Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 13, No. 2, 2025, pp. 357-372
DOI:10.22034/cmde.2024.59100.2508

Interpolating MLPG method to investigate predator-prey population dynamic with complex characters

Mostafa Abbaszadeh[1,*], Alireza Bagheri Salec[2], and Afaq Salman Alwan[2]

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

[2]Department of Mathematics, Faculty of Basic Science, University of Qom, Alghadir Blvd., Qom, Iran.

Abstract

The predator-prey model is a pair of first-order nonlinear differential equations which are used to explain the dynamics of biological systems. These systems contain two species interacting, one as a predator and the other as prey. This work proposes a meshless local Petrov-Galerkin (MLPG) method based upon the interpolating moving least squares (IMLS) approximation, for the numerical solution of the predator-prey systems. With this aim, the space derivative is discretized by the MLPG technique in which the test and trial functions are chosen from the shape functions of IMLS approximation. Next, a semi-implicit finite difference approach is utilized to discretize the time derivative. The main aim of this work is to bring forward a flexible numerical procedure to solve predator-prey systems on complicated geometries.

Keywords. Predator-prey model, Meshless local Petrov-Galerkin (MLPG) method, Interpolating IMLS approximation.

2010 Mathematics Subject Classification. 65M12, 65N12, 65N30, 65N40.

1. Introduction

A system of the predator-prey model can be introduced by [38]

$$\begin{cases} u_t = \nabla \cdot (\boldsymbol{D}\nabla \boldsymbol{u}) + G(\boldsymbol{u}), & \boldsymbol{x} \in \Omega, \quad t > 0, \\ \nabla \boldsymbol{u} \cdot \overrightarrow{\boldsymbol{n}} = 0, & \boldsymbol{x} \in \partial\Omega, \quad t > 0, \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}), & \boldsymbol{x} \in \overline{\Omega}, \end{cases} \tag{1.1}$$

where G is nonlinear functions. Several numerical techniques were investigated to simulate model (1.1) such as finite difference method [14], finite volume method [32], finite element method [5, 18, 23], element-free Galerkin method [10, 11], Chebyshev spectral method [37], Legendre spectral element method [13], spectral meshless radial point interpolation approach [34], local radial basis function [30], Convolutional neural network [39], nonstandard FDM [8, 15], etc.

According to the assumption in [6, 20], we can write

$$\begin{cases} \dfrac{\partial \Phi}{\partial t} = d_1 \Delta \Phi + r\Phi\left(1 - \dfrac{\Phi}{k}\right) - \kappa\left(\dfrac{\Phi}{\Phi + a}\right) v, \\ \dfrac{\partial u}{\partial t} = d_2 \Delta u + I - ru - \phi\left(\dfrac{u}{u + \ell}\right) v, \\ \dfrac{\partial v}{\partial t} = d_3 \Delta v + \varepsilon\kappa\left(\dfrac{\Phi}{\Phi + a}\right) v + b\phi\left(\dfrac{u}{u + \ell}\right) v - \hbar v, \end{cases} \tag{1.2}$$

where

Received: 05 November 2023 ; Accepted: 09 February 2024.
∗ Corresponding author. Email: m.abbaszadeh@aut.ac.ir.

(1) Φ is the prey population,
(2) v denotes the predator population,
(3) u shows the quantity of subsidy,
(4) d_1, d_2, and d_3 are the positive diffusion coefficients.

Authors of [7] presented a formulation that achieves high levels of accuracy and efficiency by properly solving the Poisson equations at each step of the solution process by formulating a Localized RBF Collocation Meshless Method (LRC-MM) solution approach for the approximation of the diffusive and convective derivatives while employing the same framework to implement a Dual-Reciprocity Boundary Element Method (DR-BEM) for the solution of the ensuing Poisson equations. A method is developed in [29] for solving an inverse geometric problem is presented by reconstructing the unknown subsurface cavity geometry with the boundary element method (BEM) and a genetic algorithm in combination with the Nelder-Mead non-linear simplex optimization method. Authors of [2] described an extension of the boundary element method (BEM) and the dual reciprocity boundary element method (DRBEM) formulations developed for one- and two-dimensional steady-state problems, to analyze transient convection–diffusion problems associated with first-order chemical reaction. A numerical reduced order model framework is developed in [9] to simulate the physics of the thermo-mechanical processes that occur during c-Si photovoltaic (PV) cell fabrication. The relationship between prey and predator is analyzed in [24] from the early nineteenth century by considering different types of functional responses and ecological effects. The main aim of [19] is to present a modified ratio-dependent model by incorporating the supply of additional food to the predators. A predator–prey system with Allee effect and gestation delay is established in [22]. A meshless approximation based on GMLS is applied in [12] to solve the reaction-diffusion equations on the sphere and red-blood-cell surfaces. Authors of [16] developed a Sinc function interpolation collocation method to simulate a class of predator-prey systems with complex dynamics characters.

Now, we want to propose a MLPG method for the numerical solution of The main problem. The MLPG method is constructed by Atluri [3, 4, 36]. The MLPG method uses some regular, simple and independent sub-domains for the numerical integral. The spectral meshless radial point interpolation (SMRPI) technique is used in [35] to the solution of pattern formation in nonlinear reaction-diffusion systems. Authors of [28] proposed a numerical solution for the two-dimensional system of nonlinear partial differential equations by a global radial basis function collocation method (GRBFCM). The meshless local radial point interpolation (MLRPI) methods are employed in [33] to simulate two-dimensional wave equations subject to given appropriate initial and Neumann's boundary conditions. A meshless numerical technique is proposed in [17] for solving the generalized variable coefficient Schrödinger equation and Schrödinger-Boussinesq system with electromagnetic fields.

Integrating of MLS shape functions is needed in the MLPG technique which it cause of increasing the used CPU time. To treatment this issue, the direct MLPG (DMLPG) method based on the GMLS approximation is proposed [26, 27] nad it is used for solving various problems [1, 25, 31].

2. Interpolating moving least squares (IMLS) approximation

Let $\boldsymbol{\varpi} = \{\boldsymbol{\varpi}_i\}_{i=1}^N$ be a set of scattered data in $\Omega \subset \mathbb{R}^n$. The fill distance is

$$h_{X,\Omega} = \sup_{x\in\Omega} \min_{1\le j\le N} \|\varpi - \varpi_j\|_2, \qquad q_X = \frac{1}{2}\min_{i\neq j} \|\varpi_i - \varpi_j\|_2. \tag{2.1}$$

Also

$$\Im(\boldsymbol{\varpi}) \triangleq B(\boldsymbol{\varpi}, \delta) = \{\boldsymbol{\varpi}^* \in \mathbb{R}^n \ : \ \|\boldsymbol{\varpi} - \boldsymbol{\varpi}^*\| < \delta(\boldsymbol{\varpi})\}, \tag{2.2}$$

is the influence domain of node $\boldsymbol{\varpi}$ [21] and the influence domain of point $\boldsymbol{\varpi}_i$ is

$$\Im_i \triangleq \Im(\boldsymbol{\varpi}_i) = \{\boldsymbol{\varpi}^* \in \mathbb{R}^n \ : \ \|\boldsymbol{\varpi}_i - \boldsymbol{\varpi}^*\| < \delta_i\}, \tag{2.3}$$

in which δ_i is the radius of $\Im_i$. Also, the following weight function is employed [21]

$$w_i(\boldsymbol{\varpi}) = \begin{cases} \varphi\left(\frac{\|\boldsymbol{\varpi}-\boldsymbol{\varpi}_i\|_2}{\delta_i}\right)\left\|\frac{\boldsymbol{\varpi}-\boldsymbol{\varpi}_i}{\delta_i}\right\|_2^{-\alpha}, & x \in \Im_i, \\ 0, & x \notin \Im_i, \end{cases} \tag{2.4}$$

in which the function φ is nonnegative, compactly supported in the unit ball $B(0,1)$, k-th times continuously differentiable, and its derivatives up to order k are bounded. Moreover, the function φ can be chosen to be the constant one or any weight functions used in the MLS approximation. Also, in the simulation, we assume $\alpha = 2$. We set

$$\boldsymbol{p}(\boldsymbol{\varpi}) = \left[\begin{array}{cccc} p_0(\boldsymbol{\varpi}) & p_1(\boldsymbol{\varpi}) & \dots & p_{m-1}(\boldsymbol{\varpi}) \end{array}\right]^{\mathrm{T}}, \qquad \boldsymbol{\varpi} \in \Omega. \tag{2.5}$$

Consider

$$span\left\{p_0(\boldsymbol{\varpi}), p_1(\boldsymbol{\varpi}), \dots, p_m(\boldsymbol{\varpi})\right\},$$

and [21]

$$q_0(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) = \frac{p_0(\boldsymbol{\varpi})}{(p_0, p_0)_{\boldsymbol{\varpi}}^{\frac{1}{2}}} = \frac{1}{\left[\sum\limits_{i \in \mathcal{E}(\boldsymbol{\varpi})} w_i(\boldsymbol{\varpi})\right]^{\frac{1}{2}}}, \tag{2.6}$$

in which

$$(f, g)_{\boldsymbol{\varpi}} = \sum_{i \in \mathcal{E}(\boldsymbol{\varpi})} w_i(\boldsymbol{\varpi}) f(\boldsymbol{\varpi}_i) g(\boldsymbol{\varpi}_i). \tag{2.7}$$

Also, we set [21]

$$q_i(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) = p_i(\overline{\boldsymbol{\varpi}}) - \sum_{l \in \mathcal{E}(\boldsymbol{\varpi})} v_l(\boldsymbol{\varpi}) p_i(\boldsymbol{\varpi}_l), \tag{2.8}$$

in which

$$v_l(\boldsymbol{\varpi}) = \frac{w_l(\boldsymbol{\varpi})}{\sum\limits_{j \in \mathcal{E}(\boldsymbol{\varpi})} w_j(\boldsymbol{\varpi})}. \tag{2.9}$$

To approximate the unknown function $u(\boldsymbol{\varpi})$ at $\boldsymbol{\varpi}$, we put [21]

$$u_h(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) = \sum_{i=0}^{m} q_i(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) a_i(\boldsymbol{\varpi}) = q_0(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) a_0(\boldsymbol{\varpi}) + \boldsymbol{q}^{\mathrm{T}}(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) \boldsymbol{a}(\boldsymbol{\varpi}), \tag{2.10}$$

such that $\{a_i(\boldsymbol{\varpi})\}_{i=0}^{m}$ are the unknown coefficients. These unknown parameters will be driven by minimizing the following functional [21]

$$J(\boldsymbol{\varpi}) = \sum_{i \in \mathcal{E}(\boldsymbol{\varpi})} w_i(\boldsymbol{\varpi})[u(\boldsymbol{\varpi}_i) - u_h(\boldsymbol{\varpi}, \boldsymbol{\varpi}_i)]^2 = \sum_{i \in \mathcal{E}(\boldsymbol{\varpi})} w_i(\boldsymbol{\varpi}) \left[u(\boldsymbol{\varpi}_i) - \sum_{i=0}^{m} q_i(\boldsymbol{\varpi}, \boldsymbol{\varpi}_i) a_i(\boldsymbol{\varpi})\right]^2. \tag{2.11}$$

According to relation (2.7), Eq. (2.11) can be rewritten as follows

$$(u(\cdot) - u_h(\boldsymbol{\varpi}, \cdot), q_i(\boldsymbol{\varpi}, \cdot))_{\boldsymbol{\varpi}} = 0, \qquad 0 \le i \le m, \tag{2.12}$$

such that [21]

$$a_0(\boldsymbol{\varpi}) = (u - q_0(\boldsymbol{\varpi}, \cdot))_{\boldsymbol{\varpi}}, \tag{2.13}$$

$$\sum_{i=1}^{m} (q_i(\boldsymbol{\varpi}, \cdot), q_j(\boldsymbol{\varpi}, \cdot))_{\boldsymbol{\varpi}} a_i(\boldsymbol{\varpi}) = (u, q_j(\boldsymbol{\varpi}, \cdot))_{\boldsymbol{\varpi}}, \qquad j = 1, 2, \dots, m. \tag{2.14}$$

Thus, Eq. (2.14) can be changed as

$$\boldsymbol{A}(\boldsymbol{\varpi})\boldsymbol{a}(\boldsymbol{\varpi}) = \boldsymbol{B}(\boldsymbol{\varpi})\boldsymbol{u}, \tag{2.15}$$

where

$$\boldsymbol{u} = \left[\begin{array}{cccc} u(\boldsymbol{\varpi}_{l_1}) & u(\boldsymbol{\varpi}_{l_2}) & \dots & u(\boldsymbol{\varpi}_{l_{\eta(\boldsymbol{\varpi})}}) \end{array}\right]^{\mathrm{T}} \boldsymbol{A}(\boldsymbol{\varpi}) = \boldsymbol{B}(\boldsymbol{\varpi})\boldsymbol{Q}(\boldsymbol{\varpi}), \tag{2.16}$$

$$\boldsymbol{Q}(\boldsymbol{\varpi}) = \left[\begin{array}{cccc} \boldsymbol{q}(\boldsymbol{\varpi}, \boldsymbol{\varpi}_{l_1}) & \boldsymbol{q}(\boldsymbol{\varpi}, \boldsymbol{\varpi}_{l_2}) & \dots & \boldsymbol{q}(\boldsymbol{\varpi}, \boldsymbol{\varpi}_{l_{\eta(\boldsymbol{\varpi})}}) \end{array}\right], \tag{2.17}$$

and also [21]

$$\boldsymbol{B}_{ij}(\boldsymbol{\varpi}) = \begin{cases} w_{l_j}(\boldsymbol{\varpi}) q_i(\boldsymbol{\varpi}, \boldsymbol{\varpi}_{l_j}), & \boldsymbol{\varpi} \neq \boldsymbol{\varpi}_{l_j}, \\ \displaystyle\sum_{k \in \mathcal{E}(\boldsymbol{\varpi}), k \neq j} w_k(\boldsymbol{\varpi}) \left[p_i(\boldsymbol{\varpi}_{l_j}) - p_i(\boldsymbol{\varpi}_k)\right], & \boldsymbol{\varpi} = \boldsymbol{\varpi}_{l_j}. \end{cases} \tag{2.18}$$

Eq. (2.15) results

$$\boldsymbol{a}(\boldsymbol{\varpi}) = \boldsymbol{A}^{-1}(\boldsymbol{\varpi})\boldsymbol{B}(\boldsymbol{\varpi})\boldsymbol{u}. \tag{2.19}$$

Now, we have [21]

$$q_0(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) a_0(\boldsymbol{\varpi}) = q_0(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}})(u, q_0(\boldsymbol{\varpi}, \cdot))_{\boldsymbol{\varpi}} = \sum_{i \in \mathcal{E}(\boldsymbol{\varpi})} v_i(\boldsymbol{\varpi}) u(\boldsymbol{\varpi}_i) = \boldsymbol{\beta}^{\mathrm{T}}(\boldsymbol{\varpi})\boldsymbol{u}, \tag{2.20}$$

in which

$$\boldsymbol{\beta}(\boldsymbol{\varpi}) = \left[\begin{array}{cccc} v_{l_1}(\boldsymbol{\varpi}) & v_{l_2}(\boldsymbol{\varpi}) & \dots & v_{l_{\eta(\boldsymbol{\varpi})}}(\boldsymbol{\varpi}) \end{array}\right]^{\mathrm{T}}. \tag{2.21}$$

Putting Eqs. (2.19) and (2.20) into Eq. (2.10) yields

$$u_h(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) = \boldsymbol{\beta}^{\mathrm{T}}(\boldsymbol{\varpi}) u + \boldsymbol{q}^{\mathrm{T}}(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) \boldsymbol{A}^{-1}(\boldsymbol{\varpi}) \boldsymbol{B}(\boldsymbol{\varpi}) \boldsymbol{u}. \tag{2.22}$$

Thus, we have [21]

$$u(\boldsymbol{\varpi}) \approx u_h(\boldsymbol{\varpi}) = u_h(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}})|_{\overline{\boldsymbol{\varpi}} = \boldsymbol{\varpi}} = \left[\boldsymbol{\beta}^{\mathrm{T}}(\boldsymbol{\varpi}) + \boldsymbol{q}^{\mathrm{T}}(\boldsymbol{\varpi}, \overline{\boldsymbol{\varpi}}) \boldsymbol{A}^{-1}(\boldsymbol{\varpi}) \boldsymbol{B}(\boldsymbol{\varpi})\right] \boldsymbol{u}, \tag{2.23}$$

where [21]

$$\phi_i(\boldsymbol{\varpi}) = \begin{cases} v_i(\boldsymbol{\varpi}) + \displaystyle\sum_{j=1}^{m} q_j(\boldsymbol{\varpi}, \boldsymbol{\varpi}) \left[\boldsymbol{A}^{-1}(\boldsymbol{\varpi}) \boldsymbol{B}(\boldsymbol{\varpi})\right]_{jk}, & i = I_k \in \mathcal{E}(\boldsymbol{\varpi}), \\ 0, \quad \notin \mathcal{E}(\boldsymbol{\varpi}), \end{cases} \tag{2.24}$$

3. Numerical formulation for predator-prey population dynamic

In the current section, a full-discrete scheme will be obtained for the following mathematical model

$$\begin{cases} \dfrac{\partial \Phi}{\partial t} = d_1 \Delta \Phi + r\Phi\left(1 - \dfrac{\Phi}{k}\right) - \kappa\left(\dfrac{\Phi}{\Phi + a}\right) v, \\ \dfrac{\partial u}{\partial t} = d_2 \Delta u + I - ru - \phi\left(\dfrac{u}{u+\ell}\right) v, \qquad (x_1, x_2) \in \Omega, \\ \dfrac{\partial v}{\partial t} = d_3 \Delta v + \varepsilon\kappa\left(\dfrac{\Phi}{\Phi + a}\right) v + b\phi\left(\dfrac{u}{u+\ell}\right) v - \hbar v. \\ \Phi(x_1, x_2, 0) = \Phi_0(x_1, x_2), \quad u(x_1, x_2, 0) = u_0(x_1, x_2), \quad v(x_1, x_2, 0) = v_0(x_1, x_2), \quad (x_1, x_2) \in \overline{\Omega}, \\ \Phi(x_1, x_2, t) = \Phi_{1,0}(x_1, x_2, t), \quad u(x_1, x_2, t) = u_{1,0}(x_1, x_2, t), \quad v(x_1, x_2, t) = v_{1,0}(x_1, x_2, t), \quad (x_1, x_2) \in \Gamma_D, \\ \dfrac{\partial \Phi}{\partial x_i} n_j = q_{0,\Phi}(x_1, x_2, t), \quad \dfrac{\partial u}{\partial x_i} n_j = q_{0,u}(x_1, x_2, t), \quad \dfrac{\partial v}{\partial x_i} n_j = q_{0,v}(x_1, x_2, t), \quad (x_1, x_2) \in \Gamma_{Nu}. \end{cases} \tag{3.1}$$

In the MLPG approach, a local weak form is needed. Thus, for each node (x_1, x_2) a sub-domain $\Omega_i^s \subset \overline{\Omega}$ is selected as integration domain. The local weak form of Eq. (3.1) for every interior point $(x_1, x_2) \in \Omega_i^s$ is

$$\int_{\Omega_i^s} \frac{\partial \Phi}{\partial t} \xi_1 d\Omega - d_1 \int_{\Gamma_i} \xi_1 \nabla \Phi \cdot n d\Gamma + d_1 \int_{\Omega_i^s} \nabla \Phi \cdot \nabla \xi_1 d\Omega = r \int_{\Omega_i^s} \Phi\left(1 - \frac{\Phi}{k}\right) \xi_1 d\Omega + \kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v \xi_1 d\Omega, \tag{3.2}$$

$$\int_{\Omega_i^s} \frac{\partial u}{\partial t} \xi_2 d\Omega - d_2 \int_{\Gamma_i} \xi_2 \nabla u \cdot n d\Gamma + d_2 \int_{\Omega_i^s} \nabla u \cdot \nabla \xi_2 d\Omega = I \int_{\Omega_i^s} \xi_2 d\Omega - r \int_{\Omega_i^s} u \xi_2 d\Omega - \phi \int_{\Omega_i^s} \left(\frac{u}{u+\ell}\right) v \xi_2 d\Omega, \tag{3.3}$$

$$\int_{\Omega_i^s} \frac{\partial v}{\partial t} \xi_3 d\Omega - d_3 \int_{\Gamma_i} \xi_3 \nabla v \cdot n d\Gamma + d_3 \int_{\Omega_i^s} \nabla v \cdot \nabla \xi_3 d\Omega = \varepsilon\kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v \xi_3 d\Omega + b\phi \int_{\Omega_i^s} \left(\frac{u}{u+\ell}\right) v \xi_3 d\Omega - \hbar \int_{\Omega_i^s} v \xi_3 d\Omega, \tag{3.4}$$

whre ξ_i for $i = 1, 2, 3$ are test functions. Now, each $\partial\Omega_i^s$ can be divided to $L_i \cup \Gamma_i$. This point is depicted in Figure

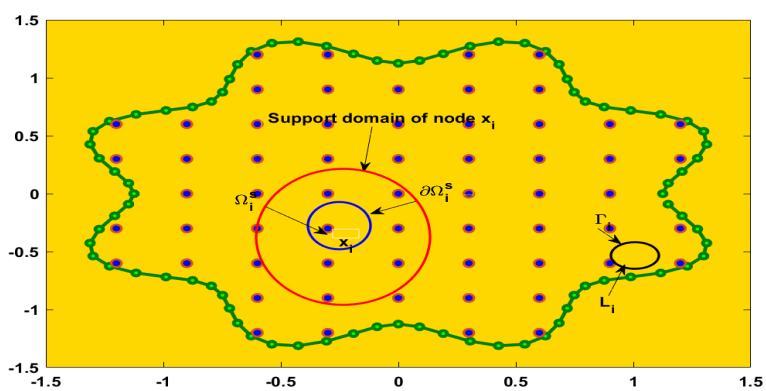


Figure 1. Subdomain for very interior and boundary nodes: Source. Authors work

1. Also, $\Gamma_i = \Omega_i^s \cap \partial\Omega$ and $L_i = \partial\Omega_i^s - \Gamma_i$. Now, the local weak forms for boundary nodes are

$$\begin{aligned}
\int_{\Omega_i^s} \frac{\partial \Phi}{\partial t}\xi_1 d\Omega &- d_1 \int_{\Gamma_i} \xi_1 \nabla\Phi \cdot n d\Gamma - d_1 \int_{L_i} \xi_1 \nabla\Phi \cdot n d\Gamma + d_1 \int_{\Omega_i^s} \nabla\Phi \cdot \nabla\xi_1 d\Omega \\
&= r \int_{\Omega_i^s} \Phi\left(1 - \frac{\Phi}{k}\right)\xi_1 d\Omega + \kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v\xi_1 d\Omega, \qquad (3.5)\\
\int_{\Omega_i^s} \frac{\partial u}{\partial t}\xi_2 d\Omega &- d_2 \int_{\Gamma_i} \xi_2 \nabla u \cdot n d\Gamma - d_2 \int_{L_i} \xi_2 \nabla u \cdot n d\Gamma + d_2 \int_{\Omega_i^s} \nabla u \cdot \nabla\xi_1 d\Omega \\
&= I \int_{\Omega_i^s} \xi_2 d\Omega - r \int_{\Omega_i^s} u\xi_2 d\Omega - \phi \int_{\Omega_i^s} \left(\frac{u}{u + \ell}\right) v\xi_2 d\Omega, \qquad (3.6)\\
\int_{\Omega_i^s} \frac{\partial v}{\partial t}\xi_3 d\Omega &- d_3 \int_{\Gamma_i} \xi_3 \nabla v \cdot n d\Gamma - d_3 \int_{L_i} \xi_3 \nabla v \cdot n d\Gamma + d_3 \int_{\Omega_i^s} \nabla v \cdot \nabla\xi_3 d\Omega \\
&= \varepsilon\kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v\xi_3 d\Omega + b\phi \int_{\Omega_i^s} \left(\frac{u}{u + \ell}\right) v\xi_3 d\Omega - \hbar \int_{\Omega_i^s} v\xi_3 d\Omega. \qquad (3.7)
\end{aligned}$$

The Neumann boundary conditions can be applied in Eqs. (3.5), (3.6), and (3.7), and the direct approach can be used to impose the Dirichlet boundary conditions. Now, we consider the following weight function

$$\xi_1(x,y) = \xi_2(x,y) = \xi_3(x,y) = \begin{cases} 1, & (x,y) \in \overline{\Omega}_i^s, \\ 0, & (x,y) \notin \overline{\Omega}_i^s, \end{cases}$$

then Eqs. (3.2)-(3.4) and (3.5)-(3.7) will be changed as

$$\int_{\Omega_i^s} \frac{\partial \Phi}{\partial t} d\Omega - d_1 \int_{\Gamma_i} \nabla\Phi \cdot n d\Gamma = r \int_{\Omega_i^s} \Phi\left(1 - \frac{\Phi}{k}\right) d\Omega + \kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v d\Omega, \qquad (3.8)$$

$$\int_{\Omega_i^s} \frac{\partial \Phi}{\partial t} d\Omega - d_1 \int_{\Gamma_i} \nabla\Phi \cdot n d\Gamma - d_1 \int_{L_i} \nabla\Phi \cdot n d\Gamma = r \int_{\Omega_i^s} \Phi\left(1 - \frac{\Phi}{k}\right) d\Omega + \kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v d\Omega, \qquad (3.9)$$

$$\int_{\Omega_i^s} \frac{\partial u}{\partial t} d\Omega - d_2 \int_{\Gamma_i} \nabla u \cdot n d\Gamma + d_2 \int_{\Omega_i^s} \nabla u \cdot \nabla d\Omega = I \int_{\Omega_i^s} d\Omega - r \int_{\Omega_i^s} u d\Omega - \phi \int_{\Omega_i^s} \left(\frac{u}{u + \ell}\right) v d\Omega, \qquad (3.10)$$

$$\int_{\Omega_i^s} \frac{\partial u}{\partial t} d\Omega - d_2 \int_{\Gamma_i} \nabla u \cdot n d\Gamma - d_2 \int_{L_i} \nabla u \cdot n d\Gamma = I \int_{\Omega_i^s} d\Omega - r \int_{\Omega_i^s} u d\Omega - \phi \int_{\Omega_i^s} \left(\frac{u}{u + \ell}\right) v d\Omega, \qquad (3.11)$$

$$\int_{\Omega_i^s} \frac{\partial v}{\partial t} d\Omega - d_3 \int_{\Gamma_i} \nabla v \cdot n d\Gamma = \varepsilon\kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v d\Omega + b\phi \int_{\Omega_i^s} \left(\frac{u}{u + \ell}\right) v d\Omega - \hbar \int_{\Omega_i^s} v d\Omega, \qquad (3.12)$$

$$\int_{\Omega_i^s} \frac{\partial v}{\partial t} d\Omega - d_3 \int_{\Gamma_i} \nabla v \cdot n d\Gamma - d_3 \int_{L_i} \nabla v \cdot n d\Gamma = \varepsilon\kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v d\Omega + b\phi \int_{\Omega_i^s} \left(\frac{u}{u + \ell}\right) v d\Omega - \hbar \int_{\Omega_i^s} v d\Omega. \qquad (3.13)$$

At the current moment, let $\tau = \dfrac{T}{N}$ and $t_n = k\tau$ for $k = 0, 1, 2, \ldots, N$, Then, we have

$$\int_{\Omega_i^s} \Phi^n d\Omega - \tau d_1 \int_{\Gamma_i} \nabla \Phi^n \cdot n d\Gamma = \int_{\Omega_i^s} \Phi^{n-1} d\Omega + \tau r \int_{\Omega_i^s} \Phi^n \left(1 - \frac{\Phi^n}{k}\right) d\Omega + \tau\kappa \int_{\Omega_i^s} \left(\frac{\Phi^n}{\Phi^n + a}\right) v^n d\Omega,$$

$$\int_{\Omega_i^s} \Phi^n d\Omega - \tau d_1 \int_{\Gamma_i} \nabla \Phi \cdot n d\Gamma - \tau d_1 \int_{L_i} \nabla \Phi \cdot n d\Gamma = \int_{\Omega_i^s} \Phi^{n-1} d\Omega + \tau r \int_{\Omega_i^s} \Phi \left(1 - \frac{\Phi}{k}\right) d\Omega + \tau\kappa \int_{\Omega_i^s} \left(\frac{\Phi}{\Phi + a}\right) v d\Omega,$$

$$\int_{\Omega_i^s} u^n d\Omega - d_2 \int_{\Gamma_i} \nabla u^n \cdot n d\Gamma = \int_{\Omega_i^s} u^{n-1} d\Omega + I \int_{\Omega_i^s} d\Omega - r \int_{\Omega_i^s} u^n d\Omega - \phi \int_{\Omega_i^s} \left(\frac{u^n}{u^n + \ell}\right) v^n d\Omega,$$

$$\int_{\Omega_i^s} u^n d\Omega - d_2 \int_{\Gamma_i} \nabla u^n \cdot n d\Gamma - d_2 \int_{L_i} \nabla u^n \cdot n d\Gamma = \int_{\Omega_i^s} u^{n-1} d\Omega + I \int_{\Omega_i^s} d\Omega - r \int_{\Omega_i^s} u^n d\Omega - \phi \int_{\Omega_i^s} \left(\frac{u^n}{u^n + \ell}\right) v^n d\Omega,$$

$$\int_{\Omega_i^s} v^n d\Omega - d_3 \int_{\Gamma_i} \nabla v^n \cdot n d\Gamma = \int_{\Omega_i^s} v^{n-1} d\Omega + \varepsilon\kappa \int_{\Omega_i^s} \left(\frac{\Phi^n}{\Phi^n + a}\right) v^n d\Omega + b\phi \int_{\Omega_i^s} \left(\frac{u^n}{u^n + \ell}\right) v^n d\Omega - \hbar \int_{\Omega_i^s} v^n d\Omega,$$

$$\begin{aligned}\int_{\Omega_i^s} v^n d\Omega - d_3 \int_{\Gamma_i} \nabla v^n \cdot n d\Gamma - d_3 \int_{L_i} \nabla v^n \cdot n d\Gamma &= \int_{\Omega_i^s} v^{n-1} d\Omega + \varepsilon\kappa \int_{\Omega_i^s} \left(\frac{\Phi^n}{\Phi^n + a}\right) v^n d\Omega \\ &\quad + b\phi \int_{\Omega_i^s} \left(\frac{u^n}{u^n + \ell}\right) v^n d\Omega - \hbar \int_{\Omega_i^s} v^n d\Omega.\end{aligned}$$

For interior and boundary nodes, let the approximate solution be

$$\Phi^n(x_1, x_2) = \sum_{i=1}^{M} \Phi_i^n \phi_i(x_1, x_2), \tag{3.14}$$

$$u^n(x_1, x_2) = \sum_{i=1}^{M} u_i^n \phi_i(x_1, x_2), \tag{3.15}$$

$$v^n(x_1, x_2) = \sum_{i=1}^{M} v_i^n \phi_i(x_1, x_2), \tag{3.16}$$

where ϕ_i is the shape functions of IMLS approximation and Φ_i^n, u_i^n and v_i^n are unknown coefficients. Substituting the approximate solutions (3.14), (3.15), and (3.16) into the local weak forms, gives the following nonlinear algebraic system of equations

$$\boldsymbol{F}(\boldsymbol{\Phi}^n, \boldsymbol{u}^n, \boldsymbol{v}^n,) = 0. \tag{3.17}$$

The Broyden's method will be used to solve Eq. (3.17).

4. Numerical Strategy

The simulations are presented via MATLAB 2022b software on an Intel Core i7 machine with 64 GB of memory. Let

$$\mathcal{E}_{\Phi,\infty}^N = \left\|\Phi^e - \Phi^N\right\|_\infty,$$
$$\mathcal{E}_{u,\infty}^N = \left\|u^e - u^N\right\|_\infty,$$
$$\mathcal{E}_{v,\infty}^N = \left\|v^e - v^N\right\|_\infty,$$

TABLE 1. Errors obtained and the used CPU time(s) for Experiment 1.

N	$\mathcal{E}^N_{\Phi,\infty}$	$\mathcal{E}^N_{u,\infty}$	$\mathcal{E}^N_{v,\infty}$	CPU time
400	5.6253×10^{-3}	3.9776×10^{-2}	3.9579×10^{-3}	2.2
600	2.7118×10^{-3}	1.9175×10^{-2}	1.9080×10^{-3}	10.3
800	1.3267×10^{-3}	9.3811×10^{-3}	9.3346×10^{-4}	31.5
1000	6.5571×10^{-4}	4.6366×10^{-3}	4.6136×10^{-4}	87.4
1200	3.2593×10^{-4}	2.3047×10^{-3}	2.2932×10^{-4}	187.3
1600	1.6248×10^{-4}	1.1489×10^{-3}	1.1432×10^{-4}	305.1
2000	6.4877×10^{-5}	4.5875×10^{-4}	4.5647×10^{-5}	1869.1

where Φ^e, u^e and v^e are there vectors that they contain the numerical solutions at $\tau = 10^{-3}$ and $N = 4000$ distributed nodes and Φ^N, u^N and v^N are there vectors including the numerical solution with time step τ. The obtained solution with $\tau = 10^{-3}$ and $N = 4000$ distributed nodes using the present methods is named Φ^e, u^e and v^e as the reference solution.

4.1. **Experiment 1.** For the first problem, we study the following system [16]

$$\begin{cases} \dfrac{\partial \Phi}{\partial t} = r\Phi\left(1 - \dfrac{\Phi}{k}\right) - \kappa\left(\dfrac{\Phi}{\Phi + a}\right)v, \\[2ex] \dfrac{\partial u}{\partial t} = I - ru - \phi\left(\dfrac{u}{u+\ell}\right)v, \\[2ex] \dfrac{\partial v}{\partial t} = \varepsilon\kappa\left(\dfrac{\Phi}{\Phi + a}\right)v + b\phi\left(\dfrac{u}{u+\ell}\right)v - \hbar v, \end{cases} \tag{4.1}$$

where

d_1	d_2	d_3	r	k	ϕ	ε	b	$\hbar$	ℓ	a	κ	I
0.001	0.001	0.001	1	5	5	0.1	0.1	0.1	1	1	0.4	0.3

Numerical outputs and pattern formation of Experiment 1 with the following initial condition

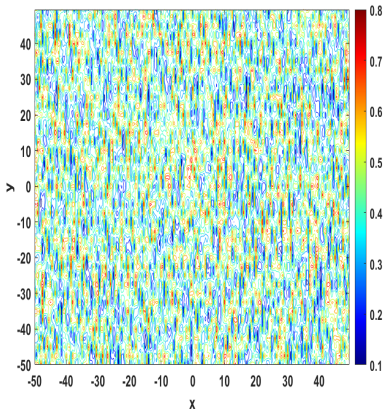
$$\Phi(x, y, 0) = 0.6\text{sech}\left(\frac{x}{0.2} + y\right) rand(N), \tag{4.2}$$

and $v(x, y, 0) = 1$ at the different T are reported.

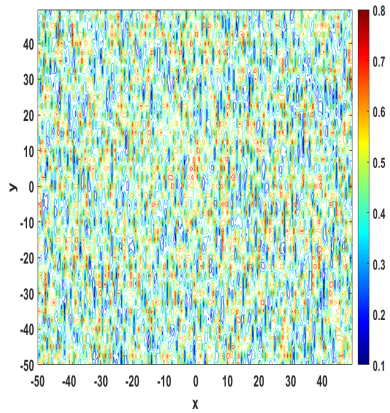
Figure 2 illustrates the pattern formation with $N = 4000$ scattered nodes in the physical domain, $\tau = 10^{-4}$, initial condition (4.2) and different final time T for Experiments 1. Since the random initial condition is used for Figure 2, thus, `MATLAB` command "**rand**" is applied. From Figure 2 it is obvious that with the use of the random initial condition, the pattern formation will be steady state. Table 1 reports the errors obtained and the used CPU time(s) for Experiment 1 based on the reference solution approach.

4.2. **Experiment 2.** Here, let the following model [16]

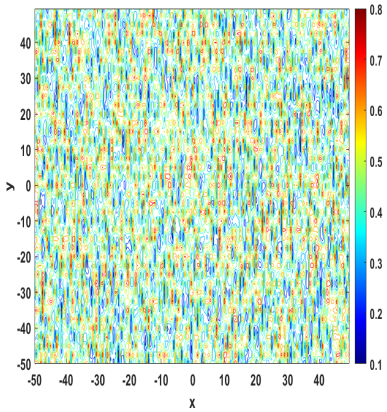
$$\begin{cases} \dfrac{\partial \Phi}{\partial t} = r\Phi\left(1 - \dfrac{\Phi}{k}\right) - \kappa\left(\dfrac{\Phi}{\Phi + a}\right)v, \\[2ex] \dfrac{\partial u}{\partial t} = I - ru - \phi\left(\dfrac{u}{u+\ell}\right)v, \\[2ex] \dfrac{\partial v}{\partial t} = \varepsilon\kappa\left(\dfrac{\Phi}{\Phi + a}\right)v + b\phi\left(\dfrac{u}{u+\ell}\right)v - \hbar v, \end{cases} \tag{4.3}$$



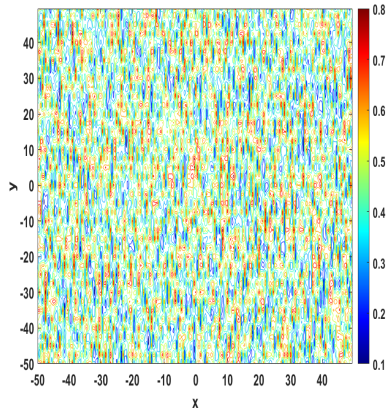
$T = 0.1.$



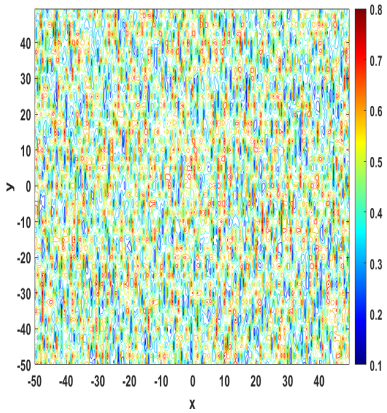
$T = 0.2.$



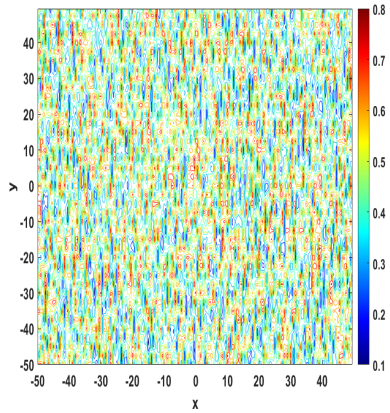
$T = 0.3.$



$T = 0.4.$

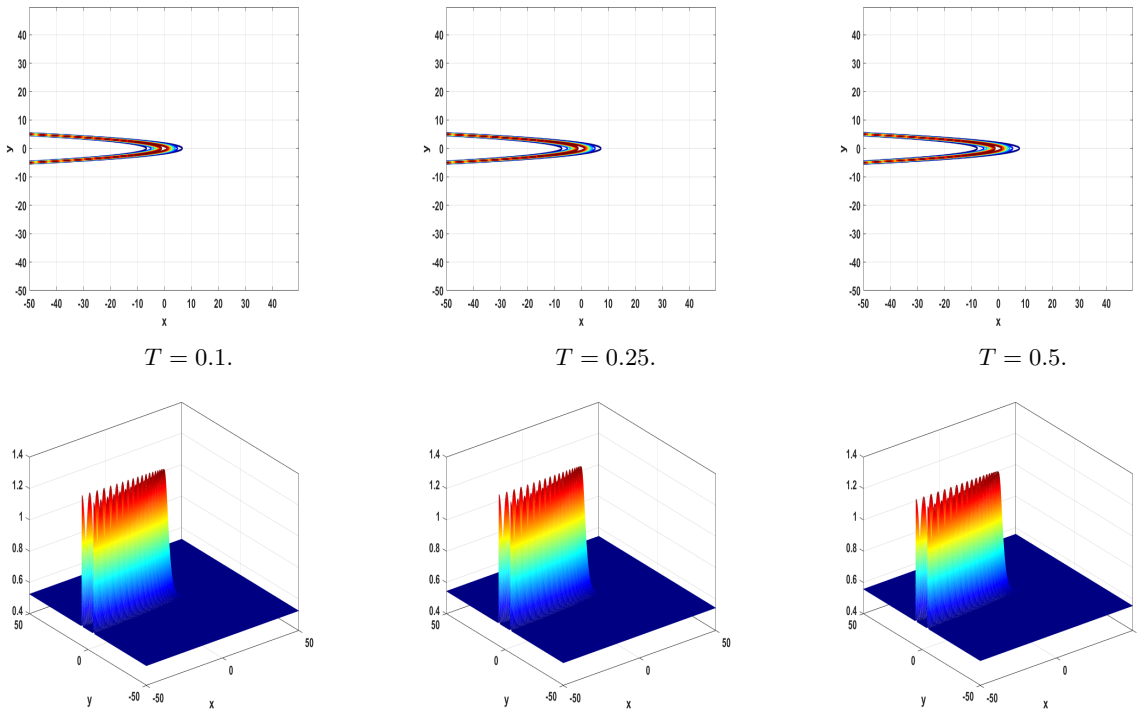


$T = 0.5.$



$T = 0.6.$

FIGURE 2. Pattern formation of approximate solution for Experiment 1.



$T = 0.1.$ $T = 0.25.$ $T = 0.5.$

$T = 0.1.$ $T = 0.25.$ $T = 0.6.$

FIGURE 3. Pattern formation of approximate solution for Experiment 2.

with homogeneous Neumann boundary conditions. Numerical outputs and pattern formation of Experiment 1 with the following data

$$\Phi(x, y, 0) = \sin\left(\operatorname{sech}\left(\frac{x}{2} + y^2\right)\right) + 0.5, \tag{4.4}$$

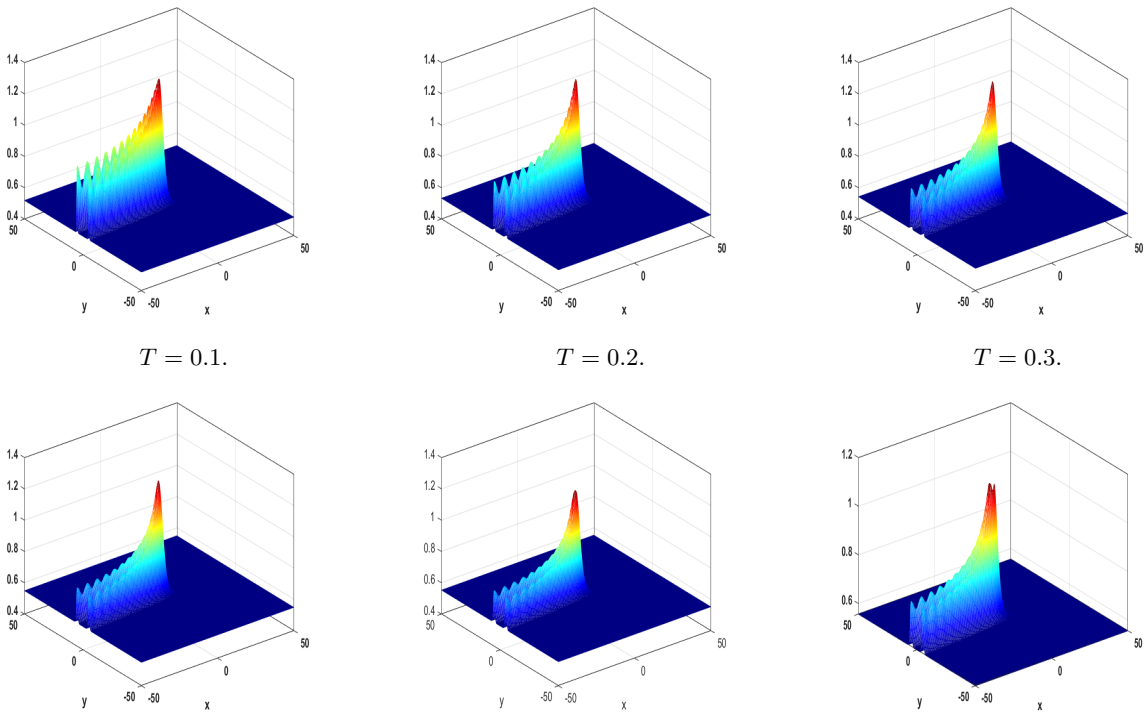
The used parameters for Figure 3 are as follows

d_1	d_2	d_3	r	k	ϕ	ε	b	$\hbar$	ℓ	a	κ	I
0.001	0.001	0.001	1	5	5	0.1	0.1	0.1	1	1	0.4	0.3

The pattern formations with $N = 6000$ collocation points in the computational domain, $\tau = 10^{-4}$, initial condition (4.4) and different final time T for Experiments 2 are shown in Figure 3. The initial condition (4.4) is a non-smooth function, however, from Figure 3, the initial condition (4.4) tends to the steady state response. On the hand, for Figures 4 and 5 the following parameters are considered

d_1	d_2	d_3	r	k	ϕ	ε	b	$\hbar$	ℓ	a	κ	I
0.1	0.1	0.1	1	5	5	0.001	0.1	0.01	1	1	0.4	0.3

The pattern formations with $N = 6000$ collocation nodes, $\tau = 10^{-4}$, initial data (4.4) and different final time T for Experiments 2 are displayed in Figure 4 and 5. Table 2 introduces the errors obtained and the used CPU time(s) for Experiment 2 based on the reference solution approach.



$T = 0.1.$ $T = 0.2.$ $T = 0.3.$

$T = 0.4.$ $T = 0.5.$ $T = 0.6.$

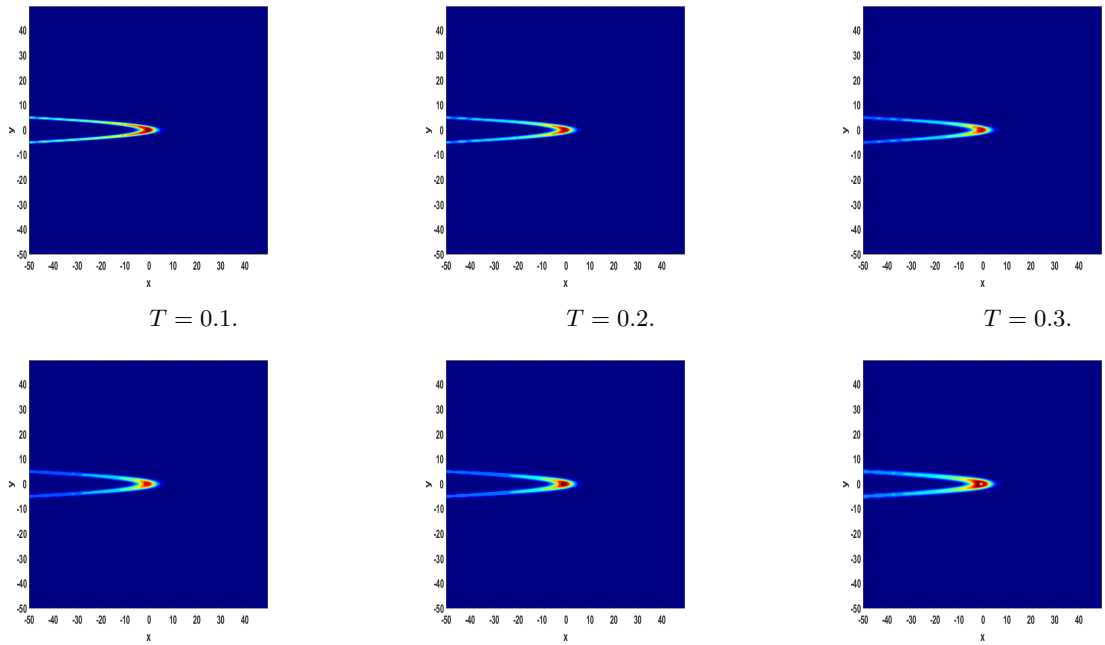
FIGURE 4. Pattern formation of approximate solution for Experiment 2.

TABLE 2. Errors obtained and the used CPU time(s) for Experiment 2.

N	$\mathcal{E}^N_{\Phi,\infty}$	$\mathcal{E}^N_{u,\infty}$	$\mathcal{E}^N_{v,\infty}$	CPU time
400	2.1512×10^{-3}	1.5211×10^{-2}	1.5136×10^{-3}	2.2
600	6.9998×10^{-4}	4.9496×10^{-3}	4.9250×10^{-4}	10.3
800	5.2334×10^{-4}	3.7006×10^{-3}	3.6822×10^{-4}	31.5
1000	4.1788×10^{-4}	2.9549×10^{-3}	2.9402×10^{-4}	87.4
1200	3.4779×10^{-4}	2.4593×10^{-3}	2.4471×10^{-4}	187.3
1600	2.9784×10^{-4}	2.1060×10^{-3}	2.0956×10^{-4}	305.1
2000	2.6043×10^{-4}	1.8415×10^{-3}	1.8324×10^{-4}	1869.1

4.3. **Experiment 3.** For the last problem, we investigate the following model [16]

$$\begin{cases} \dfrac{\partial \Phi}{\partial t} = r\Phi\left(1 - \dfrac{\Phi}{k}\right) - \kappa\left(\dfrac{\Phi}{\Phi + a}\right)v, \\[2ex] \dfrac{\partial u}{\partial t} = I - ru - \phi\left(\dfrac{u}{u+\ell}\right)v, \\[2ex] \dfrac{\partial v}{\partial t} = \varepsilon\kappa\left(\dfrac{\Phi}{\Phi + a}\right)v + b\phi\left(\dfrac{u}{u+\ell}\right)v - \hbar v, \end{cases} \tag{4.5}$$



$T = 0.1.$ $T = 0.2.$ $T = 0.3.$

$T = 0.4.$ $T = 0.5.$ $T = 0.6.$

FIGURE 5. Pattern formation of approximate solution for Experiment 2.

where

d_1	d_2	d_3	r	k	ϕ	ε	b	$\hbar$	ℓ	a	κ	I
0.0001	0.0001	0.0001	1	5	5	0.01	0.01	0.1	1	1	0.4	0.3

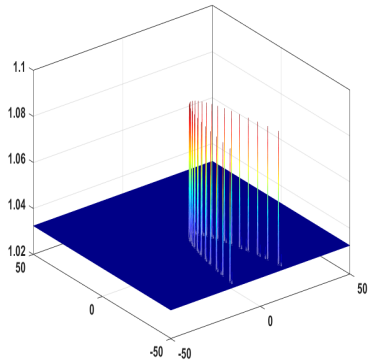
In the current example, we used the following initial condition

$$\Phi(x, y, 0) = \texttt{ones}(N) + \operatorname{sech}\left(\frac{x^2}{0.02} - 9 + \frac{2y}{0.01}\right). \tag{4.6}$$

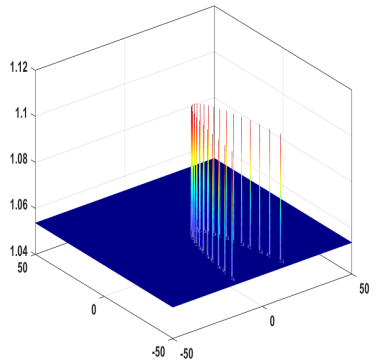
The surface of the pattern formations with $N = 6000$ collocation points in the computational domain, $\tau = 10^{-4}$, initial condition (4.6) and different final time T for Experiments 3 are depicted in Figure 6. On the other hand, the contour of the pattern formations with $N = 6000$ collocation points in the computational domain, $\tau = 10^{-4}$, initial condition (4.6) and different final time T for Experiments 3 are demonstrated in Figure 7. Table 2 proposes the errors obtained and the used CPU time(s) for Experiment 2 based on the reference solution approach.

5. Conclusion

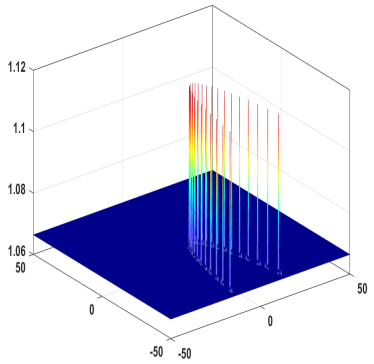
The application of the predator-prey system can be found in mathematical biology. Since this model does not have any exact solution, then the numerical solution will be important. In this study, a meshless local Petrov-Galerkin (MLPG) the method is utilized. Here, the test and trial sets contain the shape functions of IMLS approximation. These shape functions have the δ-Kronecker property. According to this advantage, the Dirichlet boundary conditions can be applied, directly. The spatial direction is approximated by the MLPG approach. Furthermore, the temporal direction is discretized by a finite difference formula which produces a nonlinear algebraic system of equations. The constructed equation is solved by Broyden's method.



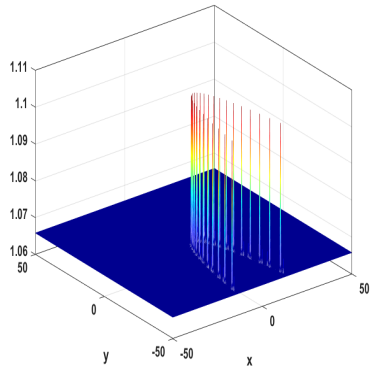
$T = 0.1.$



$T = 0.25.$



$T = 0.5.$

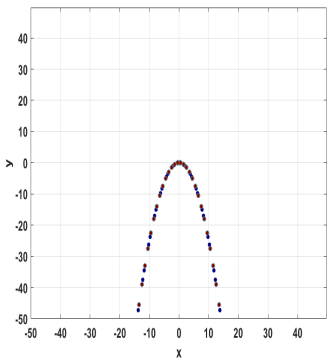


$T = 0.6.$

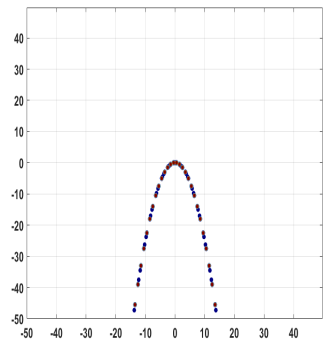
Figure 6. Pattern formation of approximate solution for Experiment 3.

Author contributions

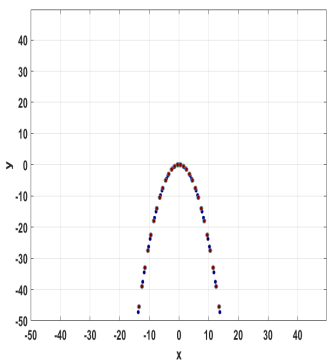
All authors contributed to writing, review and editing.



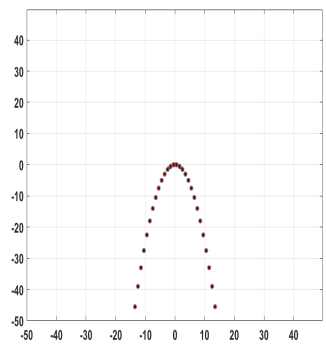
$T = 0.1.$



$T = 0.25.$



$T = 0.5.$



$T = 0.6.$

FIGURE 7. Pattern formation of approximate solution for Experiment 3.

References

[1] M. Abbaszadeh and M. Dehghan, *Direct meshless local Petrov-Galerkin (DMLPG) method for time-fractional fourth-order reaction–diffusion problem on complex domains*, Computers & Mathematics with Applications, *79*(3) (2020) 876–888.

[2] S. A. Al-Bayati and L. C. Wrobel, *Numerical modelling of convection-diffusion problems with first-order chemical reaction using the dual reciprocity boundary element method*, International Journal of Numerical Methods for Heat & Fluid Flow.

[3] S. N. Atluri, *The Meshless Method (MLPG) for Domain and BIE Discretizations*, Forsyth, *GA* (2004).

[4] S. N. Atluri and T. Zhu, *A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics*, Computational Mechanics, *2*(2) (1998), 117–127.

[5] R. Barreira, C. M. Elliott, and A. Madzvamuse, *The surface finite element method for pattern formation on evolving biological surfaces*, Journal of Mathematical Biology, *63*(6) (2011), 1095–1119.

[6] A. Bassett, A. L. Krause, and R. A. Van Gorder, *Continuous dispersal in a model of predator–prey-subsidy population dynamics*, Ecological Modelling, *354* (2017), 115–122.

[7] L. A. Bueno, E. A. Divo, and A. J. Kassab, *A coupled localized rbf meshless/drbem formulation for accurate modeling of incompressible fluid flows*, International Journal of Computational Methods and Experimental Measurements *5*(3) (2017) 359–368.

[8] Q. A. Dang and M. T. Hoang, *Nonstandard finite difference schemes for a general predator–prey system*, Journal of Computational Science, *36* (2019), 101015.

[9] A. Das, A. Khoury, E. Divo, V. Huayamave, A. Ceballos, R. Eaglin, A. Kassab, A. Payne, V. Yelundur, and H. Seigneur, *Real-time thermomechanical modeling of pv cell fabrication via a pod-trained rbf interpolation network*, Computer Modeling in Engineering & Sciences, *122*(3) (2020), 757–777.

[10] M. Dehghan and M. Abbaszadeh, *Numerical study of three-dimensional Turing patterns using a meshless method based on moving Kriging element free Galerkin (EFG) approach*, Computers & Mathematics with Applications, *72*(3) (2016), 427–454.

[11] M. Dehghan, M. Abbaszadeh, and A. Mohebbi, *The use of element free galerkin method based on moving Kriging and radial point interpolation techniques for solving some types of Turing models*, Engineering Analysis with Boundary Elements, *62* (2016), 93–111.

[12] M. Dehghan and N. Narimani, *Approximation of continuous surface differential operators with the generalized moving least-squares (gmls) method for solving reaction–diffusion equation*, Computational and Applied Mathematics, *37* (2018), 6955–6971.

[13] M. Dehghan and M. Sabouri, *A Legendre spectral element method on a large spatial domain to solve the predator–prey system modeling interacting populations*, Applied Mathematical Modelling, *37*(3) (2013), 1028–1038.

[14] R. Dillon, P. Maini, and H. Othmer, *Pattern formation in generalized Turing systems*, Journal of Mathematical Biology, *32*(4) (1994), 345–393.

[15] D. T. Dimitrov and H. V. Kojouharov, *Nonstandard finite-difference methods for predator–prey models with general functional response*, Mathematics and Computers in Simulation, *78*(1) (2008) 1–11.

[16] M. Du, P. Ning, and Y. Wang, *Numerical solution of a class of predator-prey systems with complex dynamics characters based on a sinc function interpolation collocation method*, Complexity, *2020* (2020) 1–34.

[17] G. Karamali, M. Abbaszadeh, and M. Dehghan, *The smoothed particle hydrodynamics method for solving generalized variable coefficient schrodinger equation and schrodinger-boussinesq system*, Computational Methods for Differential Equations, *6*(2) (2018), 215–237.

[18] L. J. Khaled-Abad and R. Salehi, *Numerical and theoretical study of weak Galerkin finite element solutions of Turing patterns in reaction–diffusion systems*, Numerical Methods for Partial Differential Equations, *37*(1) (2021), 302–340.

[19] D. Kumar and S. P. Chakrabarty, *A predator–prey model with additional food supply to predators: dynamics and applications*, Computational and Applied Mathematics, *37* (2018), 763–784.

[20] D. Levy, H. A. Harrington, and R. A. Van Gorder, *Role of seasonality on predator–prey–subsidy population dynamics*, Journal of Theoretical Biology, *396* (2016), 163–181.

[21] X. Li and Q. Wang, *Analysis of the inherent instability of the interpolating moving least squares method when using improper polynomial bases*, Engineering Analysis with Boundary Elements, *73* (2016), 21–34.

[22] Y. Liu, Q. Cao, and W. Yang, *Influence of allee effect and delay on dynamical behaviors of a predator–prey system*, Computational and Applied Mathematics, *41*(8) (2022), 396.

[23] A. Madzvamuse, P. K. Maini, and A. J. Wathen, *A moving grid finite element method for the simulation of pattern generation by Turing models on growing domains*, Journal of Scientific Computing, *24*(2) (2005), 247–262.

[24] P. Majumdar, B. Mondal, S. Debnath, and S. Sarkar, and U. Ghosh, *Effect of fear and delay on a prey-predator model with predator harvesting*, Computational and Applied Mathematics, *41*(8) (2022), 357.

[25] A. Mazzia, G. Pini, and F. Sartoretto, *Numerical investigation on direct MLPG for 2D and 3D potential problems*, Computer Modeling in Engineering & Sciences(CMES), *88*(3) (2012), 183–209.

[26] D. Mirzaei and R. Schaback, *Direct meshless local Petrov-Galerkin (DMLPG) method: a generalized MLS approximation*, Applied Numerical Mathematics, *68* (2013), 73–82.

[27] D. Mirzaei, R. Schaback, and M. Dehghan, *On generalized moving least squares and diffuse derivatives*, IMA Journal of Numerical Analysis, *32*(3) (2012), 983–1000.

[28] M. Nemati, M. Shafiee, and H. Ebrahimi, *A meshless technique based on the radial basis functions for solving systems of partial differential equations*, Computational Methods for Differential Equations, *10*(2) (2022), 526–537.

[29] M. W. Ni, A. J. Kassab, and E. Divo, *Automated hybrid singularity superposition and anchored grid pattern bem algorithm for the solution of inverse geometric problems*, Engineering Analysis with Boundary Elements, *73* (2016), 69–78.

[30] S. A. Sarra, *A local radial basis function method for advection–diffusion–reaction equations on complexly shaped domains*, Applied Mathematics and Computation, *218*(19) (2012), 9853–9865.

[31] F. Sartoretto, A. Mazzia, and G. Pini, *The DMLPG meshless technique for Poisson problems*, Appl Math Sci, *8*(164) (2014) 8233–8250.

[32] F. Shakeri and M. Dehghan, *The finite volume spectral element method to solve Turing models in the biological pattern formation*, Computers & Mathematics with Applications, *62*(12) (2011), 4322–4336.

[33] E. Shivanian, M. Hosseini, and A. Rahimi, *Meshless local radial point interpolation (mlrpi) to two dimensional wave equation with neumann's boundary conditions*, Computational Methods for Differential Equations, *8*(1) (2020), 155–172.

[34] E. Shivanian and A. Jafarabadi, *Turing models in the biological pattern formation through spectral meshless radial point interpolation approach*, Engineering with Computers, *36*(1) (2020), 271–282.

[35] E. Shivanian and A. Jafarabadi, *Numerical investigation based on a local meshless radial point interpolation for solving coupled nonlinear reaction-diffusion system*, Computational Methods for Differential Equations, *9*(2) (2021), 358–374.

[36] J. Sladek, V. Sladek, and Y. Hon, *Inverse heat conduction problems by meshless local Petrov-Galerkin method*, Engineering Analysis with Boundary Elements *30*(8) (2006), 650–661.

[37] M. Tehseen Saleem and I. Ali, *Numerical simulations of turing patterns in a reaction-diffusion model with the chebyshev spectral method*, The European Physical Journal Plus, *133*(10) (2018), 1–9.

[38] A. M. Turing, *The chemical basis of morphogenesis*, Bulletin of Mathematical Biology, *52*(1) (1990), 153–197.

[39] L. Zhu and L. He, *Two different approaches for parameter identification in a spatial–temporal rumor propagation model based on Turing patterns*, Communications in Nonlinear Science and Numerical Simulation, *107* (2022), 106174.