



# The Lie-group method based on radial basis functions for solving nonlinear high dimensional generalized Benjamin–Bona–Mahony–Burgers equation in arbitrary domains



M. Hajiketabi<sup>a,\*</sup>, S. Abbasbandy<sup>a</sup>, F. Casas<sup>b</sup>

<sup>a</sup> Department of Mathematics, Imam Khomeini International University, Ghazvin 34149-16818, Iran

<sup>b</sup> Departament de Matemàtiques and IMAC, Universitat Jaume I, 12071, Castellón, Spain

## ARTICLE INFO

### Keywords:

Nonlinear generalized Benjamin–Bona–Mahony–Burgers (GBBMB) equation  
Kansas approach  
Meshless method  
Radial basis functions (RBFs)  
Group preserving scheme (GPS)  
Non-regular geometrical domains

## ABSTRACT

The aim of this paper is to introduce a new numerical method for solving the nonlinear generalized Benjamin–Bona–Mahony–Burgers (GBBMB) equation. This method is combination of group preserving scheme (GPS) with radial basis functions (RBFs), which takes advantage of two powerful methods, one as geometric numerical integration method and the other meshless method. Thus, we introduce this method as the Lie-group method based on radial basis functions (LG–RBFs). In this method, we use Kansas approach to approximate the spatial derivatives and then we apply GPS method to approximate first-order time derivative. One of the important advantages of the developed method is that it can be applied to problems on arbitrary geometry with high dimensions. To demonstrate this point, we solve nonlinear GBBMB equation on various geometric domains in one, two and three dimension spaces. The results of numerical experiments are compared with analytical solutions and the method presented in Dehghan et al. (2014) to confirm the accuracy and efficiency of the presented method.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

In mathematics and physics, nonlinear partial differential equations are partial differential equations with nonlinear terms. As said in [1,2], many phenomena in engineering and applied sciences are modeled by nonlinear evolution equations. They describe many different physical systems, ranging from gravitation to fluid dynamics, and have been used in mathematics to solve problems such as the Poincaré conjecture and the Calabi conjecture. They are hard to study: there are almost no generic methods that work for all such equations, and usually each particular equation has to be studied as a separate problem. The science of exact form solutions of the nonlinear partial differential equations helps the experiment of numerical solvers, aids in the stability analysis of solutions and guides to a better conception of nonlinear phenomena that these equations model [3]. But finding an exact solution for nonlinear PDEs is difficult. Therefore, many researchers are studying numerical methods for solving the nonlinear PDEs.

\* Corresponding author.

E-mail addresses: [marcela.agudelo@insp.mx](mailto:marcela.agudelo@insp.mx), [mohammadhajiketabi@gmail.com](mailto:mohammadhajiketabi@gmail.com) (M. Hajiketabi).

One particular class of such numerical methods that have been highly regarded in recent years is formed by geometric numerical integration methods [17,18]. The concept of a Lie group is very helpful in creating some trusty and potent numerical approaches to integrate ODEs, whilst preserving their invariant features. Indeed, preserving the Lie group structure under discretization plays an serious role in the improvement of a appropriate behavior in the error minimization [19]. So, by subscription the geometric structure and invariance of the original ODEs, new techniques can be devised, which are more strong, accurate and stable than the formal numerical approaches [16]. From the class of numerical Lie group methods, the so-called group preserving scheme (GPS), introduced by Liu [20], is a numerical approach which utilizes the Cayley transformation and the Padé approximations in the Minkowski space. Avoiding the spurious solutions and ghost fixed points are main benefits of this method [21–26].

In recent years, different techniques have been employed to apply the GPS for solving partial differential equations [27–32]. In most of these methods, using a technique known as the method of lines, all variables in the problem except one, are discrete and a system of ordinary differential equations is produced. Then, the GPS is applied for solving this system of ordinary differential equations. But some problems may arise in this process. The method is very sensitive in selecting the size of discrete step, this means that many criteria including stability, consistency and etc. should be considered. So, if the size of the discrete step selected is very small or large the consistency and stability of the method can be problematic and the solution will not be reliable. On the other hand, if for obtaining good results we select the length of discrete step too small, this will result in a large number of ordinary differential equations. Thus the method requires a lot of calculations. With these explanations, implementation of GPS on complex non-linear partial differential equations such as the nonlinear GBBMB equation is virtually impossible. For dealing with these difficulties, in this paper we propose a technique that combines the GPS with radial basis functions. We have named this scheme the Lie-group method based on radial basis functions.

### 1.1. A brief review of the GBBMB equation

In this paper, we consider the nonlinear GBBMB equation of the following form

$$u_t - \Delta u_t - \Delta u + \nabla \cdot u = \nabla \cdot (F(u)) + f, \quad \mathbf{x} \in \Omega, \quad 0 \leq t \leq T, \quad (1)$$

with initial condition

$$u(\mathbf{x}, 0) = \omega(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2)$$

and boundary condition

$$u(\mathbf{x}, t) = h(t), \quad \mathbf{x} \in \partial\Omega, \quad (3)$$

where  $F(u)$  is a nonlinear function,  $f$  is a function of space and time variables,  $\Omega \subseteq \mathbb{R}^n$ ,  $\Delta$  and  $\nabla$  are the Laplacian and gradient operators, respectively. By choosing  $F(u) = \frac{1}{2}u^2$ , Eq. (1) is called the Benjamin–Bona–Mahony–Burgers (BBMB) equation. If remove the term  $\Delta u$ , we have the generalized Benjamin–Bona–Mahony (GBBM) equation. If we remove the term  $\Delta u$  and also put  $F(u) = \frac{1}{2}u^2$ , Eq. (1) is called the Benjamin–Bona–Mahony (BBM) equation.

Among the applications of BBM equation we can mention the following [4–6]:

- analysis of the surface waves of long wavelength in liquids;
- hydromagnetic waves in cold plasma;
- acoustic-gravity waves in compressible fluids;
- acoustic waves in harmonic crystals.

Different analytical and numerical methods have been used for solving the BBMB equation in the literature. Here, we review some of them. The authors in [4] use the first integral method for getting an analytic solution of the modified BBM equation. The BBM equation is solved using a generalization of the well-known tanh-coth method by Gómez et al. [7]. The nonlinear stability of planar viscous shock profiles of the GBBMB equation in two dimensions is analyzed in [8]. Guo and Fang [9] studied the optimal decay rates of solutions for the GBBMB equation in multi-dimensional space  $n \geq 3$ . Authors of [10] showed an exponential decay rate of the solutions to the Cauchy problem of the GBBMB equation by employing the space-time weighted energy method. The BBM and modified BBM equations are solved using the exp-function method to obtain some new soliton solutions by Noor et al. [11]. Authors in [12] studied the convergence rate of the global solutions of the GBBMB equation to the corresponding degenerate boundary layer solutions in the half-space. A Crank–Nicolson-type finite difference scheme with the proof of the second-order convergence in the discrete  $H^1$ -norm is discussed in [13] for solving the BBM equation. Omrani [14] presented a technique based on the standard Galerkin method and Crank–Nicolson formula for space and time variables, respectively, and he proved the convergence of a linearized Galerkin modified scheme. The interested readers are referred to [5,6,15].

## 2. Group preserving scheme

Liu [20] has derived a Lie group transformation for the augmented dynamical system on the future cone, and developed the group preserving scheme for an effective numerical solution of nonlinear ODEs. Consider a system of  $n$  ordinary differential equations:

$$\mathbf{u}'(x) = \mathbf{f}(\mathbf{u}(x), x), \quad \mathbf{u}(x) \in \mathbb{R}^n, \quad x \in \mathbb{R}, \quad (4)$$

where  $\mathbf{u}(x)$  is an  $n$ -dimensional vector,  $x$  is an independent variable and  $\mathbf{f}$  is a vector-valued function of  $\mathbf{u}$  and  $x$  (the vector field). We can transform Eq. (4) into the following  $(n + 1)$ -dimensional augmented dynamical system:

$$\frac{d}{dx} \begin{bmatrix} \mathbf{u} \\ \|\mathbf{u}\| \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n \times n} & \frac{\mathbf{f}(\mathbf{u}, x)}{\|\mathbf{u}\|} \\ \frac{\mathbf{f}^T(\mathbf{u}, x)}{\|\mathbf{u}\|} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \|\mathbf{u}\| \end{bmatrix}. \tag{5}$$

Here we assume  $\|\mathbf{u}\| > 0$ , and hence the above system is well-defined.

Obviously, the first equation in Eq. (5) is the same as the original Eq. (4), but the addition of the second equation gives us a Minkowski structure in the set of the augmented state variables of  $\mathbf{U} = (\mathbf{u}^T, \|\mathbf{u}\|)^T$ , satisfying the cone condition

$$\mathbf{U}^T \mathbf{g} \mathbf{U} = 0, \tag{6}$$

where

$$\mathbf{g} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times n} & -1 \end{bmatrix}, \tag{7}$$

is a Minkowski metric,  $\mathbf{I}_n$  is the identity matrix, and the superscript  $T$  stands for the transpose. In terms of  $(\mathbf{u}, \|\mathbf{u}\|)$ , Eq. (6) becomes

$$\mathbf{U}^T \mathbf{g} \mathbf{U} = \mathbf{u} \cdot \mathbf{u} - \|\mathbf{u}\|^2 = \|\mathbf{u}\|^2 - \|\mathbf{u}\|^2 = 0, \tag{8}$$

where the dot between two  $n$ -dimensional vectors denotes the Euclidean inner product. The cone condition is thus the most natural constraint that we can impose on the dynamical system (5).

Consequently, we have a  $n + 1$ -dimensional augmented differential system,

$$\mathbf{U}' = \mathbf{A} \mathbf{U}, \tag{9}$$

with a constraint (6), where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{n \times n} & \frac{\mathbf{f}(\mathbf{u}, x)}{\|\mathbf{u}\|} \\ \frac{\mathbf{f}^T(\mathbf{u}, x)}{\|\mathbf{u}\|} & 0 \end{bmatrix}, \tag{10}$$

satisfying

$$\mathbf{A}^T \mathbf{g} + \mathbf{g} \mathbf{A} = \mathbf{0}, \tag{11}$$

is an element of the Lie algebra  $\mathfrak{so}(n, 1)$  of the proper orthochronous Lorentz group  $SO_0(n, 1)$ . Notice that, although not stated explicitly, the matrix  $\mathbf{A}$  depends indeed on  $\mathbf{U}$  through  $\mathbf{u}$  and  $\|\mathbf{u}\|$ .

Once the problem is formulated in  $SO_0(n, 1)$ , it is crucial that any numerical scheme used to solve (9) preserve the Lie group structure. Otherwise, the cone condition (6) is not satisfied and the numerical approximation is no longer consistent. This can be done if one is able to generate approximations  $\mathbf{U}_k$  by

$$\mathbf{U}_{k+1} = \mathbf{G}(k) \mathbf{U}_k, \tag{12}$$

where  $\mathbf{U}_k$  denotes the numerical value of  $\mathbf{U}$  at the discrete  $x_k$ , and  $\mathbf{G}(k) \in SO_0(n, 1)$  verifies

$$\mathbf{G}^T \mathbf{g} \mathbf{G} = \mathbf{g}, \tag{13}$$

$$\det \mathbf{G} = 1, \tag{14}$$

$$G_0^0 > 0, \tag{15}$$

where  $G_0^0 > 0$  is the 00th component of  $\mathbf{G}$ .

A first-order approximation is obtained by taking

$$\mathbf{U}_{k+1} = \exp \left[ \int_{x_k}^{x_{k+1}} \mathbf{A} dx \right] \mathbf{U}_k. \tag{16}$$

Liu, in [20], considers the Cayley transformation and Padé approximants. Here we propose instead the Lie-group version of the simple forward Euler method, namely

$$\mathbf{U}_{k+1} = \exp[\Delta x \mathbf{A}(k)] \mathbf{U}_k. \tag{17}$$

Since  $\mathbf{A}(k) \in \mathfrak{so}(n, 1)$  and  $\mathbf{U}_k \in SO_0(n, 1)$ , then  $\mathbf{U}_{k+1} \in SO_0(n, 1)$  by construction. The exponential mapping  $\exp[\Delta x \mathbf{A}(k)]$  admits the closed-form representation

$$\exp[\Delta x \mathbf{A}(k)] = \begin{bmatrix} \mathbf{I}_n + \frac{(\alpha_k - 1)}{\|\mathbf{f}_k\|^2} \mathbf{f}_k \mathbf{f}_k^T & \frac{\beta_k \mathbf{f}_k}{\|\mathbf{f}_k\|} \\ \frac{\beta_k \mathbf{f}_k^T}{\|\mathbf{f}_k\|} & \alpha_k \end{bmatrix}, \quad (18)$$

where

$$\alpha_k = \cosh\left(\frac{\Delta x \|\mathbf{f}_k\|}{\|\mathbf{u}_k\|}\right), \quad (19)$$

and

$$\beta_k = \sinh\left(\frac{\Delta x \|\mathbf{f}_k\|}{\|\mathbf{u}_k\|}\right). \quad (20)$$

For notational convenience, we have used  $\mathbf{f}_k = \mathbf{f}(\mathbf{u}_k, x_k)$ . Substituting the above  $\exp[\Delta x \mathbf{A}(k)]$  for  $\mathbf{G}(k)$  into Eq. (12), we obtain

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \frac{(\alpha_k - 1)\mathbf{f}_k \cdot \mathbf{u}_k + \beta_k \|\mathbf{u}_k\| \|\mathbf{f}_k\|}{\|\mathbf{f}_k\|^2} \mathbf{f}_k = \mathbf{u}_k + \eta_k \mathbf{f}_k, \quad (21)$$

$$\|\mathbf{u}_{k+1}\| = \frac{\beta_k (\mathbf{f}_k \cdot \mathbf{u}_k) + \alpha_k \|\mathbf{u}_k\| \|\mathbf{f}_k\|}{\|\mathbf{f}_k\|}. \quad (22)$$

This scheme therefore preserves group properties for all  $\Delta x > 0$ . In the practical numerical calculation, we only need Eq. (21) such that by knowing the initial value  $\mathbf{u}_0$  can obtain  $\mathbf{u}_k$ ,  $k = 1, 2, 3, \dots$

**Theorem 1.** The cone condition is preserved by Eqs. (21) and (22) for every  $\Delta x$ .

**Proof.** As a matter of fact, this is a trivial consequence of the fact that  $\mathbf{U}_k^T \mathbf{g} \mathbf{U}_k = 0$  and (12), with  $\mathbf{G}(k) = \exp[\Delta x \mathbf{A}(k)]$ . Alternatively, one can show that

$$\mathbf{u}_{k+1} \cdot \mathbf{u}_{k+1} - \|\mathbf{u}_{k+1}\|^2 = 0 \quad (23)$$

as follows. From Eq. (21), we have

$$\mathbf{u}_{k+1} \cdot \mathbf{u}_{k+1} = \left( \mathbf{u}_k + \frac{(\alpha_k - 1)\mathbf{f}_k \cdot \mathbf{u}_k + \beta_k \|\mathbf{u}_k\| \|\mathbf{f}_k\|}{\|\mathbf{f}_k\|^2} \mathbf{f}_k \right) \cdot \left( \mathbf{u}_k + \frac{(\alpha_k - 1)\mathbf{f}_k \cdot \mathbf{u}_k + \beta_k \|\mathbf{u}_k\| \|\mathbf{f}_k\|}{\|\mathbf{f}_k\|^2} \mathbf{f}_k \right), \quad (24)$$

after doing the above inner product and possible simplifications, we obtain

$$\mathbf{u}_{k+1} \cdot \mathbf{u}_{k+1} = \frac{(1 + \beta_k^2) \|\mathbf{u}_k\|^2 \|\mathbf{f}_k\|^2 + (\alpha_k^2 - 1) (\mathbf{f}_k \cdot \mathbf{u}_k)^2 + 2\alpha_k \beta_k (\mathbf{f}_k \cdot \mathbf{u}_k) \|\mathbf{u}_k\| \|\mathbf{f}_k\|}{\|\mathbf{f}_k\|^2}. \quad (25)$$

Therefore, according to definition  $\alpha_k$  and  $\beta_k$  from Eqs. (19) and (20) we obtain

$$\begin{aligned} \mathbf{u}_{k+1} \cdot \mathbf{u}_{k+1} &= \frac{\alpha_k^2 \|\mathbf{u}_k\|^2 \|\mathbf{f}_k\|^2 + \beta_k^2 (\mathbf{f}_k \cdot \mathbf{u}_k)^2 + 2\alpha_k \beta_k (\mathbf{f}_k \cdot \mathbf{u}_k) \|\mathbf{u}_k\| \|\mathbf{f}_k\|}{\|\mathbf{f}_k\|^2} \\ &= \frac{(\alpha_k \|\mathbf{u}_k\| \|\mathbf{f}_k\| + \beta_k (\mathbf{f}_k \cdot \mathbf{u}_k))^2}{\|\mathbf{f}_k\|^2} = \|\mathbf{u}_{k+1}\|^2, \end{aligned} \quad (26)$$

where the last equal is obtained from Eq. (22). Thus, the proof is finished.  $\square$

**Theorem 2.** Scheme (21) unconditionally preserves the fixed points of the differential equation (4) and their stability properties.

**Proof.** First, we prove  $\eta_k = \eta(x_k) > 0$  for each  $x_k$ . From  $\alpha_k > 1$ ,  $\forall \Delta x > 0$  and  $-\|\mathbf{f}_k\| \|\mathbf{u}_k\| \leq \mathbf{f}_k \cdot \mathbf{u}_k \leq \|\mathbf{f}_k\| \|\mathbf{u}_k\|$ , we can prove that

$$\frac{\|\mathbf{u}_k\|}{\|\mathbf{f}_k\|} \left[ \exp\left(\frac{\Delta x \|\mathbf{f}_k\|}{\|\mathbf{u}_k\|}\right) - 1 \right] \geq \eta_k \geq \frac{\|\mathbf{u}_k\|}{\|\mathbf{f}_k\|} \left[ 1 - \exp\left(-\frac{\Delta x \|\mathbf{f}_k\|}{\|\mathbf{u}_k\|}\right) \right] > 0, \quad \forall \Delta x > 0. \quad (27)$$

Therefore, it is obvious that

$$\mathbf{u}_{k+1} = \mathbf{u}_k \iff \mathbf{f}_k = \mathbf{0}. \quad (28)$$

This means that  $\mathbf{u}_k$  is a fixed point of the discretized mapping (21) if and only if the point  $\mathbf{u}_k$  is an equilibrium (critical, fixed) point of the system (4). We next check the property of the fixed point. The Jacobian of the map (21) is

$$J := \frac{\partial \mathbf{u}_{k+1}}{\partial \mathbf{u}_k} = \mathbf{I}_n + \mathbf{f}_k \left( \frac{\partial \eta_k}{\partial \mathbf{u}_k} \right)^T + \eta_k \frac{\partial \mathbf{f}_k}{\partial \mathbf{u}_k}. \quad (29)$$

At the fixed point  $\mathbf{f}_k = 0$ , we thus have

$$J = \mathbf{I}_n + \eta_k \frac{\partial \mathbf{f}_k}{\partial \mathbf{u}_k}. \tag{30}$$

Recalling that  $\eta_k > 0$ , the property of the fixed point is not altered by the map (21). More precisely, the map (21) has the same type of stability that the system (4) has.  $\square$

According to the preceding theorems, the long-term behavior of the original system can be described very well by this numerical scheme.

### 3. The LG-RBFs for solving GBBMB equation

#### 3.1. 1-dimensional case

Clearly,  $\Omega$  is now an arbitrary interval in  $\mathbb{R}$ . We consider the following expansion of  $u(x, t)$  [33–41]:

$$u(x, t) = \sum_{j=1}^M c_j(t)\varphi(r_j) + c_{M+1}(t)x + c_{M+2}(t), \tag{31}$$

in which

$$\varphi(r_j) = \sqrt{(x - x_j)^2 + c^2}, \tag{32}$$

where  $r_j = \{\|x - x_j\|, x_j \in \Omega\}$  denotes the Euclidean distance between  $x$  and  $x_j$  ( $\{x_j\}_{j=1}^M$  are usually called center nodes). As mentioned in [42] the accuracy of many schemes for interpolating scattered data with radial basis functions depends on a shape parameter  $c$  of the radial basis function. Most authors use the trial and error method for obtaining a good shape parameter that results in best accuracy. In this paper we also use from this method.

In Eq. (31) we have  $M + 2$  unknown variables. Besides the  $M$  equations resulting from collocating Eq. (31) at  $M$  points  $\{x_i\}_{i=1}^M$ , two additional equations are required that add two condition as follows:

$$\sum_{j=1}^M c_j^n = \sum_{j=1}^M c_j^n x_j = 0, \tag{33}$$

where  $c_j^n \equiv c_j(t_n)$ . For the use of Kansas method, we let  $\{x_i\}_{i=1}^M$  be  $M$  collocation points in  $\Omega$  in which  $\{x_i\}_{i=2}^{M-1}$  are interior points and  $x_1$  and  $x_M$  are boundary points. For each point  $x_i$ , let us denote

$$\varphi(r_{ij}) = \sqrt{(x_i - x_j)^2 + c^2} = \sqrt{r^2 + c^2}. \tag{34}$$

By substituting Eq. (31) for interior points in Eq. (1) we obtain

$$\begin{aligned} & \frac{\partial}{\partial t} \left( \sum_{j=1}^M c_j(t)\varphi(r_{ij}) + c_{M+1}(t)x_i + c_{M+2}(t) - \sum_{j=1}^M c_j(t)\Delta\varphi(r_{ij}) \right) \\ &= \sum_{j=1}^M c_j(t)\Delta\varphi(r_{ij}) - \sum_{j=1}^M c_j(t)\nabla\varphi(r_{ij}) - c_{M+1}(t) \\ &+ \nabla \left( F \left( \sum_{j=1}^M c_j(t)\varphi(r_{ij}) + c_{M+1}(t)x_i + c_{M+2}(t) \right) \right) + f(x_i, t), \quad i = 2, \dots, M - 1, \end{aligned} \tag{35}$$

where  $\nabla\varphi(r_{ij}) = \{\varphi'(r_j)\}|_{x=x_i}$  and  $\Delta\varphi(r_{ij}) = \{\varphi''(r_j)\}|_{x=x_i}$ . Now by defining the following vectors

$$\begin{aligned} \mathbf{u}(t) &= \left[ \sum_{j=1}^M c_j(t)\varphi(r_{ij}) + c_{M+1}(t)x_i + c_{M+2}(t) - \sum_{j=1}^M c_j(t)\Delta\varphi(r_{ij}) \right]_{i=2}^{M-1}, \\ \mathbf{f}(t, \mathbf{u}(t)) &= \left[ \sum_{j=1}^M c_j(t)\Delta\varphi(r_{ij}) - \sum_{j=1}^M c_j(t)\nabla\varphi(r_{ij}) - c_{M+1}(t) \right. \\ &\quad \left. + \nabla \left( F \left( \sum_{j=1}^M c_j(t)\varphi(r_{ij}) + c_{M+1}(t)x_i + c_{M+2}(t) \right) \right) + f(x_i, t) \right]_{i=2}^{M-1}, \end{aligned} \tag{36}$$

Eq. (35) is converted in  $\mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t))$ , i.e., an equation of the form (4). Now, we use the GPS (21) and solve Eq. (35) respect to  $t$ . We define

$$t_k = n\tau, \quad n = 0, 1, 2, \dots, N, \tag{37}$$

where  $\tau = T/N$  is the step size of time variable and  $T$  is the final time. By using Eq. (21) for  $n = 0$  we get the following form:

$$\sum_{j=1}^M c_j^1 \varphi(r_{ij}) + c_{M+1}^1 x_i + c_{M+2}^1 - \sum_{j=1}^M c_j^1 \Delta \varphi(r_{ij}) = (\omega(x_i) - \omega''(x_i)) + \eta_0 \mathbf{f}_0, \quad i = 2, \dots, M - 1, \tag{38}$$

where  $\mathbf{f}_0 = [\omega''(x_i) - \omega'(x_i) + F'(\omega(x_i)) + f(x_i, 0)]_{i=2}^{M-1}$  and  $\eta_0$  is calculated from Eq. (21). For  $1 \leq n \leq N - 1$  we have

$$\begin{aligned} & \sum_{j=1}^M c_j^{n+1} \varphi(r_{ij}) + c_{M+1}^{n+1} x_i + c_{M+2}^{n+1} - \sum_{j=1}^M c_j^{n+1} \Delta \varphi(r_{ij}) \\ &= \left( \sum_{j=1}^M c_j^n \varphi(r_{ij}) + c_{M+1}^n x_i + c_{M+2}^n - \sum_{j=1}^M c_j^n \Delta \varphi(r_{ij}) \right) + \eta_n \mathbf{f}_n, \quad i = 2, \dots, M - 1, \end{aligned} \tag{39}$$

where

$$\mathbf{f}_n = \left[ \sum_{j=1}^M c_j^n \Delta \varphi(r_{ij}) - \sum_{j=1}^M c_j^n \nabla \varphi(r_{ij}) - c_{M+1}^n + \nabla \left( F \left( \sum_{j=1}^M c_j^n \varphi(r_{ij}) + c_{M+1}^n x_i + c_{M+2}^n \right) \right) + f(x_i, t_n) \right]_{i=2}^{M-1},$$

and  $\eta_n$  is calculated from Eq. (21). Using Eqs. (33), (38), (39) and condition boundary (3) we have the following matrix form:

$$\mathbf{A} \mathbf{c}^{n+1} = \mathbf{B}^{n+1}, \tag{40}$$

where

$$\mathbf{A} = \begin{bmatrix} \varphi(r_{1,1}) & \varphi(r_{1,2}) & \cdots & \varphi(r_{1,M-1}) & \varphi(r_{1,M}) & x_1 & 1 \\ \Pi(\varphi(r_{2,1})) & \Pi(\varphi(r_{2,2})) & \cdots & \Pi(\varphi(r_{2,M-1})) & \Pi(\varphi(r_{2,M})) & x_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \Pi(\varphi(r_{M-1,1})) & \Pi(\varphi(r_{M-1,2})) & \cdots & \Pi(\varphi(r_{M-1,M-1})) & \Pi(\varphi(r_{M-1,M})) & x_{M-1} & 1 \\ \varphi(r_{M,1}) & \varphi(r_{M,2}) & \cdots & \varphi(r_{M,M-1}) & \varphi(r_{M,M}) & x_M & 1 \\ x_1 & x_2 & \cdots & x_{M-1} & x_M & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 & 0 & 0 \end{bmatrix}, \tag{41}$$

is  $(M + 2) \times (M + 2)$  matrix, the operator  $\Pi$  is

$$\Pi(\varphi(r_{i,j})) = \{\varphi(r_j) - \Delta \varphi(r_j)\} |_{x=x_i}, \quad 2 \leq i \leq M - 1, \quad 1 \leq j \leq M,$$

also we have

$$\mathbf{c}^{n+1} = [c_1^{n+1} \quad c_2^{n+1} \quad \cdots \quad c_M^{n+1} \quad c_{M+1}^{n+1} \quad c_{M+2}^{n+1}]^T, \quad \mathbf{B}^{n+1} = [b_1^{n+1} \quad b_2^{n+1} \quad \cdots \quad b_M^{n+1} \quad 0 \quad 0]^T, \tag{42}$$

in which

$$b_i^1 = \{\omega(x) - \omega''(x) + \eta_0 \mathbf{f}_0\} |_{x=x_i}, \quad 1 < i < M, \tag{43}$$

and also we obtain

$$b_i^{n+1} = \sum_{j=1}^M c_j^n \varphi(r_{ij}) + c_{M+1}^n x_i + c_{M+2}^n - \sum_{j=1}^M c_j^n \Delta \varphi(r_{ij}) + \eta_n \mathbf{f}_n, \quad 1 < i < M, \quad 1 \leq n \leq N - 1, \tag{44}$$

and finally we can write

$$b_1^{n+1} = h(a, t_{n+1}), \quad b_M^{n+1} = h(b, t_{n+1}), \quad 0 \leq n \leq N - 1. \tag{45}$$

After solving the algebraic system of equations  $\mathbf{A} \mathbf{c}^{n+1} = \mathbf{B}^{n+1}$  at each time step, we can construct the solution using Eq. (31). We use the LU decomposition method for solving linear algebraic system of equations  $\mathbf{A} \mathbf{c}^{n+1} = \mathbf{B}^{n+1}$ .

### 3.2. n-Dimensional case ( $n \geq 2$ )

Now  $\Omega$  is an arbitrary interval in  $\mathbb{R}^n$ ,  $n \geq 2$ . For  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ , we construct an approximate expansion of  $u(\mathbf{x}, t)$  of the form

$$u(\mathbf{x}, t) = \sum_{j=1}^M c_j(t) \varphi(\mathbf{r}_j), \tag{46}$$

in which

$$\varphi(\mathbf{r}_j) = \sqrt{(\mathbf{x} - \mathbf{x}_j)^2 + c^2}, \tag{47}$$

where  $\mathbf{r}_j = \{\|\mathbf{x} - \mathbf{x}_j\|, \mathbf{x}_j \in \Omega\}$  denotes the Euclidean distance between  $\mathbf{x}$  and  $\mathbf{x}_j$ . For the use of Kansas method, we let  $\{\mathbf{x}_i\}_{i=1}^M$  be  $M$  collocation points in  $\Omega$  in which  $\{\mathbf{x}_i\}_{i=1}^{M_l}$  are boundary points and  $\{\mathbf{x}_i\}_{i=M_l+1}^M$  are interior points. For each point  $\mathbf{x}_i$ , let us denote

$$\varphi(\mathbf{r}_{ij}) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^2 + c^2} = \sqrt{\mathbf{r}^2 + c^2}. \tag{48}$$

By substituting Eq. (46) for interior points in Eq. (1) we obtain

$$\begin{aligned} & \frac{\partial}{\partial t} \left( \sum_{j=1}^M c_j(t) \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j(t) \Delta \varphi(\mathbf{r}_{ij}) \right) \\ &= \sum_{j=1}^M c_j(t) \Delta \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j(t) \nabla \varphi(\mathbf{r}_{ij}) + \nabla \left( F \left( \sum_{j=1}^M c_j(t) \varphi(\mathbf{r}_{ij}) \right) \right) \\ &+ f(\mathbf{x}_i, t), \quad i = M_l + 1, \dots, M, \end{aligned} \tag{49}$$

where  $\nabla \varphi(\mathbf{r}_{ij}) = \{\nabla \varphi(\mathbf{r}_j)\} |_{\mathbf{x}=\mathbf{x}_i}$  and  $\Delta \varphi(\mathbf{r}_{ij}) = \{\Delta \varphi(\mathbf{r}_j)\} |_{\mathbf{x}=\mathbf{x}_i}$ . Now by definition the following vectors

$$\begin{aligned} \mathbf{u}(t) &= \left[ \sum_{j=1}^M c_j(t) \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j(t) \Delta \varphi(\mathbf{r}_{ij}) \right]_{i=M_l+1}^M, \\ \mathbf{f}(t, \mathbf{u}(t)) &= \left[ \sum_{j=1}^M c_j(t) \Delta \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j(t) \nabla \varphi(\mathbf{r}_{ij}) + \nabla \left( F \left( \sum_{j=1}^M c_j(t) \varphi(\mathbf{r}_{ij}) \right) \right) + f(\mathbf{x}_i, t) \right]_{i=M_l+1}^M, \end{aligned} \tag{50}$$

Eq. (49) has the form of Eq. (4). We use the group preserving scheme (21) and solve Eq. (49) respect to  $t$ . We define

$$t_k = n\tau, \quad n = 0, 1, 2, \dots, N, \tag{51}$$

where  $\tau = T/N$  is the step size of time variable and  $T$  is the final time. For  $n = 0$  we get the following form:

$$\sum_{j=1}^M c_j^1 \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j^1 \Delta \varphi(\mathbf{r}_{ij}) = (\omega(\mathbf{x}_i) - \Delta \omega(\mathbf{x}_i)) + \eta_0 \mathbf{f}_0, \quad i = M_l + 1, \dots, M, \tag{52}$$

where  $\mathbf{f}_0 = [\Delta \omega(\mathbf{x}_i) - \nabla \omega(\mathbf{x}_i) + \nabla F(\omega(\mathbf{x}_i)) + f(\mathbf{x}_i, 0)]_{i=M_l+1}^M$  and  $\eta_0$  is calculated from Eq. (21). For  $1 \leq n \leq N - 1$  we have

$$\begin{aligned} & \sum_{j=1}^M c_j^{n+1} \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j^{n+1} \Delta \varphi(\mathbf{r}_{ij}) \\ &= \left( \sum_{j=1}^M c_j^n \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j^n \Delta \varphi(\mathbf{r}_{ij}) \right) + \eta_n \mathbf{f}_n, \quad i = M_l + 1, \dots, M, \end{aligned} \tag{53}$$

where

$$\mathbf{f}_n = \left[ \sum_{j=1}^M c_j^n \Delta \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j^n \nabla \varphi(\mathbf{r}_{ij}) + \nabla \left( F \left( \sum_{j=1}^M c_j^n \varphi(\mathbf{r}_{ij}) \right) \right) + f(\mathbf{x}_i, t_n) \right]_{i=M_l+1}^M,$$

and  $\eta_n$  is calculated from Eq. (21). Using Eqs. (52), (53) and condition boundary (3) we have the following matrix form:

$$\mathbf{A} \mathbf{c}^{n+1} = \mathbf{B}^{n+1}, \tag{54}$$

in which

$$\mathbf{A} = \begin{bmatrix} \varphi(\mathbf{r}_{1,1}) & \varphi(\mathbf{r}_{1,2}) & \cdots & \varphi(\mathbf{r}_{1,M_l}) & \varphi(\mathbf{r}_{1,M_l+1}) & \cdots & \varphi(\mathbf{r}_{1,M}) \\ \varphi(\mathbf{r}_{2,1}) & \varphi(\mathbf{r}_{2,2}) & \cdots & \varphi(\mathbf{r}_{2,M_l}) & \varphi(\mathbf{r}_{2,M_l+1}) & \cdots & \varphi(\mathbf{r}_{2,M}) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ \varphi(\mathbf{r}_{M_l,1}) & \varphi(\mathbf{r}_{M_l,2}) & \cdots & \varphi(\mathbf{r}_{M_l,M_l}) & \varphi(\mathbf{r}_{M_l,M_l+1}) & \cdots & \varphi(\mathbf{r}_{M_l,M}) \\ \Upsilon(\varphi(\mathbf{r}_{M_l+1,1})) & \Upsilon(\varphi(\mathbf{r}_{M_l+1,2})) & \cdots & \Upsilon(\varphi(\mathbf{r}_{M_l+1,M_l})) & \Upsilon(\varphi(\mathbf{r}_{M_l+1,M_l+1})) & \cdots & \Upsilon(\varphi(\mathbf{r}_{M_l+1,M})) \\ \Upsilon(\varphi(\mathbf{r}_{M_l+2,1})) & \Upsilon(\varphi(\mathbf{r}_{M_l+2,2})) & \cdots & \Upsilon(\varphi(\mathbf{r}_{M_l+2,M_l})) & \Upsilon(\varphi(\mathbf{r}_{M_l+2,M_l+1})) & \ddots & \Upsilon(\varphi(\mathbf{r}_{M_l+2,M})) \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \Upsilon(\varphi(\mathbf{r}_{M,1})) & \Upsilon(\varphi(\mathbf{r}_{M,2})) & \cdots & \Upsilon(\varphi(\mathbf{r}_{M,M_l})) & \Upsilon(\varphi(\mathbf{r}_{M,M_l+1})) & \cdots & \Upsilon(\varphi(\mathbf{r}_{M,M})) \end{bmatrix}, \quad (55)$$

is  $M \times M$  matrix, the operator  $\Upsilon$  is

$$\Upsilon(\varphi(\mathbf{r}_{i,j})) = \{\varphi(\mathbf{r}_j) - \Delta\varphi(\mathbf{r}_j)\} |_{\mathbf{x}=\mathbf{x}_i}, \quad M_l + 1 \leq i \leq M, \quad 1 \leq j \leq M,$$

also we have

$$\mathbf{c}^{n+1} = [c_1^{n+1} \quad c_2^{n+1} \quad \cdots \quad c_M^{n+1}]^T, \quad \mathbf{B}^{n+1} = [b_1^{n+1} \quad b_2^{n+1} \quad \cdots \quad b_M^{n+1}]^T, \quad (56)$$

where

$$b_i^1 = \{\omega(\mathbf{x}) - \Delta\omega(\mathbf{x}) + \eta_0 \mathbf{f}_0\} |_{\mathbf{x}=\mathbf{x}_i}, \quad M_l + 1 \leq i \leq M, \quad (57)$$

and also we obtain

$$b_i^{n+1} = \sum_{j=1}^M c_j^n \varphi(\mathbf{r}_{ij}) - \sum_{j=1}^M c_j^n \Delta\varphi(\mathbf{r}_{ij}) + \eta_n \mathbf{f}_n, \quad M_l + 1 \leq i \leq M, \quad 1 \leq n \leq N-1, \quad (58)$$

and finally we can write

$$b_i^{n+1} = h(\mathbf{x}_i, t_{n+1}), \quad 1 \leq i \leq M_l, \quad 0 \leq n \leq N-1. \quad (59)$$

After solving the algebraic system of equations  $\mathbf{A}\mathbf{c}^{n+1} = \mathbf{B}^{n+1}$  at each time step with LU decomposition method, we can construct the solution using Eq. (46).

#### 4. Numerical results

In this section we present the numerical results obtained with the proposed method on four test problems. We test the accuracy and stability of the method described in this paper by applying it with different values of  $h$  (distance points in space) and  $\tau$  (length of time step). We performed our computations using Mathematica 10 software on a Pentium V, 2800 MHz CPU machine with 2 GB of memory. In particular we compute the following error norms:

$$L_\infty = \max_{1 \leq i \leq M} |u(\mathbf{x}_i, T) - U(\mathbf{x}_i, T)|, \quad L_2 = \sqrt{\sum_{i=1}^M |u(\mathbf{x}_i, T) - U(\mathbf{x}_i, T)|^2},$$

$$\text{RMS} = \sqrt{\frac{1}{M} \sum_{i=1}^M |u(\mathbf{x}_i, T) - U(\mathbf{x}_i, T)|^2}, \quad \text{absolute error} = \sum_{i=1}^M \sum_{j=1}^N |u(\mathbf{x}_i, t_j) - U(\mathbf{x}_i, t_j)|,$$

where RMS means root-mean-square,  $T$  is the final time,  $M$  is the total number of points and  $N$  is the total number of time discretization.

##### 4.1. Test problem 1

We consider the 1-dimensional nonlinear GBBMB equation

$$\frac{\partial u(x, t)}{\partial t} - \frac{\partial}{\partial t} \left( \frac{\partial^2 u(x, t)}{\partial x^2} \right) - \frac{\partial^2 u(x, t)}{\partial x^2} + \frac{\partial u(x, t)}{\partial x} = u(x, t) \frac{\partial u(x, t)}{\partial x} + f(x, t), \quad (60)$$

with initial condition

$$u(x, 0) = \text{sech}(x), \quad a < x < b, \quad (61)$$

and boundary conditions

$$u(a, t) = \text{sech}(a - t), \quad u(b, t) = \text{sech}(b - t), \quad 0 < t \leq T. \quad (62)$$



**Table 1**

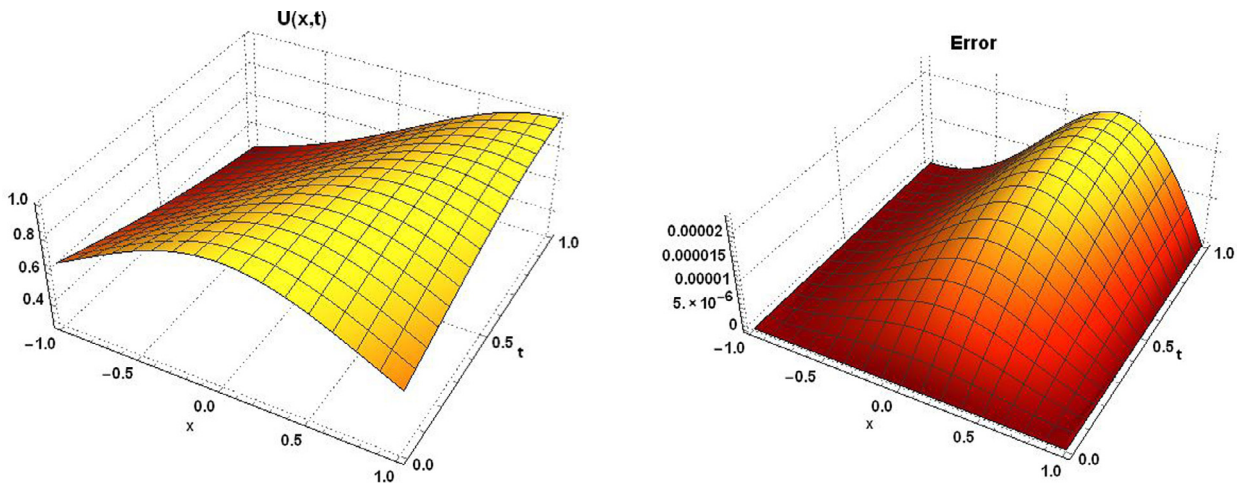
Errors obtained for the present method on interval [0, 1] with  $h = 1/10$  and  $T = 1$  for test problem 1.

$\tau$	RMS	$L_\infty$	$L_2$	$c$
1/10	$7.6989 \times 10^{-3}$	$1.1899 \times 10^{-2}$	$2.5534 \times 10^{-2}$	0.1
1/20	$4.2509 \times 10^{-3}$	$6.4885 \times 10^{-3}$	$1.4099 \times 10^{-2}$	0.2
1/40	$2.1131 \times 10^{-3}$	$3.2224 \times 10^{-3}$	$7.0083 \times 10^{-3}$	0.3
1/80	$1.0548 \times 10^{-3}$	$1.6079 \times 10^{-3}$	$3.4985 \times 10^{-3}$	0.4
1/160	$5.2964 \times 10^{-4}$	$8.0634 \times 10^{-4}$	$1.7566 \times 10^{-3}$	0.5
1/320	$2.6710 \times 10^{-4}$	$4.0580 \times 10^{-4}$	$8.8586 \times 10^{-4}$	0.6
1/640	$1.3503 \times 10^{-4}$	$2.0463 \times 10^{-4}$	$4.4784 \times 10^{-4}$	0.7
1/1280	$6.8336 \times 10^{-5}$	$1.0328 \times 10^{-4}$	$2.2665 \times 10^{-4}$	0.8
1/2560	$3.4594 \times 10^{-5}$	$5.2139 \times 10^{-5}$	$1.1474 \times 10^{-4}$	0.9

**Table 2**

Comparison between errors of present method and method of [6] for example 1 with  $h = 1/10$ .

$\tau$	$L_\infty, T = 1$		$L_\infty, T = 5$	
	Method of [6]	Present method	Method of [6]	Present method
1/10	$1.1306 \times 10^{-2}$	$1.1899 \times 10^{-2}$	$1.0335 \times 10^{-3}$	$5.4835 \times 10^{-4}$
1/20	$5.6005 \times 10^{-3}$	$6.4885 \times 10^{-3}$	$5.8509 \times 10^{-4}$	$2.8818 \times 10^{-4}$
1/40	$2.7770 \times 10^{-3}$	$3.2224 \times 10^{-3}$	$2.9635 \times 10^{-4}$	$1.5032 \times 10^{-4}$
1/80	$1.3862 \times 10^{-3}$	$1.6079 \times 10^{-3}$	$1.4801 \times 10^{-4}$	$7.7386 \times 10^{-5}$
1/160	$6.9513 \times 10^{-4}$	$8.0634 \times 10^{-4}$	$7.3858 \times 10^{-5}$	$3.9435 \times 10^{-5}$
1/320	$3.4956 \times 10^{-4}$	$4.0580 \times 10^{-4}$	$3.6874 \times 10^{-5}$	$1.9972 \times 10^{-5}$



**Fig. 1.** Graphs of approximate solution and absolute error with  $h = 1/100$ ,  $\tau = 1/10,000$  and  $c = 0.1$  on interval  $[-1, 1]$  with  $t \in [0, 1]$  for test problem 1.

with

$$f(x, t) = \operatorname{sech}(t-x) (\operatorname{sech}^2(t-x) (1 - 5 \tanh(t-x)) - \operatorname{sech}(t-x) \tanh(t-x) + \tanh^2(t-x) (\tanh(t-x) - 1)). \tag{63}$$

The exact solution of this problem is  $u(x, t) = \operatorname{sech}(x - t)$ . We solve this problem with the new method presented in this article with several values of  $h, \tau, a$  and  $b$  at different final time  $T$ . We fix  $h = 1/10$  and Table 1 shows errors obtained on interval  $[0, 1]$  for the present method with  $T = 1$  for different values of  $\tau$ . We can see from this Table when  $\tau$  is smaller the errors obtained are less. To demonstrate the accuracy of the present method we compare the results obtained of our method with the method applied in [6] for example 1 in Table 2. We can see the results for larger final time  $T = 5$  is better than method of [6]. Also, Figs. 1 and 2 show graphs of approximate solution and absolute error on interval  $[-1, 1]$  with  $t \in [0, 1]$  and on interval  $[-10, 10]$  with  $t \in [0, 10]$ , respectively, for this example.

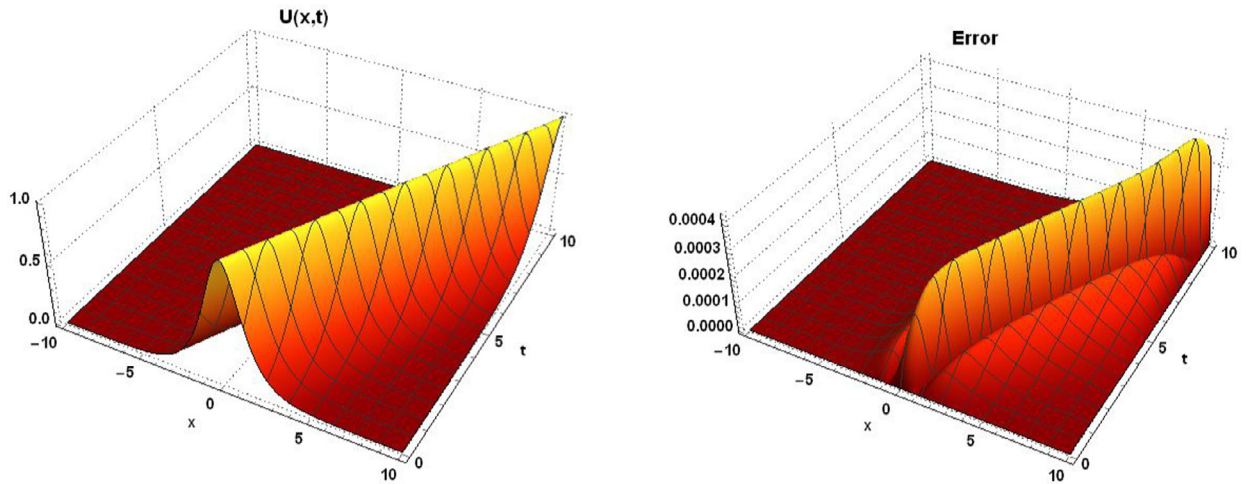


Fig. 2. Graphs of approximate solution and absolute error with  $h = 1/10$ ,  $\tau = 1/1000$  and  $c = 0.9$  on interval  $[-10, 10]$  with  $t \in [0, 10]$  for test problem 1.

Table 3

Errors obtained for the present method with  $h = 1/10$  and  $T = 5$  for test problem 2 on rectangular domain  $[0, 1] \times [0, 1]$ .

$\tau$	RMS	$L_\infty$	$L_2$	$c$
1/10	$7.7736 \times 10^{-4}$	$1.8468 \times 10^{-3}$	$8.5510 \times 10^{-3}$	0.5
1/20	$3.03854 \times 10^{-4}$	$8.5218 \times 10^{-4}$	$3.3423 \times 10^{-3}$	0.5
1/40	$1.5630 \times 10^{-4}$	$3.5909 \times 10^{-4}$	$1.7194 \times 10^{-3}$	0.6
1/80	$7.9652 \times 10^{-5}$	$1.8606 \times 10^{-4}$	$8.7617 \times 10^{-4}$	0.7
1/160	$3.9893 \times 10^{-5}$	$9.4986 \times 10^{-5}$	$4.3882 \times 10^{-4}$	0.8
1/320	$1.9555 \times 10^{-5}$	$4.7289 \times 10^{-5}$	$2.1511 \times 10^{-4}$	0.9
1/640	$8.7013 \times 10^{-6}$	$2.4801 \times 10^{-5}$	$9.5714 \times 10^{-5}$	0.95
1/1280	$5.7507 \times 10^{-6}$	$8.7045 \times 10^{-6}$	$5.0258 \times 10^{-5}$	0.95
1/2560	$1.1250 \times 10^{-6}$	$5.6461 \times 10^{-6}$	$1.2375 \times 10^{-5}$	0.95

#### 4.2. Test problem 2

Next we consider the two dimensional nonlinear GBBMB equation

$$\begin{aligned} \frac{\partial u(x, y, t)}{\partial t} - \frac{\partial}{\partial t} \left( \frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} \right) - \frac{\partial^2 u(x, y, t)}{\partial x^2} - \frac{\partial^2 u(x, y, t)}{\partial y^2} \\ + \frac{\partial u(x, y, t)}{\partial x} + \frac{\partial u(x, y, t)}{\partial y} = u(x, y, t) \frac{\partial u(x, y, t)}{\partial x} + u(x, y, t) \frac{\partial u(x, y, t)}{\partial y} + f(x, y, t), \end{aligned} \tag{64}$$

with initial condition

$$u(x, y, 0) = 1, \quad x, y \in \Omega, \tag{65}$$

and boundary conditions

$$u(x, y, t) = 1 + t \sin(x + y), \quad x, y \in \partial\Omega, \quad 0 < t \leq T, \tag{66}$$

with

$$f(x, y, t) = \sin(x + y)(3 + 2t - 2t^2 \cos(x + y)). \tag{67}$$

Then the exact solution of this test problem is  $u(x, y, t) = 1 + t \sin(x + y)$ . We solve this problem with LG-RBFs with several values of  $h$ ,  $\tau$  at different final time on the rectangular  $\Omega_1 = \{(x, y) : 0 \leq x, y \leq 1\}$ , triangular  $\Omega_2 = \{(x, y) : y \geq 0, y + x \leq 1, y - x \leq 1\}$ , and circular  $\Omega_3 = \{(x, y) : x^2 + y^2 \leq 1\}$  domains that are shown in Fig. 3. Also, we solve this problem on irregular domain that its criterion is  $r = \frac{1}{n^2}[1 + 2n + n^2 - (n + 1)\cos(n\theta)]$ . We plot this irregular domains for  $n = 4$  and  $n = 8$  in Fig. 4 which are marked with  $\Omega_4$  and  $\Omega_5$ , respectively. Tables 3 and 5 show errors obtained for the present method with  $h = 1/10$ ,  $T = 5$  on rectangular domain and with  $h = 1/10$ ,  $T = 10$  on triangular domain for different value of  $\tau$ , respectively. From this tables we can see the method is convergent when  $\tau$  is smaller. In Table 4 we compare the our results with method of [6] for final times  $T = 5$  and  $T = 10$ . We can see in two cases the results of our method is better. In Fig. 5 we plot graph of approximation solution and absolute error using LG-RBFs on rectangular domain  $[0, 1] \times [0, 1]$  with  $T = 1$ . From Fig. 6 we can show graphs of approximation solution and absolute error on larger rectangular domain

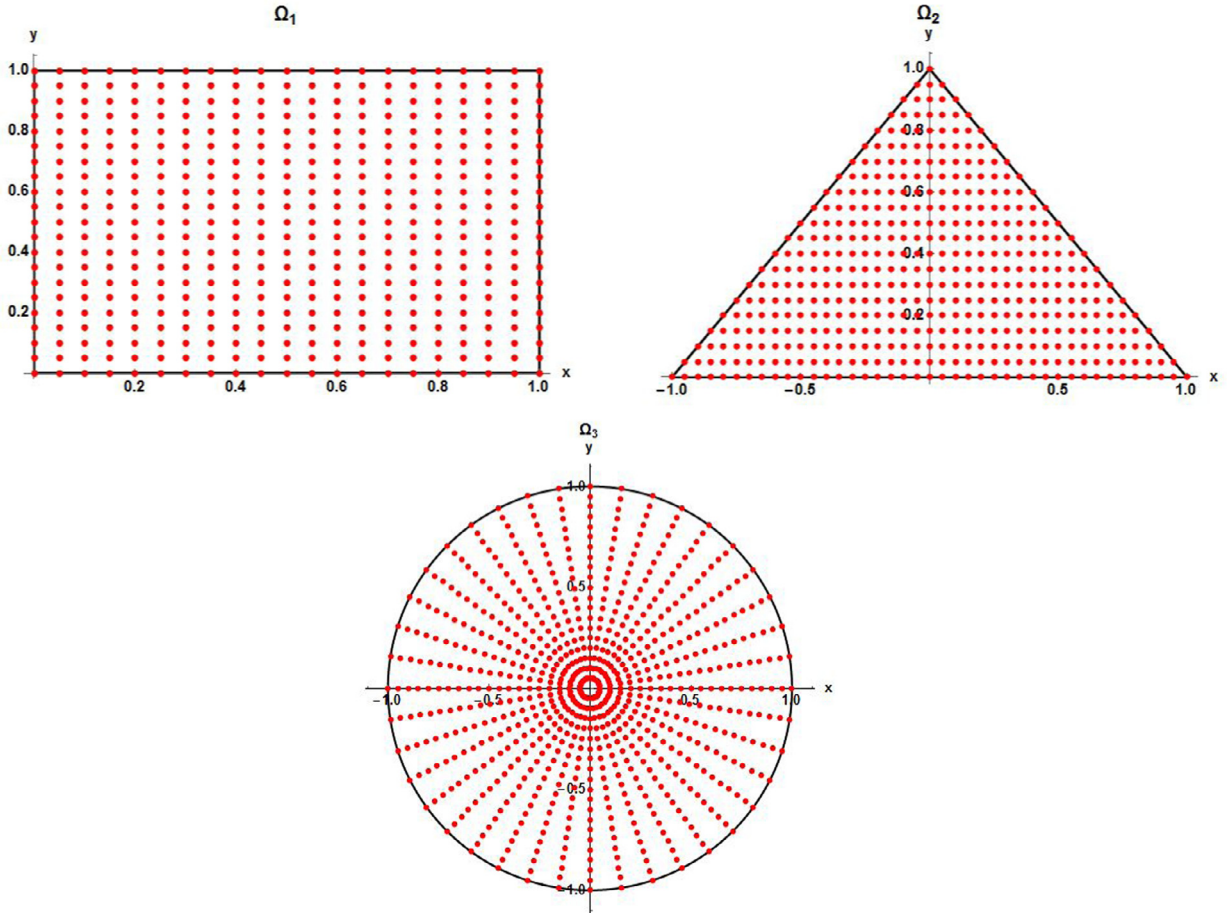


Fig. 3. Nodal points distribution on rectangular, triangular and circular domains.

Table 4

Comparison between errors of present method and method of [6] for example 2 with  $h = 1/10$  on rectangular domain  $[0, 1] \times [0, 1]$ .

$\tau$	$L_\infty, T = 5$		$L_\infty, T = 10$	
	Method of [6]	Present method	Method of [6]	Present method
1/10	$2.0927 \times 10^{-2}$	$1.8468 \times 10^{-3}$	$4.4655 \times 10^{-2}$	$4.9445 \times 10^{-4}$
1/20	$8.7888 \times 10^{-3}$	$8.5218 \times 10^{-4}$	$1.8978 \times 10^{-2}$	$2.5911 \times 10^{-4}$
1/40	$3.7341 \times 10^{-3}$	$3.5909 \times 10^{-4}$	$8.1482 \times 10^{-3}$	$1.3078 \times 10^{-4}$
1/80	$1.6029 \times 10^{-3}$	$1.8606 \times 10^{-4}$	$3.5189 \times 10^{-3}$	$6.3677 \times 10^{-5}$
1/160	$7.0137 \times 10^{-4}$	$9.4986 \times 10^{-5}$	$1.5485 \times 10^{-3}$	$5.1413 \times 10^{-5}$
1/320	$3.1392 \times 10^{-4}$	$4.7289 \times 10^{-5}$	$7.0350 \times 10^{-4}$	$3.9876 \times 10^{-5}$
1/640	$1.4378 \times 10^{-4}$	$2.4801 \times 10^{-5}$	$3.3268 \times 10^{-4}$	$2.8596 \times 10^{-5}$
1/1280	$6.8364 \times 10^{-5}$	$8.7045 \times 10^{-6}$	$1.6439 \times 10^{-4}$	$1.5596 \times 10^{-5}$

Table 5

Errors obtained for the present method with  $h = 1/10$  and  $T = 10$  for test problem 2 on triangular domain.

$\tau$	RMS	$L_\infty$	$L_2$	$c$
1	$1.4798 \times 10^{-3}$	$3.8469 \times 10^{-3}$	$1.6278 \times 10^{-2}$	0.4
1/2	$7.4514 \times 10^{-4}$	$1.7500 \times 10^{-3}$	$8.1966 \times 10^{-3}$	0.5
1/4	$3.7388 \times 10^{-4}$	$8.8739 \times 10^{-4}$	$4.1127 \times 10^{-3}$	0.6
1/8	$1.8684 \times 10^{-4}$	$4.6471 \times 10^{-4}$	$2.0552 \times 10^{-3}$	0.7
1/16	$9.3123 \times 10^{-5}$	$2.4038 \times 10^{-4}$	$1.0243 \times 10^{-3}$	0.8
1/32	$4.6516 \times 10^{-5}$	$1.2329 \times 10^{-4}$	$5.1168 \times 10^{-4}$	0.9
1/64	$2.3639 \times 10^{-5}$	$6.3175 \times 10^{-5}$	$2.6003 \times 10^{-4}$	1
1/128	$1.2168 \times 10^{-5}$	$3.1685 \times 10^{-5}$	$1.3385 \times 10^{-4}$	1.1
1/256	$6.6453 \times 10^{-6}$	$1.6033 \times 10^{-5}$	$7.3098 \times 10^{-5}$	1.2

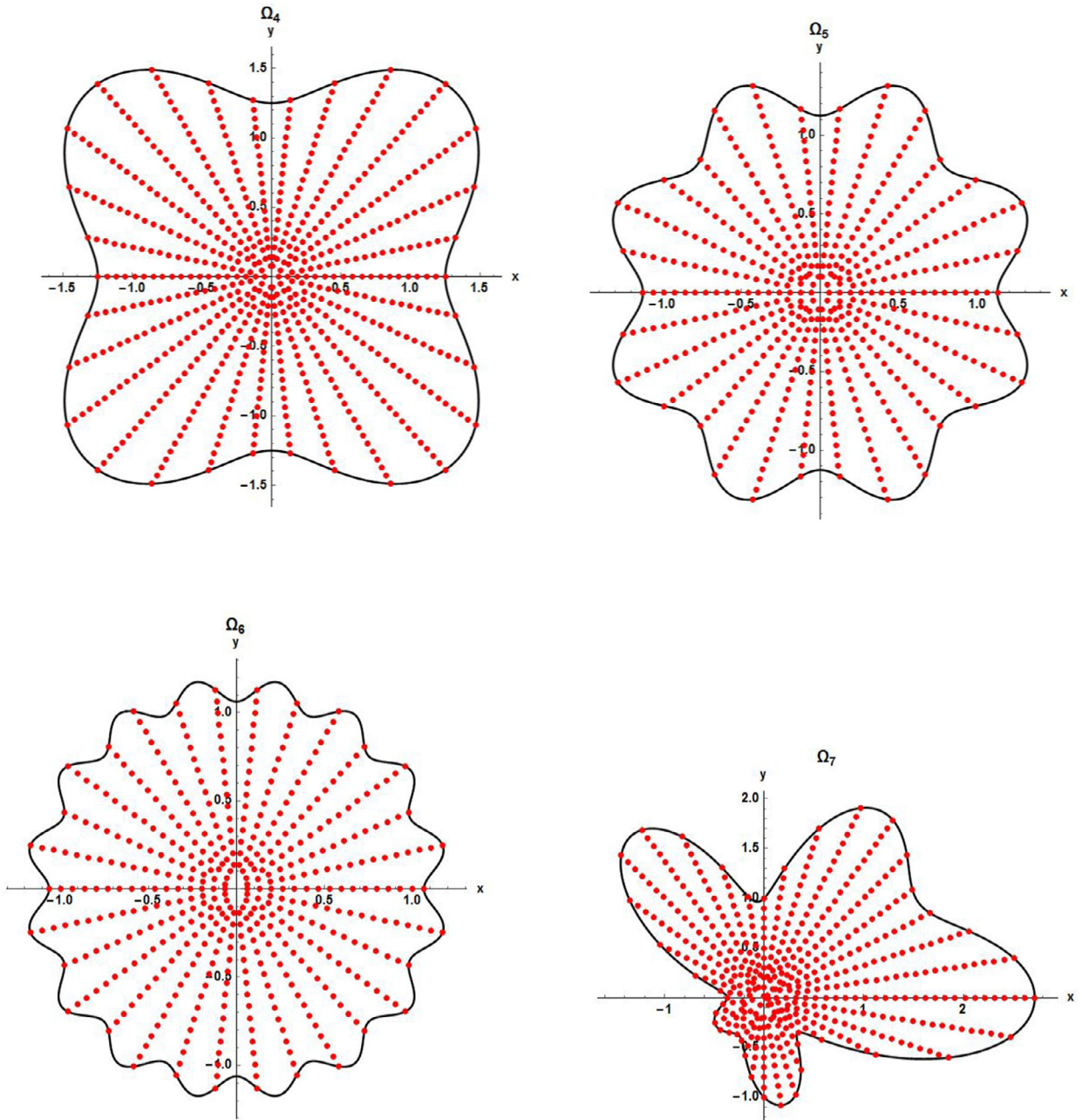


Fig. 4. Nodal points distribution on irregular domains.

$[-5, 5] \times [-5, 5]$  with larger final time  $T = 5$ . Finally, from Figs. 7–10 we can see graphs of approximation solution and absolute error using LG-RBFs on triangular, circular and irregular domains, respectively, for different final time for test problem 2.

4.3. Test problem 3

For our next problem we take again a 2-dimensional nonlinear GBBMB equation,

$$\begin{aligned} \frac{\partial u(x, y, t)}{\partial t} - \frac{\partial}{\partial t} \left( \frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} \right) - \frac{\partial^2 u(x, y, t)}{\partial x^2} - \frac{\partial^2 u(x, y, t)}{\partial y^2} \\ + \frac{\partial u(x, y, t)}{\partial x} + \frac{\partial u(x, y, t)}{\partial y} = \cos(u(x, y, t)) \frac{\partial u(x, y, t)}{\partial x} + \cos(u(x, y, t)) \frac{\partial u(x, y, t)}{\partial y} + f(x, y, t), \end{aligned} \tag{68}$$



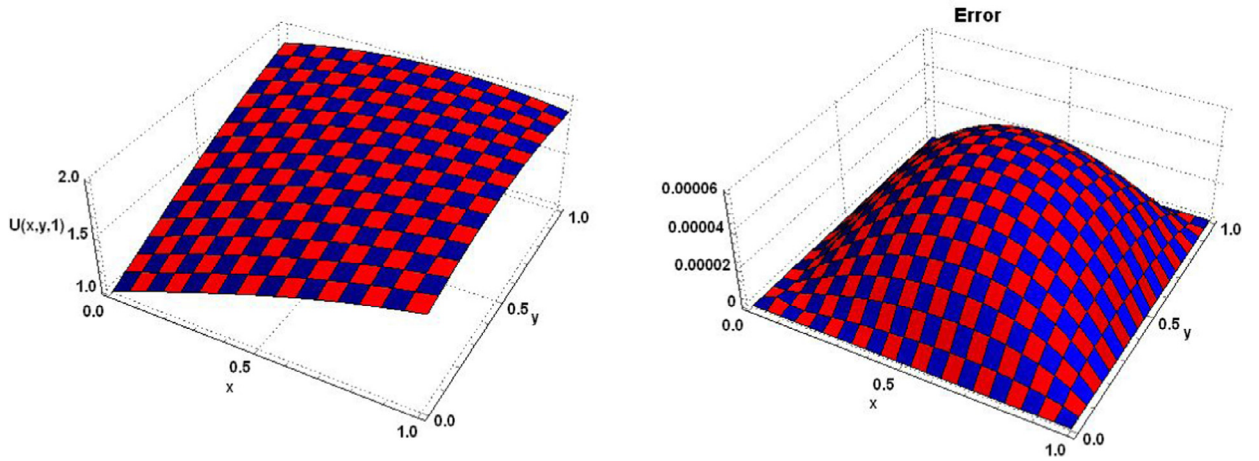


Fig. 5. Graphs of approximate solution and absolute error with  $h = 1/20$ ,  $\tau = 1/1000$  and  $c = 0.45$  on rectangular domain  $[0, 1] \times [0, 1]$  with  $T = 1$  for test problem 2.

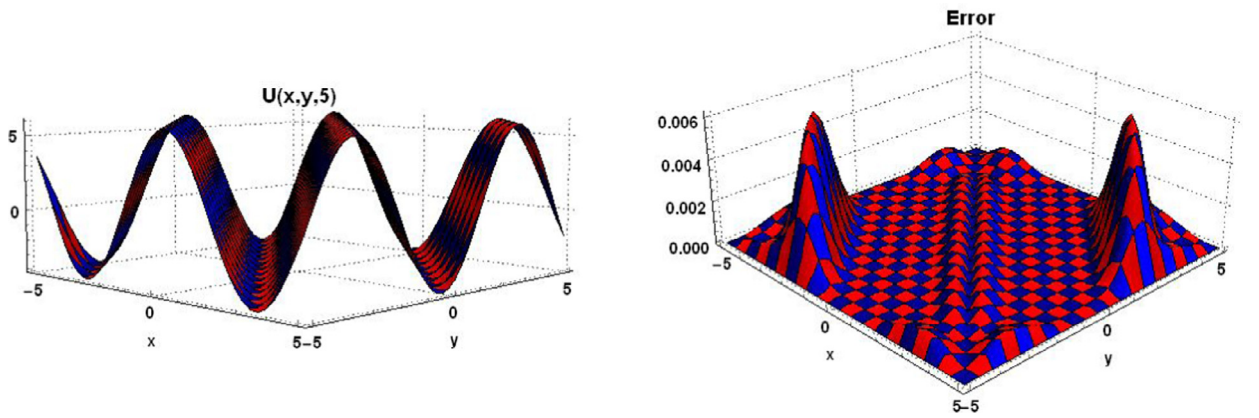


Fig. 6. Graphs of approximate solution and absolute error with  $h = 1/4$ ,  $\tau = 1/200$  and  $c = 1.9$  on rectangular domain  $[-5, 5] \times [-5, 5]$  with  $T = 5$  for test problem 2.

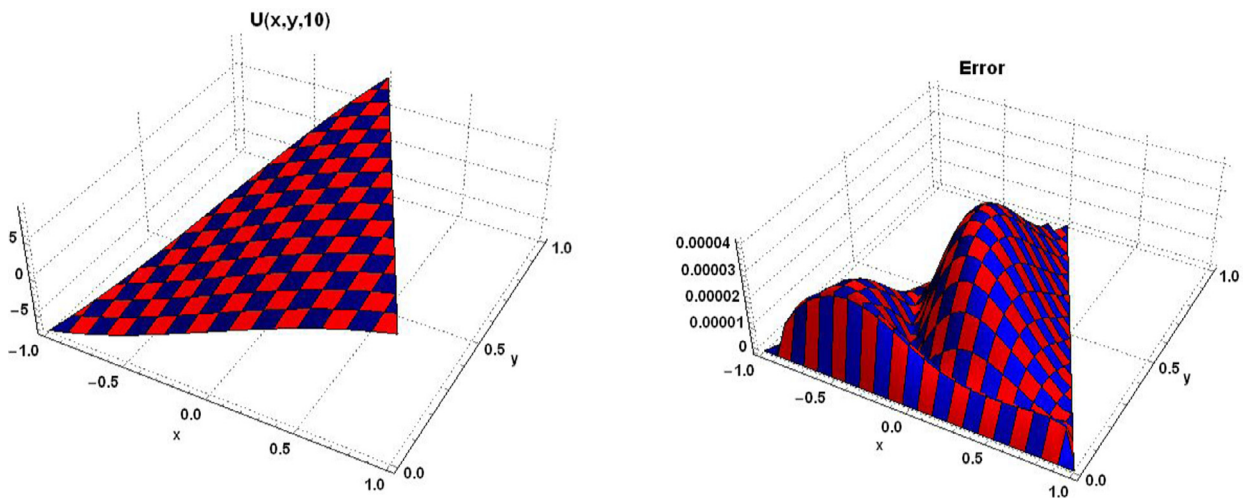


Fig. 7. Graphs of approximate solution and absolute error with  $h = 1/20$ ,  $\tau = 1/100$  and  $c = 0.45$  on triangular domain with  $T = 10$  for test problem 2.

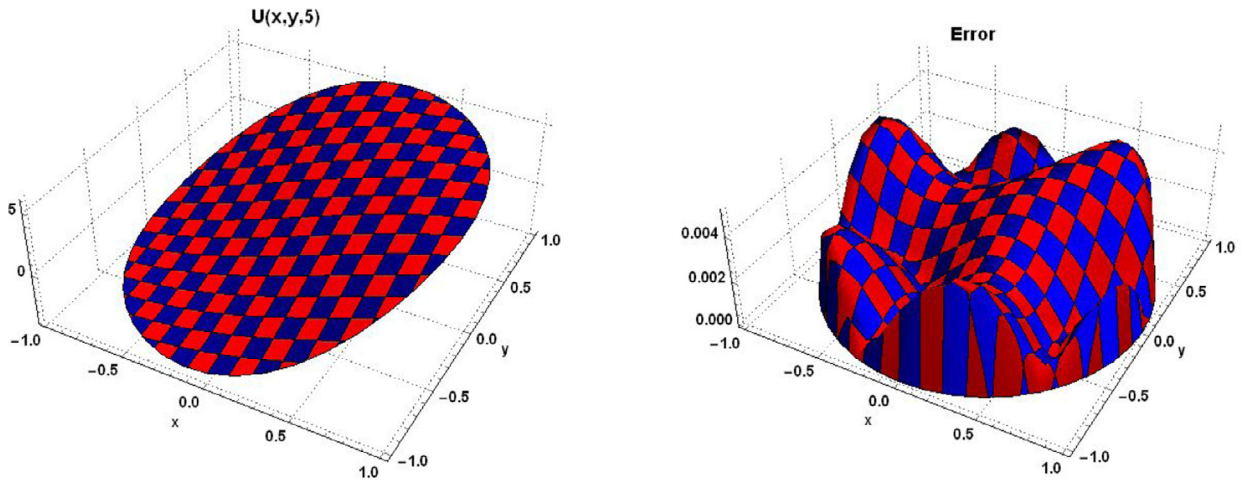


Fig. 8. Graphs of approximate solution and absolute error with  $h = 1/20$ ,  $\tau = 1/200$  and  $c = 0.1$  on circular domain with  $T = 5$  for test problem 2.

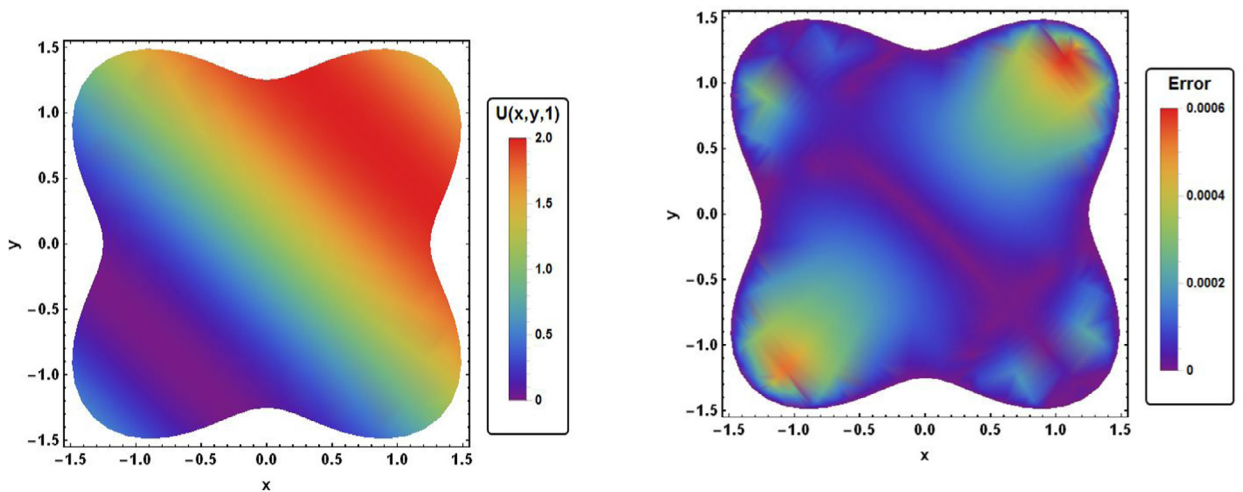


Fig. 9. Graphs of approximate solution and absolute error with  $h = 1/15$ ,  $\tau = 1/1000$  and  $c = 0.5$  on irregular domain ( $n = 4$ ) with  $T = 1$  for test problem 2.

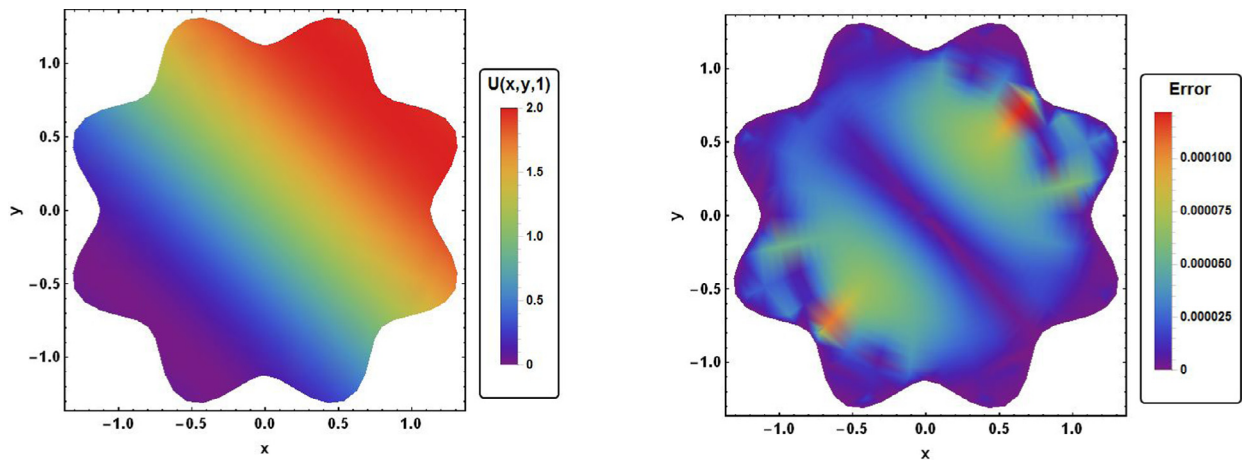
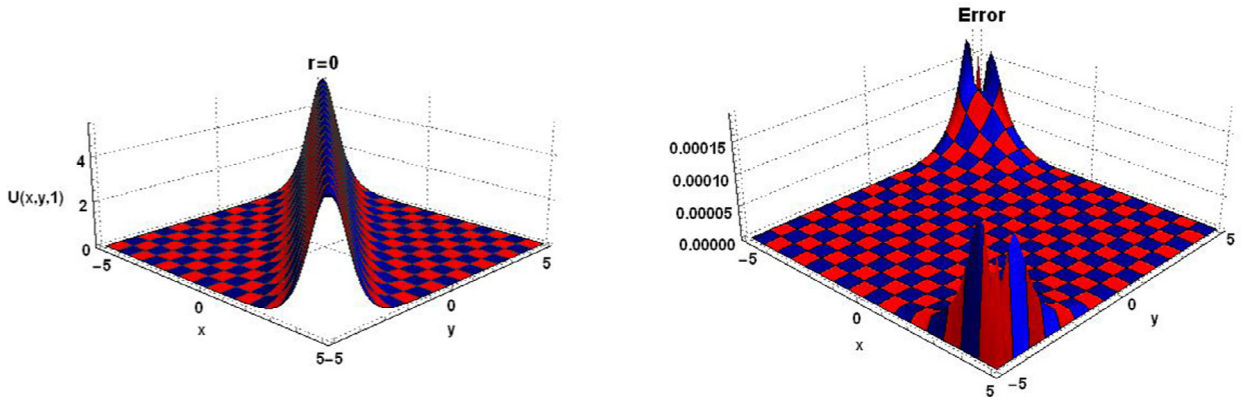


Fig. 10. Graphs of approximate solution and absolute error with  $h = 1/15$ ,  $\tau = 1/1000$  and  $c = 0.5$  on irregular domain ( $n = 8$ ) with  $T = 1$  for test problem 2.

**Table 6**

Comparison between errors of present method and method of [6] for example 3 with  $r = 0$ ,  $h = 1/5$  on rectangular domain  $[-2, 2] \times [-2, 2]$ .

$\tau$	$L_\infty, T = 1$		$L_\infty, T = 2$	
	Method of [6]	Present method	Method of [6]	Present method
1/10	$1.2187 \times 10^{-1}$	$1.8855 \times 10^{-4}$	$7.7952 \times 10^{-1}$	$5.1091 \times 10^{-4}$
1/20	$6.0296 \times 10^{-2}$	$1.6804 \times 10^{-4}$	$3.8698 \times 10^{-1}$	$4.1586 \times 10^{-4}$
1/40	$2.9981 \times 10^{-2}$	$1.3225 \times 10^{-4}$	$1.9274 \times 10^{-1}$	$2.1213 \times 10^{-4}$
1/80	$1.4939 \times 10^{-2}$	$1.1248 \times 10^{-4}$	$9.6149 \times 10^{-2}$	$1.1227 \times 10^{-4}$
1/160	$7.4461 \times 10^{-3}$	$9.9553 \times 10^{-5}$	$4.7985 \times 10^{-2}$	$9.1236 \times 10^{-5}$
1/320	$3.7069 \times 10^{-3}$	$7.2345 \times 10^{-5}$	$2.3937 \times 10^{-2}$	$8.1253 \times 10^{-5}$
1/640	$1.8389 \times 10^{-3}$	$5.8967 \times 10^{-5}$	$1.1920 \times 10^{-2}$	$5.1270 \times 10^{-5}$



**Fig. 11.** Graphs of approximate solution and absolute error with  $h = 1/5$ ,  $\tau = 1/10$  and  $c = 1.5$  on rectangular domain  $[-5, 5] \times [-5, 5]$  with  $r = 0$  and  $T = 1$  for test problem 3.

but now with initial condition

$$u(x, y, 0) = \operatorname{sech}^2(x + y - r) + \operatorname{sech}^2(x + y + r), \quad x, y \in \Omega, \tag{69}$$

and boundary conditions

$$u(x, y, t) = \exp(t)(\operatorname{sech}^2(x + y - r) + \operatorname{sech}^2(x + y + r)), \quad x, y \in \partial\Omega, \quad 0 < t \leq T. \tag{70}$$

We have chosen  $f(x, y, t)$  so that the exact solution of this test problem is  $u(x, y, t) = \exp(t)(\operatorname{sech}^2(x + y - r) + \operatorname{sech}^2(x + y + r))$ . We solve this problem with the new method presented in this article with several values of  $h$ ,  $\tau$  and  $r$  at different final time on the rectangular, triangular, and circular domains. Also, we solve this problem on irregular domains that their criterion is  $\Omega_6 = \{\frac{1}{n^2}(1 + 2n + n^2 - (n + 1)\cos(n\theta)), (n = 16)\}$  and  $\Omega_7 = \{\exp(\sin(\theta))\sin^2(2\theta) + \exp(\cos(\theta))\cos^2(2\theta)\}$  which we plot this irregular domains in Fig. 4. To show the accuracy of the method from Table 6 we can see the comparison between the present method and method of [6]. From Figs. 11–18 we can show the numerical results for this problem on different domains for various  $r$ .

**4.4. Test problem 4**

In the last example, we consider the three-dimensional nonlinear GBBMB equation

$$\begin{aligned} \frac{\partial u(x, y, z, t)}{\partial t} - \frac{\partial}{\partial t} \left( \frac{\partial^2 u(x, y, z, t)}{\partial x^2} + \frac{\partial^2 u(x, y, z, t)}{\partial y^2} \right) - \frac{\partial^2 u(x, y, z, t)}{\partial x^2} - \frac{\partial^2 u(x, y, z, t)}{\partial y^2} \\ + \frac{\partial u(x, y, z, t)}{\partial x} + \frac{\partial u(x, y, z, t)}{\partial y} = u(x, y, z, t) \frac{\partial u(x, y, z, t)}{\partial x} + u(x, y, z, t) \frac{\partial u(x, y, z, t)}{\partial y} + f(x, y, z, t), \end{aligned} \tag{71}$$

with initial condition

$$u(x, y, z, 0) = 1, \quad x, y, z \in \Omega, \tag{72}$$

and boundary conditions

$$u(x, y, z, t) = 1 + t\sin(x + y + z), \quad x, y, z \in \partial\Omega, \quad 0 < t \leq T, \tag{73}$$



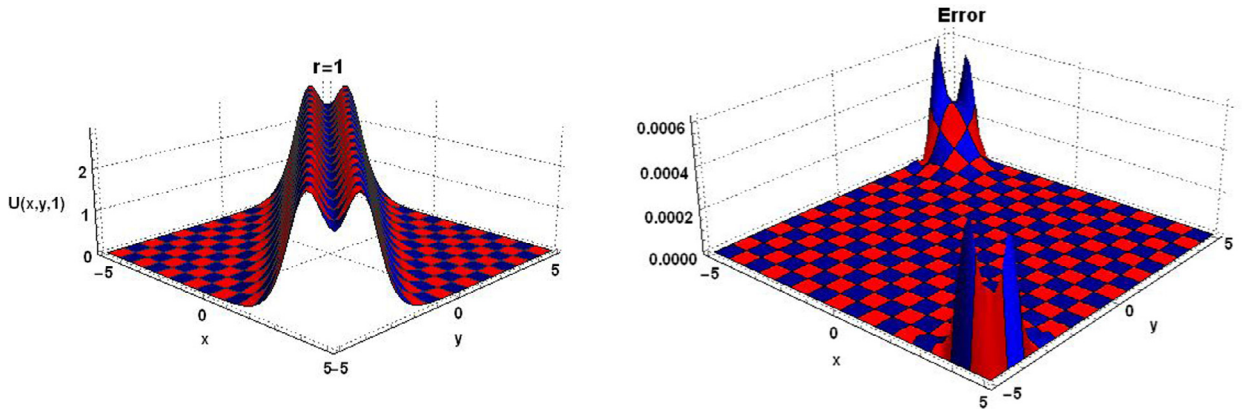


Fig. 12. Graphs of approximate solution and absolute error with  $h = 1/5$ ,  $\tau = 1/10$  and  $c = 1.5$  on rectangular domain  $[-5, 5] \times [-5, 5]$  with  $r = 1$  and  $T = 1$  for test problem 3.

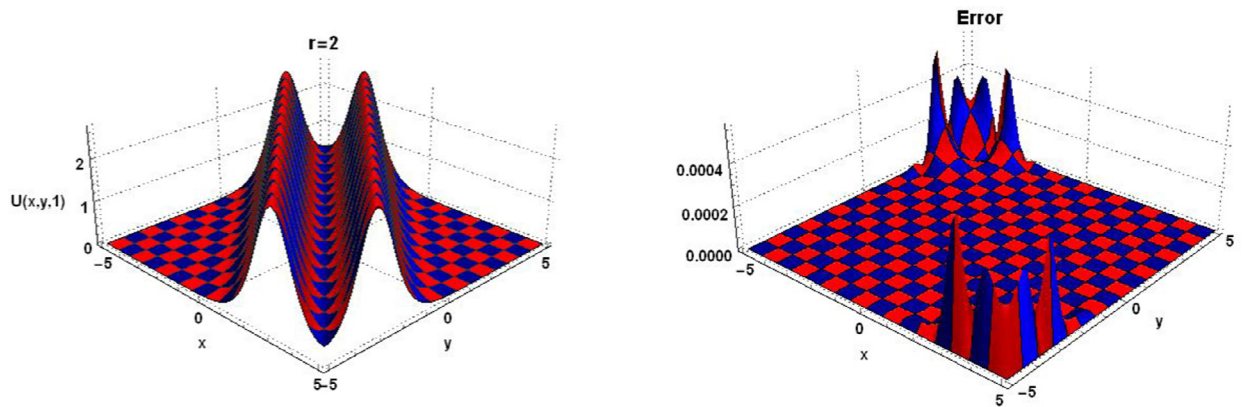


Fig. 13. Graphs of approximate solution and absolute error with  $h = 1/5$ ,  $\tau = 1/10$  and  $c = 1.5$  on rectangular domain  $[-5, 5] \times [-5, 5]$  with  $r = 2$  and  $T = 1$  for test problem 3.

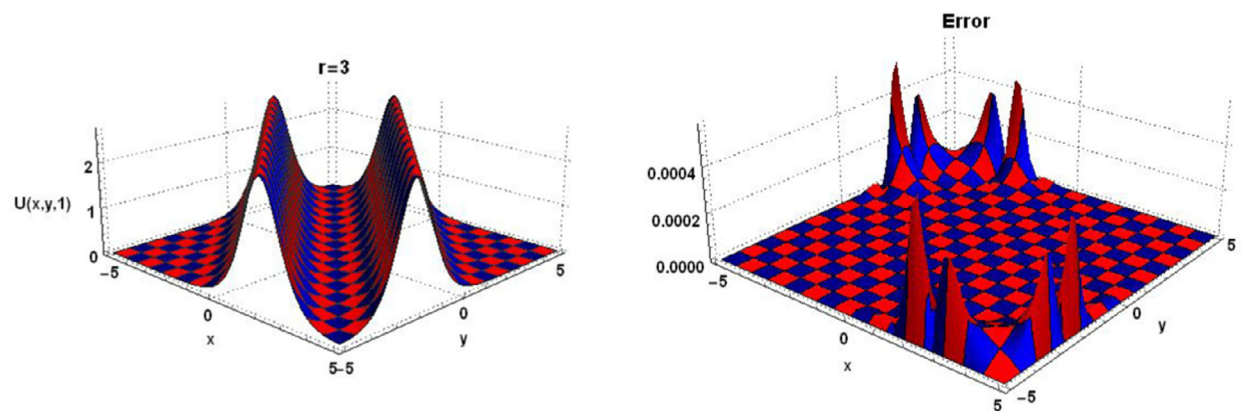


Fig. 14. Graphs of approximate solution and absolute error with  $h = 1/5$ ,  $\tau = 1/10$  and  $c = 1.5$  on rectangular domain  $[-5, 5] \times [-5, 5]$  with  $r = 3$  and  $T = 1$  for test problem 3.

with

$$f(x, y, z, t) = (4 + 3t)\sin(x + y + z) + t\cos(x + y + z)(1 - 2t\sin(x + y + z)) - t\cos(x + y + z)(1 + t\sin(x + y + z)). \tag{74}$$

Then the exact solution of this test problem is  $u(x, y, t) = 1 + t\sin(x + y + z)$ . We solve this test problem with LG-RBFs in the current article with several values of  $h$  and  $\tau$  on the cubic and spherical domains that are shown in Fig. 19. From



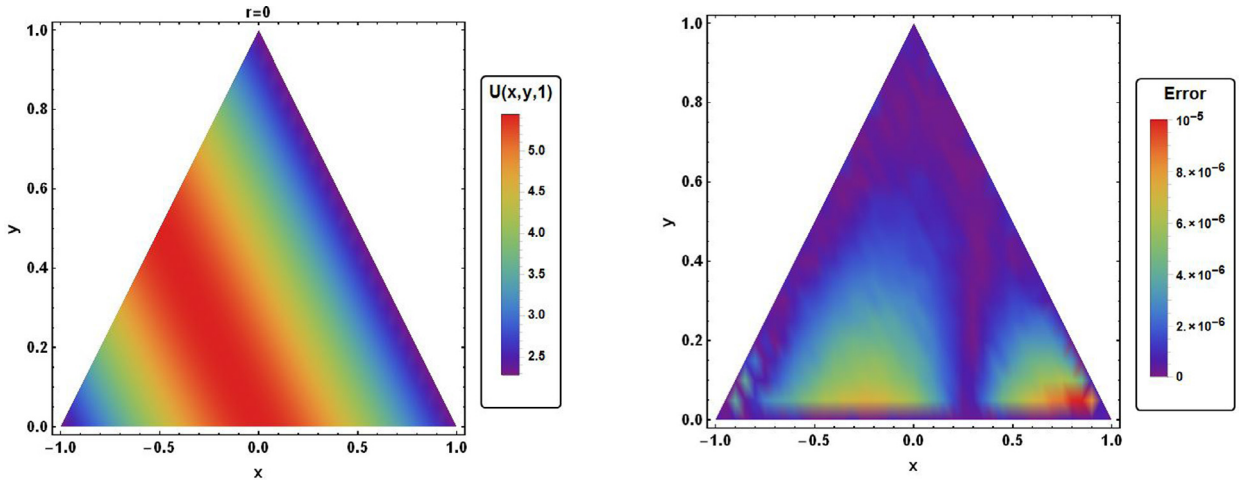


Fig. 15. Graphs of approximate solution and absolute error with  $h = 1/20$ ,  $\tau = 1/10$  and  $c = 0.52$  on triangular domain with  $r = 0$  and  $T = 1$  for test problem 3.

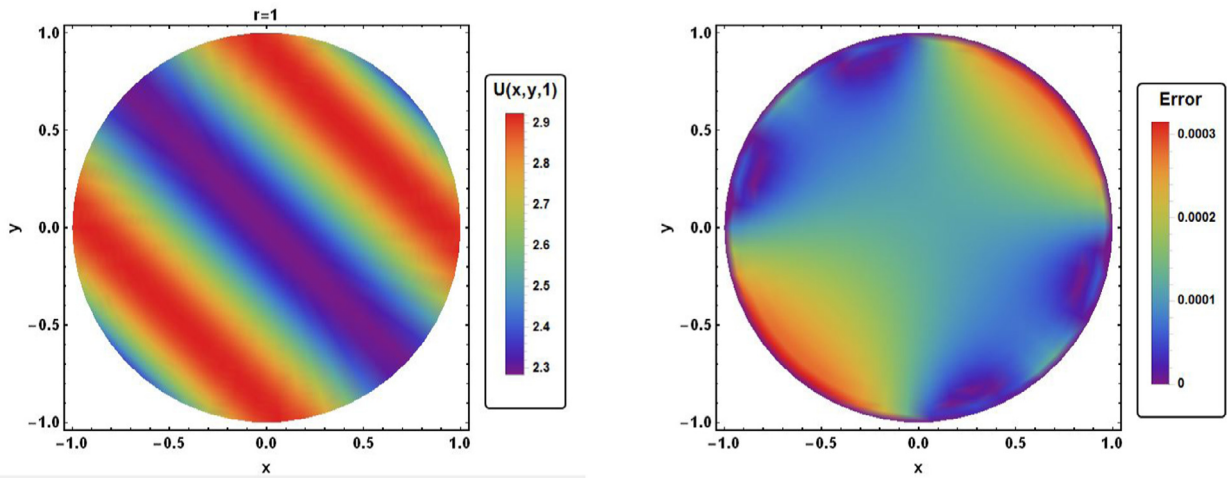


Fig. 16. Graphs of approximate solution and absolute error with  $h = 1/20$ ,  $\tau = 1/10$  and  $c = 0.2$  on circular domain with  $r = 1$  and  $T = 1$  for test problem 3.

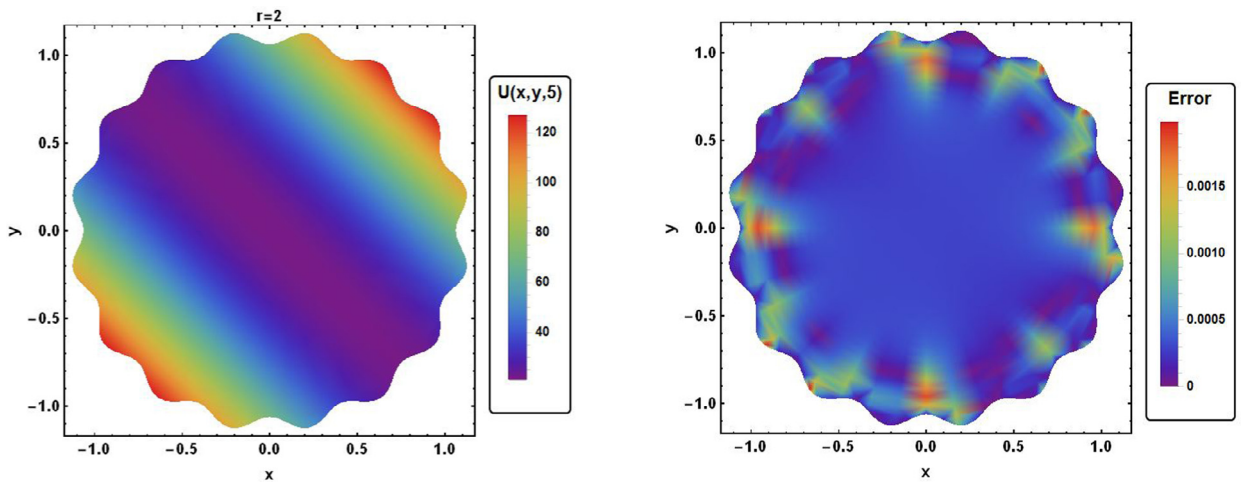


Fig. 17. Graphs of approximate solution and absolute error with  $h = 1/20$ ,  $\tau = 1/20$  and  $c = 0.4$  on irregular domain ( $n = 16$ ) with  $r = 2$  and  $T = 5$  for test problem 3.

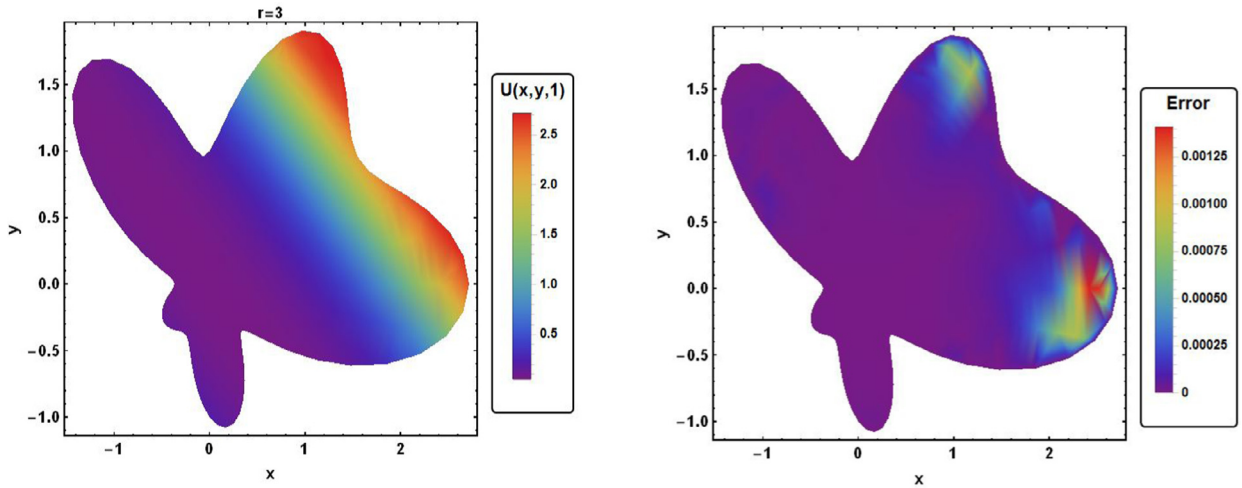


Fig. 18. Graphs of approximate solution and absolute error with  $h = 1/10$ ,  $\tau = 1/100$  and  $c = 0.8$  on irregular domain with  $r = 3$  and  $T = 1$  for test problem 3.

Table 7

Comparison between errors of present method and method of [6] for example 4 with  $h = 1/10$  and  $T = 1$  at different points of domain on cubic domain.

$(x, y, z)$	Method of [6]	Present method
(0.1, 0.1, 0.1)	$4.4851 \times 10^{-2}$	$2.91301 \times 10^{-4}$
(0.2, 0.2, 0.2)	$4.4363 \times 10^{-2}$	$1.40227 \times 10^{-3}$
(0.3, 0.3, 0.3)	$4.6234 \times 10^{-2}$	$2.93329 \times 10^{-3}$
(0.4, 0.4, 0.4)	$4.6553 \times 10^{-2}$	$4.18388 \times 10^{-3}$
(0.5, 0.5, 0.5)	$4.5498 \times 10^{-2}$	$4.61567 \times 10^{-3}$
(0.6, 0.6, 0.6)	$4.4917 \times 10^{-2}$	$4.08167 \times 10^{-3}$
(0.7, 0.7, 0.7)	$4.4182 \times 10^{-2}$	$2.83632 \times 10^{-3}$
(0.8, 0.8, 0.8)	$4.3156 \times 10^{-2}$	$1.39163 \times 10^{-3}$
(0.9, 0.9, 0.9)	$4.5028 \times 10^{-3}$	$3.23101 \times 10^{-4}$

Table 8

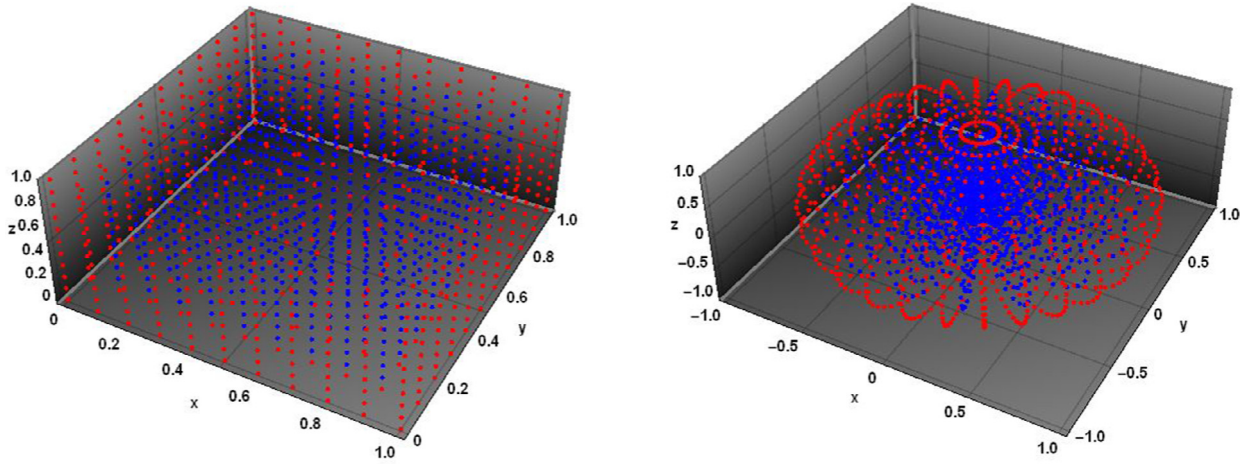
Errors obtained for the present method with  $h = 1/10$ ,  $\tau = 1/100$ ,  $c = 0.9$  and  $T = 1$  in differnt times for test problem 4 on cubic domain.

$t$	RMS	$L_\infty$	$L_2$
0.1	$1.1410 \times 10^{-4}$	$3.0817 \times 10^{-4}$	$4.2556 \times 10^{-3}$
0.2	$1.9009 \times 10^{-4}$	$5.1315 \times 10^{-4}$	$7.0896 \times 10^{-3}$
0.3	$2.4208 \times 10^{-4}$	$6.5332 \times 10^{-4}$	$9.0286 \times 10^{-3}$
0.4	$2.7788 \times 10^{-4}$	$7.4956 \times 10^{-4}$	$1.0364 \times 10^{-2}$
0.5	$3.0225 \times 10^{-4}$	$8.1453 \times 10^{-4}$	$1.1273 \times 10^{-2}$
0.6	$3.1836 \times 10^{-4}$	$8.5641 \times 10^{-4}$	$1.1874 \times 10^{-2}$
0.7	$3.2841 \times 10^{-4}$	$8.8066 \times 10^{-4}$	$1.2249 \times 10^{-2}$
0.8	$3.3413 \times 10^{-4}$	$8.9087 \times 10^{-4}$	$1.2462 \times 10^{-2}$
0.9	$3.3711 \times 10^{-4}$	$8.8931 \times 10^{-4}$	$1.2573 \times 10^{-2}$
1	$3.3946 \times 10^{-4}$	$8.7721 \times 10^{-4}$	$1.2661 \times 10^{-2}$

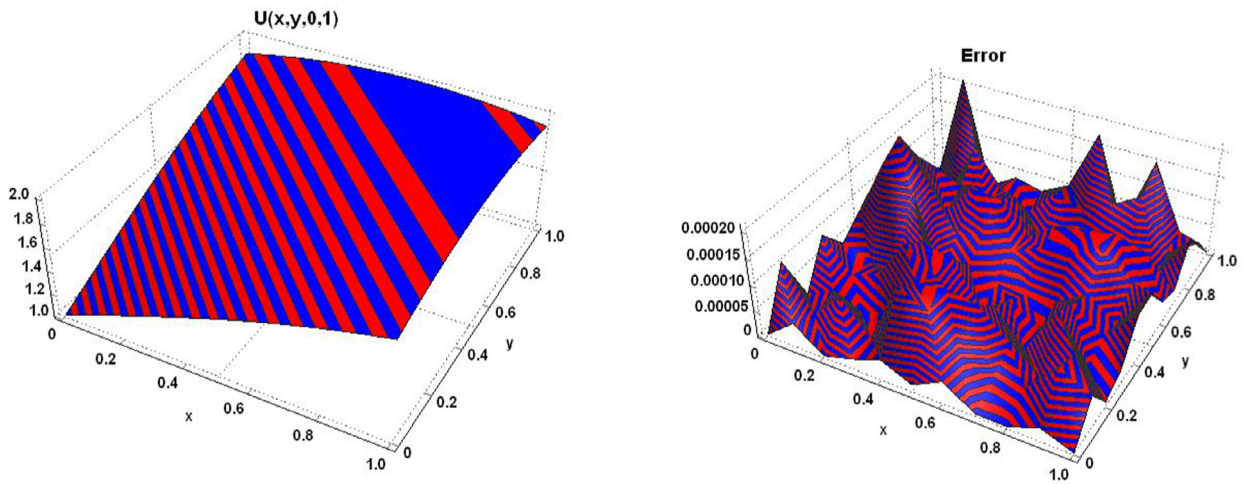
Table 9

Errors obtained for the present method with  $h = 1/10$ ,  $\tau = 1/100$ ,  $c = 0.15$  and  $T = 1$  in differnt times for test problem 4 on spherical domain.

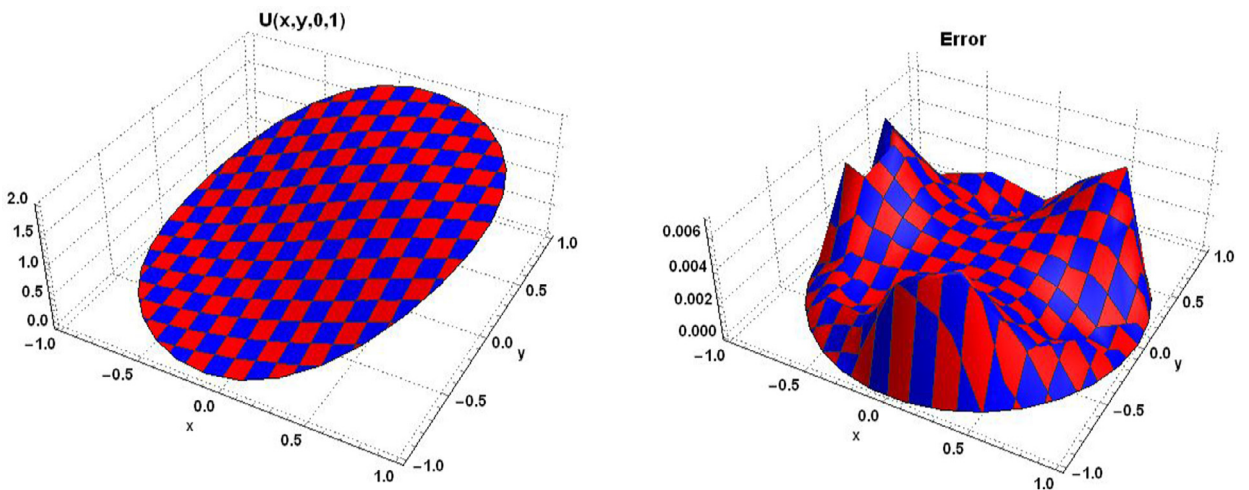
$t$	RMS	$L_\infty$	$L_2$
0.1	$1.5450 \times 10^{-3}$	$3.1080 \times 10^{-3}$	$7.8261 \times 10^{-2}$
0.2	$1.5842 \times 10^{-3}$	$4.1034 \times 10^{-3}$	$8.0249 \times 10^{-2}$
0.3	$1.6373 \times 10^{-3}$	$5.0939 \times 10^{-3}$	$8.2939 \times 10^{-2}$
0.4	$1.7032 \times 10^{-3}$	$6.0856 \times 10^{-3}$	$8.6278 \times 10^{-2}$
0.5	$1.7810 \times 10^{-3}$	$7.0936 \times 10^{-3}$	$9.0217 \times 10^{-2}$
0.6	$1.8697 \times 10^{-3}$	$8.1387 \times 10^{-3}$	$9.4712 \times 10^{-2}$
0.7	$1.9686 \times 10^{-3}$	$9.1955 \times 10^{-3}$	$9.9719 \times 10^{-2}$
0.8	$2.0768 \times 10^{-3}$	$1.0265 \times 10^{-2}$	$1.0520 \times 10^{-1}$
0.9	$2.1939 \times 10^{-3}$	$1.1348 \times 10^{-2}$	$1.1113 \times 10^{-1}$
1	$2.3192 \times 10^{-3}$	$1.2444 \times 10^{-2}$	$1.1748 \times 10^{-1}$



**Fig. 19.** Nodal points distribution on the cubic and spherical domains for test problem 4. The boundary points are red and interior points are blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).



**Fig. 20.** Graphs of approximation solution and absolute error using the present method with  $h = 1/10$ ,  $\tau = 1/100$  and  $c = 0.9$  on cubic domain in  $z = 0$  and  $T = 1$  for test problem 4.



**Fig. 21.** Graphs of approximation solution and absolute error using the present method with  $h = 1/8$ ,  $\tau = 1/100$  and  $c = 0.15$  on spherical domain in  $z = 0$  and  $T = 1$  for test problem 4.



Table 7 we can see the numerical results of our method is more accurate than method of [6] at different points. From Tables 8 and 9 we see the errors obtained for this example in different times with  $h = 1/10$  and  $\tau = 1/100$  on cubic and spherical domains, respectively. Also, we fixed  $z = 0$  and  $T = 1$  and plot the approximation solution and absolute error using the present method for this example on the cubic and spherical domains in Figs. 20 and 21, respectively.

## 5. Conclusions

The nonlinear GBBMB is an equation which is widely used in various sciences. In this paper, we present a technique for solving this equation in one, two and three dimensions and different types of domains by combining a first-order Lie-group method with radial basis functions. Numerical results show the efficiency and accuracy of the new technique and its feasibility for solving different types of partial differential equations, in particular on domains with complicated geometry and with high dimensions. Extensions to higher orders in time will be the subject of a subsequent study.

## References

- [1] Z.Z. Ganji, D.D. Ganji, H. Bararnia, Approximate general and explicit solutions of nonlinear BBMB equations by exp-function method, *Appl. Math. Model.* 33 (2009) 1836–1841.
- [2] M.A. Noor, K.I. Noor, A. Waheed, E.A. Al-Said, Some new solitary solutions of the modified Benjamin–Bona–Mahony equation, *Comput. Math. Appl.* 62 (2011) 2126–2131.
- [3] C.A. Gómez, A.H. Salas, B.A. Frias, New periodic and soliton solutions for the generalized BBM and BBM–Burgers equations, *Appl. Math. Comput.* 217 (2010) 1430–1434.
- [4] S. Abbasbandy, A. Shirzadi, The first integral method for modified Benjamin–Bona–Mahony equation, *Commun. Nonlinear Sci. Numer. Simul.* 15 (2010) 1759–1764.
- [5] M. Dehghan, M. Abbaszadeh, A. Mohebbi, The use of interpolating element-free Galerkin technique for solving 2d generalized Benjamin–Bona–Mahony–Burgers and regularized long-wave equations on non-rectangular domains with error estimate, *J. Comput. Appl. Math.* 286 (2015) 211–231.
- [6] M. Dehghan, M. Abbaszadeh, A. Mohebbi, The numerical solution of nonlinear high dimensional generalized Benjamin–Bona–Mahony–Burgers equation via the meshless method of radial basis functions, *Comput. Math. Appl.* 68 (2014) 212–237.
- [7] C.A. Gómez, A.H. Salas, B.A. Frias, New periodic and soliton solutions for the generalized BBM and BBM–Burgers equations, *Appl. Math. Comput.* 217 (2010) 1430–1434.
- [8] Q. Xiao, H. Zhao, Nonlinear stability of generalized Benjamin–Bona–Mahony–Burgers shock profiles in several dimensions, *J. Math. Anal. Appl.* 406 (2013) 165–187.
- [9] C. Guo, S. Fang, Optimal decay rates of solutions for a multi-dimensional generalized Benjamin–Bona–Mahony equation, *Nonlinear Anal.* 75 (2012) 3385–3392.
- [10] H. Yin, J. Hu, Exponential decay rate of solutions toward traveling waves for the cauchy problem of generalized Benjamin–Bona–Mahony–Burgers equations, *Nonlinear Anal.* 73 (2010) 1729–1738.
- [11] M.A. Noor, K.I. Noor, A. Waheed, E.A. Al-Said, Some new solitary solutions of the modified Benjamin–Bona–Mahony equation, *Comput. Math. Appl.* 62 (2011) 2126–2131.
- [12] X. Qinghua, C.Z. Zheng, Degenerate boundary layer solutions to the generalized Benjamin–Bona–Mahony–Burgers equation, *Acta Math. Sci.* 32 (2012) 1743–1758.
- [13] T. Achouri, N. Khiari, K. Omrani, On the convergence of difference schemes for the Benjamin–Bona–Mahony (BBM) equation, *Appl. Math. Comput.* 182 (2006) 999–1005.
- [14] K. Omrani, The convergence of fully discrete Galerkin approximations for the Benjamin–Bona–Mahony (BBM) equation, *Appl. Math. Comput.* 180 (2006) 614–621.
- [15] A. Mekki, M.M. Ali, Numerical simulation of Kadomtsev–Petviashvili–Benjamin–Bona–Mahony equations using finite difference method, *Appl. Math. Comput.* 219 (2013) 11214–11222.
- [16] M.S. Hashemi, D. Baleanu, Numerical approximation of higher-order time-fractional telegraph equation by using a combination of a geometric approach and method of line, *J. Comput. Phys.* 316 (2016) 10–20.
- [17] C.J. Budd, A. Iserles, Geometric integration: numerical solution of differential equations on manifolds, *Philos. Trans. R. Soc. Lond. A* 357 (1999) 945–956.
- [18] E. Hairer, Geometric integration of ordinary differential equations on manifolds, *BIT Numer. Math.* 41 (2001) 996–1007.
- [19] A. Iserles, H.Z. Munthe-Kaas, P.S. Norsett, A. Zanna, Lie-group methods, *Acta Numer.* 9 (2000) 215–365.
- [20] C.S. Liu, Cone of non-linear dynamical system and group preserving schemes, *Int. J. Nonlinear Mech.* 36 (2001) 1047–1068.
- [21] S. Abbasbandy, R.A.V. Gorder, M. Hajiketabi, M. Mesrizadeh, Existence and numerical simulation of periodic traveling wave solutions to the Casimir equation for the ito system, *Commun. Nonlinear Sci. Numer. Simul.* 27 (2015) 254–262.
- [22] H.C. Lee, C.K. Chen, C.I. Hung, A modified group-preserving scheme for solving the initial value problems of stiff ordinary differential equations, *Appl. Math. Comput.* 133 (2002) 445–459.
- [23] H.C. Lee, C.S. Liu, The fourth-order group preserving methods for the integrations of ordinary differential equations, *Comput. Model. Eng. Sci.* 41 (2009) 1–26.
- [24] S.Y. Zhang, Z.C. Deng, Group preserving schemes for nonlinear dynamic system based on RKMK methods, *Appl. Math. Comput.* 175 (2006) 497–507.
- [25] S. Abbasbandy, R.A.V. Gorder, M. Hajiketabi, The lie-group shooting method for radial symmetric solutions of the Yamabe equation, *Comput. Model. Eng. Sci.* 104 (2015) 67–81.
- [26] S. Abbasbandy, M.S. Hashemi, C.-S. Liu, The lie-group shooting method for solving the Bratu equation, *Commun. Nonlinear Sci. Numer. Simul.* 16 (2011) 4238–4249.
- [27] C.W. Chang, C.S. Liu, An implicit lie-group iterative scheme for solving the nonlinear Klein–Gordon and Sine–Gordon equations, *Appl. Math. Model.* 40 (2016) 1157–1167.
- [28] C.S. Liu, Group preserving scheme for Backward heat conduction problems, *Int. J. Heat Mass Transf.* 47 (2004) 2567–2576.
- [29] C.S. Liu, Two-dimensional bilinear oscillator: group-preserving scheme and steady-state motion under harmonic loading, *Int. J. Non-linear Mech.* 38 (2003) 1581–1602.
- [30] C.W. Chang, C.S. Liu, The backward group preserving scheme for multi-dimensional nonhomogeneous and nonlinear backward wave problems, *Appl. Math. Model.* 38 (2014) 4027–4048.
- [31] S. Abbasbandy, M.S. Hashemi, Group preserving scheme for the Cauchy problem of the laplace equation, *Eng. Anal. Bound. Elem.* 35 (2011) 1003–1009.
- [32] C.S. Liu, C.W. Chang, A novel mixed group preserving scheme for the inverse Cauchy problem of elliptic equations in annular domains, *Eng. Anal. Bound. Elem.* 36 (2012) 211–219.
- [33] E.J. Kansa, Multiquadrics a scattered data approximation scheme with applications to computational fluid-dynamics-i, *Comput. Math. Appl.* 19 (1990) 127–145.
- [34] E.J. Kansa, Multiquadrics a scattered data approximation scheme with applications to computational fluid dynamics-II, *Comput. Math. Appl.* 19 (1990) 147–161.

- [35] E.J. Kansa, R.C. Aldredge, L. Ling, Numerical simulation of two-dimensional combustion using mesh-free methods, *Eng. Anal. Bound. Elem.* 33 (2009) 940–950.
- [36] S.K. Vanani, A. Aminataei, On the numerical solution of neural delay differential equations using multiquadric approximation scheme, *Bull. Korean Math. Soc.* 45 (2008) 663–670.
- [37] R.L. Hardy, Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.* 76 (1971) 1705–1915.
- [38] S. Abbasbandy, H.R. Ghehsareh, I. Hashim, A meshfree method for the solution of two-dimensional cubic nonlinear Schrödinger equation, *Eng. Anal. Bound. Elem.* 37 (2013) 885–898.
- [39] S. Abbasbandy, H.R. Ghehsareh, I. Hashim, Numerical analysis of a mathematical model for capillary formation in tumor angiogenesis using a meshfree method based on the radial basis function, *Eng. Anal. Bound. Elem.* 36 (12) (2012) 1811–1818.
- [40] S. Abbasbandy, H.R. Ghehsareh, M. Alhuthali, H. Alsulami, Comparison of meshless local weak and strong forms based on particular solutions for a non-classical 2-d diffusion model, *Eng. Anal. Bound. Elem.* 39 (2014) 121–128.
- [41] A. Shirzadi, L. Ling, S. Abbasbandy, Meshless simulations of the two-dimensional fractional-time convection–diffusion–reaction equations, *Eng. Anal. Bound. Elem.* 36 (2012) 1522–1527.
- [42] S. Rippa, An algorithm for selecting a good value for the parameter  $c$  in radial basis function interpolation, *Adv. Comput. Math.* 11 (1999) 193–210.