Available at
http://pvamu.edu/aam
Appl. Appl. Math.
ISSN: 1932-9466

Applications and Applied Mathematics:
An International Journal
(AAM)

Vol. 13, Issue 2 (December 2018), pp. 199 – 210

Modified Gaussian Radial Basis Function Method for the Burgers Systems

[1]Hossein Aminikhah and [2]Mostafa Sadeghi

Department of Applied Mathematics and Computer Science
Faculty of Mathematical Sciences
University of Guilan
P.O. Box 19141, P.C. 41938
Rasht, Iran
[1]aminikhah@guilan.ac.ir; [2]mostafasadeghi76@gmail.com

Received: November 21, 2018; Accepted: April 26, 2019

Abstract

In this paper, the systems of variable-coefficient coupled Burgers equation are solved by a free mesh method. The method is based on the collocation points with the modified Gaussian (MGA) radial basis function (RBF). Dependent parameters and independent parameters and their effect on the stability are shown. The accuracy and efficiency of the method has been checked by two examples. The results of numerical experiments are compared with analytical solutions by calculating errors infinity-norm.

Keywords: Partial differential equations; Radial basis function; Coupled Burgers equation; Discretizing; Taylor expansion; Shape parameter; Stability

MSC 2010 No.: 65M06, 65M22, 65N12

1. Introduction

In 1971 the multiquadric (MQ) method was presented by Hardy (1971). The importance of shape parameter in MQ method is shown by Tarwater (1985) and Micchelli (1986) had spoken about convergence of the RBFs approximation. Kansa (1990) introduced a modified MQ method for the numerical solution of PDEs. In recent years, Radial Basis Functions as a meshless method is used for numerical solutions of functional differential equations (Alqezweeni et al. (2018), Dehghan et al. (2014), Dehghan et al. (2015), Iurlaro et al. (2014), Islam et al. (2009), Kazemi et al. (2013),

Luan et al. (2018), Madych et al. (1992), Tarwater (1985), Zhong et al. (2014)). The modified Gaussian radial basis function method (MGA-RBF) is the component of functions that has the best approximation properties for the numerical solution of partial differential equations (PDEs). The MGA-RBF method prevents further computation in time, independent of the position of points, flexible for geometric position, high spectral accuracy and it's excellent for interpolation of data in several dimensions.

In this work, we use of the MGA-RBF method for numerical solution of the Burger systems. The systems of variable coefficient coupled Burgers equation can be written in the following basic form:

$$\begin{cases} \frac{\partial u}{\partial t} = r_1(t)\frac{\partial^2 u}{\partial x^2} + s_1(t)u\frac{\partial u}{\partial x} + p_1(t)\frac{\partial (uv)}{\partial x}, \\ \frac{\partial v}{\partial t} = r_2(t)\frac{\partial^2 v}{\partial x^2} + s_2(t)v\frac{\partial v}{\partial x} + p_2(t)\frac{\partial (vu)}{\partial x}, \end{cases} \quad x \in \Omega = (a,b)\,,\ t > 0, \tag{1}$$

where $r_1(t), r_2(t), s_1(t), s_2(t), p_1(t)$ and $p_2(t)$ are arbitrary smooth functions of t.

This paper is organized as follows. In Section 2, the structure of the method was introduced. Stability of the method is checked by applying the matrix multiplication method in Section 3. To demonstrate the efficiency of the proposed method, numerical experiments are carried out for two test problems and results are given in Section 4. In the last Section, we'll evaluation the performance of the proposed method.

2. Structure of the method

We have the basic form of the Burger equations (1) subject to the initial conditions:

$$u(x,0) = f(x),\ \ v(x,0) = g(x),\ \ x \in \overline{\Omega} = [a,b]\,, \tag{2}$$

and the boundary conditions:

$$\begin{cases} u(a,t) = g_a(t),\ \ u(b,t) = g_b(t), \\ v(a,t) = h_a(t),\ \ v(b,t) = h_b(t), \end{cases} \quad t \geq 0. \tag{3}$$

Discretizing Equation (1) in time, by taking the time weighting factor $0 \leq \theta \leq 1$, we get

$$\begin{aligned} \frac{u^{n+1}-u^n}{\tau} &= \theta r_1(t)\left(u_{xx}\right)^{n+1} + (1-\theta)\, r_1(t)\left(u_{xx}\right)^n \\ &\quad + \theta s_1(t)\left(uu_x\right)^{n+1} + (1-\theta)\, s_1(t)\left(uu_x\right)^n \\ &\quad + \theta p_1(t)\left(uv_x\right)^{n+1} + (1-\theta)\, p_1(t)\left(uv_x\right)^n \\ &\quad + \theta p_1(t)\left(u_x v\right)^{n+1} + (1-\theta)\, p_1(t)\left(u_x v\right)^n, \\ \frac{v^{n+1}-v^n}{\tau} &= \theta r_2(t)\left(v_{xx}\right)^{n+1} + (1-\theta)\, r_2(t)\left(v_{xx}\right)^n \\ &\quad + \theta s_2(t)\left(vv_x\right)^{n+1} + (1-\theta)\, s_2(t)\left(vv_x\right)^n \\ &\quad + \theta p_2(t)\left(vu_x\right)^{n+1} + (1-\theta)\, p_2(t)\left(vu_x\right)^n \\ &\quad + \theta p_2(t)\left(v_x u\right)^{n+1} + (1-\theta)\, p_2(t)\left(v_x u\right)^n,\ \ n = 0,1,\ldots,K-1, \end{aligned} \tag{4}$$

in which u^n and v^n are the values of $u(x,t_n)$ and $v(x,t_n)$ respectively where $t_n = \tau n$, $n = 0,1,\ldots,K$, $\tau = T/K$ and T is final time and $(uu_x)^{n+1}$, $(uv_x)^{n+1}$, $(u_x v)^{n+1}$, $(vv_x)^{n+1}$, $(vu_x)^{n+1}$

and $(v_x u)^{n+1}$ are the nonlinear term that approximated using the Taylor expansion as follows (Islam et al. (2009)):

$$\begin{aligned}
&(uu_x)^{n+1} \approx u^{n+1}u_x^n + u^n u_x^{n+1} - u^n u_x^n, \quad (uv_x)^{n+1} \approx u^{n+1}v_x^n + u^n v_x^{n+1} - u^n v_x^n,\\
&(u_x v)^{n+1} \approx u_x^{n+1}v^n + u_x^n v^{n+1} - u_x^n v^n, \quad (vv_x)^{n+1} \approx v^{n+1}v_x^n + v^n v_x^{n+1} - v^n v_x^n,\\
&(vu_x)^{n+1} \approx v^{n+1}u_x^n + v^n u_x^{n+1} - v^n u_x^n, \quad (v_x u)^{n+1} \approx v_x^{n+1}u^n + v_x^n u^{n+1} - v_x^n u^n.
\end{aligned} \tag{5}$$

By substituting (5) in (4) we have

$$\begin{aligned}
&u^{n+1} - \tau\theta r_1(t)(u_{xx})^{n+1} - \theta s_1(t)\tau\left(u^{n+1}u_x^n + u^n u_x^{n+1}\right)\\
&\quad - \tau\theta p_1(t)\left(u^{n+1}v_x^n + u^n v_x^{n+1} + u_x^{n+1}v^n + u_x^n v^{n+1}\right)\\
&= u^n + r_1(t)\tau(1-\theta)(u_{xx})^n + (1-2\theta)s_1(t)\tau(u^n u_x^n)\\
&\quad + (1-2\theta)p_1(t)\tau\left(u^n v_x^n + u_x^n v^n\right),\\
&v^{n+1} - \tau\theta r_2(t)(v_{xx})^{n+1} - \theta s_2(t)\tau\left(v^{n+1}v_x^n + v^n v_x^{n+1}\right)\\
&\quad - \tau\theta p_2(t)\left(v^{n+1}u_x^n + v^n u_x^{n+1} + v_x^{n+1}u^n + v_x^n u^{n+1}\right)\\
&= v^n + r_2(t)\tau(1-\theta)(v_{xx})^n + (1-2\theta)s_2(t)\tau(v^n v_x^n)\\
&\quad + (1-2\theta)p_2(t)\tau\left(v^n u_x^n + v_x^n u^n\right).
\end{aligned} \tag{6}$$

For approximation u^n, v^n and their derivatives in collocation points $\{x_i\}_{i=1}^N \subseteq \Omega$ where $h = x_{i+1} - x_i$, $i = 0, 1, \ldots, N-1$, we put

$$\begin{cases}
u(x_i, t^n) = \sum\limits_{j=1}^{N} \lambda_j^n \phi(r_{ij}),\\
v(x_i, t^n) = \sum\limits_{j=1}^{N} k_j^n \phi(r_{ij}),
\end{cases} \tag{7}$$

in which λ_j^n, k_j^n are unknown coefficients and ϕ is the MGA RBF and r_{ij} is the corresponding radius where introduced in Table 1.

Table 1. The MGA RBF and its derivatives on the left and the corresponding radius on the right

$\phi(r_{ij}) = e^{-\varepsilon^2 r_{ij}} + c_0$	$r_{ij} = (x_i - x_j)^2 + a$
$\phi'(r_{ij}) = -\varepsilon^2 e^{-\varepsilon^2 r_{ij}} + c_1$	$r_{ij} = (x_i - x_j)^2 + b$
$\phi''(r_{ij}) = \varepsilon^4 e^{-\varepsilon^2 r_{ij}} + c_2$	$r_{ij} = (x_i - x_j)^2 + d$

Also from Equations (7), we get the following equation for the boundary points,

$$\begin{cases}
\sum\limits_{j=1}^{N} \lambda_j^n \phi(r_{ij}) = g_k(t_n),\\
\sum\limits_{j=1}^{N} k_j^n \phi(r_{ij}) = h_k(t_n),
\end{cases} \tag{8}$$

where $k = 1, 2$ for $i = 1, N$, respectively.

In Table 1 the derivatives of MGA RBF and radius related to each of them also given, where ε and c_0, c_1, c_2, a, b, d are shape parameter and arbitrary parameters, respectively.

The matrix form of equation (7) is as follows:

$$\begin{cases} U^n = A\lambda^n, \\ V^n = AK^n, \end{cases} \tag{9}$$

where $\lambda^n = [\lambda_1^n, ..., \lambda_N^n]^T$, $K^n = [k_1^n, ..., k_N^n]^T$ and $A = [\phi(r_{ij})]_{i,j=1}^N$.

The matrix A can be split into two matrices A_d and A_b corresponding to $N-2$ interior points and two boundary points in the following form:

$$A = A_d + A_b,$$

where

$$A_d = [\phi(r_{ij}) : i = 2, .., N-1,\ j = 1, ..., N \text{ and } 0 \text{ else where}]_{N\times N},$$
$$A_b = [\phi(r_{ij}) : i = 1, N,\ j = 1, ..., N \text{ and } 0 \text{ else where}]_{N\times N}.$$

Now from Equations (6), (8) and (9) we obtain

$$\begin{aligned} &(A - \tau\theta r_1(t)C - \tau\theta s_1(t)\,(u_x^n * A_d + u^n * B) \\ &\quad -\tau\theta p_1\,(t)\,(v_x^n * A_d + u^n * \mathrm{B} + v^n * \mathrm{B} + u_x^n * A_d))\,\lambda^{n+1} \\ &= (A_d + r_1(t)\tau\,(1-\theta)\,C + (1-2\theta)\,s_1(t)\tau(u_x^n * A_d) \\ &\quad + (1-2\theta)\,p_1(t)\tau\,(v_x^n * A_d + u_x^n * A_d))\,\lambda^n + G^{n+1}, \\ &(A - \tau\theta r_2(t)C - \tau\theta s_2(t)\,(v_x^n * A_d + v^n * B) \\ &\quad -\tau\theta p_2\,(t)\,(u_x^n * A_d + v^n * \mathrm{B} + u^n * \mathrm{B} + v_x^n * A_d)\,K^{n+1} \\ &= (A_d + r_2(t)\tau\,(1-\theta)\,C + (1-2\theta)\,s_2(t)\tau(v_x^n * A_d) \\ &\quad + (1-2\theta)\,p_2(t)\tau\,(v_x^n * A_d + u_x^n * A_d))\,K^n + H^{n+1}, \end{aligned} \tag{10}$$

where

$$G^{n+1} = [g_a(t_{n+1})\ , 0, \ldots, 0,\ g_b(t_{n+1})]^T, \quad H^{n+1} = [h_a(t_{n+1})\ , 0, \ldots, 0\ , h_b(t_{n+1})]^T,$$
$$B = \left[\phi'\,(r_{ij}) = -\varepsilon^2 e^{-\varepsilon^2 r_{ij}} + c_1 : i = 2, .., N-1,\ j = 1, ..., N \text{ and } 0 \text{ else where}\right]_{N\times N},$$
$$C = \left[\phi''(r_{ij}) = \varepsilon^4 e^{-\varepsilon^2 r_{ij}} + c_2 : i = 2, .., N-1,\ j = 1, ..., N \text{ and } 0 \text{ else where}\right]_{N\times N},$$

and $v_x^n = BK^n$, $u_x^n = B\lambda^n$.

The symbol "$*$" means that the multiplication is componentwise.

Equation (10) can be rewritten as

$$\begin{cases} \lambda^{n+1} = M_1^{-1}Q_1\lambda^n + M_1^{-1}G^{n+1}, \\ K^{n+1} = M_2^{-1}Q_2K^n + M_2^{-1}H^{n+1}, \end{cases} \tag{11}$$

where

$$\begin{aligned}
M_1 &= \big(A - \tau\theta r_1(t)C - \theta s_1(t)\tau(u_x^n * A_d + u^n * B) \\
&\quad - \tau\theta p_1(t)(v_x^n * A_d + u^n * B + v^n * B + u_x^n * A_d)\big), \\
Q_1 &= \big(A_d + r_1(t)\tau\,(1-\theta)\,C + (1-2\theta)\,s_1(t)\tau(u_x^n * A_d) \\
&\quad + (1-2\theta)p_1(t)\tau(v_x^n * A_d + u_x^n * A_d)\big), \\
M_2 &= (A - \tau\theta r_2(t)C - \theta s_2(t)\tau\,(v_x^n * A_d + v^n * B) \\
&\quad - \tau\theta p_2\,(t)\,(u_x^n * A_d + v^n * \mathrm{B} + u^n * \mathrm{B} + v_x^n * A_d))\,, \\
Q_2 &= \big(A_d + r_2(t)\tau(1-\theta)C + (1-2\theta)s_2(t)\tau(v_x^n * A_d) \\
&\quad + p_2(t)(1-2\theta)\tau(u_x^n * A_d + v_x^n * A_d)\big).
\end{aligned}$$

By multiplying the matrix A on both sides of equations (11) and considering to equations (9), we arrive at the following equations:

$$\begin{cases} U^{n+1} = AM_1^{-1}Q_1A^{-1}U^n + AM_1^{-1}G^{n+1}, \\ V^{n+1} = AM_2^{-1}Q_2A^{-1}V^n + AM_2^{-1}H^{n+1}. \end{cases} \tag{12}$$

The results of this section can be summarized in the following algorithm.

Algorithm of the method:

Step 1. Choose the parameters τ, T and θ such that $0 \le \theta \le 1$.

Step 2. Choose n collocation points in Ω.

Step 3. Set $n = 0$.

Step 4. Calculate the initial solution u^0, v^0 from equations (2).

Step 5. Using equations (9) to find $\lambda^n = A^{-1}U^n$, $K^n = A^{-1}V^n$.

Step 6. Calculate matrices M_1, M_2 and vectors $Q_1\lambda^n + G^{n+1}$, $Q_2K^n + H^{n+1}$.

Step 7. The unknown vectors λ^{n+1}, K^{n+1} are calculated from equations (11).

Step 8. Calculate the approximate solutions U^{n+1}, V^{n+1} from step 7 and equations (9).

3. Stability analysis

In this section, the stability of equations (10) is checked by applying the matrix multiplication method. Now, we can write the error vectors at the time levels n and $n+1$ as following:

$$\begin{aligned} e_1^n &= u_{exact}^n - u_{app}^n, \\ e_2^n &= v_{exact}^n - v_{app}^n, \end{aligned}$$

where v_{app}^n, u_{app}^n and v_{exact}^n, u_{exact}^n are numerical and exact solutions at the time levels n, respectively.

The error equations for the equations (10) can be written as:

$$\begin{cases} [I + Z_1]\, e_1^{n+1} = [I + H_1]\, e_1^n, \\ [I + Z_2]\, e_2^{n+1} = [I + H_2]\, e_2^n, \end{cases} \tag{13}$$

where

$$\begin{aligned} Z_1 &= (-\tau\theta r_1(t)C - \theta s_1(t)\tau\,(u_x^n * A_d + u^n * B) \\ &\quad -\tau\theta p_1\,(t)\,(v_x^n * A_d + u^n * \mathrm{B} + v^n * \mathrm{B} + u_x^n * A_d))\, A^{-1}, \\ H_1 &= (r_1(t)\tau\,(1-\theta)\,C + (1-2\theta)\,s_1(t)\tau(u_x^n * A_d) \\ &\quad + (1-2\theta)\,p_1(t)\tau\,(v_x^n * A_d + u_x^n * A_d))\, A^{-1}, \\ Z_2 &= (-\tau\theta r_2(t)C - \theta s_2(t)\tau\,(v_x^n * A_d + v^n * B) \\ &\quad - \tau\theta p_2\,(t)\,(u_x^n * A_d + v^n * \mathrm{B} + u^n * \mathrm{B} + v_x^n * A_d))\, A^{-1}, \\ H_2 &= (r_2(t)\tau\,(1-\theta)\,C + (1-2\theta)\,s_2(t)\tau(v_x^n * A_d) \\ &\quad + p_2(t)\,(1-2\theta)\,\tau\,(v_x^n * A_d + u_x^n * A_d))\, A^{-1}. \end{aligned}$$

From (12) we have

$$\begin{cases} e_1^{n+1} = E e_1^n, \\ e_2^{n+1} = P e_2^n, \end{cases} \tag{14}$$

where $E = [I + Z_1]^{-1}\,[I + H_1]$, $P = [I + Z_2]^{-1}\,[I + H_2]$. Let $\eta_{H_1}, \eta_{H_2}, \eta_{Z_1}, \eta_{Z_2}$ be the eigenvalues of the matrices H_1, H_2, Z_1, Z_2, respectively. It can be seen that

$$\left\| \frac{1 + \eta_{H_1}}{1 + \eta_{Z_1}} \right\| < 1, \ \left\| \frac{1 + \eta_{H_2}}{1 + \eta_{Z_2}} \right\| < 1, \tag{15}$$

or $\rho(E) < 1$, $\rho(P) < 1$, where $\rho(.)$ denotes the spectral radius of a matrix. If $\theta = 0$ in the equations (14), we'll arrive at the following equations:

$$\|1 + \eta_{H_1}\| < 1, \quad \|1 + \eta_{H_2}\| < 1.$$

Hence, the scheme (10) is stable.

The error bound of the proposed method with the fixed nodes is as follows (Madych et al. (1992)):

$$\|u_{exact} - u_{approximat}\|_\infty = e^{-K(\varepsilon\theta)},$$

where $K(\varepsilon\theta)$ is dependent on the parameters ε and θ.

4. Numerical results

In this section, two examples of the systems of the variable-coefficient coupled Burgers equation are considered and will be solved using proposed method. The versatility and the accuracy of the proposed method are measured using absolute errors.

Example 4.1.

Consider the following coupled Burgers Equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + 2u\frac{\partial u}{\partial x} - \frac{\partial (uv)}{\partial x},$$
$$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} + 2v\frac{\partial v}{\partial x} - \frac{\partial (uv)}{\partial x},$$

subject to the boundary conditions:

$$u(0,t) = 0, u(1,t) = e^{-t}\sin(1),$$
$$v(0,t) = 0, v(1,t) = e^{-t}\sin(1),$$

and the initial conditions

$$u(x,0) = v(x,0) = \sin(x).$$

The exact solutions of the equation are $u\,(x,t) = v\,(x,t) = e^{-t}\sin x$.

We take $c_0 = 0.96$, $c_1 = 5000$, $c_2 = 0$, $a = 2.9$, $b = d = 2$, $\theta = 0.5$, $T = 1$, $\tau = 0.25$, $\varepsilon = 0.2$ and $h = 0.1$. The exact and numerical solutions and absolute errors, are reported for u and v, in Table 2 and Table 3. The space-time graphs of the exact and numerical solution for u are shown in Figures 1.

Table 2. Numerical results of Example 4.1 for u

x	u_{exact}	$u_{approximate}$	Absolute error
0.0	0	0.000000379513949	3.795×10^{-7}
0.1	0.036726661526271	0.036727142054588	4.805×10^{-7}
0.2	0.073086362390792	0.073086904361844	5.419×10^{-7}
0.3	0.108715808481443	0.108716357965022	5.494×10^{-7}
0.4	0.143259002150416	0.143259536940604	5.347×10^{-7}
0.5	0.176370799225032	0.176371328998357	5.297×10^{-7}
0.6	0.207720357574227	0.207720902282745	5.447×10^{-7}
0.7	0.236994442773761	0.236995002254844	5.594×10^{-7}
0.8	0.263900557841047	0.263901113532484	5.556×10^{-7}
0.9	0.288169865768312	0.288170400541276	5.347×10^{-7}
1.0	0.309559875653112	0.309560436755419	5.611×10^{-7}

Table 3. Numerical results of Example 4.1 for v

x	v_{exact}	$v_{approximate}$	Absolute error
0.0	0	0.000000379513949	3.795×10^{-7}
0.1	0.036726661526271	0.036727142054588	4.805×10^{-7}
0.2	0.073086362390792	0.073086904361844	5.419×10^{-7}
0.3	0.108715808481443	0.108716357965022	5.494×10^{-7}
0.4	0.143259002150416	0.143259536940604	5.347×10^{-7}
0.5	0.176370799225032	0.176371328998357	5.297×10^{-7}
0.6	0.207720357574227	0.207720902282745	5.447×10^{-7}
0.7	0.236994442773761	0.236995002254844	5.594×10^{-7}
0.8	0.263900557841047	0.263901113532484	5.556×10^{-7}
0.9	0.288169865768312	0.288170400541276	5.347×10^{-7}
1.0	0.309559875653112	0.309560436755419	5.611×10^{-7}

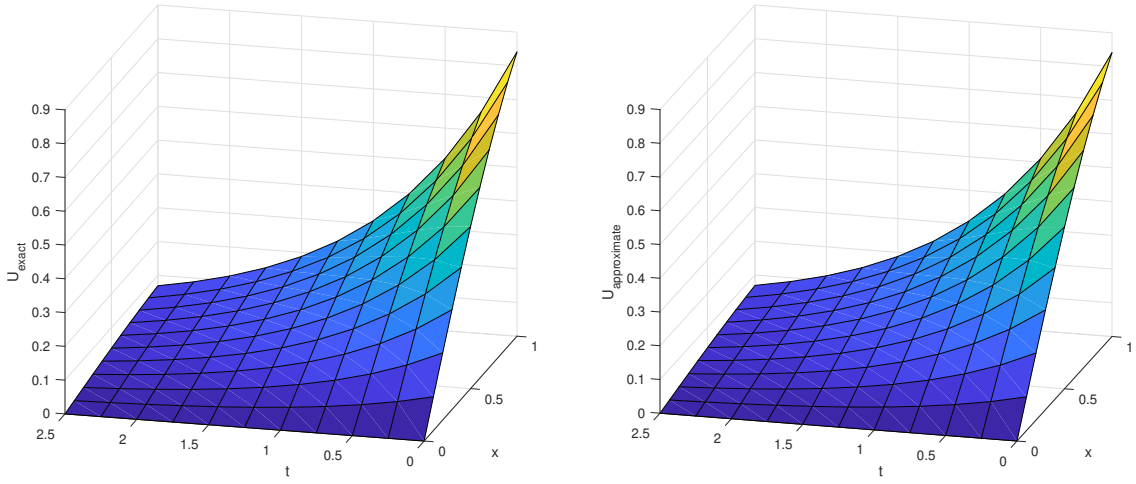

Figure 1. The exact and numerical answers of u example 4.1 for $x = 0 : 0.1 : 1$ and $t = 0 : 0.25 : 2.5$

Example 4.2.

Consider the following coupled Burgers equation:

$$\frac{\partial u}{\partial t} = \frac{t}{1-t}\frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} + \frac{1+t}{1-t}\frac{\partial (uv)}{\partial x},$$
$$\frac{\partial v}{\partial t} = \frac{t}{1+t}\frac{\partial^2 v}{\partial x^2} + v\frac{\partial v}{\partial x} - \frac{1-t}{1+t}\frac{\partial (uv)}{\partial x},$$

subject to the boundary conditions:

$$u(0,t)=0, u(1,t)=\frac{1}{1-t},$$
$$v(0,t)=0, v(1,t)=\frac{1}{1+t},$$

and the initial conditions

$$u(x,0)=x, \quad v(x,0)=x.$$

The exact solutions of the equation are

$$u\left(x,t\right)=\frac{x}{1-t}, \quad v\left(x,t\right)=\frac{x}{1+t}.$$

We take $c_0=3$, $c_1=5000$, $c_2=0$, $a=0.01$, $b=1.6$, $d=2$, $\theta=0.5393$, $T=3.5$, $\tau=0.25$, $\varepsilon=0.00039$ and $h=0.1$. In Table 4 and Table 5 the exact and numerical solutions and absolute errors, are reported for u and v, respectively. The space-time graphs of the exact and numerical solution for u are shown in Figures 2. In Figure 3, the exact and numerical solutions are also depicted for v.

Table 4. Numerical results of Example 4.2 for u

x	u_{exact}	$u_{approximate}$	Absolute error
0.0	0	-0.000000014901161	1.49×10^{-8}
0.1	-0.04	-0.039999999105930	8.94×10^{-10}
0.2	-0.08	-0.079999983310699	1.67×10^{-8}
0.3	-0.12	-0.119999982416630	1.76×10^{-8}
0.4	-0.16	-0.159999970346689	2.96×10^{-8}
0.5	-0.20	-0.199999980628490	1.94×10^{-8}
0.6	-0.24	-0.239999983459711	1.65×10^{-8}
0.7	-0.28	-0.279999975115061	2.49×10^{-8}
0.8	-0.32	-0.319999989122152	1.09×10^{-8}
0.9	-0.36	-0.359999999403954	5.96×10^{-10}
1.0	-0.40	-0.400000013411045	1.34×10^{-8}

Table 5. Numerical results of Example 4.2 for v

x	v_{exact}	$v_{approximate}$	Absolute error
0.0	0	0.000000039115548	3.91×10^{-8}
0.1	0.022222222222222	0.022222245112062	2.29×10^{-8}
0.2	0.044444444444444	0.044444456696510	1.22×10^{-8}
0.3	0.066666666666667	0.066666670143604	3.48×10^{-9}
0.4	0.088888888888889	0.088888887315989	1.57×10^{-9}
0.5	0.111111111111111	0.111111108213663	2.90×10^{-9}
0.6	0.133333333333333	0.133333329111338	4.22×10^{-9}
0.7	0.155555555555556	0.155555550009012	5.55×10^{-9}
0.8	0.177777777777778	0.177777769044042	8.73×10^{-9}
0.9	0.200000000000000	0.199999995529652	4.47×10^{-9}
1.0	0.222222222222222	0.222222222015262	2.07×10^{-10}

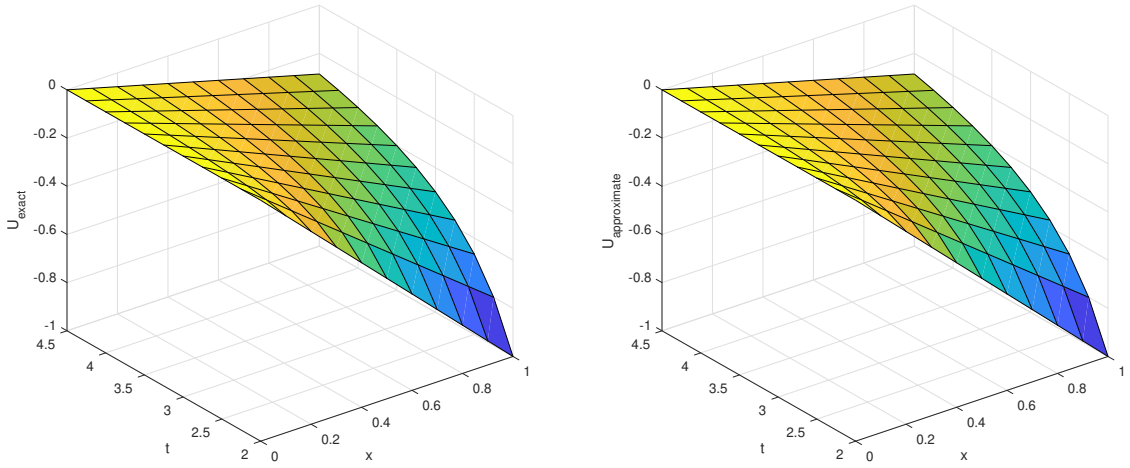

Figure 2. The exact and numerical answers of u example 4.2 for $x = 0 : 0.1 : 1$ and $t = 2 : 0.25 : 4.5$

5. Conclusion

In this paper, we presented a MGA-RBF method for solving two special variants of nonlinear Burgers systems. The results show that this scheme is an efficient approach for the solution of such type of nonlinear equations. Stability analysis is performed by the matrix method. In the proposed method when the optimal parameters are determined, we don't need to change them at the next levels of time and has low computational cost at all levels. Further work is required to find optimum value of the parameters, theoretically.

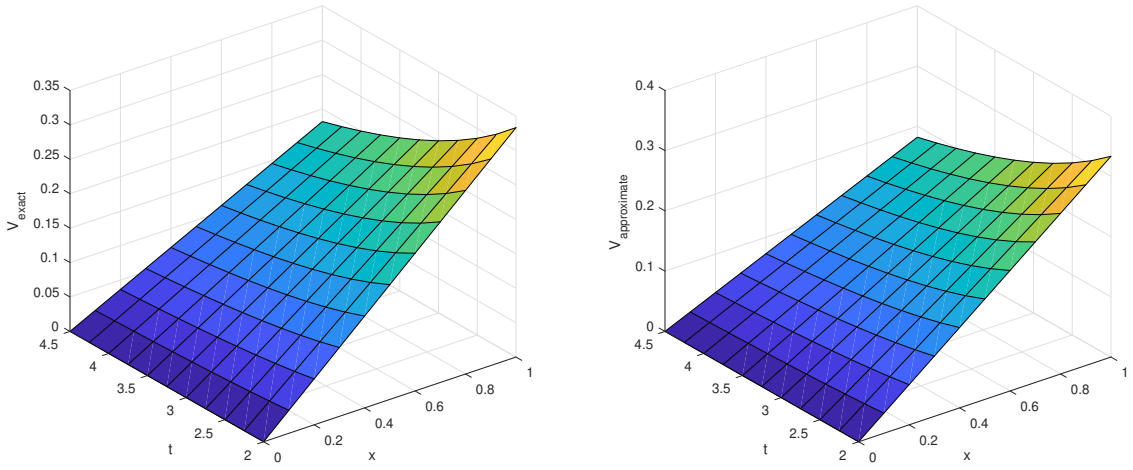

Figure 3. The exact and numerical answers of v example 4.2 for $x = 0 : 0.1 : 1$ and $t = 2 : 0.25 : 4.5$

Acknowledgment

We are very grateful to anonymous referees for their careful reading and valuable comments which led to the improvement of this paper.

REFERENCES

Alqezweeni, M. M., Gorbachenko, V. I., Zhukov, M. V. and Jaafar, M. S. (2018). Efficient Solving of Boundary Value Problems Using Radial Basis Function Networks Learned by Trust Region Method, International Journal of Mathematics and Mathematical Sciences, Vol. 2018, Article ID 9457578, pp. 1–4.

Dehghan, M., Abbaszadeh, M. and Mohebbi, A. (2014). The numerical solution of nonlinear high dimensional generalized Benjamin Bona Mahony Burgers equation via the meshless method of radial basis functions, Computers & Mathematics with Applications, Vol. 68, No. 3, pp. 212–237.

Dehghan, M. and Mohammadi, V. (2015). The numerical solution of Cahn Hilliard (CH) equation in one, two and three-dimensions via globally radial basis functions (GRBFs) and RBFs-differential quadrature (RBFs-DQ) methods, Engineering Analysis with Boundary Elements, Vol. 51, pp. 74–100.

Hardy, Rolland L. (1971). Multiquadric equations of topography and other irregular surfaces, Journal of Geophysical Research, Vol. 768, No. 8, pp. 1905–1915.

Iurlaro, L., Gherlone, M. and Sciuva, M. Di. (2014). Energy based approach for shape parameter selection in radial basis functions collocation method, Composite Structures, Vol. 107, pp. 70–78.

Islam, S. u., Haq, S. and Ali, A. (2009). A meshfree method for the numerical solution of the RLW

equation, Journal of Computational and Applied Mathematics, Vol. 223, No. 2, pp. 997–1012.

Kansa, E. J. (1990). Multiquadrics –A scattered data approximation scheme with applications to computational fluid-dynamics–II solutions to parabolic, hyperbolic and elliptic partial differential equations, Computers & Mathematics with Applications, Vol. 19, No. 8-9, pp. 147–161.

Kazemi, B. F. and Ghoreishi, F. (2013). Error estimate in fractional differential equations using multiquadratic radial basis functions, JJournal of Computational and Applied Mathematics, Vol. 245, pp. 133–147.

Luan, T., Sun, M., Xia, G. and Chen, D. (2018). Evaluation for Sortie Generation Capacity of the Carrier Aircraft Based on the Variable Structure RBF Neural Network with the Fast Learning Rate, Complexity, Vol. 2018, Article ID 6950124, pp. 1–19.

Madych, W. R. and Nelson, S. A. (1992). Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation, Journal of Approximation Theory, Vol. 70, No. 1, pp. 94–114.

Micchelli, Charles A. (1986). Interpolation of scatterted data: distance matrix and conditionally positive definite functions, Construct. Approx., Vol. 2, pp. 11–22.

Tarwater, A.E. (1985). Parameter study of Hardy's multiquadric method for scattered data interpolation, Lawrence Livermore National Laboratory, Technical Report UCRL–54670.

Zhong, Y., Huang, Xu, Meng, Pu and Li, F. (2014). PSO-RBF Neural Network PID Control Algorithm of Electric Gas Pressure Regulator, Abstract and Applied Analysis, Vol. 2014, Article ID 731368, pp. 1–7.