Finite Element Method of BBM-Burgers Equation with Dissipative Term Based on Adaptive Moving Mesh

Changna Lu,1 Qianqian Gao,1 Chen Fu,1 and Hongwei Yang2,3

1School of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing 210044, China
2College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China
3Key Laboratory of Meteorological Disaster, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China

Correspondence should be addressed to Hongwei Yang; hwyang1979@163.com

Received 17 July 2017; Revised 24 October 2017; Accepted 6 November 2017; Published 29 November 2017

Research Article

1. Introduction

In mathematics, physics, and engineering, nonlinear partial differential equations play an important role due to their rich mathematical structures and features [1–4] as well as important applications in fluid dynamics, plasma physics, fiber optics, condensed matter physics, and chemistry [5–11]. It is important to search for solutions. These nonlinear partial differential equations can be solved exactly or numerically to understand the properties of solutions.

In this paper, we consider the nonlinear multidimensional Benjamin-Bona-Mahony-Burgers (BBM-Burgers) equation with a dissipative term and use the finite element method based on the moving mesh to solve its numerical solution. The equation has the following form:

\[ u_t + \sum_{i=1}^{d} \alpha_i u_x + \sum_{i=1}^{d} \beta_i uu_x - \gamma \Delta u - \mu \Delta u_t + \sum_{j=1}^{p} \delta_{ij} \frac{\partial^{2m+2} u}{\partial x_i^{2m+2}} = F, \quad X \in \Omega, \quad t \in (0, T) \]

where \( X = (x_1, x_2, \ldots, x_d) \), \( d \) stands for the dimension, \( x_i \) stands for the corresponding spatial vector, the coefficients \( \alpha_i, \beta_i \in \mathbb{R}, \gamma, \mu, \Delta_n > 0, \Omega \subseteq \mathbb{R}^d \), \( \Delta \) is a Laplace operator, and \( p \) is one constant to define the order of the dissipative term. In the equation, \( F \) is a sufficiently smooth function and stands for the source term, and \( g(X) \) and \( u_0(X) \) are given functions.

Equation (1) is related to the well-known Benjamin-Bona-Mahony (BBM) equation, which was advocated by Benjamin et al. [12] in 1972 as a refinement of the Korteweg-de-Vries (KdV) equation. There are several different transformations of this equation; when the coefficients \( \mu = 0, F = 0 \), the equation becomes \( u_t + \sum_{i=1}^{d} \alpha_i u_x + \sum_{i=1}^{d} \beta_i uu_x = 0 \), which is the Burgers equation with a nonlinear term. When parameters \( \gamma, \delta_n = 0 \), (1) becomes \( u_t + \sum_{i=1}^{d} \alpha_i u_x + \sum_{i=1}^{d} \beta_i uu_x - \mu \Delta u_t = F \), which is called Regularized Long Wave (RLW)
equation, also called BBM equation. The RLW equation is usually used to analyze small amplitude long waves on the surface of water in fluid dynamics. When $\delta_n = 0$, (1) becomes a generalized BBM-Burgers equation, which is a mathematical model of propagation of small-amplitude long wave in nonlinear dispersive media and better than Korteweg-de-Vries-Burgers (KdV-B) equation in fluid mechanics; when $F = 0$, $d = 1$, and $g(x) = 0$, the BBM-Burgers equation satisfies two conservation laws:

\[
\begin{align*}
\text{Mass function } E_1(t) &= \int_{\Omega} u \, dx, \\
\text{Energy function } E_2(t) &= \int_{\Omega} \left( \frac{1}{2} u^2 + \frac{\mu}{2} (u_x)^2 \right) dx + \alpha \int_0^T \int_{\Omega} u_x^2 (\zeta) \, dx \, d\zeta.
\end{align*}
\]

If $d = 2$, $p = 1$, (1) becomes

\[
\begin{align*}
&u_t + \alpha_1 u_x + \alpha_2 u_y + \beta_1 u_{xx} + \beta_2 u_{yy} - \gamma \Delta u - \mu \Delta u_t \\
+ \delta (u_{xxxx} + u_{yyyy}) = F, \quad (x, y) \in \Omega, \quad t \in (0, T] \\
u(x, y, t) = g(x, y, t), \quad (x, y) \in \partial \Omega, \quad t \in (0, T] \\
u(x, y, 0) = u_0(x, y), \quad (x, y) \in \Omega.
\end{align*}
\]

In physical phenomena, the effect of dispersion in the BBM-Burgers equation with a high-order dissipative term is similar to the RLW equation; the effect of dissipation is similar to the Burgers equation.

To get the approximate solution of the BBM-Burgers equation with a high-order dissipative term, some numerical and analytical methods have been proposed in recent years. In the aspect of mathematical theory, Zhao et al. consider the existence of the generalized BBM-Burgers equation with a dissipative term and verify the corresponding convergence in H-measures [13, 14]. Kondo and Webler apply kinetic decomposition for the generalized BBM-Burgers equations with a dissipative term to obtain the approximate transport equation, and then take advantage of the averaging lemma to get the convergence [15]. Seok states the existence and conservation laws of the generalized BBM-Burgers equation with a dissipative term [16]. For more theoretical analyses, refer to the references in [13–16]. In the aspect of numerical solutions, many methods have been used, like Painlevé test, Darboux Transformation, bilinear method, symmetry method [17], and so on [18–21]. Hong et al. apply the homotopy perturbation method and He’s variational iteration method to solve the BBM-Burgers equation. Ganji et al. put forward the notion that the method they used in [22] is better than the homotopy analysis method proposed in [23]. Finite element Galerkin methods have been discussed by Kadri et al. [24], Lee [25], Yin and Piao [26], and Dehghan et al. [27] to solve the BBM-Burgers equations with a dissipative term. Finite difference methods have also been widely used to solve the equations in [28–30]; as for other numerical methods, refer to [31–34].

The references introducing the numerical simulations above are mostly based on fixed meshes; in this paper, we consider the finite element method based on the adaptive moving mesh to solve the BBM-Burgers equation with a high-order dissipative term. The moving mesh method is a kind of adaptive mesh method which adds elements near the place where the numerical solution changes rapidly and decreases elements in the solution changing slowly; the total element number still remains unchanged. When we compare the finite element method based on adaptive moving mesh with methods based on fixed mesh or proposed in [26], we find that the moving mesh method takes fewer elements to get the same distinguishability and simulates steep waves and the transition of gap distinctly. In recent years, the moving mesh method has witnessed further development and extensive application [35]. This method was applied to approximate the anisotropic diffusion-convection-reaction problems to satisfy a discrete maximum principle, and it was adopted to study the porous medium equation and approximate the solution [36]. For more information about the moving mesh method, please refer to [37–41].

The remainder of this paper is divided into three sections. In Section 2, we introduce the method we used, the discretization of the BBM-Burgers equation with a fourth-order dissipative term in one dimension, and the implementation of the moving mesh method. Section 3 presents 1D and 2D numerical examples of BBM-Burgers equations with a dissipative term. Finally, Section 4 shows the conclusion we draw from the numerical examples.

### 2. Finite Element Method Based on Moving Mesh

In this section, we mainly introduce the linear finite element method based on moving mesh and make use of the method to solve the BBM-Burgers equation with a high-order dissipative term. For convenience, in Section 2.2, we introduce the finite element method based on adaptive moving mesh. Section 3 presents 1D and 2D numerical examples of BBM-Burgers equations with a dissipative term. Finally, Section 4 shows the conclusion we draw from the numerical examples.

#### 2.1. Adaptive Moving Mesh Method

In this subsection, we mainly want to show the generation of physical mesh $\mathcal{T}_h^{n+1}$ at a new time level by an adaptive moving mesh strategy based on the Moving Mesh Partial Differential Equation (MMPDE) [42]; at the current time level, the physical mesh $\mathcal{T}_h^n = \{X_k^n\}$ and the numerical solution $(u^h)^n$ are known (the mesh and value will be introduced clearly in the following parts). During the strategy, one of the crucially important rules is to view the nonuniform mesh (i.e., the physical mesh in this paper) as a uniform mesh under the metric tensor $M$; the tensor $M$ is a symmetric and positive definite matrix. Under the metric tensor $M$, the physical mesh $\mathcal{T}$ and the computational mesh $\mathcal{T}_c$ can be transformed into each other if and only if it satisfies (the following notations and concepts about the mesh are the same as those in [36])

\[
|K| \det \left( \frac{M_K}{|\Omega|} \right)^{1/2} = \frac{|K| \sigma_h}{|\Omega|}, \quad \forall K \in \mathcal{T}_h
\]
Consider the equation to be just one-dimensional, (6) will have dimensional, parameters and $I$ are usually defined as dimensionless parameters. When we consider the equation to be just one-dimensional, (6) will have a relationship with (4) and $\theta = 0$; when the equation is two-dimensional, parameters $\theta$, $P$ are usually defined as $1/3$ and 2, and $d$ stands for the dimension in space. The energy function $I_h$ is also a function about the physical mesh coordinate $x_j = (x_j, y_j)$ and the computational mesh coordinate $\xi_j = (\xi_j, \eta_j)$ ($j = 0, \ldots, J$), where $J + 1$ stands for the number of mesh vertices. The coordinate $\xi = \xi(X, t)$ is an inverse transformation of $X = X(\xi, t)$; when $j = 0$ or $j = J$, the coordinate $\xi_j$ is the same as $X_j$. The energy function is too long to write; we write it in a simple form:

$$I_h = \sum_K |K| G \left( (F_K^{-1}) \circ (\mathbb{M}_K) \circ (F_K^{-T}) \right) \delta_{\theta}.$$  

Assume that the current physical mesh $\mathcal{T}_h^n$ and $(u_h^n)$ are known; by minimizing the energy function, we can get the corresponding computational mesh of the next time level, which can be expressed by

$$\mathcal{T}_c^{n+1} = \arg \min_{\mathcal{T}_c^n} I_h \left( \mathcal{T}_h^n; \mathcal{T}_c^n \right).$$  

In order to solve the minimum of energy function in (6), we need to take the derivative of the coordinate with respect to the energy function; that is,

$$\frac{\partial I_h}{\partial x_j} = -\frac{\partial I_h}{\partial \xi_j}, \quad j = 0, \ldots, J.$$  

It is not difficult to find that the equations are nonlinear and difficult to solve directly; in order to solve this problem, many scholars put forward a large number of methods to improve it. In this paper, we use the MMPDE [42] method to get the minimum value of the energy function (6). For convenience and efficiency, $\xi$-formulation MMPDE is made use of rather than $x$-formulation MMPDE; during the calculation, $\xi$-formulation only involves the differentiation of $G$ with respect to its first two arguments in the energy function, and there is no need for the metric tensor $\mathbb{M}$ to update. The minimization process can be replaced as the derivative of the computational mesh coordinate with respect to $t$, which is an ordinary differential equation and has the following form:

$$\frac{d\xi_j}{dt} = \frac{P_j}{\tau} \left[ \frac{\partial I_h}{\partial \xi_j} \right]^T,$$

where the derivative $\partial I_h/\partial \xi_j$ is seen as a row vector, $\tau > 0$ is a relaxation factor which is used for adjusting the time scale on moving mesh, and $P = \{P_0, \ldots, P_J\}$ is a positive function used to make (10) have some desired properties above; in this paper, $\tau = 10^{-4}, P_j = -\det(\mathbb{M}(X_j))^{1/2}$. At the current physical mesh $\mathcal{T}_h^n$ $(n = 0, \ldots, K)$ and computational solution $(u_h^n)$, we define a reference computational mesh $\mathcal{T}_c = \{\xi_0, \ldots, \xi_J\}$, which has uniform elements; the mesh $\mathcal{T}_c$ also has the same relationship, grid connection, and number of mesh vertices as the physical mesh $\mathcal{T}_h$ and computational mesh $\mathcal{T}_c$. The three kinds of meshes just differ in the location of the mesh coordinate. During the mesh generation, by integrating (10) from $t_n$ to $t_{n+1}$, we get the new computational mesh $\mathcal{T}_c^{n+1} = \{\xi_0^{n+1}, \ldots, \xi_J^{n+1}\}$; there is no need for the metric tensor $\mathbb{M}$ to update. The relationship between the physical mesh and the computational mesh can be expressed by an affine mapping $\mathcal{T}_c^{n+1} = \Phi_h(\mathcal{T}_c^{n-1})$; the physical mesh in the next time level can take the reference computational mesh by using a form of linear interpolation to get the approximate result:

$$\mathcal{T}_h^{n+1} = \Phi_h(\mathcal{T}_c^{n+1}).$$

2.2. Linear Finite Element Discretization of the 1D BBM-Burgers Equation with a Dissipative Term. In this subsection, we mainly introduce the linear finite element discretization of the 1D BBM-Burgers equation with a dissipative term. When $(u_h^n, x^n)$, and the discretization of the equation are known, we will get the value $(u_h^{n+1}, x^{n+1})$ at the new time level. The 1D BBM-Burgers equation with a fourth-order dissipative term has the formulation

$$u_t + uu_x + \beta uu_x - \gamma u_{xx} - \mu u_{xxx} + \delta u_{xxxx} = F,$$

$$x \in \Omega, \quad t \in (0, T].$$
Before solving (11) by the finite element method based on moving mesh, we have to introduce a new variable \( w = u_{\text{xx}} \) rather than solving the equation directly; because a mixed derivative of the finite element approximation based on the moving mesh method is unable to be defined, then (11) becomes a coupled system. Before the discretization of the 2D equation with a fourth-order dissipative term, we need to introduce two new different variables to replace \( u_{\text{xx}} \) and \( u_{\text{yy}} \); the equation then becomes three coupled equations; the other steps in the 2D case are similar to the 1D case. The new formulation of (11) is as follows:

\[
\begin{align*}
\frac{d}{dt}u_i - \mu u_i + \alpha u_x + \beta uu_x - \gamma w + \delta u_{\text{xx}} &= F_i, \\
x &\in \Omega, \ t \in (0, T], \\
w &= u_{\text{xx}}, & (12)
\end{align*}
\]

The basis function \( \varphi \) is just related to \( x \); on a moving mesh, it is also related to \( x \) and \( t \). The basis function spans the space \( V_0^h = \{ \varphi_0(x,t), \ldots, \varphi_{J-1}(x,t) \} \), \( \forall \varphi \in V_0^h \). We need to find whether \( u^h(x,t) \in V_0^h \) and \( u^h(x,t) \in V_0^h \) with the Dirichlet boundary condition equal to zero can satisfy the following system:

\[
\begin{align*}
\int_\Omega \left( \frac{d}{dt}u_i - \mu u_i + \alpha u_x + \beta uu_x - \gamma w - F_i \right) \varphi dx &- \int_\Omega \delta w_i \varphi_i dx = 0, \quad \forall \varphi \in V_0^h (\Omega), \ t \in (0, T], \\
\int_\Omega \left( w \varphi + \mu u_x \varphi_x \right) dx &= 0, \quad \forall \varphi \in V_0^h (\Omega), \ t \in (0, T].
\end{align*}
\]

Take \( u_i = \sum_{j=1}^{J-1} u_j(t) \varphi_j(x,t), \quad w_i = \sum_{j=1}^{J-1} w_j(t) \varphi_j(x,t), \) where the derivative of the basis function \( \varphi_j \) with respect to \( t \) is \( \frac{\partial \varphi_j}{\partial t} = -\frac{\partial \varphi_j}{\partial x} \dot{X}(x,t) \), and \( \dot{X} \) has the following formulation: \( \dot{X}(x,t) = \sum_{j=1}^{J-1} \dot{x}_j \varphi_j(x,t) \) (in the two-dimensional equation, \( \dot{X}(x,y,t) = \sum_{j=1}^{J-1} \dot{x}_j \varphi_j(x,y,t) \), so \( (u^h) \) can be expressed as

\[
\begin{align*}
(u^h) = \sum_{j=1}^{J-1} \frac{d}{dt} \varphi_j - \sum_{j=1}^{J-1} \frac{\partial \varphi_j}{\partial x} \dot{X}
\end{align*}
\]

Take the basis function \( \varphi = \varphi_k \) \( (k = 1, 2, \ldots, J - 1) \) and substitute (14) into (13); we can get

\[
\begin{align*}
\int_\Omega \left( \frac{d}{dt}u_i - \mu u_i + \alpha u_x + \beta uu_x - \gamma w_i \right) \varphi_k dx &- \int_\Omega \delta w_i \varphi_k dx = 0, \\
\int_\Omega \left( w_i \varphi_k + \mu u_x \varphi_{x_k} \right) dx &= 0.
\end{align*}
\]

We rewrite (15) into a matrix form:

\[
M \frac{d}{dt} (u - \mu w) + f - \delta Aw = 0,
\]

\[
Mw + Aw = 0.
\]

Only if the ODE (17) is solved can the solution \( u \) of the BBM-Burgers equation with a dissipative term be expressed, where \( u = -A^{-1}Mw \).

All the process of discretization is based on the homogeneous boundary condition \( u_0 = g \); when the boundary condition is nonhomogeneous, we just need to subtract the integration at the boundary.

2.3. The Implementation of the Method Combined with Finite Element and Moving Mesh. In this subsection, we mainly introduce the implementation of the two-dimensional method. The ways to solve the equation usually can be divided into two kinds: a simultaneous solution procedure and an alternate solution procedure. The simultaneous solution procedure combines the finite element discretization with the mesh equation (10) into a large system and solves them together; the alternate solution procedure solves one by one. In this paper, we introduce a formulation of the second method which is called MP (the method solves the moving equation firstly and solves the physical equation later), which is more flexible and efficient. The method contains four important parts: initialize the mesh, move the mesh, discretize the equation, and integrate time from \( t_0 \) to \( t_K \).
Step 1 (initialize the mesh). In the whole computational domain, we first divide the domain into \( N \) uniform elements; the corresponding initial physical mesh is \( \mathcal{F}_h^0 \), and the initial coordinate, initial value of the BBM-Burgers equation with a dissipative term, initial time step, and initial time are known; they are \( X_j^0 = (x_j^0, y_j^0), (h_j^0), d\tau_0 \), and \( t_0 \).

Step 2 (move the mesh). (a) Suppose the current time \( t = t_n (n = 0, 1, \ldots, K) \), the numerical solution \((u_h)^n\) of the equation, the coordinate \( X_j^n, \mathcal{F}_j^n \) of the mesh, and the time step \( d\tau_n \) are known; (b) we make use of the energy function (9) and ODE function (10) to obtain the coordinate \( X_j^{n+1} = (\xi_j^{n+1}, \eta_j^{n+1}) \), and then we derive the relationship \( \Phi_h(t) \) between the new computational mesh \( \mathcal{F}_c^{n+1} = \{\xi_i^{n+1}\} \) and the physical mesh \( \mathcal{F}_h^n \), where \( \mathcal{F}_h^n \) can be expressed by \( \Phi_h(\mathcal{F}_c^{n+1}) \). When the reference computational mesh \( \tilde{\mathcal{F}}_c \) and \( \Phi_h(t) \) are known, we get the new physical mesh \( \mathcal{F}_h^{n+1} = \Phi_h(\tilde{\mathcal{F}}_c) \) approximately; the new coordinate is \( \tilde{X}_j^{n+1} \); (c) when \( t \in [t_n, t_{n+1}] \), using linear interpolation \( X_j(t) = ((t_{n+1} - t)/d\tau_n)X_j^n + ((t - t_n)/d\tau_n)\tilde{X}_j^{n+1} \), the velocity of the corresponding grid node is \( \tilde{X}_j(t) = (\tilde{X}_j^{n+1} - X_j^n)/d\tau_n \). The fifth-order Radau II A method is used in time discretization; if you are interested in more details, please refer to [44]. From the method, we can determine the time step \( d\tau_n \); if the real time step \( d\tau_n < d\tau_m \), the next time level, the mesh coordinate, and the time step of next time level update: \( t_{n+1} \leftarrow t_n + d\tau_n, X_j^{n+1} \leftarrow X_j^n + d\tau_nX_j, \) and \( d\tau_{n+1} \leftarrow d\tau_n \).

Step 3. On the new physical mesh \( \mathcal{F}_h^{n+1} \), use the discrete version (16) based on the linear finite element method to solve (12) from time \( t_n \) to \( t_{n+1} \); we get the new numerical solution at the new time level; the solution is \((u_h)^{n+1}\).

Step 4. When \( t_{n+1} < T, n \rightarrow n + 1 \), go to Step 2; else, stop the computation and save the value of solution.

### 3. Numerical Examples

**Example 1** (a single solitary wave in 1D BBM-Burgers equation). In the one-dimensional model,

\[
u_t + \alpha u_x + \beta uu_x - \gamma u_{xx} - \mu u_{xxx} = F,
\]

\[x \in (-20, 30), t \in (0, T],
\]

where the coefficients \( \alpha = \gamma = \mu = 1, \beta = -1 \), the final time \( T = 10 \), the exact solution of the equation is \( u_{ex}(x, t) = \text{sech}(x-t) \), and the source term \( F = F(x, t) = (1 - 6 \tanh^2(x-t) - 2 \tanh^2(x-t) + (\text{sech}(x-t) + 5) \tanh(x-t)) \text{sech}(x-t) \).

We solve the numerical solution with the method we introduced; with the different element numbers, the corresponding error and the order of convergence for both moving mesh and fixed mesh are reported in Table 1. When the element number is the same, the error based on moving mesh is always smaller than that on fixed mesh, and the convergence order of the two kinds of meshes can lead to the second order in the spatial direction. Comparison between the exact solution and the numerical solution based on moving mesh at the final time \( T = 10 \) is shown in Figure 1. From the figure, we can see that the grid points mainly focus on the domain, where the solution fluctuates greatly, and match the exact solution well. With time going on, the moving mesh trajectory is plotted in Figure 1; the feature of the moving process indicates that most of the points focus on nearby solitary waves and move with the propagation of the solitary wave, which also demonstrates the superiority of the moving mesh method.

**Example 2** (the elastic collision of three solitary waves in 1D BBM-Burgers equation). In this example, we study the elastic collision of three solitary waves and simulate numerical solutions based on moving mesh and fixed mesh; the BBM-Burgers equation has the following form:

\[
u_t + u_x + uu_x - 0.01u_{xx} - u_{xxx} = F,
\]

\[x \in (-50, 300), t \in (0, 50],
\]

\[u(-50, t) = u(300, t) = 0,
\]

\[u(-50, 0) = u(300, 0) = 0,
\]

\[u(x, 0) = u_0(x),
\]

\[x \in (-50, 300),
\]

where \( u_0(x) \) is a single solitary wave in 1D BBM-Burgers equation.
with the initial condition $u_0(x) = \sum_{j=1}^{3} 3d_j \text{sech}^2(k_j(x-x_j))^2$, where $k_1 = 0.39$, $k_2 = 0.3$, $k_3 = 0.3$, $x_1 = 10, x_2 = 28, x_3 = 52$, $d_j = 4k_j^2/(1 - 4k_j^2)$, and the final time $T = 50$.

In this example, the exact solution of the equation is unknown; we mainly want to show the superiority of the moving mesh. In Figure 2, the conservation laws (2) are plotted; we can see that $\Delta E_1(T)$ is close to zero and $\Delta E_2(T)$ is a little bigger than zero. The two conservation laws are not conserved by finite element approximation on the moving mesh, because, in this case, $X \equiv 0$ and both matrices $A$ and $M$ are time-dependent (if you want to know more, read [45]). Since the exact solution is unknown, we take the element number $N = 8000$ on the fixed mesh as a reference solution and plot the numerical solutions of moving mesh and fixed mesh with element number $N = 400$ at final time $T = 50$ in Figure 3. It can be seen that the numerical solution of fixed mesh is on the left of the reference solution and the wave peaks are shorter than those in the reference solution; there are several small waves in the left of the first wave on the fixed mesh. The result also shows that the numerical solution of moving mesh can match the reference better than that based on the fixed mesh. If we consider the computer memory and computational efficiency, the moving mesh is better for that it moves its mesh points quickly to the place where the numerical solution changes rapidly to get higher distinguishability with fewer elements.

Figure 1: Example 1. (a) The exact solution and the numerical solution based on adaptive moving mesh with element number $N = 200$ at $t = 10$; (b) the corresponding mesh trajectories.

Figure 2: Example 2. The variation of mass function ($E_1$) and energy function ($E_2$) for the elastic collision of three solitary waves in the BBM-Burgers equation.
The propagation process of the three solitary waves is plotted in Figure 4.

Example 3 (different coefficients in 1D BBM-Burgers equation with a high-order dissipative term). In this case, we mainly consider the following BBM-Burgers equation with a fourth-order dissipative term; the equation is

\[ u_t + u_x + 2uu_x - u_{xxt} - \gamma u_{xxx} + \delta u_{xxxx} = 0, \]
\[ x \in (-100, 150), \quad t \in (0, 20], \] \[ u(-100, t) = u(150, t) = 0, \quad t \in (0, 20], \]
\[ u(x, 0) = u_0(x), \quad x \in (-100, 150). \]

(20)

The initial condition \( u_0(x) = 0.15 \text{sech}(k(x-40))^2 \), where \( k = 0.5 \times \sqrt{1/11} \); when the coefficients \( \gamma, \delta \) change, the maximum value \( U_{\text{max}} \) of the numerical solution changes as well.

In this example, we mainly want to talk about the dissipation effect of the dissipative terms \( u_{xx} \) and \( u_{xxxx} \) in the BBM equation. Numerical solutions obtained with different coefficients and one of the coefficients fixed at \( T = 20 \) are shown in Figure 5. When two BBM equations with one dissipative term \( u_{xx} \) or \( u_{xxxx} \) have the same dissipation effect or similar wave peak at the final time, the corresponding coefficient of dissipative term is plotted in Figure 6. From these figures, we find that when the coefficient \( \gamma \) of the second-order dissipative term increases and the coefficient \( \delta \)
of the fourth-order dissipative term is fixed, the numerical solution decreases a lot; however, when the coefficient $\delta$ increases and the coefficient $\gamma$ is fixed, the numerical solution changes a little. The numerical simulation provides three sets of data to show the result in Figure 5: $\delta = 1$; when $\gamma = 0.1, 1, 10$, the corresponding maximum values of the numerical solutions are $U_{\text{max}} = 0.139, 0.1324, 0.1091$, $\gamma = 1$; when $\delta = 0.1, 1, 10$, the corresponding values are $U_{\text{max}} = 0.0947, 0.0936, 0.0866$. A lot of numerical tests have been carried out to adjust and confirm the corresponding coefficient of different dissipative terms to keep the equation with dissipation effect or similar wave peak at $T = 20$. The value of the wave peak is $U_{\text{max}} = 0.0764$ at $T = 20$, and the corresponding data are shown in Figure 6. When $\delta = 0$, $\gamma = 1.92$, with time going on, $U_{\text{max}}$ drops steadily; however, when $\gamma = 0$, $\delta = 100$, the value drops quickly before time $t = 2$, ...
and then the value drops slowly and steadily. We simulate a series of numerical examples with different coefficients; all of these data show us that the dissipative term $u_{xx}$ plays a more important role in dissipation than the dissipative term $u_{xxxx}$.

**Example 4** (two solitary waves in 2D BBM-Burgers equation). This example has two solitary waves and we want to verify the convergence order for a 2D BBM-Burgers equation; the equation has the following formulation:

$$u_t + (\alpha + \beta \cos(u))(u_x + u_y) - \gamma (u_{xx} + u_{yy}) - \mu (u_{xxt} + u_{yyt}) = F, \quad (x, y) \in \Omega, \ t \in (0, T],$$

(21)

where $\alpha, \gamma, \mu = 1$, $\beta = -1$, and the exact solution of the equation is $u_{ex}(x, y, t) = e^{t} (\text{sech}^2(x_1) + \text{sech}^2(x_2))$, where $x_1 = x + y - 1, x_2 = x + y + 1, \Omega = (-2, 2) \times (-2, 2), T = 1$. To maintain the balance of the equation, $F = F(x, y, t) = e^{t}((9(\text{sech}^2(x_1) + \text{sech}^2(x_2)) - 4(\text{sech}^2(x_1) \tan h(x_1) + \text{sech}^2(x_2) \tan h(x_2)) - 24(\text{sech}^2(x_1) \tan h^2(x_1) + \text{sech}^2(x_2) \tan h^2(x_2)) + 4 \cos(e^t (\text{sech}^2(x_1) + \text{sech}^2(x_2))))(\text{sech}^2(x_1) \tan h(x_1) + \text{sech}^2(x_2) \tan h(x_2)))$.

Numerical solutions are shown in Table 2 and Figure 7. From Table 2, we can find that the finite element method leads to the same second order of convergence for both fixed and moving meshes, and the convergence result is similar to Example 1. The error of the numerical solution on the moving mesh is smaller than that on the fixed mesh with the same element number. What is more, with fewer mesh elements, the moving mesh reaches second-order convergence more quickly than the fixed mesh. The trajectory and the numerical solution of both the moving mesh and the fixed mesh at final time can be seen in Figure 7. From the figure, we can see that most points focus on the place where the numerical solution changes a lot.

**Figure 7:** Example 4. Meshes (a, c) and the corresponding numerical solutions (b, d) with $N = 6400$. 

(a) Moving mesh  
(b) The corresponding numerical solution  
(c) Fixed mesh  
(d) The corresponding numerical solution
4. Conclusion

In this paper, we study the numerical model based on adaptive moving mesh finite element method to simulate 1D and 2D BBM-Burgers equations with a high-order dissipative term. The method can be used in equations with mixed derivatives and high-order derivative terms; on the moving mesh, new variables need to be introduced to transform the equation into coupled equations. A series of numerical examples demonstrate that the finite element method based on moving mesh keeps a second-order convergence in space as that based on fixed mesh. With the same element number, the finite element moving mesh method improves the distinguishability and provides a smaller error than the fixed mesh. Meanwhile, numerical tests indicate that the dissipative term $u_{xx}$ plays a more important role than $u_{xxxx}$ in dissipation. We compare two BBM equations with different dissipative terms $u_{xx}$ and $u_{xxxx}$; to keep the maximum value $U_{\max}$ of the numerical solutions the same at the final time, the coefficient of $u_{xxxx}$ is around 50 times that of $u_{xx}$ in the same BBM equation with different dissipative terms in the dissipation aspect.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by NSFC (41375115), Open Fund of the Key Laboratory of Meteorological Disaster of the Ministry of Education (Nanjing University of Information Science and Technology) (KLME1507), and Open Fund of the Key Laboratory of Ocean Circulation and Waves, Chinese Academy of Sciences (KLOCAW1401).

References


