



Approximation of continuous surface differential operators with the generalized moving least-squares (GMLS) method for solving reaction–diffusion equation

Mehdi Dehghan¹ · Niusha Narimani¹

Received: 9 September 2016 / Revised: 29 August 2018 / Accepted: 15 September 2018 /

Published online: 7 November 2018

© SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional 2018

Abstract

In this paper, a meshless approximation based on generalized moving least squares is applied to solve the reaction–diffusion equations on the sphere and red-blood cell surfaces. The proposed method is based on the projected gradient of the shape functions, and it approximates the Laplace operator defined on the surfaces that is called Laplace–Beltrami. This technique only requires nodes at locations on the surface and the corresponding normal vectors to the surface. To discretize the time variable, an explicit time technique based on the fourth-order Runge–Kutta is used. Some numerical results on Turing and Fitzhugh–Nagumo partial differential equations are given for showing patterns which are appeared in biological phenomena.

Keywords Generalized moving least-squares approximation · Projected gradient of the shape functions · Runge–Kutta time discretization · Turing and Fitzhugh–Nagumo models · Biological pattern formation · Spot and stripe patterns · Spiral wave patterns in excitable media

Mathematics Subject Classification MSC 65M70 · MSC 65L06

1 Introduction

In this paper, we consider the following reaction–diffusion equation on the sphere and red-blood cell surfaces (Fuselier and Wright 2013):

Communicated by Abimael Loula.

✉ Mehdi Dehghan
mdehghan@aut.ac.ir ; mdehghan.aut@gmail.com

Niusha Narimani
n.narimani.aut@gmail.com

¹ Department of Applied Mathematics, Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology, No. 424, Hafez Ave., Tehran 15914, Iran

$$\begin{cases} \frac{\partial u}{\partial t} = \delta_u \Delta_{\mathbb{M}} u + f_u(u, v, t), \\ \frac{\partial v}{\partial t} = \delta_v \Delta_{\mathbb{M}} v + f_v(u, v, t), \end{cases} \quad (1.1)$$

in which $u, v : \mathbb{M} \rightarrow \mathbb{R}$, $\delta_u, \delta_v \geq 0$, f_u, f_v are (possibly non-linear) scalar functions, and $\Delta_{\mathbb{M}}$ is the Laplace–Beltrami operator defined on the surface $\mathbb{M} \subset \mathbb{R}^3$ (Fuselier and Wright 2013). Here, we introduce two important reaction–diffusion equations in developmental biology defined on the surfaces.

1.1 Turing model

The mathematical model of Turing pattern formation on the surfaces is of the following form (Fuselier and Wright 2013):

$$\begin{cases} \frac{\partial u}{\partial t} = \delta_u \Delta_{\mathbb{M}} u + \alpha u(1 - \tau_1 v^2) + v(1 - \tau_2 u), \\ \frac{\partial v}{\partial t} = \delta_v \Delta_{\mathbb{M}} v + \beta v(1 + \frac{\alpha \tau_1}{\beta} uv) + u(\gamma + \tau_2 v), \end{cases} \quad (1.2)$$

where u and v represent concentrations of the chemical substances. In fact, the substances u and v are called the activator with a slow diffusion rate and the inhibitor with a fast diffusion rate, respectively (Gierer and Meinhardt 1972). Moreover, the cubic coupling parameter τ_1 points the stripes patterns, while the quadratic coupling parameter τ_2 refers to spot patterns (Fuselier and Wright 2013). As is said in Sugai et al. (2017), the activator increases the concentrations of both the activator and the inhibitor, while the inhibitor represses the concentration of the activator. We can also derive different patterns such as spots and stripes by changing diffusivity rates (Fuselier and Wright 2013; Mosekilde 1996). As mentioned in Fuselier and Wright (2013) and related literature works, the spot pattern formations are more robust than the stripes and take far less time to reach steady state.

The history of primary Turing model comes back to 1952. In that year, Alan Turing in his paper Turing (1952) showed how chemical reactions can create patterns in the nature by reaction–diffusion equation. As is said in Maini et al. (2012), Turing hypothesized that the patterns which we observe during embryonic development arise in response to a spatial pre-pattern in biochemicals which he called morphogens. Until now, Turing mechanism is considered experimentally in different morphological events such as, hair pattern of mammals, feather patterns of birds, regeneration of hydra, and pigment pattern formation in the skin of zebrafish (Kondo et al. 2009; Murray 1993). As is said in Sheth et al. (2012), to achieve a leopard spot pattern, the activator u produces a dark colored fur, while the inhibitor v prevents the dark color formation which gives the background gold color, and the overall result is black spots on a gold background (Sheth et al. 2012). Besides, there are other complex patterns which can be derived via modified reaction–diffusion system (see Fig. 1).

Several Turing models have been developed such as the Gierer–Meinhardt model (Gierer and Meinhardt 1972), Gray–Scott model (Gray and Scott 1983), Lengyel–Epstein model (Lengyel and Epstein 1991), Brusselator model (Prigogine and Lefever 1968), Schnakenberg model (Schnakenberg 1979), and Selkov model (Selkov 1968). The solutions of Turing models show the pattern formation in the developing animal embryo through interactions between two diffusible substances (Kondo and Miura 2010). Since the exact solutions for Turing models are not available, finding the numerical solutions will be important. For example, spot pattern on cheetah is simulated by applying numerical method on reaction–diffusion

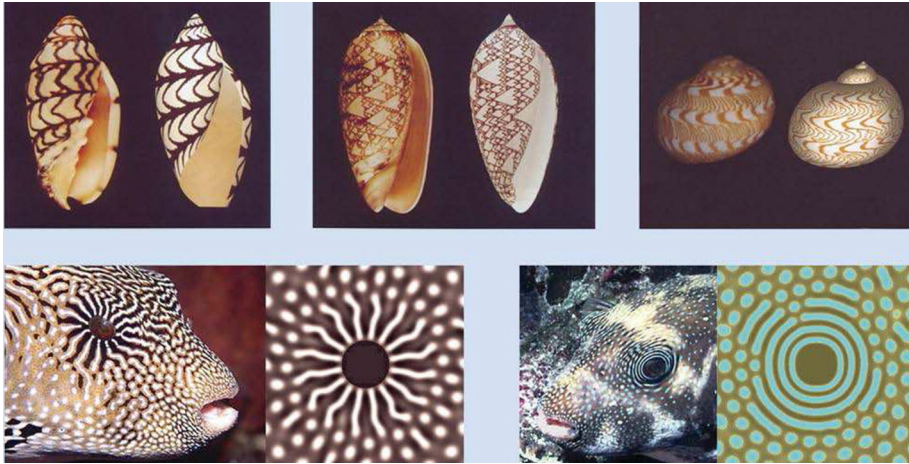


Fig. 1 Biological patterns created by modified reaction–diffusion mechanisms for types of seashell and fish. Taken from Kondo and Miura (2010)

equation in Cooper and Leeuwen (2013) and Varvruska (2015). Besides, patterns on the wings of a butterfly have been simulated in Sekimura et al. (2000) with a numerical method. The other numerical techniques for showing Turing patterns can be found in the literature works.

1.2 Fitzhugh–Nagumo model

The well-known Fitzhugh–Nagumo (FHN) model is formulated as a reaction–diffusion system in the following form (Fuselier and Wright 2013):

$$\begin{cases} \frac{\partial u}{\partial t} = \delta_u \Delta_{\mathbb{M}} u + \frac{1}{\alpha} u(1 - u)(u - \frac{v + b}{a}), \\ \frac{\partial v}{\partial t} = \delta_v \Delta_{\mathbb{M}} v + (u - v), \end{cases} \tag{1.3}$$

where u denotes the membrane potential, and v is a recovery variable (Tonnelier 2002). The parameters a , b , and α govern the reaction Kinetic. In addition, δ_u and δ_v represent the diffusivities rates (Fuselier and Wright 2013). As is mentioned in Fuselier and Wright (2013), when $\alpha \ll 1$, the value of u is 0 or 1 almost everywhere (Fuselier and Wright 2013). The FHN model is often applied as a generic model for excitable media (Nomura and Glass 1996).

The patterns of action potential propagation along the axon of a single neuron or in a network of cortical neurons can be obtained by numerical simulations of FHN model (Alford and Auchmuty 2006). In addition, this model can be applied to simulate the propagation of wave in heart tissue (Cherry and Fenton 2008) (see Fig. 2).

The history of FHN model returns to 1928 when Van der Pol and Van der Mark introduced a mathematical model for showing the heart’s dynamic (Van der Pol and Van der mark 1928). In 1952, Hodgkin and Huxley presented another mathematical model known as Hodgkin–Huxley for explaining the behavior of nerve cells in a squid giant axon (Hodgkin Huxley 1952). In 1961, Fitzhugh proposed a new model of the electrical activities of nerve membrane by using of simple representative of Hodgkin-Huxley model (Fitzhugh 1961). After that, in

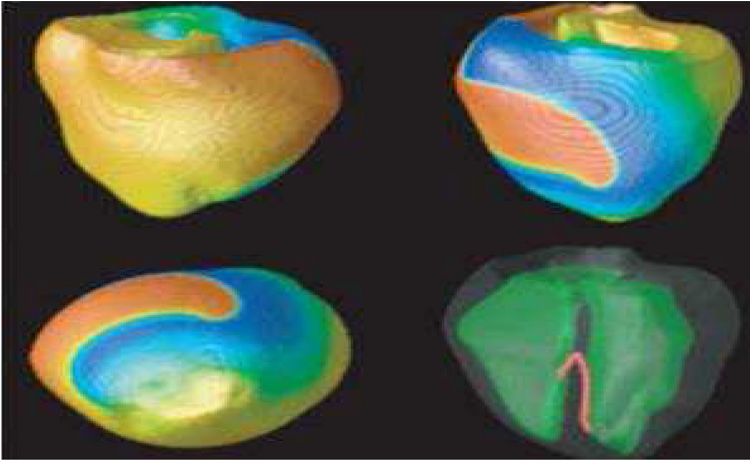


Fig. 2 Spiral wave on the surface of a human heart. Taken from Cherry and Fenton (2008)

1962, Nagumo et al. (1962) confirmed Fitzhugh's model and proposed a new mathematical model of the nerve axon. Finally, this model is known as Fitzhugh–Nagumo which interprets qualitative characteristics of electrical impulses along nerve and cardiac fibers (Xu et al. 2014). The other types of FHN model which have been derived from the original one can be found in Izhikevich (2010), Kostova et al. (2004), and Roqoreanu et al. (2000).

1.3 The literature review

In recent years, there are different research works on the numerical solutions of the reaction-diffusion equations. For example, Discontinuous Galerkin (DG) method (Epshteyn and Kurganov 2008, and Zhu et al. 2009), various finite difference (FD) techniques (Smiely 2009; Tyson et al. 1999, 2000), and finite volume methods (Chertock and Kurganov 2008; Chiu and Yu 2007). Authors of Shakeri and Dehghan (2011) combined the spectral element method with finite volume method, and obtained a numerical scheme for solving Turing model in two-dimension. Authors of Tatari et al. (2011) applied the moving least-squares (MLS) approximation to solve Turing equation in \mathbb{R}^2 . In Ilati and Dehghan (2015), the meshless local weak form has been applied to solve Brusselator model and spike dynamics in the Gierer–Meinhardt system. In addition, they solved Turing-type models via direct meshless local Petrov–Galerkin (DMLPG) method (Ilati and Dehghan 2017). In Guin and Mandal (2014), the finite difference scheme was used to solve predator–prey model. More numerical simulations of the reaction–diffusion equations can be found in Gambino et al. (2013), Guin et al. (2012), and Guin (2015). Also, for solving Brusselator model, there are some methods such as Adomian decomposition method (Adomian 1995), second-order finite difference scheme (Twizell et al. 1999), modified Adomian decomposition method (Wazwaz 2000), dual-reciprocity boundary element method (Ang 2003), differential quadrature method (Mittal and Jiwari 2011), and radial basis functions collocation method (Islam et al. 2010). Recently, the authors of Mohammadi et al. (2014) solved this problem with reproducing kernel Hilbert space technique. In Dehghan and Fakhar-Izadi (2011), pseudospectral method has been applied to find the numerical solution of Nagumo equation in one-dimensional space. Authors of Dehghan and Abbaszadeh (2016) obtained the numerical solution of Turing and

FHN models in two-dimensional space with element-free Galerkin (EFG) method which is combined with the moving kriging least-squares technique and radial point interpolation scheme. A high-order kernel method based on radial basis functions (RBFs) is applied to solve reaction–diffusion equations on the surfaces embedded in \mathbb{R}^3 (Fuselier and Wright 2013). In Shankar et al. (2015), the authors introduced a radial basis function (RBF)–finite difference (FD) method and applied it for solving reaction–diffusion equations on the surfaces. Authors of (Lehto et al. 2017) provided a radial basis function (RBF) compact finite difference (FD) scheme for solving reaction–diffusion equations on surfaces. In Peng and Zhang (2016), the method of multiple time scales is employed to derive the amplitude equations which are the cubic Stuart–Landau equation in the supercritical case and the quintic in the subcritical case. In Fua et al. (2016), a new mixed finite-element method is introduced for approximating the steady reaction–diffusion equations. Other numerical methods for finding the numerical solutions of Turing models can be observed in Bergdorf et al. (2010), Bertalmio et al. (2001), Calhoun and Helzel (2009), Gomatam and Amdjadi (1997), MacDonald and Ruuth (2009), Piret (2012), Ruuth and Merriman (2008), and references therein.

1.4 The structure of remainder of this paper

In this paper, we use a direct method based on generalized moving least squares (GMLS) for approximating the spatial variables of Turing and FHN equations on the unit sphere and red-blood cell surfaces. The remainder of the paper is as follows: Sect. 2 gives a brief discussion of the GMLS approximation. The discussion of the continuous surface differential operators is given in the next section. In Sect. 4, we approximate the Laplace–Beltrami operator using the projected gradient of the shape functions. In Sect. 5, the time discretization of the semi-discretized reaction–diffusion equation using the fourth-order Runge–Kutta method is given. In Sect. 6, some numerical simulations are reported for solving Turing and FHN models on the unit sphere and red-blood cell surfaces.

2 Generalized moving least squares approximation

The generalized moving least squares approximation on the spheres has been introduced by Mirzaei (2017). In this section, we briefly review the main idea of this approximation on the two-dimensional unit sphere, i.e., \mathbb{S}^2 . For further information, the interested reader can refer to Mirzaei (2017) and the other references such as Fasshauer (2007), Mirzaei (2016), Mirzaei et al. (2012), and Wendland (2005).

Suppose that $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ is N scattered points on the unit sphere. In addition, $\mathbf{P} = \{p_1, p_2, \dots, p_Q\}$ is the basis of the space of all spherical harmonics of degree m which is denoted by Y_m^2 . Besides, we consider u is a scalar real-valued function defined on \mathbb{S}^2 . The approximation of u by GMLS can be written as follows (Mirzaei 2016, 2017; Mirzaei et al. 2012):

$$u(\mathbf{x}) \approx \widehat{u(\mathbf{x})} = \sum_{j \in I(\mathbf{x})} a_j(\mathbf{x})u(\mathbf{x}_j), \quad \mathbf{x} \in \mathbb{S}^2,$$

where $a_j(\mathbf{x})$ $j = 1, 2, \dots, |I(\mathbf{x})|$ are the shape functions which are obtained from the minimizing of the weighted discrete- ℓ_2 (Lancaster and Salkauskas 1981, Mirzaei 2017, Salehi and Dehghan 2013). In addition, $I(\mathbf{x})$ denotes the family of indices of points in support $\mathbf{x} \in \mathbb{S}^2$, and it is defined as follows:

$$I(\mathbf{x}) = \{j \in \{1, 2, \dots, N\} : \text{dist}(\mathbf{x}, \mathbf{x}_j) \leq \delta\},$$

in which δ is the radius of support domain, and $\text{dist}(\mathbf{x}, \mathbf{x}_j)$ shows the geodesic distance between \mathbf{x} and \mathbf{x}_j on \mathbb{S}^2 .

Definition 1 The weight function W is defined as follows (Mirzaei 2016, 2017; Mirzaei et al. 2012; Wendland 2005):

$$W : \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathbb{R},$$

such that

$$W(\mathbf{x}, \mathbf{x}_j) = \begin{cases} \phi(r), & 0 \leq r \leq 1, \\ 0, & r > 1, \end{cases}$$

where ϕ is a radial function and $r := \frac{\text{dist}(\mathbf{x}, \mathbf{x}_j)}{\delta}$.

In this paper, we use the Gaussian weight function as follows (Mirzaei 2016, 2017; Mirzaei et al. 2012):

$$W(\mathbf{x}, \mathbf{x}_j) = \begin{cases} \frac{e^{-\varepsilon^2 r^2} - e^{-\varepsilon^2}}{1 - e^{-\varepsilon^2}}, & 0 \leq r \leq 1, \\ 0, & r > 1, \end{cases}$$

where ε is a constant (shape) parameter. The approximation of $D^\alpha u(\mathbf{x})$ for multi-index $\alpha \in \mathbb{N}_0^d$, where $|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_d$, can be written as follows (Mirzaei 2016, 2017; Mirzaei et al. 2012):

$$D^\alpha u(\mathbf{x}) \approx \widehat{D^\alpha u(\mathbf{x})} = \sum_{j \in I(\mathbf{x})} a_{j,\alpha}(\mathbf{x}) u_j, \quad \mathbf{x} \in \mathbb{S}^2.$$

The vector form of $\widehat{D^\alpha u(\mathbf{x})}$ can be written as follows:

$$\widehat{D^\alpha u(\mathbf{x})} = [a_{1,\alpha}(\mathbf{x}), a_{2,\alpha}(\mathbf{x}), \dots, a_{|I(\mathbf{x})|,\alpha}(\mathbf{x})] \mathbf{u} := D^\alpha p^T(\mathbf{x}) (P^T W P)^{-1} P^T W \mathbf{u},$$

in which

$$\begin{aligned} D^\alpha p^T(\mathbf{x}) &= [D^\alpha p_1(\mathbf{x}), D^\alpha p_2(\mathbf{x}), \dots, D^\alpha p_Q(\mathbf{x})], \\ P &= P(\mathbf{x}) = (P_k(\mathbf{x}_j)) \in \mathbb{R}^{|I(\mathbf{x})| \times Q}, \\ W &= W(\mathbf{x}) = \text{diag}\{W(\mathbf{x}, \mathbf{x}_j)\} \in \mathbb{R}^{|I(\mathbf{x})| \times |I(\mathbf{x})|}, \\ \mathbf{u} &= [u_1, u_2, \dots, u_{|I(\mathbf{x})|}]^T \in \mathbb{R}^{|I(\mathbf{x})|}. \end{aligned}$$

3 Continuous surface differential operators

In this section, we present the mathematical formulations of the differential operators which are defined on the surfaces embedded in \mathbb{R}^3 . All discussions proposed here are taken from Fuselier and Wright (2013). For more details, the interested reader can refer to Fuselier and Wright (2013) and references therein.

We suppose that \mathbb{M} is a smooth-embedded surface (sub-manifold) of \mathbb{R}^3 with no boundary. At each point $\mathbf{x} = (x, y, z)^T$ on \mathbb{M} , we denote the normal vector as $\mathbf{n} = (n^x, n^y, n^z)^T$, and the tangent vector to \mathbb{M} at \mathbf{x} is shown with $T_x \mathbb{M}$, i.e., Fuselier and Wright (2013):

$$\nabla_{\mathbb{M}} := P \nabla = (I - \mathbf{n} \mathbf{n}^T) \nabla, \tag{3.1}$$

such that P projects vectors in \mathbb{R}^3 to $T_x\mathbb{M}$. The vector form of $\nabla_{\mathbb{M}}$ can be written as follows:

$$\nabla_{\mathbb{M}} = \begin{bmatrix} (1 - n_x n_x) \frac{\partial}{\partial x} - n_x n_y \frac{\partial}{\partial y} - n_x n_z \frac{\partial}{\partial z} \\ -n_x n_y \frac{\partial}{\partial x} + (1 - n_y n_y) \frac{\partial}{\partial y} - n_y n_z \frac{\partial}{\partial z} \\ -n_x n_z \frac{\partial}{\partial x} - n_y n_z \frac{\partial}{\partial y} + (1 - n_z n_z) \frac{\partial}{\partial z} \end{bmatrix} := \begin{bmatrix} \eta^x \\ \eta^y \\ \eta^z \end{bmatrix}. \tag{3.2}$$

Due to the definition $\nabla_{\mathbb{M}}$, i.e., Eq. (3.2), the Laplace–Beltrami operator can be written as follows:

$$\Delta_{\mathbb{M}} := \nabla_{\mathbb{M}} \cdot \nabla_{\mathbb{M}} = (P \nabla) \cdot (P \nabla) = \eta^x \eta^x + \eta^y \eta^y + \eta^z \eta^z. \tag{3.3}$$

4 The projected gradient of the shape functions

Our goal of this section is to approximate the Laplace–Beltrami operator defined on the sphere and red-blood cell surfaces at given set of points X by GMLS technique. For this purpose, the approximation of the surface differential operators which is based on deriving the projected gradient of the shape functions is introduced. Another approach with details to approximate Laplace–Beltrami operator defined on the spheres by GMLS can be found in Mirzaei (2017).

According to (3.2), at point $\mathbf{x} \in \mathbb{M}$, we have

$$\nabla_{\mathbb{M}} a_j(\mathbf{x}) = \begin{bmatrix} (1 - n_x n_x) \frac{\partial a_j(\mathbf{x})}{\partial x} - n_x n_y \frac{\partial a_j(\mathbf{x})}{\partial y} - n_x n_z \frac{\partial a_j(\mathbf{x})}{\partial z} \\ -n_x n_y \frac{\partial a_j(\mathbf{x})}{\partial x} + (1 - n_y n_y) \frac{\partial a_j(\mathbf{x})}{\partial y} - n_y n_z \frac{\partial a_j(\mathbf{x})}{\partial z} \\ -n_x n_z \frac{\partial a_j(\mathbf{x})}{\partial x} - n_y n_z \frac{\partial a_j(\mathbf{x})}{\partial y} + (1 - n_z n_z) \frac{\partial a_j(\mathbf{x})}{\partial z} \end{bmatrix}, \tag{4.1}$$

for $j = 1, 2, \dots, |I(\mathbf{x})|$. We again consider the following approximation for the function u defined on the surface \mathbb{M} :

$$u(\mathbf{x}) = \sum_{j \in I(\mathbf{x})} a_j(\mathbf{x}) u_j, \quad \mathbf{x} \in \mathbb{M}. \tag{4.2}$$

Due to (4.2), $\nabla_{\mathbb{M}} u$ at point $\mathbf{x} \in \mathbb{M}$ can be written as follows:

$$\nabla_{\mathbb{M}} u(\mathbf{x}) = \sum_{j \in I(\mathbf{x})} \nabla_{\mathbb{M}} a_j(\mathbf{x}) u_j. \tag{4.3}$$

According to (4.1) and (4.3), the components of $\nabla_{\mathbb{M}} u(\mathbf{x})$ can be written as follows:

$$\begin{aligned} \eta^x u(\mathbf{x}) &:= \sum_{j \in I(\mathbf{x})} \eta^x a_j(\mathbf{x}) u_j = \sum_{j \in I(\mathbf{x})} \left[(1 - n_x n_x) \frac{\partial a_j(\mathbf{x})}{\partial x} - n_x n_y \frac{\partial a_j(\mathbf{x})}{\partial y} - n_x n_z \frac{\partial a_j(\mathbf{x})}{\partial z} \right] u_j, \\ \eta^y u(\mathbf{x}) &:= \sum_{j \in I(\mathbf{x})} \eta^y a_j(\mathbf{x}) u_j = \sum_{j \in I(\mathbf{x})} \left[-n_x n_y \frac{\partial a_j(\mathbf{x})}{\partial x} + (1 - n_y n_y) \frac{\partial a_j(\mathbf{x})}{\partial y} - n_y n_z \frac{\partial a_j(\mathbf{x})}{\partial z} \right] u_j, \\ \eta^z u(\mathbf{x}) &:= \sum_{j \in I(\mathbf{x})} \eta^z a_j(\mathbf{x}) u_j = \sum_{j \in I(\mathbf{x})} \left[-n_x n_z \frac{\partial a_j(\mathbf{x})}{\partial x} - n_y n_z \frac{\partial a_j(\mathbf{x})}{\partial y} + (1 - n_z n_z) \frac{\partial a_j(\mathbf{x})}{\partial z} \right] u_j, \end{aligned} \tag{4.4}$$

where $\mathbf{x} \in \mathbb{M}$. Applying the collocation technique on Eq. (4.4) with $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ yields

$$\begin{aligned} \eta_X^x u(\mathbf{x}_i) &= \sum_{j \in I(\mathbf{x}_i)} \eta_X^x a_j(\mathbf{x}_i) u_j \\ &= \sum_{j \in I(\mathbf{x}_i)} \left[(1 - n_{x_i} n_{x_i}) \frac{\partial a_j(\mathbf{x}_i)}{\partial x} - n_{x_i} n_{y_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial y} - n_{x_i} n_{z_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial z} \right] u_j, \end{aligned} \tag{4.5}$$

for $i = 1, 2, \dots, N$. $\eta_X^y u(\mathbf{x}_i)$ and $\eta_X^z u(\mathbf{x}_i)$ will be obtained in a similar way. We now define

$$\begin{aligned} B_X^x &:= \left[(1 - n_{x_i} n_{x_i}) \frac{\partial a_j(\mathbf{x}_i)}{\partial x} - n_{x_i} n_{y_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial y} - n_{x_i} n_{z_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial z} \right]_{1 \leq i, j \leq N}, \\ B_X^y &:= \left[-n_{x_i} n_{y_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial x} + (1 - n_{y_i} n_{y_i}) \frac{\partial a_j(\mathbf{x}_i)}{\partial y} - n_{y_i} n_{z_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial z} \right]_{1 \leq i, j \leq N}, \\ B_X^z &:= \left[-n_{x_i} n_{z_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial x} - n_{y_i} n_{z_i} \frac{\partial a_j(\mathbf{x}_i)}{\partial y} + (1 - n_{z_i} n_{z_i}) \frac{\partial a_j(\mathbf{x}_i)}{\partial z} \right]_{1 \leq i, j \leq N}. \end{aligned} \tag{4.6}$$

According to (4.6), the discrete version of the approximation of Laplace–Beltrami operator can be written as follows:

$$\Delta_{\mathbb{M}|X} = \nabla_{\mathbb{M}} \cdot \nabla_{\mathbb{M}|X} \approx L_X := B_X^x B_X^x + B_X^y B_X^y + B_X^z B_X^z. \tag{4.7}$$

5 Time discretization

Applying GMLS approximation on Eq. (1.1) gives the following semi-discretized equation:

$$\frac{d\mathbf{u}_X}{dt} = \delta L \mathbf{u}_X + F(\mathbf{u}_X, t) =: R(\mathbf{u}_X, L, t), \tag{5.1}$$

where \mathbf{u}_X is the vector of approximation solution at points X due to the spatial discretization. Also, L is a matrix as follows:

$$L = \begin{bmatrix} L_X & \text{zeros}(N, N) \\ \text{zeros}(N, N) & L_X \end{bmatrix}_{2N \times 2N},$$

and

$$F(\mathbf{u}_X, t) = [f_u(u_X, v_X, t), f_v(u_X, v_X, t)]^T.$$

We now discretize Eq. (5.1) due to the time variable using the fourth-order Runge–Kutta method Quarteroni et al. (2007). We suppose that $t \in [0, T]$ where T is the final time. By dividing the time interval $[0, T]$ into M sub-intervals, such that $T = M\Delta t$ where Δt is the time step and by defining $t_n := n\Delta t$, the Runge–Kutta method for Eq. (5.1) can be written as follows:

$$\begin{aligned}
 k_1 &= \Delta t R(\mathbf{U}_X^n, L, t_n), \\
 k_2 &= \Delta t R\left(\mathbf{U}_X^n + \frac{k_1}{2}, L, t_n + \frac{\Delta t}{2}\right), \\
 k_3 &= \Delta t R\left(\mathbf{U}_X^n + \frac{k_2}{2}, L, t_n + \frac{\Delta t}{2}\right), \\
 k_4 &= \Delta t R(\mathbf{U}_X^n + k_3, L, t_n + \Delta t), \\
 \mathbf{U}_X^{n+1} &= \mathbf{U}_X^n + \frac{1}{6}(k_1 + 2(k_2 + k_3) + k_4), \quad n = 0, 1, \dots, M - 1,
 \end{aligned}$$

in which \mathbf{U}_X^{n+1} represents the vector of approximation solution due to the time and spatial discretizations at t_{n+1} .

6 Numerical simulations

In this section, we have reported some numerical results on the unit sphere and red-blood cell surfaces using the proposed method for solving Turing and FHN models. The surface of the unit sphere is formulated as follows (Fuselier and Wright 2013):

$$\mathbb{M} = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1\}. \tag{6.1}$$

As mentioned in St Clair (2006), the sphere has a more smooth, constant curvature throughout the surface. This gives the patterns to distribute evenly in comparison to the other surfaces.

The surface of red-blood cell is a mathematical model for human red-blood cells in static equilibrium conditions, and in the spherical coordinate, it is defined as follows (Fuselier and Wright 2013):

$$\begin{aligned}
 \mathbb{M} &= \{(x, y, z) \in \mathbb{R}^3 : x = r_0 \cos \lambda \cos \theta, \quad y = r_0 \sin \lambda \cos \theta, \\
 z &= \frac{1}{2} \sin \theta (c_0 + c_2 \cos^2 \theta + c_4 \cos^4 \theta)\}, \tag{6.2}
 \end{aligned}$$

in which $-\pi/2 \leq \theta \leq \pi/2, -\pi \leq \lambda \leq \pi, r_0 = 3.91/3.39, c_0 = 0.81/3.39, c_2 = 7.83/3.39, c_4 = -4.39/3.39$.

To show the convergence of the presented method due to the time discretization, the numerical order of convergence is calculated by the following formula:

$$\frac{\log\left(\frac{e_1}{e_2}\right)}{\log\left(\frac{\Delta t_1}{\Delta t_2}\right)},$$

in which e_1 and e_2 denote the numerical errors due to L_∞ -norm for time steps Δt_1 and Δt_2 , respectively. Since the exact solution of this problem is not available, to show the order of convergence with respect to time discretization, we compute $e = \left\| u_{\Delta t_1}^T - u_{\Delta t_2}^T \right\|_{L_\infty}$, where $u_{(\cdot)}^T$ denotes the approximation solution at the final time T with respect to Δt . Here, the number of points is considered $N = 9989$, the radius of support domain is fixed $\delta = c/\sqrt{N}$, where c is a constant parameter, and $\varepsilon = 4$ is chosen as shape parameter for the Gaussian weight function. All simulations presented here are carried out in MATLAB environment.

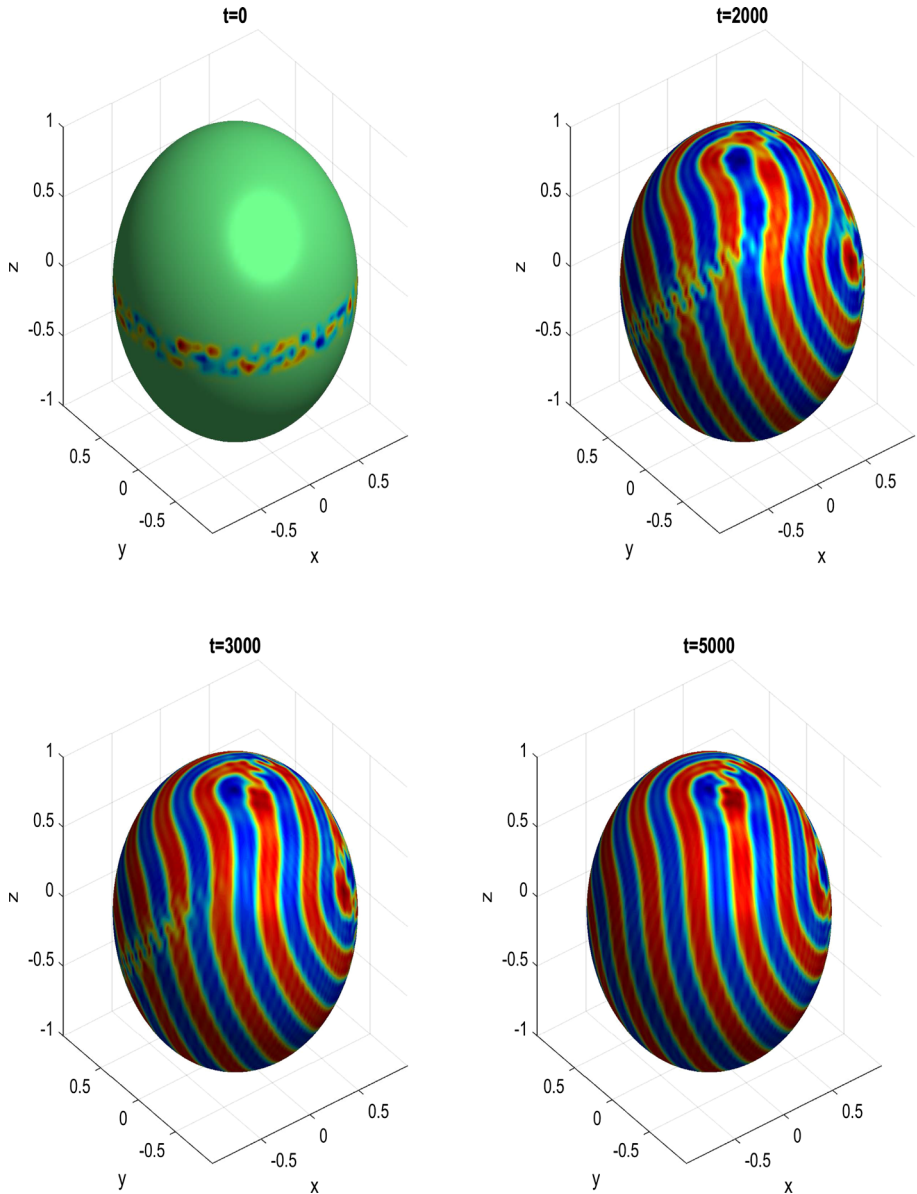


Fig. 3 Turing stripe pattern with the initial condition on the unit sphere surface for u at $t = 2000, 3000,$ and $t = 5000$ in Test problem 1

6.1 Turing patterns on the unit sphere and red-blood cell surfaces

6.1.1 Test problem 1 (stripe pattern formations)

In this example, the presented method is applied to simulate Turing patterns on the unit sphere and red-blood cell surfaces. The initial conditions of u and v are considered as random values

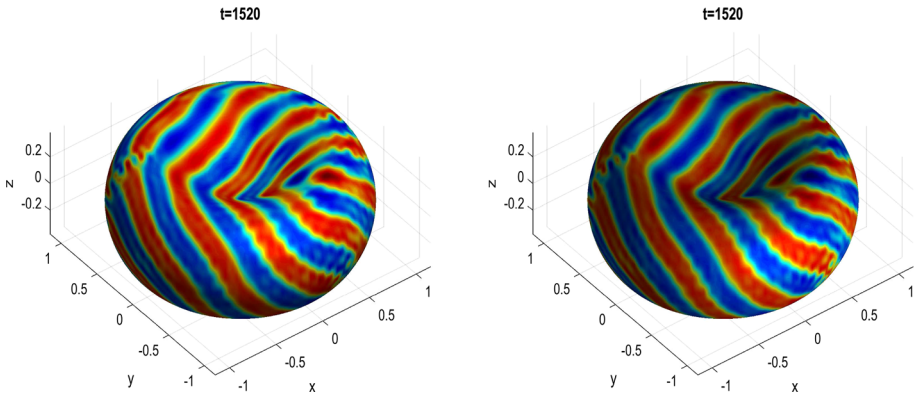


Fig. 4 Turing stripe pattern on red-blood cell surface for u (left panel) and v (right panel) at $t = 1520$ in Test problem 1

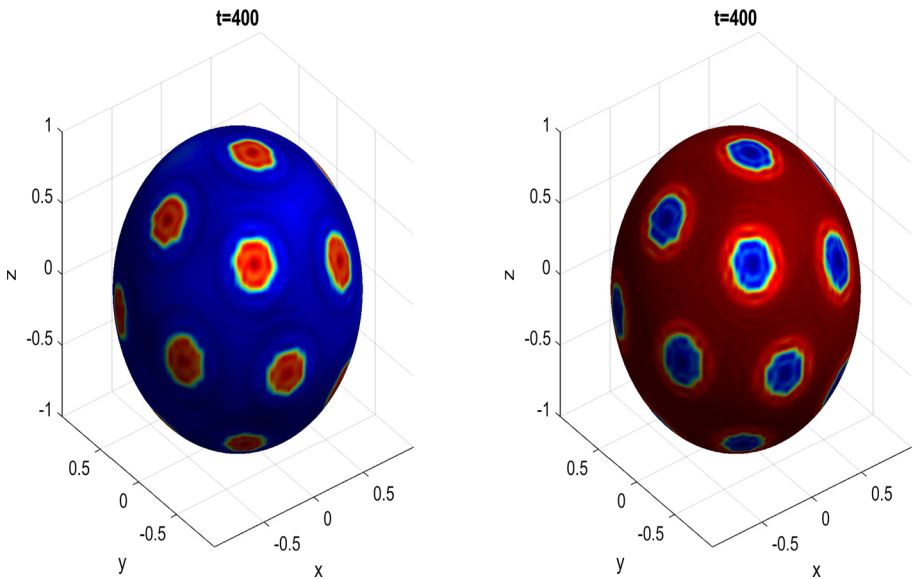


Fig. 5 Turing spot pattern on the unit sphere surface for u (left panel) and v (right panel) at $t = 400$ in Test problem 2

between -0.5 and 0.5 in a thin strip around the “equator” of each surface and $u = v = 0$ elsewhere (Fuselier and Wright 2013). To obtain stripe patterns on both surfaces, we set $\delta_v = 0.0021$, $\delta_u = 0.516\delta_v$, $\tau_1 = 3.5$, $\tau_2 = 0$, $\alpha = 0.899$, $\beta = -0.91$, and $\gamma = -\alpha$ (Fuselier and Wright 2013). Figure 3 shows the numerical solutions of u with the initial condition at different time levels $t = 2000, 3000$ and $t = 5000$ on the unit sphere using $\delta = 20/\sqrt{N}$ and $\Delta t = 0.1$. This figure demonstrates stripe pattern as is observed in the literature works. Moreover, the numerical solutions of u and v on red-blood cell surface are displayed in Fig. 4 at $t = 1520$ using $\Delta t = 0.05$.

6.1.2 Test problem 2 (spot pattern formations)

This test demonstrates the spot patterns on the unit sphere and red-blood cell surfaces with the presented method. Similar to the previous test, the initial conditions of u and v are considered as random values between -0.5 and 0.5 in a thin strip around the "equator" of each surface and $u = v = 0$ elsewhere (Fuselier and Wright 2013). To simulate spot patterns on both surfaces, the parameters are chosen as follows: $\delta_v = 0.0045$, $\delta_u = 0.516\delta_v$, $\tau_1 = 0.02$, $\tau_2 = 0.2$, $\alpha = 0.899$, $\beta = -0.91$, $\gamma = -\alpha$ (Fuselier and Wright 2013), and $t = 400$ with $\delta = 30/\sqrt{N}$ and $\Delta t = 0.05$. The numerical simulations of u and v at steady state are shown in Fig. 5 on the unit sphere. In addition, the numerical solution of u on red-blood cell is shown at time levels $t = 400$ in Fig. 6. In Table 1, the numerical L_∞ error with the order of convergence of the proposed time discretization is given by different time steps. In Table 2, we have compared the proposed approximation with RBFs' interpolation (Fuselier and Wright 2013) due to CPU time which is needed to reach the steady-state solution. Here, we have used $N = 6561$ points generated on the unit sphere and red-blood cell surfaces (Fuselier and Wright 2013) and $\Delta t = 0.05$.

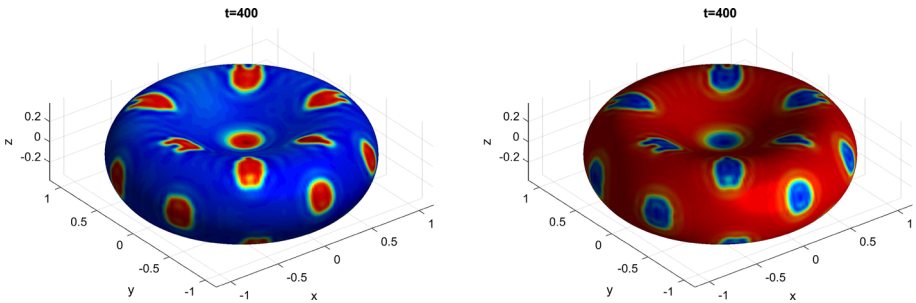


Fig. 6 Turing spot pattern on red-blood cell surface for u (left panel) and v (right panel) at $t = 400$ in Test problem 2

Table 1 The L_∞ error with the order of convergence of time discretization

$\Delta t_1 - \Delta t_2$	L_∞	Orders
$0.1 - \frac{0.1}{2}$	$2.87e-7$	—
$\frac{0.1}{2} - \frac{0.1}{4}$	$1.68e-8$	4.10
$\frac{0.1}{4} - \frac{0.1}{8}$	$1.01e-9$	4.05
$\frac{0.1}{8} - \frac{0.1}{16}$	$6.22e-11$	4.02

Table 2 CPU time required for spot patterns on the considered surfaces

Surface	GMLS approximation	RBF interpolation (Fuselier and Wright 2013)
	CPU time(s)	CPU time(s)
Sphere	1172.24	835.96
Red-blood cell	1333.23	705.94

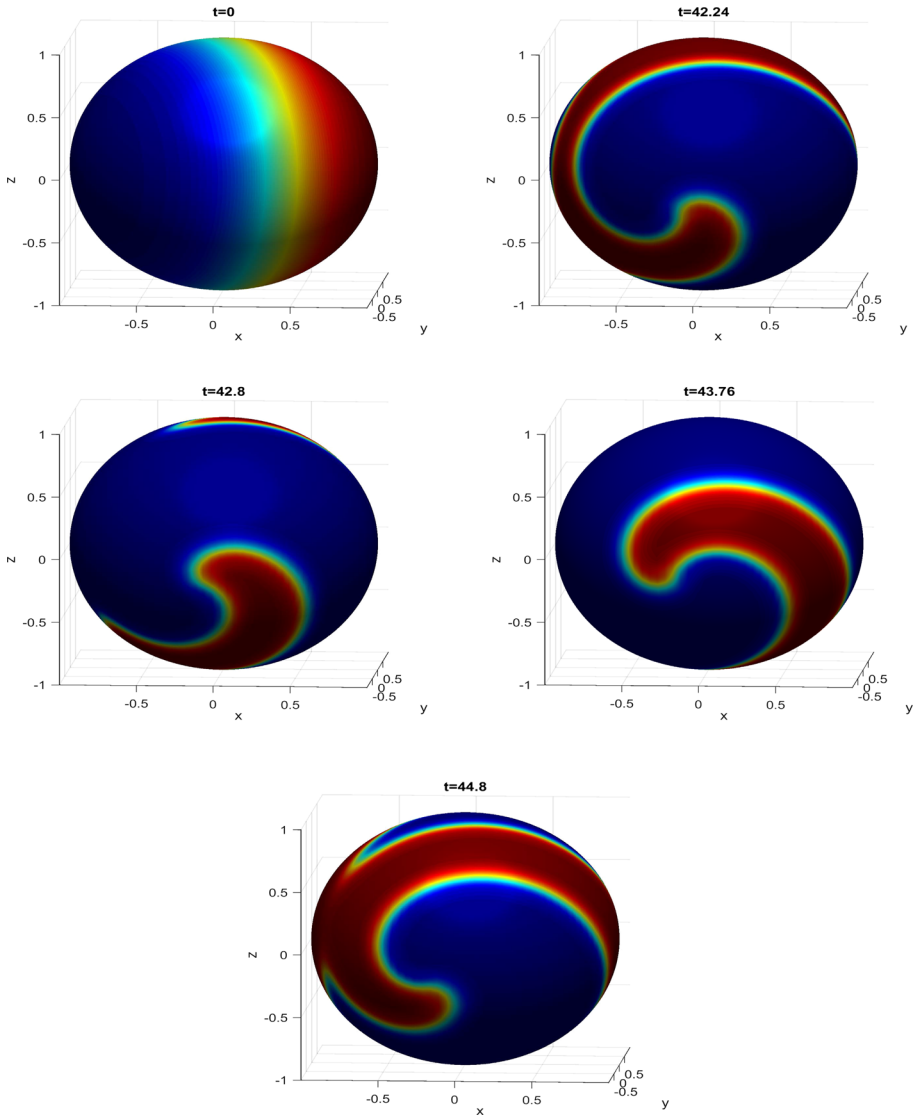


Fig. 7 Spiral wave pattern with the initial condition for u on the unit sphere surface, $u = 0$ (blue color) and $u = 1$ (red color) at $t = 42.24, 42.80, 43.76$ and $t = 44.80$ in Test problem 3

6.2 Spiral waves on the unit sphere and red-blood cell surfaces

6.2.1 Test problem 3 (Spiral waves)

In this test, we have obtained the numerical solution of FHN model on the unit sphere and red-blood cell surfaces via the presented method. The initial conditions of u and v are considered as follows (Fuselier and Wright 2013):

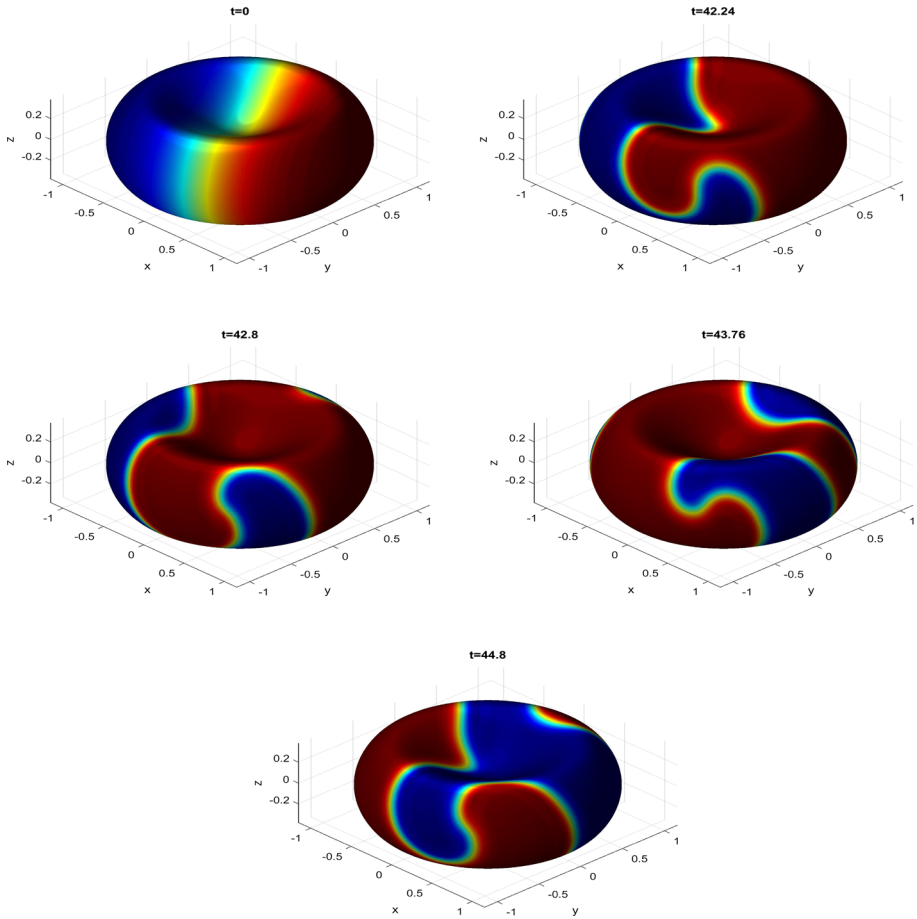


Fig. 8 Spiral wave pattern with the initial condition for u on red-blood cell surface, $u = 0$ (blue color) and $u = 1$ (red color) at $t = 42.24, 42.80, 43, 76$ and $t = 44.80$ in Test problem 3

$$u(x, y, z, 0) = \frac{1}{2} [1 + \tanh(2x + y)],$$

$$v(x, y, z, 0) = \frac{1}{2} [1 - \tanh(3z)].$$

To show spiral wave patterns on both surfaces, $\delta_v = 0, \delta_u = 2.5(2\pi/50)^2, \alpha = 0.02, a = 0.75, b = 0.02$ are considered as the parameters of the model. The numerical solutions of u with the initial condition at different time levels $t = 42.24, 42.80, 43, 76$ and $t = 44.80$ are drawn in Fig. 7 using $\delta = 30/\sqrt{N}$ and $\Delta t = 0.02$. Moreover, in Fig. 8, spiral wave pattern is simulated on red-blood cell surface using the proposed approach. As can be seen here and observations reported in Fuselier and Wright (2013), during the simulation time, these spiral waves remain intact, but they are meandering around the surfaces.

7 Conclusion

In this paper, we have successfully applied a numerical meshless method based on generalized moving least squares for solving reaction–diffusion systems on the sphere and red-blood cell surfaces which show animal coat patterns such as spot and stripe, and spiral wave patterns in excitable media. We have used the projected gradient to construct discrete approximations to the Laplace–Beltrami operator. An explicit time discretization based upon fourth-order Runge–Kutta method is used to approximate the time variable. Some numerical simulations are given for Turing and Fitzhugh–Nagumo (FHN) equations on the considered surfaces which are embedded in \mathbb{R}^3 .

Acknowledgements The authors are very grateful to reviewer for carefully reading this paper and for his (her) comments and suggestions which have improved the paper.

References

- Adomian G (1995) The diffusion Brusselator equation. *Comput Math Appl* 29:1–3
- Alford JG, Auchmuty G (2006) Rotating wave solutions of the Fitzhugh–Nagumo equations. *J Math Biol* 53(5):797–819
- Ang WT (2003) The two-dimensional reaction–diffusion Brusselator system: a dual-reciprocity boundary element solution. *Eng Anal Bound Elem* 27:897–903
- Bergdorf M, Sbalzarini I, Koumoutsakos P (2010) A Lagrangian particle method for reaction–diffusion systems on deforming surfaces. *J Math Biol* 61:649–663
- Bertalmio M, Cheng L, Osher S, Sapiro G (2001) Variational problems and partial differential equations on implicit surfaces. *J Comput Phys* 174:759–780
- Calhoun DA, Helzel C (2009) A finite volume method for solving parabolic equations on logically Cartesian curved surface meshes. *SIAM J Sci Comput* 31(6):4066–4099
- Cherry EM, Fenton FH (2008) Visualization of spiral and scroll waves in simulated and experimental cardiac tissue. *New J Phys* 10:125016
- Chertock A, Kurganov A (2008) A second-order positivity preserving central-upwind scheme for chemotaxis and haptotaxis models. *Numer Math* 111:169–205
- Cooper SB, Van Leeuwen J (2013) Alan Turing: his work and impact. Elsevier, Atlanta
- Chiu C, Yu JL (2007) An optimal adaptive time-stepping scheme for solving reaction–diffusion–chemotaxis systems. *Math Biosci Eng* 4(2):187–203
- Dehghan M, Abbaszadeh M (2016) The use of element-free Galerkin method based on moving Kriging and radial point interpolation techniques for solving some types of Turing models. *Eng Anal Bound Elem* 62:93–111
- Dehghan M, Fakhar-Izadi F (2011) Pseudospectral methods for Nagumo equation. *Int J Numer Methods Biomed Eng* 27:553–561
- Epshteyn Y, Kurganov A (2008) New interior penalty discontinuous Galerkin methods for the Keller–Segel chemotaxis model. *SIAM J Numer Anal* 47:386–408
- Fasshauer GE (2007) Meshfree approximation methods with MATLAB. World Scientific, Hackensack
- Fitzhugh R (1961) Impulses and physiological states in theoretical models of nerve membrane. *Biophys J* 1:445–466
- Fua H, Guoa H, Houb J, Zhao J (2016) A stabilized mixed finite element method for steady and unsteady reaction–diffusion equations. *Comput Methods Appl Mech Eng* 304:102–117
- Fuselier EJ, Wright GB (2013) A high-order kernel method for diffusion and reaction–diffusion equations on surfaces. *J Sci Comput* 56:535–565
- Gambino G, Lombardo MC, Sammartino M (2013) Pattern formation driven by cross-diffusion in a 2D domain. *Nonlinear Anal. RWA*. 14:1755–1779
- Gierer A, Meinhardt H (1972) A theory of biological pattern formation. *Kybernet* 12:30–39
- Gomatam J, Amdjadi F (1997) Reaction–diffusion equations on a sphere: meandering of spiral waves. *Phys Rev E* 56:3913–3919
- Gray P, Scott SK (1983) Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Isolates and other forms of multistability. *Chem Eng Sci* 38:29–43

- Guin LN, Mandal PK (2014) Effect of prey refuge on spatiotemporal dynamics of the reaction–diffusion system. *Comput Math Appl* 68:1325–1340
- Guin LN, Haque M, Mandal PK (2012) The spatial patterns through diffusion-driven instability in a predator–prey model. *Appl Math Model* 36:1825–1841
- Guin LN (2015) Spatial patterns through Turing instability in a reaction–diffusion predator–prey model. *Math Comput Simul* 109:174–185
- Hodgkin Huxley A (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol* 117:500–544
- Ilati M, Dehghan M (2015) Meshless local weak form method based on a combined basis function for numerical investigation of Brusselator model and spike dynamics in the Gierer–Meinhardt system. *Comput Model Eng Sci (CMES)* 109:325–360
- Ilati M, Dehghan M (2017) Application of direct meshless local Petrov–Galerkin (DMLPG) method for some Turing-type models. *Eng Comput* 33:107–124
- Islam SU, Ali A, Haq S (2010) A computational modeling of the behavior of the two-dimensional reaction–diffusion Brusselator system. *Appl Math Model* 34:3896–3909
- Izhikevich EM (2010) Hybrid spiking models. *Philos Trans R Soc A* 368:5061–5070
- Kondo S, Iwashita M, Yamaguchi M (2009) How animals get their skin patterns: fish pigment pattern as a live Turing wave. *Int J Dev Biol* 53:851–856
- Kondo S, Miura T (2010) Reaction–Diffusion model as a framework for understanding biological pattern formation. *Science* 329:1616–1620
- Kostova T, Ravindran R, Schonbek M (2004) Fitzhugh–Nagumo revisited: types of bifurcations, periodical forcing and stability regions by a Lyapunov functional. *Int J Bifurcat Chaos* 14:913–925
- Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. *Math Comput* 37:141–158
- Lehto E, Shankar V, Wright GB (2017) A radial basis function (RBF) compact finite difference (FD) scheme for reaction–diffusion equations on surfaces. *SIAM J Sci Comput* 39(5):2129–2151
- Lengyel I, Epstein IR (1991) Modeling of Turing structures in the chlorite-iodidemalonate acid-starch reaction system. *Science* 251:650–652
- MacDonald CB, Ruuth SJ (2009) The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM J Sci Comput* 31:4330–4350
- Maini PK, Woolley TE, Baker RE, Gaffney EA, Lee SS (2012) Turing’s model for biological pattern formation and the robustness problem. *Interface Focus* 2:487–496
- Mirzaei D (2016) Error bounds for GMLS derivatives approximations of Sobolev functions. *J Comput Appl Math* 294(1):93–101
- Mirzaei D (2017) Direct approximation on spheres using generalized moving least squares. *BIT Numer. Math.* 57:1041–1063
- Mirzaei D, Schaback R, Dehghan M (2012) On generalized moving least squares and diffuse derivatives. *IMA J Numer Anal* 32(3):983–1000
- Mittal RC, Jiwari R (2011) Numerical study of two-dimensional reaction–diffusion Brusselator system by differential quadrature method. *Int J Comput Methods Eng Sci Mech* 12:14–25
- Mohammadi M, Mokhtari R, Schaback R (2014) Simulating the 2D Brusselator system in reproducing kernel Hilbert space. *Comput Model Eng Sci* 101:113–138
- Mosekilde E (1996) Topics in nonlinear dynamics: applications to physics, biology and economic systems. World Scientific, London
- Murray JD (1993) *Mathematical biology*. Springer, Heidelberg
- Nagumo J, Arimoto S, Yoshizawa S (1962) An active pulse transmission line simulating nerve axon. *Proc. IRE* 50:2061–2070
- Nomura T, Glass L (1996) Entrainment and termination of reentrant wave propagation in a periodically stimulated ring of excitable media. *Phys. Rev. E.* 53:6353–6360
- Peng Y, Zhang T (2016) Turing instability and pattern induced by cross-diffusion in a predator–prey system with Allee effect. *Appl. Math. Comput.* 275:1–12
- Piret C (2012) The orthogonal gradients method: a radial basis functions method for solving partial differential equations on arbitrary surfaces. *J. Comput. Phys.* 231(20):4662–4675
- Prigogine I, Lefever R (1968) Symmetry breaking instabilities in dissipative systems II. *J. Chem. Phys.* 48:1695–1700
- Quarteroni A, Sacco R, Saleri F (2007) *Numerical mathematics*. Springer, New York
- Roqoreanu C, Georgescu A, Giurgiteanu N (2000) *The FitzHugh–Nagumo model: Bifurcation and dynamics*. Springer, New York
- Ruuth SJ, Merriman B (2008) A simple embedding method for solving partial differential equations on surfaces. *J Comput Phys* 227:1943–1961

- Salehi R, Dehghan M (2013) A generalized moving least square reproducing kernel method. *J Comput Appl Math* 249:120–132
- Schnakenberg J (1979) Simple chemical reaction system with limit cycle behavior. *J Theor Biol* 81:389–400
- Sekimura T, Madzvamuse A, Wathen A, Maini P (2000) A model for colour pattern formation in the butterfly wing of *Papilio dardanus*. *Proc R Soc Lond Ser B* 26:852–859
- Selkov EE (1968) Self-oscillations in glycolysis. *Eur J Biochem* 4:79–86
- Shakeri F, Dehghan M (2011) The finite volume spectral element method to solve Turing models in the biological pattern formation. *Comput Math Appl* 62:4322–4336
- Shankar V, Wright GB, Fogelson AL, Kirby RM (2015) A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction–diffusion equations on surfaces. *J Sci Comput* 63:745–768
- Sheth R, Marcon L, Bastida MF, Junco M, Quintana L, Dahn R, Kmita M, Sharpe J, Ros M (2012) Hox genes regulate digit patterning by controlling the wavelength of a Turing-type mechanism. *Science* 338:1476–1480
- Smiely M (2009) An efficient implementation of a numerical method for a chemotaxis system. *Int J Comput Math* 86:219–235
- St Clair NM (2006) Pattern formation in partial differential equations, Thesis
- Sugai SS, Ode KL, Ueda HR (2017) A design principle for an autonomous post-translational pattern formation. *Cell Rep* 19:863–874
- Tatari M, Kamranian M, Dehghan M (2011) The finite point method for reaction–diffusion systems in developmental biology. *Comput Model Eng Sci CMES* 82:1–27
- Tonnellier A (2002) The McKean’s caricature of the Fitzhugh–Nagumo model I. The space-clamped system. *SIAM J Appl Math* 63:459–484
- Turing AM (1952) The chemical basis of morphogenesis. *Phil Trans R Soc Ser B* 237(641):37–72
- Twizell EH, Gumel AB, Cao Q (1999) A second-order scheme for the “Brusselator” reaction–diffusion system. *J Math Chem* 26:297–316
- Tyson R, Lubkin S, Murray J (1999) Model and analysis of chemotactic bacterial patterns in a liquid medium. *J Math Biol* 38:359–375
- Tyson R, Stern L, LeVeque R (2000) Fraction step methods applied to a chemotaxis model. *J Math Biol* 41:455–475
- Van der Pol B, Van der mark J (1928) The heart beat considered as a relaxation oscillation, and an electrical model of the heart. *Philos Mag* 6:763–775
- Varvruska T (2015) Turing patterns and butterfly wings. Thesis, Carroll university
- Wazwaz AM (2000) The decomposition method applied to systems of partial differential equations and to the reaction–diffusion Brusselator model. *Appl Math Comput* 110:251–264
- Wendland H (2005) *Scattered Data Approximation*. Cambridge University Press
- Xu B, Binczak S, Jacquir S, Pont O, Yahia H (Aug 2014) Parameters analysis of FitzHugh–Nagumo model for a reliable simulation. 36th Annual international conference of the IEEE engineering in Medicine and Biology Society (EMBC’14) Chicago. United States, IEEE, p 2014
- Zhu J, Zhang YT, Newman SA, Alber M (2009) Application of discontinuous Galerkin methods for reaction–diffusion systems in developmental biology. *J Sci Comput* 40:391–418