World Applied Sciences Journal 23 (12): 29-40, 2013
ISSN 1818-4952
© IDOSI Publications, 2013
DOI: 10.5829/idosi.wasj.2013.23.12.2840

A Numerical Meshless Technique for the Solution of the two Dimensional Burger's Equation Using Collocation Method

[1]*Iltaf Hussain*, [2]*Safyan Mukhtar and* [2]*Arshed Ali*

[1]Department of Basic Sciences and Islamiat,
University of Engineering and Technology Peshawar (Mardan Campus), Khyber Pakhtunkhwa, Pakistan
[2]Department of Mathematics, Bacha Khan University, Palosa Charsadda, Khyber Pakhtunkhwa, Pakistan

Abstract: In this paper we propose a meshfree technique for the numerical solution of the two dimensional Burger's equation. Collocation method using the Radial Basis Functions (RBFs) is coupled with first order accurate finite difference approximation. Different types of RBFs are used for this purpose. Performance of the proposed method is successfully tested in terms of various error norms. In the case of non-availability of exact solution, performance of the new method is compared with the results obtained from the existing methods available in the literature. The elementary stability analysis is established theoretically and is also supported by numerical results.

Key words: Two dimensional Burger's equations • Meshless method • Radial Basis Functions (RBFs) • Lattice Boltzmann Method (LBM) • Stability analysis

INTRODUCTION

Mathematical models of basic flow equations describing unsteady transport problems consist of a class of time dependent PDEs. One of such important models is known as the two dimensional Burger's equation [1]. The two dimensional Burger's equation occur in a large number of physical problems such as the phenomena of turbulence, flow through a shock wave traveling in a viscous fluid, sedimentation of two kinds of particles in fluid suspensions under the effect of gravity [2-4]. Analytical solution of the two dimensional Burger's equation is given by Fletcher [5] under restricted conditions. Various numerical methods have been introduced for the numerical solution of integrable and non-integrable two dimensional Burger's equation. Fletcher [6] has used finite element and finite difference methods for the numerical solution of the two dimensional model. Bahadir [3], Dehghan [4] and Radwan [7] have discussed variety of finite difference numerical schemes for solution of the problem. Khater *et al*. [8] have obtained numerical solution of the two dimensional Burger's equation by using spectral collocation method. Velivelli and Bryden [9] have obtained numerical solution of the problem on parallel processors by using traditional finite difference methods and lattice Boltzamann approach. Very recently, Duan and Liu [10], Zhang and Yan [11] have proposed various types of Boltzmann methods for the numerical solution of the two dimensional Burger's equation.

In this paper, we develop a meshfree algorithm for the numerical solution of the two dimensional nonlinear Burger's equation. The aim of this approach is to obtain approximate solution in a simple and effective manner free of mesh structure depending entirely on nodal points inside and/or in the boundary. A typical RBF approximation has the form

$$u(x) = \sum_{j=1}^{N} \lambda_i \varphi(r_i)$$

where φ is the RBF (with or without shape parameter c) listed in Table 1.

In 1990 Kansa [12] has used Multiquadric (MQ) to find approximate solution of different types of PDEs. Chen and Pepper [13] have used RBFs for simulating 1-D and 2-D groundwater contaminant transport models. Later on, this idea was modified by Fasshauer [14] to the Hermite type collocation method for invertibility of the collocation matrix. Siraj-ul-Isam *et al*. [15] have used different types of RBFs to obtain numerical solution of another class of PDEs known as RLW equation. The choice of radial basis functions is a flexible feature of meshfree methods. Infinitely differentiable RBFs like multiquadric (MQ), Inverse multiquadric (IMQ) and Gaussian (GA) contain a free

Corresponding Author: Iltaf Hussain, Department of Basic Sciences and Islamiat, University of Engineering and Technology Peshawar, Mardan Campus, Khyber Pakhtunkhwa, Pakistan

Table 1:

Name of the RBF	Definition
Multiquadric (MQ)	$\varphi(r,c)=\sqrt{c^2+r^2}$
Seventh degree spline (r^7)	$\varphi(r) = r^7$

parameter, called shape parameter, which affects both accuracy of the solution and condition number of the collocation matrix. The optimal value of the shape parameter c is an important factor in finding the maximal accuracy while maintaining numerical stability [16]. The choice of optimum value of the shape parameter is an open problem which is still under thorough investigation. Some authors like Carlson and Foley [17] have reported the dependency of the shape parameter on the function to be approximated by RBFs. They have observed that for rapidly varying functions, a small value of c be used, but a large value be used if the function has a large curvature. Tarwater [18] has also observed strong dependency of the Root Mean Square (RMS) error of the approximate solution on different values of the shape parameter c. Cheng *et al*. [19] showed that when c is very large then the RBFs system error is of exponential order. But there is a certain limit for the value c after which the solution breaks down. In general, as the value of the shape parameter c increases, the matrix of the system to be solved becomes highly ill-conditioned and hence the condition number can be used in determining the critical value of the shape parameter c for an accurate solution.

The contribution of this paper is the numerical study is to provide a straight forward approach for the solution of nonlinear PDEs. The new method can be implemented on a single desktop computer instead of using parallel processors for realization of the method. We look at the benchmark problems with different initial and boundary conditions to test the performance of the RBF method with the existing finite difference methods [3, 7] and lattice Boltzmann methods [9]. Further more, we discuss stability, space and time convergence of the new method.

The rest of the paper is organized as follows. In Section 2, formulation of the meshfree method for the numerical solution of the problem is given. Section 3, is devoted to stability analysis of the method. In Section 4, we apply the method to different types of problems related to the two dimensional Burger's equation. In Section 5, we summarize the results.

CONSTRUCTION OF THE METHOD

In order to implement the meshfree approach, we consider the two dimensional Burger's equation:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+u\frac{\partial u}{\partial y}-\nu\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]=0,(x,y)\in\Omega,\ t>0 \quad (1)$$

where the viscosity coefficient v = 1/R and R is the Reynolds number. Eq. (1) reduces to hyperbolic partial differential equation with shock like wave fronts for high value of Reynolds number. Eq. (1) accompanies the following Dirichlet boundary condition,

$$u(x,y,t)=f(x,y,t)(x,y)\in\partial\Omega,\ t>0 \quad (2)$$

and appropriate initial condition,

$$u(x,y,0)=g(x,y),\ (x,y)\in\Omega \quad (3)$$

where u(x,y,t) velocity component to be determined is, f and g are known functions, $\Omega\subset\mathbb{R}^2$ is the domain set and $\partial\Omega$ is the boundary of the domain set Ω.

We discretize the time derivatives of the two dimensional Burger's Eq. (1) using first order forward difference formula and applying the θ-weighted scheme ($0\le\theta\le1$) to the space derivatives at two successive time levels n and n+1 as below:

$$\frac{u^{n+1}-u^{n}}{\delta t}+\theta\left[u^{n+1}\left(\frac{\partial u}{\partial x}\right)^{n+1}+u^{n+1}\left(\frac{\partial u}{\partial y}\right)^{n+1}-\nu\left\{\left(\frac{\partial^2 u}{\partial x^2}\right)^{n+1}+\left(\frac{\partial^2 u}{\partial y^2}\right)^{n+1}\right\}\right]+(1-\theta)\left[u^{n}\left(\frac{\partial u}{\partial x}\right)^{n}+u^{n}\left(\frac{\partial u}{\partial y}\right)^{n}-\nu\left\{\left(\frac{\partial^2 u}{\partial x^2}\right)^{n}+\left(\frac{\partial^2 u}{\partial y^2}\right)^{n}\right\}\right]=0 \quad (4)$$

where $u^n=u(x,y,t^n),t^n=t^{n-1}+\delta t$ and δt is time step size.

The nonlinear terms in Eq. (4) can be approximated by using the following formula which can be obtained by applying the Taylor expansion,

$$u^{n+1}(\frac{\partial u}{\partial x})^{n+1}=u^{n+1}(\frac{\partial u}{\partial x})^{n}+u^{n}(\frac{\partial u}{\partial x})^{n+1}-u^{n}(\frac{\partial u}{\partial x})^{n} \quad (5)$$

Using Eq. (5) in Eq. (4) we obtain

$$u^{n+1}+\theta\delta t\left[u^{n}\left(\frac{\partial u}{\partial x}\right)^{n+1}+u^{n+1}\left(\frac{\partial u}{\partial x}\right)^{n}+u^{n}\left(\frac{\partial u}{\partial y}\right)^{n+1}+u^{n+1}\left(\frac{\partial u}{\partial y}\right)^{n}-n\left\{\left(\frac{\partial^2 u}{\partial x^2}\right)^{n+1}+\left(\frac{\partial^2 u}{\partial y^2}\right)^{n+1}\right\}\right]$$
$$=u^{n}-\delta t(1-2\theta)\left[u^{n}\left(\frac{\partial u}{\partial x}\right)^{n}+u^{n}\left(\frac{\partial u}{\partial y}\right)^{n}\right]+\delta t(1-\theta)\left[n\left\{\left(\frac{\partial^2 u}{\partial x^2}\right)^{n}+\left(\frac{\partial^2 u}{\partial y^2}\right)^{n}\right\}\right] \quad (6)$$

Let (x_i,y_i), $i = 1,2,\ldots,N$ be the collocation points in the domain set Ω such that (x_i,y_i), $i = 1,2,\ldots,N_d$ are interior points of the domain set Ω and (x_i,y_i), $i = N_d+1, N_d+2,\ldots,N$ are boundary points of the domain set Ω. Solution of Eq. (1) can be approximated by

$$u^{n}(x,y)=\sum_{j=1}^{N}\lambda_i^{n}\,\varphi(r_j) \quad (7)$$

where φ is radial basis function

$$r_j=\sqrt{\left(x-x_j\right)^2+\left(y-y_j\right)^2}$$

represents the Euclidean distance between the points (x,y) and (x_j,y_j). The points (x_j,y_j), $j = 1,2,\ldots,N$ are known as centers. The unknown parameters λ_j $j = 1,2,\ldots,N$ in Eq. (7) are to be determined by the collocation method. Therefore for each collocation point (x_j,y_j) Eq. (7) can be written as

$$u^{n}(x_i,y_i)=\sum_{j=1}^{N}\lambda_j^{n}\,\varphi(r_{ij}),\, i=1,\ 2,\ldots,N \quad (8)$$

Eq. (8) can be expressed in a matrix form as

$$\mathbf{U}^{n}=\mathbf{A}\,l^{\,n} \quad (9)$$

where

$$\mathbf{A}=\begin{bmatrix}\varphi(r_{11}) & \varphi(r_{12})\ldots & \varphi(r_{1N})\\ \varphi(r_{21}) & \varphi(r_{22})\ldots & \varphi(r_{2N})\\ . & . & .\\ . & . & .\\ \varphi(r_{N1}) & \varphi(r_{N2})\ldots & \varphi(r_{NN})\end{bmatrix}$$

and $\lambda^{n}=[\lambda_1^{n},\lambda_2^{n},\ldots,\lambda_N^{n}]^{T}$.

The matrix A can be written as $A = A_d+A_b$, where

$$\mathbf{A_d}=\left[\varphi(r_{ij}):i=1,2,\ldots,N_d,\, j=1,2,\ldots,N \text{and 0 elsewhere}\right]$$

$$\mathbf{A_b}=\left[\varphi(r_{ij}):i=N_d+1,N_d+2,\ldots,N, j=1,2,\ldots,N \text{and 0 elsewhere}\right]$$

The RBFs listed in Table 1 are used for meshfree approximation of the problem. The shape parameter c plays an important role in getting accurate solution. The value of c is to be found numerically for each radial basis functions and for each problem separately.

Using Eq. (9) in Eq. (6), we get the following equation for the interior points of the domain set Ω

$$\sum_{j=1}^{N}\lambda_j^{n+1}\varphi(r_{ij})+\theta\delta t\left[\begin{array}{l}\sum_{j=1}^{N}\lambda_j^{n}\varphi(r_{ij})\sum_{j=1}^{N}\lambda_j^{n+1}\varphi_x(r_{ij})\\ +\sum_{j=1}^{N}\lambda_j^{n+1}\varphi(r_{ij})\sum_{j=1}^{N}\lambda_j^{n}\varphi_x(r_{ij})\end{array}\right.$$
$$+\sum_{j=1}^{N}\lambda_j^{n}\varphi_j(r_{ij})\sum_{j=1}^{N}\lambda_j^{n+1}\varphi_y(r_{ij})+\sum_{j=1}^{N}\lambda_j^{n+1}\varphi_j(r_{ij})\sum_{j=1}^{N}\lambda_j^{n}\varphi_y(r_{ij})$$
$$\left.-\nu\left\{\sum_{j=1}^{N}\lambda_j^{n+1}\varphi_{xx}(r_{ij})+\sum_{j=1}^{N}\lambda_j^{n+1}\varphi_{yy}(r_{ij})\right\}\right]$$
$$=\sum_{j=1}^{N}\lambda_j^{n}\varphi(r_{ij})-\delta t(1-2\theta)\left[\begin{array}{l}\sum_{j=1}^{N}\lambda_j^{n}\varphi(r_{ij})\sum_{j=1}^{N}\lambda_j^{n}\varphi_x(r_{ij})\\ +\sum_{j=1}^{N}\lambda_j^{n}\varphi(r_{ij})\sum_{j=1}^{N}\lambda_j^{n}\varphi_y(r_{ij})\end{array}\right]$$
$$+\delta t(1-\theta)\left[\nu\left\{\sum_{j=1}^{N}\lambda_j^{n}\varphi_{xx}(r_{ij})+\sum_{j=1}^{N}\lambda_j^{n}\varphi_{yy}(r_{ij})\right\}\right]$$
$$i=1,2,\ldots,\ N_d \quad (10)$$

where

$$\varphi_x\left(r_{ij}\right)=\frac{\partial}{\partial x}\varphi\left(\sqrt{\left(x-x_j\right)^2+\left(y-y_j\right)^2}\right)\Bigg|_{\substack{x=x_i\\ y=y_i}}$$

$$\varphi_{xx}\left(r_{ij}\right)=\frac{\partial^2}{\partial x^2}\varphi\left(\sqrt{\left(x-x_j\right)^2+\left(y-y_j\right)^2}\right)\Bigg|_{\substack{x=x_i\\ y=y_i}}$$

and φ_y and φ_{yy} can be defined in similar way.

Also from Eqs. (9) and (3), we get the following equation for the boundary points,

$$\sum_{j=1}^{N}\lambda_j^{n+1}\varphi(r_{ij})=f(x_i,y_i,t^{n+1}),\, i=N_d+1,N_d+2,\ldots,N \quad (11)$$

Eqs. (10)-(11) can be written in matrix form as

$$\left[\mathbf{A}+\delta t\theta\left\{\mathbf{U}^{n}{}_{x}*\mathbf{A}+\mathbf{U}^{n}*\mathbf{B}_1+\mathbf{U}^{n}{}_{y}*\mathbf{A}+\mathbf{U}^{n}*\mathbf{B}_2-\nu(\mathbf{B}_3+\mathbf{B}_4)\right\}\right]l^{\,n+1}$$
$$=\left[\mathbf{A}-\delta\mathbf{t}(\mathbf{1}-\mathbf{2}\theta)\left\{\mathbf{U}^{n}*\mathbf{B}_1+\mathbf{U}^{n}*\mathbf{B}_2\right\}+\nu(1-\theta)\left\{\mathbf{B}_3+\mathbf{B}_4\right\}\right]l^{\,n}+\mathbf{C} \quad (12)$$

where

$$\mathbf{C}=\left[f(x_1, y_1, t^{n+1}), f(x_2, y_2, t^{n+1}), \ldots, f(x_N, y_N, t^{n+1})\right]^T$$

$$\mathbf{B_1}=\left[\varphi_x(r_{ij}) : i=1,2,\ldots,N_d\,, j=1,2,\ldots,N\text{and 0 elsewhere}\right]$$

$$\mathbf{B_2}=\left[\varphi_y(r_{ij}) : i=1,2,\ldots,N_d\,, j=1,2,\ldots,N\text{and 0 elsewhere}\right]$$

$$\mathbf{B_3}=\left[\varphi_{xx}(r_{ij}) : i=1,2,\ldots,N_d,\ j=1,2,\ldots,N\text{and 0 elsewhere}\right]$$

$$\mathbf{B_4}=\left[\varphi_{yy}(r_{ij}) : i=1,2,\ldots,N_d,\ j=1,2,\ldots,N\text{and 0 elsewhere}\right]$$

The symbol “*” means that ith component of the vector U^n is multiplied to every element of the ith row of the matrix B_1. Compact form of Eq. (12) is given by

$$\mathbf{M}\, l^{n+1} = \mathbf{N}\, l^{n} + \mathbf{C} \tag{13}$$

where

$$\mathbf{M}=\left[\mathbf{A}+\delta t\theta\left\{\begin{matrix}\mathbf{U}^n{}_x * \mathbf{A} + \mathbf{U}^n * \mathbf{B}_1 + \mathbf{U}^n{}_y * \mathbf{A} \\ +\mathbf{U}^n * \mathbf{B}_2 - \nu(\mathbf{B}_3 + \mathbf{B}_4)\end{matrix}\right\}\right]$$

and

$$\mathbf{N}=\left[\mathbf{A}\text{ - }\delta\mathbf{t(1\text{-}2}\theta)\left\{\mathbf{U}^n * \mathbf{B}_1 + \mathbf{U}^n * \mathbf{B}_2\right\} + \nu(1-\theta)\left\{\mathbf{B}_3 + \mathbf{B}_4\right\}\right] l^{n}$$

$$l^{n+1} = \mathbf{M}^{-1}\mathbf{N}\, l^{n} + \mathbf{M}^{-1}\mathbf{C} \tag{14}$$

Using Eq. (14) in Eq. (9) at (n+1) time level, we get

$$\mathbf{U}^{n+1} = \mathbf{A\,M^{-1}N\,A^{-1}U}^{n} + \mathbf{AM^{-1}C} \tag{15}$$

The non singularity of the matrix M cannot be shown in general [12], therefore, it is not possible to show that the scheme is well-posed in all such cases. However, the cases of singularity in the practical problems are rare. Eq. (12) represents a system of “N” linear equations in “N” unknown parameters λ_j, j = 1,2,…,N. This system can be solved by the Gaussian elimination method. The approximate solution can be found from Eq. (9) at any point in the domain set Ω, after finding the values of the unknown parameters λ_j, j = 1,2,…,N at each time level. The results of this section can be summarized in the following algorithm.

Algorithm

The algorithm works in the following manner:

1. Choose N collocation points from the domain set Ω.
2. Choose the parameters δt and θ such that ($0\leq\theta\leq1$).
3. Calculate the initial solution U^0 from Eq. (3) and then use Eq. (9) to find $\lambda^n = \mathbf{A}^{-1}\mathbf{U}^n$
4. The parameters λ_j^{n+1} (j = 1,2,…,N) are calculated from Eq. (14).
5. The approximate solution U^{n+1} at the successive time levels is obtained by combination of step 4 and Eq. (9).

STABILITY ANALYSIS

In this section, we present the stability analysis of the RBF approximation given in Eq. (12) using the matrix method. The error e^n at the *n*th time level is given by

$$\mathbf{e}^n = \mathbf{U}^n_{exact} - \mathbf{U}^n_{app}$$

where $\mathbf{U}^n_{exact}$, $\mathbf{U}^n_{app}$ are the exact and the approximate solutions at the *n* th time level. The error equation for the linearized two dimensional Burger’s equation can be written as:

$$\left[\mathbf{MA}^{-1}\right]\mathbf{e}^{n+1} = \left[\mathbf{NA}^{-1}\right]\mathbf{e}^{n}$$

$$[\mathbf{I}+\delta t\theta\mathbf{R}]\mathbf{e}^{n+1} = [\mathbf{I}+\delta t\mathbf{S}]\mathbf{e}^{n} \tag{16}$$

where R = DA^{-1} and

$$\mathbf{S} = \{(1-\theta)F\mathbf{A}^{-1} - (1-2\,\theta)\,E\mathbf{A}^{-1}\}$$

Eq. (16) can be rewritten as

$$e^{n+1} = \mathbf{T}e^{n} \tag{17}$$

where

$$\mathbf{T} = [\mathbf{I}+\delta t\theta\mathbf{R}]^{-1}[\mathbf{I}+\delta t\mathbf{S}]$$

The numerical scheme is stable if $\|T\|_2\leq1$, which is equivalent to $\rho(T)\leq1$, where $\rho(T)$ denotes the spectral radius of the matrix T. The sufficient condition for stability of the method (15) is guaranteed if all the eigenvalues of the matrix $[I+\delta t\theta R]^{-1}[I\_\delta tS]$ satisfy the following condition

$$\left|\frac{1+\delta t\lambda_{\mathbf{S}}}{1+\delta t\theta\lambda_{\mathbf{R}}}\right| \leq 1 \tag{18}$$

where λ_S and λ_R are eigenvalues of the matrices S and R respectively.

For θ = 0.5, the inequality (18) reduces to

$$\left|\frac{1+0.5\delta t\lambda_{\mathbf{Z}}}{1+0.5\delta t\lambda_{\mathbf{R}}}\right| \leq 1 \tag{19}$$

where Z = FA^{-1}.

The inequality (19) is satisfied if $\lambda_R \geq \lambda_Z$. This shows that for θ = 0.5, the scheme (12) will be unconditionally stable if $\lambda_R \geq \lambda_Z$. When θ = 0, the inequality (18) becomes $|1+\delta t\lambda_L| \leq 1$, where $\Lambda = F\mathbf{A}^{-1} - E\mathbf{A}^{-1}$, i.e. $\delta t \leq \frac{-2}{\lambda_L}$ and $\lambda_\Lambda \leq 0$. Thus for θ = 0, the scheme is conditionally stable.

The sufficient condition for stability of the scheme (12) and condition number of the component matrices R, S of the matrix T depend on the weight parameter θ, the number collocation points N and the shape parameter c. Cheng *et al*. [19] showed that when c is very large then the RBFs system error is of exponential order and the system leads to ill-conditioning. In the case of ill-conditioned system, the numerical solution thus produced is not stable. For fixed number of collocation points, co-relation between the condition number of the matrix T and the different values of the shape parameter c enable us to determine interval of stability for the method. This relationship is shown in Table 8 and Fig. 5 corresponding to Problem 1 for MQ. The interval of stability in this case is (0.02, 0.38) corresponding to v = 1, N = 400, δt = 0.005 and t = 1.0. It is clear from the Table 8 and Fig. 5, that if the value of shape parameter c is greater than the critical value c = 0.38, then the solution breaks down and hence the method becomes unstable. It can be seen from the Table 6, 7 and Fig. 1-4 that the sufficient condition of stability given in Eq. (19) is satisfied for the test Problem 4.1. In case of the parameter free RBFs such as seventh degree spline (r^7), the stability and conditioning depend on the weight parameter θ, eigenvalues λ_R, λ_Z and the number collocation points N.

NUMERICAL TESTS AND RESULTS

In this section we present the results of numerical tests of our scheme for the solution of the two dimensional Burger's equation (1)-(3). The accuracy of the scheme is measured in terms of error norms L_∞, L_R and L_2 which are defined below:

$$L_\infty = \max_j \left|(u_{exact})_j - (u_{app})_j\right|$$

$$L_R = \left(\frac{\sum_{j=1}^{N_d} \left((u_{exact})_j - (u_{app})_j\right)^2}{\sum_{j=1}^{N_d} \left((u_{exact})_j\right)^2} \right)^{1/2}$$

$$L_2 = \left(\sum_{j=1}^{N_d} \left((u_{exact})_j - (u_{app})_j\right)^2 \right)^{\frac{1}{2}}$$

where u_{exact} and u_{app} represent the exact and approximate solutions respectively. The value of the weight parameter θ used in the main scheme (12) is taken as 0.5 for each problem. The performance of the method is also compared with some of the published papers [7, 8, 15, 21, 22].

Test problem: Consider the two dimensional unsteady Burger's equation (1) that is dominated by moderate gradients [8] with the following boundary conditions

$$u(0,y,t) = \frac{1}{1+e^{(y-t)/2v}}$$

$$u(1,y,t) = \frac{1}{1+e^{(1+y-t)/2v}}$$

$$u(x,0,t) = \frac{1}{1+e^{(x-t)/2v}}$$

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

and initial condition

$$u(x,y,0) = \frac{1}{1+e^{(x+y)/2v}},\ \ 0 \leq x \leq 1, 0 \leq y \leq 1$$

The numerical tests of the two dimensional Burger's equation are performed for radial basis functions MQ and seventh degree spline (r^7). The algorithm is examined up to maximum time t = 0.25. The error norms E_∞ and E_2 are computed for various values of the parameters N, δt, R using MQ and r^7 RBFs. These results are shown in Table 2 and 3. The performance of the new method is compared with LBM [8]. Table 2 clearly shows that the results of MQ are better for large N and small v (specially in L_∞ error norm). The performance of L_2 error norm of [8] is slightly better than our method. The results produced by spline basis (r^7) are given in Table 3 along with the results of [8]. Like previous case L_∞ norm of our method is slightly better than [8] whereas L_2 norm of method [8] is marginally better than our method. Improved accuracy of the new method for small values of v is evident of the fact that it can handle steep shock like wave front accurately in the same way as [8]. Time and spatial rates of convergence for Problem 4.1 are presented in Table 4 and 5.

Table 2: L_∞ and L_2 errors at different viscosities and c = 0.18, $\delta t = 0.0025$, t = 0.25 for problem 4.1

		L_∞ error		L_2 error	
Viscosity	Collocation points N (Grid)	MQ	[8]	MQ	[8]
v = 1	100(10×10)	7.8844×10^{-4}	7.882×10^{-5}	4.0331×10^{-3}	2.016×10^{-5}
	400(20×20)	8.5596×10^{-5}	1.483×10^{-4}	8.4763×10^{-4}	2.652×10^{-5}
	1600(40×40)	2.2577×10^{-6}	2.921×10^{-4}	4.5880×10^{-5}	3.681×10^{-5}
v = 0.1	100(10×10)	7.1842×10^{-4}	1.061×10^{-2}	3.6837×10^{-3}	2.565×10^{-3}
	400(20×20)	3.5395×10^{-4}	3.077×10^{-3}	1.9617×10^{-3}	4.536×10^{-4}
	1600(40×40)	7.4860×10^{-6}	1.626×10^{-2}	6.0343×10^{-5}	1.875×10^{-3}
v = 0.01	1600(40×40)	6.4518×10^{-2}	--	8.5465×10^{-2}	--

Table 3: L_∞ and L_2 errors at different viscosities and $\delta t = 0.0025$, t = 0.25 for problem 4.1

		L_∞ error		L_2 error	
Viscosity	Collocation points N (Grid)	r^7	[8]	r^7	[8]
v = 1	100(10×10)	8.5437×10^{-5}	7.882×10^{-5}	6.8373×10^{-4}	2.016×10^{-5}
	400(20×20)	2.6616×10^{-6}	1.483×10^{-4}	4.1168×10^{-5}	2.652×10^{-5}
	1600(40×40)	1.0841×10^{-7}	2.921×10^{-4}	3.0647×10^{-6}	3.681×10^{-5}
v = 0.1	100(10×10)	3.1857×10^{-4}	1.061×10^{-2}	9.4443×10^{-4}	2.565×10^{-3}
	400(20×20)	3.2712×10^{-5}	3.077×10^{-3}	1.1452×10^{-4}	4.536×10^{-4}
	1600(40×40)	6.4858×10^{-6}	1.626×10^{-2}	6.4032×10^{-5}	1.875×10^{-3}
v = 0.01	1600(40×40)	6.3565×10^{-3}	--	2.3144×10^{-2}	--

Table 4: Time rate of convergence produced by MQ at c = 0.18, t = 0.25, N = 400, v = 0.01 for problem 4.1

δt	L_∞	Order	L_2	Order
0.1	2.1196×10^{-3}	--	3.8717×10^{-3}	--
0.05	4.2705×10^{-4}	2.3113	7.0465×10^{-4}	2.4580
0.02	3.1285×10^{-4}	0.3396	5.6663×10^{-4}	0.2379
0.01	3.4560×10^{-4}	-0.1436	6.6977×10^{-4}	-0.2413
0.005	3.5395×10^{-4}	-0.0344	7.0062×10^{-4}	-0.0650

Table 5: Spatial rate of convergence produced by MQ at time c = 0.18, t = 0.25, v = 0.1 for problem 4.1

N	L_∞	Order	L_2	Order
25	1.7665×10^{-2}	--	5.3605×10^{-2}	--
100	3.3948×10^{-3}	2.3795	7.5466×10^{-4}	2.8284
400	3.5395×10^{-4}	3.2617	7.0062×10^{-4}	3.4291
625	1.3136×10^{-4}	4.4421	2.5296×10^{-4}	4.5653
1600	7.4860×10^{-6}	6.0955	9.9991×10^{-6}	6.8738

The point wise rate of convergence in space and time is calculated by using the following formulae:

$$\frac{\log_{10}\left(\left\|u_{exact}-u_{h_i}\right\| / \left\|u_{exact}-u_{h_{i+1}}\right\|\right)}{\log_{10}(h_i / h_{i+1})} \quad \text{and} \quad \frac{\log_{10}\left(\left\|u_{exact}-u_{\delta t_i}\right\| / \left\|u_{exact}-u_{\delta t_{i+1}}\right\|\right)}{\log_{10}(\delta t_i / \delta t_{i+1})}$$

The term u_{exact} is the exact solution, whereas u_{h_i} and $u_{\delta t_i}$ are the numerical solutions with spatial step size h_i and time step size δt_i respectively. Computations are carried out with the different spatial and time step sizes to examine the point rates of convergence in space and time for MQ. These results are shown in Fig. 3-5 as well.

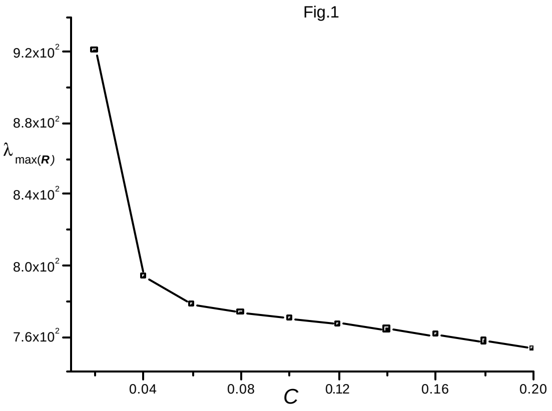

Fig. 1: Graph of maximum eigenvalue of the matrix R corresponding to v = 0.1, N = 400 and t = 0.25 using MQ

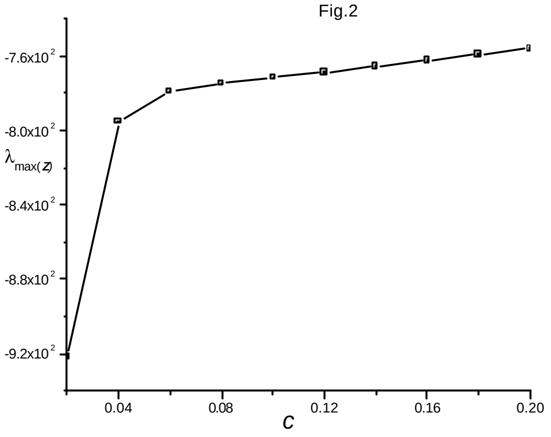

Fig. 2: Graph of maximum eigenvalue of the matrix Z corresponding to v = 0.1, N = 400 and t = 0.25 using MQ

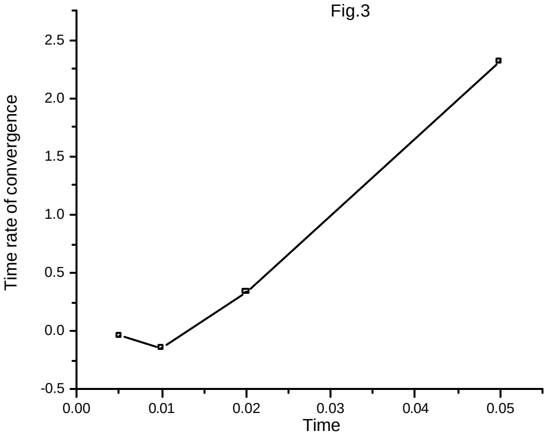

Fig. 3: Graph of time rat of convergence produced by MQ at c = 0.18, t = 0.25, N = 400, v = 0.1

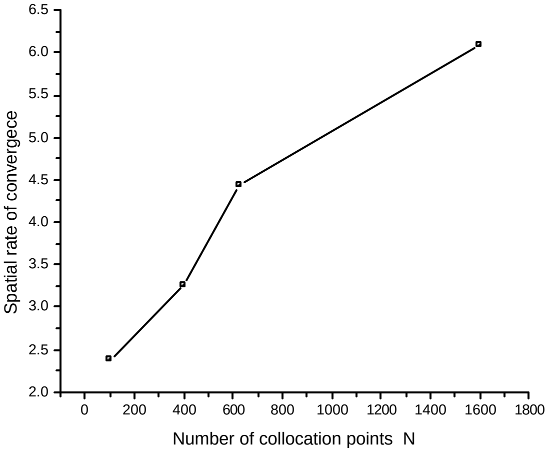

Fig. 4: Graph of time rat of convergence produced by MQ at c = 0.18, t = 0.25, N = 400, v = 0.1

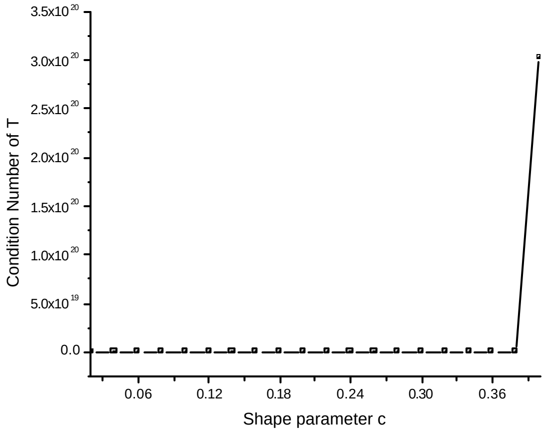

Fig. 5: Graph of condition number versus shape parameter c at t = 1.0, v = 0.1, δt = 0.005 and N = 400

Test problem: Consider the two dimensional unsteady Burger's equation (1) having domain set [-0.5,0.5]×[-0.5,0.5]. The exact solution of the equation is given in [20]

$$u=0.5-\tanh\left(\frac{x+y-t}{2\nu}\right)$$

The boundary conditions can be extracted from the exact solution. Values of L_2 and L_∞ error norms corresponding to MQ and r^7 are listed in Table 9 and 10. The authors in [21] have reported only graphical solutions. Graphical solution of the problem is shown in Fig. 6 and 7. In [21] the author have taken the grid size 200×200 and v = 0.002. We have restricted our computations up to collocation points 2500 and

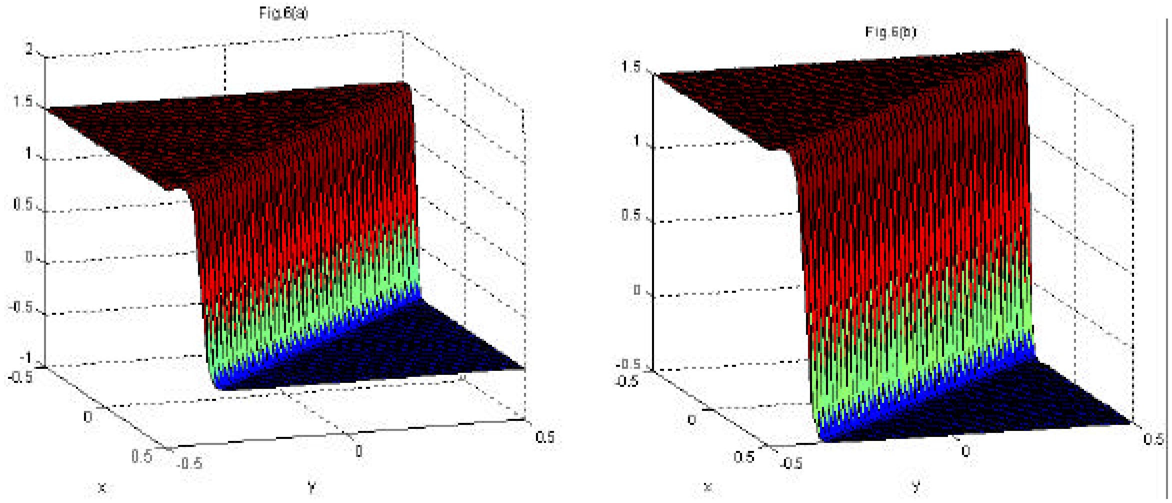

Fig. 6: Numerical (6a) and Exact (6b) solution at c = 0.1, N = 2500, v = 0.01, δt = 0.001 and t = 0.01

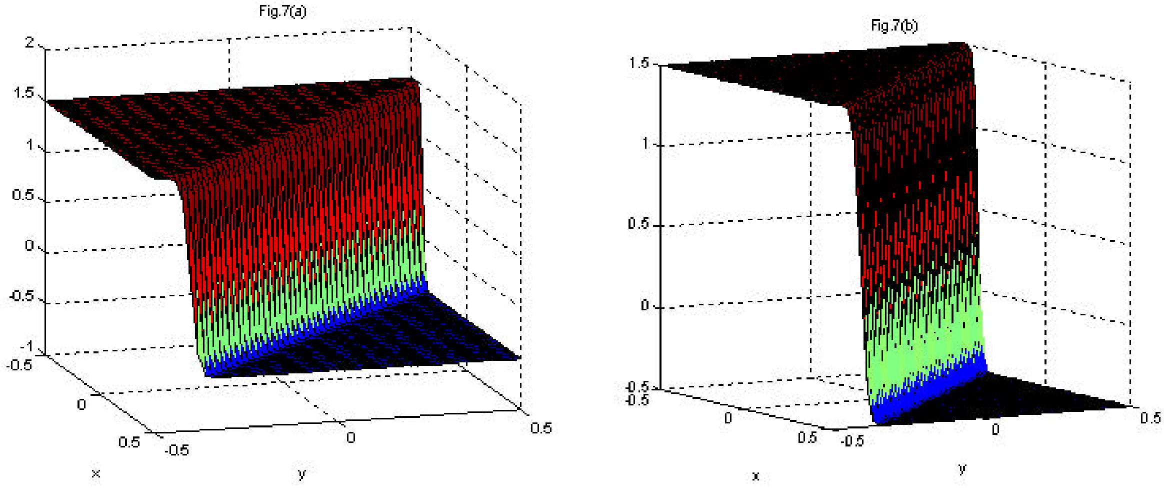

Fig. 7: Numerical (7a) and Exact (7b) solutions at N = 2500, v = 0.01, δt = 0.001 and t = 0.1 using r^7

Reynolds number up to 100 due to limited computational resources.

Test problem: Consider the two dimensional unsteady Burger's equation with a steep oblique shock defined on the domain set $-0.1 \le x \le 0.1$ and $-0.05 \le y \le 0.05$ with the following Dirichlet boundary conditions [8, 18]:

$$u(-0.1, y,\ t) = -\tanh\left(\frac{-0.1 - 0.4y}{2\nu}\right)$$

$$u(0.1, y, t) = -\tanh\left(\frac{0.1 + 0.4y}{2\nu}\right)$$

$$u(x, -0.05, t) = -\tanh\left(\frac{x + 0.02}{2\nu}\right)$$

$$u(x, 0.05, t) = -\tanh\left(\frac{x - 0.02}{2\nu}\right)$$

The shape parameter c is chosen as 0.025, v = 0.002, t = 0.1 and collocation points N = 200, 800 (grid sizes 20×10 and 40×20). Figure 8(a) and 8(b) show numerical and exact graphical solutions corresponding to collocation points N = 200 respectively. Figure 9(a) and 9(b) show numerical and exact graphical solutions corresponding to collocation points N = 800 respectively. From these figures it is clear that MQ-dependent meshfree method is capable of producing convergent and stable solution on relatively coarse grid. In [8], the authors have solved the problem with high Reynolds numbers 5000 and 10000 on finer grids 200×100×400×200 (N = 20000, 80000) respectively.

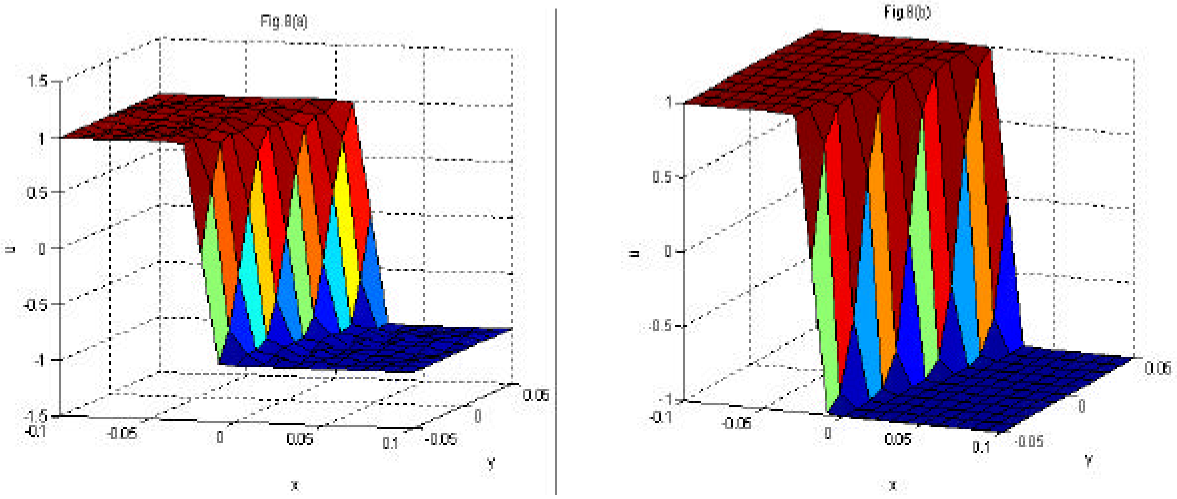

Fig. 8: Numerical (8a) and Exact (8b) solutions at c = 0.025, v = 0.002, t = 0.1 and N = 200

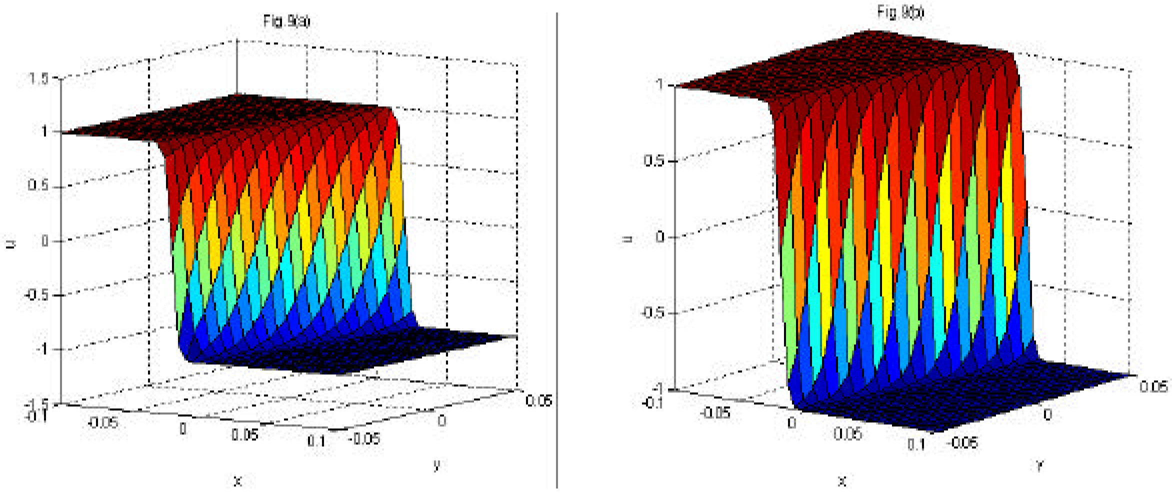

Fig. 9: Numerical (9a) and Exact (9b) solutions at c = 0.025, v = 0.002, t = 0.1 and N = 800

We have not implemented these cases because in the meshfree approach a single processor runs out of memory due to large size of collocation matrices. The graphical solution shown in Fig. 8 and 9 resembles the previous published papers [8, 18].

Test problem: We consider the Burger's equation (1) defined on the domain set $(x,y) \in [0,1] \times [0,1]$, with the periodic boundary conditions and the corresponding initial condition

$$u(x,y,0) = \sin(2\pi x)\cos(2\pi y)$$

The exact solution of this problem is unknown [8]. We compute meshfree solution by using spline basis r^7 and MQ RBFs for the total time t = 1/8, v = 0.002, δt = 0.025 and collocation points N = 2500 (grid size

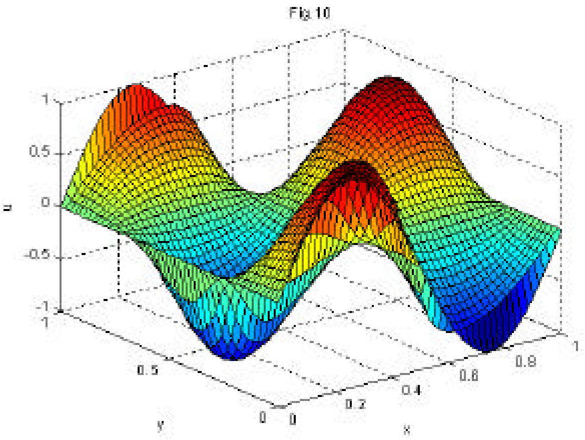

Fig. 10: Graph at v = 0.002, δt = 0.025 and t = 1/8 using r^7

Table 6: Maximum and minimum eigenvalues of the matrices R and Z for problem 4.1 corresponding to v = 0.1, N = 400 and t = 0.25 using MQ

c	Maximum eigenvalue of the matrix R	Minimum eigenvalue of the matrix R	Maximum eigenvalue of the matrix Z	Minimum eigenvalue of the matrix Z	Maximum error
0.02	9.2101×10^{2}	0	0	-9.2160×10^{2}	1.0279×10^{-2}
0.04	7.9395×10^{2}	0	0	-7.9468×10^{2}	3.3552×10^{-3}
0.06	7.7823×10^{2}	0	0	-7.7912×10^{2}	2.2099×10^{-3}
0.08	7.7369×10^{2}	0	0	-7.7470×10^{2}	1.5815×10^{-3}
0.10	7.7041×10^{2}	0	0	-7.7162×10^{2}	1.1542×10^{-3}
0.12	7.6723×10^{2}	0	0	-7.6860×10^{2}	8.6134×10^{-4}
0.14	7.6401×10^{2}	0	0	-7.6553×10^{2}	6.4385×10^{-4}
0.16	7.6071×10^{2}	0	0	-7.6237×10^{2}	4.8876×10^{-4}
0.18	7.5729×10^{2}	0	0	-7.5910×10^{2}	3.7201×10^{-4}
0.20	7.5376×10^{2}	0	0	-7.5571×10^{2}	2.8352×10^{-4}

Table 7: Maximum and minimum eigenvalues of the matrices R and Z for Problem 4.1 correspondin g to v = 0.1, N = 400 and t = 0.25 using seventh degree spline

N	Maximum eigenvalue of R	Minimum eigenvalue of R	Maximum eigenvalue of Z	Minimum eigenvalue of Z	Maximum error
25	3.7203×10	0	0	-3.7971×10	3.4323×10^{-3}
100	1.8384×10^{2}	0	0	-1.8481×10^{2}	3.1858×10^{-4}
400	7.7870×10^{2}	0	0	-7.7974×10^{2}	3.2710×10^{-5}
625	2.6278×10^{3}	0	0	-2.4601×10^{3}	1.4320×10^{-5}
1600	0	-3.8384×10^{6}	3.7524×10^{6}	0	6.4900×10^{-6}

Table 8: Condition number versus shape parameter c at t = 1.0, v = 0.1, $\delta t = 0.005$, N = 400 for problem 4.1

C	ondition num ber of the matrix T	L_∞ error	L_R error	c	ondition number of the matrix T	L_∞ error	L_R error
0.02	2.4288×10^{4}	1.2509×10^{-2}	9.6401×10^{-2}	0.22	7.6419×10^{10}	9.3660×10^{-5}	9.6764×10^{4}
0.04	7.2001×10^{4}	2.5293×10^{-3}	2.3104×10^{-2}	0.24	4.0241×10^{11}	6.4868×10^{-5}	6.7208×10^{4}
0.06	2.4473×10^{5}	1.1895×10^{-3}	1.2658×10^{-2}	0.26	2.1222×10^{12}	4.4483×10^{-5}	4.5426×10^{4}
0.08	9.4258×10^{5}	8.7006×10^{-4}	8.9349×10^{-3}	0.28	1.1194×10^{13}	3.0198×10^{-5}	2.9492×10^{4}
0.10	4.2398×10^{6}	6.3817×10^{-4}	6.5786×10^{-3}	0.30	5.8977×10^{13}	1.9876×10^{-5}	1.8009×10^{4}
0.12	2.0792×10^{7}	4.7046×10^{-4}	4.8650×10^{-3}	0.32	3.1004×10^{14}	1.2520×10^{-5}	1.0128×10^{4}
0.14	1.0488×10^{8}	3.4864×10^{-4}	3.5839×10^{-3}	0.34	1.6345×10^{15}	7.5948×10^{-6}	5.9176×10^{5}
0.16	5.3772×10^{8}	2.5584×10^{-4}	2.6229×10^{-3}	0.36	8.8764×10^{15}	7.2237×10^{-6}	5.9532×10^{5}
0.18	2.7873×10^{9}	1.8557×10^{-4}	1.9035×10^{-3}	0.38	3.6117×10^{16}	1.0213×10^{-5}	7.8407×10^{5}
0.20	1.4557×10^{10}	1.3283×10^{-4}	1.3667×10^{-3}	0.40	3.0299×10^{19}	1.9094×10^{4}	4.0603×10^{4}

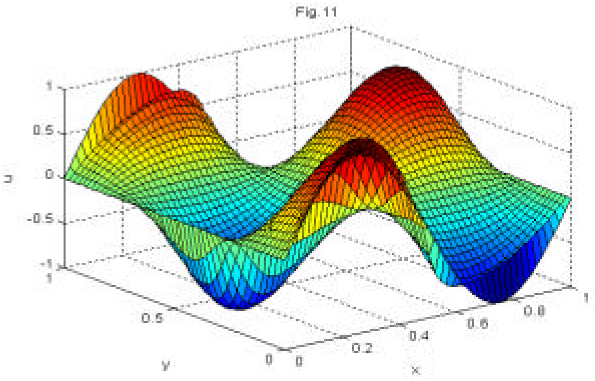

Fig. 11: Graph of Problem 4.4 at v = 0.002, $\delta t = 0.025$ and t = 1/8

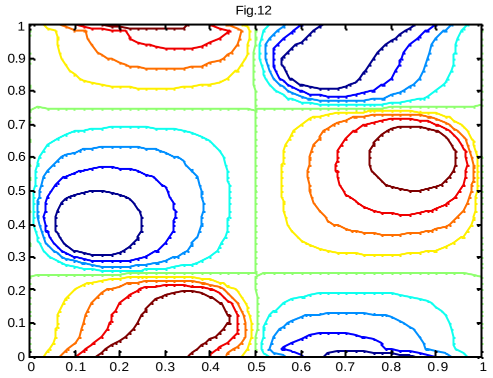

Fig. 12: Contour graph at v = 0.002, $\delta t = 0.025$ and t = 1/8 using r^{7}

Table 9: L_2 and L_∞ errors at different viscosities, c = 0.1, $\delta t = 0.001$ and t = 0.1 using MQ for problem 4.2

Reynolds's number	Collocation points N (Grid)	L_∞-error	L_R error
50	100(10×10)	6.8664×10^{-1}	2.4916×10^{-1}
	400(20×20)	5.6976×10^{-2}	1.4246×10^{-2}
	625(25×25)	1.6570×10^{-3}	1.4477×10^{-4}
	1600(40×40)	1.6751×10^{-3}	1.4478×10^{-4}
	2500(50×50)	8.5264×10^{-4}	5.1064×10^{-5}
75	100(10×10)	2.0799×10^{0}	7.6995×10^{-1}
	400(20×20)	3.4852×10^{-1}	8.6102×10^{-2}
	625(25×25)	5.6839×10^{-2}	8.5390×10^{-3}
	1600(40×40)	1.4951×10^{-2}	2.0516×10^{-3}
	2500(50×50)	8.6537×10^{-3}	5.8278×10^{-4}
100	100(10×10)	4.9638×10^{0}	1.8367×10^{0}
	400(20×20)	1.0958×10	2.1046×10^{-1}
	625(25×25)	4.2997×10^{-1}	4.3134×10^{-2}
	1600(40×40)	1.1912×10^{-1}	1.0705×10^{-2}
	2500(50×50)	5.2733×10^{-2}	3.3677×10^{-3}

Table 10: L_2 and L_∞ errors at different viscosities, $\delta t = 0.001$ and t = 0.1 using seventh degree spline for problem 4.2

Reynolds's number	Collocation points N (Grid)	L_∞-error	L_R-error
50	100(10×10)	8.3293×10^{-1}	2.9692×10^{-1}
	400(20×20)	8.7849×10^{-2}	1.6642×10^{-2}
	625(25×25)	$3.982\ \times10^{-2}$	3.6980×10^{-3}
	1600(40×40)	9.0810×10^{-3}	4.9316×10^{-4}
	2500(50×50)	4.7320×10^{-3}	2.0895×10^{-4}
75	100(10×10)	3.2960×10^{0}	1.1038×10^{0}
	400(20×20)	4.6264×10^{-1}	8.8256×10^{-2}
	625(25×25)	1.0323×10^{-1}	1.1860×10^{-2}
	1600(40×40)	2.848×10^{-2}	2.2677×10^{-3}
	2500(50×50)	1.1610×10^{-2}	6.2848×10^{-4}
100	100(10×10)	1.3610×10^{1}	4.3248×10^{0}
	400(20×20)	1.3752×10^{0}	2.3737×10^{-1}
	625(25×25)	4.9658×10^{-1}	4.7720×10^{-2}
	1600(40×40)	8.1025×10^{-2}	1.0077×10^{-2}
	2500(50×50)	3.0953×10^{-2}	2.7972×10^{-3}

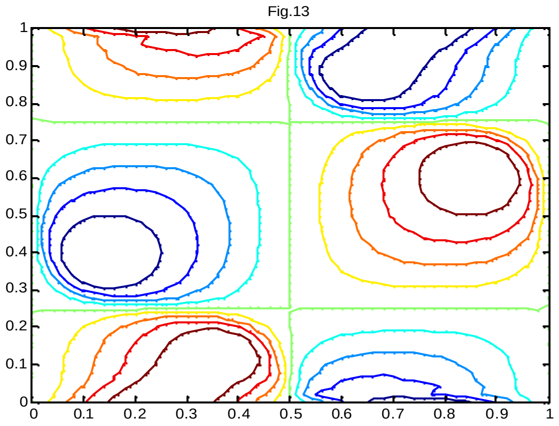

Fig. 13: Contour graph at v = 0.002, δt = 0.025 and t = 1/8 using MQ

50×50). The value of shape parameter is c = 0.025 for MQ only. Surface plots of the numerical solution are shown in Fig. 10 and 11. Contours graphs are shown in Fig. 12 and 13. It is clear from these figures that contours as well as surface plots agree with the results of the previous papers [8, 21] for small grid size.

CONCLUSIONS

A numerical scheme based on collocation method using different types of RBFs, has been presented for the solution of the two dimensional Burger's equation with different sets of initial and boundary conditions. This approach is generally more applicable than the traditional methods like finite difference and finite element methods as it can be scaled up to higher dimensions in convenient way and does not require nodal or elements connectivity. The accuracy is examined in terms of L_∞, L_R and L_2 error norms. The advantage of using spline basis (r^7) is that it is independent of shape parameter c. The problems presented in this paper suggest that meshfree approximation methods should be considered as one of possible ways of solving these kinds of nonlinear partial differential equations.

ACKNOWLEDGMENT

The first author is thankful to University of Engineering & Technology Peshawar (Mardan Campus) for providing research facilities.

REFERENCES

1. Burger, J.M., 1948. A mathematical model illustrating the theory of turbulence. Adv. Appl. Mech., I: 171-199.
2. Arminjon, P. and C. Beauchamp, 1979. Numerical Solution of Burger's Equations in Two-Space Dimensions. Computer Methods in Appl. Mech. and Eng., 19 (3): 351-365.
3. Bahadir, A.R., 2003. A fully implicit finite-difference scheme for two-dimensional Burger's equations. Appl. Math. Comput., 137: 131-137.
4. Dehghan, M., 2007. Time-splitting procedures for the solution of the two-dimensional transport equation. Kybernetes, 36: 791-805.
5. Fletcher, J.D., 1983. Generating exact solutions of the two-dimensional Burger's equations. Int. J. Numer. Meth. Fluids, 3: 213-216.
6. Fletcher, C.A.J., 1983. A comparison of finite element and finite difference solution of the one-and two-dimensional Burger's equations. J. Comput. Phys., 51: 159-188.

7. Radwan, S.F., 2005. Comparison of higher order accurate schemes for solving two dimensional unsteady Burger's equation. J. Comput. Appl. Math., 174: 383-397.
8. Khater, A.H., R.S. Temsah and M.M. Hassan, 2007. A Chebyshev spectral collocation method for solving Burger's-type equations. J. Comput. Appl. Math.
9. Velivelli, A.C. and K.M. Bryden, 2006. Parallel performance and accuracy of lattice Boltzmann and traditional finite difference methods for solving the unsteady two dimensional Burger's equation. Physica A, 362: 139-145.
10. Nee, J. and J. Duan, 1998. Limit set of trajectories of the coupled viscous Burgers equations. Appl. Math. Lett., 11 (1): 57-61.
11. Zhang, J. and G. Yan, 2008. Lattice Boltzmann method for one and two dimensional Burger's equation. Physica A, 387: 4771-4786.
12. Kansa, E.J., 1990. Multiquadrics scattered data approximation scheme with applications to Computational fluid-dynamics I, surface approximations and partial derivative estimates. Comput. Math. Appl., 19: 127-145.
13. Li, J., Y. Chen and D. Pepper, 2003. Radial basis function method for 1-D and 2-D groundwater contaminant transport modeling. Comput. Mech., 32: 10-15.
14. Fasshauer, G.E., 1996. Solving partial differential equations by collocation with radial basis functions. In: Me´chaute´, A.L. (Ed.). Proceedings of Chamonix, Vanderbilt University Press, Nashville, TN, pp: 1-8.
15. Siraj-ul-Islam, Sirajul Haq and Arshed Ali, 2008. A meshfree method for the numerical solution of RLW equation. J. Comp. Appl. Math, doi: 10.1016.
16. Sara, S.A., 2006. Integrated Multiquadric Radial Basis Function Approximation Methods. Comp. Math. Appl., 51: 1283-1296.
17. Carlson, R.E. and T.A. Foley, 1991. The parameter R^2 in multiquadric interpolation. Comput. Math. Appl., 21: 29-42.
18. Tarwater, A.E., 1985. A parameter study of Hardy's multiquadric method for scattered data interpolation. Lawrence Livermore National Laboratory, Technical Report UCRL-54670.
19. Cheng, A.H.D., M.A. Golberg, E.J. Kansa and G. Zammito, 2003. Exponential convergence and H-c multiquadric collocation method for partial differential equations. Numer. Methods for Partial Diff. Eq., 19: 571-594.
20. Hon, Y.C. and R. Schaback, 2001. On unsymmetric collocation by radial basis functions. Appl. Math. Comput., 119: 177-186.
21. Cole, J.D., 1951. On a quasilinear parabolic equations occurring in aerodynamics. Quart. Appl. Math., 9: 225-236.