Journal of Computational Physics 230 (2011) 3780–3802

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations

Chaopeng Shen [a,*,1], Jing-Mei Qiu [b], Andrew Christlieb [c]

[a] Department of Civil and Environmental Engineering, Michigan State University, East Lansing, 48824, United States
[b] Department of Mathematical and Computer Science, Colorado School of Mines, Golden, 80401, United States
[c] Department of Mathematics, Michigan State University, East Lansing, 48824, United States

ARTICLE INFO

Article history:
Received 8 July 2010
Received in revised form 5 February 2011
Accepted 8 February 2011
Available online 13 February 2011

Keywords:
Adaptive mesh refinement (AMR)
WENO
High order finite difference
Multiscale simulations

ABSTRACT

In this paper, we propose a finite difference AMR-WENO method for hyperbolic conservation laws. The proposed method combines the adaptive mesh refinement (AMR) framework [4,5] with the high order finite difference weighted essentially non-oscillatory (WENO) method in space and the total variation diminishing (TVD) Runge–Kutta (RK) method in time (WENO-RK) [18,10] by a high order coupling. Our goal is to realize mesh adaptivity in the AMR framework, while maintaining very high (higher than second) order accuracy of the WENO-RK method in the finite difference setting. The high order coupling of AMR and WENO-RK is accomplished by high order prolongation in both space (WENO interpolation) and time (Hermite interpolation) from coarse to fine grid solutions, and at ghost points. The resulting AMR-WENO method is accurate, robust and efficient, due to the mesh adaptivity and very high order spatial and temporal accuracy. We have experimented with both the third and the fifth order AMR-WENO schemes. We demonstrate the accuracy of the proposed scheme using smooth test problems, and their quality and efficiency using several 1D and 2D nonlinear hyperbolic problems with very challenging initial conditions. The AMR solutions are observed to perform as well as, and in some cases even better than, the corresponding uniform fine grid solutions. We conclude that there is significant improvement of the fifth order AMR-WENO over the third order one, not only in accuracy for smooth problems, but also in its ability in resolving complicated solution structures, due to the very low numerical diffusion of high order schemes. In our work, we found that it is difficult to design a robust AMR-WENO scheme that is both conservative and high order (higher than second order), due to the mass inconsistency of coarse and fine grid solutions at the initial stage in a finite difference scheme. Resolving these issues as well as conducting comprehensive evaluation of computational efficiency constitute our future work.

1. Introduction

Recently high order (higher than second order) methods have attracted increasing attention from many computational fields of study. High order schemes are often considered expensive, unstable and unnecessary. However, it is argued in some studies that the expenses of high order schemes are compensated by their high order accuracy and very low numerical

[*] Corresponding author.
E-mail addresses: cshen@lbl.gov (C. Shen), jingqiu@mines.edu (J.-M. Qiu), Christlieb@math.msu.edu (A. Christlieb).
[1] Now at: Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States. Tel.: +1 510 486 7286.

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.02.008

diffusion, because high order methods can resolve solution structures with relatively coarse numerical meshes. As the error tolerance for the problem gets very small, or the integration time gets very long in a time-dependent problem, high order schemes often become more efficient than lower order ones [27]. Shu [29] demonstrated that high order schemes can resolve solution structures which are impractically expensive to obtain if a low order scheme is used, specifically when the solution structures become very complicated in the long time setting. However, the accusation that high order schemes are still more or less valid, as more computational dollars per node are needed by a higher order scheme. One can not use an arbitrarily coarse grid because the mesh needs to be fine enough to resolve the physical phenomena of interest. Popular classes of high order methods for hyperbolic equations include the finite difference or finite volume WENO method [31,18], the finite element discontinuous Galerkin (DG) method [11,12], the spectral method [15], the space time conservative element and the solution element method [8,9] and so on.

Adaptive mesh refinement (AMR) was originally developed by Berger et. al. [5,6]. The method refines the mesh locally to focus computational effort where it is most needed. AMR has found popularity in a wide range of fields such as computational fluid dynamics, astrophysics, oceanography, biophysics and many others [26]. In particular, it has been shown that the AMR method is advantageous for physical systems with vastly different spatial scales. For example, the evolution of a hyperbolic equation often leads to local shocks, near which numerical methods can have large errors. These errors might propagate and further contaminate the solution across the entire domain. Moreover, the solution structures of different scales might interact, and the failure to address features on one scale can jeopardize the quality of the entire solution. The multi-resolution nature of the AMR algorithm makes it ideal for these cases, where computational savings can be enormous [5]. There have been many efforts aimed at further developing the AMR algorithm, for example, for high dimensional problems [3], for use with unstructured meshes [23], for coupling with high order integrators [6,27,1,22], and for elliptic equations [2]. A detailed comparison of AMR versus high order schemes is given in [17] for a range of different problems. It was concluded in [17] that the AMR scheme will be advantageous if the AMR region is below a certain proportion of the entire domain; and that it is advantageous to have an AMR scheme with as high order as the regularity of the PDE.

Given the above mentioned features of the AMR algorithm, it is natural to combine the AMR algorithm with numerical schemes that are highly accurate, robust and efficient. There have been many research efforts in this direction. Sebastian & Shu [28] introduced a multi-domain WENO scheme but the refinement region was fixed and therefore not adaptive. Li [22] combined high order finite difference WENO with AMR. However, a linear prolongation in space was used, which reduces the scheme to second order accurate. Further, the scheme in [22] does not have temporal refinement, thus the temporal step size is limited by the CFL restriction of the finest mesh. Baeza [1] also described an AMR-WENO scheme, but the method was only second order accurate in time. There has also been much effort in designing adaptive versions of the DG and spectral methods for hyperbolic problems [14,16,34,24,33,27]. In the present work, we will focus on designing an adaptive finite difference WENO method. Compared with the DG and spectral methods, the finite difference WENO methods, when combined with the total variation diminishing (TVD) Runge–Kutta (RK) method in time, are more robust in resolving discontinuous shock structures.

In this paper, we couple the AMR framework with a high order finite difference WENO-RK scheme. The finite difference WENO-RK scheme is well-known for its high order accuracy, robustness and efficiency, as well as its straightforward extension to multi-dimensional problems (dimension by dimension) compared with finite volume schemes. There are a few major roadblocks that need to be cleared in order to realize a high order coupling. First, it is difficult to maintain high order accuracy across several levels of grids. In the AMR algorithm, the fine mesh solutions are interpolated from the coarse mesh solutions. Solutions at boundary points are needed not only at the current time but also at intermediate fine grid time steps (subcycles) as well as RK sub-stages. The same temporal and spatial accuracy are required in the data prolongation procedures as those of the base integrator. Secondly, introducing a high order accurate scheme in the AMR setting makes it harder to maintain local mass conservation. Thirdly, robust and inexpensive refinement criteria are difficult to find. In the original AMR [6], the refinement criterion relies on Richardson extrapolation (RE). In the finite difference WENO method, nonlinear WENO weights could be used as a measure of solution smoothness, hence providing an inexpensive refinement criterion. Alternatively, the difference between solutions on a coarse and fine grid can be used as a refinement criterion. Karni [19] also suggested a smoothness indicator based on weak local truncation error of the numerical solution, which can be used as a refinement criterion.

In the discussion that follows, we focus on the high order aspect of the proposed AMR-WENO algorithm. We compare different implementations of the AMR-WENO scheme, e.g. with integrators of different orders (third and fifth order), with different refinement criteria (RE and WENO weights), and with different data restriction procedures (a point value replacement and a conservative update). The classical AMR procedure is briefly reviewed in Section 2. We describe the proposed AMR-WENO scheme in Section 3. In order to describe our algorithm clearly, most of our discussions are based on a 1D hyperbolic equation,

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

The 1D algorithm can be generalized to high-dimensional problems without much technical difficulty. In Section 4, computational results for several 1D and 2D hyperbolic problems are demonstrated, illustrating the high order accuracy, robustness and efficiency of the proposed AMR-WENO scheme. Conclusions are given in Section 5.

2. Review of AMR

The AMR algorithm was developed for hyperbolic conservation laws [4,5] in order to focus the mesh resolution and computational effort around complicated solution structures. To save space and avoid repetition, we will only give a brief outline of the AMR algorithm in this section. A complete description of the original AMR algorithm is quite involved, with many subtle implementation techniques. Readers are referred to [4,5] for detailed descriptions.

The AMR algorithm starts with a collection of logically rectangular meshes that make up the coarse grid. Depending on the problem specification, at each regridding time, the algorithm scans the solution (typically by an error estimation technique, e.g., the RE) and tags the regions where finer resolution is needed for fine scale features. Buffer zones are created by flagging coarse grid points that are sufficiently close to the tagged regions. The tagged regions are organized into logically rectangular patches, which are populated by either copying data from existing fine grid solutions or by prolongation from coarse grid solutions. Boundary values for fine grid evolution are prolongated from coarse grid solutions in both space and time. Refinement patches are recursively nested in the coarse grid, until a given level of accuracy is attained. The time step is also refined with the same ratio as the spatial mesh. The fine grid solutions obtained at the end of one time step are used to correct (or 'update') the coarse grid solutions, therefore achieving better accuracy and resolution of small scale structures. In the original second order algorithm [4,5], the solutions are evolved using a second order conservative finite volume method. A second order linear interpolation in both space and time is used for data prolongation. The conservation of mass, between the fine and coarse grids, is ensured by a flux correction step [5].

3. AMR-WENO

In this section, we introduce the proposed high order finite difference AMR-WENO algorithm by describing the computational grid structure in Section 3.1, the general framework of AMR in Section 3.2, and details of numerical integration, detection of refinement regions, data prolongation in both space and in time, and data restriction in Sections 3.3–3.7 respectively. We briefly outline the multi-dimensional implementation and data management in Section 3.8.

3.1. Grid description

The spatial domain is first discretized into a coarse, uniform computational grid called the 'root grid'. Finer meshes, called the child of the root grid, can be created to focus computational efforts on parts of the domain. Finer and finer meshes inside the child grids can be created until some prescribed criteria are satisfied. These grids can be organized into a tree/hierarchy structure as shown in Fig. 1. Each non-root grid may have zero to multiple children but only one unequivocal parent. We use G_k^l to denote a grid, represented by a node on the tree, where k is the node index in the tree and $l = 0, 1, \ldots, l_{max}$ marks the level. Strict grid nesting is enforced such that any grid on level l, say G_k^l, must be contained completely within its parent grid on level $l-1$. We use G^0 to denote the root grid, use G_{kp}^{l-1} to denote the parent of G_k^l, and use $G_{kc_j}^{l+1}$ where $j = 1, 2, \ldots, q$ to denote its q children. Each node in the tree, say G_k^l, corresponds to a uniform rectangular mesh with $\mathbf{x}_k^l$ as the coordinates for the grid points in the mesh, and Δx_k^l and Δt_k^l as the spatial and temporal mesh sizes. The refinement ratio, denoted as r, is always set to be an odd number. For example, Fig. 2 shows the grid refinement with $r = 3$. Such design of nesting grids is different from [5] and is convenient in the finite difference setting. Specifically, a simple spatial correspondence between grid points at different levels is automatically established, so that a direct point replacement is possible in the data restriction step described in Section 3.7. Such a configuration also ensures consistency in the flux location between coarse and fine grids.

3.2. Flow map of AMR-WENO

The AMR-WENO algorithm is summarized as a flow map in Fig. 3. The algorithm revolves around a procedure, hereby called 'gMarch', which is recursively called upon for all active grids. We use gMarch (l,k) for grid G_k^l as an example to

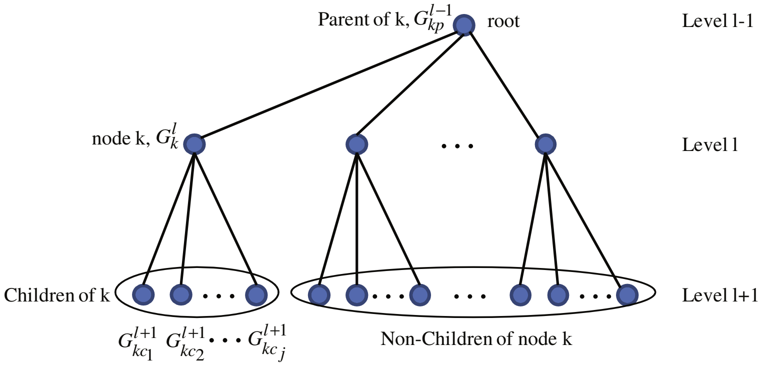

Fig. 1. Grid structure for AMR-WENO (adapted from [25]).

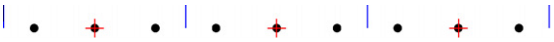

Fig. 2. The AMR grid nesting with refinement ratio $r = 3$. The coarse grids are marked by red crosses ('+') and the fine grids are marked by black solid circles. '|' are the cell boundaries of the cell coarse grid.

illustrate the flow map. The gMarch first invokes a numerical integrator, e.g. a finite difference WENO scheme described in Section 3.3, to march the solution over a time step Δt_k^l. The algorithm then checks if certain simple criteria are met (the 'norefine' in Fig. 3), e.g., whether the maximum level of refinement has been reached or whether the grid spacing is less than a threshold. If 'norefine' is true, the algorithm finishes its time step. Otherwise, the algorithm begins to scan the solution vector and locate regions of refinement interest (RRI), see Section 3.4. Finer meshes as child grids of G_k^l are then created within each of RRI, housing refined solution vectors obtained either by using existing fine grid solutions or by data prolongation. The data prolongation provides the initial and boundary conditions for child grids of G_k^l by high order spatial and temporal interpolation (upper left corner in Fig. 3), described in Sections 3.5 and 3.6. After these preparations, gMarch calls a numerical integrator and marches each of the child grid solutions for Δt_k^l for several finer time steps (lower left corner in Fig. 3). In the end, the solutions from the child grids are used to update the solution on G_k^l, which is called the data restriction step, described in Section 3.7. The basic framework in the flow map is similar to that in [4,5]. However, the details for numerical integration and high order coupling between AMR and finite difference WENO, described in later subsections, are quite different.

3.3. Numerical integration

A key component of the AMR-WENO is to use the fifth order finite difference WENO scheme coupled with the TVD RK method as the base numerical integrator. Below, we give a brief review of it. We refer readers to [10,30] for more details.

For a particular grid G_k^l encompassing spatial domain $[a,b]$, we adopt the following spatial discretization

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = b, \tag{2}$$

where $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, $i = 1,\ldots,N$ are uniform numerical cells with centers $x_i = \frac{1}{2}\left(x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}}\right)$ and cell sizes $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} = (b-a)/N$. We use $u_i = u(x_i, t)$ to denote the point value of the solution at $x = x_i$. $\{x_i\}_{i=1}^N$ are collectively called $\mathbf{x}_k^l$. Similarly, $\{u_i\}_{i=1}^N$ are termed $\mathbf{u}_k^l$. The finite difference scheme evolves the point values of the solution u_i, $i = 1,\ldots,N$, by approximating the differential form of Eq. (1) directly. The scheme is of conservative form

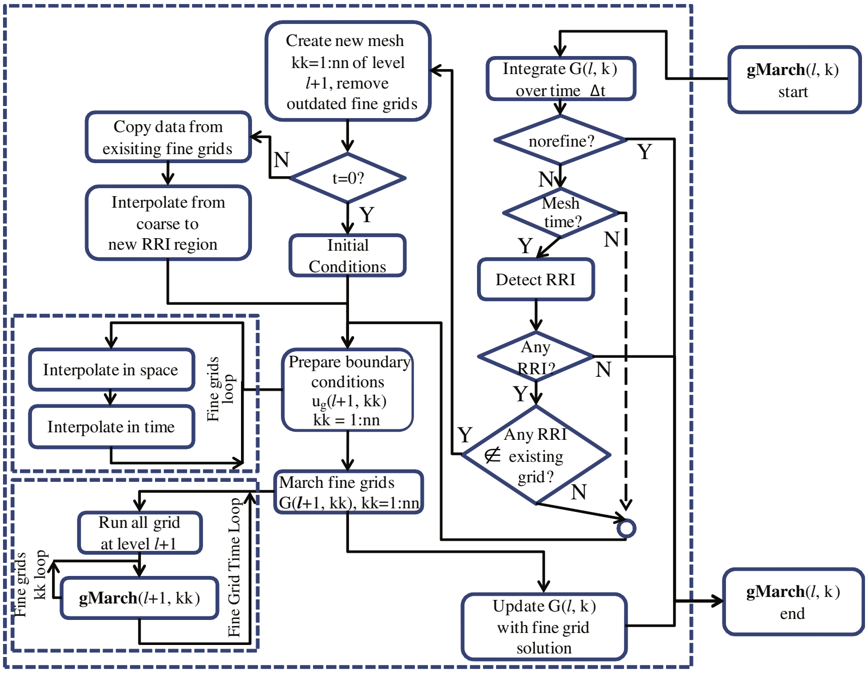
Create new mesh kk=1:nn of level l+1, remove outdated fine grids
Integrate G(l, k) over time Δt
gMarch(l, k) start
Copy data from exisiting fine grids
N
t=0?
Y
norefine?
Y
N
Interpolate from coarse to new RRI region
Initial Conditions
Mesh time?
N
Y
Detect RRI
Interpolate in space
Interpolate in time
Fine grids loop
Prepare boundary conditions u_g(l+1, kk) kk = 1:nn
Any RRI?
N
Y
Y
Any RRI ∉ existing grid?
N
March fine grids G(l+1, kk), kk=1:nn
Run all grid at level l+1
gMarch(l+1, kk)
Fine grids kk loop
Fine Grid Time Loop
Update G(l, k) with fine grid solution
gMarch(l, k) end

Fig. 3. Flow map of the AMR-WENO algorithm. In the diagram, 'gMarch' is an AMR marching procedure that is recursively called upon in the hierarchy grid structure as shown in Fig. 1; $G(l,k)$ is the numerical solution, corresponding to the node G_k^l in the tree; 'RRI' refers to regions of refinement interest discussed in Section 3.4; $\mathbf{u}_g(l+1,kk)$ stands for the solutions at ghost points of the child grid G_{kk}^{l+1}.

$$\frac{d}{dt}u_i = -\frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right), \tag{3}$$

where the numerical flux $\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_{i-p}, \ldots, u_{i+q})$ is consistent with the physical flux $f(u)$ and is Lipschitz continuous with respect to all arguments. The stencil $\{u_{i-p}, \ldots, u_{i+q}\}$ is chosen to be upwind biased. Specifically, when $f'(u) \geqslant 0$, one more point from the left ($p = q$) will be taken to reconstruct $\hat{f}_{i+\frac{1}{2}}$; when $f'(u) < 0$, one more point from the right ($p = q - 2$) will be taken. When $f'(u)$ changes sign over the domain, then a flux splitting, e.g. the Lax–Friedrichs flux splitting can be applied [30]. The spatial accuracy of the scheme is determined by how well $\frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}})$ approximates $f(u)_x$. To obtain a high order approximation, Shu and Osher [31] introduced a sliding average function $h(x)$, such that

$$\frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi)d\xi = f(u), \tag{4}$$

for a fixed t. Taking the x derivative of the above equation gives

$$\frac{1}{\Delta x}\left(h(x+\frac{\Delta x}{2}) - h\left(x - \frac{\Delta x}{2}\right)\right) = f(u)_x. \tag{5}$$

Therefore the numerical flux $\hat{f}_{i+\frac{1}{2}}$ in Eq. (3) can be taken as $h(x_{i+\frac{1}{2}})$, which can be reconstructed from neighboring cell averages of $h(x)$, $\bar{h}_j = \frac{1}{\Delta x}\int_{I_j} h(\xi)d\xi \overset{(4)}{=} f(u_j)$, $j = i - p, \ldots, i + q$ by WENO reconstruction [30]. Eq. (3) is further discretized in time by a method of line (MOL) procedure, the most popular of which is the third order TVD RK method [13]. The finite difference WENO scheme can be easily extended to higher dimensions using a dimension-by-dimension approach. The initial and boundary conditions for G_k^l are obtained either from given physical conditions, or from existing fine grid solutions, or from prolongation from its parent grid as described in Sections 3.5 and 3.6.

3.4. Detection of RRI

The need for refinement may come from different aspects of the solution, such as non-smoothness, small scale features, variable physical parameters and so on. We attempted two different approaches in our numerical experiments. One is based on the error estimate of RE [5], and the other is based on using WENO weights as an indicator for the smoothness of the solution structure. The rationale for using WENO weights as smoothness indicators is partly that they have already been computed in the numerical integration step, allowing us to detect RRI at low cost. However, we found it difficult to set a threshold on WENO weights to detect the RRI in our numerical experiments. The WENO weights can attenuate gradually as shocks are being inevitably smeared out. Therefore, we applied the traditional error estimation of RE, suggested in the original AMR algorithm [4,5], to detect RRI in all of our numerical examples in Section 4. Specifically, at each regridding time, we create a transient grid, denoted as 'G_{RE}^l', with twice the spatial and temporal mesh sizes of the current grid G_k^l. Two sets of solutions are evolved on these two grids, with the RE error computed as:

$$\varepsilon = \mathbf{u}_k^l\left(\mathbf{x}_{RE}^l, t^{n+1}\right) - \mathbf{u}_{RE}^l\left(\mathbf{x}_{RE}^l, t^{n+1}\right),$$

where $\mathbf{x}_{RE}^l$ is the mesh for G_{RE}^l. A cell will be tagged for refinement if $\varepsilon > \kappa$, where κ is a preset parameter. After RRI are detected, buffer zones are created to ensure shocks do not propagate out of the RRI before the next regridding time. Fig. 4 depicts the process of detecting RRI and creating buffer zones, followed by creation of fine meshes and data prolongation for initial and boundary conditions, which are described in the next two subsections.

3.5. Creation of fine grids and data prolongation

In the following, we describe the life cycle of a grid G_k^l, generated from its parent grid G_{kp}^{l-1}. The detected RRI on G_{kp}^{l-1} are organized into rectangular clusters using a smart bisection method [6,21]. These clusters are then refined in space. The time step is also refined with the same ratio. A new grid G_k^l, as a child of G_{kp}^{l-1}, is then created for one of the rectangular patches. Existing data at overlapping fine grid regions are copied into the new patch. Most likely, there are regions in G_k^l that are not covered by previously existing fine grids. The data in these regions must be obtained from spatial interpolation of the coarse grid solutions $\mathbf{u}_{kp}^{l-1}$. We define $\mathcal{SI}(\mathbf{u}, \mathbf{x})$ as an operation which spatially interpolates data $\mathbf{u}$ to locations $\mathbf{x}$. The data prolongation from G_{kp}^{l-1} to G_k^l can be expressed as,

$$\mathbf{u}_k^l(\mathbf{x}^o) = \mathcal{SI}(\mathbf{u}_{kp}^{l-1}, \mathbf{x}^o), \tag{6}$$

where $\mathbf{x}^o$ denotes 'open windows', i.e. locations on the newly generated grids that are not covered by previously existing grids. The data prolongation operator $\mathcal{SI}$ is designed to be of high order in order to maintain the accuracy of the AMR-WENO method.

Two spatial interpolation procedures have been considered, Lagrangian and point-to-point WENO interpolation. The point-to-point WENO interpolation is a nonlinear version of the Lagrangian interpolation. The idea of designing WENO

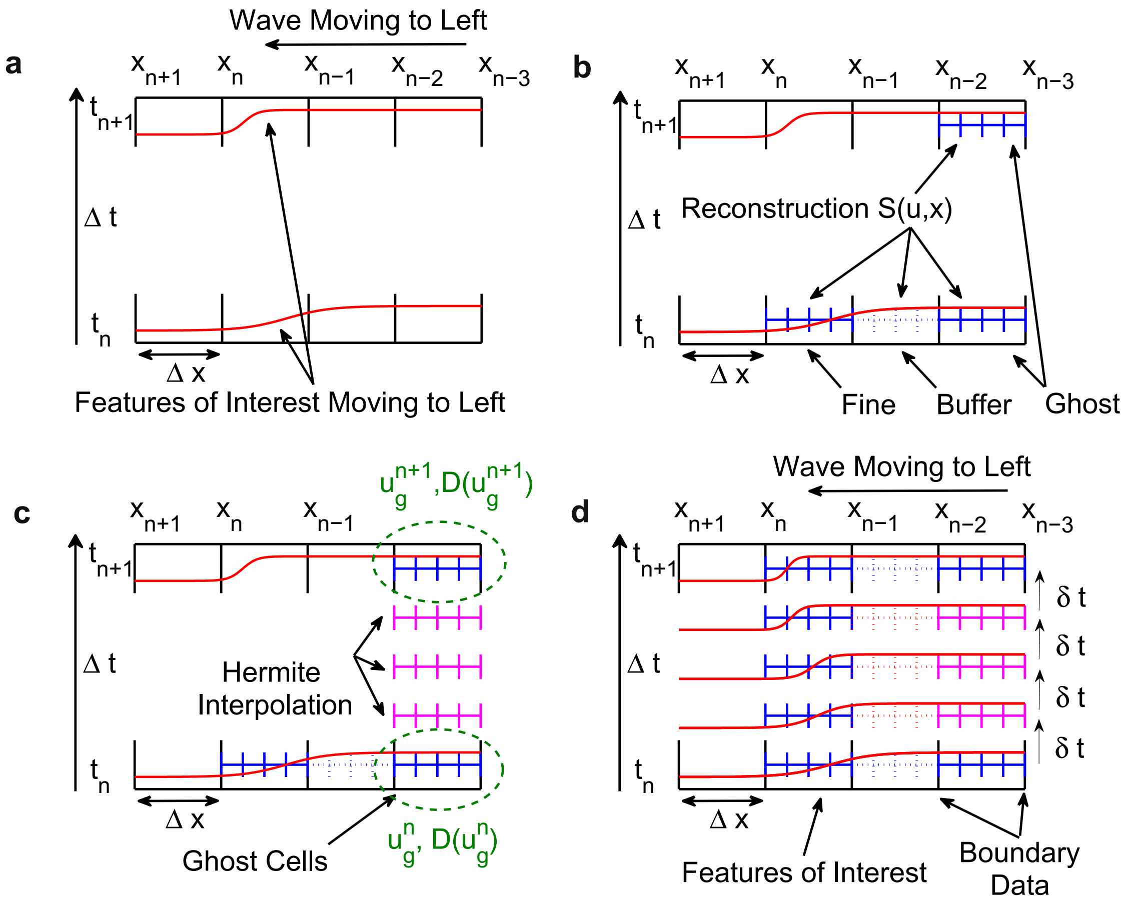

Fig. 4. A graphical depiction of the creation of the fine mesh, buffer and ghost points in the AMR-WENO algorithm. For simplicity, only buffer and ghost points behind the refinement region are depicted. Figure (a) shows detection of a feature of interest on the coarse grid at time t_n using either WENO weights or RE error estimate as an indicator function. Figure (b) shows the creation of the fine grid, buffer cells and ghost points at time t_n and the creation of ghost points at time t_{n+1}. Using WENO reconstruction, the solutions at fine grid, buffer cells and ghost points are prolongated from coarse grid solutions at time t_n. Similarly, the solutions at ghost points at time t_{n+1} are prolongated from coarse grid solutions at t^{n+1}. Figure (c) shows the construction of boundary data at the ghost points using Hermite interpolation with the data $(\mathbf{u}_g(t_n),(D_t\, u)_g(t_n)))$ and $(\mathbf{u}_g(t_{n+1}),(D_t u)_g(t_{n+1})))$. Hermite interpolation is done to facilitate high order time stepping. Figure (d) shows time stepping of the solution on the fine grid and buffer cell regions using the prolongated initial and boundary data.

interpolation is to view the high order Lagrangian interpolant on a large reconstruction stencil as a convex combination of several lower order interpolants from smaller substencils with linear weights. The WENO mechanism perturbs the linear weights to some nonlinear WENO weights, computed based on the solution structures. These WENO weights are designed to be very close to linear weights around smooth and well-resolved solution structures to maintain high order accuracy, and are designed to avoid oscillations around discontinuities by distributing very little weight to non-smooth substencils. Appendix A gives the formulas for 5th order WENO interpolation for obtaining fine grid solutions from coarse grid solutions with an refinement ratio of 5. These formulas are used in all of our test examples in the next section. The detailed procedure of point-to-point WENO interpolation is given in [28]. It was reported in [28] that simple Lagrangian interpolations are as effective as WENO interpolations. However, we choose to use WENO interpolations in all of our numerical experiments to avoid any potential oscillations. The 1D WENO interpolation procedures in Appendix A can be easily extended to multi-dimensional problems using a dimension-by-dimension approach. Fig. 4(b) shows the creation of the fine grid, buffer cells and ghost cells at time level t_n and the creation of ghost cells at time level t_{n+1}. Using WENO interpolation, the solution values on the coarse mesh are used to construct the solution vector on the fine mesh, buffer cells and ghost cells at time t_n and on the ghost cells at time t_{n+1}. For simplicity, only buffer and ghost cells behind the refinement region are depicted. In reality, buffer cells and ghost cells surround the entire RRI.

3.6. Building boundary conditions for the solution evolution on fine grids

Another key step in the AMR-WENO algorithm is to provide boundary conditions, not only at the initial stage, but also at intermediate fine grid time steps and their RK substages, for the solution evolution on fine grids. Fig. 4(c) shows the construction of boundary data in the ghost cells. The algorithm for prescribing ghost point values, which serve as the numerical boundary conditions for a fine grid, follows the two steps specified below. Because buffer zones are added to the RRI in our algorithm, the boundaries are guaranteed to be in the smooth regions of the solution. Again, our description is based on a fine grid G_k^l, generated from its parent grid G_{kp}^{l-1}. We note that before the fine grid evolution on G_k^l, the numerical solution on its parent grid G_{kp}^{l-1} has completed its evolution for a time step $\Delta t \doteq \Delta t_{kp}^{l-1}$, say from t^n to t^{n+1}. We use subscript g to denote the ghost cells, e.g., $\mathbf{u}_g$ are the fine grid solutions at ghost cell locations $\mathbf{x}_g$. We use δt to denote the fine time step for the grid G_k^l.

1. *Spatial interpolation.* The ghost point values for G_k^l at t^n and t^{n+1} are reconstructed by WENO interpolation as mentioned in the previous subsection,

$$\mathbf{u}_g(t^n) = \mathcal{SI}(\mathbf{u}_{kp}^{l-1}(t^n), \mathbf{x}_g), \quad \mathbf{u}_g(t^{n+1}) = \mathcal{SI}\left(\mathbf{u}_{kp}^{l-1}(t^{n+1}), \mathbf{x}_g\right).$$

Furthermore, the temporal derivatives of the parent grid solutions on G_{kp}^{l-1} are known from their spatial derivatives at both t^n and t^{n+1},

$$(D_t u)_{kp}^{l-1} \doteq \frac{du}{dt}|_{\mathbf{x}_{kp}^{l-1}} = -f(u)_x|_{\mathbf{x}_{kp}^{l-1}}.$$

The corresponding values at the ghost points of G_k^l can be constructed by the same WENO interpolation procedures,

$$(D_t u)_g(t^n) = \mathcal{SI}((D_t u)_{kp}^{l-1}(t^n), \mathbf{x}_g), \quad (D_t u)_g(t^{n+1}) = \mathcal{SI}((D_t u)_{kp}^{l-1}(t^{n+1}), \mathbf{x}_g),$$

where $(D_t u)_g$ are temporal derivatives of u at ghost points. These temporal derivatives are preparations for the Hermite interpolation in the next step. Fig. 4(b) shows the reconstruction of ghost point values at time levels t_n and t_{n+1} using operator $\mathcal{SI}$.

2. *Temporal interpolation.* We perform a Hermite interpolation using the function values $\mathbf{u}_g$ and their temporal derivatives $(D_t u)_g$ to obtain ghost point values at fine grid time intervals. Specifically, $\mathbf{u}_g(t^n)$, $(D_t u)_g(t^n)$, $\mathbf{u}_g(t^{n+1})$, $(D_t u)_g(t^{n+1})$ are used to construct a third order polynomial $H(t)$ such that

$$H(t^n) = u_g(t^n), \quad H(t^{n+1}) = u_g(t^{n+1}), \quad H'(t^n) = (D_t u)_g(t^n), \quad H'(t^{n+1}) = (D_t u)_g(t^{n+1}), \tag{7}$$

with the Hermite interpolation. In addition to reconstructing the ghost point values at fine time steps, we need to reconstruct ghost point values at intermedia RK stages within each fine time step. We follow the idea in [7] to ensure the third order convergence in time. If we take a fine time step $[t^n, t^n + \delta t]$ for example, the first and second stages in the third order TVD RK method can be reconstructed as

$$u_g(t_n) = H(t^n), \quad u_g^{(1)} = H(t^n) + \delta t H'(t^n), \quad u_g^{(2)} = H(t^n) + \frac{\delta t}{2} H'(t^n) + \frac{\delta t^2}{4} H''(t^n), \tag{8}$$

where $^{(1)}$ and $^{(2)}$ denote the first and second RK stages respectively. More implementation details are included in Appendix B. When another RK method is used as the time integrator, a similar strategy proposed in [7] can be applied. In our numerical experiments, we also implemented an AMR-WENO scheme that is third order accurate in space and second order in time, as a comparison to the AMR-WENO method that is fifth order in space and third order in time. In that case, the Hermite interpolation mentioned above is replaced by a second order linear interpolation in time.

With the set up of initial and boundary conditions by spatial and temporal prolongations, the solution on the fine grid G_k^l can be evolved to time t^{n+1} by several fine time steps δt, see Fig. 4(c) and (d).

3.7. Data restriction from fine to coarse grid

Data restriction is the procedure which uses the fine grid solution to update the coarse grid solution. Two restriction methods are considered in our study. The first one is called the direct point value replacement (PR). From the finite difference perspective, it is very natural to directly replace the coarse grid solution with the fine grid solution at corresponding locations. However, the coarse grid solution after such direct PR is no longer mass conservative. There have been several research efforts in the past to demonstrate or to address the mass conservation problem in the AMR framework. In [6], a flux correction step was introduced to preserve the mass conservation for the traditional finite volume AMR method. When a space–time conservative formulation [8,9] was considered, no flux correction was needed. In [28], several numerical examples experienced the wrong speed of shock propagation for the multi-domain WENO method, which we believe is largely due to the fact that the moving shock crosses the stationary multi-domain boundary. In the AMR setting, however, the RRI boundaries are dynamically moving, and are always located in the smooth solution regions. Therefore we expect the influence of such mass-conservation errors to be insignificant. This hypothesis is tested by our numerical experiments in Section 4.2. In the following, we analyze the conservation error by PR. We also consider a new procedure, termed a 'conservative update' (CU). The CU is designed to maintain the mass conservation and the high order accuracy of the scheme. At the same time, the CU should preserve the scheme's ability to capture shocks without oscillations.

Conservation error by PR. As shown in the flow map in Fig. 3, the AMR algorithm involves (1) the interpolation from the coarse to the fine grid solution, serving as the initial and boundary conditions for the solution evolution on the fine grid, (2) solution evolution on the fine grid with small fine time steps by a numerical integrator, (3) updating the coarse grid solution using the fine grid solution. In each of the above mentioned steps, there is potential to lose the mass conservation property on the coarse grid solution.

1. There is a mass difference between the numerical solution on the coarse and the fine grids at the initial stage
$$E^n = \Delta x \sum_{ic} u^n_{c,ic} - \delta x \sum_{if} u^n_{f,if}, \tag{9}$$
assuming Δx and δx are the spatial mesh sizes for the coarse and fine grids and $u^n_{c,ic}$ and $u^n_{f,if}$ are the coarse and fine grid solutions at t^n respectively.
2. The numerical fluxes into/out of the RRI boundaries are different during the solution evolution for the coarse grid and the fine grid. Typically, a flux correction procedure is employed [6].
3. There is a mass difference between the numerical solutions on the coarse and the fine grids, if PR is used
$$E^{n+1} = \Delta x \sum_{ic} u^{n+1}_{c,ic} - \delta x \sum_{if} u^{n+1}_{f,if}.$$

If we intend to preserve the mass conservation in each of the steps above, the accuracy of the scheme will degenerate to second order for smooth problems. Specifically, the mass difference in E^n and E^{n+1} above in the finite difference setting is $\mathcal{O}(\Delta x^2)$. On the other hand, it is known that both the fine and the coarse grid solutions at coarse grid $\mathbf{x}^{l-1}_{kp}$ are high order approximations to the true solution, assuming the solution is smooth

$$u^l_k|_{x^{l-1}_{kp}} - u(x^{l-1}_{kp}) = \mathcal{O}(\delta x^5), \quad u^{l-1}_{kp} - u(x^{l-1}_{kp}) = \mathcal{O}(\Delta x^5).$$

Therefore, in regions where the solution is smooth, the mass difference between the coarse and fine meshes is small,

$$u^l_k|_{x^{l-1}_{kp}} - u^{l-1}_{kp} = \mathcal{O}(\Delta x^5). \tag{10}$$

Hence the conservation error introduced by PR is very small: $\mathcal{O}(\Delta x^4)$ if the size of RRI is of $\mathcal{O}(1)$, and $\mathcal{O}(\Delta x^5)$ if the size of RRI is of $\mathcal{O}(\Delta x)$. However, around the shock, the conservation error can be as large as $\mathcal{O}(\Delta x)$.

Conservative update (CU). To ensure the mass conservation on the coarse grid, we propose a CU method, based on the PR. The idea is to first perform PR, then distribute the mass difference analyzed in Eq. (10) among coarse grid points. Since the mass difference is of high order in smooth solution regions, the high order accuracy of the scheme can be maintained. The distribution is designed in a weighted way such that the CU would preserve the scheme's ability to capture shocks without oscillations. Below, the coarse grid solution is denoted as u^c_i, the PR update solution is denoted as u^f_i, for each coarse grid point x_i. The error $e_i = u^f_i - u^c_i$ can be grouped by its sign (positive or negative). The overall mass difference is $e_{mass} \doteq \sum e_i = \sum_{e_i \geqslant 0} e_i + \sum_{e_i<0} e_i$. Without loss of generality, we assume e_{mass} to be positive, therefore $e_{mass} \leqslant \sum_{e_i \geqslant 0} e_i$. We then distribute the e_{mass} according to the weighting w_i defined below:

$$\tilde{w}_i = \begin{cases} \frac{e_i}{\sum_{e_j \geqslant 0} e_j}, & e_i \geqslant 0 \\ 0, & \text{otherwise} \end{cases} \quad \text{and } w_i = \frac{\tilde{w}_i}{\sum_j \tilde{w}_j}. \tag{11}$$

With such a design, the solution after the CU can be proven to be always bounded by u^c_i and u^f_i to control possible oscillations caused by the distribution of e_{mass}.

In the numerical experiments presented in Section 4, we test both updating methods in the 1D cases. When there is a mass inconsistency in the initial stage, see Eq. (9), the CU might generate some unphysical solutions. The performance of the PR seems to be more robust than the CU and the computational cost is less. We therefore choose to tolerate the conservation error of PR in our 2D simulations. A more detailed study on designing a scheme that is high order accurate and robust and preserves the mass conservation constitutes our future research.

3.8. Multi-dimensional implementations

The multi-dimensional implementation of finite difference WENO-RK integrators and spatial prolongation of the coarse grid solution to the fine grid solutions in the AMR-WENO can be performed in a simple dimension-by-dimension fashion while maintaining high order. The main difficulty in multi-dimensional implementation is the organization of RRI. For example, in a 2D problem, the RRI regions are organized into rectangular patches, which might interact with each other at the patch boundaries as they evolve. To resolve this problem, we have used a 'pools' method for the numerical integration step.

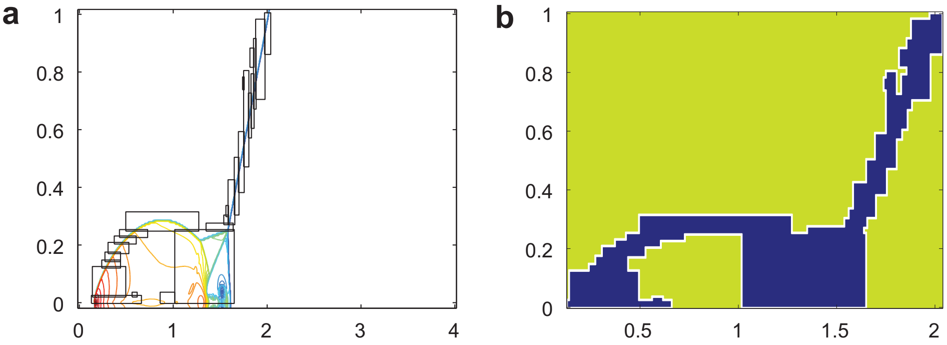

Fig. 5. The 'pools' mode for a 2D simulation. The left figure shows the feature of interest. The black boxes indicate the refinement regions. In the right figure, the light colored (yellow) region is where the solution is smooth and no refinement is needed. The dark-colored (blue) region is the refinement region which will be marched in time simultaneously. The hollow lines wrapping the dark-colored region are the ghost cells at the boundary which are provided by the data prolongation procedure in Section 3.6. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In this mode, abutting patches are evolved simultaneously in one integral grid. Take, for example, the simulation of an inclined feature across the domain as shown in Fig. 5(a), where the black boxes indicate the refinement regions. In our 'pools' mode, an integral fine grid is generated for these connected patches, resembling pools connected together. In Fig. 5(b), the yellow (light-colored) region is where the solution is smooth and no refinement is needed. The blue (dark-colored) region is the domain with fine grids. The fine grid solution in this region will be marched in time simultaneously. The implementations of the numerical integrator need to be slightly modified to compute from the starting to the ending address of each line in the 'pools'. The hollow spaces wrapping the dark-colored region are the ghost cells at the boundary. The boundary conditions are provided by the data prolongation procedure in Section 3.6. In the 'pools' mode, the need to exchange boundary conditions between neighboring rectangular patches is avoided. Note that in the 'pools' mode, only regions sharing boundaries are pooled, thus there may be more than one pools in the computational domain.

3.9. Data management and extensions

As commented in [5], the complexity of the original AMR algorithm is already intimidating to new users. The proposed WENO-AMR algorithm is even more complicated, because of the attempt to maintain high order accuracy. The algorithm can be intractable without proper data management and modularized programming. We have implemented the algorithm in an object-based fashion. For each grid, its solution vectors, spatial coordinates, mesh management data and other AMR-related information are stored together as a grid object. The implementation is modularized so that users may plug in their own integrator without knowing the details of the algorithm, only having to make minimal changes to accomodate the 'pools' mode. The algorithm can be extended to multi-level grid refinement by grid hierarchy generation [5,3], to a system of equations by characteristic decomposition [10], and to multi-dimensional problems in a dimension by dimension fashion, when the finite difference WENO scheme is used [10,28].

4. Numerical results

In this section we first examine the spatial and temporal convergence of the proposed AMR-WENO algorithm (5th order in space and 3rd order in time) by a linear advection equation with smooth initial conditions in Section 4.1. We demonstrate the main features of the AMR-WENO algorithm by a few classical 1D scalar test cases, such as the linear advection equation, Burgers' equation in Section 4.2, and by several 1D systems, such as the shallow water equations and the Euler equations in Section 4.3. The most challenging test cases in the literature were selected in order to test the effectiveness and robustness of the proposed scheme. We discuss the conservation property of the proposed scheme by comparing the numerical performance of the scheme with PR and with CU as the data restriction algorithm as discussed in Section 3.7. We also compare the proposed high order AMR-WENO scheme with a lower order version of it (3rd order in space and 2nd order in time) to demonstrate the advantage of using a very high order version of the AMR-WENO method. Finally, we present numerical results for the 2D Euler equations in Section 4.4.

In all of our numerical examples below, we use '*root solution*' to denote the AMR solution living on G^0, the root grid in the grid hierarchy. The root grid covers the entire computational domain. We use '*nA-root solution*' to denote the non-AMR, uniform solution with the same mesh resolution as G^0. The gain of solution quality by applying the AMR algorithm can be best demonstrated by comparing the nA-root solution with the root solution. We use '*full solution*' to denote the non-AMR solution living on a uniform grid with the same grid spacing as the finest grid in the AMR grid hierarchy. The quality of the full solution is what we hope to achieve using our AMR algorithm. For a better presentation of the figures, we will show the solution over the entire computational domain, as well as zoomed-in images of fine grid solutions around the shocks or complicated solution features.

4.1. Convergence study

We examine the spatial and temporal convergence of the AMR-WENO algorithm using the linear advection equations:

$$\text{1D}: \quad u_t + u_x = 0, \tag{12}$$

$$\text{2D}: \quad u_t + u_x + u_y = 0 \tag{13}$$

with periodic boundary condition. Since smooth solutions do not automatically invoke mesh refinement, a refinement signal is arbitrarily placed around the center of the computational domain. The numerical results are presented in Table 1. The spatial accuracy is tested by setting *CFL* = 0.1, so that the spatial error is the dominant error. The spatial convergence rate is observed to be very close to the desired 5th order as expected. The temporal accuracy is tested by varying *CFL*, while fixing Δx. It is harder to test the temporal order, as the available test window for Δt is quite small. On one hand, Δt is bounded from above by the CFL condition; and on the other, as Δt gets small, the spatial error may become the dominate error and contaminate the temporal convergence rate. The temporal order is observed to be slightly less than the expected 3rd order over such a small test window. For smooth test cases, the convergence rate is very close to that of a finite difference WENO scheme without the AMR (data not reported here to save space). The results indicate that AMR neither positively nor negatively impact the convergence rate for a smooth problem.

4.2. 1D scalar equation

Example 4.1 (*Burgers' equation*). We apply the AMR-WENO to Burgers' equation,

$$u_t + \left(\frac{u^2}{2}\right)_x = 0. \tag{14}$$

with two sets of initial conditions that were tested in Sebastian and Shu [28] for the multi-domain WENO method. In [28], the wrong shock propagation speed was observed due to the conservation error in the multi-domain WENO method. In the proposed AMR-WENO algorithm, if direct PR is used to update the coarse grid solution from the fine grid solution, then the scheme is not conservative either. On the other hand, the CU proposed in Section 3.7 assures the mass conservation of the AMR-WENO.

In the first test, the initial condition is

$$u(x,0) = \begin{cases} 1.02, & x < 0.03; \\ -1.0, & x \geqslant 0.03. \end{cases} \tag{15}$$

with $x \in [0,1]$ and Dirichlet boundary conditions. The exact solution contains a slowly traveling shock. In our numerical experiment, we use 40 uniform grid points as the root grid, and a refinement ratio $r = 5$, which corresponds to a full solution of 200 uniform grid points for one level of mesh refinement (or 1000 points for two levels of mesh refinement). We note that the use of 200 uniform grid points is in the same resolution as in [28] for comparison. Fig. 6(a) and (c) demonstrate the numerical solution at $t = 90$ for one and two levels of grid refinement. Fig. 6(b) and (d) are plots, zoomed in around the shock locations in Fig. 6 (a) and (c) respectively. As expected, the AMR root solution produces a much sharper shock profile than the nA-root solution. The shocks are observed to be even sharper with two levels of mesh refinement. We also implement the CU in updating the root solution from the AMR fine grid solution. The obtained numerical results are almost identical to Fig. 6, thus they are omitted to save space.

Table 1
Temporal and spatial convergence of AMR-WENO.

	Temporal Convergence, Δx fixed			Spatial Convergence, *CFL* = 0.1		
	CFL	error	order	Δx	error	order
1D	0.8000	6.29E-05		0.0625	2.46E-04	
	0.6000	2.80E-05	2.8097	0.0313	8.45E-06	4.8618
	0.5750	2.50E-05	2.6880	0.0208	1.17E-06	4.8692
	0.5500	2.22E-05	2.7162	0.0182	5.99E-07	4.9305
	0.5000	1.72E-05	2.6831	0.0156	2.91E-07	4.7778
2D	0.8485	5.85E-05		0.0667	7.20E-04	
	0.8132	5.22E-05	2.6954	0.0500	1.73E-04	4.9601
	0.7778	4.62E-05	2.7245	0.0333	2.38E-05	4.8946
	0.7425	4.09E-05	2.6322	0.0250	5.81E-06	4.8970
	0.7071	3.59E-05	2.7005			

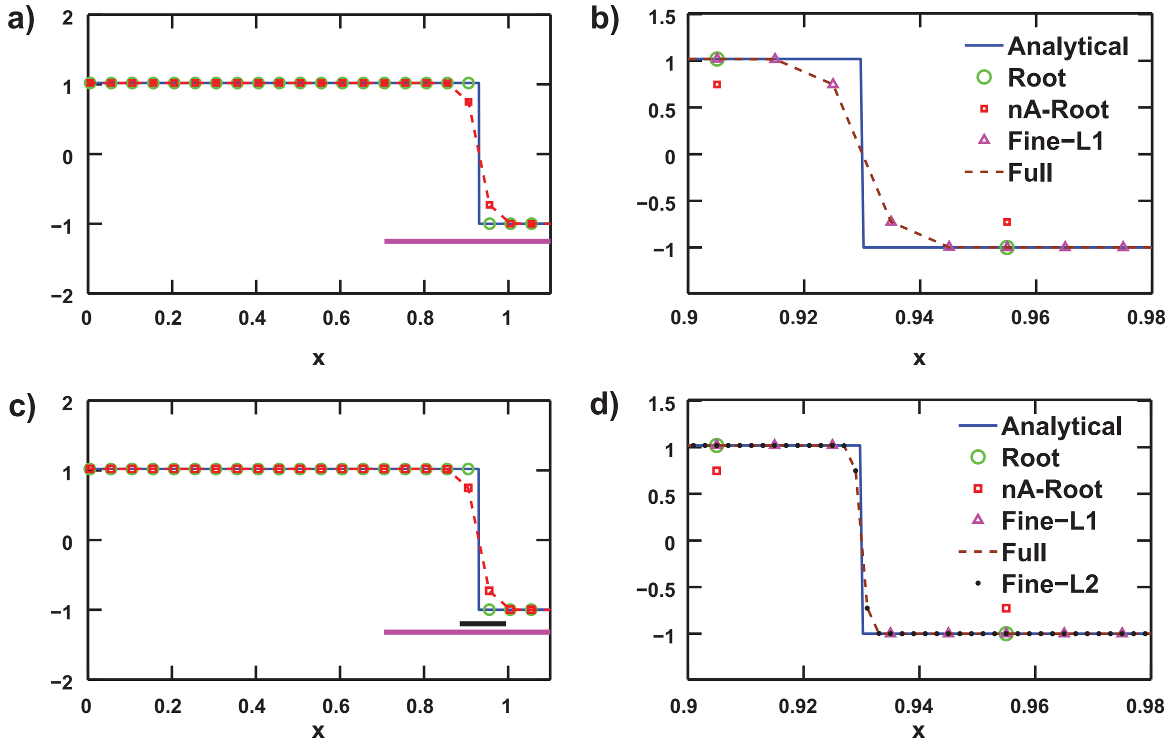

Fig. 6. Burgers' equation with the initial condition (15): the AMR solution with one (upper panels) and two (lower panels) levels of mesh refinement. The root grid contains 40 uniform grid points, with a refinement ratio of $r = 5$. The right panels are the zoom-in plots around the shock region of the left panels. The AMR regions are denoted by horizontal lines below the solution in Figures (a) and (c).

In the second test, the initial condition is:

$$u(x,0) = 17.4 + 13.3\sin(\pi x), \quad -1 \leqslant x \leqslant 1 \tag{16}$$

with $x \in [-1,1]$ and periodic boundary conditions. This is a much stronger shock than the previous one. The wrong shock propagation speed was observed in Fig. 9 of Sebastian and Shu [28]. Fig. 7 shows the AMR-WENO solution after a long time evolution ($t = 24$). We place 40 uniform grid points on the root grid, and use one level of grid refinement with the refinement ratio of $r = 5$. Again, we observe comparable performances between the PR and the CU implementations. Unlike the multi-domain WENO method, the numerical solutions of the AMR-WENO are observed to match the exact solution very well. The superior performance of the AMR-WENO method in capturing the correct shock speed, compared to the multi-domain WENO method, can be explained by the dynamic movement of the AMR region. Specifically, in the multi-domain WENO,

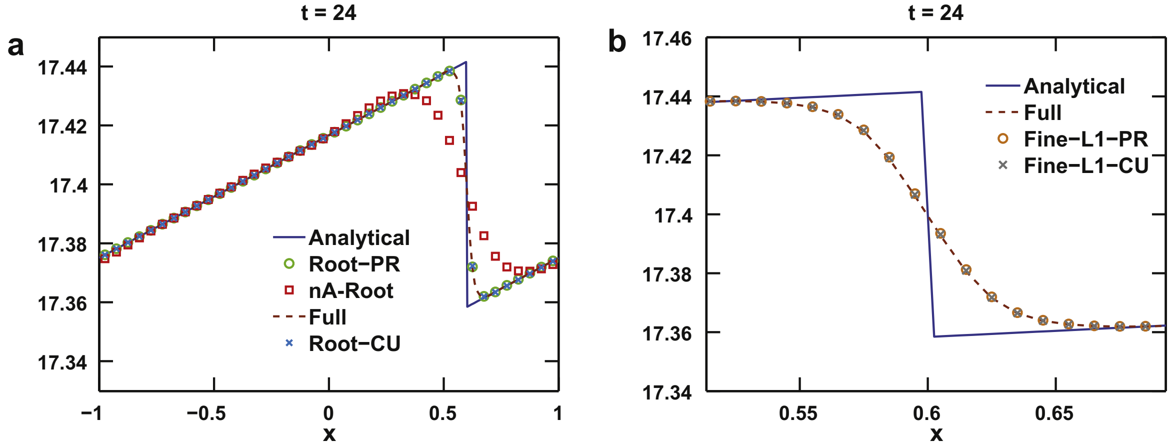

Fig. 7. Burger's equation with the initial condition (16): the AMR solution with one level of grid refinement. The root grid contains 40 grid points, with a refinement ratio of $r = 5$. The right panel is the zoomed-in plot around the shock region of the left panel.

shock passes through domain interfaces. At that moment, the loss of mass conservation becomes very significant. As a result, incorrect shock propagation speed is observed. On the other hand, the dynamic movement of the AMR regions in the AMR-WENO algorithm ensures that the shock is always located *within* the AMR region and doesn't pass through the interface of different grid sets. Since the solution is always smooth around the boundary of AMR refinement regions, the loss of mass conservation is relatively small, despite the non-conservative nature of the PR method. The shock is observed to propagate at the correct speed even after very long time propagation.

Example 4.2 (*Linear advection equation*). It is well known that the solutions of the linear advection equations might contain contact discontinuities that pose challenges to shock capturing schemes. The shocks or sharp corners in the solution profile can be easily smeared out, especially after long time propagation. In this test example, we use the initial condition tested in [37],

$$u(x,0)=\begin{cases} -x\sin\left(\frac{3\pi}{2}x^2\right), & -1<x\leqslant -\frac{1}{3},\\ |\sin(2\pi x)|, & -\frac{1}{3}<x\leqslant \frac{1}{3},\\ 2x-1-\frac{1}{6}\sin(3\pi x), & \frac{1}{3}<x\leqslant 1. \end{cases} \tag{17}$$

This initial condition is considered to be challenging because it contains many shocks and sharp corners. In our numerical experiment, we use 150 uniform grid points as the root grid, to ensure the jumps in the initial condition (at $x = 0$ or $\pm 1/3$) are placed exactly at cell interfaces. Fig. 8(a) and (c) show the results at $t = 11$ (5.5 periods) with one level of refinement using, respectively, the PR and CU for the data restriction. The small panels on the right are the corresponding zoomed-in plots around interesting features of the solution. As expected, AMR-WENO is observed to capture the contact discontinuities better than the nA-root solution. The fine grid solution of AMR has similar performance as the full solution for regions with interesting structures. Similar to Example 4.1, the AMR-WENO with PR captures the shock location correctly, despite the 'loss of mass conservation'.

An interesting observation is made when we use 151 points in the root grid for the same problem. The initial jumps are no longer placed exactly at the cell interfaces (at $x = 0$ or $\pm 1/3$). The result from AMR-WENO using the CU at $t = 11$ is shown in Fig. 9(a) and its zoomed-in plot in Fig. 9(b). The PR results are essentially the same as Fig. 8 and are thus omitted. Significant deviations between the root solution (green circles) and the AMR fine grid solution (magenta triangles) near the jumps are observed. For example, around $x = -0.65$, the root solution appears to be unphysically high when compared to the full solution. This error stays throughout the simulation, which is expected for the linear advection equation. The problem shown

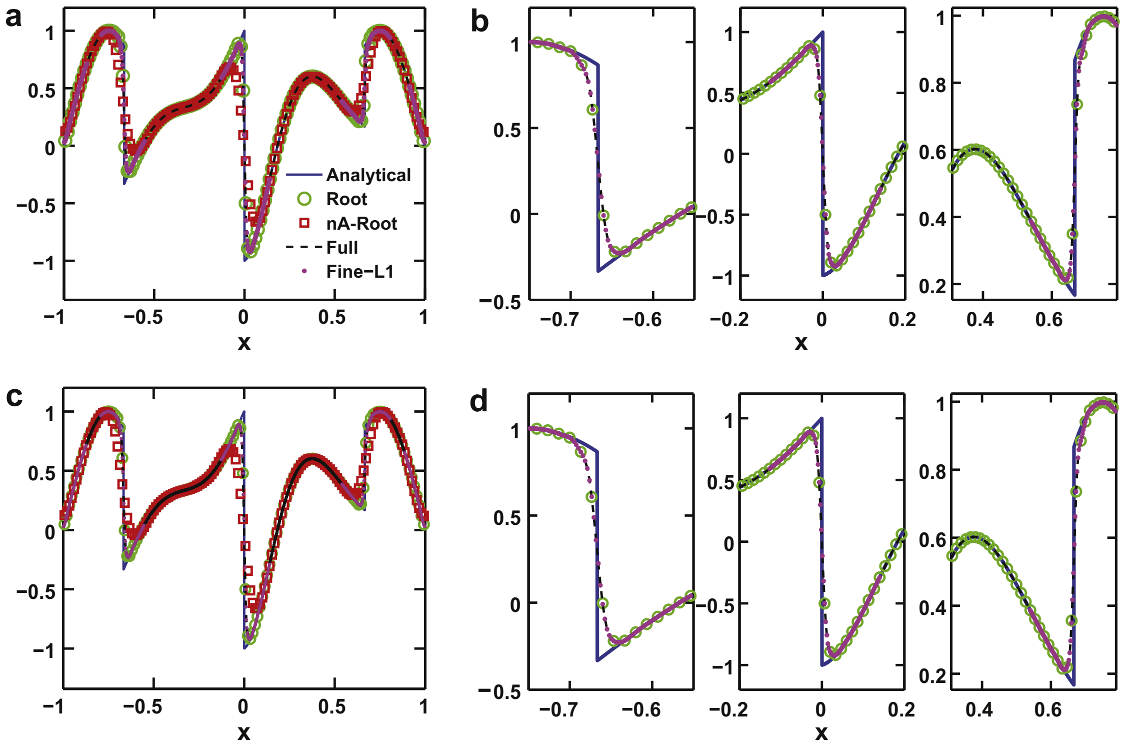

Fig. 8. Linear advection equation: the AMR solution with 150 grid points as the root grid and one level of grid refinement. The right panels are zoomed-in plots that focus on shocks/sharp corners of the left panels. The upper/lower panels are from the AMR-WENO with PR/CU data restriction, respectively.

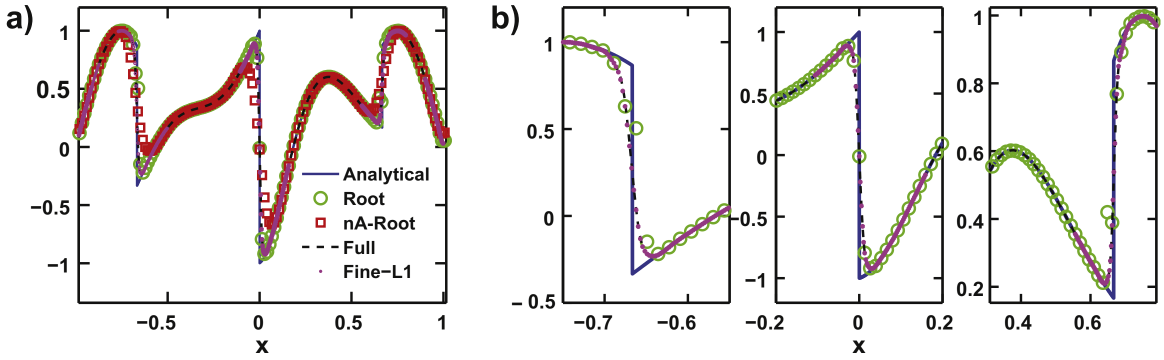

Fig. 9. Linear advection equation: the same example as Fig. 8, but 151 uniform grid points are used as the root grid and using the CU. Note that in the zoom-in figures, the coarse grid solution (green circles) deviates from the fine grid solution (magenta dots), illustrating the problem with the CU method due to initial mass inconsistency. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in this pathological case is not due to the AMR algorithm. Rather, the problem comes from the mass inconsistency between the root grid solution and the fine grid solution when they are both initialized using point values of the initial conditions, e.g. see Eq. (9). In a smooth solution, such initial mass difference is small. However when jumps are present, there can be large difference between the two initially, on the order of $\mathcal{O}(\Delta x)$. Most of the mass difference might be distributed to one transition point by the CU formula (11), yielding unphysical solution.

Fig. 10 shows the solution obtained using a lower order AMR-WENO algorithm (3rd order in space and 2nd order in time) with the same resolution as Fig. 8. Third order WENO interpolation is applied to prolongate the fine grid solution from the coarse grid solution. Second order linear interpolation in time is applied for boundary conditions. The solution from the lower order AMR-WENO is observed to be more diffusive than that obtained from the proposed high order AMR-WENO. The jumps are more smeared with the corners showing 'flattened' tops and troughs. In addition, the 3rd order AMR creates larger AMR refinement regions than the 5th order one. This is because the local truncation error estimated by RE from the 3rd order WENO scheme is greater than that of the 5th order one. Taking this into account, part of the computational savings obtained using a cheaper and lower order scheme may in turn be spent by the demand for larger refinement regions. Despite the higher computational cost per grid point for a high order scheme, the results clearly indicate that the 5th order AMR-WENO has much better resolution than the 3rd order version for this test example. We remark that the lower order AMR-WENO scheme used in this paper should have comparable performance to that of the traditional second order finite volume AMR method [4,5] in most of our test examples.

4.3. 1-D systems

Example 4.3 (*Shallow water equations*). The shallow water equations, also called the Saint Venant equations, are:

$$\begin{pmatrix} h \\ hu \end{pmatrix}_t + \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}_x = 0. \tag{18}$$

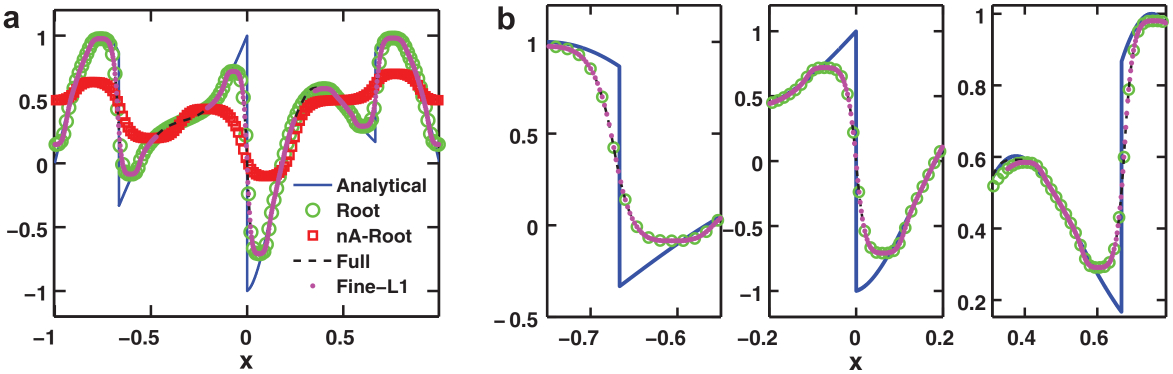

Fig. 10. Linear advection equation: the same example as Fig. 8, but computed with lower order AMR-WENO (3rd in space and 2nd order in time). The solution is observed to be more diffusive than the higher order version of AMR-WENO.

Source terms are not considered in this study. The shallow water equations, together with a jump in the initial water depth:

$$h = \begin{cases} h_0 & x < x_0, \\ h_1 & x \geqslant x_0 \end{cases}$$

becomes the dam-break problem. In our numerical study, we test the case of $h_0 = 10$ and $h_1 = 5$. The solution of this problem contains a shock propagating downstream and a rarefaction wave traveling upstream. Comprehensive discussions of this problem have been given in [36,20]. Our base scheme is the fifth order finite difference WENO scheme using a characteristic decomposition based on Roe averages and global Lax–Friedrichs flux splitting [20,10]. Fig. 11(a) and (b) show the results from AMR-WENO with 1 level of refinement using the CU and PR, respectively. The AMR solution matches very well with the full solution. However, if we initialize the root grid with 51 points with the above initial condition, unphysical solutions in the root grid are again produced by the scheme with CU, see Fig. 11(c) and (d). This is again due to the 'mass inconsistency' issue discussed in the previous example. The mass initialized on different levels of grids are inconsistent with each other. The difference is forced onto the transition points during the conservative update. Whereas this issue does not exist with the method with PR. For the proposed AMR-WENO with CU, users must be aware of the 'mass inconsistency' issue that could potentially destroy the solution. Fig. 12 shows the results of the AMR-WENO with 2 levels of refinement and with the PR. It is observed that the second level fine grids are placed both at the shock front and at the rarefaction wave. In the zoomed-in plots (Fig. 12(b) and (d)), we see that the 2nd level fine grid solution matches well with the full solution, which has 1250 (=50 · 5 · 5) points.

From the above 1D test examples, we make the following observations, when we compare the performance of AMR-WENO with PR and with CU in the data restriction step.

(1) Both the PR and the CU maintains the high order accuracy of the base integrator for smooth problems.
(2) For problems with discontinuities that we have tested, the method with PR produces equally good, or sometimes superior, solutions to the method with CU. The conservation error of the method with PR does not have any noticeable negative impact in capturing the correct shock propagating speed, thanks to the dynamic movement of the AMR refinement region.
(3) The CU formula may occasionally yield unphysical solutions on the root grid. It is important to have consistent mass on different grid levels in the initialization, especially when jumps are present. Otherwise the initial mass discrepancy may lead to unphysical solutions from the AMR-WENO with CU.

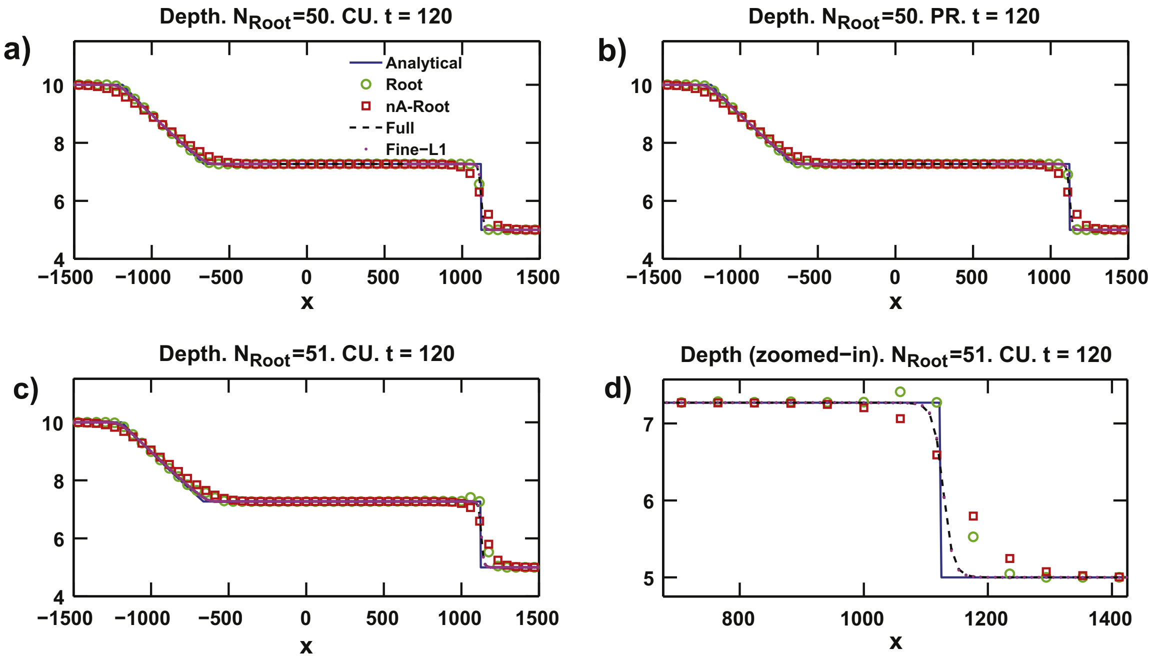

Fig. 11. Shallow water equations: the AMR solution at $t = 120$ with 1 level of refinement. (a) 50 points in the root grid, PR; (b) 50 points in the root grid, CU; (c) 51 points in the root grid, CU, overview; (d) 51 points in the root grid, CU, zoomed-in. Note with the CU, the numerical solution deviates slightly from the exact solution when the number of points on the root grid is 51, similar to what we observed in Fig. 9. This is due to the 'mass inconsistency' between the coarse and fine solutions in the initial set up. The solution for the AMR-WENO with the PR does not have such a problem, and thus is omitted.

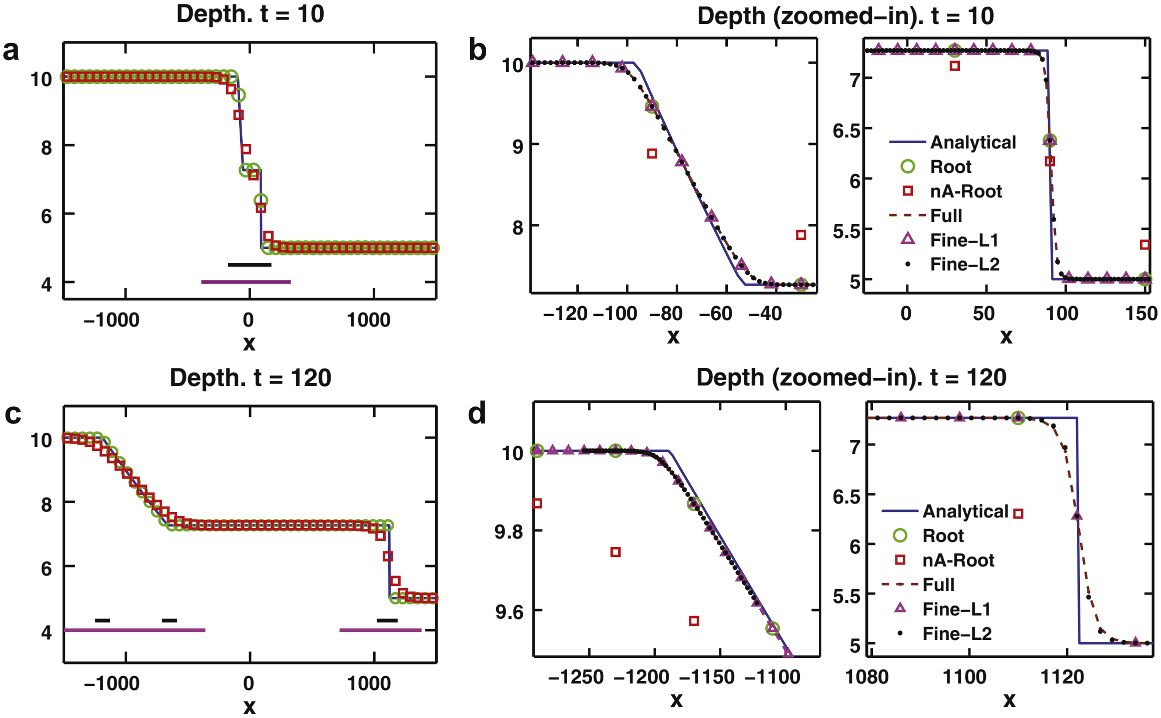

Fig. 12. Shallow water equations: the AMR solution at $t = 10$ (upper row) and $t = 120$ (lower row) with 50 grid points in the root grid, two levels of refinement and PR. The panels on the second and third columns are the zoomed-in plots of the solution around rarefaction waves and shocks respectively. The 2nd level fine grid solution is observed to match well with the full solution. The AMR regions are denoted by horizontal lines below the solution in figures (a) and (c).

Based on these observations, we conclude that the AMR-WENO with PR in data restriction step, although not mass conservative, is a better approach than the method with CU. For the rest of numerical tests in this paper, we will focus on the AMR-WENO with PR.

Example 4.4 (*Euler equations*). Consider the shock/turbulence interaction problem described by the system of Euler equations,

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E+p)u \end{pmatrix}_x = 0, \tag{19}$$

where $p = (\gamma - 1)(E - \frac{1}{2}\rho u^2)$ and $\gamma = 1.4$. The initial condition is

$$\begin{aligned} &\rho = 3.857143; \quad u = 2.629369; \quad p = 10.3333 \quad \text{when} \quad x < -4 \\ &\rho = 1 + 0.2\sin 5x; \quad u = 0; \quad p = 1 \quad \text{when} \quad x \geqslant -4 \end{aligned} \tag{20}$$

The fifth order finite difference WENO scheme using characteristic decomposition based on Roe averages and global Lax–Friedrichs flux splitting is used as the base numerical integrator. This problem showcases the advantages of the AMR-WENO algorithm because normally a very fine grid is needed for the resulting high frequency perturbations. Readers are refered to Fig. 14(c) in [32] for the solution to this problem produced by a second order MUSCL type TVD scheme.

In our numerical test, we start with a root grid with 100 points and 2 levels of mesh refinement. Fig. 13 shows the results at $t = 0.4$ (upper row) and 1.8 (lower row), with the small panels on the right showing the close-ups of different refinement regions. The AMR refinement regions are denoted by horizontal lines below the solution in the left panels of the figure. It is observed that the first level of mesh refinement covers almost the whole domain. This is because the root grid is too coarse to support the initial sine density profile. The second level fine grids, on the other hand, are only generated at sharp corners and high frequency regions. The AMR solution matches the full solution almost exactly whereas the nA-root solution completely fails to describe the fine scale perturbations. The multi-resolution nature of the AMR-WENO algorithm offers significant advantages for this type of problem, since it does not need to know *a priori* where the refinement regions are.

Fig. 14 shows the numerical results for the same problem computed by a lower order AMR-WENO method (3rd in space and 2nd in time). For comparison, the full solution from the 5th order WENO integrator is shown as a solid blue line in the figure. The 3rd order scheme is observed to perform reasonably well for the overall solution, showing great improvement

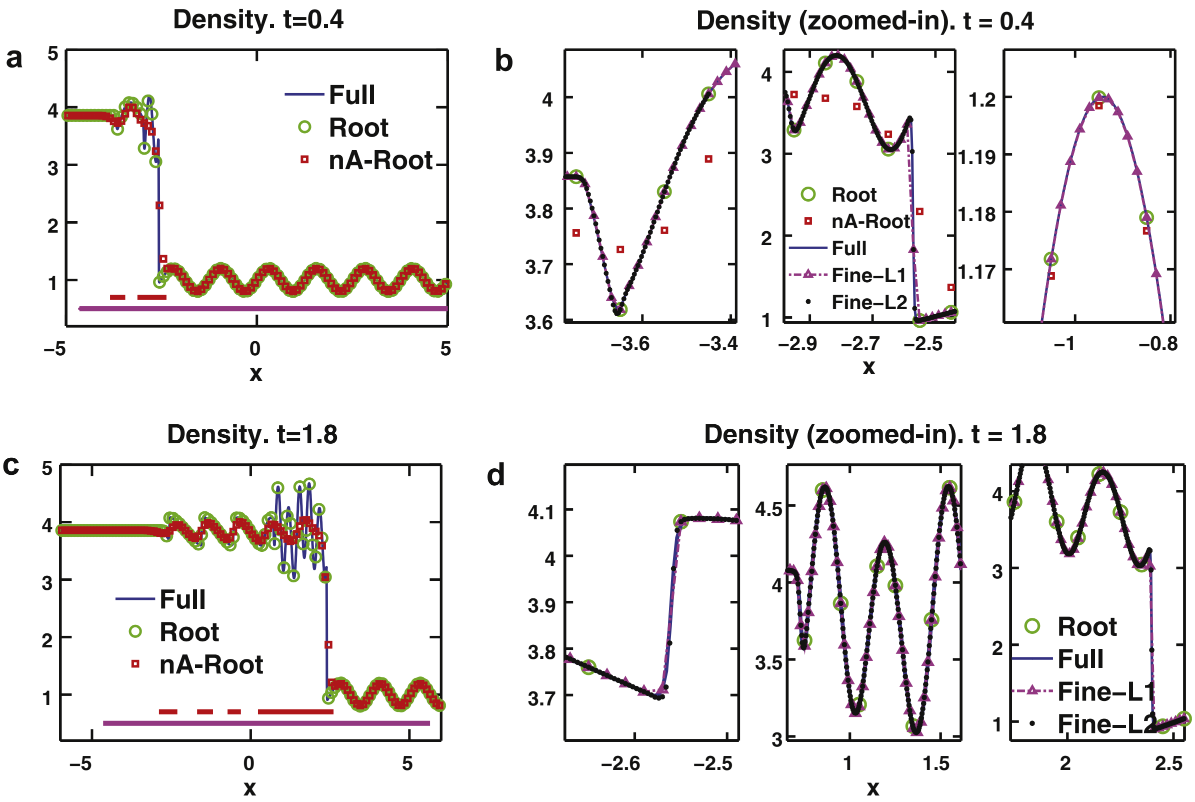

Fig. 13. Euler equation: the AMR solution at $t = 0.4$ (upper row) and $t = 1.8$ (lower row) with 100 grid points in the root grid and two levels of grid refinement with a refinement ratio of $r = 5$. The right panels are the zoom-in plots of the solution in regions with rich structures. The AMR refinement regions are denoted by thick solid horizontal lines below the solution in (a) and (c).

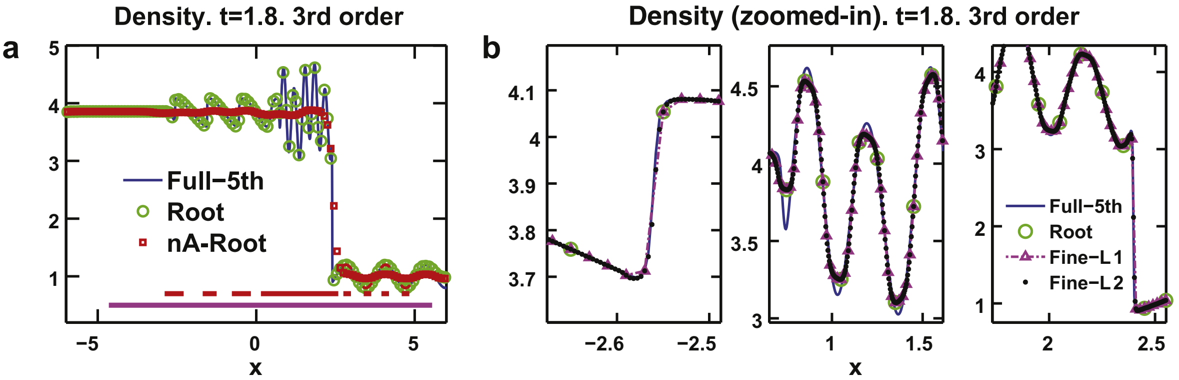

Fig. 14. Euler equation: the same example as Fig. 13, but computed with the lower order AMR-WENO (3rd in space and 2nd in time). The solution is observed to be more diffuse than that in Fig. 13. We also observe larger refinement regions compared to those in Fig. 13.

over the nA-root solution. However, it doesn't capture the high frequency peaks and troughs as competently as the higher order AMR-WENO, see the third plot in Fig. 14.

Example 4.5 (*Two blast wave problem*). Consider the two blast wave problem [35,28], which is described by the Euler Eq. (20) with the initial condition

$$\begin{cases} \mathbf{u}_L, & 0 < x \leqslant 0.1, \\ \mathbf{u}_M, & 0.1 < x \leqslant 0.9, \\ \mathbf{u}_R, & 0.9 < x < 1, \end{cases}$$

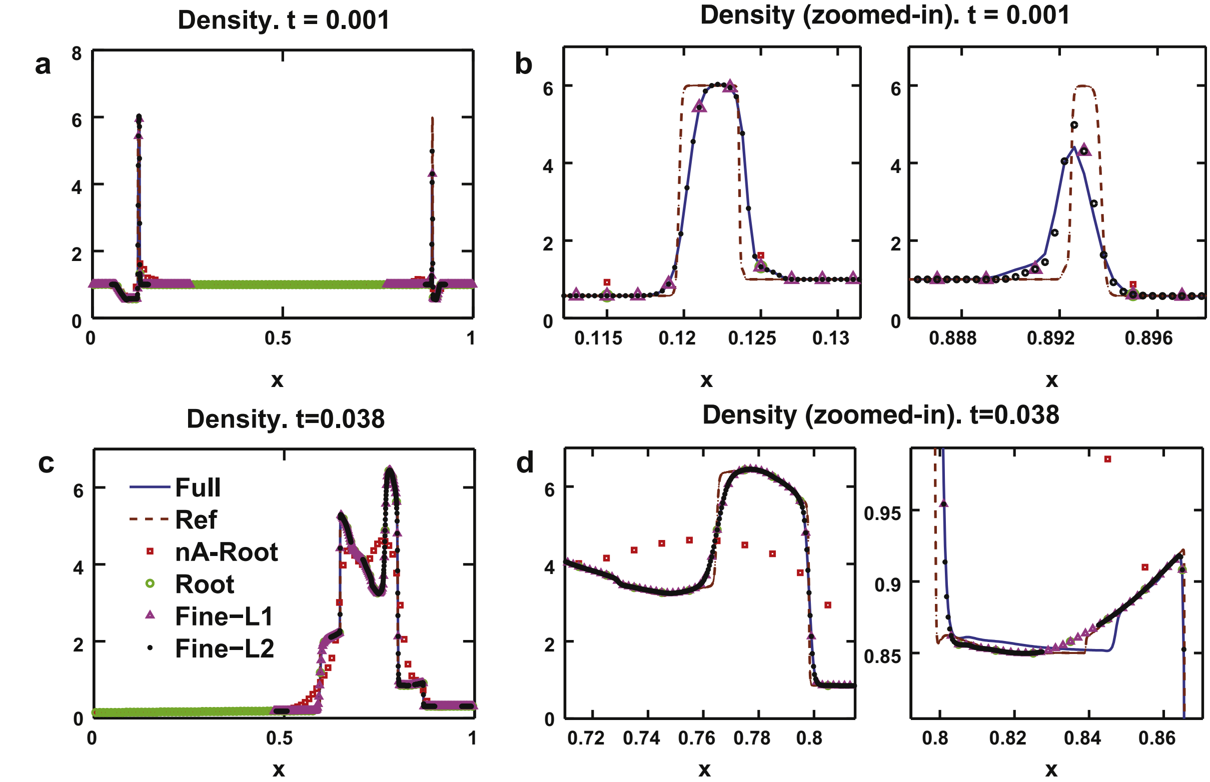

Fig. 15. The two blast wave problem: the high order AMR-WENO solution at $t = 0.001$ (upper) and $t = 0.038$ (lower). 300 uniform grids with two levels of refinement are used. The right panels are zoom-in plots of solution regions with rich structures.

where $\mathbf{u}_L = (1,0,10^3)'$, $\mathbf{u}_M = (1,0,10^{-1})'$, $\mathbf{u}_R = (1,0,10^2)'$. Reflective boundary conditions are applied at $x = 0$ and $x = 1$. Strong shocks, rarefactions and contact discontinuities interact multiple times during the evolution. Many of the interactions take place in a small volume which presents a challenge to the numerical simulation with a uniform Eulerian grid [35], but highlights the effectiveness of the AMR algorithm. We apply the AMR-WENO algorithm with the PR update to this problem. 300 uniform grid points are used in the root grid and two levels of mesh refinement are employed. The AMR, full and nA-root solutions, as well as a reference solution produced by a uniform grid of *16000* points, are shown in Fig. 15 at two different times.

In the following, we discuss the behavior of the AMR and non-AMR solutions at early ($t = 0.001$) and later ($t = 0.038$) stages of the solution evolution separately. At $t = 0.001$, see the plots in the upper row of Fig. 15, the discontinuities in the density create two blast waves, both of which are identified by the first level of the refinement. The second level of grid refinements are placed around the two shocks. It is interesting to note that at this time, the density peak on the right computed by the AMR is higher (less diffused) than that computed by the full solution, compared with the reference solution, see zoomed-in plots on the right panels of the Fig. 15. This 'superior' performance can be explained by the global Lax-Friedrichs flux splitting employed in this particular WENO solver. Specifically, the numerical flux in the finite difference scheme is $\hat{f}_{i+1/2} = \hat{f}^-_{i+1/2} + \hat{f}^+_{i+1/2}$, where $\hat{f}^\pm_{i+1/2}$ is reconstructed from $\{f^\pm_i\}_{i=1}^N$, with $f^\pm_i = f(u_i) \pm \alpha u_i$. α is selected as the greatest characteristic speed in the computational domain. The larger the computational domain is (the entire domain for the full solution vs. local AMR refinement region for the AMR fine grid solution), the larger the α will be, consequently more numerical diffusion. The two waves later collide, producing a very high density spike in a very thin volume. The two AMR regions follow the two shocks and merge into one refinement patch prior to the collision of the two waves. The numerical solutions at $t = 0.038$ are given in the lower row of Fig. 15. It is observed that the AMR solution matches well with the full solution. Again, the AMR fine grid solution even matches better with the reference solution, due to less numerical diffusion introduced by Lax-Friedrichs flux splitting.

Fig. 16 compares the solution at $t = 0.038$ using the fifth order vs. third order AMR-WENO algorithm. The performance of the third order AMR-WENO seems to be decent. However, around the region where the solution has small scale structures, the fifth order AMR-WENO method outperforms the third order one. This example demonstrates the ability of the proposed high order AMR-WENO algorithm in resolving complex interactions.

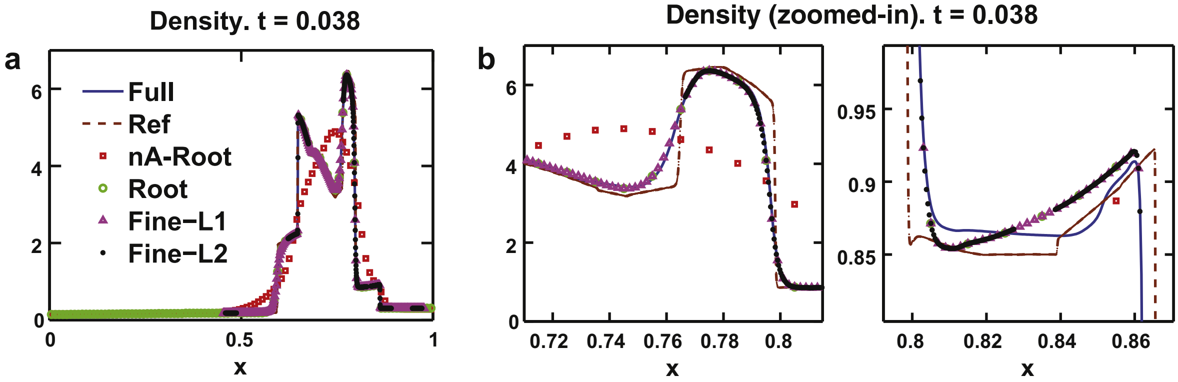

Fig. 16. The two blast wave problem: the same problem set up as in Fig. 15), but computed with the lower order AMR-WENO (3rd in space and 2nd in time). The legend is the same as in Fig. 15

4.4. 2D Euler equation

Example 4.6 (*Double Mach reflection problem*). Consider the 2D Euler equation:

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}_y = 0, \quad (x,y) \in [0,4] \times [0,1]. \tag{21}$$

This test problem simulates reflection of planar shocks in air from wedges. Inflow from the upper left boundary drives a Mach 10 shock at a 60 degree angle of impact to the reflective wall placed along the x-axis. The undisturbed fluid has $\rho = 1.5$ and $p = 1.0$. The top boundary conditions are set to describe the exact motion of this shock. Exact post shock conditions are imposed along the bottom boundary from $x = 0$ to $x = 1/6$. Starting from $x = 1/6$, a reflecting wall condition is imposed for the rest of the bottom boundary. Inflow and outflow conditions are applied for the left and right boundaries. This example has been studied extensively in the literature by the traditional second order AMR method and high order finite difference WENO method [5,10].

In our numerical test, we use the proposed very high order AMR-WENO with PR. The root grid with 180 × 720 points in the x-y-plane ($\Delta y = \Delta x = \frac{1}{180}$) and a refinement ratio of $r = 5$ are used. The base scheme is the fifth order finite difference WENO scheme using characteristic decomposition based on Roe averages and a global Lax–Friedrichs flux splitting. Fig. 17 shows the refinement regions at $t = 0.01$ (left), 0.03 (middle) and 0.12 (right). The regions around density jump and the reflection point, where the solution structures are rich, are automatically chosen to be refined. The reflected shock region grows as time evolves and the refinement regions start to split apart to cover the expansion edges of the reflection. The plots in the upper row in Fig. 18 show the density field at time $t = 0.2$ computed with a uniform fine grid of 900 × 3600 (left) and the AMR-WENO with the AMR refinement regions marked by thin black boxes (right). The plots in the lower row in Fig. 18 are the corresponding zoomed-in plots around the regions with rich solution structures. Both the full and AMR solutions resolve the fine scale Kelvin–Helmholtz rolls along the principle slip line and the reflective wall very well. We also notice maturely developed vortex structures underneath the triple point ($x = 2.4, y = 0.18$), which was not previously observed in [5]. Slight differences are observed when the AMR solution is compared with the full solution. The rolls in the AMR solution

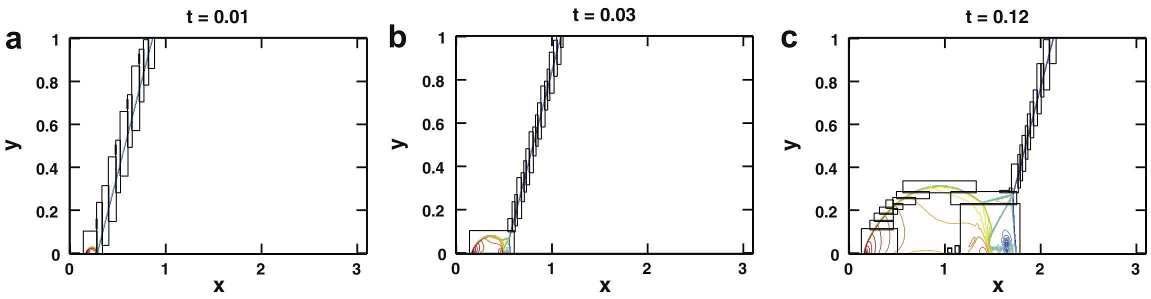

Fig. 17. The double Mach reflection problem: the AMR region, highlighted in thin black boxes, at $t = 0.01$ (left), $t = 0.03$ (middle) and $t = 0.12$ (right).

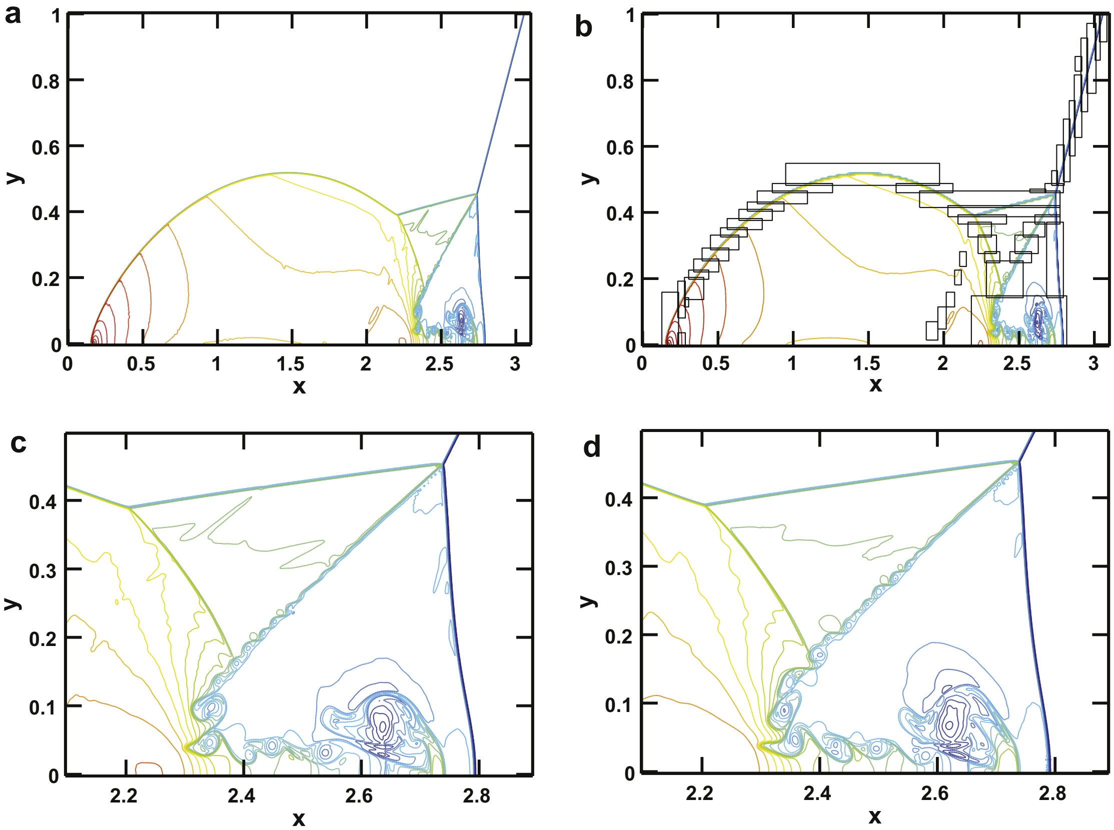

Fig. 18. The double Mach reflection problem: the full solution with 900 × 3600 grid points (left) and the AMR solution with a root grid of 180 × 720 points and one level of grid refinement (right). The density is shown with 30 equally spaced contours from $\rho = 1.5$ to $\rho = 22.9705$. The refinement regions are marked by thin black boxes. The lower panels are the zoomed-in plots of the solution in regions with rich structures.

are larger in shape and their outlines are sharper. This is likely due to the fact that the AMR-WENO has less numerical diffusion than the fine WENO solution, as discussed in Example 4.5. Because of the relative coarse spatial mesh and large Δt in the non-refined region, the computational cost of the proposed high order AMR-WENO is approximately 10% of that of the full uniform resolution.

Fig. 19 show the entire solution (left) and the corresponding zoom-in plot (right) from a lower order AMR-WENO algorithm (3rd order in space and 2nd order in time) with the same grid resolution. It is observed from Fig. 19(b) that both the vortices and the Kelvin–Helmholtz rolls are missing. There is only one visible vortex near the bottom wall. We believe

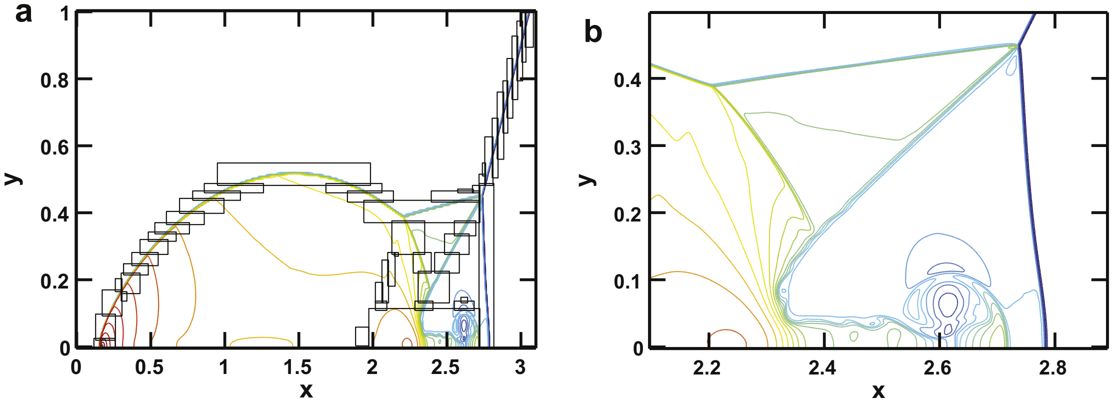

Fig. 19. The double Mach reflection problem: the same example as in Fig. 18, but computed with the lower order AMR-WENO (3rd order in space and 2nd order in time). (a) Overview of the density field and refinement regions (marked by thick black boxes) at t = 0.2, (b) zoomed-in plot of the density field in regions with rich structures. The density is shown with 30 equally spaced contours from $\rho = 1.5$ to $\rho = 22.9705$.

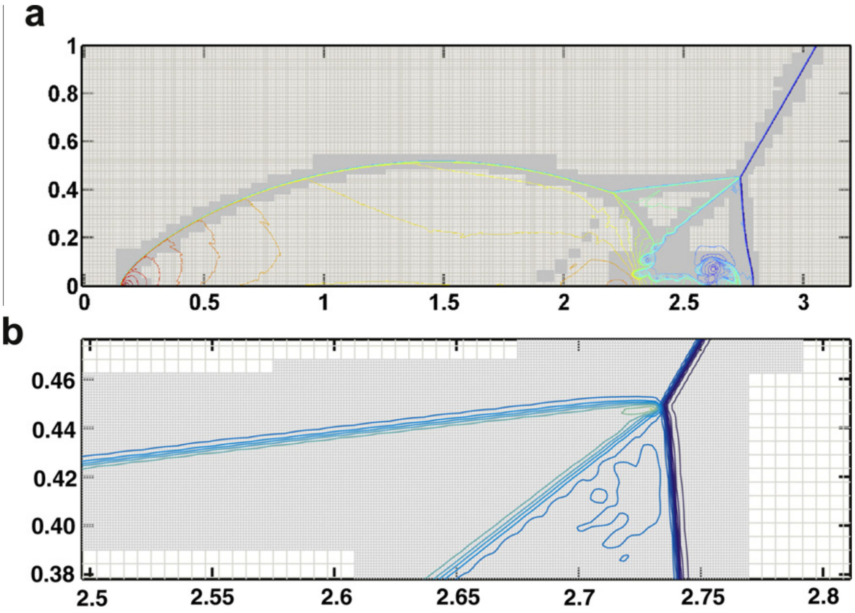

Fig. 20. (a) Grid resolution is shown beneath the solution contours. In this figure, the fine grids are too dense. Thus they seem like grayed patches; (b) Zoom-in around the triple point. The refinement ratio is 5.

that the high order accuracy of the scheme plays a major role in resolving these fine scale structures, not only due to the high order accuracy around smooth solution regions, but also due to the much less numerical diffusion introduced when there are discontinuities. Finally, to give the readers an idea of the grid resolution for the 2D simulation presented in Fig. 19, we show the actual grid in Fig. 20.

5. Conclusions

We draw the following conclusions about the proposed high order AMR-WENO algorithm based on the numerical tests we have performed.

1. The proposed AMR-WENO algorithm maintains the high order accuracy (5th order in space and 3rd order in time) of the finite difference WENO scheme in problems with smooth solutions. The AMR-WENO scheme produces solutions that match very well with the corresponding uniform grid solution in various challenging problems with shocks or complicated solution structures. In cases when global Lax–Friedrichs flux splitting is used, the AMR solution may even produce better solution than the full solution computed on a uniform grid.
2. The very high order (5th order in space and 3rd order in time) AMR-WENO algorithm performs significantly better than the lower order version (3rd order in space and 2nd order in time) not only in accuracy but also in reducing numerical diffusion and resolving fine-scale features. Compared to the lower order AMR, the higher order scheme also has a smaller refinement region due to the local truncation error from RE, thus the higher order cost per grid point is partially compensated.
3. The AMR-WENO with PR in data restriction works well for various test problems, despite the fact that it is not conservative. This is because the shocks are always contained within the refinement regions and do not cross interfaces.

Our future research includes the following: (1) Compare solution quality and computational efficiency between the lower order and the higher order AMR-WENO schemes, and with the traditional finite volume AMR method [5]. We expect the answers to the question 'which scheme is better' will differ from case to case. (2) The issue of 'mass inconsistency' among different levels of grids needs to be further addressed and, hopefully, will lead to an algorithm that is high order, stable and conservative. (3) The high order AMR-WENO algorithm will be applied to other test problems.

Acknowledgements

We thank Dr. Chi-Wang Shu for incisive suggestions and generous help. We thank Dr. Randy Leveque for constructive advice. We thank the anonymous reviewers, as well as Dr. W. Nicolas G. Hitchon whose useful comments and suggestions led to the improved version of the manuscript. The first author was supported by NOAA Center of Excellence for Great Lakes and Human Health. The second and third authors were supported by Air Force Office of Scientific Computing and the National Science Foundation.

Appendix A. WENO interpolation in space

The details of WENO interpolation have been provided in [28,22], etc. For completeness, we briefly outline the procedures. Given a stencil of $(2k-1)$ points, there are k candidate substencils:

$$S_s(i) = \{x_{i-s}, \dots, x_{i-s+k-1}\}, \quad s = 0, \dots, k-1.$$

A Lagrangian polynomial of order $k-1$ can be produced based on each candidate substencil S_s,

$$u_L^{(s)}(x) = \sum_{j=0}^{k-1} u_{i-s+j} c_{sj}(x),$$

where $c_{sj}(x)$ are constants that can be pre-determined for any specific location x:

$$c_{sj}(x) = \prod_{\substack{l=0 \\ l \neq j}}^{k-1} \frac{x - x_{i-s+l}}{x_{i-s+j} - x_{i-s+l}}.$$

The Lagrangian polynomial constructed based on the large stencil of $(2k-1)$ points can be viewed as a linear combination of the substencil polynomials

$$u_L^{(x)} = \sum_{s=0}^{k-1} d_s(x) u_L^{(s)}(x),$$

where $d_s(x)$ are linear weights. Borrowing the idea of WENO in [18], the linear weights $d_s(x)$ can be perturbed to be nonlinear weights $\omega_s(x)$

$$\omega_s(x) = \frac{\tilde{\omega}_s(x)}{\sum_{s=0}^{k-1} \tilde{\omega}_s(x)}, \quad \tilde{\omega}_s(x) = \frac{d_s(x)}{(\varepsilon + \beta_s(x))^2},$$

where $\beta_s(x)$ is a smoothness indicator of each substencil computed from

$$\beta_s = \sum_{l=1}^{k-1} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x^{2l-1} \left(\frac{d^l}{dx^l} u_L^{(s)}(x) \right)^2 dx.$$

The WENO interpolation

$$u_w(x) = \sum_{s=0}^{k-1} \omega_s(x) u_L^{(s)}(x),$$

is a nonlinear convex combination of substencil polynomials. It maintains $(2k-1)$th order accuracy when $u(x)$ is smooth in all the substencils. When the solution in a substencil is not smooth, the nonlinear weight generated by WENO would be small compared to others. In this way, oscillations are avoided and stability is ensured. In the spatial data prolongation procedure in the proposed AMR-WENO algorithm, data will be interpolated to the fixed locations $x_i + \delta \frac{\Delta x}{r}$, where r is the ratio of refinement and $\delta = -(r-1)/2, \dots, (r-1)/2$. The interpolation constants $C_{sj}(x)$ and linear weights $d_s(x)$ for $r = 5$ are tabulated in Table A1.

Appendix B. Hermite interpolation in time

For easier implementation, the Hermite interpolation polynomial $H(t)$ is rescaled as $H(\tau) = b_3\tau^3 + b_2\tau^2 + b_1\tau + b_0$, $\tau \in [0,1]$. The conditions given by (7) can be written in a matrix form:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \mathbf{Ab} = \begin{bmatrix} u_g(t^n) \\ \Delta t (D_t u)_g (t^n) \\ u_g(t^{n+1}) \\ \Delta t (D_t u)_g (t^{n+1}) \end{bmatrix} = \mathbf{Y}.$$

Then $H(\tau)$, $H'(\tau)$ and $H''(\tau)$ can be evaluated as

Table A1
WENO interpolation constants and linear weights for $r = 5$.

δ	s	d	C_{sj}		
			j = 0	j = 1	j = 2
-2/5	1	2/25	42/25	−24/25	7/25
	2	16/25	7/25	21/25	−3/25
	3	7/25	−3/25	16/25	12/25
-1/5	1	3/25	33/25	−11/25	3/25
	2	33/50	3/25	24/25	−2/25
	3	11/50	−2/25	9/25	18/25
1/5	1	11/50	18/25	9/25	−2/25
	2	33/50	−2/25	24/25	3/25
	3	3/25	3/25	−11/25	33/25
2/5	1	7/25	12/25	16/25	−3/25
	2	16/25	−3/25	21/25	7/25
	3	2/25	7/25	−24/25	42/25

$$
\begin{aligned}
H(\tau) &= \begin{bmatrix} \tau^3 & \tau^2 & \tau & 1 \end{bmatrix}\mathbf{A}^{-1}\mathbf{Y},\\
H'(\tau) &= \begin{bmatrix} 3\tau^2 & 2\tau & 1 & 0 \end{bmatrix}\mathbf{A}^{-1}\mathbf{Y},\\
H''(\tau) &= [6\tau \quad 2 \quad 0 \quad 0]\mathbf{A}^{-1}\mathbf{Y}.
\end{aligned}
$$

All the terms above except for **Y** can be pre-computed and stored, thus only one small matrix vector multiplication is involved per ghost point in Hermite interpolation.

References

[1] A. Baeza, P. Mulet, Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations, International Journal for Numerical Methods in Fluids 52 (2006) 455–471.

[2] M. Barad, P. Colella, A fourth-order accurate local refinement method for Poisson equation, Journal of Computational Physics 209 (2005) 1–18.

[3] J. Bell, M. Berger, J. Saltzman, M. Welcome, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws, SIAM Journal on Scientific Computing 15 (1994) 127.

[4] M.J. Berger, Adaptive mesh refinement for hyperbolic partial differential equations, Phd. Thesis, Standford University, 1982.

[5] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, Journal of Computational Physics 82 (1989) 64–84.

[6] M.J. Berger, R.J. Leveque, Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, SIAM Journal on Numerical Analysis 35 (1998) 2298–2316.

[7] M. Carpenter, D. Gottlieb, S. Abarbanel, W. Don, The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a careful study of the boundary error, SIAM Journal on Scientific Computing 16 (1995) 1241–1252.

[8] S. Chang, The method of space-time conservation element and solution element – a new approach for solving the Navier–Stokes and Euler equations, Journal of Computational Physics 119 (1995) 295–324.

[9] S. Chang, X. Wang, C. Chow, The space-time conservation element and solution element method: a new high-resolution and genuinely multidimensional paradigm for solving conservation laws, Journal of Computational Physics 156 (1999) 89–136.

[10] B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer, New York, 1998.

[11] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Mathematics of Computation 52 (1989) 411–435.

[12] B. Cockburn, C.-W. Shu, The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V^{*} 1:: Multidimensional Systems, Journal of Computational Physics 141 (1998) 199–224.

[13] S. Gottlieb, D. Ketcheson, C.-W. Shu, High order strong stability preserving time discretizations, Journal of Scientific Computing 38 (2009) 251–289.

[14] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, Journal of Computational Physics 183 (2002) 508–532.

[15] J. Hesthaven, S. Gottlieb, D. Gottlieb, Spectral Methods for Time-dependent Problems, Cambridge University Press, 2007.

[16] P. Houston, B. Senior, E. Suli, hp-discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity, International Journal for Numerical Methods in Fluids 40 (2002) 153–169.

[17] L. Jameson, AMR vs high order schemes, Journal of Scientific Computing 18 (2003) 1–24.

[18] G.S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1996) 202–228.

[19] S. Karni, A. Kurganov, G. Petrova, A smoothness indicator for adaptive algorithms for hyperbolic systems, Journal of Computational Physics 178 (2002) 323–341.

[20] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, New York, NY, 2002.

[21] R.J. LeVeque, e. M.J. Berger, Clawpack software 4.3. Available from: <www.clawpack.org>, 2009 (Accessed 28.04.2009).

[22] S.T. Li and J. Mac Hyman, Adaptive mesh refinement for finite difference WENO schemes, 2003 <http://math.lanl.gov/shenli/publications/amrweno.pdf>.

[23] D. Mavriplis, A. Jameson, Multigrid solution of the Euler equations on unstructured and adaptive meshes, in: Copper Mountain Conference on Multigrid Methods, 1987.

[24] T. Oliver, A high-order, adaptive, discontinuous Galerkin finite element method for the Reynolds-averaged Navier–Stokes equations, Phd Thesis, Citeseer, 2008.

[25] M. Pan, E.F. Wood, D.B. McLaughlin, D. Entekhabi, L. Luo, A multiscale ensemble filtering system for hydrologic data assimilation. Part I: implementation and synthetic experiment, Journal of Hydrometeorology 10 (2009) 794–806.

[26] T. Plewa, T. Linde, e. Gregory Weirs, V., Adaptive mesh refinement – theory and applications, in: Proceedings of the Chicago workshop on adaptive mesh refinement methods, Springer, 2003.

[27] J. Ray, C.A. Kennedy, S. Lefantzi, H.N. Najm, Using high-order methods on adaptively refined block-structured meshes: derivatives, interpolations, and filters, SIAM Journal on Scientific Computing 29 (2007) 139–181.

[28] K. Sebastian, C.-W. Shu, Multidomain WENO finite difference method with interpolation at subdomain interfaces, Journal of Scientific Computing 19 (2003) 405–438.

[29] C.-W. Shu, High-order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD, International Journal of Computational Fluid Dynamics 17 (2003) 107–118.

[30] C.-W. Shu, High order weighted essentially non-oscillatory schemes for convection dominated problems, SIAM Review 51 (2009) 82–126.

[31] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (1988) 439–471.

[32] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes.2, Journal of Computational Physics 83 (1989) 32–78.

[33] A. Tam, D. Ait-Ali-Yahia, M. Robichaud, M. Moore, V. Kozel, W. Habashi, Anisotropic mesh adaptation for 3D flows on structured and unstructured grids, Computer Methods in Applied Mechanics and Engineering 189 (2000) 1205–1230.

[34] L. Wang, D. Mavriplis, Adjoint-based hp adaptive discontinuous Galerkin methods for the 2D compressible Euler equations, Journal of Computational Physics 228 (2009) 7643–7661.

[35] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.

[36] Y.-L. Xing, C.-W. Shu, High order finite difference WENO schemes with the exact conservation property for the shallow water equations, Journal of Computational Physics 208 (2005) 206–227.

[37] Z.-F. Xu, C.-W. Shu, Anti-diffusive flux corrections for high order finite difference WENO schemes, Journal of Computational Physics 205 (2005) 458–485.