A MOVING MESH WENO METHOD FOR ONE-DIMENSIONAL CONSERVATION LAWS[*]

XIAOBO YANG[†], WEIZHANG HUANG[‡], AND JIANXIAN QIU[§]

Abstract. In this paper, we develop an efficient moving mesh weighted essentially nonoscillatory (WENO) method for one-dimensional hyperbolic conservation laws. The method is based on the quasi-Lagrange approach of the moving mesh strategy in which the mesh is considered to move continuously in time. Several issues arising from the implementation of the scheme, including mesh smoothness, mesh movement restriction, and computation of transformation relations, and their effects on the accuracy of the underlying scheme have been addressed. Particularly, it is found that a least squares smoothing can be used to effectively smooth the mesh, and the transformation relations can be computed using either high order finite differences or WENO applied to some geometric conservation laws. Moreover, mesh movement can cause WENO schemes to become unconditionally unstable. A simple strategy is used to restrict the mesh movement and recover the stability. Numerical results are presented to demonstrate the accuracy and shock-capturing ability of the new scheme.

Key words. WENO, finite difference method, moving mesh method, GCL, equidistribution

AMS subject classifications. 65M06, 35L50

DOI. 10.1137/110856381

1. Introduction. We are concerned with the numerical solution of one-dimensional conservation laws in the form

$$(1.1) \qquad \begin{cases} \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \\ u(x,0) = u_0(x). \end{cases}$$

A major challenge in the numerical solution of this type of system is to capture discontinuous solutions such as shock waves with reasonable accuracy. High order schemes have proven to be advantageous in achieving this goal in many applications. We are particularly interested in WENO schemes in this paper. Weighted essentially nonoscillatory (WENO) schemes are designed based on the successful essentially nonoscillatory (ENO) schemes by Harten [7, 6]. The first WENO scheme was constructed by Liu, Osher, and Chan [15] for a third order finite volume version in one dimension. Third and fifth order finite difference WENO schemes in multidimensions were constructed by Jiang and Shu [13] with a general framework for the design of the smoothness indicators and nonlinear weights. Hu and Shu implemented the WENO

scheme for triangular meshes [9]. Qiu and Shu [17] developed finite difference WENO with Lax–Wendroff-type time discretization. Both ENO and WENO use the idea of adaptive stencils in the reconstruction procedure. ENO uses the smoothest stencil for reconstruction while WENO uses a convex combination of the candidate stencils, each being assigned a nonlinear weight depending on the local smoothness of the numerical solution based on that stencil. WENO improves ENO with better robustness, smoothness of fluxes, steady state convergence, provable convergence properties, and efficiency. WENO schemes have been widely used in applications, especially for problems with both discontinuities and complex smooth regions such as compressible turbulence simulations and aerodynamics [13, 19].

Solutions of nonlinear hyperbolic conservations laws often exhibit a wide range of localized structures, such as shock waves, contact discontinuities, and rarefaction waves. Mesh adaptation has become an indispensable tool for use in the efficient numerical solution of this type of problem (e.g., see [11]). Loosely speaking, there are three types of mesh adaptation methods. The first one is h-methods, which generate a new mesh by adding or removing points to an existing mesh. Typically, mesh points are added in the regions where the solution variation or error is large, and mesh points are removed in the regions where the solution is smooth. The second type is p-methods with which the order of polynomial approximation varies from place to place according to a certain error estimate or indicator. The third type is r-methods, also called moving mesh methods, which relocate mesh point positions while maintaining the total number of mesh points and the mesh connectivity.

In this paper we study moving mesh methods for the numerical solution of conservation laws. There exist a few research works in this aspect. Noticeably, Stockie, Mackenzie, and Russell [22] developed a Godunov-type moving mesh method for solving one-dimensional hyperbolic conservation laws. Particularly, they constructed different monitor functions to capture shocks and discontinuities in different regions of the physical solution. Wan-Lung and Tan [26] solved the Boussinesq equation using a moving mesh method developed in [24]. Tang and Tang [23] solved one- and two-dimensional conservation laws using rezoning moving mesh methods where physical PDEs were integrated on a uniform mesh over a time step and conservative interpolation was used to transfer solutions from the old mesh to the new mesh. The methods are shown to work well generally for hyperbolic conservation laws although it is unclear that they can be high order since the conservative interpolation technique used is only second order in space.

It is worth pointing out that although tremendous work has been done with WENO and adaptive mesh methods separately, only a few works have combined finite difference WENO with an adaptive mesh environment. For example, Shen, Qiu, and Christlieb [18] developed a finite difference WENO scheme with AMR (adaptive mesh refinement) which achieves the fifth order spatial accuracy and third order temporal accuracy.

Li and Hyman [14] also developed a finite difference WENO scheme with AMR which maintains the conservation property and high accuracy of WENO. Yoon, Kim, and Hwang [28] developed a simple finite difference WENO scheme with AMR. The scheme employs a special time level integration to maintain the conservation and order of WENO. Moreover, there is basically no research work done to implement finite difference WENO schemes with adaptive moving meshes in the so-called quasi-Lagrange approach (in which the mesh is considered to move continuous in time; see [11]). A major difficulty in using a high order finite difference WENO scheme in and adaptive mesh environment is that the mesh should be sufficiently smooth in both

time and space [16, 20]. Our experience also indicates that the convergence order of a high order finite difference WENO scheme can deteriorate dramatically when the mesh is not sufficiently smooth. To improve the mesh smoothness, we apply a least squares smoothing directly to the mesh. This is in addition to the commonly used smoothing of the monitor function via a low-pass filter. Numerical results show that the least squares smoothing is very effective.

With the quasi-Lagrange approach of moving mesh methods, the moving mesh is viewed as a time-dependent coordinate transformation from a computational (reference) domain to the physical domain. The associated transformation relations such as the Jacobian matrix and determinant must be approximated numerically for the discretization of the underlying physical PDEs. It is shown that the approximations of the relations should be high order to maintain the convergence order of WENO schemes. We consider several high order finite difference approximations for this purpose. We also use so-called geometric conservation laws (GCLs) [8, 25] and the WENO scheme to update the Jacobian of the coordinate transformation. Effects of satisfaction of GCLs by a numerical scheme have been extensively studied; e.g., see [1, 5, 8, 25].

Numerical experiment also shows that abrupt mesh movement can cause WENO schemes to become unconditionally unstable. To avoid this difficulty, we use a simple strategy to restrict the mesh movement. It turns out that the mesh movement restriction plays a crucial role in recovering the stability of the underlying scheme while allowing a high level of mesh adaptivity.

An outline of this paper is as follows. The moving mesh WENO method is presented in section 2. The accuracy of the method and the effects of mesh smoothness, mesh restriction, and computation of transformation relations are addressed in section 3. In section 4, we examine the performance and shock capturing capability of the new methods with a collection of examples with discontinuities. Finally, section 5 contains concluding remarks.

2. Method description. In this section, we describe the moving mesh WENO method for solving the one-dimensional conservation law (1.1). The method is based on the quasi-Lagrange approach of the moving mesh method [11] in which we first transform the physical PDE from the physical coordinates (x, t) into the computational coordinates (ξ, t) and then discretize it using a finite difference WENO scheme in space and a Runge–Kutta scheme in time.

We view a moving mesh as the image of a computational mesh under a time-dependent coordinate transformation from the computational domain to the physical domain. Denote such a coordinate transformation by $x = x(\xi, t)$: $[0, 1] \to [a, b]$. Under the transformation, we can write (1.1) as

$$(2.1) \qquad \frac{\partial u}{\partial t} + \xi_t \frac{\partial u}{\partial \xi} + \xi_x \frac{\partial f(u)}{\partial \xi} = 0.$$

Letting $J = x_\xi$ $(= 1/\xi_x)$ (the Jacobian) and multiplying both sides of (2.1) with J, we get

$$(2.2) \qquad J \frac{\partial u}{\partial t} + J \xi_t \frac{\partial u}{\partial \xi} + \frac{\partial f(u)}{\partial \xi} = 0.$$

From the identities

$$J \frac{\partial u}{\partial t} = \frac{\partial (Ju)}{\partial t} - u \frac{\partial J}{\partial t}, \quad J \xi_t \frac{\partial u}{\partial \xi} = \frac{\partial (Ju\xi_t)}{\partial \xi} - u \frac{\partial (J\xi_t)}{\partial \xi},$$

we can rewrite (2.2) as

$$(2.3) \qquad \frac{\partial(Ju)}{\partial t} + \frac{\partial(Ju\xi_t + f(u))}{\partial \xi} - u\left(\frac{\partial J}{\partial t} + \frac{\partial(J\xi_t)}{\partial \xi}\right) = 0.$$

Moreover, from

$$\frac{\partial J}{\partial t} = \frac{\partial x_\xi}{\partial t} = x_{\xi t}, \quad \frac{\partial(J\xi_t)}{\partial \xi} = \frac{\partial(x_\xi \xi_t)}{\partial \xi} = \frac{\partial(-x_t)}{\partial \xi} = -x_{\xi t},$$

we have

$$(2.4) \qquad \frac{\partial J}{\partial t} + \frac{\partial(J\xi_t)}{\partial \xi} = 0.$$

Combining this with (2.3) leads to

$$(2.5) \qquad \frac{\partial(Ju)}{\partial t} + \frac{\partial(Ju\xi_t + f(u))}{\partial \xi} = 0.$$

Equation (2.4) is often referred to as the geometric conservation law (GCL) in the literature. The GCL is an identity satisfied by the continuous coordinate transformation. It reflects the reproduction of the uniform flow, and the satisfaction of GCL by a numerical scheme often has a significant effect on the accuracy and stability of the scheme (cf. section 2.2). Equation (2.5) can be rewritten as

$$(2.6) \qquad \frac{\partial \widehat{u}}{\partial t} + \frac{\partial \widehat{f}(\widehat{u})}{\partial \xi} = 0,$$

where

$$\widehat{u} = Ju, \quad \widehat{f}(\widehat{u}) = \widehat{u}\xi_t + f(J^{-1}\widehat{u}).$$

The moving mesh WENO method is based on the conservation law (2.6) in the new coordinate system. Following the standard WENO notation, we define a uniform computational mesh as

$$(2.7) \qquad 0 = \xi_{\frac{1}{2}} < \xi_{\frac{3}{2}} < \cdots < \xi_{N-\frac{1}{2}} < \xi_{N+\frac{1}{2}} = 1,$$

where $\xi_{i+\frac{1}{2}} = i\Delta\xi$, $i = 0, \ldots, N$ and $\Delta\xi = 1/N$. The mesh cells and cell centers are denoted by

$$I_i = [\xi_{i-\frac{1}{2}}, \xi_{i+\frac{1}{2}}], \quad \xi_i = \frac{1}{2}(\xi_{i-\frac{1}{2}} + \xi_{i+\frac{1}{2}}), \quad i = 1, 2, \ldots, N.$$

The corresponding moving mesh in the physical domain is denoted by

$$(2.8) \qquad a = x_{\frac{1}{2}}(t) < x_{\frac{3}{2}}(t) < \cdots < x_{N-\frac{1}{2}}(t) < x_{N+\frac{1}{2}}(t) = b.$$

We assume that the mesh points vary linearly in time in each time slab, i.e.,

$$x_{i+\frac{1}{2}}(t) = \frac{t_{n+1} - t}{\Delta t_n} x_{i+\frac{1}{2}}^n + \frac{t - t_n}{\Delta t_n} x_{i+\frac{1}{2}}^{n+1}, \quad i = 0, \ldots, N, \quad t \in [t_n, t_{n+1}].$$

The new mesh $x_{i+1/2}^{n+1}$, $i = 0, \ldots, N$, is generated via the so-called equidistribution principle (cf. section 2.3.1). Having determined the moving mesh, the conservation

law (2.6) can be discretized in space (on a uniform computational mesh) and time. It should be pointed out that the transformation quantities J and ξ_t have to be approximated numerically in the discretization and their approximations have significant effects on the stability and accuracy of the underlying scheme. In our computation, we approximate $J = x_\xi$ using central finite differences (cf. (2.10)) initially and GCL (2.4) later; see the discussion in section 2.2. Quantity ξ_t is calculated through the relation

$$(2.9) \qquad\qquad \xi_t = -J^{-1}\dot{x}.$$

In the following, we give a flowchart of the moving mesh WENO scheme. Important components of the algorithm such as the WENO discretization, satisfaction of GCL, mesh movement and smoothing, and mesh movement restriction will be elaborated later.

The moving mesh WENO algorithm.

Step 1. Initialization.

(a) Start with a uniform mesh and an initial solution.

(b) An adaptive mesh based on the initial solution is computed using de Boor's algorithm (to be described in section 2.3.1). This mesh is denoted by x^0 and used as the initial mesh for the integration of conservation laws. Compute the initial solution u^0 on mesh x^0.

(c) Approximate the transformation quantity $J = x_\xi$ by a fourth order finite difference (FD4),

(2.10)

$$J(\xi_{i+1/2}, t_0) \approx \frac{x^0_{i-\frac{3}{2}} - 8x^0_{i-\frac{1}{2}} + 8x^0_{i+\frac{1}{2}} + x^0_{i+\frac{3}{2}}}{12\Delta\xi}, \quad i = 2, \dots, N-1.$$

The approximation near the boundary is computed using a periodic or extrapolative extension corresponding to the periodic or inflow/outflow boundary condition of the underlying problem.

(d) Set $n = 0$, $t = t_0$, and $\widetilde{\Delta}t_0 = \Delta t_0$.

Step 2. Mesh movement. Assume that the physical solution u^n, the mesh x^n, and the metric derivative $J^n = x^n_\xi$, and the time step size $\widetilde{\Delta}t_n$ are known at time $t = t_n$.

(a) *The prediction step.* Integrate the conservation laws using a finite difference WENO (fifth order or third order) scheme on mesh x^n. The mesh is *fixed* during the integration. The obtained physical solution is denoted by $\bar{u}^n$ and used for computing the new adaptive mesh.

(b) *Generating the new mesh.* Compute the monitor function $M^n(x) = M(t_n, x)$ based on $\bar{u}^n$ and x^n. The needed solution derivatives are calculated using a gradient recovery technique based on the nodal values of the computed solution. The new mesh, denoted by $\widetilde{x}^{n+1}$, is obtained using de Boor's algorithm. The mesh speed is computed as

$$(2.11) \qquad \dot{x}^n_{i+\frac{1}{2}} = \frac{\widetilde{x}^{n+1}_{i+\frac{1}{2}} - x^n_{i+\frac{1}{2}}}{\widetilde{\Delta}t_n}, \quad i = 0, \dots, N.$$

(c) *Mesh movement restriction.* The mesh movement is restricted so that the mesh points do not move too far away from their current locations

over a time step; see section 2.3.3. This step is necessary for ensuring the stability of the numerical method. The restriction typically results in a smaller time step, $\Delta t_n \leq \widetilde{\Delta} t_n$. The mesh is updated as

$$(2.12) \qquad x_{i+\frac{1}{2}}^{n+1} = x_{i+\frac{1}{2}}^n + \Delta t_n \dot{x}_{i+\frac{1}{2}}^n, \quad i = 0, \ldots, N.$$

(d) The transformation quantity $J^{n+1} = x_\xi^{n+1}$ for the new mesh is computed using the WENO scheme (which is the same as that used in Step 2(a) and Step 3) for GCL (2.4) (cf. section 2.2).

Step 3. Integration of conservation laws. The conservation laws (2.6) are integrated using the WENO scheme (to be described in section 2.1) from t_n to $t_{n+1} = t_n + \Delta t_n$ and the obtained solution is denoted by u^{n+1}.

Step 4. If $t_{n+1} < T$, compute $\widetilde{\Delta} t_{n+1}$ using the CFL condition (3.1) and go to Step 2.

2.1. WENO for the transformed conservation laws. In this subsection, we give a brief description of the finite difference WENO method applied to the transformed conservation law (2.6) on the uniform mesh (2.7) in ξ. A semidiscrete conservative high order finite difference scheme for (2.6) can be cast in the form

$$(2.13) \qquad \frac{d\widehat{u}_i}{dt} = -\frac{1}{\Delta \xi}(\widehat{f}_{i+1/2} - \widehat{f}_{i-1/2}),$$

where $\widehat{u}_i(t) \approx \widehat{u}(\xi_i, t)$, $\widehat{f}_{i+1/2} \approx h(\xi_{i+1/2})$, and $h(\xi)$ is a function implicitly defined [13] as

$$\widehat{f}(\widehat{u}(\xi)) = \frac{1}{\Delta \xi} \int_{\xi - \Delta \xi/2}^{\xi + \Delta \xi/2} h(s)ds.$$

For the case where (2.6) is a scalar equation and $\widehat{f}'(\widehat{u}) \geq 0$, the numerical flux of the fifth order finite difference WENO scheme is given by

$$(2.14) \qquad \widehat{f}_{i+\frac{1}{2}} = \omega_1 \widehat{f}_{i+\frac{1}{2}}^{(1)} + \omega_2 \widehat{f}_{i+\frac{1}{2}}^{(2)} + \omega_3 \widehat{f}_{i+\frac{1}{2}}^{(3)},$$

where $\widehat{f}_{i+\frac{1}{2}}^{(i)}$ are three third order fluxes on three different stencils given by

$$\widehat{f}_{i+\frac{1}{2}}^{(1)} = \frac{1}{3}\widehat{f}(\widehat{u}_{i-2}) - \frac{7}{6}\widehat{f}(\widehat{u}_{i-1}) + \frac{11}{6}\widehat{f}(\widehat{u}_i),$$

$$\widehat{f}_{i+\frac{1}{2}}^{(2)} = -\frac{1}{6}\widehat{f}(\widehat{u}_{i-1}) + \frac{5}{6}\widehat{f}(\widehat{u}_i) + \frac{1}{3}\widehat{f}(\widehat{u}_{i+1}),$$

$$\widehat{f}_{i+\frac{1}{2}}^{(3)} = \frac{1}{3}\widehat{f}(\widehat{u}_i) + \frac{5}{6}\widehat{f}(\widehat{u}_{i+1}) - \frac{1}{6}\widehat{f}(\widehat{u}_{i+2}),$$

and the nonlinear weights ω_i are given by

$$\omega_i = \frac{\tilde{\omega}_i}{\sum_{k=1}^3 \tilde{\omega}_k}, \quad \tilde{\omega}_k = \frac{\gamma_k}{(\epsilon + \beta_k)^2}.$$

Here, ϵ is a small positive parameter used to avoid the denominator to become zero (taken as $\epsilon = 10^{-6}$ in our computation), γ_k's are the linear weights defined as

$$\gamma_1 = \frac{1}{10}, \qquad \gamma_2 = \frac{3}{5}, \qquad \gamma_3 = \frac{3}{10},$$

and β_k's are the smoothness indicators given by

$$\beta_1 = \frac{13}{12}(\widehat{f}(\widehat{u}_{i-2}) - 2\widehat{f}(\widehat{u}_{i-1}) + \widehat{f}(\widehat{u}_i))^2 + \frac{1}{4}(\widehat{f}(\widehat{u}_{i-2}) - 4\widehat{f}(\widehat{u}_{i-1}) + 3\widehat{f}(\widehat{u}_i))^2,$$

$$\beta_2 = \frac{13}{12}(\widehat{f}(\widehat{u}_{i-1}) - 2\widehat{f}(\widehat{u}_i) + \widehat{f}(\widehat{u}_{i+1}))^2 + \frac{1}{4}(\widehat{f}(\widehat{u}_{i-1}) - \widehat{f}(\widehat{u}_{i+1}))^2,$$

$$\beta_3 = \frac{13}{12}(\widehat{f}(\widehat{u}_i) - 2\widehat{f}(\widehat{u}_{i+1}) + \widehat{f}(\widehat{u}_{i+2}))^2 + \frac{1}{4}(3\widehat{f}(\widehat{u}_i) - 4\widehat{f}(\widehat{u}_{i+1}) + \widehat{f}(\widehat{u}_{i+2}))^2.$$

For scalar equations not satisfying the property $\widehat{f}'(\widehat{u}) \geq 0$, we can use the flux splitting

$$\widehat{f}(\widehat{u}) = \widehat{f}^+(\widehat{u}) + \widehat{f}^-(\widehat{u}), \quad \frac{d\widehat{f^+}}{d\widehat{u}} \geq 0, \quad \frac{d\widehat{f^-}}{d\widehat{u}} \leq 0.$$

In our computation we use Lax–Friedrichs flux splitting, where

$$\widehat{f}^\pm(\widehat{u}) = \frac{1}{2}(\widehat{f}(\widehat{u}) \pm \alpha\widehat{u}), \quad \alpha = \max_{\widehat{u}}|\widehat{f}'(\widehat{u})|.$$

With the numerical fluxes defined, scheme (2.13) together with appropriate boundary conditions forms a system of ordinary differential equations which can be written as

$$\widehat{\mathbf{u}}_t = L(\widehat{\mathbf{u}}, t).$$

The system can be integrated using a Runge–Kutta scheme. In our computation, we use a third order Runge–Kutta scheme, i.e.,

$$(2.15) \quad \begin{cases} \widehat{\mathbf{u}}^{(1)} = \widehat{\mathbf{u}}^n + \Delta t L(\widehat{\mathbf{u}}^n, t^n), \\[4pt] \widehat{\mathbf{u}}^{(2)} = \frac{3}{4}\widehat{\mathbf{u}}^n + \frac{1}{4}\widehat{\mathbf{u}}^{(1)} + \frac{1}{4}\Delta t L(\widehat{\mathbf{u}}^{(1)}, t^n + \Delta t), \\[4pt] \widehat{\mathbf{u}}^{n+1} = \frac{1}{3}\widehat{\mathbf{u}}^n + \frac{2}{3}\widehat{\mathbf{u}}^{(2)} + \frac{2}{3}\Delta t L(\widehat{\mathbf{u}}^{(2)}, t^n + \frac{1}{2}\Delta t). \end{cases}$$

For the case where (2.6) is a system of conservation laws, the WENO approximation is constructed using a local characteristic decomposition. To save space, we refer the interested reader to [19] for details of the decomposition.

2.2. Geometric conservation law. We have seen that GCL (2.4) is an identity satisfied by the continuous coordinate transformation. It reflects the so-called uniform flow reproduction in the sense that it can be obtained by inserting the uniform flow solution $u(x, t) \equiv 1$ into (2.6). Unfortunately, the finite difference WENO scheme described in the previous subsection generally does not preserve this property; i.e., $u_i^n \equiv 1$ does not necessarily imply $u_i^{n+1} \equiv 1$. This means that the scheme is not consistent with the underlying differential equations. The violation of GCL typically affects the accuracy and stability of the underlying numerical scheme.

A remedy for this difficulty was proposed by Hindman [8]. Applying to our case, it requires that the Jacobian $J = x_\xi$ be considered as an unknown variable and computed by applying the same finite difference WENO scheme to GCL (2.4). (This is in contrast to a direct finite difference approximation of x_ξ.) Our numerical examples show that this enforcement of GCL is crucial to the accuracy of the moving mesh WENO scheme.

2.3. Mesh movement, restriction, and smoothing. In this subsection, we describe the mesh generation with emphasis on mesh movement, restriction, and smoothing.

2.3.1. De Boor's algorithm. For the moment, we assume that a monitor function $M = M(x)$ has been defined on the current mesh $x_{i+\frac{1}{2}}^n$, $i = 0, \dots, N$. The new mesh is determined by the so-called equidistribution principle [4]. Indeed, the principle requires that $M(x)$ be evenly distributed among the mesh cells, i.e.,

$$(2.16) \qquad \int_{\widetilde{x}_{i-1/2}^{n+1}}^{\widetilde{x}_{i+1/2}^{n+1}} M(x)dx = \frac{1}{N} \int_a^b M(x)dx, \quad i = 1, \dots, N.$$

The above equation can be solved for the mesh using de Boor's algorithm [4]. With the algorithm, the monitor function $M(x)$ is considered as a piecewise constant function, i.e.,

$$(2.17) \qquad M(x) = \frac{1}{2}\left(M_{i-1/2}^n + M_{i+1/2}^n\right), \quad x \in [x_{i-1/2}^n, x_{i+1/2}^n], \ i = 1, \dots, N,$$

where $M_{i+1/2}^n$ denotes the nodal value of the monitor function at $x = x_{i+1/2}^n$. Denoting

$$P(x) = \int_a^x M(x)dx,$$

it is easy to see that

$$P(x_{j+1/2}^n) = \sum_{i=1}^j (x_{i+1/2}^n - x_{i-1/2}^n)\frac{M_{i+1/2}^n + M_{i-1/2}^n}{2}, \quad j = 1, \dots, N.$$

It follow from (2.16) that

$$P(\widetilde{x}_{i+1/2}^{n+1}) = \xi_{i+1/2}P(b), \qquad i = 1, \dots, N-1.$$

Since $P(x)$ is piecewise linear, it is easy to show that

$$(\widetilde{x}_{i+1/2}^{n+1} - x_{k-1/2}^n)\frac{M_{k-1/2}^n + M_{k+1/2}^n}{2} = \xi_{i-1/2}P(b) - P(x_{k-1/2}^n)$$

or

$$\widetilde{x}_{i+1/2}^{n+1} = x_{k-1/2}^n + \frac{2(\xi_{i+1/2}P(b) - P(x_{k-1/2}^n))}{M_{k-1/2}^n + M_{k+1/2}^n},$$

where k is an index satisfying

$$P(x_{k-1/2}^n) < \xi_{i+1/2}P(b) \le P(x_{k+1/2}^n).$$

It is remarked that de Boor's algorithm described above is simple and robust in one dimension but cannot be applied to multidimensional mesh generation. Multidimensional extensions of the equidistribution principle and related meshing strategies can be seen, e.g., in Huang and Russell [10, 11].

2.3.2. Monitor functions. In the preceding subsection we have seen that the monitor function controls the concentration of the mesh points, and thus its choice is the key to the success of the mesh adaptation strategy. The choice of the optimal monitor function according to interpolation error estimates has been extensively discussed; e.g., see [2, 3, 12]. In our computation, we use

$$(2.18) \qquad M = \left(1 + \frac{1}{\alpha}|u'|^2\right)^{\frac{1}{3}},$$

where the intensity controlling parameter α is defined as

$$\alpha = \max\left\{1, \left[\frac{1}{b-a}\int_a^b |u'|^{\frac{2}{3}}dx\right]^3\right\}.$$

The main reason for choosing this monitor function is that it involves only the first derivative of the solution whose recovery (see the discussion below) is generally smoother and less expensive to compute than higher derivatives.

In practical computation, the solution derivative involved in (2.18) is replaced by a numerical approximation based on the nodal values of the computed solution at the current time step. Such an approximation can be obtained using a direct finite difference or a more sophisticated gradient recovery technique. We use in our computation a least squares method [11, section 2.5.1]. To be specific, for any given point $x_{j+1/2}$, let $x_{j+1/2}^i$, $i = 1, \ldots, 2p+1$ $(p \geq 1)$, be the $2p + 1$ neighboring mesh points which are closest to $x_{j+1/2}$ (including $x_{j+1/2}$ itself). Define the center of those points as $\hat{x}_{j+1/2}$. Let $H_{j+1/2} = \max_{i=1,\ldots,2p+1} |x_{j+1/2}^i - \hat{x}_{j+1/2}|$, and denote the first three Legendre polynomials by

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}.$$

A least squares quadratic polynomial

$$(2.19) \qquad q(x) = \sum_{k=0}^{2} a_k P_k\left(\frac{x - \hat{x}_{j+1/2}}{H_{j+1/2}}\right)$$

is determined by minimizing

$$(2.20) \qquad \min_{a_0, a_1, a_2} \sum_{i=1}^{2p+1} (q(x_{j+1/2}^i) - u(x_{j+1/2}^i))^2.$$

Then $q'(x_{j+1/2})$ is used as the approximation to the solution derivative at $x_{j+1/2}$. We take $p = 1$ in the gradient recovery.

Generally speaking, the recovered solution derivative contains numerical oscillations which typically lead to a nonsmooth mesh. To obtain a smoother mesh, a common approach in the moving mesh context is to smooth the monitor function each time after it has been computed. We use a low-pass filter

$$(2.21) \qquad \tilde{M}_{i+1/2}^n = \frac{1}{4}M_{i-1/2}^n + \frac{1}{2}M_{i+1/2}^n + \frac{1}{4}M_{i+3/2}^n, \quad i = 1, \ldots, N-1,$$

with appropriate modifications for the boundary points. Several sweeps of the filter are used each time. (Twenty sweeps are used for all examples considered in this

paper except for Examples 4.4, 4.5, and 4.6 for which 40, 100, and 200 sweeps are used, respectively, for extra smoothness of the mesh.)

Our numerical experience also shows that the accuracy of the moving mesh WENO scheme relies crucially on the smoothness of the mesh. To further smooth the mesh, we apply to the newly generated mesh a least squares smoothing similar to (2.19) and (2.20) used for gradient recovery. Specifically, a quadratic polynomial $q(\xi)$ associated with a given point $\xi_{j+\frac{1}{2}}$ is constructed by the least squares fitting to a $(2p+1)$-neighboring-point subset of the data sets $(\xi_{i+\frac{1}{2}}, \widetilde{x}_{i+\frac{1}{2}}^{n+1})$, $i = 0, \ldots, N$. Then the mesh location at $\xi_{j+\frac{1}{2}}$ is updated as the value of the quadratic polynomial at the point, i.e., $q(\xi_{j+\frac{1}{2}}) \to \widetilde{x}_{j+\frac{1}{2}}^{n+1}$. The parameter p is taken as $p = 10$ in most of our computation. Its effects on the accuracy of the underlying scheme is discussed in section 3.2.

2.3.3. Mesh movement restriction. An important component of the moving mesh WENO scheme is the restriction of mesh movement. Numerical experiments show that the scheme becomes unstable (even for a tiny time step) when mesh points move too far away from their current location over one time step (cf. section 3.2). A simple, heuristic strategy [11, section 2.6.1] is used for this purpose. Define the mesh speed $\dot{x}_{i+1/2}^n$ as in (2.11). We then require that the new mesh location $x_{i+1/2}^{n+1}$ defined in (2.12) stay within the interval $[x_i^n, x_{i+1}^n]$, i.e.,

$$x_i^n = \frac{1}{2}\left(x_{i-1/2}^n + x_{i+1/2}^n\right) \leq x_{i+1/2}^n = x_{i+1/2}^n + \Delta t_n \dot{x}_{i+1/2}^n \leq x_{i+1}^n$$
$$= \frac{1}{2}\left(x_{i+1/2}^n + x_{i+3/2}^n\right).$$

This implies that

(2.22) $$\Delta t_n \leq \begin{cases} -\frac{1}{2\dot{x}_{i+1/2}^n}\left(x_{i+1/2}^n - x_{i-1/2}^n\right) & \text{if } \dot{x}_{i+1/2}^n < 0, \\ +\frac{1}{2\dot{x}_{i+1/2}^n}\left(x_{i+3/2}^n - x_{i+1/2}^n\right) & \text{if } \dot{x}_{i+1/2}^n \geq 0. \end{cases}$$

The new time step size Δt_n is then chosen as the minimum of the right-hand term of the above equation over all the mesh points.

3. Numerical tests. In this section we present numerical results obtained with the moving mesh WENO scheme described in the previous section for a selection of problems. Both the third and fifth order WENO schemes are used in our computation. We observed that they attain the predicted convergence orders for the accuracy test examples and produce sharp, ENO shock transitions for shock test examples. To save space, we show only the results obtained by the fifth order WENO scheme. For all computation, the third order TVD Runge–Kutta method (2.15) is used for time integration. The CFL number (for the time step selection) is taken as $CFL = 0.8$ and $\widetilde{\Delta t_n}$ is chosen as

(3.1) $$\alpha \frac{\widetilde{\Delta t_n}}{\Delta \xi} = CFL,$$

where $\alpha = \sup_{\hat{u}} |\hat{f}'(\hat{u})|$. An exception is the accuracy tests where a smaller time step $\alpha \frac{\widetilde{\Delta t_n}}{\Delta \xi^{5/3}} = CFL$ is used so that the temporal discretization is sufficiently small

and the solution error for the fifth order WENO scheme is dominated by the spatial discretization error.

It is noted that the time step is controlled by both the CFL condition and the mesh movement restriction. As shown in the flowchart of the moving mesh WENO scheme in section 2, the time step $\widetilde{\Delta t}_n$ is first computed using (3.1) and then a time step Δt_n is computed to satisfy conditions $\Delta t_n \le \widetilde{\Delta t}_n$ and (2.22). This double control mechanism is necessary to ensure the stability of the underlying scheme. CFL condition (3.1) is the stability condition for the WENO scheme with a uniform mesh. Moreover, as will be seen in section 3.2, for problems with strong discontinuities the mesh points can move rapidly into the area where the discontinuities form. In this case, without mesh movement restriction the moving mesh WENO scheme can become unstable even for very small CFL numbers. On the other hand, when the solution is smooth and there is no significant mesh movement, the CFL condition should govern the time step.

3.1. Accuracy test. We first examine the accuracy of the scheme using three examples with exact solutions: a linear scalar problem, a nonlinear scalar problem, and a nonlinear system.

Example 3.1. We consider a linear scalar conservation law equation

$$(3.2) \qquad\qquad u_t + u_x = 0,$$

with the initial condition $u(x,0) = 1 + 0.2\sin(x)$ and a periodic boundary condition, the computational domain $[0, 2\pi]$, and the final time $T = 2.0$. The exact solution of this problem is given by

$$u = 1 + 0.2\sin(x - t).$$

The error of the numerical solution obtained with the moving mesh WENO (WENO5) is shown in Figure 1. It can be seen that the convergence order is about fifth, which is consistent with the theoretical prediction.
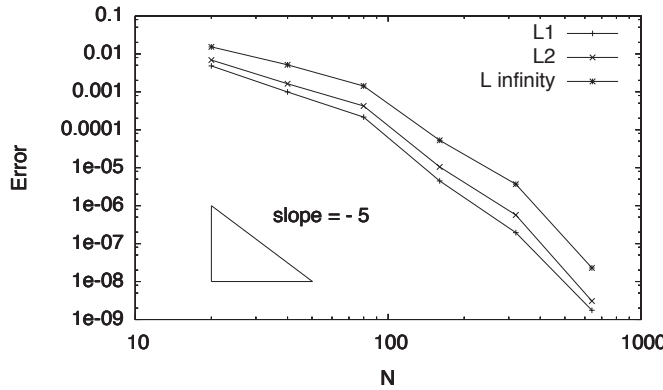

FIG. 1. *Example 3.1 (linear scalar conservation law equation). The L_1, L_2, and L_∞ norm of the error in the numerical solution at $T = 2$ is plotted as a function of the number of mesh points.*

Example 3.2. The second example is a nonlinear scalar equation—Burgers' equation,

$$(3.3) \qquad\qquad u_t + \left(\frac{u^2}{2}\right)_x = 0,$$

subject to the initial condition $u(x,0) = \frac{1}{3} + \frac{2}{3}\sin(x)$ and a periodic boundary condition. The exact solution for this problem is defined implicitly through the equation

$$u(x,t) = \frac{1}{3} + \frac{2}{3}\sin(x - u(x,t)t).$$

The exact solution is smooth before $t = 1$ and can be computed using the Newton iteration. The error at $T = 1$ is plotted in Figure 2, which confirms the fifth order convergence of the underlying scheme.
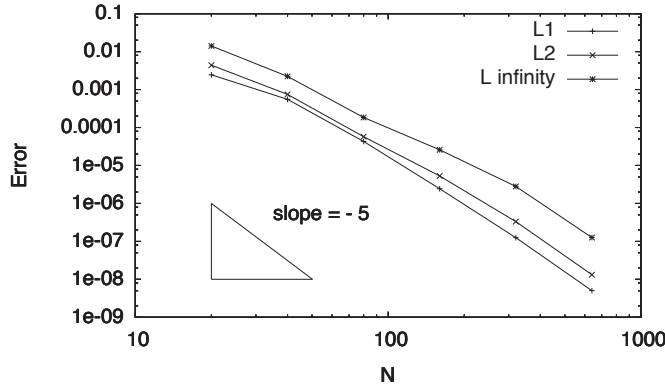

FIG. 2. *Example 3.2 (Burgers' equation). The L_1, L_2, and L_∞ norm of the error in the numerical solution at $T = 1$ is plotted as a function of the number of mesh points.*

Example 3.3. This example is a system of nonlinear equations—the Euler equations,

(3.4) $w_t + f(w)_x = 0,$

where

$$w = (\rho, \rho u, E)^T, \quad f(w) = (\rho u, \rho u^2 + p, u(E + p))^T,$$

ρ is the density, u is the velocity, E is the energy density, and p is the pressure. The equation of state is $E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2$ with $\gamma = 1.4$. A periodic boundary condition is used and the initial condition is set to be $\rho(x,0) = 1.0 + 0.2\sin(x)$, $u(x,0) = 0.5$, and $p(x,0) = 1.0$. The exact solution for this problem is

$$\rho(x,t) = 1.0 + 0.2\sin(x - 0.5t), \quad u(x,t) = 0.5, \quad p(x,t) = 1.$$

The final time for the computation is $T = 2.0$. We used the entropy $S = \log(\rho^\gamma/p)$ for computing the monitor function (cf. section 2.3.2). The error in the density is plotted in Figure 3, which, once again, shows the fifth order convergence of the scheme.

3.2. Effects of GCL satisfaction, mesh smoothness, and mesh movement restriction. These effects are demonstrated using Example 3.2 (Burgers' equation). First, to demonstrate the effect of GCL satisfaction we consider the use of second (FD2), fourth (FD4, (2.10)), and sixth (FD6) order finite difference approximations for $J^{n+1} = x_\xi^{n+1}$ in Step 2(d) of the algorithm described in section 2. The numerical results obtained using these approximations and GCL are shown in Figure 4. From the figure, we can see that FD2 is not acceptable. Moreover, FD4 is
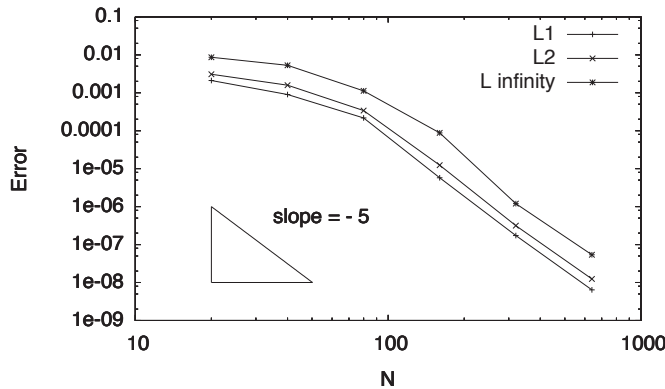
FIG. 3. *Example 3.3 (Euler equations). The L_1, L_2, and L_∞ norm of the error in the density at $T = 2$ is plotted as a function of the number of mesh points.*
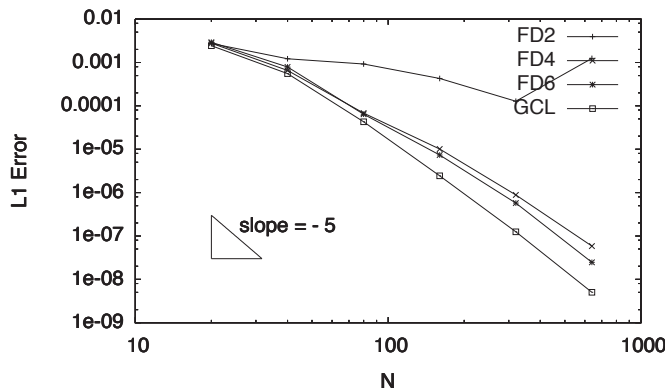

FIG. 4. *Example 3.2 (Burgers' equation). The L_1 norm of the error at $T = 1$ in the numerical solution is plotted as a function of the number of mesh points. The moving mesh WENO scheme with finite difference approximations or the GCL procedure for the Jacobian of the coordinate transformation is used.*

acceptable but leads to only a fourth order convergence. Finally, FD6 gives the correct (fifth) order of the WENO scheme although the results are slightly worse than those obtained based on GCL. These results show that the satisfaction of GCL is not necessary, but it does improve the accuracy of the underlying scheme. It can be replaced with a sufficiently high order finite difference approximation of x_ξ.

Next, we examine the effects of smoothness of the mesh using the least squares smoothing described in the last paragraph of section 2.3.2. The results obtained with different values of p are shown in Figure 5. Notice that the greater p is, the smoother the mesh. The results confirm the observation (e.g., see Shu [20]) that the mesh needs to be sufficiently smooth in order to achieve the designed accuracy order.

Finally, we examine the effects of mesh movement restriction on stability of the moving mesh WENO scheme. They are significant only when the mesh points move fast enough. For this reason, we integrate Burgers' equation for a long time $T = 2$ when a sharp shock wave has formed and propagated to the right end of the domain. The computation was done without mesh movement restriction for CFL numbers
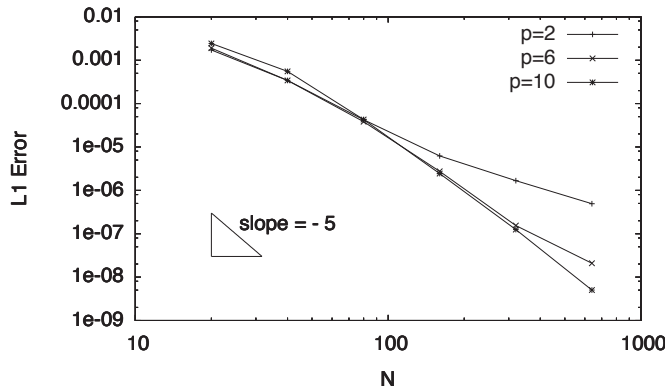
FIG. 5. *Example 3.2 (Burgers' equation). The L_1 norm of the error at $T = 1$ in the numerical solution is plotted as a function of the number of mesh points. The moving mesh WENO scheme with various values of p in the least squares smoothing of the mesh is used.*

ranging from 0.001 to 0.8. All of the numerical solutions blow up, indicating that the underlying scheme without mesh movement restriction is unconditionally unstable. On the other hand, as will be seen in Example 4.2, the scheme with mesh movement restriction is stable as long as the time step satisfies the CFL condition.

4. Examples with shocks. We now present numerical results obtained with the moving mesh WENO scheme for a number of examples containing discontinuities (shocks, rarefaction waves, and contact discontinuities). These examples either do not have exact solutions or have discontinuous solutions. The L_1 error is computed for the three system examples (Lax problem, Shu–Osher problem, and blast-wave problem) against a numerical solution obtained with 6400 uniform mesh points. The performance of the scheme is examined by comparing moving mesh results with uniform mesh results.

Example 4.1. This example is the linear equation (3.2) subject to a periodic boundary condition and the initial condition

$$u(x,0) = \begin{cases} \frac{1}{6}(G(x, z - \delta) + G(x, z + \delta) + 4G(x, z)) & \text{for } x \in (-0.8, -0.6), \\ 1 & \text{for } x \in (-0.4, -0.2), \\ 1 - |10(x - 0.1)| & \text{for } x \in (0, 0.2), \\ \frac{1}{6}(F(x, a - \delta) + F(x, a + \delta) + 4F(x, a)) & \text{for } x \in (0.4, 0.6), \\ 0 & \text{otherwise,} \end{cases}$$

where

$$G(x, z) = e^{-\beta(x-z)^2}, \quad F(x, a) = \sqrt{\max(1 - \alpha^2(x - a)^2, 0)},$$

$$a = 0.5, \quad z = -0.7, \quad \delta = 0.005, \quad \alpha = 10, \quad \beta = \frac{\log 2}{36\delta^2}.$$

The exact solution is a traveling wave formed by the combination of Gaussian and square-, sharp triangle-, and half ellipse-shaped waves. The domain is taken as $[-1, 1]$, and the final time is $T = 2$. A moving mesh solution ($N = 200$) is compared with the solutions obtained with uniform meshes of $N = 400$ and $N = 800$ in Figure 6.
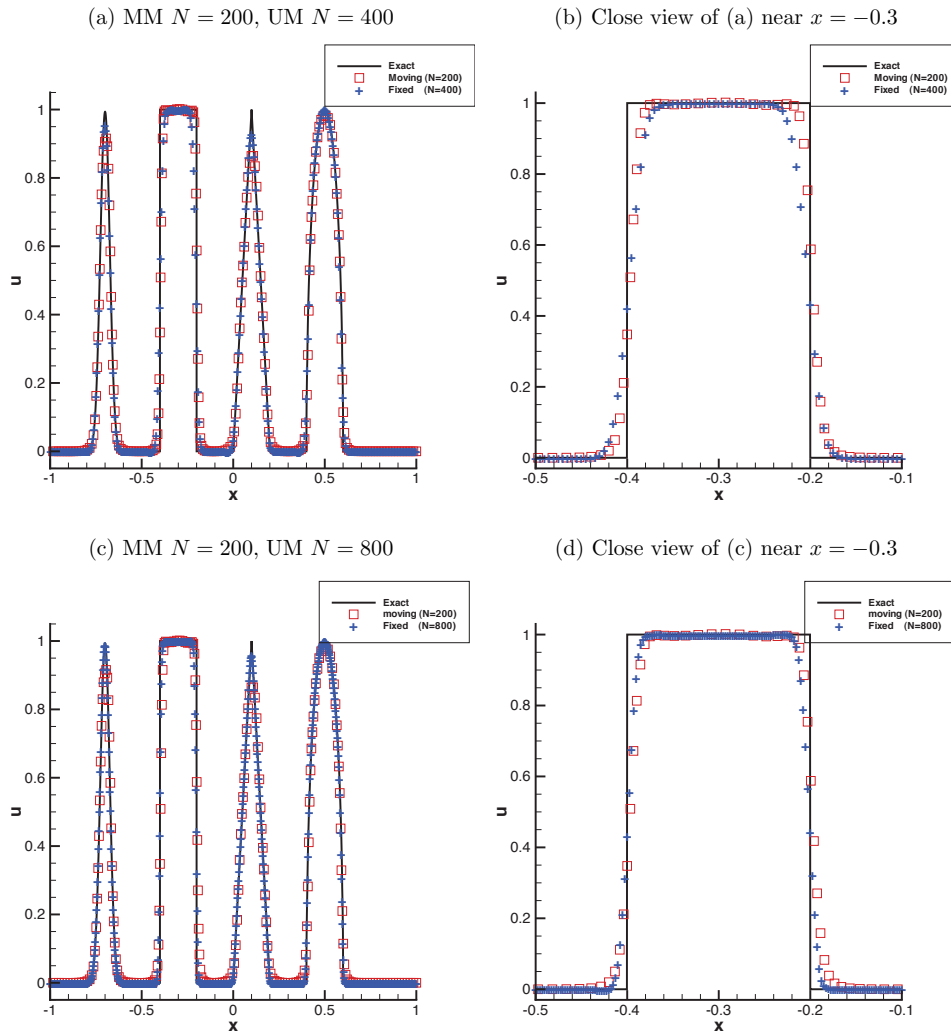
(a) MM $N = 200$, UM $N = 400$ (b) Close view of (a) near $x = -0.3$

(c) MM $N = 200$, UM $N = 800$ (d) Close view of (c) near $x = -0.3$

FIG. 6. *Example 4.1 (linear scalar conservation laws). The moving mesh solution with $N = 200$ is compared with the uniform mesh solutions with $N = 400$ and $N = 800$. Here, MM and UM mean moving mesh and uniform mesh, respectively.*

The figure shows that the result obtained by the moving mesh method with $N = 200$ is better than that with $N = 400$ uniform points and comparable to that with $N = 800$ uniform points.

Example 4.2. This example is basically the same as Example 3.2 except Burgers' equation is integrated for a longer time $T = 2.0$ when a shock has formed and is propagating towards the right end of the domain. The trajectories of a moving mesh $N = 80$ are shown in Figure 7, and Figure 8 shows that the moving mesh solution ($N = 80$) is better than those obtained with 160 and 320 uniform points.

Example 4.3. We consider in this example the nonlinear Buckley–Leverett problem

$$u_t + \left(\frac{4u^2}{4u^2 + (1 - u)^2} \right)_x = 0, \quad x \in (-1, 1), \quad t < 0.4,$$
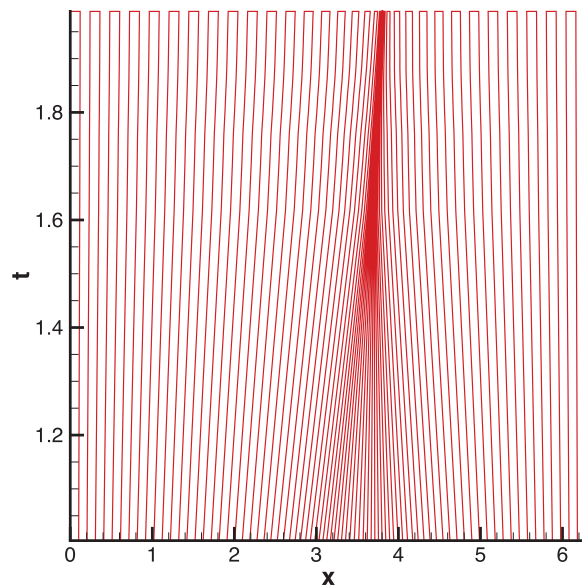
FIG. 7. *Example 4.2 (Burgers' equation). The trajectories of a moving mesh with $N = 80$ are plotted.*

subject to the initial condition

$$u = \begin{cases} 1 & \text{for } x \in (-\frac{1}{2}, 0), \\ 0 & \text{otherwise} \end{cases}$$

and the inflow/outflow boundary condition. The exact solution is a shock-rarefaction-contact discontinuity mixture. The trajectories of a moving mesh $N = 80$ are shown in Figure 9, and the moving mesh solution ($N = 80$) is compared with those obtained with 160 and 320 uniform points in Figure 10. It can be seen that the moving mesh solution is better than that obtained with 160 uniform points and comparable with that obtained with 320 uniform points.

Example 4.4. This example is the Lax problem of the Euler equations (3.4) subject to the inflow/outflow boundary condition and a Riemann initial condition

$$(\rho, v, p) = \begin{cases} (0.445, 0.698, 3.528) & \text{for } x \le 0, \\ (0.5, 0, 0.571) & \text{for } x > 0. \end{cases}$$

The physical domain is taken as $(-5, 5)$ and the integration is stopped at $T = 1.3$. As in Example 3.3, the monitor function is computed based on the entropy. The trajectories of a moving mesh are shown in Figure 11. From the figure (particularly Figure 11(b)), one can see that there is a sharp readjustment of mesh points at the first time step. This is because the mesh is equidistributing to the initial solution and the solution at the first time step that have different discontinuity strengths. The initial solution for this example is discontinuous at $x = 0$ and its strength of discontinuity is infinity. On the other hand, this initial discontinuity is smeared by the WENO scheme (and spatial discretization), and the strength of discontinuity of the solution at the first time step is at most conversely proportional to the minimal spacing of the initial mesh. As a result of the different discontinuity strengths, the
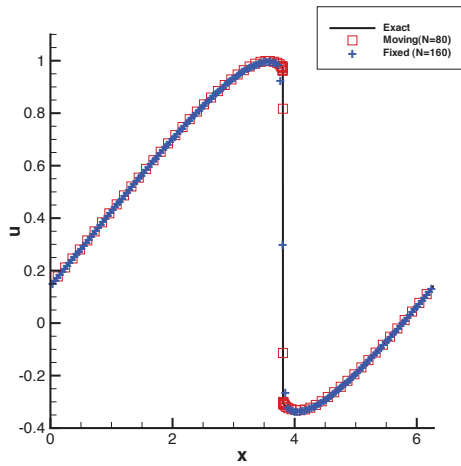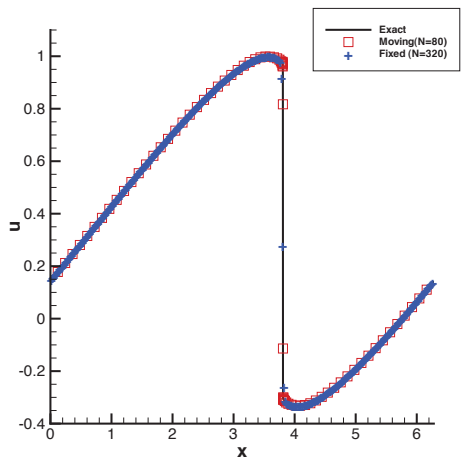
(a) MM $N = 80$, UM $N = 160$ (b) Close view of (a) near $x = 3.8$
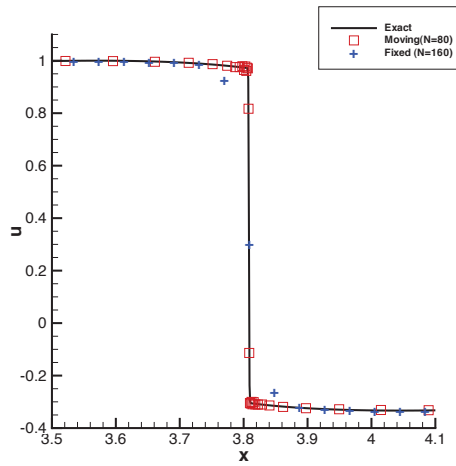
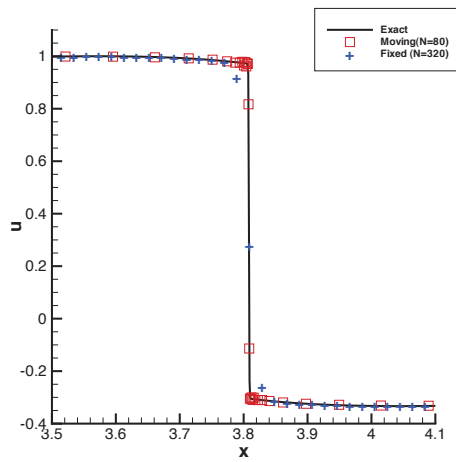(c) MM $N = 80$, UM $N = 320$ (d) Close view of (c) near $x = 3.8$

FIG. 8. *Example 4.2 (Burgers' equation). The moving mesh solution with $N = 80$ is compared with the uniform mesh solutions with $N = 160$ and $N = 320$.*

mesh has different degrees of concentration initially and at the first time step, and thus the sharp readjustment occurs.

The moving mesh solution obtained with $N = 200$ is compared with uniform mesh solutions obtained with $N = 400$ and $N = 800$ in Figure 12. From the figure we can see that the shock and contact discontinuity are captured by the monitor function. The moving mesh solution of shock is clearly better than that obtained with 400 uniform points and even comparable to that with 800 uniform points. One may also notice from Figures 12(c) and (f) that there is a dip to the right of the rarefaction in the uniform mesh solutions with 400 and 800 points but not in the moving mesh solution, which can be considered an advantage of the moving mesh method.

Finally, we will consider the L_1 error of moving mesh and uniform mesh solutions. The L_1 errors of density are shown in Table 1. From it we see that the L_1 error of
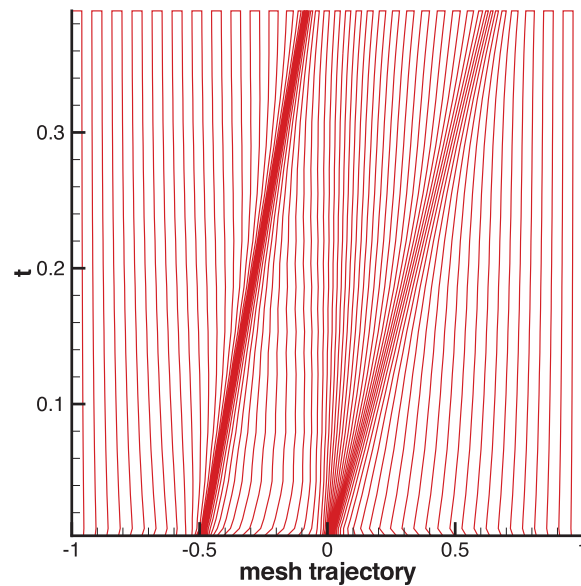
FIG. 9. *Example 4.3 (Buckley–Leverett equation). The trajectories of a moving mesh with N = 80 are plotted.*

the moving mesh solution with $N = 200$ is clearly smaller than that obtained with 600 uniform points, and the L_1 error of the moving mesh solution with $N = 300$ is smaller than that obtained with 800 uniform points. These results are consistent with what is shown in Figure 12.

Example 4.5. We now consider the Shu–Osher problem [21], which contains both shocks and complex smooth region structures. It models a shock interaction with entropy waves and is governed by the Euler equations (3.4) subject to the inflow/outflow boundary condition and the initial condition

$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{for } x < -4, \\ (1 + \varepsilon \sin(5x), 0, 1) & \text{for } x \geq -4. \end{cases}$$

We used $\varepsilon = 0.2$ and the physical domain as $(-5, 5)$ in our computation. The computed density is plotted at $T = 1.8$ against the reference "exact" solution obtained using the fifth order finite difference WENO scheme [13] with 6400 uniform points. The trajectories of a typical moving mesh are shown in Figure 13. The moving mesh solutions obtained with $N = 200$ and 400 are compared with the uniform mesh solutions obtained with $N = 200, 300, 400$ and $N = 400, 600, 800$ in Figures 14 and 15, respectively. From Figure 14(a) and (b), we can see that for the same number of mesh points, the moving mesh results are better than the uniform mesh ones. Particularly, the complex oscillations in the middle of the computational region are well resolved by the moving mesh but not by the uniform mesh. From Figure 14(c), we see that the moving mesh solution with $N = 200$ is better than that obtained with 300 uniform points.

If we increase the number of uniform mesh points to $N = 400$ while keeping the number of moving mesh points the same ($N = 200$), we can see from Figure 14(d) that the first few oscillations on the left are resolved better with 400 uniform points. But in the rest of the physical domain, the moving mesh results are comparable or

(a) MM $N = 80$, UM $N = 160$ (b) Close view of (a) near $x = -0.075$

(c) MM $N = 80$, UM $N = 320$ (d) Close view of (c) near $x = -0.075$
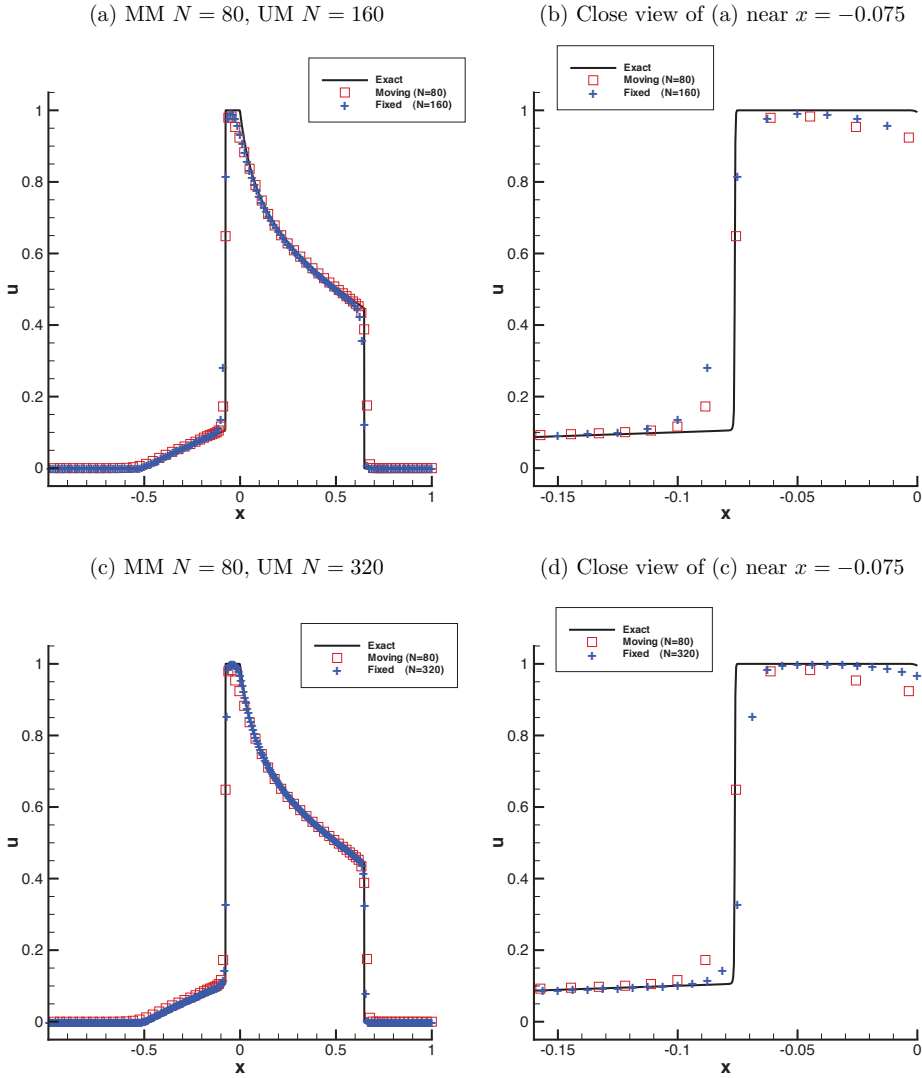

FIG. 10. *Example 4.3 (Buckley–Leverett equation). The moving mesh solution with $N = 80$ is compared with the uniform mesh solutions with $N = 160$ and $N = 320$.*

even better than the uniform mesh results. The same observation can be made from Figure 15, where the moving mesh solution with $N = 400$ is better than those with uniform meshes $N = 400$ and 600 and comparable to the uniform mesh solution with $N = 800$.

Next, we compare the accuracy of the moving mesh method with the uniform mesh method. The L_1 errors of density are listed in Table 2. We can see that the L_1 error of the moving mesh solution with $N = 200$ is clearly smaller than that with uniform meshes $N = 200$ and 300. Similarly, the L_1 error with the moving mesh $N = 400$ is smaller than those with uniform meshes $N = 400$ and 600 and comparable to the one with 800 uniform points. It is remarked that this Shu–Osher problem is a difficult test problem for moving mesh methods since a large portion of the physical domain is filled with features that require fine resolution. Nevertheless,
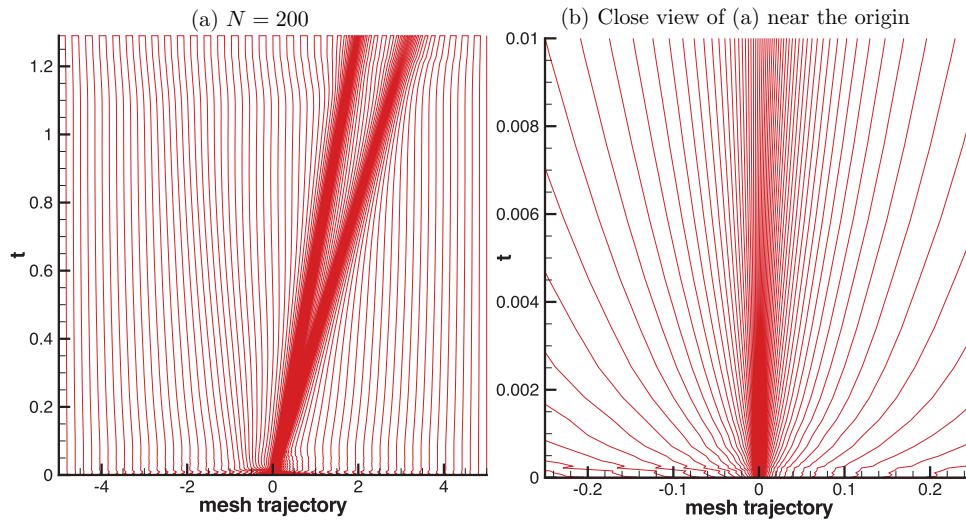
(a) $N = 200$ (b) Close view of (a) near the origin

Fig. 11. *Example 4.4 (Lax problem). The trajectories of a moving mesh with $N = 200$ are plotted.*

(a) MM $N = 200$, UM $N = 400$ (b) Close view of (a) near shock   (c) Close view of (a) near rarefaction wave

(d) MM $N = 200$, UM $N = 800$ (e) Close view of (d) near shock   (f) Close view of (d) near rarefaction wave
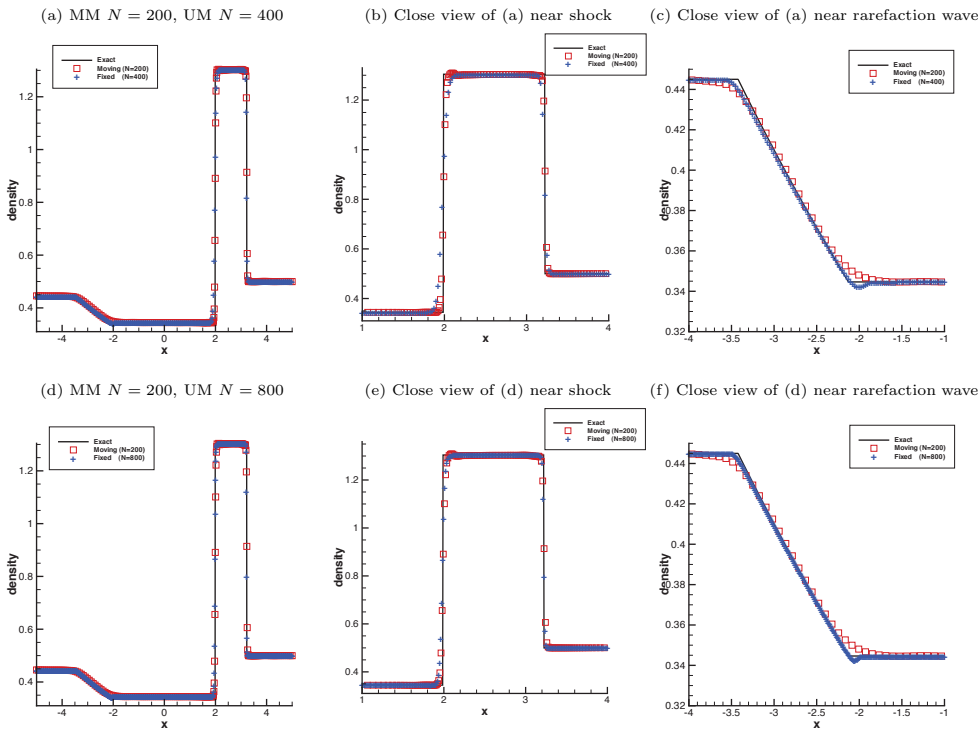
Fig. 12. *Example 4.4 (Lax problem). The moving mesh solution (density) with $N = 200$ is compared with the uniform mesh solutions with $N = 400$ and $N = 800$.*

TABLE 1
(Lax problem) L_1 errors of density for uniform and moving meshes.

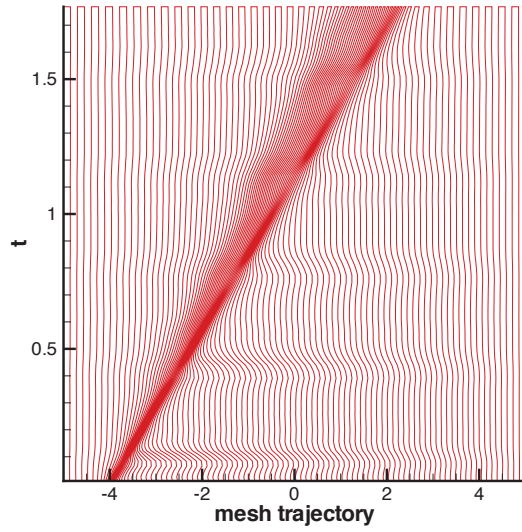| N	200	300	400	600	800
Uniform mesh	1.21E-02	8.64E-03	5.49E-03	4.18E-03	2.48E-03
Moving mesh	3.59E-03	2.30E-03	1.39E-03	1.12E-03	7.81E-04

FIG. 13. *Example 4.5 (Shu–Osher problem). The trajectories of a moving mesh with $N = 200$ are plotted.*

our moving mesh method still shows some savings: the ratio of the numbers of moving mesh points and uniform mesh points required to achieve a specific level of accuracy is about 1:1.5.

Example 4.6. The last example is the blast-wave problem modeling the interaction of two blast waves. It is governed by the Euler equations (3.4) subject to the initial condition

$$(\rho, v, p) = \begin{cases} w_L, & 0 < x < 0.1, \\ w_M, & 0.1 < x < 0.9, \\ w_R, & 0.9 < x < 1, \end{cases}$$

where

$$w_L = (1, 0, 10^3), \qquad w_M = (1, 0, 10^{-2}), \qquad w_R = (1, 0, 10^2).$$

The physical domain is taken as $(0, 1)$ and at both ends the reflective boundary condition is applied. The system is integrated up to $T = 0.038$. The computed density is plotted against the reference "exact" solution obtained with the fifth order finite difference WENO scheme [13] with 6400 uniform mesh points. The results are shown in Figures 16, 17, and 18. This problem was first used by Woodward and Colella [27] as a test problem for various numerical schemes.

The mesh trajectories in Figure 16 show that the two blast waves propagate in
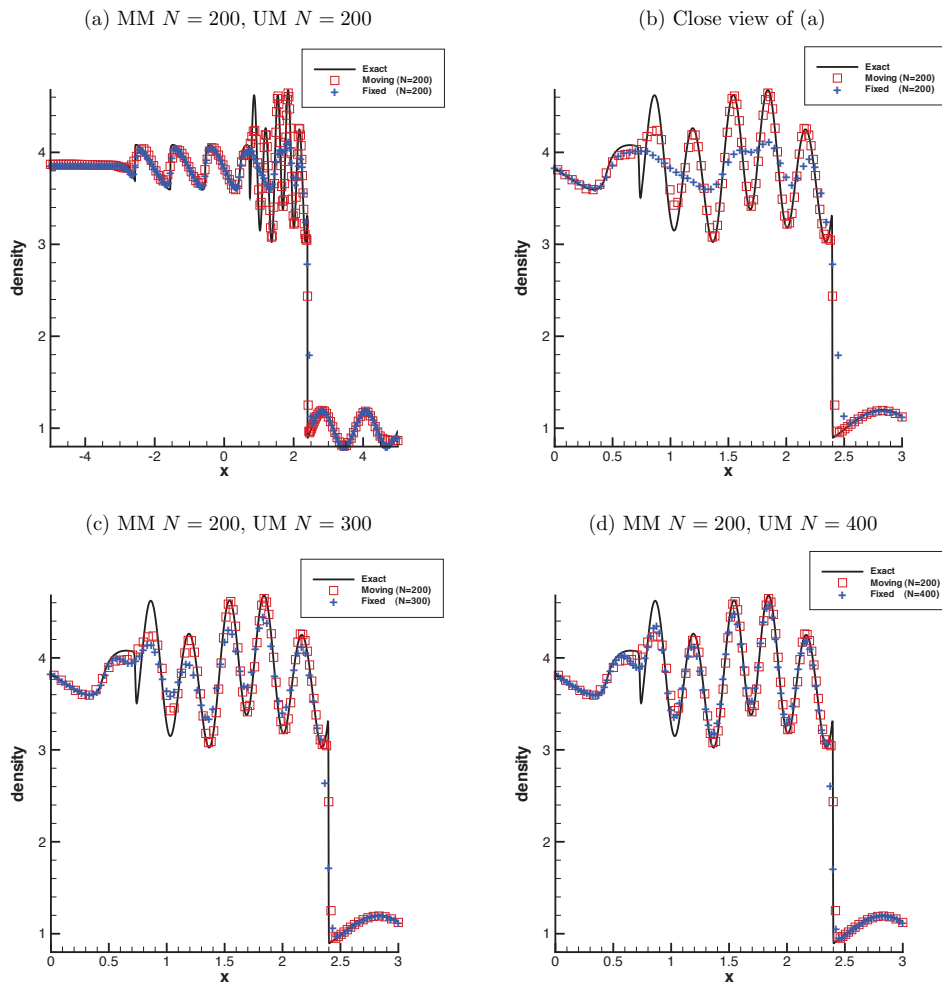
(a) MM $N = 200$, UM $N = 200$ (b) Close view of (a)

(c) MM $N = 200$, UM $N = 300$ (d) MM $N = 200$, UM $N = 400$

FIG. 14. *Example 4.5 (Shu–Osher problem). Density. The moving mesh solution with $N = 200$ is compared with the uniform mesh solutions with $N = 200$, $N = 300$, and $N = 400$.*

the right direction and finally collide. Two dense mesh trajectory lines form near $x = 0.1$ and $x = 0.9$ initially. For the time being, the two waves collide at $t = 0.027$.

Next, we will discuss our moving mesh results compared with uniform mesh results. From Figures 17 and 18 (particularly the close views in (b) and (c)), we can see that when the same number of mesh points $N = 200$ is used, the moving mesh results are better than the uniform mesh results. When the number of uniform mesh points increases, we can see that the moving mesh solution with $N = 200$ is comparable to that obtained with twice the number of uniform mesh points. The same can be observed with 400 moving mesh points (cf. Figures 18).

The L_1 errors of density for the moving and uniform mesh methods are listed in Table 3. We can see that the L_1 error of the moving mesh solution with $N = 200$ is smaller than the uniform mesh solution $N = 200$ and comparable to those with $N = 300$ and $N = 400$ of uniform mesh. The L_1 error of the moving mesh solution with $N = 300$ is smaller than that obtained with $N = 400$ and comparable to the uniform mesh solution of $N = 600$. For moving mesh with $N = 400$, we also can see
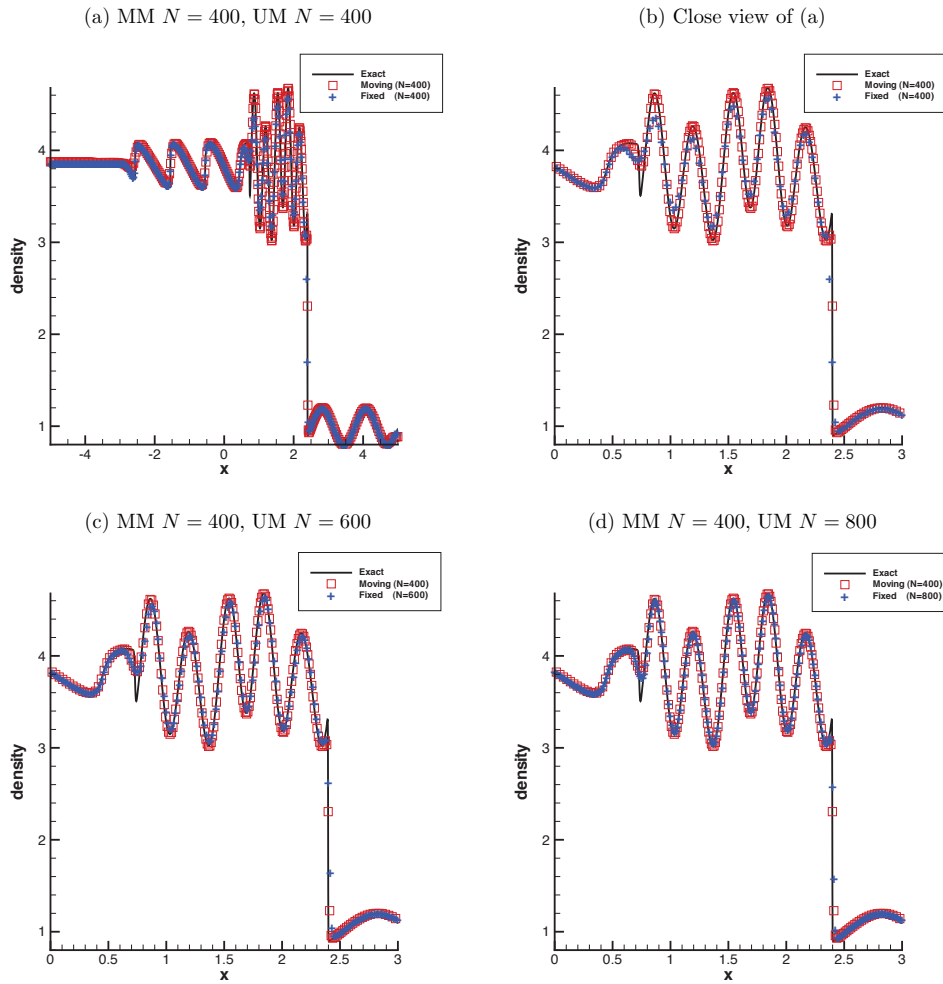
(a) MM $N = 400$, UM $N = 400$ (b) Close view of (a)

(c) MM $N = 400$, UM $N = 600$ (d) MM $N = 400$, UM $N = 800$

FIG. 15. *Example 4.5 (Shu–Osher problem). Density. The moving mesh solution with $N = 400$ is compared with the uniform mesh solutions with $N = 400$, $N = 600$, and $N = 800$.*

TABLE 2
(Shu–Osher problem) L_1 errors of density for uniform and moving meshes.

N	200	300	400	600	800
Uniform mesh	8.65E-02	4.50E-02	2.55E-02	2.51E-02	1.87E-02
Moving mesh	4.10E-02	2.21E-02	2.00E-02	1.38E-02	1.06E-02

that the L_1 error is smaller than that obtained by uniform mesh with $N = 600$ and comparable to that with $N = 800$.

5. Concluding remarks. In the previous sections we have developed and implemented a moving mesh WENO scheme for solving one-dimensional conservation laws on moving meshes. It is a nontrivial combination of the WENO scheme and an adaptive moving mesh strategy. Like many other high order schemes, the scheme requires the mesh to be sufficiently smooth to achieve the designed order of accuracy. To improve the smoothness of the mesh, we have employed the commonly used approach of smoothing the monitor function with a low-pass filter (cf. (2.21)) and a not
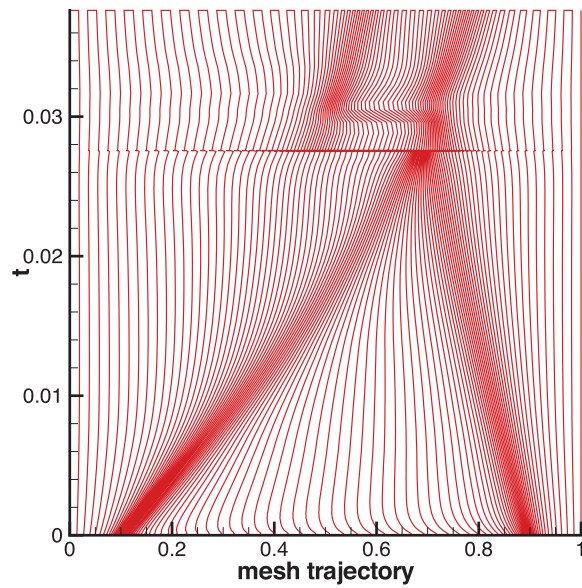
FIG. 16. *Example 4.6 (blast-wave problem). The trajectories of a moving mesh with $N = 200$ are plotted.*
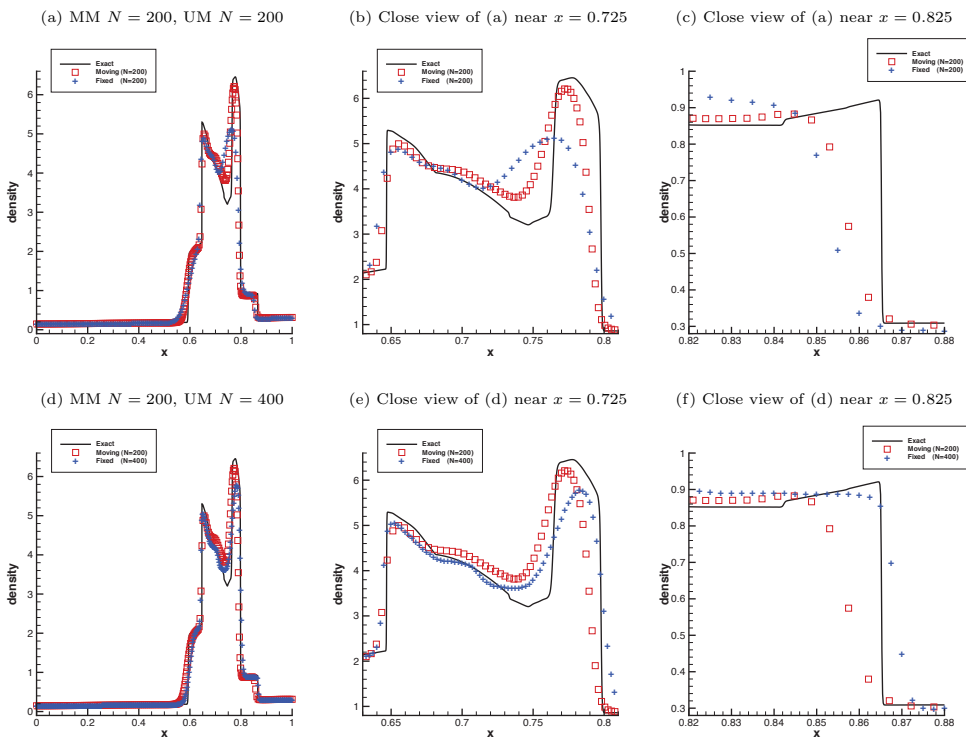

FIG. 17. *Example 4.6 (blast-wave problem). Density. The moving mesh solution with $N = 200$ is compared with the uniform mesh solutions with $N = 200$ and $N = 400$.*

(a) MM $N = 400$, UM $N = 400$ (b) Close view of (a) near $x = 0.725$ (c) Close view of (a) near $x = 0.825$

(d) MM $N = 400$, UM $N = 800$ (e) Close view of (d) near $x = 0.725$ (f) Close view of (d) near $x = 0.825$
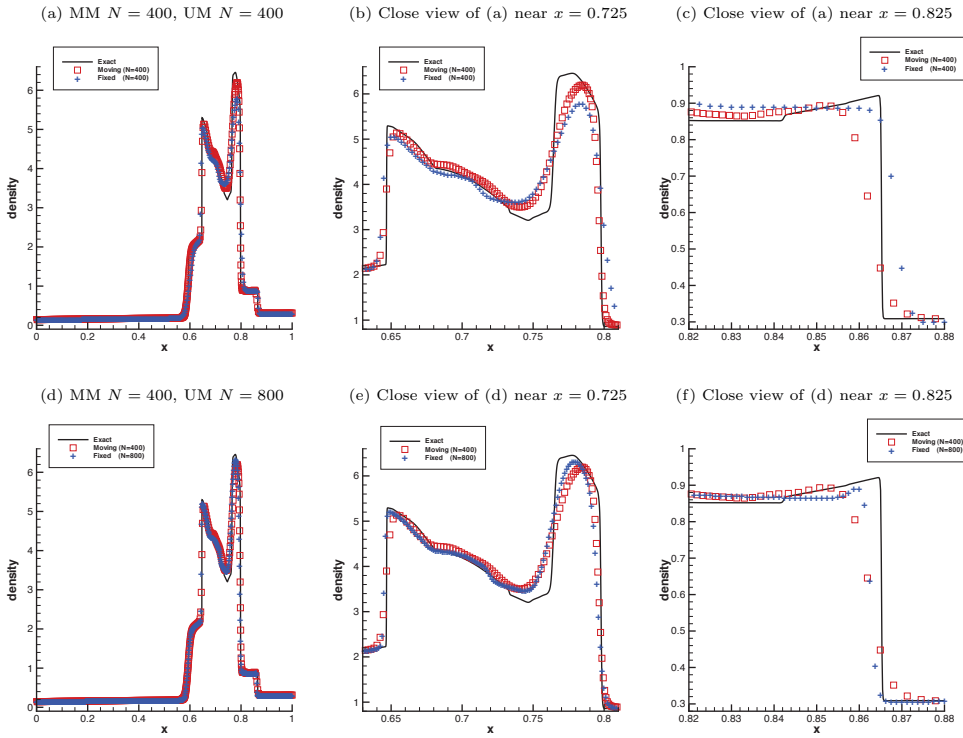

FIG. 18. *Example 4.6 (blast-wave problem). Density. The moving mesh solution with $N = 400$ is compared with the uniform mesh solutions with $N = 400$ and $N = 800$.*

TABLE 3
(Blast-wave problem) L_1 errors of density for uniform and moving meshes.

N	200	300	400	600	800
Uniform mesh	2.08E-01	1.33E-01	1.05E-01	8.47E-02	6.67E-02
Moving mesh	1.39E-01	9.59E-02	7.42E-02	4.82E-02	3.56E-02

so commonly used approach of directly smoothing the mesh with least squares fitting. Numerical results have demonstrated that the combination of these approaches is effective in improving the smoothness of the mesh.

It has also been shown that the transformation relation $J = x_\xi$ has to be approximated sufficiently accurately in order for the scheme to maintain the designed accuracy. This can be done using a high order finite difference approximation or the so-called GCL approach with which a geometric conservation law is used to update J at the new time step. In the mean time, it has been shown that the GCL satisfaction is not necessary nor sufficient for the scheme to maintain the designed accuracy and stability, but it does improve accuracy.

More importantly, we have shown that the mesh movement can cause the moving mesh WENO scheme to become unconditionally unstable. A simple strategy has been proposed to restrict the movement of the mesh. Numerical results have demonstrated that the strategy can recover the stability of the scheme while maintaining a reasonable level of mesh adaptivity.

The moving mesh WENO method has been applied to a number of examples with

discontinuities. For scalar conservation laws such as Example 4.1–4.3 and systems of conservation laws such as Example 4.4, the moving mesh method produces a comparable solution as the uniform mesh WENO scheme using about four times as many mesh points. For more difficult systems of conservation laws such as the Shu–Osher (Example 4.5) and blast-wave (Example 4.6) problems, the ratio of the numbers of moving mesh points and uniform mesh points required to achieve the same level of accuracy is approximately 1:1.5.

Finally, it is worth pointing out that the basic idea of the moving mesh WENO scheme can be applied to problems in two and three dimensions. Indeed, the generalization of the scheme to two-dimensional Euler equations and other physical systems is underway.

REFERENCES

[1] D. BOFFI AND L. GASTALDI, *Stability and geometric conservation laws for ALE formulations*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 4717–4739.
[2] W. CAO, W. HUANG, AND R. D. RUSSELL, *An r-adaptive finite element method based upon moving mesh PDEs*, J. Comput. Phys., 149 (1999), pp. 221–244.
[3] W. CAO, W. HUANG, AND R. D. RUSSELL, *A study of monitor functions for two-dimensional adaptive mesh generation*, SIAM J. Sci. Comput., 20 (1999), pp. 1978–1994.
[4] C. DE BOOR, *Good approximation by splines with variables knots*, II, in Conference on the Numerical Solution of Differential Equations (Univ. Dundee, Dundee, 1973), Lecture Notes in Math. 363, G. A. Watson, ed., Springer-Verlag, Berlin, 1974, pp. 12–20.
[5] L. FORMAGGIA AND F. NOBILE, *Stability analysis for second-order time accurate schemes for ALE-FEM*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 4097–4116.
[6] A. HARTEN, B. ENGQUIST, S. OSHER, AND S. CHAKRAVARTHY, *Uniformly high order accurate essentially non-oscillatory schemes*, J. Comput. Phys., 131 (1997), pp. 3–47.
[7] A. HARTEN AND S. OSHER, *Uniform high-order accurate nonoscillatory schemes*, SIAM J. Numer. Anal., 24 (1987), pp. 279–309.
[8] R. G. HINDMAN, *Generalized coordinate forms of governing fluid equations and associated geometrically induced errors*, AIAA J., 20 (1982), pp. 1359–1367.
[9] C. HU AND C.-W. SHU, *Weighted Essentially Non-oscillatory Schemes on Triangular Meshes*, NASA/CR-(1998)-208459, ICASE Report 98-32, 1998.
[10] W. HUANG, *Mathematical principles of anisotropic mesh adaptation*, Comm. Comput. Phys., 1 (2006), pp. 276–310.
[11] W. HUANG AND R. D. RUSSELL, *Adaptive Moving Mesh Methods*, Springer-Verlag, New York, 2011.
[12] W. HUANG AND W. SUN, *Variational mesh adaptation Part* II: *Error estimates and monitor functions*, J. Comput. Phys., 184 (2003), pp. 619–648.
[13] G. JIANG AND C.-W. SHU, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
[14] S. LI AND J. M. HYMAN, *Adaptive Mesh Refinement for Finite Difference WENO Scheme*, Technical Report LA-UR-03-8927, Los Alamos National Laboratory, 2003.
[15] X. D. LIU, S. OSHER, AND T. CHAN, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.
[16] S. OSHER AND S. CHAKRAVARTHY, *Upwind schemes and boundary conditions with applications to Euler equations in general geometries*, J. Comput. Phys., 50 (1983), pp. 447–481.
[17] J. QIU AND C.-W. SHU, *Finite difference WENO schemes with Lax–Wendroff-type time discretizations*, SIAM J. Sci. Comput., 24 (2003), pp. 2185–2198.
[18] C.-P. SHEN, J.-M. QIU, AND A. CHRISTLIEB, *Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations*, J. Comput. Phys., 230 (2011), pp. 3780–3802.
[19] C.-W. SHU, *Essential Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws*, NASA/CR-97-206253, ICASE Report 97-65, 1997.
[20] C.-W. SHU, *High order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD*, Internat. J. Comput. Fluids, 17 (2003), pp. 107–118.
[21] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock capturing schemes* II, J. Comput. Phys., 83 (1989), pp. 32–78.
[22] J. M. STOCKIE, J. A. MACKENZIE, AND R. D. RUSSELL *A moving mesh method for one-*

dimensional hyperbolic conservation laws, SIAM J. Sci. Comput., 22 (2001), pp. 1791–1813.

[23] H.-Z. Tang and T. Tang, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal., 41 (2003), pp. 487–515.

[24] H.-Z. Tang, T. Tang, and P. Zhang, *An adaptive mesh redistribution method for nonlinear Hamilton-Jacobi equations in two and three dimensions*, J. Comput. Phys., 188 (2003), pp. 543–572.

[25] P. D. Thomas and C. K. Lombard, *Geometric conservation law and its applications to flow computations on moving grids*, AIAA J., 17 (1979), pp. 1030–1037.

[26] L. Wan-Lung and Z. Tan, *Moving mesh methods for Boussinesq equation*, Internat. J. Numer. Methods Fluids, 61 (2009), pp. 1161–1178.

[27] P. R. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys., 54 (1984), pp. 115–173.

[28] D. Yoon, H. Kim, and W. Hwang, *Adaptive mesh refinement for weighted essentially non-oscillatory scheme*, Bull. Korean Math. Soc., 45 (2008), pp. 781–795.