THE NUMERICAL SOLUTION OF SHALLOW WATER EQUATION BY MOVING MESH METHODS

SUYEON SHIN* AND WOONJAE HWANG**

Abstract. This paper presents a moving mesh method for solving the hyperbolic conservation laws. Moving mesh method consists of two independent parts: PDE evolution and mesh-redistribution. We compute numerical solution of shallow water equation by using moving mesh methods. In comparison with computations on a fixed grid, the moving mesh method appears more accurate resolution of discontinuities.

1. Introduction

The moving mesh methods have been used for solving differential equations that involve singular solution. It gives an improvement in accuracy by moving mesh points that they are concentrate in regions of larger solution variations. Because of this advantage, many useful schemes have been applied on moving grids in recent years.

Several moving mesh techniques have been introduced, in which one of the most advocated methods is the one based on solving elliptic PDEs first proposed by Winslow [25]. Harten and Hyman [6] began the earliest study in this direction, by moving the grid along the characteristic direction to increase the accuracy of solution. After their work, many other moving mesh methods have been proposed. Salari and Steinberg [13] developed a FCT method on a moving grid based on adaptive grid generation algorithms. The moving mesh method based on harmonic
mapping was suggested by Dvinsky [4]. His method can be viewed as a generalization and extension of Winslow’s method. Motivated by the work of Dvinsky, a moving mesh finite element strategy based on harmonic mapping was proposed and studied by the Li et al. [9, 10]. The moving mesh PDEs were studied by Russell et al. [7, 12, 16], and Li and Petzold [11]. In recent years, there are many works about moving mesh methods [1, 2, 3, 14, 15, 19, 20, 21, 22, 23, 24]. Adaptive mesh method consists of two independent parts [9, 19]: a mesh-redistribution algorithm and a solution algorithm. The first part is an iteration procedure. Meshes are redistributed by an equidistribution principle. For the resulting elliptic, adaptive mesh PDEs, a Gauss-Seidel type iteration method is used. The underlying numerical solution on the new grids are updated by a conservative-interpolation formula. The second part will be independent of the first one, and it can be any of the standard codes for solving the given PDEs.

The organization of this paper is as follows. In Section 2, the moving mesh method is briefly described. Numerical experiments of the one-dimensional hyperbolic conservation laws are provided in Section 3.

2. Moving mesh method

In Tang and Tang [19, 24], the basic idea of the moving mesh method can be summarized by two independent parts: mesh-redistribution and PDE evolution.

2.1. Mesh-redistribution based on Gauss-Jacobi iteration

Meshes are redistributed by an equidistribution principle. The mesh-redistribution equation is

\[(wx_\xi)_\xi = 0, 0 < \xi < 1\]

where the function \(w\) is called monitor function [16, 17], which designed specifically to resolve discontinuous solutions. To solve the mesh redistribution equation (2.1), the following Gauss-Jacobi type iteration can be used:

\[w(u_{j+\frac{1}{2}})(x_{j+1} - \tilde{x}_j) - w(u_{j-\frac{1}{2}})(\tilde{x}_j - x_{j-1}) = 0,\]

and then the underlying numerical solution on the new grids \(\{\tilde{x}_{j+\frac{1}{2}}\}\) are updated by a conservative-interpolation formula. The conservative-interpolation formula is following:

\[\Delta \tilde{x}_{j+\frac{1}{2}} \tilde{u}_{j+\frac{1}{2}} = \Delta x_{j+\frac{1}{2}} u_{j+\frac{1}{2}} - ((cu)_{j+1} - (cu)_j),\]
The numerical solution by moving mesh methods 565

where \( \Delta \tilde{x}_{j+\frac{1}{2}} = \tilde{x}_{j+1} - \tilde{x}_j \). The linear flux \( cu \) in (2.3) is approximated by second-order numerical flux. The second-order numerical flux is defined by

\[
(cu)_j = \frac{c_j}{2}(u_j^+ + u_j^-) - \frac{|c_j|}{2}(u_j^+ - u_j^-).
\]

The wave speed \( c_j \) above is defined by \( c_j = x_j - \tilde{x}_j \). In (2.4), \( u_j^{n,\pm} \) are defined by

\[
u_n^{\pm} = u_{j+\frac{1}{2}}^{n} \pm \frac{1}{2}(x_{j+1} - x_{j})\tilde{S}_{j+\frac{1}{2}},
\]

where \( \tilde{S}_{j+\frac{1}{2}} \) is an approximation of the slope \( u_x \) at \( x_{j+\frac{1}{2}} \). \( \tilde{S}_{j+\frac{1}{2}} \) is defined by

\[
\tilde{S}_{j+\frac{1}{2}} = (\text{sign}(\tilde{S}_{j+\frac{1}{2}}^+) + \text{sign}(\tilde{S}_{j+\frac{1}{2}}^-)) \frac{|\tilde{S}_{j+\frac{1}{2}}^+ - \tilde{S}_{j+\frac{1}{2}}^-|}{|\tilde{S}_{j+\frac{1}{2}}^+| + |\tilde{S}_{j+\frac{1}{2}}^-|},
\]

with

\[
\tilde{S}_{j+\frac{1}{2}}^+ = \frac{u^n_{j+\frac{3}{2}} - u^n_{j+\frac{1}{2}}}{x_{j+\frac{3}{2}} - x_{j+\frac{1}{2}}}, \quad \tilde{S}_{j+\frac{1}{2}}^- = \frac{u^n_{j+\frac{1}{2}} - u^n_{j-\frac{1}{2}}}{x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}}.
\]

The equation (2.5)~(2.7) are the MUSCL (monotone upstream-centered scheme for conservation laws)-typed finite volume method. The new grid procedure by the Gauss-Jacobi type iteration and the updated solution procedure using the conservative-interpolation are repeated for a fixed number of iterations or until \( \|x^{[n+1]} - x^{[n]}\| \leq \epsilon \).

2.2. PDE evolution

The mesh redistribution is based on an iteration procedure and the PDE evolution is independent of the first part. It can be solved by using any high resolution finite volume methods. In this paper, a second-order finite volume scheme with a numerical flux is the Lax-Friedrichs flux:

\[
\hat{f}(a, b) = \frac{1}{2}[f(a) + f(b) - \max_u\{|f_u|}(b - a)].
\]

The details can be found in Tang and Tang [19].
3. Numerical experiments

In this section, we consider the one-dimensional hyperbolic conservation laws

\[ u_t + f(u)_x = 0, \quad t > 0. \]  

As a scalar model, we consider Buckley-Leverett equation. For a system model, we consider the shallow water equation [5, 18]. We consider the Riemann problem for both equations. Some details are the following: the number of Jacobi iterations is 5; the scheme for evolving this equation is a (formally) second-order MUSCL finite volume scheme (with the Lax-Friedrichs flux) together with a second-order Runge-Kutta discretization; the CFL number used is 0.3.

**Example 3.1.** We consider Buckley-Leverett problem [8],

\[ u_t + f(u)_x = 0, \quad f(u) = \frac{u^2}{u^2 + a(1-u)^2}, \quad a = 0.5. \]

The initial data are

\[ u(x, 0) = \begin{cases} 
1 & \text{if } x < 0 \\
0 & \text{if } x > 0.
\end{cases} \]

This may be used to model gas and oil in a reservoir, where \( u \) is the fluid saturation (water). This flux is nonconvex with a single inflection point. The characteristic speed is \( f'(u) = \frac{2au(1-u)}{(u^2 + a(1-u)^2)^2} \). By following characteristics, we can construct the triple-valued solution. By the

![Figure 1](image)

**Figure 1.** (a) Numerical solutions by MUSCL, (b) Numerical solutions by MUSCL with MMM (J=50).
equal-area rule, this triple-valued solution replaced a shock. Thus Rie-
mann solution involves both a shock and a rarefaction wave and is called
a compound wave.

Figure 2. (a),(b), and (c) show numerical solutions by
MUSCL with MMM of monitor parameter 1, 10, and 20,
respectively. (a’),(b’), and (c’) show mesh trajectories of
monitor parameter 1, 10, and 20, respectively. (J=50)
Figure 1 (a) and (b) show the numerical solutions at $t = 0.5$ for uniform grid and nonuniform grid, respectively obtained with the number of grids $J = 50$. The monitor function used in the computation is

Figure 3. Height: (a),(b), and (c) show numerical solutions by MUSCL of $J = 50$, 75, and 100, respectively. (a’),(b’), and (c’) show numerical solutions by MUSCL with MMM of $J = 50$, 75, and 100, respectively.
$w = \sqrt{1 + 10 |u_\xi|^2}$. In this example, the monitor function is taken as $w = \sqrt{1 + \beta |u_\xi|^2}$, $\beta > 0$ where several values of $\beta$ are used. In Figure 2, the numerical solution and mesh trajectories with three different

![Graphs showing numerical solutions](image)

**Figure 4.** Velocity: (a),(b), and (c) show numerical solutions by MUSCL of $J = 50$, 75, and 100, respectively. (a’),(b’), and (c’) show numerical solutions by MUSCL with MMM of $J = 50$, 75, and 100, respectively.
monitor functions ($\beta = 1, 10, 20$) are plotted. It is observed that the grid distributions seem more reasonable, with more points where the value of curvature is large. However, we avoid clustering too many points in the neighborhood of the discontinuities. Thus parameter $\beta = 10$ of monitor function is efficient choice. Obviously, the study on how to choose the monitor constants seems very useful. In principle, the monitor function $w$ can be any appropriately chosen measure of the numerical error in the solution of the PDE. At points where the error is large, $w$ should also be large so that mesh points will tend to concentrate in those areas where higher resolution is needed.

![Figure 5](image_url)

**Figure 5.** (a), (b), and (c) show mesh trajectories for $J = 50$, 75, and 100, respectively.
Example 3.2. We apply moving mesh algorithm to the one-dimensional shallow water equations,

\[
\left( \begin{array}{c} h \\ hu \end{array} \right)_t + \left( \begin{array}{c} hu \\ hu^2 + \frac{1}{2}gh^2 \end{array} \right)_x = 0
\]

Figure 6. Height: (a), (b), and (c) show numerical solutions by MUSCL of \( J = 20, 40, \) and 80, respectively. (a’), (b’), and (c’) show numerical solutions by MUSCL with MMM of \( J = 20, 40, \) and 80, respectively.
where \( h, u \) and \( g \) are height, velocity and gravitational constant, respectively. Consider the shallow water equation with the piecewise-constant

\[
\begin{align*}
\frac{\partial u}{\partial t} + \frac{\partial (hu)}{\partial x} &= 0, \\
\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} &= -gh. 
\end{align*}
\]

Figure 7. Velocity: (a), (b), and (c) show numerical solutions by MUSCL of \( J = 20, 40, \) and 80, respectively. (a’), (b’), and (c’) show numerical solutions by MUSCL with MMM of \( J = 20, 40, \) and 80, respectively.
initial data

\[ h(x,0) = \begin{cases} 
3 & \text{if } x < 0 \\
1 & \text{if } x > 0, 
\end{cases} \]

\[ u(x,0) = 0. \]

This is special case of the Riemann problem in which \( u_l = u_r = 0 \), and is called the dam-break problem because it models what happens if a dam separating two levels of water bursts at time \( t = 0 \).

Figures 3 and 4 show the solutions of height and velocity, respectively at \( t = 0.5 \) for uniform and nonuniform grid, obtained with \( J = 50 \),

**Figure 8.** (a), (b), and (c) show mesh trajectories for \( J = 50, 75, \) and 100, respectively.
Figure 9. Height: (a), (b), and (c) show numerical solutions by MUSCL with MMM of monitor parameter 1, 10, and 100 respectively. (a’), (b’), and (c’) show mesh trajectories of monitor parameter 1, 10, and 100, respectively. (J=40)
$J = 75$, and $J = 100$. Figure 5 shows the trajectories of the grid points at $t=0.5$, obtained with $J=50$, $J=75$, and $J=100$. In this example, the monitor function used in the computation is $w = \sqrt{1 + 100 \left( \frac{[h_\xi]}{\max_{\xi} |h_\xi|} \right)^2}$.

**Example 3.3.** Consider the shallow water equation with the piecewise-constant initial data

$$h(x, 0) = 1,$$

$$u(x, 0) = \begin{cases} 
1 & \text{if } x < 0 \\
-1 & \text{if } x > 0.
\end{cases}$$

The solution is symmetric in $x$ with $h(-x, t) = h(x, t)$ and $u(-x, t) = -u(x, t)$ at all times. A shock waves moves in direction, bringing the fluid to rest, since the middle state must have $u_m = 0$ by symmetry.

Figures 6 and 7 show the solutions of height and velocity, respectively at $t = 1$ for uniform and nonuniform grid, obtained with $J = 20$, $J = 40$, and $J = 80$. Figure 8 shows the trajectories of the grid points at $t=1$, obtained with $J=20$, $J=40$, and $J=80$. In this example, the monitor function used in the computation is $w = \sqrt{1 + 10 \left( \frac{|u_\xi|}{\max_{\xi} |u_\xi|} \right)^2}$. In this example, the monitor function is taken as $w = \sqrt{1 + \beta \left( \frac{|u_\xi|}{\max_{\xi} |u_\xi|} \right)^2}, \beta > 0$ where several values of $\beta$ are used. In Figure 9, the mesh trajectories and numerical solution with three different monitor functions ($\beta = 1, 10, 100$) are plotted.

**References**


* Department of Mathematics
Korea University
Seoul 136-701, Republic of Korea
E-mail: angelic52@korea.ac.kr

** Department of Information and Mathematics
Korea University
Jochiwon 339-700, Republic of Korea
E-mail: woonjae@korea.ac.kr