East Asian Math. J.
Vol. 33 (2017), No. 1, pp. 053–065
http://dx.doi.org/10.7858/eamj.2017.006

CUBIC B-SPLINE FINITE ELEMENT METHOD FOR THE ROSENAU-BURGERS EQUATION

Ge-Xing Xu, Chun-Hua Li, and Guang-Ri Piao*

Abstract. Numerical solutions of the Rosenau-Burgers equation based on the cubic B-spline finite element method are introduced. The backward Euler method is used for discretization in time, and the obtained nonlinear algebraic system is changed to a linear system by the Newton's method. We show that those methods are unconditionally stable. Two test problems are studied to demonstrate the accuracy of the proposed method. The computational results indicate that numerical solutions are in good agreement with exact solutions.

1. Introduction

Standard Lagrangian finite element basis functions offer only simple C^0-continuity, therefore they cannot be used for the spatial discretization of higher-order partial differential equations (e.g., the equation involving third-order or forth-order differential operators with regard to spatial variables). The B-spline basis function can, however, achieve C^1-continuity globally and such basis functions are often used to solve higher-order partial differential equations.

In study of dynamics of dense discrete systems, the cases of wave-wave and wave-wall interactions cannot be described by the well-known KDV equation. To overcome this shortcoming of the KDV equation, Rosenau [14, 15] proposed the following Rosenau equation

$$u_t + u_{xxxxt} + \gamma u_x + uu_x = f, \qquad x \in \Omega, \qquad t \in [0, T], \tag{1}$$

where $\Omega = [0, L]$, γ is a constant, and f is a forcing term. The existence and uniqueness of the solution to (1) were proved in [12], but it is still difficult to find the analytical solution to (1). For this reason, many works have been done on the numerical methods for solving (1)(see e.g., [1, 2, 5, 8] and also the

Received December 6, 2016; Accepted December 14, 2016.
2010 *Mathematics Subject Classification.* 35K30,74S05.
Key words and phrases. Cubic B-spline, Backward Euler method, Newton method, Crank-Nicolson difference scheme, Rosenau-Burgers equation.
* Corresponding author.
This work was financially supported by the National Natural Science Foundation of China (11461074).

(pISSN 1226-6973, eISSN 2287-2833)

references therein). On the other hand, with a further consideration about a nonlinear wave, a viscous term $-\nu u_{xx}$ needs to be included, and hence

$$u_t + u_{xxxxt} - \nu u_{xx} + \gamma u_x + u u_x = f, \tag{2}$$

with $\nu > 0$. This equation is usually called the Rosenau-Burgers equation, because its dissipative effect is the same as the Burgers' equation. The great number of works have been devoted to the Cauchy problem of the Rosenau-Burgers equation (see e.g., [6, 7, 9, 10]). Recently, the numerical solutions to an initial boundary value problem of the Rosenau-Burgers equation have been studied using finite difference schemes (see e.g., [3, 4, 11, 17] and also the references therein). In this paper, we suggest a cubic B-spline Galerkin finite element method for solving the Rosenau-Burgers equation (2) with boundary conditions

$$u(0,t) = u(L,t) = 0, \quad u_x(0,t) = u_x(L,t) = 0, \quad t \in [0,T] \tag{3}$$

and an initial condition

$$u(x,0) = u_0(x), \quad x \in [0,L]. \tag{4}$$

In this paper, we employ the Method of Manufactured Solutions(MMS) to ensure that the code produces accurate results and the approximations generated converge to a known solution. The MMS allows us to evaluate the error produced by numerical discretizations. The method develops exact solutions that are designed to test interactions in code and thereby verify the code.

The rest of this paper is organized as follows. In Section 2, we describe the B-spline finite element approximation of a solution to the Rosenau-Burgers equation. We analyze the stability of the proposed scheme in Section 3, and then present some numerical examples and their results in Section 4.

2. B-Spline Finite Element Approximation

Consider the Rosenau-Burgers equation with boundary conditions and an initial condition. We use a variational formulation to help to define a finite element method to approximate (1). A variational formulation of the problem (1) is as following: find $u \in L^2(0,T; H_0^2(\Omega))$ such that

$$\begin{cases} \displaystyle\int_\Omega u_t v dx + \int_\Omega u_{xxt} v'' dx + \nu \int_\Omega u_x v' dx + \gamma \int_\Omega u_x v dx \\ \displaystyle\qquad\qquad + \int_\Omega u u_x v dx = \int_\Omega f v dx \quad \text{for all } v \in H_0^2(\Omega), \\ u(0,x) = u_0(x) \quad \text{in } \Omega, \end{cases} \tag{5}$$

where $H_0^2(\Omega) = \{w \in H^2(\Omega) : w(0) = w(L) = 0, w_x(0) = w_x(L) = 0\}$ and $H^2(\Omega) = \{w \in L^2(\Omega) : v_x \in L^2(\Omega), v_{xx} \in L^2(\Omega)\}$. We write the first spatial derivative as "$\frac{d}{dx} ='$" and the second spatial derivative as "$\frac{d^2}{dx^2} =''$".

A typical finite element approximation of (2) is defined as follows: we first choose conforming finite element subspaces $V^h \subset H^2(\Omega)$ and then define $V_0^h = V^h \cap H_0^2(\Omega)$. One then seeks $u^h(t,\cdot) \in V_0^h$ such that

$$\begin{cases} \displaystyle\int_\Omega u_t^h v^h dx + \int_\Omega u_{xxt}^h (v^h)'' dx + \nu \int_\Omega u_x^h (v^h)' dx + \gamma \int_\Omega u_x^h v^h dx \\ \qquad\qquad \displaystyle + \int_\Omega u^h u_x^h v^h dx = \int_\Omega f v^h dx \quad \text{for all } v^h \in V_0^h(\Omega), \\ u^h(0,x) = u_0^h(x) \quad \text{in } \Omega, \end{cases} \tag{6}$$

where $u_0^h(x) \in V_0^h$ is an approximation, e.g., a projection, of $u_0(x)$.

Let the interval $\Omega = [0, L]$ be divided into N finite elements with an equal length h and x_i denote the knots such that $0 = x_0 < x_1 < \cdots < x_N = L$. The set of splines $\{B_{-1}, B_0, B_1 \cdots, B_N, B_{N+1}\}$ forms a basis for functions defined on Ω. Cubic B-splines $B_i(x)$ with the required properties are defined by [13]

$$B_i(x) = \frac{1}{h^3} \begin{cases} (x - x_{i-2})^3, & x \in [x_{i-2}, x_{i-1}), \\ h^3 + 3h^2(x - x_{i-1}) + 3h(x - x_{i-1})^2 - & \\ 3(x - x_{i-1})^3, & x \in [x_{i-1}, x_i), \\ h^3 + 3h^2(x - x_{i+1}) + 3h(x - x_{i+1})^2 + & \\ 3(x - x_{i+1})^3, & x \in [x_i, x_{i+1}), \\ (x_{i+2} - x)^3, & x \in [x_{i+1}, x_{i+2}), \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $h = x_{i+1} - x_i$ and $i =, -1, 0, \cdots, N, N+1$.

Each cubic B-spline covers four elements, or equivalently each element is covered by four cubic B-splines. The values of $B_i(x)$ and their derivatives can be tabulated as in Table 1.

Table 1: Values of cubic B-splines and its derivatives at knots

x	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
$B_i(x)$	0	1	4	1	0
$B_i^{'}(x)$	0	$3/h$	0	$-3/h$	0
$B_i^{''}(x)$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0

Our numerical treatment for solving Equation (2) by the cubic B-spline finite elements with the backward Euler-Newton's methods as a time marching method finds an approximate solution $u^h(x,t)$ in the form

$$u^h(x,t) = \sum_{i=-1}^{N+1} \alpha_i(t) B_i(x), \tag{8}$$

where $\alpha_i(t)$ are unknown time-dependent quantities to be determined from (6).

Using cubic B-spline functions at (7) and an approximate function at (8), the approximate values of $u^h(x,t)$ and the first and second derivatives of $u^h(x,t)$ at the knots (or nodes) are determined in terms of the time parameters α_i as follows. For the sake of simplicity, we assume $u_i = u^h(x_i,t)$, then

$$\begin{cases} u_i = \alpha_{i-1} + 4\alpha_i + \alpha_{i+1}, \\ u_i' = 3(\alpha_{i+1} - \alpha_{i-1})/h, \\ u_i'' = 6(\alpha_{i-1} - 2\alpha_i + \alpha_{i+1})/h^2. \end{cases} \tag{9}$$

From (9) and the boundary conditions (3), we obtain

$$\begin{cases} \alpha_{-1} = \alpha_1, \quad \alpha_0 = -\alpha_1/2, \\ \alpha_{N+1} = \alpha_{N-1}, \quad \alpha_N = -\alpha_{N-1}/2. \end{cases} \tag{10}$$

Using (10) in (8), we have

$$u^h(x,t) = \sum_{i=1}^{N-1} \alpha_i(t)\tilde{B}_i(x), \tag{11}$$

where

$$\begin{cases} \tilde{B}_1(x) = [2B_{-1}(x) - B_0(x) + 2B_1(x)]/2, \\ \tilde{B}_i(x) = B_i(x), \qquad i = 2, \cdots, N-2, \\ \tilde{B}_{N-1}(x) = [2B_{N-1}(x) - B_N(x) + 2B_{N+1}(x)]/2. \end{cases}$$

A differential equation discretized by the finite element method is not expressed in terms of the nodal parameters u_i, u_i' but the element parameters α_i. Thus, we shall not determine the nodal values directly as the case of usual finite element formulations, however they can always be recovered by (9).

According to the Galerkin method, the test function $v^h(x)$ in (6) is chosen to be $v_i^h(x) = \tilde{B}_i(x)(i = 1, 2, \cdots, N-1)$. Substituting (11) into (6), we obtain

$$\begin{cases} \displaystyle\sum_{i=1}^{N-1}\left(\int_\Omega \tilde{B}_i\tilde{B}_j dx\right)\frac{d\alpha_i(t)}{dt} + \sum_{i=1}^{N-1}\left(\int_\Omega \tilde{B}_i''\tilde{B}_j'' dx\right)\frac{d\alpha_i(t)}{dt} \\ \displaystyle + \nu\sum_{i=1}^{N-1}\left(\int_\Omega \tilde{B}_i'\tilde{B}_j' dx\right)\alpha_i(t) + \gamma\sum_{i=1}^{N-1}\left(\int_\Omega \tilde{B}_i'\tilde{B}_j dx\right)\alpha_i(t) \\ \displaystyle + \sum_{i=1}^{N-1}\sum_{k=1}^{N-1}\left(\int_\Omega \tilde{B}_i\tilde{B}_k'\tilde{B}_j dx\right)\alpha_i(t)\alpha_k(t) = \int_\Omega f\tilde{B}_j dx, \\ \displaystyle\sum_{i=1}^{N-1}\left(\int_\Omega \tilde{B}_i\tilde{B}_j dx\right)\alpha_i(0) = \int_\Omega u_0(x)\tilde{B}_j dx, \quad j = 1, 2, \cdots, N-1, \end{cases}$$

which can be written in matrix form as

$$\begin{cases} (\mathbb{M}+\mathbb{D})\dfrac{d\alpha}{dt}+(\nu\mathbb{S}+\gamma\mathbb{C})\alpha+\alpha^T\mathbb{N}\alpha=\mathbf{f}, \\ \mathbb{M}\alpha^0=\mathbf{u}^0, \end{cases} \tag{12}$$

where $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_{N-1})^T$ and $\alpha^0=(\alpha_1^0,\alpha_2^0,\cdots,\alpha_{N-1}^0)^T$. Elements (or components) of the $(N-1)\times(N-1)$ matrices $\mathbb{M},\mathbb{D},\mathbb{S},\mathbb{C}$, the $(N-1)\times(N-1)\times(N-1)$ tensor $\mathbb{N}$, and vectors $\mathbf{f}$ and $\mathbf{u}_0$ are given by

$$\begin{cases} \mathbb{M}_{ij}=\displaystyle\int_\Omega \tilde{B}_i\tilde{B}_j dx, \quad \mathbb{D}_{ij}=\int_\Omega \tilde{B}_i^{''}\tilde{B}_j^{''}dx, \mathbb{S}_{ij}=\int_\Omega \tilde{B}_i^{'}\tilde{B}_j^{'}dx, \\ \mathbb{C}_{ij}=\displaystyle\int_\Omega \tilde{B}_i^{'}\tilde{B}_j dx, \quad \mathbb{N}_{ijk}=\int_\Omega \tilde{B}_i\tilde{B}_k^{'}\tilde{B}_j dx, \quad f_j=\int_\Omega f\tilde{B}_j dx, \\ u_j^0=\displaystyle\int_\Omega u_0(x)\tilde{B}_j dx, \qquad i,j,k=1,2,\cdots,N-1. \end{cases}$$

The system of nonlinear ordinary differential equations (12) consists of $N-1$ equations and $N-1$ unknowns. An associated $(N-1)\times(N-1)$ matrix $\mathbb{G}$ is given by

$$\mathbb{G}_{ij}=\sum_{k=1}^{N-1}\mathbb{N}_{ijk}\alpha_k,$$

which depends on the parameters α and will be used in the following theoretical discussions.

We can determine the matrices $\mathbb{M},\mathbb{D},\mathbb{S},\mathbb{C}$, and $\mathbb{G}$ from (7) algebraically, which have a septadiagonal form. This implies that the general row for each matrix has the following form:

$$\begin{cases} \mathbb{M}:(h/140)(1,120,1191,2416,1191,120,1), \\ \mathbb{D}:(1/h^3)(6,0,-54,96,-54,0,6) \\ \mathbb{S}:(-1/10h)(3,72,45,-240,45,72,3) \\ \mathbb{C}:(1/100)(5,280,1225,0,-1225,-280,-5) \\ \mathbb{G}:(1/840)(-(5,108,129,10,0,0,0)\alpha,-(21,1944,8130,3888,129, \\ \qquad 0,0)\alpha,-(-21,0,17841,35682,8130,108,0)\alpha,(5,1944,17841,0, \\ \qquad -17841,-1944,-5)\alpha,(0,108,8130,35682,17841,0,-21)\alpha, \\ \qquad (0,0,129,3888,8130,1944,21)\alpha,(0,0,0,10,129,108,5)\alpha), \end{cases} \tag{13}$$

where $\alpha=(\alpha_{i-3},\alpha_{i-2},\alpha_{i-1},\alpha_i,\alpha_{i+1},\alpha_{i+2},\alpha_{i+3})$ for the ith row. The matrices $\mathbb{M},\mathbb{D}$, and $\mathbb{S}$ are symmetric and the matrix $\mathbb{C}$ is skew symmetric. The matrix $\mathbb{G}$ has a relatively more complex structure.

As mentioned before, we solve the system (12) by using the following Euler and Newton's methods specifically.

(1) The interval $[0,T]$ is divided into M subintervals with length $\triangle t = T/M$, where T is a total time and M is chosen as a positive integer.
(2) Suppose α^n are parameters at time $t_n = n\triangle t$, then according to the backward Euler method, (12) can be written as

$$\begin{aligned}\mathcal{J}(\alpha^n) = (\mathbb{M}+\mathbb{D}+\triangle t\nu\mathbb{S}+\triangle t\gamma\mathbb{C})\alpha^n + \triangle t\mathbb{G}\alpha^n \\ -(\mathbb{M}+\mathbb{D})\alpha^{n-1} - \triangle t\mathbf{f}^n = 0, \qquad n = 1,2,\cdots,M,\end{aligned} \tag{14}$$

where $\mathbf{f}^n$ is a given vector. The matrices $\mathbb{M},\mathbb{D},\mathbb{S}$, and $\mathbb{C}$ are independent of time, hence they will remain constants throughout the calculations. While the matrix $\mathbb{G}$ is dependent on time, therefore it must be recalculated at each time step. For the sake of convenience, the function $\mathcal{J}$ is introduced.
(3) The Newton's method is employed to linearize the nonlinear algebraic system (14). We observe that

$$\mathcal{J}'(\alpha^{n,l-1})(\alpha^{n,l}-\alpha^{n,l-1}) = -\mathcal{J}(\alpha^{n,l-1}), \tag{15}$$

where the derivative of function $\mathcal{J}$ is given by

$$\mathcal{J}'(\cdot) = \mathbb{M}+\mathbb{D}+\triangle t\nu\mathbb{S}+\triangle t\gamma\mathbb{C}+\triangle t\tilde{\mathbb{G}}(\cdot),$$

and the index l is an inner iteration number at each time step n. The matrix $\tilde{\mathbb{G}}$ is a derivative of nonlinear term $(\alpha^n)^T\mathbb{N}\alpha^n(=\mathbb{G}\alpha^n)$ with respect to α^n.

The time evolution of the approximate solution $u^h(x,t)$ is determined by the time evolution of the vector α^n. This is found by repeatedly solving the system (15), once the initial vector $\alpha^0 = \mathbb{M}^{-1}\mathbf{u}^0$ has been computed from the initial conditions. The concrete solving process at one time step $[t_{n-1},t_n]$ is as follows:

(1) When $l=1$, for the initial step of the inner iteration, set $\alpha^{n,0} \leftarrow \alpha^{n-1}$, and calculate $\alpha^{n,1}$ from (15).
(2) When the other $l = 2,3,\cdots$, compute $\alpha^{n,l}$ by using (15).
(3) If $\|\alpha^{n,l}-\alpha^{n,l-1}\| < tolerence$, quit; otherwise, go back to Step 2.

It is not difficult to find that the backward Euler method is applied to time discretization process, but the Newton method is used only for linearization of the nonlinear algebraic equation (14) or update the α included in the $\mathbb{G}$ within each time step. Thus, we only use the formula (14) rather than employing (15) in the following stability analysis.

3. The Stability Analysis

An investigation into stability of the numerical scheme (14) is based on the Von Neumann theory. We define the growth factor of a typical Fourier mode as

$$\alpha_m^n = \hat{\alpha}^n e^{imkh}, \tag{16}$$

where k is a mode number and h is an element size, which is required for a linearization of the numerical scheme (14).

In this linearization, we assume that the quantity u in the nonlinear term uu_x is a local constant. This is equivalent to assuming that the corresponding values of α_j are also constant and equal to d.

The stability of time discretization for linear evolution equations can be expressed in terms of stability for the case as the right-hand side and boundary data are zero. Therefore, we replace (14) with the following system

$$(\mathbb{M}+\mathbb{D}+\triangle t\nu\mathbb{S}+\triangle t\gamma\mathbb{C}+\triangle t\mathbb{G})\alpha^n=(\mathbb{M}+\mathbb{D})\alpha^{n-1}, \tag{17}$$

and then make the stability analysis.

A linearized recurrence relationship corresponding to (17) is then given by

$$\begin{aligned}&k_1\alpha_{j-3}^n+k_2\alpha_{j-2}^n+k_3\alpha_{j-1}^n+k_4\alpha_j^n+k_5\alpha_{j+1}^n+k_6\alpha_{j+2}^n+k_7\alpha_{j+3}^n=\\&l_1\alpha_{j-3}^{n-1}+l_2\alpha_{j-2}^{n-1}+l_3\alpha_{j-1}^{n-1}+l_4\alpha_j^{n-1}+l_3\alpha_{j+1}^{n-1}+l_2\alpha_{j+2}^{n-1}+l_1\alpha_{j+3}^{n-1},\end{aligned} \tag{18}$$

where

$$\begin{aligned}&k_1=l_1-r_1+r_2-r_3,\quad k_2=l_2-24r_1+56r_2-56r_3,\\&k_3=l_3-15r_1+245r_2-245r_3,\quad k_4=l_4+80r_1,\\&k_5=l_3-15r_1-245r_2+245r_3,\quad k_6=l_2-24r_1-56r_2+56r_3,\\&k_7=l_1-r_1-r_2+r_3,\\&l_1=\frac{h}{140}+\frac{6}{h^3},\quad l_2=\frac{6h}{7},\quad l_3=\frac{1191h}{140}-\frac{54}{h^3},\quad l_4=\frac{604h}{35}+\frac{96}{h^3},\\&r_1=\frac{3\nu\triangle t}{10h},\quad r_2=\frac{\gamma\triangle t}{20},\quad r_3=\frac{3d\triangle t}{10}.\end{aligned}$$

Substituting (16) into (18), we obtain

$$(a+b+ic)\hat{\alpha}^n=a\hat{\alpha}^{n-1}, \tag{19}$$

where

$$\begin{aligned}&i=\sqrt{-1},\\&a=l_1\cos3kh+l_2\cos2kh+l_3\cos kh+\frac{l_4}{2},\\&b=r_1(40-\cos3kh-24\cos2kh-15\cos kh),\\&c=(r_3-r_2)(\sin3kh+56\sin2kh+245\sin kh).\end{aligned} \tag{20}$$

Let us write $\hat{\alpha}^n=g\hat{\alpha}^{n-1}$ and substitute it into (18), which gives

$$g=\frac{a}{a+b+ic},$$

where g is the growth factor for the mode. Since a,b are greater than zero, the modulus of the growth factor is

$$|g|=\sqrt{g\bar{g}}=\sqrt{\frac{a^2}{(a+b)^2+c^2}}<1.$$

Therefore, the linearized scheme is unconditionally stable.

4. Computational Experiments and Conclusions

We consider numerical solutions of the Rosenau-Burgers equation for two test problems now. To measure the accuracy of the numerical algorithm, we compute the difference between analytic solutions and numerical solutions at each mesh points on specified time steps. We also compute the discrete L_2- and $L_\infty-$ error norms in Example 1. These error norms are defined as

$$L_2 = \|u^e - u^h\|_2 = [h\sum_{j=0}^{N}(u_j^e - u_j^h)^2]^{\frac{1}{2}},$$
$$L_\infty = \|u^e - u^h\|_\infty = \max_j |u_j^e - u_j^h|,$$

where h is a spatial step size, u^e is an analytic solution, and u^h is an numerical solution.

In order to show the accuracy of the present method more clearly, we construct the exact solution to the first example.

Example 1. When the parameters $\nu = 1, \gamma = 1$, we consider the problem with the boundary conditions (3) and exact solution given by

$$u(x,t) = 4e^{-t}x(1-x)\sin(\pi x). \tag{21}$$

The initial condition is given by

$$u_0(x) = 4x(1-x)\sin(\pi x).$$

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

$$f(x,t) = 4x(1-x)\sin(\pi x).$$

When implementing a numerical computation, we take $L = 1$, and a forcing term f is chosen by substituting (21) into (2). Figure 1 shows us that the behavior of the numerical solution from $t = 0$ to 4. The numerical and exact solutions are given in Table 2. The results of error norms given in Tables 3, 4, 5, and 6 present the changes of two error norms L_2 and L_∞ at different times when distinct spatial steps and time steps are taken. Table 4 shows us a good accuracy of the numerical solution for small time step and appropriate spatial step. In particular, we observe that the error and time step rates are constant in Tables 4, 5, and 6. The characteristics are consistent with those computed by using Euler method. From Tables 7 and 8, we find that the error obtained by the present algorithm is smaller than the error computed by Crank-Nicolson difference method [16].

To illustrate the effectiveness of the present method nicely, we study more general cases in the second example.

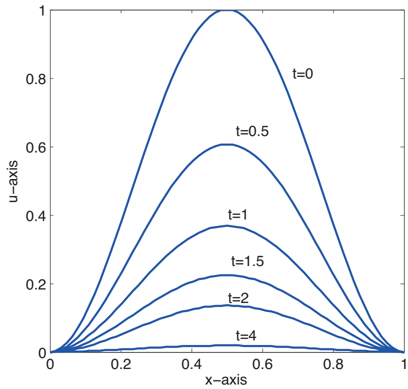

FIGURE 1. The behavior of the numerical solution with $h = 1/64$, $\triangle t = 0.001$ at different times for Example 1.

Example 2. In the equation (2), we set the forcing term f be zero for the following several cases.

(i) Consider the case with $\nu = 0.01, \gamma = 80, \triangle t = 0.01$ and initial condition $u_0(x) = \frac{4x(1-x)}{1+exp(2x^2)}$, at different times $t = 3$ and 7. The numerical solutions are given in Table 9, with distinct spatial step sizes h.

(ii) Figures 2, 3, and 4 show the behavior of the numerical solutions for various constant coefficients ν and γ, diverse parameters h and $\triangle t$, different times, and distinct initial conditions $u_0(x) = 4x(x-1)\sin(\pi x)$, $u_0(x) = 4x(1-x)\sin(2\pi x)$ and $u_0(x) = \frac{4x(1-x)}{1+exp(0.25x^2)}$ respectively.

We see that the algorithm proposed here by using Galerkin's method with cubic spline shape functions gives results in a good accuracy. The L_2- and $L_\infty-$error norms keep satisfactorily small during the simulations. A linear stability analysis based on the Von Neumann theory shows that the numerical scheme is unconditionally stable. We conclude that a finite element approach based on Galerkin's method with cubic spline shape functions is eminently suitable for the computation of solutions to the Rosenau-Burgers equation. We think that this approach is a proper method for other applications where the continuity of derivatives is essential.

Table 2: Comparison between exact solution and numerical solution at different times with $h = 1/64$ and $\triangle t = 0.001$ for Example 1.

x	$t = 0.5$ Exact	$t = 0.5$ Numer	$t = 2$ Exact	$t = 2$ Numer	$t = 4$ Exact	$t = 4$ Numer
0.125	0.10154	0.10157	0.02266	0.02273	0.00307	0.00314
0.25	0.32166	0.32176	0.07177	0.07199	0.00971	0.00995
0.375	0.52533	0.52550	0.11722	0.11757	0.01586	0.01624
0.5	0.60653	0.60671	0.13534	0.13574	0.01832	0.01875
0.625	0.52533	0.52550	0.11722	0.11756	0.01586	0.01624
0.75	0.32166	0.32176	0.07177	0.07198	0.00971	0.00994
0.875	0.10154	0.10157	0.02266	0.02272	0.00307	0.00313

Table 3: Error norms at different times for $h = 1/10$ and $\triangle t = 0.05$ for Example 1.

	$t = 2$	$t = 2.5$	$t = 3$	$t = 4$	$t = 5$
$L_2 \times 10^2$	1.123	1.176	1.208	1.214	1.217
$L_\infty \times 10^2$	1.824	1.915	1.962	1.986	1.994

Table 4: Error norms at different times for $h = 1/32$ and $\triangle t = 0.001$ for Example 1.

	$t = 1$	$t = 1.5$	$t = 2$	$t = 3$	$t = 4$
$L_2 \times 10^4$	1.344	1.636	1.803	1.938	2.716
$L_\infty \times 10^4$	2.264	2.955	3.038	3.266	3.296

Table 5: Error norms at different times for $h = 1/32$ and $\triangle t = 0.01$ for Example 1.

	$t = 1$	$t = 1.5$	$t = 2$	$t = 3$	$t = 5$
$L_2 \times 10^3$	1.879	2.289	2.524	2.716	2.710
$L_\infty \times 10^3$	3.021	3.681	4.058	4.365	4.353

Table 6: Error norms at different times for $h = 1/32$ and $\triangle t = 0.1$ for Example 1.

	$t = 1$	$t = 1.5$	$t = 2$	$t = 3$	$t = 5$
$L_2 \times 10^2$	1.905	2.321	2.559	2.754	2.748
$L_\infty \times 10^2$	3.051	3.717	4.097	4.409	4.397

Table 7: Comparison of error norms $L_2 \times 10^3$ at different times with $h = 1/32$ and $\triangle t = 0.01$ for Example 1.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Present	1.878	2.507	2.694	2.722	2.691
C-N scheme	1.879	2.524	2.716	2.743	2.710

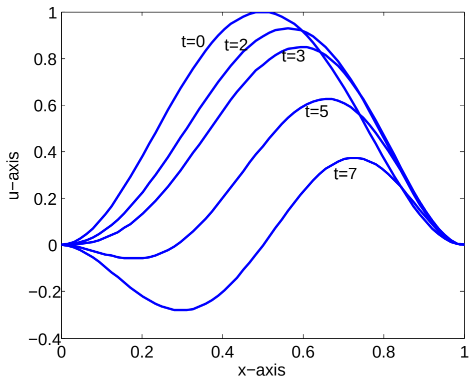

FIGURE 2. The behavior of the numerical solution with $\nu = 0.1, \gamma = 90$ and $h = 1/64$, $\triangle t = 0.01$ at different times for Example 2.

Table 8: Comparison of error norms $L_\infty \times 10^3$ at different times with $h = 1/32$ and $\triangle t = 0.01$ for Example 1.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Present	3.021	4.058	4.365	4.407	4.353
C-N scheme	3.131	4.146	4.441	4.481	4.429

Table 9: Comparison of results at different times with $\nu = 0.01$, $\gamma = 80$ and $\triangle t = 0.01$ for distinct spatial step size h, Example 2.

x	$t = 3$ $h = \frac{1}{16}$	$t = 3$ $h = \frac{1}{32}$	$t = 3$ $h = \frac{1}{64}$	$t = 7$ $h = \frac{1}{16}$	$t = 7$ $h = \frac{1}{32}$	$t = 7$ $h = \frac{1}{64}$
0.125	0.20333	0.20619	0.20669	0.15516	0.15818	0.15870
0.25	0.35005	0.35117	0.35123	0.24840	0.24979	0.24989
0.375	0.44038	0.44082	0.44084	0.35035	0.35115	0.35121
0.5	0.46160	0.46182	0.46184	0.44715	0.44775	0.44782
0.625	0.40209	0.40227	0.40228	0.47387	0.47437	0.47443
0.75	0.27158	0.27186	0.27187	0.37315	0.37363	0.37367
0.875	0.11228	0.11295	0.11306	0.16464	0.16535	0.16548

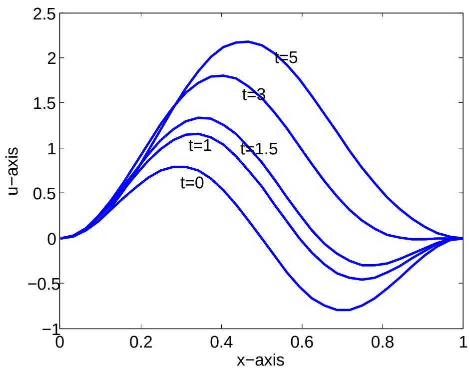

FIGURE 3. The behavior of the numerical solution with $\nu = 0.0001, \gamma = 100$ and $h = 1/32$, $\triangle t = 0.1$ at different times for Example 2.

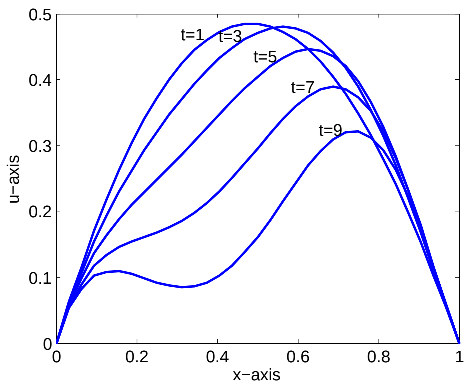

FIGURE 4. The behavior of the numerical solution with $\nu = 0.01, \gamma = 80$ and $h = 1/64$, $\triangle t = 0.01$ at different times for Example 2.

References

[1] S.K. Chung, Finite difference approximate solutions for the Rosenau equation, Appl. Anal. 69 (1.2) (1998) 149-156.

[2] S.K. Chung, A.K. Pani, Numerical methods for the Rosenau equation, Appl. Anal. 77 (2001) 351-369.

[3] B. Hu, Y.C. Xu, J.S. Hu, Crank-Nicolson finite difference scheme for the Rosenau-Burgers equation, Appl. Math. Comput. 204 (2008) 311-316.

[4] J.S. Hu, B. Hu, Y.C. Xu, Average implicit linear difference scheme for generalized Rosenau-Burgers equation, Appl. Math. Comput. 217 (2011) 7557-7563.

[5] Y.D. Kim, H.Y. Lee, The convergence of finite element Galerkin solution for the Rosenau equation, Korean J. Comput. Appl. Math. 5 (1998) 171-180.

[6] L. Liu, M. Mei, A better asymptotic profile of Rosenau-Burgers equation, Appl. Math. Comput. 131 (2002) 147-170.
[7] L. Liu, M. Mei, Y.S. Wong, Asymptotic behavior of solutions to the Rosenau-Burgers equation with a periodic initial boundary, Nonlinear Anal. 67 (2007) 2527-2539.
[8] S.A. Manickam, A.K. Pani, S.K. Chung, A second-order splitting combined with orthogonal cubic spline collocation method for the Rosenau equation, Numer. Meth. Partial Diff. Eq. 14 (1998) 695-716.
[9] M. Mei, Long-time behavior of solution for Rosenau-Burgers equation (I), Appl. Anal. 63 (1996) 315-330.
[10] M. Mei, Long-time behavior of solution for Rosenau-Burgers equation (II), Appl. Anal. 68 (1998) 333-356.
[11] X.T. Pan, L.M. Zhang, A new finite difference scheme for the Rosenau-Burgers equation, Appl. Math. Comput. 218 (2012) 8917-8924.
[12] M.A. Park, On the Rosenau equation, Math. Appl. Comput. 9 (1990) 145-152
[13] P.M. Prenter, Splines and variational methods, John Wiley and Sons, New York, 1975.
[14] P. Rosenau, A quasi-continuous description of a nonlinear transmission line, Phys. Scripta 34 (1986) 827-829.
[15] P. Rosenau, Dynamics of dense discrete systems, Progr. Theor. Phys. 79 (1988) 1028-1042.
[16] A.A. Soliman, A Galerkin solution for Burgers equation using Cubic B-spline finite elements. Abstr. Appl. Anal. 2012 (2012). Hindawi Publishing Corporation.
[17] G.Y. Xue, L.M. Zhang, A new finite difference scheme for generalized Roseau-Burgers equation. Appl. Math. Comput. 222 (2013) 490-496.

Ge-Xing Xu
Department of Mathematics, Yanbian University, Yanji 133002, China
E-mail address: 785037520@qq.com

Chun-Hua Li
Department of Mathematics, Yanbian University, Yanji 133002, China
E-mail address: sxlch@ybu.edu.cn

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