Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 5, No. 4, 2017, pp. 301-309

An approximation to the solution of Benjamin-Bona-Mahony-Burgers equation

Mohammad Zarebnia*
Department of Mathematics and Applications,
University of Mohaghegh Ardabili, 56199-11376, Ardabil, Iran.
E-mail: zarebnia@uma.ac.ir

Maryam Aghili
Department of Mathematics and Applications,
University of Mohaghegh Ardabili, 56199-11376, Ardabil, Iran.
E-mail: mina.aghili66@yahoo.com

Abstract In this paper, numerical solution of the Benjamin-Bona-Mahony-Burgers(BBMB) equation is obtained by using the mesh-free method based on the collocation method with radial basis functions(RBFs). Stability analysis of the method is discussed. The method is applied to several examples and accuracy of the method is tested in terms of L_2 and L_∞ error norms.

Keywords. Radial basis functions, Mesh-free method, BBMB equation, Stability.

2010 Mathematics Subject Classification. 35B35, 41A30, 65N35.

1. Introduction

The mathematical model of propagation of small-amplitude long waves in nonlinear dispersive media is described by the following Benjamin-Bona-Malony-Burgers (BBMB) equation:

$$u_t - u_{xxt} - \alpha u_{xx} + \beta u_x + \gamma u u_x = 0,\ (x,t) \in \Omega \times (0,T], \tag{1.1}$$

with the boundary conditions:

$$u(a,t) = g_1(t), \quad u(b,t) = g_2(t),\ t \in (0,T], \tag{1.2}$$

and initial condition

$$u(x,0) = f(x),\ x \in \Omega. \tag{1.3}$$

where $\Omega = (0,1)$ and $\alpha, \beta, \gamma > 0$. The BBMB problem has been numerically tackled and investigated by many authors. A spline collocation method for approximating the solution of (1.1) can be found in Manickam et al.[8]. A mesh-free method based on radial basis function will discussed in this paper for finding approximate solution of BBMB equation. This method introduced by Hardy in 1971 [6]. Kansa [5] in 1990 used modified MQ scheme to solve partial differential equations. So, Frank in 1982, observed radial basis function is better than all other methods regarding efficiency,

Received: 17 January 2017 ; Accepted: 24 September 2017.
∗ Corresponding author.

stability, uniqueness. Also convergence of the method was discussed by Michelson [10], Michelli [9] and Frank [4].
We apply this method by RBFs such as: $\phi(r) = \sqrt{r^2+c^2}(MQ)$, $\phi(r) = 1/(r^2 + c^2)(IQ)$, $\phi(r) = 1/\sqrt{r^2+c^2}(IMQ)$ and $\phi(r) = exp(-c^2r^2)(GA)$ to obtain solution of BBMB equation.
The layout of this paper is as follows. In section 2, we will illustrate how the RBFs method may be used to Equation (1.1) by an explicit system of algebraic equations. Section 3, is devoted to stability analysis of the method. In section 4, several examples are solved and accuracy of numerical scheme is tested. In section 5, we conclude our results.

2. Structure of the method

We consider the BBMB equation

$$u_t - u_{xxt} - \alpha u_{xx} + \beta u_x + \gamma u u_x = 0, \qquad a \le x \le b, \quad t \ge 0, \tag{2.1}$$

with the initial and boundary conditions

$$u(x,0) = f(x), \qquad a < x < b, \tag{2.2}$$

(2.3)

$$u(a,t) = g_1(t), \quad u(b,t) = g_2(t), \qquad t \ge 0. \tag{2.4}$$

By applying Crank-Nicolson scheme to Equation (2.1) we obtain:

$$\begin{aligned}&[\frac{u^{n+1}-u^n}{\partial t}] - [\frac{(u_{xx})^{n+1}-(u_{xx})^n}{\partial t}] - \alpha[\frac{(u_{xx})^{n+1}+(u_{xx})^n}{2}]\\ &+\beta[\frac{(u_x)^{n+1}+(u_x)^n}{2}] + \gamma[\frac{(uu_x)^{n+1}+(uu_x)^n}{2}] = 0.\end{aligned} \tag{2.5}$$

Where $u^{n+1} = u(x, t^{n+1})$, $t^{n+1} = t^n + \delta t$ and δt is the time step.
The term $(uu_x)^{n+1}$ in above equation shows that Equation (2.5) is nonlinear. For linearization this term we apply following formula:

$$(uu_x)^{n+1} \approx u^n u_x^{n+1} + u^{n+1}u_x^n - u^n u_x^n. \tag{2.6}$$

Substituting Equation (2.6) in Equation (2.5) we will have

$$\begin{aligned}u^{n+1} - u_{xx}^{n+1} + \frac{\delta t}{2}[-\alpha(u_{xx})^{n+1} + \beta(u_x)^{n+1} + \gamma(u)^{n+1}u_x^n + \gamma(u)^n u_x^{n+1}]\\ = u^n - u_{xx}^n - \frac{\delta t}{2}[-\alpha(u_{xx})^n + \beta(u_x)^n],\end{aligned} \tag{2.7}$$

where u^n is approximate solution in nth time step. Let us approximate the solution of Equation (2.1) by:

$$u^n(x_i) = \sum_{j=0}^{N} \lambda_j^n \varphi(r_{ij}), \tag{2.8}$$

where ϕ is radial basis function. $r_{ij} = \|x_i - x_j\|$ is Euclidian distance, $x_j = a + j\delta x$, $j = 0(1)N$ are centers in [a,b], and $x_i = a + i\delta x$ are collocation points in [a,b]. Using Equation (2.7) and (2.8), for $x_i, i = 1(1)N$ we get the following equation:

$$\begin{aligned}
&\sum_{j=0}^{N}\lambda_j^{n+1}\varphi(r_{ij}) - \sum_{j=0}^{N}\lambda_j^{n+1}\varphi^{''}(r_{ij}) + \frac{\delta t}{2}[-\alpha\sum_{j=0}^{N}\lambda_j^{n+1}\varphi^{''}(r_{ij}) + \beta\sum_{j=0}^{N}\lambda_j^{n+1}\varphi^{'}(r_{ij}) \\
&\qquad +\gamma\sum_{j=0}^{N}\lambda_j^{n+1}\varphi(r_{ij})\sum_{j=0}^{N}\lambda_j^{n}\varphi^{'}(r_{ij}) + \gamma\sum_{j=0}^{N}\lambda_j^{n}\varphi(r_{ij})\sum_{j=0}^{N}\lambda_j^{n+1}\varphi^{'}(r_{ij})] \\
&= \sum_{j=0}^{N}\lambda_j^{n}\varphi(r_{ij}) - \sum_{j=0}^{N}\lambda_j^{n}\varphi^{''}(r_{ij}) - \frac{\delta t}{2}[-\alpha\sum_{j=0}^{N}\lambda_j^{n}\varphi^{''}(r_{ij}) + \beta\sum_{j=0}^{N}\lambda_j^{n}\varphi^{'}(r_{ij})], \quad (2.9)
\end{aligned}$$

where $\varphi^{'}(r_{ij}) = \frac{\partial}{\partial x}\varphi(\|x - x_j\|)|_{x=x_i}$ and $\varphi^{''}(r_{ij}) = \frac{\partial^2}{\partial x^2}\varphi(\|x - x_j\|)|_{x=x_i}$ and $i = 1(1)N-1$. Also, from Equation (2.8) and (2.4) we obtain following equation for the boundary conditions,

$$\sum_{j=0}^{N}\lambda_j^{n+1}\varphi(r_{0j}) = g_1(t), \quad \sum_{j=0}^{N}\lambda_j^{n+1}\varphi(r_{Nj}) = g_2(t). \tag{2.10}$$

The system (2.9) and (2.10) contain $N+1$ equations and $N+1$ unknowns λ_j^{n+1} which can be obtain by Gaussian elimination method. First, we find value of u^0 from initial condition and then determine value of λ_j^0 from Equation (2.8). Matrix form of this system can be written as:

$$\begin{aligned}
&[A_1 - D_2 + \frac{\delta t}{2}[-\alpha D_2 + \beta D_1 + \gamma(A_2 * u_x^n + u^n * D_1)]]\lambda_j^{n+1} \\
&\qquad = [A_2 - D_2 - \frac{\delta t}{2}(-\alpha D_2 + \beta D_1)]\lambda_j^n + G^{n+1}, \quad (2.11)
\end{aligned}$$

where $A_1 = [\varphi(r_{ij})]_{i,j=0}^{N}$ and

$$A_2 = [\varphi(r_{ij}) : 1 \le i \le N-1\ ,\ 0 \le j \le N\ and\ 0\ elsewhere],$$

$$D_k = [\varphi^{(k)}(r_{ij}) : 1 \le i \le N-1\ ,\ 0 \le j \le N\ and\ 0\ elsewhere],\ k = 1, 2,$$

$$G^{n+1} = [g_1^{n+1}(t), 0, \ldots, 0, g_2^{n+1}(t)]^T,$$

and

$$u^n = A_2\lambda^n, \qquad u_x^n = D_1\lambda^n.$$

The symbol " $*$ " means the ith component of the vector u^n and u_x^n are multiplied to all element in the ith row of the matrices D_1 and A_2 respectively. By attention to Equation (2.11), we have:

$$\lambda^{n+1} = M^{-1}N\lambda^n + M^{-1}G^{n+1}, \tag{2.12}$$

where

$$M = [A_1 - D_2 + \frac{\delta t}{2}[-\alpha D_2 + \beta D_1 + \gamma(A_2 * u_x^n + u^n * D_1)]],$$
$$N = [A_2 - D_2 - \frac{\delta t}{2}[-\alpha D_2 + \beta D_1]].$$

by using Equation (2.8) and (2.12) we can write

$$u^{n+1} = A_1 M^{-1} N A_1^{-1} u^n + A_1 M^{-1} G^{n+1}. \tag{2.13}$$

From Equation (2.13) we can find the solution at any time level n. For distinct collocation points, A_1 is always invertible [9]. Invertibility of matrix M cannot be provide, but in case of parameter-dependent RBFs, invertibility of M depends on shape parameters c. Optimal value of c calculate numerically in any problem.

Algorithm

1. choose N collocation point from the domain set [a,b].
2. choose the parameter δt.
3. Obtain the initial solution u^0 from Equation (2.2) and then find $\lambda^0 = A_1^{-1} u^0$ from Equation (2.8).
4. The parameters λ_j^{n+1} are calculate from Equation (2.12).
5. Finally, u^{n+1} at the successive time levels is obtained from step 4 and Equation (2.13).

3. Stability analysis

In this section we discuss stability of presented scheme (2.11), using the matrix method. To apply this method, we have linearized the non-linear term uu_x by assuming u as a constant. The error e^n at the nth time level is given by:

$$e^n = u^n_{exact} - u^n_{app}, \tag{3.1}$$

where u^n_{exact} and u^n_{app} are the exact and approximate solution at the nth time level respectively. The error equation for Equation (1.1) is as follows:

$$[H + \frac{\delta t}{2} K] e^{n+1} = [B - \frac{\delta t}{2} K] e^n, \tag{3.2}$$

where $K = [-\alpha D_2 + \beta D_1] A_1^{-1}$, $H = [A_1 - D_2] A_1^{-1}$ and $B = [A_2 - D_2] A_1^{-1}$. Let $P = [H + \frac{\delta t}{2} K]^{-1} [B - \frac{\delta t}{2} K]$, now we can write Equation (3.2) as follow:

$$e^{n+1} = P e^n. \tag{3.3}$$

Numerical scheme is stable if $\|P\|_2 \leq 1$, which is equivalent to $\rho(P) \leq 1$, where $\rho(P)$ denotes the spectral radius of the matrix P. By attention to above subjects, stability is assured if maximum eigenvalue of P satisfied in below condition:

$$|\frac{\lambda_B - \frac{\delta t}{2} \lambda_K}{\lambda_H + \frac{\delta t}{2} \lambda_K}| \leq 1, \tag{3.4}$$

where λ_H, λ_B and λ_K, are eigenvalue of the matrices H, B and K, respectively. For real eigenvalues, the inequality (3.4) hold true if $-\lambda_H \leq \lambda_B$ and $\lambda_B \leq \lambda_H + \lambda_K \delta t$. This shows that the scheme (2.11), is stable if

$$-\lambda_H \leq \lambda_B \leq \lambda_H + \lambda_K \delta t. \tag{3.5}$$

For complex eigenvalue $\lambda_B = a_b + ib_b$, $\lambda_H = a_h + ib_h$ and $\lambda_K = a_k + ib_k$,where a_b, b_b, a_h, b_h, a_k and b_k are real numbers, Equation (3.2) takes the following form:

$$|\frac{(a_b - \frac{\delta t}{2}a_k) + i(b_b - \frac{\delta t}{2}b_k)}{(a_h + \frac{\delta t}{2}a_k) + i(b_h + \frac{\delta t}{2}b_k)}| \leq 1. \tag{3.6}$$

The Equation (3.6) is satisfied if:

$$\delta t[a_k(a_b + a_h) + b_k(b_h + b_b)] + (b_h^2 - b_b^2) \geq 0, \tag{3.7}$$

and the scheme is stable.
The stability of the scheme (2.11) and conditioning of the component matrices H, K and B of the matrix P depend on the minimum distance between any two collocation points δx, in the domain set $[a, b]$, and the local shape parameter c.

4. Numerical solution

In this section we consider examples that solved by presented method in previous section. In order to illustrate the accuracy of the method, we used the error norm L_2 and L_∞ which are defined as follows:

$$L_2 = \|u^{exact} - u^{app}\|_2 = [\delta x \sum_{j=0}^{N} (u^{exact} - u^{app})^2]^{1/2}, \tag{4.1}$$

$$\tag{4.2}$$

$$L_\infty = \|u^{exact} - u^{app}\|_\infty = \max_j |u^{exact} - u^{app}|, \tag{4.3}$$

where δx is spatial step.

Example 4.1. *Consider the BBMB equation*

$$u_t - u_{xxt} - \alpha u_{xx} + \beta u_x + \gamma u u_x = 0, \tag{4.4}$$

with the following initial condition:

$$u(x, 0) = \sin x. \tag{4.5}$$

Exact solution of the above problem is given by

$$u(x, t) = e^{-t} \sin x, \tag{4.6}$$

where $\alpha = \beta = \gamma = 1$. The boundary conditions are taken from the exact solution. We solved the Example 4.1 for different values of t, $\delta t = 0.02, N = 20$ and [a,b]=[-10,10]. We used MQ, IMQ and GA radial basis functions with shape parameter respectively 0.001, 0.01 and 4.5. Table 1 shows the L_2 and L_∞ in $t = 0.02, 1, 10, 15$.

TABLE 1. Numerical results for Example 4.1.

RBF	Time	L_∞	L_2
GA			
	0.02	4.66869×10^{-4}	1.44763×10^{-3}
	1	8.8652×10^{-3}	2.74763×10^{-2}
	10	1.2221×10^{-5}	3.78939×10^{-5}
	15	1.31572×10^{-7}	4.07967×10^{-7}
MQ			
	0.02	9.67623×10^{-5}	2.30278×10^{-4}
	1	1.84825×10^{-3}	4.32048×10^{-3}
	10	2.56153×10^{-6}	6.13605×10^{-6}
	15	2.779×10^{-8}	7.00093×10^{-8}
IMQ			
	0.02	1.30952×10^{-6}	4.02232×10^{-6}
	1	2.45495×10^{-5}	7.54975×10^{-5}
	10	3.02594×10^{-8}	9.32299×10^{-8}
	15	3.05851×10^{-10}	9.42446×10^{-10}

TABLE 2. L_∞ and L_2 norm with increasing N for Example 4.1.

RBF	N	L_∞	L_2
MQ			
	20	2.88619×10^{-10}	7.31725×10^{-10}
	40	1.09293×10^{-9}	3.00421×10^{-9}
	60	3.25813×10^{-9}	9.56447×10^{-9}
GA			
	20	1.26062×10^{-9}	3.90885×10^{-9}
	40	1.48735×10^{-9}	4.56714×10^{-9}
	60	1.25202×10^{-8}	3.70878×10^{-8}
IMQ			
	20	2.74808×10^{-12}	8.46838×10^{-12}
	40	2.89735×10^{-12}	8.87604×10^{-12}
	60	3.11628×10^{-12}	9.47935×10^{-12}

Table 1 shows that IMQ has better accuracy than MQ and GA. The value of L_∞ and L_2 with increasing N in $t = 20$ is shown in Table 2.

Example 4.2. *Consider the BBMB Equation (1.1) with initial condition*

$$u(x,0) = sech^2(x/4), \;\; x \in R, \tag{4.7}$$

and exact solution

$$u(x,t) = sech^2(\frac{x}{4} - \frac{1}{3}t), \tag{4.8}$$

Figure 1. The surface shows the exact solution of BBMB Equation (4.4) when $\alpha = \beta = \gamma = 1$.

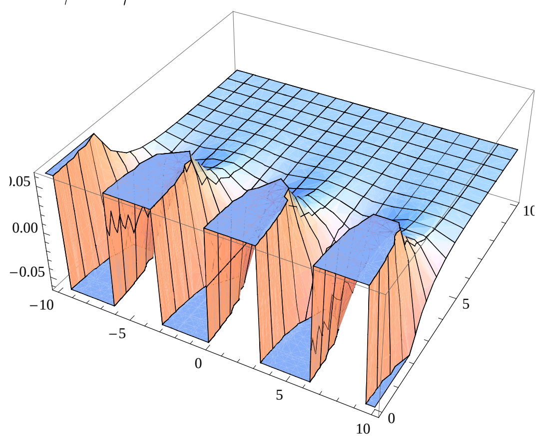

Table 3. Numerical results for Example 4.2.

RBF	Time	L_∞	L_2
MQ			
	0.1	9.61141×10^{-7}	1.33296×10^{-6}
	0.5	5.62261×10^{-6}	6.75616×10^{-6}
	1	1.39236×10^{-5}	1.40979×10^{-5}
	2	4.72334×10^{-5}	3.82213×10^{-5}
IQ			
	0.1	5.88745×10^{-7}	1.0765×10^{-6}
	0.5	2.08331×10^{-6}	4.41492×10^{-6}
	1	3.98593×10^{-6}	8.67292×10^{-6}
	2	7.82587×10^{-6}	1.71645×10^{-5}
IMQ			
	0.1	1.0724×10^{-6}	2.57134×10^{-6}
	0.5	3.1115×10^{-6}	6.35206×10^{-6}
	1	5.44985×10^{-6}	1.14811×10^{-5}
	2	9.1596×10^{-6}	1.98248×10^{-5}

where $\beta = \gamma = 1$ and $\alpha = 0$ and this equation is said BBM equation. Tables 3 shows numerical results, for $\delta t = 0.02, N = 60$, [a,b]=[-10,10], $c(MQ) = 1.1, c(IMQ) = 2.5$ and $c(IQ) = 2.5$. Table 4 shows accuracy with increasing N in $t = 0.5$.

FIGURE 2. The surface shows the exact solution of BBMB Equation (4.4) when $\alpha = 0$ and $\beta = \gamma = 1$.

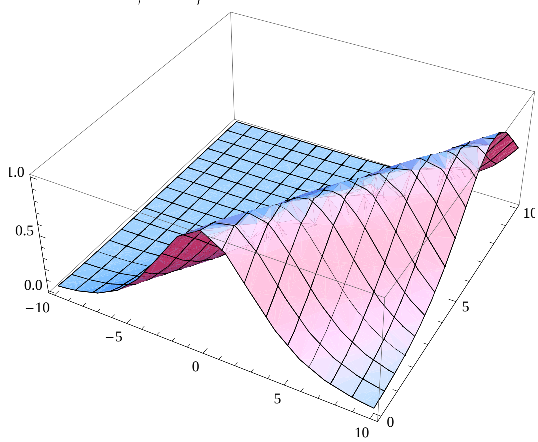

TABLE 4. L_∞ and L_2 norm with increasing N for Example 4.2.

RBF	N	L_∞	L_2
MQ			
	20	4.91306×10^{-4}	9.21948×10^{-4}
	40	1.46943×10^{-5}	1.57072×10^{-5}
	60	5.62261×10^{-6}	6.75616×10^{-6}
IQ			
	20	6.59187×10^{-6}	9.98726×10^{-6}
	40	1.88261×10^{-6}	4.33411×10^{-6}
	60	2.08331×10^{-6}	4.41492×10^{-6}
IMQ			
	20	1.15288×10^{-4}	1.64105×10^{-4}
	40	2.20596×10^{-6}	4.77047×10^{-6}
	60	3.1115×10^{-6}	6.35206×10^{-6}

5. CONCLUSIONS

In this work, we have applied mesh-free method for solution of BBMB equation based on radial basis function. The numerical results and tables show that errors are very small and this scheme is accurate and efficient approach for the solution of such type of nonlinear partial differential equations.

REFERENCES

[1] K. Al-Khaled, S. Momani, A. Alawneh, *Approximate wave solutions for generalized Benjamin-Bona-Mahony-Burgers equations*, Applied Mathematics and Computation, *171* (2005), 281–292.

[2] I. Dag, A. Dogan, B. Saka, *B-spline collocation methods for numerical solutions of the RLW equation*, Int. J. Comput. Math., *80* (2003), 743–757.

[3] I. Dag, B. Saka, D. Irk, *Galerkin method for the numerical solution of the RLW equation using quintic B-splines*, J. Comput. Appl. Math., *190* (2006), 532–547.

[4] C. Franke, *Scattered data interpolation: testes of some methods*, Math. Comput., *38* (1982), 181–200.

[5] E. J. Kansa, *Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-I. surface approximations and partial derivative estimates*, Comput. Math. Appl., *19* (1990), 127–145.

[6] M. A. Lopez-Marcos, *Numerical analysis of pseudospectral method for the kuramoto-sivashinsky equation.* IMA J. Numer. Anal., *14* (1994), 223–242.

[7] W. R. Madych, S. A. Nelson, *Multivariate interpolation and conditionally ppositive definite functions. II*, Math. Comput., *54* (1990), 211–230.

[8] S. A. V. Manickam, A. K. Pani, and S. K. Chang, *A second-order splitting combined with orthogonal cubic spline collocation method for the roseneau equation*, Numer. Methods Partial Differ. Equ., *14* (1998), 695–716.

[9] C. A. Michelli, *Interpolation of scattered data: Distance matrix and conditionally positive definite functions*, Construct. Approx., *2* (1986), 11–22.

[10] D. Michelson, *Steady solutions of the Kuramoto-Sivashinsky equation*, Physica D, *19* (1986), 89–111.

[11] S. Ul-Islam, S. Haq, A. Ali, *A meshfree method for the nimerical solution of the RLW equation*, Computational and Applied Mathematics, *223* (2009), 997–1012.

[12] Yan Xu, Chi-Wang Shu, *Local discontinuous Galerkin methods for the KuramotoSivashinsky equations and the Ito-type coupled KdV equations*, Comput. Methods Appl. Mech. Eng., *195* (2006), 3430–3447.