Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 196 (2008) 433–447
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Observations on the fifth-order WENO method with non-uniform meshes

Rong Wang [a,1], Hui Feng [b,2], Raymond J. Spiteri [a,*,1]

[a] Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada S7N 5C9
[b] Department of Mathematics, Wuhan University, Wuhan, Hubei, China

Abstract

The weighted essentially non-oscillatory (WENO) methods are a popular high-order spatial discretization for hyperbolic partial differential equations. Typical treatments of WENO methods assume a uniform mesh. In this paper, we give explicit formulas for the finite-volume, fifth-order WENO (WENO5) method on non-uniform meshes in a way that is amenable to efficient implementation. We then compare the performance of the non-uniform mesh approach with the classical uniform mesh approach for the finite-volume formulation of the WENO5 method. We find that the numerical results significantly favor the non-uniform mesh approach both in terms of computational efficiency as well as memory usage. We expect this investigation to provide a basis for future work on adaptive mesh methods coupled with the finite-volume WENO methods.

Keywords: Finite-volume discretization; Fifth-order WENO method; Non-uniform mesh

1. Introduction

The essentially non-oscillatory (ENO) methods were first introduced by Harten et al. in [1,2]. They were the first successful high-order methods for the spatial discretization of hyperbolic conservation laws that had the ENO property. This property is considered to be very useful in the numerical solution of hyperbolic conservation laws because numerical methods often produce spurious oscillations when applied to such problems, especially near shocks or other discontinuous behavior of the solution. The finite-volume ENO spatial discretization was studied in [2], where it was shown to have uniform high-order accuracy right up to the location of any discontinuities. Later Shu and Osher [3,4] developed the finite-difference ENO method. The main idea behind ENO methods is to choose from among several candidates the stencil on which the solution varies

* Corresponding author.
E-mail addresses: rong@cs.usask.ca (R. Wang), hui_feng163@tom.com (H. Feng), spiteri@cs.usask.ca (R.J. Spiteri).
[1] The work of this author was partially supported by MITACS, NSERC, and Martec, Ltd.
[2] The work of this author was partially supported by MITACS and NSET.

0096-3003/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2007.06.024

the most smoothly and then approximate the flux at the cell boundaries with a high order of accuracy, thus avoiding the large spurious oscillations caused by interpolating data across discontinuities.

Weighted ENO (WENO) methods were developed in [5,6] to address potential numerical instabilities in choosing ENO stencils. WENO methods use a convex combination of all the ENO candidate stencils; i.e., rather than choosing one specific ENO stencil, each stencil is assigned a weight between 0 and 1. Given r ENO stencils of order r, the weights for the WENO method are chosen such that the numerical flux is approximated to order $(2r-1)$ in smooth regions, while in regions near solution discontinuities WENO methods emulate ENO methods so that the ENO property is achieved. In other words, WENO methods achieve a higher order of accuracy than ENO methods in smooth regions, while retaining the ENO property at discontinuities.

Explicit formulas for WENO coefficients on uniform meshes appear in, e.g., [7,8]. A framework for deriving WENO coefficients for non-uniform meshes is established in [7]; explicit formulas appear for uniform meshes. Finite-difference WENO schemes with orders from 7 to 13 are derived in [12] for one-dimensional uniform meshes. Schemes of third and fifth order in multiple spatial dimensions on uniform meshes are derived in [6,13]. WENO coefficients on arbitrary triangular meshes are derived for second-order schemes in [14] and for third- and fourth-order schemes in [15]. Computations in two dimensions with WENO discretizations are performed in [8] on triangular and rectangular meshes.

Given a fixed uniform mesh, the finite-difference WENO methods and the finite-volume WENO methods produce identical spatial discretization operators for one-dimensional, linear, constant-coefficient partial differential equations (PDEs). Of course, they do differ in the quantities that they evolve; i.e., the finite-difference approach evolves point values whereas the finite-volume approach evolves cell averages. For a nonlinear scalar hyperbolic PDE, the equivalence of the finite-difference and finite-volume WENO spatial discretization operators does not hold anymore; however, the computational costs for the two methods are still the same.

For multi-dimensional problems, the finite-difference WENO methods are significantly less computationally expensive than the finite-volume WENO methods [7,8]. Specifically, the finite-difference WENO methods are about 4 times less expensive than the finite-volume WENO methods of the same order for two-dimensional problems. This becomes about 9 times less expensive for three-dimensional problems. In this sense the finite-difference WENO methods are more favorable than the finite-volume methods. Furthermore, when the finite-volume WENO methods are applied for multi-dimensional problems, negative weights may arise [8]; specialized techniques have been used to overcome the difficulty [8].

In recent years, adaptive mesh methods have been used with great success for parabolic problems; see, e.g., [9,10]. They have also been used to solve hyperbolic conservation laws, e.g., [11]. We note that the finite-difference WENO method of third order or higher can only be applied to uniform or smoothly varying meshes [7], i.e., a mesh such that a smooth transformation $\xi = \xi(x)$ transforms the original mesh into a uniform mesh in the new variable ξ. This eliminates the possibility of using non-uniform or adaptive meshes with finite-difference WENO methods. In this paper, we focus only on one-dimensional problems, where the computational costs of the finite-volume and finite-difference WENO methods are the same, so we restrict our comparison to the relative efficiency of the finite-volume WENO methods on uniform and non-uniform meshes.

At present it is not known whether the finite-volume WENO methods on a non-uniform (adaptive) mesh can compete with the finite-difference WENO methods on a uniform mesh in terms of efficiency. For example, there are many efficiencies afforded to implementations using uniform meshes in terms of being able to precompute coefficients. In this paper, we perform a quantitative comparison of the relative efficiency of non-uniform mesh approach with the uniform mesh approach for the finite-volume, fifth-order WENO (WENO5) method. Our numerical results show that the use of non-uniform meshes can lead to significant improvements in efficiency over the use of uniform meshes, both in terms of computational efficiency as well as memory usage. These are the first such quantitative comparisons to be made available of which we know. This leads us to hypothesize that if a suitable adaptive mesh algorithm can be derived, an adaptive finite-volume WENO approach can generally outperform the classical finite-difference WENO approach on uniform meshes. We leave the development of an adaptive mesh strategy as future work. We hope this investigation will provide a basis for future work on adaptive mesh methods coupled with the finite-volume WENO methods.

The remainder of this paper proceeds as follows. For completeness, in Section 2 we give explicit, detailed formulas for the coefficients of the finite-volume WENO5 method on non-uniform meshes. We note that these

formulas are presented in a way that takes advantage of recursive patterns in the coefficients, thus improving the efficiency of implementation. Moreover, we also note that this is a different approach in computing ENO stencils from that of the classical finite-volume WENO methods, which use the divided difference approach [7]. The smoothness indicators, which are used for computing the ENO stencil weights, are also derived explicitly for a non-uniform mesh. Eqs. (65)–(70) clearly demonstrate that the finite-volume WENO methods on non-uniform meshes still use a convex combination of ENO stencils. In Section 3, we give the results of two numerical experiments. The first experiment involves the advection equation, which we use to illustrate the correctness of our formulas on a problem with a smooth exact solution. We then solve the advection equation with a moving shock as its exact solution. Assuming that the shock location is known a priori, we manually construct fine meshes where the shock is located during the simulation. The numerical results are significantly better than the uniform mesh approach, both in terms of computational efficiency as well as memory usage. The last experiment involves the Burgers equation, where the exact solution is a shock with a fixed location that develops from a smooth initial condition. The results from this example suggest some necessary properties for a successful adaptive mesh strategy.

2. The finite-volume WENO5 method on non-uniform meshes

Consider the one-dimensional, scalar hyperbolic conservation law on a normalized spatial domain

$$u_t = -f_x(u), \quad 0 < x < 1, \ \ t > 0. \tag{1}$$

Given a non-uniform mesh

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = 1,$$

we define cells I_i and cell centers x_i by

$$I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right], \quad x_i = \frac{1}{2}\left(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}\right), \qquad i = 1, 2, \ldots, N.$$

Finite-volume methods are based on the cell averages $\bar{u}(x_i, t)$ of $u(x,t)$; i.e.,

$$\bar{u}(x_i, t) = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(\xi, t) \mathrm{d}\xi,$$

where $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ is the ith cell size, $i = 1, 2, \ldots, N$.

We now integrate (1) over I_i and obtain

$$\frac{\mathrm{d}\bar{u}(x_i, t)}{\mathrm{d}t} = -\frac{1}{\Delta x_i}\left[f\left(u\left(x_{i+\frac{1}{2}}, t\right)\right) - f\left(u\left(x_{i-\frac{1}{2}}, t\right)\right)\right]. \tag{2}$$

We now approximate the conservative method (2) by

$$\frac{\mathrm{d}\bar{u}_i(t)}{\mathrm{d}t} = -\frac{1}{\Delta x_i}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}),$$

where $\bar{u}_i(t)$ is the numerical approximation to $\bar{u}(x_i, t)$, and the numerical flux $\hat{f}_{i+\frac{1}{2}}$ is the approximation to $f(u(x_{i+\frac{1}{2}}, t))$, defined by

$$\hat{f}_{i+\frac{1}{2}} = G\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}\right), \tag{3}$$

where $u^{\pm}_{i+\frac{1}{2}}$ are obtained from the WENO reconstruction procedure that is described in the remainder of this section. $G(a,b)$ must be [16]

- consistent with the physical flux f; i.e., $G(a,a) = f(a)$;
- Lipschitz continuous in both a and b; i.e., there exists a constant $K \geqslant 0$ such that $|G(a_1,b) - G(a_2,b)| \leqslant K|a_1 - a_2| \ \forall a_1, a_2, b$, and such that $|G(a,b_1) - G(a,b_2)| \leqslant K|b_1 - b_2| \ \forall b_1, b_2, a$;
- nondecreasing in a and nonincreasing in b.

There are several well-known monotone fluxes with the above properties, such as the Godunov flux, the Engquist–Osher flux, and the Lax–Friedrichs flux [16]. In this paper, we consider only the Godunov flux

$$G(a,b)=\begin{cases}\min\limits_{a\leqslant u\leqslant b} f(u) & \text{if } a\leqslant b,\\ \max\limits_{b\leqslant u\leqslant a} f(u) & \text{if } a>b.\end{cases}$$

We now describe the WENO5 reconstruction procedure for $u^{\pm}_{i+\frac{1}{2}}$.

2.1. Spatial stencils

Given a cell I_i, in order to obtain third-order accurate spatial discretizations to use as building blocks for the WENO5 method, we can choose a *stencil* $S_r^{(i)}$ based on r cells to the left of I_i, k cells to the right of I_i, and I_i itself. If we require $r,k\geqslant 0$, and $r+k=2$, then there are only three possible stencils, i.e., $S_r^{(i)}=\{I_{i-r},I_{i-r+1},I_{i-r+2}\}$, $r=0,1,2$. Furthermore, given t_n, there is only one unique quadratic polynomial, denoted by $P_r^{(i)}(x)$, whose cell average in each cell of S_r is equal to $\bar{u}_m, m=i-r,i-r+1,i-r+2$; i.e.,

$$\frac{1}{\Delta x_m}\int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} P_r^{(i)}(\xi)\mathrm{d}\xi=\bar{u}(x_m,t_n),\quad m=i-r,i-r+1,i-r+2. \tag{4}$$

It can be shown by standard analysis that in smooth regions $P_r^{(i)}(x)$ is a third-order accurate approximation to the exact solution $u(x,t_n)$ inside I_i; i.e.,

$$P_r^{(i)}(x)=u(x,t_n)+\mathrm{O}((\Delta x_i)^3),\quad x\in I_i. \tag{5}$$

In particular, at the cell boundaries, we have

$$P_r^{(i)}\left(x_{i\pm\frac{1}{2}}\right)=u\left(x_{i\pm\frac{1}{2}},t_n\right)+\mathrm{O}((\Delta x_i)^3).$$

There are convex combinations of $P_r(x_{i\pm\frac{1}{2}})$, $r=0,1,2$, that yield fifth-order accurate approximations to $u(x_{i\pm\frac{1}{2}},t_n)$ in smooth regions; i.e.,

$$\sum_{r=0}^{2} w_r^{(i)}P_r^{(i)}\left(x_{i-\frac{1}{2}}\right)=u\left(x_{i-\frac{1}{2}},t_n\right)+\mathrm{O}((\Delta x_i)^5), \tag{6}$$

$$\sum_{r=0}^{2} \hat{w}_r^{(i)}P_r^{(i)}\left(x_{i+\frac{1}{2}}\right)=u\left(x_{i+\frac{1}{2}},t_n\right)+\mathrm{O}((\Delta x_i)^5), \tag{7}$$

where

$$w_r^{(i)},\hat{w}_r^{(i)}\geqslant 0,\qquad \sum_{r=0}^{2} w_r^{(i)}=\sum_{r=0}^{2}\hat{w}_r^{(i)}=1. \tag{8}$$

The key to the success of WENO methods is the choice of the weights w_r, $\hat{w}_r, r=0,1,2$. These weights must satisfy (6) or (7) in smooth regions and emulate the ENO property where u has discontinuous behavior; i.e., they should emulate the first-order upwind method at a discontinuity.

The WENO reconstruction for $u^{+}_{i+\frac{1}{2}}$ and $u^{-}_{i+\frac{1}{2}}$ in (3) is defined as follows:

$$u^{+}_{i+\frac{1}{2}}=\sum_{r=0}^{2} w_r^{(i+1)}P_r^{(i+1)}\left(x_{i+\frac{1}{2}}\right), \tag{9}$$

$$u^{-}_{i+\frac{1}{2}}=\sum_{r=0}^{2} \hat{w}_r^{(i)}P_r^{(i)}\left(x_{i+\frac{1}{2}}\right). \tag{10}$$

Based on (6) and (7), we see that both $u^{\pm}_{i+\frac{1}{2}}$ are fifth-order approximations to $u(x_{i+\frac{1}{2}},t_n)$.

We let $i+1 \to i$ in (9) to obtain

$$u^+_{i-\frac{1}{2}} = \sum_{r=0}^{2} w_r^{(i)} P_r^{(i)}\left(x_{i-\frac{1}{2}}\right). \tag{11}$$

Now because the terms in (10) and (11) depend on $w_r^{(i)}$, $w_r^{(i)}$, and $P_r^{(i)}(x_{i\pm\frac{1}{2}})$, for simplicity, we drop the superscript (i) from the expressions. Eqs. (10) and (11) thus become

$$u^+_{i-\frac{1}{2}} = \sum_{r=0}^{2} w_r P_r\left(x_{i-\frac{1}{2}}\right), \tag{12}$$

$$u^-_{i+\frac{1}{2}} = \sum_{r=0}^{2} \hat{w}_r P_r\left(x_{i+\frac{1}{2}}\right). \tag{13}$$

When a uniform mesh is used, detailed formulas for compute w_r, w_r, and $P_r(x_{i\pm\frac{1}{2}})$ can be found in [6,7]. We now give explicit, detailed formulas for a non-uniform mesh. For convenience, we define h_m, $m = 1,\ldots,5$, as follows:

$$h_m = \Delta x_{i-3+m}, \quad m = 1,\ldots,5.$$

From [7], we know that

$$P_r(x) = \sum_{j=0}^{2} C_{rj}(x)\bar{u}_{i-r+j}, \tag{14}$$

where

$$C_{rj}(x) = B_{rj}(x)h_{3-r+j} \tag{15}$$

and

$$B_{rj}(x) = \sum_{m=j+1}^{3} \frac{\sum_{l=0,l\neq m}^{3}\left(\prod_{q=0,q\neq m,l}^{3}\left(x - x_{i-r+q-\frac{1}{2}}\right)\right)}{\prod_{l=0,l\neq m}^{3}\left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}\right)}. \tag{16}$$

Defining $\hat{c}_{rj} = C_{rj}(x_{j+\frac{1}{2}})$ and $\hat{b}_{rj} = B_{rj}(x_{j+\frac{1}{2}}), r,j = 0,1,2$, we can compute $P_r(x_{i+\frac{1}{2}})$ in (13) as follows:

$$P_r\left(x_{i+\frac{1}{2}}\right) = \sum_{j=0}^{2} \hat{c}_{rj}\bar{u}_{i-r+j}, \tag{17}$$

where

$$\hat{c}_{rj} = \hat{b}_{rj}h_{3-r+j} \tag{18}$$

and

$$\hat{b}_{rj} = \sum_{m=j+1}^{3} \frac{\sum_{l=0,l\neq m}^{3}\left(\prod_{q=0,q\neq m,l}^{3}\left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}}\right)\right)}{\prod_{l=0,l\neq m}^{3}\left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}\right)}. \tag{19}$$

In terms of h_m, $m = 1,\ldots,5$, we find that explicit formulas for $\hat{b}_{rj}$, $r,j = 0,1,2$, are

$$\hat{b}_{22} = \frac{1}{h_1+h_2+h_3} + \frac{1}{h_2+h_3} + \frac{1}{h_3}, \tag{20}$$

$$\hat{b}_{21} = \hat{b}_{22} - \frac{(h_1+h_2+h_3)(h_2+h_3)}{(h_1+h_2)h_2h_3}, \tag{21}$$

$$\hat{b}_{20} = \hat{b}_{21} + \frac{(h_1+h_2+h_3)h_3}{h_1h_2(h_2+h_3)}, \tag{22}$$

$$\hat{b}_{12} = \frac{(h_2 + h_3)h_3}{(h_2 + h_3 + h_4)(h_3 + h_4)h_4}, \tag{23}$$

$$\hat{b}_{11} = \hat{b}_{12} + \frac{1}{h_2 + h_3} + \frac{1}{h_3} - \frac{1}{h_4}, \tag{24}$$

$$\hat{b}_{10} = \hat{b}_{11} - \frac{(h_2 + h_3)h_4}{h_2 h_3 (h_3 + h_4)}, \tag{25}$$

$$\hat{b}_{02} = -\frac{h_3 h_4}{(h_3 + h_4 + h_5)(h_4 + h_5)h_5}, \tag{26}$$

$$\hat{b}_{01} = \hat{b}_{02} + \frac{h_3(h_4 + h_5)}{(h_3 + h_4)h_4 h_5}, \tag{27}$$

$$\hat{b}_{00} = \hat{b}_{01} + \frac{1}{h_3} - \frac{1}{h_4} - \frac{1}{h_4 + h_5}. \tag{28}$$

By defining $c_{rj} = C_{rj}(x_{j-\frac{1}{2}})$ and $b_{rj} = B_{rj}(x_{j-\frac{1}{2}})$, $r, j = 0, 1, 2$, we can compute $P_r(x_{i-\frac{1}{2}})$ in (12) as follows:

$$P_r\left(x_{i-\frac{1}{2}}\right) = \sum_{j=0}^{2} c_{rj}\bar{u}_{i-r+j}, \tag{29}$$

where

$$c_{rj} = b_{rj}h_{3-r+j} \tag{30}$$

and

$$b_{rj} = \sum_{m=j+1}^{3} \frac{\sum_{l=0, l\neq m}^{3}\left(\prod_{q=0, q\neq m, l}^{3}\left(x_{i-\frac{1}{2}} - x_{i-r+q-\frac{1}{2}}\right)\right)}{\prod_{l=0, l\neq m}^{3}\left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}\right)}. \tag{31}$$

It is easy to verify that

$$b_{rj} = \hat{b}_{r-1,j} \tag{32}$$

for $j = 0, 1, 2$, and $r = 1, 2$. When $r = 0$, b_{0j}, $j = 0, 1, 2$, can be computed as follows:

$$b_{02} = \frac{h_3(h_3 + h_4)}{(h_3 + h_4 + h_5)(h_4 + h_5)h_5}, \tag{33}$$

$$b_{01} = b_{02} - \frac{h_3(h_3 + h_4 + h_5)}{(h_3 + h_4)h_4 h_5}, \tag{34}$$

$$b_{00} = b_{01} + \frac{(h_3 + h_4)(h_3 + h_4 + h_5)}{h_3 h_4 (h_4 + h_5)}. \tag{35}$$

Eqs. (17)–(35) give the expressions for $P_r(x_{i-\frac{1}{2}})$ and $P_r(x_{i+\frac{1}{2}})$. We note that the coefficients c_{rj} and $\hat{c}_{rj}, r, j = 0, 1, 2$, depend only on h_m, $m = 1, \ldots, 5$. That is, the coefficients only need to be computed once throughout the computation provided the mesh is fixed.

2.2. Smoothness measure

In order to achieve high-order accuracy in regions where the solution is smooth while emulating the first-order, upwind method in regions where the solution has discontinuous behavior, a smoothness measure for each stencil is computed as suggested in [6,7].

The smoothness measure IS_r for the rth stencil is defined by

$$IS_r = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h_3 (P_r'(x))^2 \mathrm{d}x + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (h_3)^3 (P_r''(x))^2 \mathrm{d}x. \tag{36}$$

When the stencil S_r is smooth (i.e., P_r is a smooth polynomial in I_i), IS_r satisfies

$$IS_r = (P_r'(x_i)h_3)^2(1 + \mathrm{O}((h_3)^2)), \tag{37}$$

whereas for a non-smooth stencil S_r, we have

$$IS_r = \mathrm{O}(1). \tag{38}$$

The properties (37) and (38) are important for the construction of w_r and $\hat{w}_r$ (for more details, see [6]).

From (14), we see that $P_r(x)$ is a quadratic polynomial. Thus $P_r'(x)$ is linear, and $P_r''(x)$ is constant. $P_r''(x)$ is computed using the following formulas:

$$P_r''(x) = \sum_{j=0}^{2} C_{rj}''(x)\bar{u}_{i-r+j} = \sum_{j=0}^{2} B_{rj}''(x)h_{3-r+j}\bar{u}_{i-r+j}. \tag{39}$$

Note that because $C_{rj}''(x)$ and $B_{rj}''(x)$ are constants, we can omit the argument x. The expression of B_{rj}'', $r = 0, 1, 2$, $j = 0, 1, 2$, can be computed as follows:

$$B_{r2}'' = \frac{6}{(h_{3-r} + h_{4-r} + h_{5-r})(h_{4-r} + h_{5-r})h_{5-r}}, \tag{40}$$

$$B_{r1}'' = B_{r2}'' - \frac{6}{(h_{3-r} + h_{4-r})h_{4-r}h_{5-r}}, \tag{41}$$

$$B_{r0}'' = B_{r1}'' + \frac{6}{h_{3-r}h_{4-r}(h_{4-r} + h_{5-r})}, \quad r = 0, 1, 2. \tag{42}$$

The second integral in (36) becomes

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (h_3)^3 (P_r''(x))^2 \mathrm{d}x = (h_3)^4 \left(\sum_{j=0}^{2} B_{rj}''(x)h_{3-r+j}\bar{u}_{i-r+j} \right)^2. \tag{43}$$

Because $P_r'(x)$ is linear in x, $(P_r'(x))^2$ is quadratic in x. Hence Simpson's quadrature rule can be used to compute the exact value for the first integral in (36); i.e.,

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h_3 (P_r'(x))^2 \mathrm{d}x = (h_3)^2 \left(\left(P_r'\left(x_{i-\frac{1}{2}}\right) \right)^2 + 4(P_r'(x_i))^2 + \left(P_r'\left(x_{i+\frac{1}{2}}\right) \right)^2 \right). \tag{44}$$

Based on (14), $P_r'(x_{i-\frac{1}{2}})$, $P_r'(x_i)$, and $P_r'(x_{i+\frac{1}{2}})$ can be obtained using

$$P_r'(x) = \frac{1}{6} \sum_{j=0}^{2} C_{rj}'(x)\bar{u}_{i-r+j} = \frac{1}{6} \sum_{j=0}^{2} B_{rj}'(x)h_{3-r+j}\bar{u}_{i-r+j}.$$

We obtain expressions for $B_{rj}'(x_{i-\frac{1}{2}})$, $B_{rj}'(x_i)$, and $B_{rj}'(x_{i+\frac{1}{2}})$ from (16):

$$B_{22}'\left(x_{i-\frac{1}{2}}\right) = \frac{2(h_1 + 2h_2)}{(h_1 + h_2 + h_3)(h_2 + h_3)h_3}, \tag{45}$$

$$B_{21}'\left(x_{i-\frac{1}{2}}\right) = B_{22}'\left(x_{i-\frac{1}{2}}\right) - \frac{2(h_1 + 2h_2 - h_3)}{(h_1 + h_2)h_2h_3}, \tag{46}$$

$$B_{20}'\left(x_{i-\frac{1}{2}}\right) = B_{21}'\left(x_{i-\frac{1}{2}}\right) + \frac{2(h_1 + h_2 - h_3)}{h_1h_2(h_2 + h_3)}, \tag{47}$$

$$B_{12}'\left(x_{i-\frac{1}{2}}\right) = \frac{2(h_2 - h_3)}{(h_2 + h_3 + h_4)(h_3 + h_4)h_4}, \tag{48}$$

$$B_{11}'\left(x_{i-\frac{1}{2}}\right) = B_{12}'\left(x_{i-\frac{1}{2}}\right) - \frac{2(h_2 - h_3 - h_4)}{(h_2 + h_3)h_3h_4}, \tag{49}$$

$$B_{10}'\left(x_{i-\frac{1}{2}}\right) = B_{11}'\left(x_{i-\frac{1}{2}}\right) + \frac{2(h_2 - 2h_3 - h_4)}{h_2h_3(h_3 + h_4)}, \tag{50}$$

$$B'_{02}\left(x_{i-\frac{1}{2}}\right) = -\frac{4(h_3+h_4)}{(h_3+h_4+h_5)(h_4+h_5)h_5} \tag{51}$$

$$B'_{01}\left(x_{i-\frac{1}{2}}\right) = B'_{02}\left(x_{i-\frac{1}{2}}\right) + \frac{2(2h_3+h_4+h_5)}{(h_3+h_4)h_4h_5}, \tag{52}$$

$$B'_{00}\left(x_{i-\frac{1}{2}}\right) = B'_{01}\left(x_{i-\frac{1}{2}}\right) - \frac{2(2h_3+2h_4+h_5)}{h_3h_4(h_4+h_5)}, \tag{53}$$

$$B'_{r2}(x_i) = B'_{r2}\left(x_{i-\frac{1}{2}}\right) + \frac{1}{2}h_3B''_{r2}, \tag{54}$$

$$B'_{r1}(x_i) = B'_{r1}\left(x_{i-\frac{1}{2}}\right) + \frac{1}{2}h_3B''_{r1}, \tag{55}$$

$$B'_{r0}(x_i) = B'_{r0}\left(x_{i-\frac{1}{2}}\right) + \frac{1}{2}h_3B''_{r0}, \tag{56}$$

$$B'_{r2}\left(x_{i+\frac{1}{2}}\right) = B'_{r2}\left(x_{i-\frac{1}{2}}\right) + h_3B''_{r2}, \tag{57}$$

$$B'_{r1}\left(x_{i+\frac{1}{2}}\right) = B'_{r1}\left(x_{i-\frac{1}{2}}\right) + h_3B''_{r1}, \tag{58}$$

$$B'_{r0}\left(x_{i+\frac{1}{2}}\right) = B'_{r0}\left(x_{i-\frac{1}{2}}\right) + h_3B''_{r0}. \tag{59}$$

Again we note that all the coefficients in (45)–(59) and (40)–(42) depend only on h_m, $m = 1,\ldots,5$. Therefore, they need to be computed only once throughout the computation provided the mesh is fixed.

2.3. Computation of the weights

Before we describe the computation of the weights for the WENO5 method, we now derive two convex combinations of $P_r(x_{i\pm\frac{1}{2}})$ that are fifth-order approximations to $u(x_{i\pm\frac{1}{2}}, t_n)$, respectively. In other words, they satisfy

$$\sum_{r=0}^{2} d_r P_r\left(x_{i-\frac{1}{2}}\right) = u\left(x_{i-\frac{1}{2}}, t_n\right) + \mathrm{O}((h_3)^5), \tag{60}$$

$$\sum_{r=0}^{2} \hat{d}_r P_r\left(x_{i+\frac{1}{2}}\right) = u\left(x_{i+\frac{1}{2}}, t_n\right) + \mathrm{O}((h_3)^5), \tag{61}$$

where d_r, $\hat{d}_r$ are positive constants, $r = 0, 1, 2$. It is easily verified that there is a unique solution for d_r and $\hat{d}_r$ if fifth-order accuracy is required.

Note that one of the stencils for the fifth-order ENO (or ninth-order WENO) methods depends on $\{I_{i-2},\ldots,I_{i+2}\}$ and has the following form:

$$\widetilde{P}(x) = \sum_{j=0}^{4} \widetilde{C}_j(x)\bar{u}_{i-2+j}, \tag{62}$$

where

$$\widetilde{C}_j(x) = \widetilde{B}_j(x)h_{3-r+j}$$

and

$$\widetilde{B}_j(x) = \sum_{m=j+1}^{5} \frac{\sum_{l=0,l\neq m}^{5}\left(\prod_{q=0,q\neq m,l}^{5}\left(x - x_{i+q-\frac{5}{2}}\right)\right)}{\prod_{l=0,l\neq m}^{5}\left(x_{i+m-\frac{5}{2}} - x_{i+l-\frac{5}{2}}\right)}.$$

We know that $\widetilde{P}(x)$ is a fifth-order approximation to $u(x, t_n)$. Because of the uniqueness of the fifth-order approximation, d_r and $\hat{d}_r$ can be obtained by solving

$$\widetilde{P}\left(x_{i-\frac{1}{2}}\right) = \sum_{r=0}^{2} d_r P_r\left(x_{i-\frac{1}{2}}\right), \tag{63}$$

$$\widetilde{P}\left(x_{i+\frac{1}{2}}\right) = \sum_{r=0}^{2} \hat{d}_r P_r\left(x_{i+\frac{1}{2}}\right). \tag{64}$$

The solution is

$$d_2 = \frac{(h_3+h_4)(h_3+h_4+h_5)}{(h_1+h_2+h_3+h_4)(h_1+h_2+h_3+h_4+h_5)}, \tag{65}$$

$$d_1 = \frac{(h_1+h_2)(h_3+h_4+h_5)(h_1+2h_2+2h_3+2h_4+h_5)}{(h_1+h_2+h_3+h_4)(h_2+h_3+h_4+h_5)(h_1+h_2+h_3+h_4+h_5)}, \tag{66}$$

$$d_0 = \frac{h_2(h_1+h_2)}{(h_2+h_3+h_4+h_5)(h_1+h_2+h_3+h_4+h_5)}, \tag{67}$$

$$\hat{d}_2 = \frac{h_4(h_4+h_5)}{(h_1+h_2+h_3+h_4)(h_1+h_2+h_3+h_4+h_5)}, \tag{68}$$

$$\hat{d}_1 = \frac{(h_1+h_2+h_3)(h_4+h_5)(h_1+2h_2+2h_3+2h_4+h_5)}{(h_1+h_2+h_3+h_4)(h_2+h_3+h_4+h_5)(h_1+h_2+h_3+h_4+h_5)}, \tag{69}$$

$$\hat{d}_0 = \frac{(h_2+h_3)(h_1+h_2+h_3)}{(h_2+h_3+h_4+h_5)(h_1+h_2+h_3+h_4+h_5)}. \tag{70}$$

We now define w_r and $\hat{w}_r$ as

$$w_r = \frac{\alpha_r}{\alpha_0+\alpha_1+\alpha_2}, \tag{71}$$

$$\hat{w}_r = \frac{\hat{\alpha}_r}{\hat{\alpha}_0+\hat{\alpha}_1+\hat{\alpha}_2}, \tag{72}$$

where

$$\alpha_r = \frac{d_r}{(\epsilon+IS_r)^2}, \tag{73}$$

$$\hat{\alpha}_r = \frac{\hat{d}_r}{(\epsilon+IS_r)^2}, \quad r=0,1,2, \tag{74}$$

where ϵ is a positive number that is introduced to avoid the denominator becoming zero. As suggested in [6], we take $\epsilon = 10^{-6}$ for the numerical experiments in this paper.

We see that the properties in (8) are satisfied. Furthermore, (71)–(74) and (37) suggest that

$$w_r = d_r + \mathrm{O}((h_3)^2), \tag{75}$$

$$\hat{w}_r = \hat{d}_r + \mathrm{O}((h_3)^2) \tag{76}$$

in regions where the solution is smooth. Eqs. (75), (76), and (5) guarantee the fifth-order accuracy of the WENO5 method in smooth regions; i.e., (6) and (7) are satisfied. On the other hand, (71)–(74) and (38) guarantee that the WENO5 method emulates the first-order, upwind method in regions where the solution exhibits discontinuous behavior.

2.4. Interpolation of point values from cell averages

When finite-volume WENO methods are used, the quantities that are evolved are the cell averages of the solution values. If we need point values at mesh points, we can use either (12) or (13) to approximate them. However, because a non-uniform mesh is used, point values for the solution inside cells may be required. For example, to plot the numerical solution requires an accurate interpolation from cell averages to point values.

Assume $x_i < \tilde{x} < x_{i+1}$. There are three possible interpolants; i.e., $P_r(\tilde{x})$, $r = 0, 1, 2$, where $P_r(x)$ is the quadratic polynomial defined in (4) (we omit the superscript (i) without loss of clarity). We can perform a similar procedure as is performed at the cell boundary in the WENO5 reconstruction; i.e., form a linear combination of the $P_r(\tilde{x})$ with weights $w_r(\tilde{x})$. Each $P_r(\tilde{x})$ is computed from (14)–(16), and the $w_r(\tilde{x})$ are obtained in a manner similar to what is described in Section 2.3. However, we note that, unlike the weights in the cell boundary calculation, the $w_r(\tilde{x})$ are not guaranteed to be positive numbers. In other words, this procedure does not necessarily produce a convex combination.

3. Numerical results

In this section, we study two classical scalar conservation laws: the advection equation and the Burgers equation. In both cases, the finite-volume WENO5 method is employed as the spatial discretization. Also we always use the three-stage, order-3, strong–stability–preserving (SSP) explicit Runge–Kutta method (which we call SSP(3,3) [17,18]) for the time discretization. We illustrate the efficiency of the non-uniform mesh approach by comparing the results with those obtained by using a uniform mesh. We compare the results by computing the solution at a given time T_{out} with a specified Courant number

$$\sigma = \left(\max \frac{\partial f}{\partial u}\right) \frac{\Delta t}{\min_{i=1}^{N} \Delta x_i}. \tag{77}$$

Accuracy is measured by means of the L^1-norm error

$$\begin{aligned} \|Error\|_1 = \frac{1}{x_{N+\frac{1}{2}} - x_{\frac{1}{2}}} \Bigg(& \left|\tilde{u}\left(x_{\frac{1}{2}}, T_{\text{out}}\right) - u\left(x_{\frac{1}{2}}, T_{\text{out}}\right)\right| \frac{\Delta x_1}{2} + \sum_{i=1}^{N-1} \left|\tilde{u}\left(x_{i+\frac{1}{2}}, T_{\text{out}}\right) - u\left(x_{i+\frac{1}{2}}, T_{\text{out}}\right)\right| \frac{\Delta x_i + \Delta x_{i+1}}{2} \\ & + \left|\tilde{u}\left(x_{N+\frac{1}{2}}, T_{\text{out}}\right) - u\left(x_{N+\frac{1}{2}}, T_{\text{out}}\right)\right| \frac{\Delta x_N}{2} \Bigg), \end{aligned} \tag{78}$$

where, respectively, $\tilde{u}(x_{i+\frac{1}{2}}, T_{\text{out}})$ is the numerical solution and $u(x_{i+\frac{1}{2}}, T_{\text{out}})$ is the exact solution at $x = x_{i+\frac{1}{2}}$ and $t = T_{\text{out}}$. The point values $\tilde{u}(x_{i+\frac{1}{2}}, T_{\text{out}})$ are obtained using the interpolation described in Section 2.4. More precisely, we use (13) for the interpolation of $\tilde{u}(x_{i+\frac{1}{2}}, T_{\text{out}})$, $i = 1, \ldots, N$, and (12) for the interpolation of $\tilde{u}(x_{\frac{1}{2}}, T_{\text{out}})$.

Example 1. The first example is the linear advection equation

$$u_t + u_x = 0, \quad 0 \leqslant x \leqslant 2, \quad t > 0$$

with periodic boundary conditions. We consider two different initial conditions. The first is the smooth initial condition $u(x, 0) = \sin(\pi x)$. This well-behaved problem is used to illustrate the fifth-order accuracy of the method. We set the mesh to be

$$x_{i+\frac{1}{2}} = \begin{cases} \frac{3i}{N}, & \text{if } 0 \leqslant i < \frac{N}{3}, \\ \frac{3i}{2N}, & \text{if } \frac{N}{3} \leqslant x \leqslant N. \end{cases}$$

That is, the mesh points are uniformly located in $0 \leqslant x \leqslant 1$ and in $1 \leqslant x \leqslant 2$ but with different spacings. The CFL number, σ, is chosen to be 0.1, so that the temporal error does not dominate the spatial error. Table 1 demonstrates that the observed convergence rate agrees with that predicted by theory.

Table 1
L^1-norm errors and convergence rates

N	$\|Error\|_1$	Order
30	4.6570E−4	–
60	1.6403E−5	4.8274
120	5.3608E−7	4.9354
240	1.7447E−8	4.9414

The second initial condition is the step function

$$u(x,0)=\begin{cases}1, & \text{if } 0<x<0.5 \text{ or } 1.5<x<2,\\ 0, & \text{if } 0.5\leqslant x\leqslant 1.5.\end{cases} \tag{79}$$

There are two shocks present in the exact solution. We compute the solution at $T_{\text{out}}=0.1$. Therefore, the location of the first shock moves from $x=0.5$ to $x=0.6$, and the location of the second one moves from $x=1.5$ to $x=1.6$. The mesh is carefully chosen with this in mind. We partition the domain into three types of regions; i.e., coarse-mesh regions, fine-mesh regions, and intermediate-mesh regions. Every intermediate-mesh region is located between a coarse-mesh region and a fine-mesh region. The cell sizes are gradually increased from the fine-mesh region to the coarse-mesh region in the intermediate-mesh regions. We first place fine uniform meshes in the intervals $[0.5,0.6]$ and $[1.5,1.6]$ with cell width 10^{-3}; i.e., there are 100 points in each interval. We place coarse meshes in the intervals $[0,0.428]$, $[0.672,1.428]$, and $[1.672,2]$. We place 25 uniform intervals in $[0,0.428]$ and $[1.672,2]$ (i.e., the cell width is 0.0171) and 50 uniform intervals in $[0.672,1.428]$ (i.e., the cell width is 0.0151). The four remaining regions are intermediate-mesh regions. In each intermediate-mesh region, we place 15 intervals in such a way that the cell widths are increased by 20% from one cell to the next in the direction from the fine-mesh region to the coarse-mesh region. In summary, the mesh is defined in terms of Δx_i as follows:

$$\Delta x_i=\begin{cases}0.0171, & \text{if } 1\leqslant i\leqslant 25;\ \text{i.e., } x_{i+\frac{1}{2}}\in[0,0.428];\\ 0.001\times(1.2)^{40-i}, & \text{if } 26\leqslant i\leqslant 40;\ \text{i.e., } x_{i+\frac{1}{2}}\in(0.428,0.5];\\ 0.001, & \text{if } 41\leqslant i\leqslant 140;\ \text{i.e., } x_{i+\frac{1}{2}}\in(0.5,0.6];\\ 0.001\times(1.2)^{i-141}, & \text{if } 141\leqslant i\leqslant 155;\ \text{i.e., } x_{i+\frac{1}{2}}\in(0.6,0.672];\\ 0.0151, & \text{if } 151\leqslant i\leqslant 205;\ \text{i.e., } x_{i+\frac{1}{2}}\in(0.672,1.428];\\ 0.001\times(1.2)^{220-i}, & \text{if } 206\leqslant i\leqslant 220;\ \text{i.e., } x_{i+\frac{1}{2}}\in(1.428,1.5];\\ 0.001, & \text{if } 221\leqslant i\leqslant 320;\ \text{i.e., } x_{i+\frac{1}{2}}\in(1.5,1.6];\\ 0.001\times(1.2)^{i-321}, & \text{if } 321\leqslant i\leqslant 335;\ \text{i.e., } x_{i+\frac{1}{2}}\in(1.6,1.672];\\ 0.0171, & \text{if } 336\leqslant i\leqslant 360;\ \text{i.e., } x_{i+\frac{1}{2}}\in(1.672,2].\end{cases}$$

Fig. 1 shows the solutions at $T_{\text{out}}=0.1$. The solid line is the exact solution. The crosses represent the values of the numerical solution at the mesh points of the above non-uniform mesh for $\sigma=0.5$.

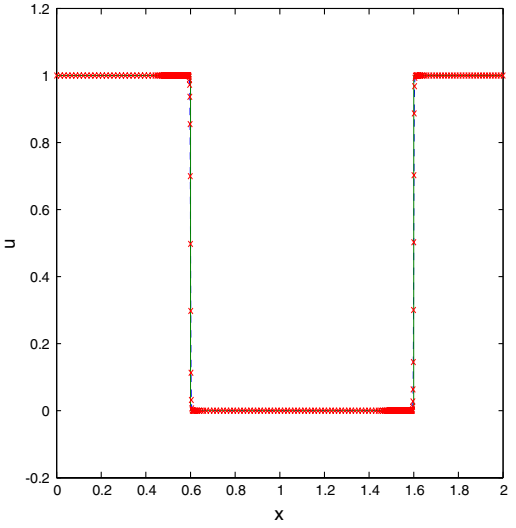

Fig. 1. The discontinuous solution at $T_{\text{out}}=0.1$.

Table 2
L^1-norm errors and CPU times for non-uniform and uniform meshes

σ	Non-uniform	Uniform		
	$N = 360$	$N = 500$	$N = 1000$	$N = 2000$
1.0	0.0045/3.82	0.0099/0.75	0.0066/2.65	0.0045/15.46
0.5	0.0016/6.24	0.0050/1.28	0.0028/5.21	0.0016/30.77
0.25	0.0016/11.19	0.0050/2.47	0.0028/10.27	0.0016/60.82

In order to compare the results from using a non-uniform mesh versus a uniform one, we compute the L^1-norm error as in (78) and the corresponding CPU time in seconds at $T_{\text{out}} = 0.1$. Different CFL numbers, σ, are chosen as well as different values of N for the uniform mesh. Table 2 shows the L^1-norm error and the CPU time in the form of A/B, where A is the L^1-norm error, and B is the CPU time.

From Table 2, we conclude that the use of the WENO5 method with a non-uniform mesh is more efficient than with a uniform mesh. For example, in order to achieve an L^1-norm error of 0.0016, a non-uniform mesh with only 360 intervals is needed compared with 2000 intervals for a uniform mesh. Furthermore, in this example the solution is generated about five times faster using a non-uniform mesh versus using a uniform one.

We have performed extensive tests on problems where shocks are present throughout the entire simulation. We have found that if it is possible to place a fine mesh that captures the location of the shock throughout the entire simulation, using a non-uniform mesh is much more efficient, both in terms of computational efficiency as well as in terms of storage requirements, than using a uniform mesh.

Example 2. The second example is the Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad 0 < x < 2, \;\; t > 0$$

subject to the initial condition $u(x,0) = \sin(\pi x)$ and periodic boundary conditions. This is a nonlinear problem of the form (1) with $f(u) = u^2/2$. Thus $\partial f/\partial u = u$. For a given σ, Δt is chosen as $\Delta t = \sigma \min_{i=1}^{N} \frac{\Delta x_i}{|u_i|}$. The exact solution of this problem does not have a closed form. A reference solution is generated using WENO5 with a fixed uniform mesh of size $N = 10{,}000$ and a CFL number $\sigma = 0.1$.

The solution of this problem evolves from a smooth initial condition to having a shock at $x = 1$. After the shock is formed, its location is fixed, and its height decreases. The non-uniform mesh is now chosen carefully using $N = 150$ intervals. Because the location of the shock does not change, we describe the construction of the non-uniform mesh starting from $x_{\frac{N+1}{2}} = 1$. To the right of $x_{\frac{N+1}{2}}$, 15 intervals are located in such a way that the first cell size is $1.d - 3$, and the cell sizes are increased by 20% as we proceed to the right, terminating with $x_{\frac{N+31}{2}} = 1.0720$. We then put 60 uniform mesh cells between $[1.0720, 2]$; i.e., the cell size is 0.0155. The mesh to the left of $x_{\frac{N+1}{2}}$ is defined similarly. In summary, the mesh is defined in terms of Δx_i as follows:

$$\Delta x_i = \begin{cases} 0.0155, & \text{if } 1 \leqslant i \leqslant 60; \text{ i.e., } x_{i+\frac{1}{2}} \in [0, 0.9280]; \\ 0.001 \times (1.2)^{75-i}, & \text{if } 61 \leqslant i \leqslant 75; \text{ i.e., } x_{i+\frac{1}{2}} \in (0.9280, 1]; \\ 0.001 \times (1.2)^{i-76}, & \text{if } 76 \leqslant i \leqslant 90; \text{ i.e., } x_{i+\frac{1}{2}} \in (1, 1.0720]; \\ 0.0155, & \text{if } 91 \leqslant i \leqslant 150; \text{ i.e., } x_{i+\frac{1}{2}} \in (1.0720, 2]. \end{cases}$$

Fig. 2 shows the solutions at $T_{\text{out}} = 1$. The solid line is the reference solution. The crosses represent the numerical solution at the mesh points with the above non-uniform mesh using $\sigma = 0.5$.

In this example, we also use non-uniform meshes with different values of N. For example, the non-uniform mesh with $N = 300$ is generated by halving each interval of the non-uniform mesh with $N = 150$; similarly the non-uniform mesh with $N = 450$ is generated by dividing each interval of the non-uniform mesh with $N = 150$ into three equally spaced intervals. In order to compare the results from using the non-uniform mesh versus

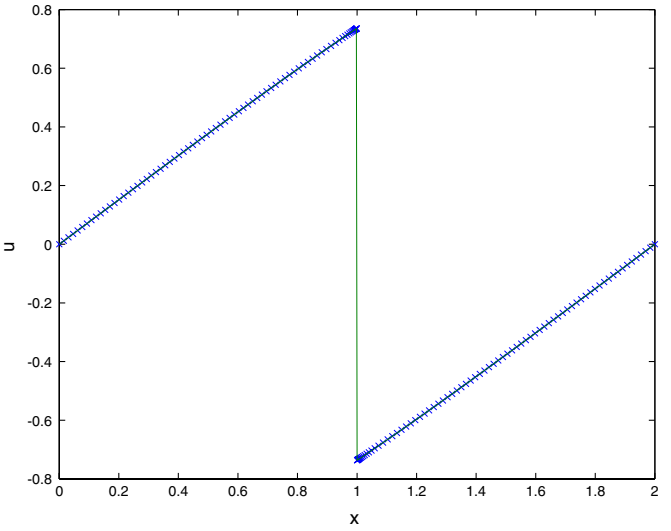

Fig. 2. The solution for the Burgers equation at $T_{\text{out}} = 1$.

using the uniform one, we compute the L^1-norm error as in (78) and the corresponding CPU time in seconds at $T_{\text{out}} = 1$. Different CFL numbers, σ, are chosen, and different values of N for both the uniform mesh and non-uniform mesh are used. Tables 3 and 4 show the L^1-norm error and the CPU time in the form of A/B, where A is the L^1-norm error, and B is the CPU time.

Based on these results, we make the following observations.

- When a non-uniform mesh is used, the error does not decrease significantly for all σ studied. In other words, when this happens the spatial error dominates the temporal error. This can be attributed to the fact that Δt is very small; we recall that it is equal to the product of σ and the smallest value of $\frac{\Delta x_i}{|u_i|}$, which is achieved at the location of the shock; in fact, it has the smallest Δx_i and the largest $|u_i|$. On the other hand, when a uniform mesh is used, the error decreases steadily for the range of σ shown. In other words, the spatial error is significantly smaller than the temporal error. This is because $\Delta t = \frac{\Delta x}{\max_{i=1}^{N}|u_i|}$ is considerably larger than it is for the non-uniform meshes.
- It is not apparent that there is much advantage in terms of computational efficiency for using a non-uniform mesh with this problem. For example, the error is 5.99×10^{-12} when using the non-uniform mesh with $N = 450$ and $\sigma = 0.5$; the corresponding CPU time is 199.7 s. This result is comparable with that when

Table 3
L^1-norm errors and corresponding CPU times with non-uniform meshes

σ	$N = 150$	$N = 300$	$N = 450$
1.0	1.4584E−9/11.7	5.6991E−11/45.5	1.2151E−11/100.1
0.5	1.3498E−9/22.6	4.3537E−11/91.5	5.9872E−12/199.7
0.25	1.3364E−9/45.3	4.1860E−11/181.1	5.4909E−12/401.6

Table 4
L^1-norm errors and corresponding CPU times with uniform meshes

σ	$N = 500$	$N = 1000$	$N = 2000$
1.0	1.9656E−8/5.9	2.4311E−9/25.5	3.0224E−10/148.9
0.5	2.4297E−9/11.5	3.0213E−10/50.3	3.7670E−11/305.9
0.25	3.0702E−10/23.2	3.7975E−11/100.6	4.7196E−12/584.4
0.125	5.3284E−11/46.0	5.6795E−12/200.6	6.4675E−13/1207.1

using a uniform mesh with $N = 1000$ and $\sigma = 0.125$; i.e., the error is 5.68×10^{-12}, and the CPU time is 200.6 s. If the shock location does not change (significantly), using a non-uniform mesh may not necessarily be superior to using a uniform one in terms of CPU time. However, if the shock location does change, it is clear that an adaptive mesh strategy that could place a fine mesh only in the immediate area of the shock location would have significant computational advantages over a strategy where a fine uniform mesh would have to be used everywhere in the spatial domain. Moreover, it is noteworthy that, despite the lack of clear computational efficiency advantages for this simple example, there are nonetheless distinct advantages in terms of reduced storage requirements when a non-uniform mesh is used.

In this example, the solution starts from a smooth function and ends with a shock whose location is fixed. This means that in order to maintain a very small error, e.g., 10^{-10}, throughout the computation, many mesh points must be used at the beginning. That is, if the computation does not begin with a sufficiently fine mesh, the errors generated at early times adversely affect the accuracy of the solution at later times. When the shock begins to form, the overall number of mesh points may be reduced without adversely affecting the overall error provided that a sufficiently fine mesh is placed in the vicinity of the shock. This is the reason why the use of non-uniform meshes does not outperform the use of uniform meshes for this example. This example also suggests that if an adaptive mesh strategy is to be effectively used with the finite-volume WENO5 method, it must have the ability to adaptively change the number of mesh points used in order to maintain a certain spatial error.

4. Conclusions and future work

In this paper, we give explicit formulas for the implementation of the finite-volume WENO5 method on an arbitrary (non-uniform) one-dimensional mesh. We compare the performance of using non-uniform meshes with that of the classical finite-difference version of the WENO5 method using uniform meshes for one-dimensional, scalar hyperbolic conservation laws in terms of computational efficiency. By means of numerical experiments on linear and nonlinear problems with shock-like solutions, we find that using non-uniform meshes can be significantly more efficient than using uniform meshes both in terms of computation time and memory required. However, we also conclude that in order for an adaptive mesh strategy for the WENO5 spatial discretization to succeed, it is critical for it to have the ability to add or remove mesh points at different time steps. We hope that these formulas and observations can be used as a starting point for the future development of an adaptive strategy for the finite-volume WENO5 method.

References

[1] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes I, SIAM J. Numer. Anal. 24 (1987) 270–309.
[2] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231–303.
[3] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[4] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, J. Comput. Phys. 83 (1989) 32–78.
[5] X.-D. Liu, S. Osher, C. Tony, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[6] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[7] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, vol. 1697, Springer-Verlag, Berlin, 1998, pp. 325–432.
[8] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (2002) 108–127.
[9] W. Huang, Y. Ren, R. Russell, Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, SIAM J. Numer. Anal. 31 (1994) 709–730.
[10] Y. Di, R. Li, T. Tang, P. Zhang, Moving mesh finite element methods for the incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 26 (3) (2005) 1036–1056.

[11] J. Stockie, J. Mackenzie, R. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws, SIAM J. Sci. Comput. 22 (5) (2000) 1791–1813.
[12] D. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, J. Comput. Phys. 160 (2000) 405–452.
[13] C.-W. Shu, High order ENO and WENO schemes for computational fluid dynamics, in: High-order Methods for Computational Physics, Lecture Notes in Computational Science and Engineering, vol. 9, Springer-Verlag, Berlin, 1999, pp. 439–582.
[14] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys. 144 (1998) 194–212.
[15] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
[16] B. van Leer, On the relation between the upwind-differencing schemes of Godunov, Engquist–Osher and Roe, SIAM J. Sci. Statist. Comput. 5 (1) (1984) 1–20.
[17] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability–preserving time discretization methods, SIAM Rev. 43 (2001) 89–112.
[18] R. Spiteri, S. Ruuth, A new class of optimal high-order strong–stability–preserving time discretization methods, SIAM J. Numer. Anal. 40 (2002) 469–491.