside the boundary so as to treat the reconstructions near the boundary as if they were in the interior of the domain [11]. For example, in the inverse Lax–Wendroff procedure [44], one substitutes normal derivatives of the solution with tangential and time derivatives using the partial differential equation in an iterative way to impose precise values for the ghost cells or points.

Semplice, Travaglia, and Puppo [11] describe how to handle the problem using the adaptive order reconstruction of Semplice and Visconti [27], which they can apply to rectangular Cartesian meshes. In a similar manner, the flexibility of the ML-WENO reconstruction allows us to tackle the difficulties in a direct way, independent of the complexity of the governing partial differential equations and intricacies of the geometry of the boundary. In this section we illustrate how one might impose inflow and outflow boundary conditions numerically without the use of ghost cells; that is, we restrict our reconstruction stencils to interior cells.

In finite volume methods, we must use the boundary condition to determine the numerical flux on the boundary. The first step is to determine whether a boundary is inflow or outflow. One can compute the reconstruction of the solution on the boundary (from the inside), call it u_B. If $\mathbf{f}(u_B)\cdot\nu > 0$, no boundary condition is imposed and the flux is simply $\mathbf{f}(u_B)\cdot\nu$. Otherwise, the inflow flux boundary condition (43) fixes the numerical flux directly as f_B. The inflow Dirichlet boundary condition (42) has its flux defined by (45), using the boundary value u_D and u_B substituting the right and left limits u^+ and u^-, ordered so u_D is outside the domain.

The key to the above strategy is to have accurate reconstructions near the boundary. It is likely that stencils are constructed in a way that some would cross the boundary (which we do not allow), and so they become smaller than expected. Therefore, our ML-WENO reconstructions will use reduced order stencil polynomials near the boundary, resulting in degraded accuracy of the numerical solution. However, ML-WENO gives us the flexibility to add stencils of the desired size that span back into the domain. This is especially needed at outflow boundaries, and inflow boundaries when the solution remains smooth. For example, an ML-WENO(5,3) reconstruction may revert to ML-WENO(3,2) near the boundary, so we would add an additional fifth order stencil polynomial to the reconstruction to rehabilitate it to ML-WENO(5,3,2).

In the case of an inflow Dirichlet boundary, there may be a discontinuous inflow shock wave. In that case, we require a swift response inside the domain to align with the specified Dirichlet value. We therefore add an extra constant stencil polynomial to the inflow boundary to accommodate the potential discontinuity. Since a constant approximation has only order one accuracy, one could further take a refined mesh near the boundary to improve the accuracy, but we do not do so here.

We present below two examples. The first handles a free outflow boundary, and the second handles an inflow Dirichlet boundary condition that generates an incoming shock wave.

5.4.1. Free outflow at the boundary

For the outflow example, we test linear transport of a sine wave traveling diagonally across the domain $(0, 1)^2$. Our true solution is exactly controllable at the inflow boundaries, having value zero there:

$$u(x,y,t) = \begin{cases} \sin^6\!\left(\dfrac{x-0.1-t}{0.8}\pi\right)\sin^6\!\left(\dfrac{y-0.1-t}{0.8}\pi\right), & 0.1+t \leq x \leq 0.9+t, \\ & 0.1+t \leq y \leq 0.9+t, \\ 0, & \text{otherwise.} \end{cases} \tag{50}$$

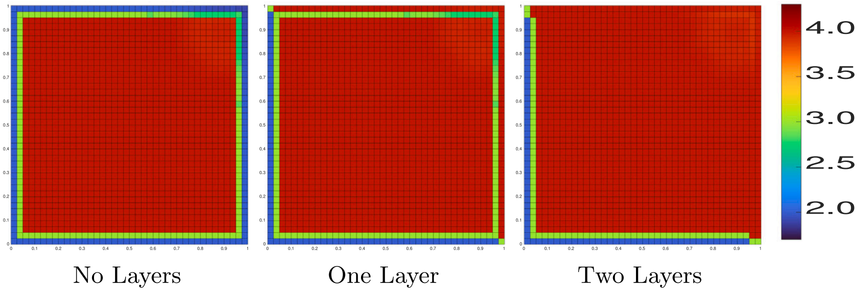


Fig. 16. Section 5.4.1, free outflow. The effective polynomial degree of the reconstructions computed on 80 × 80 mesh at time 0.4.

Table 6
Example 5.4.1, free outflow. Errors and convergence rates on rectangular meshes using ML1-WENO, adding 5th order stencils to reconstructions near the boundary in no, one, or two layers into the domain.

	No Layers		One Layer		Two Layers	
Mesh size	$L^\infty(L^1)$-err	Rate	$L^\infty(L^1)$-err	Rate	$L^\infty(L^1)$-err	Rate
20 × 20	4.41e−03	–	4.35e−03	–	4.26e−03	–
40 × 40	1.19e−04	5.21	1.13e−04	5.27	1.05e−04	5.34
80 × 80	5.05e−06	4.67	4.07e−06	4.80	3.82e−06	4.78
160 × 160	2.17e−07	4.54	1.32e−07	4.95	1.25e−07	4.93
250 × 250	2.99e−08	4.44	1.43e−08	4.98	1.36e−08	4.97

On the left and bottom sides, we impose the constant zero inflow condition. We have taken the sine wave to the 6th power in order to achieve a smooth connection between the sine wave peak and the constant zero region.

We use a rectangular mesh of $N \times N$ cells. We computed the overall ML1-WENO solution to time 0.4 using the fifth order Runge–Kutta time integrator due to Ruuth and Spiteri [33] with time step $\Delta t = 0.4h$.

The usual ML1-WENO(5,3) reconstruction drops to ML1-WENO(3,2) on the layer of (target) cells next to the boundary. At the same time, ML1-WENO(5,3) drops to ML1-WENO(4,3) on the next layer of cells away from the boundary. We consider three cases. First, we add no additional stencils (i.e., "no layers"), so the reconstructions are technically ML1-WENO(5,4,3,2), although they are ML1-WENO(5,3) in the interior of the domain. Second, we add additional fifth order stencils to the reconstructions for cells next to the outflow boundary (i.e., "one layer"). For each target cell next to the boundary, we add a single ring 2.5 stencil centered around the cell one layer inside the domain. Except near the outflow corner at (1,1), they have 33 cells, and each supports accurate approximation by a fourth degree (fifth order) stencil polynomial, even in the corner. Third, we use these same additional fifth order stencils in reconstructions but for target cells next to and one cell inside the outflow boundary (i.e., "two layers").

The effective polynomial degree of the reconstructions can be defined for each target cell as

$$p_{\text{eff}} = \sum_{\ell} \tilde{\omega}_\ell \, r_\ell - 1. \tag{51}$$

It is shown in Fig. 16, where we see good approximation (degree 4) over the middle of the domain for all three cases. The approximation near the inflow boundary matters not in this test. Near the outflow boundary, the no layer case drops to degree 2 while the other two cases are degree 4. Both the no layer and one layer cases, however, are only degree 3 in the layer of cells that are one cell inside the outflow boundary.

A reduced effective degree leads to a poorer approximation of the outflow flux. In

Table 6 we show the maximum over time of the discrete L^1 error. In the no layer case, we see a one half order loss of the convergence rate, i.e., order 4.5. In this test, we do not see a degradation of convergence in the one layer case, although the error is a bit greater than that in the two layer case.

In Fig. 17, we see the error distribution over the domain. The solution has a peak that propagates to the outflow corner, and so the error concentrates there. The error caused by the loss of reconstruction accuracy near the outflow boundary in the no layers case is evident, while the other two cases show good approximation there. We remark that the error near the outflow boundary eventually propagates out of the computational domain, and so does not pollute the solution in the interior of the domain in this test.

5.4.2. Inflow Dirichlet boundary conditions

For the inflow example, we use the Burgers-like equation $u_t + u^3 u_x = 0$ with the flux $\mathbf{f}(u) = (u^4/4, 0)$ in the domain $(0,1)^2$. The Dirichlet boundary condition on the left boundary ($x = 0$) is the continuous function

$$u_D(0, y, t) = \begin{cases} \sin^2(2\pi y), & y \le 0.25,\ \ y \ge 0.75, \\ 1, & 0.25 < y < 0.75, \end{cases} \tag{52}$$

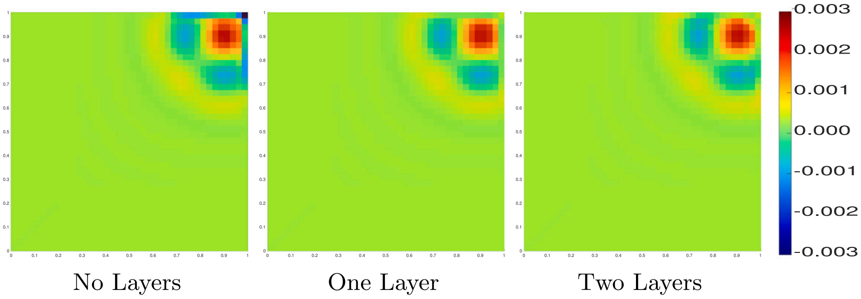


Fig. 17. Section 5.4.1, free outflow. Error distribution computed on 80 × 80 mesh at time 0.4. The error near the outflow boundary is notably diminished by incorporating additional stencils.

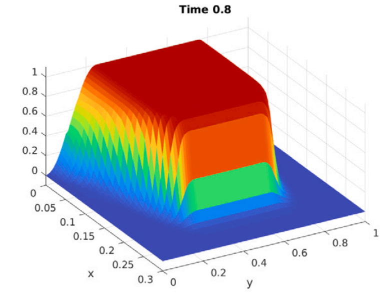


Fig. 18. Section 5.4.2, inflow propagation of nonlinear shock wave. The ML-WENO(5,4,3,2,1) reconstructed solution on the 80 × 80 rectangular mesh at time 0.8.

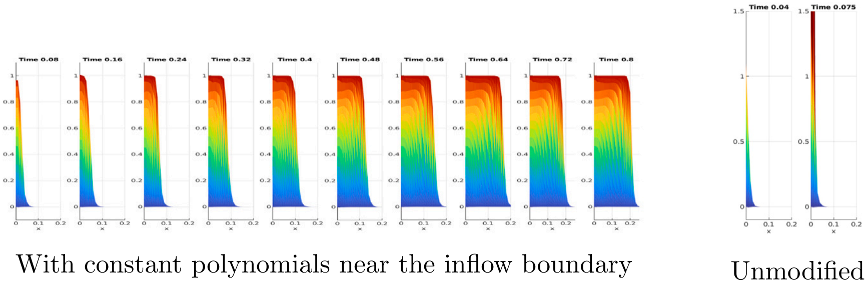


Fig. 19. Section 5.4.2, inflow propagation of nonlinear shock wave. On the left, we show the results of the ML-WENO(5,4,3,2,1) reconstructed solution at ten evenly spaced times from 0.08 to 0.8. On the right we show the ML-WENO(5,4,3,2) reconstructed solution at times 0.04 and 0.075. This solution goes unstable and the computer program returns NaN after time 0.075.

and the right boundary is a free outflow boundary. The top and bottom boundaries are Dirichlet with value zero. We test a shock wave traveling into the computational domain by setting the initial condition $u_0(x, y) = 0$.

We use a rectangular mesh with 80 × 80 cells. We compute the solution to the final time 0.8 using the same fifth order Runge–Kutta time integrator as before [33] and set the time step to be $\Delta t = 0.4\,h$ ($\Delta t = 0.005$, CFL $= 0.4$). We use the same set of stencils used in the outflow problem, which results in an overall ML-WENO(5,4,3,2) reconstruction that is ML-WENO(5,3) over most of the domain, and drops to ML-WENO(4,3) and ML-WENO(3,2) near the boundaries.

The ML-WENO(5,4,3,2,1) solution is shown in Fig. 18. It uses the usual set of stencils augmented with extra single cell stencils (constant polynomials) near the inflow boundary to resolve the incoming shock wave. The result is as we should expect, since the Rankine–Hugoniot shock speed in the x-direction near the boundary is $u_D^3/4$, which is $\sin^6(2\pi y)/4$ for $y \leq 0.25$ and $y \geq 0.75$.

On the left of Fig. 19, we show the shock wave traveling into the domain at ten evenly spaced times from 0.08 to 0.8. The wave enters with a sharp profile and very little overshoot (a small amount can be observed at time 0.16).

We compare our results to those using the ML-WENO(5,4,3,2) reconstruction, which has not been supplemented with constant polynomials near the inflow boundary. The results are shown on the right of Fig. 19. We see significant overshoot as the shock wave enters the domain. It is so severe that the solution goes well outside the expected range $[0, 1]$, and the maximum numerical wave speed greatly exceeds the presumed speed of one used within the Lax–Friedrichs numerical flux, so it does not stabilize the solution after time $t = 0.075$. The computer program returns the IEEE code for Not a Number (NaN) at the next time step. Adding additional constant stencils near the inflow boundary significantly reduces the overshoot caused by linear reconstruction of a sharp front and thereby allows the numerical flux to stabilize the solution.

We remark that the high power in the flux function is needed to see this dramatic result. Both schemes perform well if Burger's equation is used with these boundary and initial conditions. Nonetheless, it should be clear that the flexibility to add constant stencils near the boundary is a distinct advantage of ML-WENO reconstructions.

6. Summary and conclusions

We presented a general framework for solving scalar conservation laws using finite volume WENO techniques on general computational meshes in multiple space dimensions.

As the numerical results showed in Section 2.3, it is difficult, and perhaps impossible, to relate the *geometry* of a general mesh stencil to the quality of the (finite volume) stencil polynomial approximation defined on it. However, based on the Bramble–Hilbert Lemma, we were able to give an *algebraic* condition (10) that ensures the uniform error bound (8).

We suggested a practical algorithm to deal with bad stencils. If the condition (10) is not met, we keep the same stencil but reduce the order of the polynomial approximation. The algorithm will terminate at worst with a constant polynomial.

We then developed a multilevel WENO (ML-WENO) reconstruction (21) that was flexible enough to handle stencil polynomials of various degrees. The key is to scale the linear weights according to (22) before normalizing (24).

Lemmas 3.1–3.2 quantified the effect of the true solution and its smoothness indicator on the size of the weights. The nonlinear weights for both inaccurate oscillatory polynomials of higher degree (i.e., those crossing a shock or steep front) and smooth polynomials of lower degree are biased to zero, thereby allowing selection of the smooth polynomial(s) of maximal degree of approximation $r_{\max}$ (defined in (29)). The rates of bias are sufficient to lead us to the desired convergence result Theorem 3.3. We performed some numerical tests, which confirmed and illustrated the ability of the reconstruction to approximate accurately both smooth and discontinuous functions.

We saw good results when applying ML-WENO to the approximation of scalar conservation laws. We tested smooth solutions and problems with shocks and rarefactions. We also illustrated the flexibility of ML-WENO by using it to impose boundary conditions.

Preliminary application of ML-WENO to the Euler system of gas dynamics shows reasonable results, especially if ML2-WENO is used. However, some problems show excessive numerical diffusion, so we expect that some modifications are needed to handle systems of conservation laws. This is the subject of future work.

CRediT authorship contribution statement

Todd Arbogast: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Chieh-Sen Huang:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – review & editing. **Chenyu Tian:** Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] A. Harten, S. Osher, Uniformly high-order accurate nonoscillatory schemes I, SIAM J. Numer. Anal. 24 (2) (1987) 279–309.
[2] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[3] X.D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[4] A. Harten, S.R. Chakravarthy, Multi-Dimensional ENO Schemes for General Geometries, Tech. Rep. 91–76, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia, 1991, contract No. NAS1-18605.
[5] T. Sonar, On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection, Comput. Methods Appl. Mech. Engrg. 140 (1997) 157–181.
[6] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys. 144 (1998) 194–212.
[7] M. Kaŝer, A. Iske, ADER schemes on adaptive triangular meshes for scalar conservation laws, J. Comput. Phys. 205 (2005) 486–508.
[8] M. Dumbser, W. Boscheri, M. Semplice, G. Russo, Central weighted ENO schemes for hyperbolic conservation laws on fixed and moving unstructured meshes, SIAM J. Sci. Comput. 39 (6) (2017) A2564–A2591.
[9] P. Tsoutsanis, Stencil selection algorithms for WENO schemes on unstructured meshes, J. Comput. Phys. 475 (2023) 108840.
[10] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.
[11] M. Semplice, E. Travaglia, G. Puppo, One- and multi-dimensional CWENOZ reconstructions for implementing boundary conditions without ghost cells, Commun. Appl. Math. Comput. 5 (2023) 143–169, http://dx.doi.org/10.1007/s42967-021-00151-4.
[12] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.

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

[14] D.S. Balsara, S. Garain, C.-W. Shu, An efficient class of WENO schemes with adaptive order, J. Comput. Phys. 326 (2016) 780–804.

[15] T. Arbogast, C.-S. Huang, X. Zhao, Accuracy of WENO and adaptive order WENO reconstructions for solving conservation laws, SIAM J. Numer. Anal. 56 (3) (2018) 1818–1847, http://dx.doi.org/10.1137/17M1154758.

[16] D.S. Balsara, S. Garain, V. Florinski, W. Boscheri, An efficient class of WENO schemes with adaptive order for unstructured meshes, J. Comput. Phys. 404 (2020) 109062, http://dx.doi.org/10.1016/j.jcp.2019.109062.

[17] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.

[18] A. Harten, Multi-resolution analysis for ENO schemes, Tech. Rep. 23665–5225, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia, 1991, contract No. NAS1-18605.

[19] R. Abgrall, A. Harten, Multiresolution representation in unstructured meshes, SIAM J. Numer. Anal. 35 (6) (1998) 2128–2146.

[20] W. Dahmen, B. Gottschlich-Müller, S. Müller, Multiresolution schemes for conservation laws, Numer. Math. 88 (2001) 399–443.

[21] G. Chiavassa, R. Donat, S. Müller, Multiresolution-based adaptive schemes for hyperbolic conservation laws, in: T. Plewa, T. Linde, V. Weiss (Eds.), Adaptive Mesh Refinement: Theory and Applications, in: Lecture Notes in Computational Science and Engineering, vol. 41, Springer-Verlag, Berlin, 2003, pp. 137–159.

[22] R. Bürger, A. Kozakevicius, Adaptive multiresolution WENO schemes for multi-species kinematic flow models, J. Comput. Phys. 224 (2007) 1190–1222.

[23] J. Zhu, C.-W. Shu, A new type of multi-resolution WENO schemes with increasingly higher order of accuracy, J. Comput. Phys. 375 (2018) 659–683, http://dx.doi.org/10.1016/j.jcp.2018.09.003.

[24] J. Zhu, C.-W. Shu, A new type of multi-resolution WENO schemes with increasingly higher order of accuracy on triangular meshes, J. Comput. Phys. 392 (2019) 19–33, http://dx.doi.org/10.1016/j.jcp.2019.04.027.

[25] J. Zhu, C.-W. Shu, A new type of third-order finite volume multi-resolution WENO schemes on tetrahedral meshes, J. Comput. Phys. 406 (2020) 109212, http://dx.doi.org/10.1016/j.jcp.2019.109212.

[26] Y. Jiang, High order finite difference multi-resolution WENO method for nonlinear degenerate parabolic equations, J. Sci. Comput. 86 (16) (2021) http://dx.doi.org/10.1007/s10915-020-01382-y.

[27] M. Semplice, G. Visconti, Efficient implementation of adaptive order reconstructions, J. Sci. Comput. 83 (6) (2020) http://dx.doi.org/10.1007/s10915-020-01156-6.

[28] M. Dumbser, M. Käser, Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, J. Comput. Phys. 221 (2007) 693–723.

[29] M. Dumbser, M. Käser, V.A. Titarev, E.F. Toro, Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems, J. Comput. Phys. 226 (1) (2007) 204–243.

[30] J.H. Bramble, S.R. Hilbert, Estimation of linear functionals on Sobolev spaces with applications to Fourier transforms and spline interpolation, SIAM J. Numer. Anal. 7 (1970) 112–124.

[31] T. Dupont, L.R. Scott, Polynomial approximation of functions in Sobolev space, Math. Comp. 34 (1980) 441–463.

[32] F.W.J. Olver, Introduction to Asymptotics and Special Functions, Academic Press, New York, 1974.

[33] S.J. Ruuth, R.J. Spiteri, High-order strong-stability-preserving runge-kutta methods with downwind-biased spatial discretizations, SIAM J. Numer. Anal. 42 (3) (2004) 974–996.

[34] C. Talischi, G.H. Paulino, A. Pereira, I.F.M. Menezes, Polymesher: a general-purpose mesh generator for polygonal elements written in matlab, Struct. Multidiscip. Optim. 45 (2012) 309–328.

[35] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (2) (1996) 627–665.

[36] M. Castro, B. Costa, W. Don, High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws, J. Comput. Phys. 230 (2011) 1766–1792.

[37] J.P. Boris, D.L. Book, Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11 (1) (1973) 38–69, http://dx.doi.org/10.1016/0021-9991(73)90147-2.

[38] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362, http://dx.doi.org/10.1016/0021-9991(79)90051-2.

[39] M. Joshaghani, B. Rivière, M. Sekachev, Maximum-principle-satisfying discontinuous Galerkin methods for incompressible two-phase immiscible flow, Comput. Methods Appl. Mech. Engrg. 391 (2022) 114550, http://dx.doi.org/10.1016/j.cma.2021.114550.

[40] G.S. Jiang, E. Tadmor, Nonoscillatory central schemes for multidimensional hyperbolic conservation laws, SIAM J. Sci. Comput. 19 (1998) 1892–1917.

[41] K.H. Karlsen, K. Brusdal, H. Dahle, S. Evje, K.-A. Lie, The corrected operator splitting approach applied to a nonlinear advection-diffusion problem, Comput. Methods Appl. Mech. Engrg. 167 (3–4) (1998) 239–260.

[42] I. Christov, B. Popov, New non-oscillatory central schemes on unstructured triangulations for hyperbolic systems of conservation laws, J. Comput. Phys. 227 (2008) 5736–5757.

[43] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys. 230 (2011) 4248–4267, http://dx.doi.org/10.1016/j.jcp.2010.11.043.

[44] S. Tan, C.-W. Shu, Inverse lax-wendroff procedure for numerical boundary conditions of conservation laws, J. Comput. Phys. 229 (21) (2010) 8144–8166, http://dx.doi.org/10.1016/j.jcp.2010.07.014.