East Asian Math. J.
Vol. 33 (2017), No. 5, pp. 495–509
http://dx.doi.org/10.7858/eamj.2017.034

A FINITE ELEMENT SOLUTION FOR THE CONSERVATION FORM OF BBM-BURGERS' EQUATION

Yang Ning, Mingzhe Sun, and Guangri Piao*

Abstract. With the accuracy of the nonlinearity guaranteed, plenty of time and large memory space are needed when we solve the finite element numerical solution of nonlinear partial differential equations. In this paper, we use the Group Element Method (GEM) to deal with the nonlinearity of the BBM-Burgers Equation with Conservation form and perform a numerical analysis for two particular initial-boundary value (the Dirichlet boundary conditions and Neumann-Dirichlet boundary conditions) problems with the Finite Element Method (FEM). Some numerical experiments are performed to analyze the error between the exact solution and the FEM solution in MATLAB.

1. Introduction

In recent years, the nonlinear partial differential equation is widely used in many fields to build mold for all kinds of physical phenomenon, such as fluid mechanics, solid mechanics, geography, chemical physics, etc. To establish the corresponding mathematical model for the physical phenomena, many nonlinear partial differential equations arise at the historic moment, and BBM-Burgers Equation with variable coefficients is one of the most famous ones. In this parper, we mainly consider BBM-Burgers equation with respect to one dimension, which is defined as follows:

$$u_t(t,x) - u_{xxt}(t,x) - qu_{xx}(t,x) + u(t,x)u_x(t,x) + u_x(t,x) = f(t,x) \tag{1.1}$$

This is a nonlinear parabolic PDE, containing both a convective term uu_x and dissipative term u_{xx}. Eq. (1.1) is the alternative regularized long-wave equation proposed by Peregrine [3] and Benjamin et al. [24]. Eq. (1.1) features a balance between nonlinear and dispersive effects, but takes no account of dissipation. In the physical sense, Eq. (1.1) with the dissipative term qu_{xx} is proposed if

Received September 5, 2017; Accepted September 19, 2017.
2010 *Mathematics Subject Classification.* 65M60,35G60.
Key words and phrases. BBM-Burgers equation; Conservation Form; Group Element Method; Finite Element Method.
This work was financially supported by the college students' Innovation Training Project of Yanbian University in 2017.
* Corresponding author.

(pISSN 1226-6973, eISSN 2287-2833)

the good predictive power is desired, such a problem arises in the phenomena for both the bore propagation and the water waves. Since the dispersive effect of (1.1) is the same as the Benjamin-Bona-Mahony equation

$$u_t - u_{xxt} + u_x + uu_x = 0 \tag{1.2}$$

while the dissipative effect is the same as the Burgers equation

$$u_t - qu_{xx} + u_x + uu_x = 0 \tag{1.3}$$

we call (1.1) the BBM-Burgers equations, but it is proposed neither by Benjamin, Bona, and Mahony nor by Burgers, see Mei [25].

For the mathematical theory and physical significance of the Eq. (1.1), (1.2), and (1.3), we refer the reader to [3,7-8,13-14,24,28] and the references therein. Numerical methods have been proposed by several researchers, based on either finite differences [10,19-21,25], finite elements [4,11,15,22], or Adomian decomposition scheme [5-6,12].

The main contribution of this article is to use GEM to deal with the nonlinearity of the BBM-Burgers Equation with Conservation form and perform a numerical analysis for two particular initial-boundary value (the Dirichlet boundary conditions and Neumann-Dirichlet boundary conditions) problems with the Finite Element Method (FEM).

The rest of article is organized as follows, in Section 2, an introduction of the Finite Element Method is established and a numerical analysis for two particular initial-boundary value problems is performed with FEM. In Section 3, some numerical experiments are performed to analyze the error between the exact solution and the FEM solution in MATLAB.

2. The Finite Element Method

The Finite Element Method (FEM) is a rather general numerical method that is often used to approximate partial differential equations (PDEs). If the PDE is time dependent, then the problem can be reduced to a system of ODEs which can be numerically integrated by known techniques. In this paper we use standard piecewise linear basis functions for our approximations. Therefore, we divide the unit interval $[0, 1]$ into N subintervals $[x_i, x_{i+1}]$ of uniform length $h = \frac{1}{N}$ where $x_i = ih$ for $i = 0, \cdots, N$. On each interval the global basis functions are defined by

$$\phi_0(x) = \begin{cases} \frac{x_1 - x}{h} & x \in [0, x_1] \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq N - 1$,

$$\phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h} & x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{h} & x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

and

$$\phi_N(x) = \begin{cases} \frac{x-x_{N-1}}{h} & x \in [x_{N-1}, x_N] \\ 0 & \text{otherwise} \end{cases}$$

We form an approximation of $u(t,x)$ in the space spanned by the piecewise linear basis functions by setting

$$u^N(t,x) = \sum_{j=0}^{N} \alpha_j(t)\,\phi_j(x)$$

where $\alpha_j(t)$ represents the nodal unknown value of $u(t,x_j)$ at the jth node at time t.

The standard Finite Element approach yields a nonlinear time dependent equation. We also take advantage of the Group Finite Element (GFE) method, described by Fletcher [2]. This simplifies the nonlinear term so that one can take advantage of grouping similar terms. The Burgers' equation in conservation form expresses the nonlinearity $u(t,x)\,u_x(t,x)$ as $\frac{1}{2}\left[u(t,x)^2\right]_x$ leading us to the approximation

$$u(t,x)^2 \approx u^N(t,x)^2 \approx \sum_{j=0}^{N} \alpha_j(t)^2 \phi_j(x)$$

This was shown in [1,9,16,23,26] to provide improved stability and computational efficiency since matrices do not need to be assembled at each time step.

In this paper, We consider BBM-Burgers equation in conservation form given by

$$u_t(t,x) - u_{xxt}(t,x) - q u_{xx}(t,x) + \frac{1}{2}\left[u(t,x)^2\right]_x + u_x(t,x) = f(t,x)$$

The forcing f is assumed to be at least L^2 in space and time. We shall focus on two particular initial-boundary value problems. The first problem has Dirichlet boundary conditions and the other has a Neumann-Dirichlet boundary condition.

2.1. BBM-Burgers Equation with Dirichlet Boundary Conditions

Consider the first problem

$$u_t(t,x) + \frac{1}{2}\left[u(t,x)^2\right]_x - u_{xxt}(t,x) - q u_{xx}(t,x) + u_x(t,x) = f(t,x) \tag{2.1}$$

where $x \in [0,1]$, $t \in [0,t_f]$, the boundary conditions are given by

$$u(t,0) = 0 \quad u(t,1) = 0$$

and the initial condition is

$$u(0,x) = u_0(x)$$

Multiplying both sides of (2.1) by a test function $v(x)$ and integrating yields

$$\int_0^1 \left(u_t(t,x) + \frac{1}{2}\left[u(t,x)^2\right]_x + u_x(t,x)\right) v(x)dx - \int_0^1 u_{xxt}(t,x)v(x)dx$$

$$-q\int_0^1 u_{xx}(t,x)v(x)dx = \int_0^1 f(x)v(x)dx \qquad (2.2)$$

If $v(x)$ is piecewise smooth, then $v'(x) \in L^2(0,1)$, we can apply integration by parts to the second and the third term on the left, and take advantage of the essential Dirichlet boundary conditions. Thus (2.2) becomes the weak form of BBM-Burgers' equation with Dirichlet boundary conditions given by

$$\int_0^1 \left(u_t(t,x) + \frac{1}{2}\left[u(t,x)^2\right]_x + u(t,x)\right) v(x)dx + \int_0^1 q u_x(t,x)\, v_x(x)dx$$
$$+\int_0^1 u_{xt}(t,x)\, v_x(x)dx = \int_0^1 f(t,x)\, v(x)\, dx \qquad (2.3)$$

Note that (2.3) must hold for any piecewise smooth function $v(x)$.

Using the approximation by piecewise linear basis functions we write $u(t,x) \approx u^N(t,x) = \sum\limits_{j=0}^{N} \alpha_j(t)\,\phi_j(x)$ and using the group approximation $u^N(t,x)^2 \approx \sum\limits_{j=0}^{N} \alpha_j(t)^2 \phi_j(x)$. Since this identity holds for arbitrary piecewise smooth $v(x)$, for each $i = 0,1,...,N$, we can set $v(x) = \phi_i(x)$, and for all j, $\alpha_j(t)$ does not depend on x, we can move those terms outside. For each $i = 0,1,...,N$, the equation (2.3) can finally be written as a system of $N+1$ ODEs given by

$$(M+C)\dot{\alpha}(t) + \frac{1}{2}\mathfrak{B}(\alpha(t))\,\alpha(t) + qC\alpha(t) + \mathfrak{B}(\alpha(t)) = F(t) \qquad (2.4)$$

where $M_{ij} = \int_0^1 \phi_j(x)\,\phi_i(x)\,dx$, $C_{ij} = \int_0^1 \phi'_j(x)\,\phi'_j(x)\,dx$ and $F_i(t) = \int_0^1 f(t,x)\,\phi_i(x)\,dx$, $i,j = 0,1,...,N$, $\mathfrak{B}(\alpha(t))$ is a matrix that depends on $\alpha(t)$.

Taking advantage of the Dirichlet boundary conditions we know $\alpha_0(t)$ and $\alpha_N(t)$ for all t. Hence we can eliminate these equations and reduce the size of (2.4) from $N+1$ to $N-1$ equations by solving only for the internal nodes. Thus the matrices M, C and $\mathfrak{B}(\alpha(t))$ are given by

$$M = \frac{h}{6}\begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix}_{[N-1\times N-1]}$$

$$C = \frac{1}{h}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}_{[N-1\times N-1]}$$

and

$$\mathfrak{B}\left(\alpha\left(t\right)\right)=\begin{pmatrix} 0 & \frac{1}{2}\alpha_2\left(t\right) & & & \\ -\frac{1}{2}\alpha_1\left(t\right) & 0 & \frac{1}{2}\alpha_3\left(t\right) & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2}\alpha_{N-3}\left(t\right) & 0 & \frac{1}{2}\alpha_{N-1}\left(t\right) \\ & & & -\frac{1}{2}\alpha_{N-2}\left(t\right) & 0 \end{pmatrix}_{[N-1\times N-1]}$$

respectively.

During computation the term $\mathfrak{B}\left(\alpha\left(t\right)\right)\alpha\left(t\right)$ can be efficiently implemented as

$$\mathfrak{B}\left(\alpha\left(t\right)\right)\alpha\left(t\right)=\begin{pmatrix} 0 & \frac{1}{2} & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2} & 0 & \frac{1}{2} \\ & & & -\frac{1}{2} & 0 \end{pmatrix}_{[N-1\times N-1]}\begin{bmatrix} \alpha_1(t)^2 \\ \alpha_2(t)^2 \\ \vdots \\ \alpha_{N-2}(t)^2 \\ \alpha_{N-1}(t)^2 \end{bmatrix}$$

We also need to find the approximate initial condition $u_0(x)$ in terms of the basic functions $\phi_i(x)$. Thus, we assume that $u_0\left(x\right)\approx {u_0}^N\left(x\right)=\sum\limits_{j=0}^{N}\alpha_j\left(0\right)\phi_j\left(x\right)$ and note that

$$\int_0^1 {u_0}^N\left(x\right)dx=\int_0^1 u_0\left(x\right)v\left(x\right)dx \tag{2.5}$$

Using the approximation ${u_0}^N\left(x\right)=\sum\limits_{j=0}^{N}\alpha_j\left(0\right)\phi_j\left(x\right)$ and enforcing the essential boundary condition, for each $i=1,...,N-1$, the equation (2.5) generates an equation which can be written as the matrix equation

$$M\alpha(0)=G \tag{2.6}$$

where M is the same as (2.4) and $G_i=\int_0^1 u_0(x)\phi_i(x)dx$. Consequently, $\alpha(0)=M^{-1}G$.

Combining (2.4) and (2.6) we finally obtain the initial value ODE system

$$\begin{aligned}&\dot{\alpha}\left(t\right)=\left(M+C\right)^{-1}[F\left(t\right)-\frac{1}{2}\mathfrak{B}\left(\alpha\left(t\right)\right)\alpha\left(t\right)-qC\alpha\left(t\right)-\mathfrak{B}\left(\alpha\left(t\right)\right)]\\&\alpha(0)=M^{-1}G\end{aligned}$$

Here, $\alpha(t)=[\alpha_1(t),\alpha_2(t),...,\alpha_{N-1}(t)]^T$.

2.2. BBM-Burgers Equation with Neumann-Dirichlet Boundary Conditions

Now consider the second problem

$$u_t(t,x)+\frac{1}{2}\left[u(t,x)^2\right]_x-u_{xxt}(t,x)-qu_{xx}(t,x)+u_x(t,x)=f(t,x)$$

where $x \in [0,1]$, $t \in [0,t_f]$, the boundary conditions are given by

$$u_x(t,0) = \delta \in \mathbb{R} \quad u(t,1) = 0$$

and initial value given by

$$u\left(0,x\right) = u_0\left(x\right)$$

Again, multiply by a test function $v(x)$ and integrating yields we obtain

$$\int_0^1 \left(u_t(t,x) + \frac{1}{2}\left[u(t,x)^2\right]_x + u_x(t,x)\right) v(x)dx - \int_0^1 u_{xxt}(t,x)v(x)dx$$

$$-q\int_0^1 u_{xx}(t,x)v(x)dx = \int_0^1 f(x)v(x)dx \tag{2.7}$$

Also, if $v(x)$ is piecewise smooth, we can apply integration by parts on the second term on the left. We know from the right essential Dirichlet boundary condition that $u_x(t,1)v(x) = 0$ for all t and from the left boundary condition that $u_x(t,0) = \delta$. From (2.7) we obtain the weak form of Burgers' equation with Neumann-Dirichlet boundary conditions given by

$$\int_0^1 \left(u_t\left(t,x\right) + \frac{1}{2}\left[u(t,x)^2\right]_x + u\left(t,x\right)\right) v\left(x\right)dx + \int_0^1 qu_x\left(t,x\right) v_x\left(x\right)dx$$

$$+\int_0^1 u_{xt}\left(t,x\right) v_x\left(x\right)dx + q\delta v(0) = \int_0^1 f\left(t,x\right) v\left(x\right) dx \tag{2.8}$$

for any piecewise smooth function $v(x)$.

Using an approximation by piecewise linear basis functions we write $u\left(t,x\right) = \sum\limits_{j=0}^{N} \alpha_j\left(t\right)\phi_j\left(x\right)$ and using the group approximation $u^N(t,x)^2 \approx \sum\limits_{j=0}^{N} \alpha_j(t)^2\phi_j\left(x\right)$. Since this holds for arbitrary piecewise smooth $v(x)$, we let $v(x) = \phi_i(x)$ and for all j, $\alpha_j(t)$ does not depend on x, we can move those terms outside the integral. For each $i = 0,1,...,N$, the equation (2.8) generates $N+1$ equations and hence we have a system of $N+1$ ODEs given by

$$(M+C)\dot{\alpha}\left(t\right) + \frac{1}{2}\mathfrak{B}\left(\alpha\left(t\right)\right)\alpha\left(t\right) + q(D + C\alpha\left(t\right)) + \mathfrak{B}\left(\alpha\left(t\right)\right) = F\left(t\right) \tag{2.9}$$

where $M_{ij} = \int_0^1 \phi_j\left(x\right)\phi_i\left(x\right)dx$, $C_{ij} = \int_0^1 \phi'_j\left(x\right)\phi'_j\left(x\right)dx$, $D_{ij} = \delta[1,0,...,0]^T$ and $F_i\left(t\right) = \int_0^1 f\left(t,x\right)\phi_i\left(x\right)dx$, $i,j = 0,1,...,N$, $\mathfrak{B}\left(\alpha\left(t\right)\right)$ is a matrix that depends on $\alpha\left(t\right)$.

Taking advantage of the right Dirichlet boundary condition we know $\alpha_N\left(t\right)$ for all t. Hence we can eliminate this equation and reduce the size of (2.9) from $N+1$ to N equations by solving only for the internal nodes. Hence, the $[N \times N]$

matrices M,C and $\beta(\alpha(t))$ are given as

$$M = \frac{h}{6}\begin{pmatrix} 2 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix}_{[N\times 2.4N]}$$

$$C = \frac{1}{h}\begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}_{[N\times N]}$$

and

$$\mathfrak{B}\left(\alpha\left(t\right)\right) = \begin{pmatrix} -\frac{1}{2}\alpha_0\left(t\right) & \frac{1}{2}\alpha_1\left(t\right) & & & \\ -\frac{1}{2}\alpha_0\left(t\right) & 0 & \frac{1}{2}\alpha_2\left(t\right) & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2}\alpha_{N-3}\left(t\right) & 0 & \frac{1}{2}\alpha_{N-1}\left(t\right) \\ & & & -\frac{1}{2}\alpha_{N-2}\left(t\right) & 0 \end{pmatrix}_{[N\times N]}$$

respectively. Again, note that during computation the term $\mathfrak{B}(\alpha(t))$ is computed by

$$\mathfrak{B}\left(\alpha\left(t\right)\right)\alpha\left(t\right) = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2} & 0 & \frac{1}{2} \\ & & & -\frac{1}{2} & 0 \end{pmatrix}_{[N\times N]} \begin{bmatrix} \alpha_0(t)^2 \\ \alpha_1(t)^2 \\ \vdots \\ \alpha_{N-2}(t)^2 \\ \alpha_{N-1}(t)^2 \end{bmatrix}$$

We also construct the initial condition $u_0(x)$ in terms of the basic functions $\phi_i(x)$. Thus, as before we assume that $u_0\left(x\right) \approx {u_0}^N\left(x\right) = \sum\limits_{j=0}^{N} \alpha_j\left(0\right)\phi_j\left(x\right)$ and obtain

$$\int_0^1 {u_0}^N\left(x\right)dx = \int_0^1 u_0\left(x\right)v\left(x\right)dx \tag{2.10}$$

Using the approximation ${u_0}^N\left(x\right) = \sum\limits_{j=0}^{N} \alpha_j\left(0\right)\phi_j\left(x\right)$ and enforcing the right essential Dirichlet boundary condition for each $i = 1, ..., N-1$, the equation (2.10) generates an equation which can be written as the matrix equation

$$M\alpha(0) = G \tag{2.11}$$

where M is the same as (2.9) and $G_i = \int_0^1 u_0(x)\phi_i(x)dx$. Hence, $\alpha(0) = M^{-1}G$.

Combining (2.9) and (2.11) we finally obtain the initial value ODE system

$$\dot{\alpha}(t) = (M + C)^{-1}[F(t) - \frac{1}{2}\mathfrak{B}(\alpha(t))\alpha(t) - q(D + C\alpha(t)) - \mathfrak{B}(\alpha(t))]$$
$$\alpha(0) = M^{-1}G$$

Here, $\alpha(t) = [\alpha_0(t), \alpha_1(t), ..., \alpha_{N-1}(t)]^T$.

3. Numerical Experiments

Using finite elements we approximate the true Burgers' equations by finite dimensional ODE systems. We will solve these ODE systems in MATLAB to obtain the numerical solutions. The Method of Manufactured Solutions (MMS) is used to generate analytical solutions which are used to test convergence of the finite element methods. The details of the method can be found in the paper by Roache [18] or the book by Oberkampf [27]. Here we make a brief introduction of an MMS example.

3.1. Brief Introduction of an MMS Example

Consider Burger's equation operator given by

$$L[u(t,x)] := u_t(t,x) + \frac{1}{2}[u(t,x)^2]_x - qu_{xx}(t,x)$$

We want $\tilde{u}(t,x)$ to be our exact solution. Thus we apply the operator to $\tilde{u}(t,x)$ to obtain

$$L[\tilde{u}(t,x)] := \tilde{u}_t(t,x) + \frac{1}{2}[\tilde{u}(t,x)^2]_x - q\tilde{u}_{xx}(t,x)$$

By setting the $f(t,x) = L[\tilde{u}(t,x)]$, we guarantee that $\tilde{u}(t,x)$ is an exact solution to the modified equation

$$u_t(t,x) + \frac{1}{2}[u(t,x)^2]_x - qu_{xx}(t,x) = f(t,x)$$

Here, we assume the exact solution is of the form

$$\tilde{u}(t,x) = h(t)r(x)$$

Then the forcing term has the form

$$f(x,t) = \frac{d}{dt}h(t)r(x) + h(t)^2\frac{d}{dx}r(x) - qh(t)\frac{d^2}{dx^2}r(x)$$

By comparing the approximate solution and the true solution we can determine the error for a particular discretization level. The error is calculated in the L^2 norm. In particular, if $f(t,x) \in L^2([0,T] \times [0,1]; dt \times dx)$, the norm $\|L^2\|$ is given by

$$\|f\|_{L^2} = \left(\int_0^T \int_0^1 |f(t,x)|^2 dx dt\right)^{\frac{1}{2}}$$

Therefore the relative error is given by

$$Err^{rel}(N) = \frac{\left\|u - u^N\right\|_{L^2}}{\|u\|_{L^2}}$$

To evaluate the L^2 norm we utilize a 3 point gauss quadrature in space and for time we use the approximation

$$\int_0^T f(t)dt \approx \sum_{t=0}^{M} f(i)k$$

where $k = T/M$ and M is the number of times steps.

As the mesh spacing $h_N = 1/N$ is refined, standard finite element theory implies convergence to the true solution. If we assume the error for any mesh spacing has the form $e_{N_i} = Chp_{N_i}$, where C is a constant that does not depend on h, then we can calculate the observed order of convergence between two discretizations by

$$\ln \frac{e_{N_i}}{e_{N_{i+1}}} = p \ln \frac{h_{N_i}}{h_{N_{i+1}}} \to p = \frac{\ln \frac{e_{N_i}}{e_{N_{i+1}}}}{\ln \frac{h_{N_i}}{h_{N_{i+1}}}}$$

In the next section, we will verify the code using the MMS above. We present several experiments by changing the parameter value for both the Dirichlet boundary conditions and Neumann-Dirichlet boundary conditions problems. We also test several of the built-in MATLAB ODE solvers for accuracy and performance using the default settings.

3.2. Dirichlet Boundary Conditions Problem

Consider the problem with Dirichlet boundary conditions and exact solution given by

$$u(t, x) = e^{-t} sin(\pi x)$$

The initial condition is given by

$$u_0(x) = sin(\pi x)$$

This meets the boundary conditions and the associated MMS forcing term is

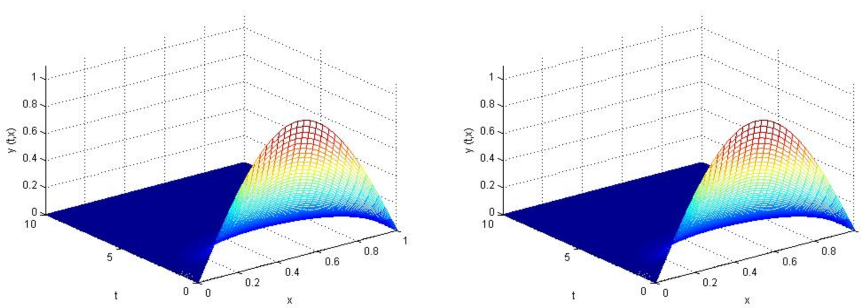
$$f(t, x) = e^{-t}(sin(\pi x)(-1 - \pi^2 + q\pi^2) + \pi cos(\pi x)(e^{-t} sin(\pi x) + 1))$$

Table 1 shows the FEM Results for $T = 10$, $Re = 100$ and Dirichlet boundary conditions with various parameter values and solvers. In general, ODE45 and ODE23 are both accurate but requires a significant amount of time for a fine mesh. We also see that ODE15s is very fast but the errors may actually grow as the mesh is refined.

Sample plots of the exact solution, the FEM solution and errors are found in Figures 1-3. In Figure 1 we compare the exact solution to the finite element numerical approximation for 16 elements when $q = 1/100$, we note the visual agreement between them. Figure 2 displays the error between the exact solution and the numerical solution. The global L^2 error is less than 3×10^{-3} in time and space. As shown in Figure 3, using $N = 32$ elements the discretization error is small and our computed numerical solution matches the exact analytic solution.

MATLAB Solver	Number of Elements	Solver time	Relerr	Order
ODE45	$N = 8$	4.9629	3.650×10^{-2}	-
	$N = 16$	10.7906	9.600×10^{-3}	1.927
	$N = 32$	40.1462	2.400×10^{-3}	2.000
	$N = 64$	200.406	5.953×10^{-4}	2.011
	$N = 128$	1407.40	1.449×10^{-4}	2.039
ODE23	$N = 8$	23.7733	3.640×10^{-2}	-
	$N = 16$	36.657	9.400×10^{-3}	1.953
	$N = 32$	115.6316	2.300×10^{-3}	2.031
	$N = 64$	457.4316	4.849×10^{-4}	2.264
	$N = 128$	2138.30	7.063×10^{-5}	2.779
ODE15s	$N = 8$	5.8159	3.680×10^{-2}	-
	$N = 16$	6.0075	9.800×10^{-3}	1.909
	$N = 32$	17.3624	1.300×10^{-3}	2.914
	$N = 64$	49.333	1.800×10^{-3}	-0.469
	$N = 128$	153.38	2.200×10^{-3}	-0.290

Table 1: FEM Results for $T = 10$, $Re = 100$ and Dirichlet boundary conditions



(a) Exact solution (b) FEM solution

Figure 1: Exact solution and FEM using ODE23 for $N = 32$; $Re = 100$ and Dirichlet boundary conditions

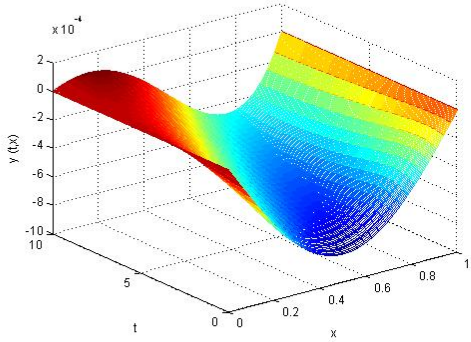


Figure 2: FEM error using ODE23 for $N = 32$; $Re = 100$ and Dirichlet boundary conditions

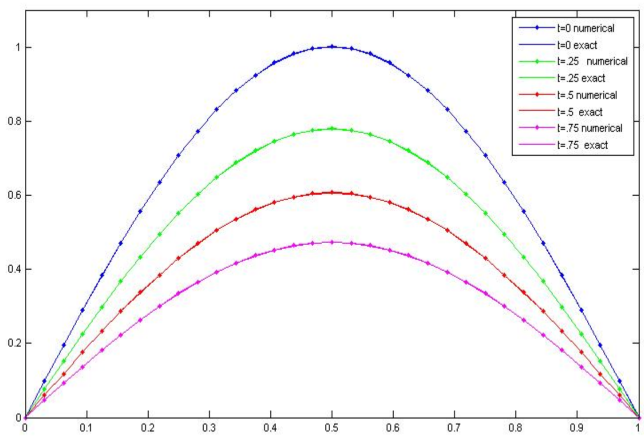


Figure 3: Cross section of exact and FEM using ODE23 for $N = 32$; $Re = 100$ and Dirichlet boundary conditions

3.3. Neumann-Dirichlet Boundary Conditions Problem

Consider the problem with Neumann-Dirichlet boundary conditions, $\delta = 0$ and exact solution given by

$$u(t,x) = e^{-t}(1 - x^2)$$

The initial condition is given by

$$u_0(x) = (1 - x^2)$$

This meets the boundary conditions and the associated MMS forcing term is

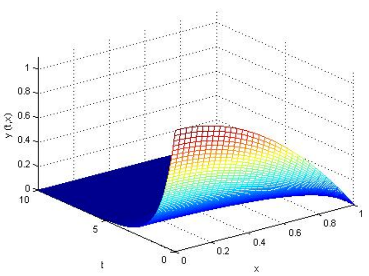
$$f(t,x) = e^{-t}(x^2 - 2x - 3 + 2q + 2e^{-t}(x^3 - x))$$

Results for various parameter values and solvers are found in Table 2. We see again that for all values of q, ODE45 produced convergent solutions as the spatial mesh is refined. However the errors of ODE23 actually grow as the mesh was refined from $N = 64$ to $N = 128$. ODE15s also demonstrate monotone convergence, and the observed order of accuracy is higher than ODE45.

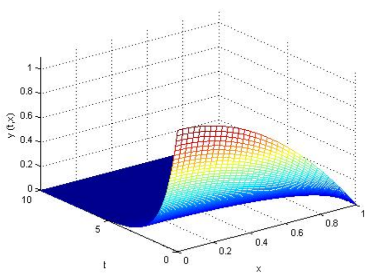
Sample plots of the exact solution, the FEM solution and errors are found in Figures 4-6. In Figure 4 we compare the exact solution to the finite element numerical approximation. Figure 5 clearly shows that the global error for the Neumann-Dirichlet boundary condition problem is larger than for the Dirichlet boundary condition problem. In Figure 6, the discretization error is small and our computed numerical solution matches the exact analytic solution.

MATLAB Solver	Number of Elements	Solver time	Relerr	Order
ODE45	$N = 8$	5.3959	7.230×10^{-2}	-
	$N = 16$	14.9935	1.750×10^{-2}	2.047
	$N = 32$	44.3118	4.100×10^{-3}	2.094
	$N = 64$	212.3113	8.205×10^{-4}	2.321
	$N = 128$	1866.10	1.823×10^{-4}	2.171
ODE23	$N = 8$	3.9322	6.990×10^{-2}	-
	$N = 16$	43.0017	1.520×10^{-2}	2.201
	$N = 32$	86.1277	2.000×10^{-3}	2.926
	$N = 64$	233.0048	1.600×10^{-3}	0.322
	$N = 128$	1058.80	2.300×10^{-3}	-0.524
ODE15s	$N = 8$	3.3187	7.820×10^{-2}	-
	$N = 16$	6.5455	2.320×10^{-2}	1.753
	$N = 32$	15.9258	9.800×10^{-3}	1.243
	$N = 64$	38.3093	6.500×10^{-3}	0.592
	$N = 128$	119.3554	5.700×10^{-3}	0.189

Table 2: FEM Results for $T = 10$, $Re = 100$ and Neumann-Dirichlet boundary conditions



(a) Exact solution



(b) FEM solution

Figure 1: Exact solution and FEM using ODE23 for $N = 32$; $Re = 100$ and Dirichlet boundary conditions

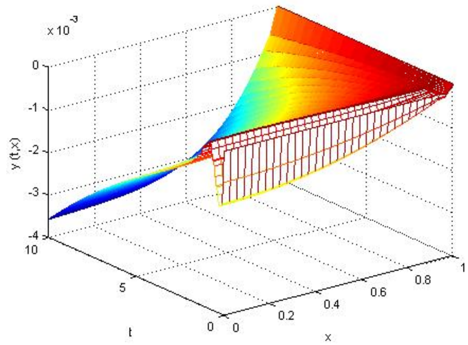


Figure 5: FEM error using ODE45 for $N = 32; Re = 100$ and Neumann-Dirichlet boundary conditions

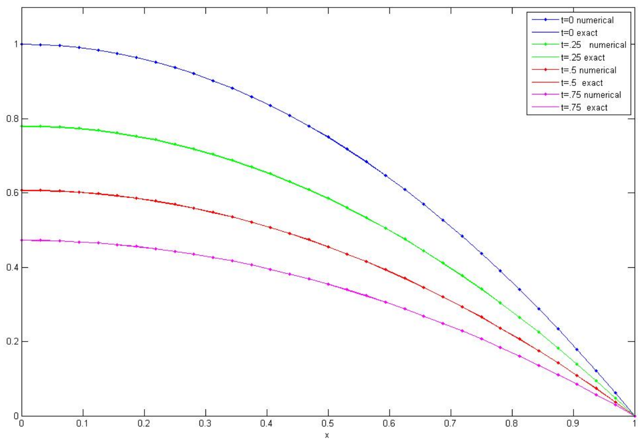


Figure 6: Cross section of exact and FEM using ODE45 for $N = 32$; $Re = 100$ and Neumann-Dirichlet boundary conditions

In conclusion, the motivation for this research is to demonstrate computational tools that can be used for design, optimization and control for PDE systems over a large parameter range. As outlined in the introduction, BBM-Burgers equation is a 2nd order nonlinear PDE model that approximates the dynamics of the Navier-Stokes system. Based on the successful previous results [21-23, 25] we used the GFE method which reduces the computational complexity of the non-linear term while maintaining accuracy. Thus, whenever applicable, we recommend the GFE over the standard FEM as a simplifying computational tool for high-fidelity simulations.

References

[1] B. Krämer. *Model Reduction of the Coupled Burgers Equation in Conservation Form.* PhD thesis, Virginia Polytechnic Institute and State University, 2011.

[2] C.A.J. Fletcher et al. *Computational galerkin methods*, volume 260. Springer-Verlag New York, 1984.

[3] D. H. Peregrine, *Calculations of the development of an undular bore*, J Fluid Mech 25 (1966), 321-326.

[4] D. N. Arnold, J. Douglas Jr., and V. Thome, *Superconvergence of finite element approximation to the solution of a Sobolev equation in a single space variable*, Math Comput 27 (1981), 737-743.

[5] D. Kaya and I. E. Inan, *Exact and numerical traveling wave solutions for nonlinear coupled equations using symbolic computation*, Appl Math Comput 151 (2004), 775-787.

[6] D. Kaya, *A numerical simulation of solitary-wave solutions of the generalized regularized long-wave equation*, Appl Math Comput 149 (2004), 833-841.

[7] H. Zhang, G. M. Wei, and Y. T. Gao, *On the general form of the Benjamin-Bona-Mahony equation in fluid mechanics*, Czech J Phys 52 (2002), 344-373.

[8] J. L. Bona and R. Smith, *The initial-value problem for the Korteweg-de Vries equation*, Philos Trans R Soc Lond A

[9] J.R. Singler. *Sensitivity analysis of partial difierential equations with applications to fluid flow.* PhD thesis, Citeseer, 2005.

[10] K. Omrani and M. Ayadi, *Finite difference discretization of the Benjamin-Bona-Mahony-Burgers(BBMB) equation*, Numer Meth Partial Diff Eq 24 (2008), 239-248.

[11] K. Omrani, *The convergence of the fully discrete Galerkin approximations for the Benjamin-Bona-Mahony (BBM) equation*, Appl Math Comput 180 (2006), 614-621.

[12] K. Al-Khaled, S. Momani, and A. Alawneh, *Approximate wave solutions for generalized Benjamin-Bona-Mahony-Burgers equations*, Appl Math Comp 171 (2005), 281-292.

[13] L. A. Medeiros and M. M. Miranda, *Weak solutions for a nonlinear dispersive equation*, J Math Anal Appl 59 (1977), 432-441.

[14] L. A. Medeiros and G. Perla Menzela, *Existence and uniqueness for periodic solutions of the Benjamin-Bona-Mahony equation*, SIAM J Math Anal 8 (1977), 792-799.

[15] L.Wahlbin, *Error estimates for a Galerkin method for a class of model equations for long waves*, Numer Math 23.

[16] L.C. Smith III. *Finite Element Approximations of BurgersEquation With RobinS Boundary Conditions.* PhD thesis, Virginia Polytechnic Institute and State University, 1997.

[17] M. Mei, *Large-time behavior of solution for generalized Benjamin-Bona-Mahony-Burgers Equations*, Nonlinear Anal

[18] P.J. Roache. *Code verification by the method of manufactured solutions.* Journal of Fluids Engineering, 124:4, 2002.

[19] R. K. Mohanty, *An $O(k^2 + h^2)$ finite difference method for one-space Burgers equation in polar coordinates*, Numer Meth Partial Diff Eq 12 (1996), 579-583.

[20] R. E. Mickens, *A finite difference scheme for travelling wave solutions to Burgers equation*, Numer Meth Partial Diff Eq 14 (1998), 815-820.

[21] R. Kannan and S. K. Chung, *Finite difference approximate solutions for the two-dimensional Burgerssystem*, Comput Math Appl 44 (2002), 193-200.

[22] R. E. Ewing, *Time-stepping Galerkin methods for nonlinear Sobolev partial differential equation*, IAMJ Numer Anal 15 (1978), 1125-1150.

[23] S.M. Pugh. *Finite element approximations of burgers' equation.* 1995.

[24] T. B. Benjamin, J. L. Bona, and J. J. Mahony, *Model equations for long waves in nonlinear dispersive systems*, Philos Trans R Soc Lond A 272 (1972), 47-78.

[25] T. Achouri, N. Khiari, and K. Omrani, *On the convergence of difference schemes for the Benjamin-Bona-Mahony (BBM) equation*, Appl Math Comput 182 (2006), 999-1005.

[26] V.Q. Nguyen. *A Numerical Study of Burgers Equation With Robin Boundary Conditions.* PhD thesis, Virginia Polytechnic Institute and State University, 2001.
[27] W.L. Oberkampf and C.J. Roy. *Verification and validation in scientific computing.* Cambridge Univ Pr, 2010.
[28] Y. Chen, B. Li, and H. Zhang, *Exact solutions of two nonlinear wave equations with simulation terms of any order*, Comm Nonlinear Sci Numer Simulation 10 (2005), 133-138.

Yang Ning
Department of Mathematics, College of Sciences, Yanbian University, Yanji 133002, China
E-mail address: fzhying@outlook.com

Mingzhe Sun
Department of Mathematics, College of Sciences, Yanbian University, Yanji 133002, China
E-mail address: mzsun@ybu.edu.cn

Guangri Piao
Department of Mathematics, College of Sciences, Yanbian University, Yanji 133002, China
E-mail address: grpiao@ybu.edu.cn