Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 7, No. 2, 2019, pp. 276-288

Analysis of meshless local radial point interpolant on a model in population dynamics

Elyas Shivanian*
Department of Applied Mathematics,
Imam Khomeini International University,
Qazvin, 34149-16818, Iran.
E-mail: shivanian@sci.ikiu.ac.ir

Hedayat Fatahi
Department of Mathematics, Baneh Branch,
Islamic Azad University, Baneh, Iran.
E-mail: h.fatahi@iaubaneh.ac.ir

Abstract In this work, we present an improvement of the spectral meshless radial point interpolation (SMRPI) method to uncover a simulation behavior of the population dynamic model which mathematically is a nonlinear partial integro-differential equation. This PDE is a kind of competition strategy in which equivalent individuals match for the same supplies. Moreover, this boundary value problem is a particular type of reaction-diffusion problem augmented to an integral term corresponding to the nonlocal consumption of resources. As a result of applying meshless method, it does not matter how the geometry of the domain is complicated because the method enjoys the element free adoption. Applying the SMRPI on the two-dimensional integral equation leads to a linear system of algebraic equations which is easy to treat. Finally, some numerical experiments are presented to show the reliable results.

Keywords. Spectral meshless radial point interpolation (SMRPI) method, Radial basis function, Partial integro-differential equation.

2010 Mathematics Subject Classification. 65M06, 65M12, 65M22.

1. Introduction

Consider the following kind of reaction-diffusion equation which is a type of nonlinear integro-differential boundary value problem [31, 33]

$$\frac{\partial u}{\partial t} = \Delta u + Ku(\mathbf{x},t)\left(1 - au(\mathbf{x},t) - b\int_{\Omega}\Psi(\mathbf{x}-\mathbf{y})u(\mathbf{y},t)\mathrm{d}\mathbf{y}\right) + h(\mathbf{x},t),\ \ \mathbf{x}\in\Omega,\ \ t\in[0,T], \tag{1.1}$$

where $\Omega = [-1,1]^2$ and the initial and boundary conditions are

$$u(\mathbf{x},t) = 0,\ \ \mathbf{x}\in\partial\Omega, \qquad u(\mathbf{x},0) = g(\mathbf{x})\ \ \mathbf{x}\in\Omega. \tag{1.2}$$

Received: 23 February 2018 ; Accepted: 11 March 2019.

∗ Corresponding author.

In Eq. (1.1), K, a and b are non-negative constants. Also, $\Psi(\mathbf{x})$ is the kernel function which is known and $h(\mathbf{x},t)$ is a known smooth function. The $u(\mathbf{x},t)$ is the mass of the population in Eq. (1.1). Furthermore, the first and the second terms of the right-hand side of Eq. (1.1) are diffusion and mortality of individuals in a population, respectively [30].

Some works have been done on developing numerical methods for solving the reaction-diffusion and reaction-subdiffusion equations [20]. As a non-complete list, one can be referred to the resources that will be coming. Henry and Wearne [24] derived a reaction-diffusion equation from a continuous-time random walk model with temporal memory and sources. Yuste and Acedo [42] combined the forward time centered space (FTCS) method to obtain an explicit FTCS scheme for solving the diffusion equation, in which the order of convergence was $\mathcal{O}(\delta t + h^2)$. Yuste in [41], discussed weighted averaged finite difference methods for solving diffusion equations with different formulae of the discretization to time derivative. In [38], Sokolov et al. showed that reaction-subdiffusion equations may not resemble the corresponding reaction-diffusion ones and are not obtained by a trivial change of the diffusion operator for a subdiffusion one. Chen et al. [14] proposed a Fourier method for analyzing the stability and convergence of an implicit difference approximation with the global accuracy, in which the order of convergence was $\mathcal{O}(\delta t + h^2)$. Cui in [15] considered a compact finite difference scheme for solving one-dimensional diffusion equation, where the order of convergence was $\mathcal{O}(\delta t + h^4)$. Chen et al. [13] considered an implicit and an explicit difference method for the reaction-subdiffusion equation with study of stability and convergence those methods by a Fourier analysis, where the convergence with the order was $\mathcal{O}(\delta t + h^2)$. Baeumer et al. [9] developed a practical method for numerical solution of reaction-diffusion equations, based on operator splitting.

Meshless methods have attracted much attention in recent years [1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 16, 17, 18, 19, 21, 22, 23, 25, 26, 27, 28, 29, 32, 35, 39, 40, 43]. One of the very useful and easy ways to apply among meshless methods has been improved by Shivanian [34, 36] which is a kind of spectral meshless radial point interpolation (SMRPI) method. As a result of applying this meshless method, it does not matter how the geometry of the domain is complicated because the method enjoys the element free adoption. Our aim in this work is the development of this technique to obtain the solution of the nonlinear integro-differential reaction-diffusion equation (1.1)-(1.2). It will be seen the method with high performance, while is numerical based, can easily overcome the nonlinear term and obtain more accurate approximate solutions in a fast way.

In section 2, we introduce the SMRPI scheme and obtain the corresponding shape functions. In section 3 and 4, time discretization approximation for implementation of the SMRPI is given and obtained. In section 5, we present algorithm with detailed material. We report the numerical experiments of solving problem (1.1)-(1.2) for two test problems in section 6. Finally a conclusion is given in section 7.

2. The outline of SMRPI

This section has been adapted from reference [37]. Consider a continuous function $u(\mathbf{x})$ defined in a domain Ω, which is represented by a set of field nodes. The $u(\mathbf{x})$ at a point of interest $\mathbf{x}$ is approximated in the form

$$u(\mathbf{x}) \simeq \sum_{i=1}^{n} R_i(\mathbf{x})a_i + \sum_{j=1}^{m} P_j(\mathbf{x})b_j = \mathbf{R}^T(\mathbf{x})\mathbf{a} + \mathbf{P}^T(\mathbf{x})\mathbf{b}, \tag{2.1}$$

where $R_i(\mathbf{x})$ is a radial basis function (RBF), n is the number of RBFs, $P_j(\mathbf{x})$ is monomial in the space coordinate $\mathbf{x}$, and m is the number of polynomial basis functions. The coefficients a_i and b_j are unknown which should be determined. In order to determine a_i and b_j in Eq. (2.1), a support domain is formed for the point of interest at $\mathbf{x}$, and n field nodes are included in the support domain (support domain is usually a disk with radius r_s). Coefficients a_i and b_j can be determined by enforcing Eq. (2.1) to be satisfied at these n nodes surrounding the point of interest $\mathbf{x}$. Therefore, by the idea of interpolation, Eq. (2.1) is converted to the following form:

$$u(\mathbf{x}) \simeq \mathbf{\Phi}^T(\mathbf{x})\mathbf{U}_s = \sum_{i=1}^{n} \phi_i(\mathbf{x})u_i. \tag{2.2}$$

where $\phi_i(\mathbf{x})$'s are called the RPIM shape functions which have the Kronecker delta function property, that is

$$\phi_i(\mathbf{x}_j) = \begin{cases} 1, & \text{for } i = j, j = 1, 2, ..., n \ , \\ 0, & \text{for } i \neq j, i, j = 1, 2, ..., n. \end{cases} \tag{2.3}$$

This is because the RPIM shape functions are created to pass thorough nodal values. Moreover, the shape functions are the partitions of unity, i.e.

$$\sum_{i=1}^{n} \phi_i(\mathbf{x}) = 1, \tag{2.4}$$

for more details about RPIM shape functions and the way they are constructed, the readers are referred to see [37]. Now, we construct operational matrices which are essential tools of present approach. Operational matrices make the technique more appropriate to handle partial differential equations with high derivatives. Suppose that the number of total nodes covering the domain of the problem, i.e., $\bar{\Omega} = \Omega \bigcup \partial\Omega$ is N. On the other hand, we know that n depends on point of interest $\mathbf{x}$ (so, after that we call it $n_{\mathbf{x}}$) in Eq. (2.2) which is the number of nodes included in support domain $\Omega_{\mathbf{x}}$ corresponding to the point of interest $\mathbf{x}$ (for example $\Omega_{\mathbf{x}}$ can be a disk centered at $\mathbf{x}$ with radius r_s). Therefore, we have $n_{\mathbf{x}} \leq N$ and then Eq. (2.2) can be modified as

$$u(\mathbf{x}) \simeq \mathbf{\Phi}^T(\mathbf{x})\mathbf{U}_s = \sum_{j=1}^{N} \phi_j(\mathbf{x})u_j. \tag{2.5}$$

In fact, corresponding to node $\mathbf{x}_j$ there is a shape function $\phi_j(\mathbf{x}), j = 1, 2, ..., N$, we define $\Omega^c_{\mathbf{x}} = \{\mathbf{x}_j : \mathbf{x}_j \notin \Omega\}$ then obviously from Eq. (2.3)

$$\forall \mathbf{x}_j \in \Omega^c_{\mathbf{x}} : \phi_j(\mathbf{x}) = 0. \tag{2.6}$$

The derivatives of $u(\mathbf{x})$ are easily obtained as

$$\frac{\partial u(\mathbf{x})}{\partial x} \simeq \sum_{j=1}^{N} \frac{\partial \phi_j(\mathbf{x})}{\partial x} u_j, \qquad \frac{\partial u(\mathbf{x})}{\partial y} \simeq \sum_{j=1}^{N} \frac{\partial \phi_j(\mathbf{x})}{\partial y} u_j, \tag{2.7}$$

and for high derivatives of $u(\mathbf{x})$

$$\frac{\partial^s u(\mathbf{x})}{\partial x^s} \simeq \sum_{j=1}^{N} \frac{\partial^s \phi_j(\mathbf{x})}{\partial x^s} u_j, \qquad \frac{\partial^s u(\mathbf{x})}{\partial y^s} \simeq \sum_{j=1}^{N} \frac{\partial^s \phi_j(\mathbf{x})}{\partial y^s} u_j, \tag{2.8}$$

where $\frac{\partial^s (.)}{\partial x^s}$ and $\frac{\partial^s (.)}{\partial y^s}$ are s'th derivatives with respect to x and y. Let $u_x^{(s)}(.) := \frac{\partial^s (.)}{\partial x^s}$, $u_y^{(s)}(.) := \frac{\partial^s (.)}{\partial y^s}$ and set $\mathbf{x} = \mathbf{x}_i$ in Eq. (2.7). Then the following matrix form is given

$$\begin{pmatrix} u'_{x_1} \\ u'_{x_2} \\ \vdots \\ u'_{x_N} \end{pmatrix} = \begin{pmatrix} \frac{\partial \phi_1(\mathbf{x}_1)}{\partial x} & \frac{\partial \phi_2(\mathbf{x}_1)}{\partial x} & \cdots & \frac{\partial \phi_N(\mathbf{x}_1)}{\partial x} \\ \frac{\partial \phi_1(\mathbf{x}_2)}{\partial x} & \frac{\partial \phi_2(\mathbf{x}_2)}{\partial x} & \cdots & \frac{\partial \phi_N(\mathbf{x}_2)}{\partial x} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \phi_1(\mathbf{x}_N)}{\partial x} & \frac{\partial \phi_2(\mathbf{x}_N)}{\partial x} & \cdots & \frac{\partial \phi_N(\mathbf{x}_N)}{\partial x} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}, \tag{2.9}$$

$$\begin{pmatrix} u'_{y_1} \\ u'_{y_2} \\ \vdots \\ u'_{y_N} \end{pmatrix} = \begin{pmatrix} \frac{\partial \phi_1(\mathbf{x}_1)}{\partial y} & \frac{\partial \phi_2(\mathbf{x}_1)}{\partial y} & \cdots & \frac{\partial \phi_N(\mathbf{x}_1)}{\partial y} \\ \frac{\partial \phi_1(\mathbf{x}_2)}{\partial y} & \frac{\partial \phi_2(\mathbf{x}_2)}{\partial y} & \cdots & \frac{\partial \phi_N(\mathbf{x}_2)}{\partial y} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \phi_1(\mathbf{x}_N)}{\partial y} & \frac{\partial \phi_2(\mathbf{x}_N)}{\partial y} & \cdots & \frac{\partial \phi_N(\mathbf{x}_N)}{\partial y} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}. \tag{2.10}$$

We denote the above coefcients matrices as Dx, Dy and also, for high order derivatives, we have the following matrix-vector multiplications:

$$U_x^{(s)} = D^{(s)}xU, \qquad U_y^{(s)} = D^{(s)}yU, \tag{2.11}$$

where

$$U_x^{(s)} = \left(u_{x_1}^{(s)}, u_{x_2}^{(s)}, ..., u_{x_N}^{(s)}\right)^{tr}, \qquad U_y^{(s)} = \left(u_{y_1}^{(s)}, u_{y_2}^{(s)}, ..., u_{y_N}^{(s)}\right)^{tr}, \tag{2.12}$$

$$D_{x_{ij}}^{(s)} = \frac{\partial^s \phi_j(\mathbf{x}_i)}{\partial x^s}, \tag{2.13}$$

$$D_{y_{ij}}^{(s)} = \frac{\partial^s \phi_j(\mathbf{x}_i)}{\partial y^s}, \tag{2.14}$$

$$U = (u_1, u_2, ..., u_N)^{tr}. \tag{2.15}$$

3. The time discretisation approximation

To treat the time derivative of the population model, let us adopt a time-stepping scheme based on Crank-Nicolson technique as

$$\begin{aligned}\frac{\partial u(\mathbf{x},t)}{\partial t} &\cong \frac{1}{\triangle t}\left(u^{k+1}(\mathbf{x})-u^{k}(\mathbf{x})\right),\\ \Delta u(\mathbf{x},t) &\cong \frac{1}{2}\left(\Delta u^{k+1}(\mathbf{x})+\Delta u^{k}(\mathbf{x})\right),\end{aligned} \tag{3.1}$$

where $u^k(\mathbf{x}) = u(\mathbf{x}, k\triangle t)$. Assuming $\lambda = \frac{\triangle t}{2}$ and $H(\mathbf{x}; k, k+1) = h(\mathbf{x}, (k+1)\triangle t) + h(\mathbf{x}, k\triangle t)$, we derive

$$\begin{aligned}(1-K\lambda)u^{(k+1)}-\lambda\Delta u^{(k+1)} = \quad & (1+K\lambda)u^{(k)}+\lambda\Delta u^{(k)}-2Ka\lambda\tilde{u}^2\\ & -2Kb\lambda\tilde{u}\int_{\Omega}\Psi(\mathbf{x}-\mathbf{y})\tilde{u}(\mathbf{y})\mathrm{d}\mathbf{y}\\ & +\lambda H(\mathbf{x};k,k+1),\end{aligned} \tag{3.2}$$

where $\tilde{u}$ is the latest and best available approximation of u.

4. Discretization of SMRPI formulation

In this part, we manipulate on Eq. (3.2) to derive the system of discrete algebraic equations. Taking into account N regularly located nodal points on the both boundary and domain of the model, we set them so that the distance between two consecutive nodes in each direction is constant and equal to h. Suppose that we have obtained $u(\mathbf{x}_i, k\triangle t)$, $i = 1, 2, ..., N$ already, our purpose, for the next step, is to obtain $u(\mathbf{x}_i, (k+1)\triangle t)$, $i = 1, 2, ..., N$. Therefore, we have N unknowns and to discover them we need N equations. It will be observed later that there is one equation corresponding to each node. For those nodes belong to inside of the domain, i.e., for $\mathbf{x}_i \in$ interior Ω, to obtain the discrete equations from Eq. (3.2), let us substitute approximation formula (2.8) into Eq. (3.2) to obtain:

$$\begin{aligned}&(1-K\lambda)u^{k+1}(\mathbf{x})-\lambda\left(\sum_{j=1}^{N}\frac{\partial^2\phi_j(\mathbf{x})}{\partial x^2}u_j^{k+1}+\sum_{j=1}^{N}\frac{\partial^2\phi_j(\mathbf{x})}{\partial y^2}u_j^{k+1}\right)=\\ &(1+K\lambda)u^{k}(\mathbf{x})+\lambda\left(\sum_{j=1}^{N}\frac{\partial^2\phi_j(\mathbf{x})}{\partial x^2}u_j^{k}+\sum_{j=1}^{N}\frac{\partial^2\phi_j(\mathbf{x})}{\partial y^2}u_j^{k}\right)\\ &-2Ka\lambda\tilde{u}^2-2Kb\lambda\tilde{u}\int_{\Omega}\Psi(\mathbf{x}-\mathbf{y})\tilde{u}(\mathbf{y})\mathrm{d}\mathbf{y}+\lambda H(\mathbf{x};k,k+1),\end{aligned} \tag{4.1}$$

Now, let us set $\mathbf{x} = \mathbf{x}_i$, $i = 1, 2, 3, ..., N_\Omega$ (N_Ω denotes the number of nodes for inside of Ω) in the above equation. Then, using notations (2.13)-(2.14), we have

$$\begin{aligned}
&\left[(1 - K\lambda)\delta_{ij} - \lambda\left(\mathbf{D}^{(2)}x_{ij} + \mathbf{D}^{(2)}y_{ij}\right)\right]u_j^{(k+1)} = \\
&\left[(1 + K\lambda)\delta_{ij} + \lambda\left(\mathbf{D}^{(2)}x_{ij} + \mathbf{D}^{(2)}y_{ij}\right)\right]u_j^{(k)} \\
&-2Ka\lambda\tilde{u_i}^2 - 2Kb\lambda\tilde{u_i}\int_\Omega \Psi(\mathbf{x} - \mathbf{y})\tilde{u}(\mathbf{y})\mathrm{d}\mathbf{y} + \lambda H(\mathbf{x}_i; k, k+1),
\end{aligned} \tag{4.2}$$

where $\tilde{u}_i,\ i = 1, 2, ..., N$ is the best available approximation of $\hat{u}_i$. We will apply a simple predictor-corrector (P-C) scheme to treat $\tilde{u}_i$ in next section.

Now, we turn back to Eq. (4.2), suppose that $u_i^{(k)},\quad i = 1, 2, ..., N$ are known already. We have to approximate the integral

$$\int_\Omega \Psi(\mathbf{x} - \mathbf{y})\tilde{u}(\mathbf{y})\mathrm{d}\mathbf{y},$$

for next iteration. The procedure is to apply two dimensional Gaussian quadrature rule, but the problem is we do not have any expression for $\tilde{u}(\mathbf{y})$. In order to overcome this problem, we use two dimensional cubic spline interpolation to estimate the values of $\tilde{u}^{(k)}(\mathbf{y})$ at Gaussian points. This is done by the command $interp2(\cdots, spline)$ easily in MATLAB software program. Therefore, we turn into

$$\int_\Omega \Psi(\mathbf{x} - \mathbf{y})\tilde{u}(\mathbf{y})\mathrm{d}\mathbf{y} = \sum_{p=1}^{n_G} \omega_p \Psi(\mathbf{x} - \mathbf{y}_p)\tilde{u}^{(k)}(\mathbf{y}_p) = \tilde{\Psi}(\mathbf{x}), \tag{4.3}$$

and then Eq. (4.2) is changed to the following form

$$\begin{aligned}
&\left[(1 - K\lambda)\delta_{ij} - \lambda\left(\mathbf{D}^{(2)}x_{ij} + \mathbf{D}^{(2)}y_{ij}\right)\right]u_j^{(k+1)} = \\
&\left[(1 + K\lambda)\delta_{ij} + \lambda\left(\mathbf{D}^{(2)}x_{ij} + \mathbf{D}^{(2)}y_{ij}\right)\right]u_j^{(k)} \\
&-2Ka\lambda\tilde{u_i}^2 - 2Kb\lambda\tilde{u_i}\tilde{\Psi}(\mathbf{x}_i) + \lambda H(\mathbf{x}_i; k, k+1).
\end{aligned} \tag{4.4}$$

for the all nodes in the interior of the domain Ω.

5. Numerical implementation for SMRPI method

From the problem description, it is straightforward to set

$$\forall k : u^{(k)}(\mathbf{x}_i) = 0, \quad \mathbf{x}_i \in \partial\Omega = \{(|x| = 1 \text{ or } |y| = 1) \text{ and } -1 \le x, y \le 1\}, \tag{5.1}$$

for nodes located on the boundary. Therefore, the matrix forms of Eqs. (4.4) and (5.1) for all N nodal points located in both inside and the boundary of the domain are defined as

$$\begin{aligned}
&\left[(1 - K\lambda)\delta_{ij} - \lambda\left(\mathbf{D}^{(2)}x_{ij} + \mathbf{D}^{(2)}y_{ij}\right)\right]u_j^{(k+1)} = \\
&\left[(1 + K\lambda)\delta_{ij} + \lambda\left(\mathbf{D}^{(2)}x_{ij} + \mathbf{D}^{(2)}y_{ij}\right)\right]\hat{u}_j^{(k)} \\
&-c_i\tilde{u}_i^2 - d_i\tilde{u}_i\tilde{\Psi}_i + e_i(k, k+1),
\end{aligned} \tag{5.2}$$

where

$$c_i = 2Ka\lambda \quad d_i = 2Kb\lambda, \quad \tilde{\Psi}_i = \tilde{\Psi}(\mathbf{x}_i), \quad e_i(k,k+1) = \lambda H(\mathbf{x}_i; k, k+1). \tag{5.3}$$

Consider the notations

$$\begin{aligned}
&\mathbf{A}_{ij} = (1 - K\lambda)\delta_{ij} - \lambda \left(\mathbf{D}^{(2)} x_{ij} + \mathbf{D}^{(2)} y_{ij} \right), \\
&\mathbf{B}_{ij} = (1 + K\lambda)\delta_{ij} + \lambda \left(\mathbf{D}^{(2)} x_{ij} + \mathbf{D}^{(2)} y_{ij} \right), \\
&\mathbf{C} = \text{diagonal}(c_1, c_2, ..., c_N), \\
&\mathbf{D} = \text{diagonal}(d_1, d_2, ..., d_N), \\
&\mathbf{E}^k = \left[e_1(k,k+1), e_2(k,k+1), ..., e_N(k,k+1) \right]^T, \\
&\hat{U} = (u_i)_{N\times 1}, \quad \tilde{U} = (\tilde{u}_i)_{N\times 1}, \\
&\tilde{\Psi} = (\tilde{\Psi}_i)_{N\times 1},
\end{aligned}$$

then, Eq. (5.2) yields

$$\mathbf{A}\hat{U}^{(k+1)} = \mathbf{B}\hat{U}^{(k)} - \mathbf{C}\tilde{U}^2 - \mathbf{D}\tilde{U}\tilde{\Psi} + \mathbf{E}^k. \tag{5.4}$$

Moreover, to satisfy Eq. (5.1), for all nodes belong to the boundary, i.e. $\mathbf{x}_i \in \partial\Omega$, we have to set

$$\mathbf{C}_{ii} = \mathbf{D}_{ii} = \mathbf{E}_i^k = 0, \quad \forall j : \mathbf{B}_{ij} = 0, \quad \mathbf{A}_{ij} = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \tag{5.5}$$

We pay attention that the product in Eq. (5.4), i.e. $\tilde{U}^2$ and $\tilde{U}\tilde{\Psi}$ is of Hadamard product type.

Now, we adopt the following procedure for dealing with the nonlinearity:
At first step, we set $\tilde{U} = \hat{U}^{(k)} (k = 0)$; then, Eq. (5.4) can be solved by a system of linear algebraic equations for the unknown $\hat{U}^{(1)}$, after that, we apply the following Crank-Nicolson scheme:

$$\bar{U}^2 = \frac{1}{2}\left[(\hat{U}^{(0)})^2 + (\hat{U}^{(1)})^2 \right], \quad \bar{U}\bar{\Psi} = \frac{1}{2}\left[\hat{U}^{(0)}\tilde{\Psi}^{(0)} + \hat{U}^{(1)}\tilde{\Psi}^{(1)} \right]. \tag{5.6}$$

Now, by Eq. (5.6), we set $\tilde{U}^2 = \bar{U}^2$ and $\tilde{U}\tilde{\Psi} = \bar{U}\bar{\Psi}$, then by the help of these updated $\tilde{U}^2$ and $\tilde{U}\tilde{\Psi}$, the system of linear algebraic equations (5.4) is resolved again for the unknown $U^{(1)}$. In general, we have the following iterative process

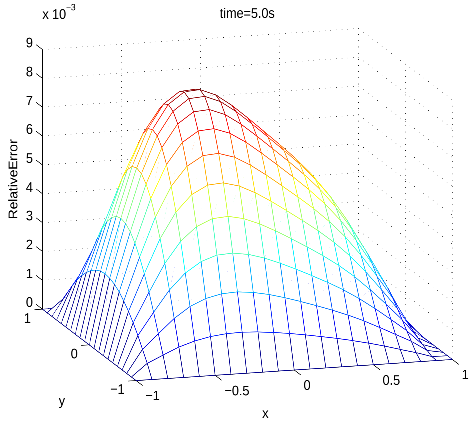
$$\begin{aligned}
\mathbf{A}\hat{U}_{l+1}^{(k)} = \quad & \mathbf{B}\hat{U}^{(k)} - \frac{1}{2}\mathbf{C}\left[(\hat{U}^{(k)})^2 + (\hat{U}_l^{(k)})^2 \right] - \frac{1}{2}\mathbf{D}\left[\hat{U}^{(k)}\tilde{\Psi}^{(k)} + \hat{U}_l^{(k)}\tilde{\Psi}_l^{(k)} \right] \\
& + \mathbf{E}^k,
\end{aligned} \tag{5.7}$$

for each time level, in where l denotes the number of iterations in each time level. This process is iterated until that unknown quantity converges to within a prescribed number of tolerance. In other words, in each time level, these iterations are continued until satisfying the following condition:

$$\left\| \hat{U}_{l+1}^{(k)} - \hat{U}_l^{(k)} \right\|_\infty \leq \epsilon, \tag{5.8}$$

where ϵ is a fixed number, in the numerical experiment it is considered to be 10^{-10} and also $\| \cdot \|_\infty$ is infinity norm.

FIGURE 1. Relative error of approximate solution at $t = 5$ for Example 1.



6. NUMERICAL EXPERIMENT

In this section, we show the results which we obtained for two examples, which have been taken from the literature [33], using the meshless method described in the above. The domain integrals are evaluated with 4 points Gaussian quadrature. To show the behavior of the solution and the efficiency of the proposed method, the following relative error (RE) is applied

$$RE = \frac{\left|U_{exact}(\mathbf{x}_i) - U_{approx}(\mathbf{x}_i)\right|}{\parallel U_{exact}(\mathbf{x}) \parallel_\infty},$$

where $U_{exact}(\mathbf{x}_i)$ and $U_{approx}(\mathbf{x}_i)$ are achieved by exact and approximate solution on points $\mathbf{x}_i$ and N is number of nodal points.

In the current work, we have used the thin plate spline (TPS) as radial basis functions in Eq. (2.1). This RBF is defined as follows:

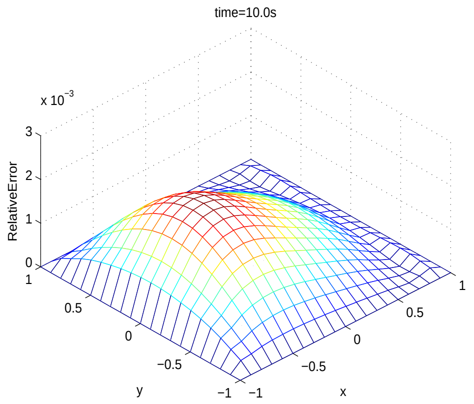
$$R(\mathbf{x}) = r^{2m}\ln(r), \quad m = 1, 2, 3, ... \tag{6.1}$$

In both problems, while setting $m = 2$, the regular node distribution is used. The radius of support domain to moving least squares approximation is $r_s = 4.2h$, where h is the distance between the nodes in x or y direction. This size is significant enough to have sufficient number of nodes (n) and gives an appropriate approximation.

Example 1. We take $K = 2, a = 1$, and $b = \frac{1}{2}$, in equation (1.1) assume that

$$u(x, y, t) = x(1 - x^2)(1 - y^2)\sinh(\frac{1}{1 + t^2}),$$

FIGURE 2. Relative error of approximate solution at $t = 10$ for Example 1.

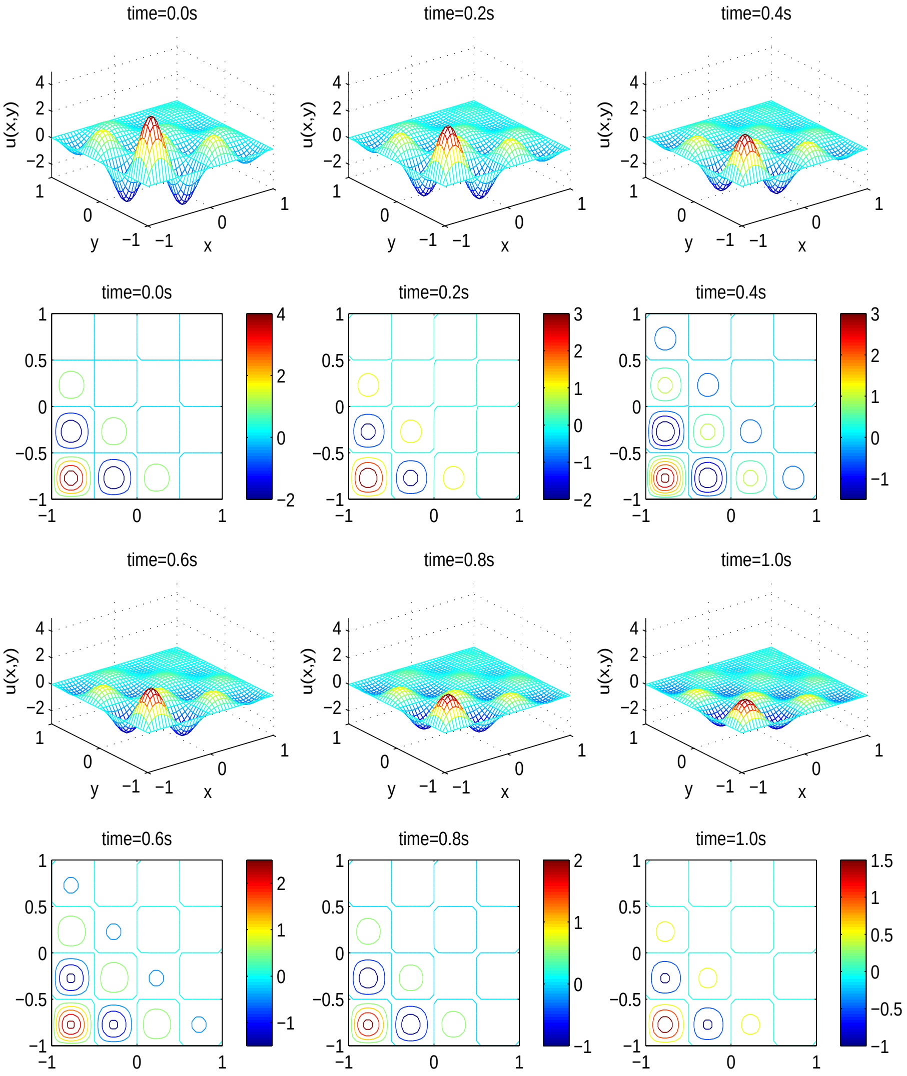


is the exact solution of the problem (1.1)-(1.2) with $\Psi(\mathbf{x}) = x + y$ and

$$\begin{aligned}
&h(x,y,t) = \\
&2x^6y^4\sinh^2\left(\frac{1}{t^2+1}\right) - 4x^6y^2\sinh^2\left(\frac{1}{t^2+1}\right) + 2x^6\sinh^2\left(\frac{1}{t^2+1}\right) - \\
&4x^4y^4\sinh^2\left(\frac{1}{t^2+1}\right) + 8x^4y^2\sinh^2\left(\frac{1}{t^2+1}\right) - 4x^4\sinh^2\left(\frac{1}{t^2+1}\right) - \\
&\frac{16}{45}x^3y^2\sinh^2\left(\frac{1}{t^2+1}\right) - 2x^3y^2\sinh\left(\frac{1}{t^2+1}\right) - \frac{2tx^3y^2\cosh\left(\frac{1}{t^2+1}\right)}{\left(t^2+1\right)^2} + \\
&\frac{16}{45}x^3\sinh^2\left(\frac{1}{t^2+1}\right) + \frac{2tx^3\cosh\left(\frac{1}{t^2+1}\right)}{\left(t^2+1\right)^2} + 2x^2y^4\sinh^2\left(\frac{1}{t^2+1}\right) - \\
&4x^2y^2\sinh^2\left(\frac{1}{t^2+1}\right) + 2x^2\sinh^2\left(\frac{1}{t^2+1}\right) + \frac{16}{45}xy^2\sinh^2\left(\frac{1}{t^2+1}\right) - \\
&4xy^2\sinh\left(\frac{1}{t^2+1}\right) + \frac{2txy^2\cosh\left(\frac{1}{t^2+1}\right)}{\left(t^2+1\right)^2} - \frac{16}{45}x\sinh^2\left(\frac{1}{t^2+1}\right) + \\
&6x\sinh\left(\frac{1}{t^2+1}\right) - \frac{2tx\cosh\left(\frac{1}{t^2+1}\right)}{\left(t^2+1\right)^2}.
\end{aligned} \tag{6.2}$$

Figures 1 and 2 presents the relative error of approximate solution at $t = 5, 10$ with $N = 441$ and $\triangle t = 0.001$.

FIGURE 3. Numerical solutions of Example 2, by SMRPI, at times $t = 0, 0.2, 0.4, 0.6, 0.8$ and 1, with $\triangle t = 0.001$ and $h = 0.05$.



Example 2. We set $K = a = 1$, and $b = 0$. The exact solution of problem (1.1)-(1.2) is taken as

$$u(x, y, t) = \sin(2\pi x)\sin(2\pi y)\exp(-x - y - t),$$

where $h(x, y, t)$ is defined accordingly i.e.

$$\begin{aligned} h(x,y,t) &= \sin^2(2\pi x)\sin^2(2\pi y)e^{-2t-2x-2y} - 4\sin(2\pi x)\sin(2\pi y)e^{-t-x-y} + \\ &\quad 8\pi^2\sin(2\pi x)\sin(2\pi y)e^{-t-x-y} + 4\pi\cos(2\pi x)\sin(2\pi y)e^{-t-x-y} + \\ &\quad 4\pi\sin(2\pi x)\cos(2\pi y)e^{-t-x-y}. \end{aligned} \tag{6.3}$$

The numerical solutions of this problem, with time step $\triangle t = 0.001$, at $t = 0, 0.2, 0.4, 0.6, 0.8$ and 1, and corresponding contours have been shown in Figure 3.

7. Conclusions

In this paper, a new spectral meshless radial point interpolation (SMRPI) method has been to solve a partial integro-differential equation arising in population dynamics. The present methods is based on meshless methods and benefits from spectral collocation techniques. The interpolation with the help of conditionally positive definite radial basis functions has been used to construct shape (basis) functions which have Kronecker delta function property. The method does not need any domain element and so it is independent of the geometry of the domain.

Acknowledgments

The authors acknowledge financial support from the Baneh Branch, Islamic Azad University. The authors are grateful to anonymous reviewers for carefully reading this paper and for their comments and suggestions which have improved the paper.

References

[1] S. Abbasbandy, H. Roohani Ghehsareh, and 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. Boundary. Elem., *36*(12) 2012, 1811–1818.

[2] S. Abbasbandy, H. Roohani Ghehsareh, and I. Hashim, *A meshfree method for the solution of two-dimensional cubic nonlinear schrödinger equation,* Eng. Anal. Boundary. Elem., *37*(6) 2013, 885–898.

[3] S. Abbasbandy and E. Shivanian, *Multiple solutions of mixed convection in a porous medium on semi-infinite interval using pseudo-spectral collocation method,* Commun. Nonlinear Sci. Numer. Simulat., *16* 2011, 2745–2752.

[4] P. Assari, H. Adibi, and M. Dehghan, *A meshless method for solving nonlinear two-dimensional integral equations of the second kind on non-rectangular domains using radial basis functions with error analysis,* Comput. Appl. Math., *239* 2013, 72–92.

[5] P. Assari, H. Adibi, and M. Dehghan, *A meshless discrete Galerkin (MDG) method for the numerical solution of integral equations with logarithmic kernels,* J. Comput. Appl. Math., *267* 2014, 160–181.

[6] S. N. Atluri, *The Meshless Method (MLPG) for Domain and BIE Discretizations*, Tech. Science Press, 2004.

[7] S. N. Atluri and S. Shen, *The meshless local Petrov-Galerkin (MLPG) method: a simple and less costly alternative to the finite element and boundary element methods,* Comput. Modeling Engrg. Sci., *3* 2002, 11–51.

[8] S. N. Atluri and S. Shen, *The Meshless Local Petrov-Galerkin (*MLPG*) Method,* Tech. Science Press, 2002.

[9] B. Baeumer, M. Kovács, and M. Meerschaert, *Numerical solutions for fractional reaction–diffusion equations,* Computers & Mathematics with Applications, *55*(10) 2008, 2212–2226.

[10] F. N. Bai, D. M. Li, J. F. Wang, and Y. M. Cheng, *An improved complex variable element-free Galerkin method for two-dimensional elasticity problems,* Chin. Phys. B, *21*(2) 2012, 020204–10.

[11] T. Belytschko, Y. Y. Lu, and L. Gu, *Element-free Galerkin methods,* Int.J. Numer. Methods Eng., *37*(2) 1994, 229–256.

[12] T. Belytschko, Y. Y. Lu, and L. Gu, *Element free Galerkin methods for static and dynamic fracture,* Int. J. Solids Struct., *32* 1995, 2547–2570.

[13] C. Chen, F. Liu, and K. Burrage, *Finite difference methods and a fourier analysis for the fractional reaction–subdiffusion equation,* Applied Mathematics and Computation, *198*(2) 2008, 754–769.

[14] C. Chen, F. Liu, I. Turner, and V. Anh, *A fourier method for the fractional diffusion equation describing sub-diffusion,* Journal of Computational Physics, *227*(2) 2007, 886–897.

[15] M. Cui, *Compact finite difference method for the fractional diffusion equation,* Journal of Computational Physics, *228*(20) 2009, 7792–7804.

[16] B. D. Dai and Y. M. Cheng, *An improved local boundary integral equation method for two-dimensional potential problems, Int. J. Appl. Mech.*, 2(2):421–436, 2010.

[17] M. Dehghan and A. Ghesmati, *Numerical simulation of two-dimensional sine-gordon solitons via a local weak meshless technique based on the radial point interpolation method (*RPIM*),* Comput. Phys. Commun., *181* 2010, 772–786.

[18] M. Dehghan and R. Salehi, *A meshless local Petrov-Galerkin method for the time-dependent maxwell equations,* J. Comput. Appl. Math., *268* 2014, 93–110.

[19] M. Dehghan and A. Shokri, *A numerical method for solution of the two dimensional sine-Gordon equation using the radial basis functions,* Math. Comput. Simul., *79* 2008, 700–715.

[20] M. Dehghan and M. Abbaszadeh, *Variational multiscale element free galerkin (vmefg) and local discontinuous galerkin (ldg) methods for solving two-dimensional brusselator reaction–diffusion system with and without cross-diffusion,* Computer Methods in Applied Mechanics and Engineering, *300* 2016, 770–797.

[21] M. Dehghan, M. Abbaszadeh, and A. Mohebbi, *A meshless technique based on the local radial basis functions collocation method for solving parabolic–parabolic patlak–keller–segel chemotaxis model,* Engineering Analysis with Boundary Elements, *56* 2015, 129–144.

[22] Y. Duan and Y. J. Tan, *A meshless Galerkin method for Dirichlet problems using radial basis functions,* J. Comput. Appl. Math., *196*(2) 2006, 394–401.

[23] A. Fili, A. Naji, and Y. Duan. *Coupling three-field formulation and meshless mixed Galerkin methods using radial basis functions,* J. Comput. Appl. Math., *234*(8) 2010, 2456–2468.

[24] B. I. Henry and S. L. Wearne, *Fractional reaction–diffusion,* Physica A: Statistical Mechanics and its Applications, *276*(3) 2000, 448–455.

[25] S. Jakobsson, B. Andersson, and F. Edelvik, *Rational radial basis function interpolation with applications to antenna design,* J. Comput. Appl. Math., *233*(4) 2009, 889–904.

[26] E. Kansa, *Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics.* I. *surface approximations and partial derivative estimates,* Comput. Math. Appl., *19*(8-9) 1990, 127–145.

[27] G. R. Liu and Y. T. Gu, *An Introduction to Meshfree Methods and Their Programing*, Springer, 2005.

[28] D. Mirzaei and M. Dehghan, *Meshless local Petrov-Galerkin (*MLPG*) approximation to the two dimensional sine-Gordon equation,* J. Comput. Appl. Math., *233*(10) 2010, 2737–2754.

[29] M. J. Peng, D. M. Li, and Y. M. Cheng, *The complex variable element-free galerkin (*CVEFG*) method for elasto-plasticity problems,* Eng. Struc., *33*(1) 2011, 127–135.

[30] A. P. S. Selvadurai, *Partial Differential Equations in Mechanics*, Springer, Berlin, 2000.

[31] F. Shakeri and M. Dehghan, *Solution of a model describing biological species living together using the variational iteration method,* Mathematical and Computer Modelling, *48*(5-6) 2008, 685–699.

[32] A. Shirzadi, L. Ling, and S. Abbasbandy, *Meshless simulations of the two-dimensional fractional-time convection-diffusion-reaction equations,* Eng. Anal. Boundary. Elem., *36* 2012, 1522–1527.

[33] E. Shivanian, *Analysis of meshless local radial point interpolation (*MLRPI*) on a nonlinear partial integro-differential equation arising in population dynamics,* Eng. Anal. Boundary. Elem., *37* 2013, 1693–1702.

[34] E. Shivanian, *Analysis of meshless local and spectral meshless radial point interpolation (*MLRPI *and* SMRPI*) on 3-d nonlinear wave equations,* Ocean Eng., *89* 2014, 173–188.

[35] E. Shivanian, *Meshless local* P*etrov-*G*alerkin (*MLPG*) method for three-dimensional nonlinear wavee quations via moving leasts quares approximation,* Eng. Anal. Boundary. Elem., *50* 2015, 249–257.

[36] E. Shivanian, *A new spectral meshless radial point interpolation (*SMRPI*) method: A well-behaved alternative to the meshless weak forms,* Eng. Anal. Boundary. Elem., *54* 2015, 1–12.

[37] E. Shivanian, *A new spectral meshless radial point interpolation (*SMRPI*) method: A well-behaved alternative to the meshless weak forms,* Engineering Analysis with Boundary Elements, *54* 2015, 1–12.

[38] I. M. Sokolov, M. G. W. Schmidt, and F. Sagués, *Reaction-subdiffusion equations,* Physical Review E, *73*(3) 2006, 031102.

[39] T. Tang, *On spectral methods for Volterra integral equations and the convergence analysis,* J. Comput. Math., *26*(6) 2008, 825–837.

[40] L. N. Trefethen, *Spectral methods in* MATLAB, PA: SIAM, Philadelphia, 2000.

[41] S. B. Yuste, *Weighted average finite difference methods for fractional diffusion equations,* Journal of Computational Physics, *216*(1) 2006, 264–274.

[42] S. B. Yuste and L. Acedo, *An explicit finite difference method and a new von neumann-type stability analysis for fractional diffusion equations,* SIAM Journal on Numerical Analysis, *42*(5) 2005, 1862–1874.

[43] Y. X. Zhang and Y. J. Tan, *Meshless schemes for unsteady* N*avier-*S*tokes equations in vorticity formulation using radial basis functions,* J. Comput. Appl. Math., *192*(2) 2009, 328–338.