Engineering with Computers (2021) 37:1159–1179
https://doi.org/10.1007/s00366-019-00877-z

ORIGINAL ARTICLE

Local radial basis function–finite-difference method to simulate some models in the nonlinear wave phenomena: regularized long-wave and extended Fisher–Kolmogorov equations

Mehdi Dehghan[1] · Nasim Shafieeabyaneh[1]

Received: 25 April 2019 / Accepted: 17 October 2019 / Published online: 12 November 2019

Abstract

In this investigation, we concentrate on solving the regularized long-wave (RLW) and extended Fisher–Kolmogorov (EFK) equations in one-, two-, and three-dimensional cases by a local meshless method called radial basis function (RBF)–finite-difference (FD) method. This method at each stencil approximates differential operators such as finite-difference method. In each stencil, it is necessary to solve a small-sized linear system with conditionally positive definite coefficient matrix. This method is relatively efficient and has low computational cost. How to choose the shape parameter is a fundamental subject in this method, since it has a palpable effect on coefficient matrix. We will employ the optimal shape parameter which results from algorithm of Sarra (Appl Math Comput 218:9853–9865, 2012). Then, we compare the approximate solutions acquired by RBF–FD method with theoretical solution and also with results obtained from other methods. The numerical results show that the RBF–FD method is suitable and robust for solving the RLW and EFK equations.

Keywords Nonlinear regularized long-wave (RLW) equation · Extended Fisher–Kolmogorov (EFK) equation · Radial basis functions (RBFs) · Local meshless method · RBF–FD method · Shape parameter

Mathematics Subject Classification 65M06 · 65M70 · 35K55

1 Introduction

Partial differential equations which have one or more nonlinear terms are called nonlinear partial differential equations (NLPDEs). Their application is extensive in many branches of sciences such as physics, chemistry, and engineering, since many phenomena can be modeled using the NLPDEs [17]. The most fundamental difficulty in studying NLPDEs is lack of a general technique for solving such equations, so any equation must be examined singly. Note that because the exact solution of NLPDEs is not easily obtained, numerical methods must be used to solve these equations [21].

1.1 A glance over the RLW equation

Here, we focus on one of the most important nonlinear equations which is called regularized long-wave (RLW) equation [12, 17]:

$$u_t + \nabla.u - \mu\Delta u_t + u.\nabla u = 0, \qquad \mathbf{x} \in \Omega, \qquad 0 < t \leq T, \quad (1.1)$$

with boundary condition

$$u(\mathbf{x}, t) = g_1(t), \qquad \mathbf{x} \in \partial\Omega,$$

and initial condition

$$u(\mathbf{x}, 0) = g_2(\mathbf{x}), \qquad \mathbf{x} \in \Omega,$$

where $\Omega \subset \mathbb{R}^d$, $d \leq 3$, and μ is positive constant. Equation (1.1) was first presented to model the treatment of the undular bore by Peregrine [54]. It plays a serious role in describing physical model with nonlinear dispersive waves. We mention some of them below [16, 47]:

- ion-acoustic and magneto-hydrodynamic waves in plasma;

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

Nasim Shafieeabyaneh
nasim.shafiee@aut.ac.ir; nasim.shafieeabyaneh@gmail.com

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

- pressure waves in liquid bubbles;
- rotating flow down a tube;
- longitudinal dispersive waves in elastic rods;
- phonon packets in nonlinear crystals.

The solution of Eq. (1.1) is special type of solitary wave which is called soliton. Solitons are localized waves of permanent form, they can collide with other solitons, so that they will appear unchanged after the collision. The reader can refer to [35] for more information about solitons. Authors of [6] found an analytical solution under the limited initial and boundary conditions for the one-dimensional RLW equation. Since this solution is not effective, the need for a suitable numerical method is highlighted. Hence, the RLW equation has been solved by various numerical methods including finite-difference scheme [6, 24, 25], finite-element method [10, 11], and pseudo-spectral method [22, 32]. Some researchers also used several kinds of meshless methods to find a numerical solution of the 1-D RLW equation. A collocation method based on radial basis functions (RBFs) has been extended for this equation in [64]. The author of [58] used the Petrov–Galerkin method to solve the GRLW equation with choosing a linear hat function and a quintic B-spline function as the trial and test function, respectively.

The exact solution for two-dimensional RLW equation has been presented in [62]. The cnoidal wave solutions of this model were found by elliptic integral method in paper [73]. The 2-D case of Eq. (1.1) which is appeared in the investigation of the Rossby wave in fluid rotating and the drift waves in plasma has been solved with a combination of the analog equation method and the boundary knot method [16]. The main purpose of [17] is to develop element-free Galerkin meshless method for solving this equation on a non-rectangular region. Author of [52] studied and analyzed the Crank–Nicolson Galerkin method for the Benjamin–Bona–Mahony (BBM) equation. In [59], two conservative difference schemes, one two-level and nonlinear implicit and the other three-level and linear implicit have been studied for solving the BBM-KdV equation. Authors of [60] proposed a high-order nonlinear conservative difference method for the RLW–KdV equation. Furthermore, they indicated the stability and convergence of the presented method. Ghiloufi et al. [31] focused on applying a nonlinear conservative fourth-order difference technique for a model of nonlinear dispersive equations. Moreover, they proved the convergence of the used method by employing the energy method.

1.2 A glance over the EFK equation

The Fisher–Kolmogorov (FK) equation is a nonlinear second-order differential equation which is used to model many phenomena related to reaction–diffusion in chemistry and biology fields [1]. Some researchers [9] added an additional stabilizing fourth-order derivative term to FK equation and called the obtained model as the extended Fisher–Kolmogorov (EFK) equation. It emerges from describing important phenomena such as:

- pattern formation [15] and spatio-temporal turbulence [9] in bistable system;
- population genetics [2];
- diffusion of domain walls in liquids crystals [74];
- traveling waves in reaction–diffusion systems [2];
- mesoscopic model of a phase transition in a binary system nearby the Lifshitz point [39].

The EFK equation is indicated as follows [40]:

$$u_t + \gamma\Delta^2 u - \Delta u + g(u) = 0, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad 0 < t \le T, \tag{1.2}$$

with initial condition

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

and boundary conditions

$$u = f_1(t), \quad \Delta u = f_2(t), \quad \mathbf{x} \in \partial\Omega, \quad t \in (0, T],$$

where $g(u) = u^3 - u$, γ is constant.

Up to now, Eq. (1.2) has attracted the attention of many scholars. In [53], its steady state has been surveyed by applying shooting technique. Authors of [15] focused on dynamics of Eq. (1.2) by choosing various values of γ. The quintic B-spline collocation scheme is offered for detecting the approximate solution of 1-D EFK equation in [51]. Danumjaya and Pani [13, 14] used the orthogonal cubic spline collocation technique and the finite-element Galerkin method to investigate Eq. (1.2). Authors of [42, 46] rendered the discretization for the 2-D EFK model by applying finite-difference schemes. Moreover, authors of [40] introduced various meshless local boundary integral equation (LBIE) techniques for solving the extended Fisher–Kolmogorov equation. In [43], a nonlinear FD scheme for the 1-D EFK equation is implemented. In addition, the convergence analysis of the obtained difference scheme has been proven by applying the energy method.

1.3 A glance over RBFs and RBF–FD schemes

Meshless methods can solve the PDEs with a set of scattered points in their computational domain. Their most advantage is that they don't require mesh generation. Radial basis functions (RBFs) which were employed by Hardy [33, 34] for the interpolation of scattered points play a significant role in meshless methods. Kansa [44, 45] was the first scholar that acquired the approximate solution of a PDE by applying the RBFs interpolation technique and using all the points in the domain. Hence, the RBFs collocation method is called Kansa's method. So far, researchers succeeded in solving many PDEs with Kansa's method, including heat transport equation [72], 1-D nonlinear Burgers equation [36], shallow water equations [37], Schrödinger equation [19], free boundary problems [38, 49], etc. In the Kansa's method, an ill-conditioned and large linear system is created that can lead to uncertain results. On the other hand, local RBF (LRBF) method has been introduced which for each point (center) uses only a limited number of points (stencil) instead of applying the entire points. This method produces a better-conditioned, sparse linear system. Therefore, it has good accuracy for bad-posed problems.

The RBF–FD method, combining the meshless and FD [20] methods, has attracted the attention of many scholars due to its advantage. The RBF–FD, which can fall into the category of LRBF techniques, is expanded in [23, 63, 66, 68, 69]. Shu and his co-authors [65] implemented the RBF–FD and LSFD for 2-D case of Poisson and Lid-Driven equations, and then, by comparing their results, they showed that the RBF–FD method has a significant accuracy. The steady convection–diffusion equations were solved by the RBF–FD method in [8]. The authors of [27] explained how to apply the RBF–FD method to the shallow water equations and they also established the accuracy and computational efficiency of the RBF–FD method. Bolling and his co-authors [7] were the first researchers to focus on the parallel performance of the RBF–FD method and introduced parallelization strategies. Bayona and his co-workers [4] presented an efficient technique for finding the solutions of PDEs using the local RBF–FD method. Moreover, they showed that the approximation error can be minimized by choosing the optimal shape parameter. The main goals of [41] are to implement the RBF–FD method for 2-D Navier–Stokes and to present an adaptive shape parameter for RBFs to improve numerical results. The authors of [67] solved atmospheric flow by a scalable RBF–FD and they also concentrated on parallelizing of this method. A new method was introduced to decrease the CPU time by combining the RBF–FD idea with the proper orthogonal decomposition technique [18]. Authors of [5] demonstrated that achieving high accuracy without choosing optimal shape parameter is possible when the RBF–FD approximations are combined with polyharmonic splines. Petras and his co-workers [55] solved PDEs on moving surface using the least-squares implicit RBF–FD method.

1.4 The main purpose and framework of this article

In current article, we are going to implement the RBF–FD for solving 1-, 2-, and 3-D cases of the RLW and EFK equations. The framework of this paper is as follows:

- In Sect. 2, the RBFs and RBF–FD schemes are described.
- In Sect. 3, we employ the RBF–FD technique for Eq. (1.1) in dimensions one and two.
- Section 4 relates to the implementation of RBF–FD technique for Eq. (1.2).
- In Sect. 5, the numerical results of the simulation of RBF–FD method are reported for RLW and EFK equations.
- Section 6 is dedicated to the overall conclusion of this investigation.

2 The RBFs and RBF–FD methods

To obtain the RBF–FD formulation, we need to RBF interpolants. Hence, we will first review the essential concepts of the RBF interpolation.

2.1 RBF interpolation

A radial function in each point can be defined based on the distance from that point to the origin. It should be noted that a set of specific centers can be considered as the origin that we show them with $\mathbf{x}_i$ for $i = 1, 2, \dots, N$. On the other words, let $\mathbf{x} \in \mathbb{R}^d$, $\psi : \mathbb{R}^d \to \mathbb{R}$ be a radial function, then $\psi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$ [26, 70], where φ is a real-valued, non-negative and single variable function and Euclidean norm denoted by $\|.\|$

We presume that a set of scattered points $\mathbf{x}_i \in \mathbb{R}^d$ and function values $u(\mathbf{x}_i)$ for $i = 1, \dots, N$ are given. Finding an interpolant as below is known as the RBFs interpolation problem [26, 28, 70]:

$$P(\mathbf{x}) = \sum_{i=1}^{N} \eta_i \psi(\|\mathbf{x} - \mathbf{x}_i\|),$$

where ψ is a radial function. The coefficients η_i are specified by exerting the interpolation conditions:

$$P(\mathbf{x}_i) = u(\mathbf{x}_i), \quad i = 1, \dots, N.$$

Thus, the above condition leads to linear system:

$$\begin{bmatrix} \psi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \psi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & \psi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \psi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \psi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & \psi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \psi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \dots & \psi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_N \end{bmatrix} = \begin{bmatrix} u(\mathbf{x}_1) \\ u(\mathbf{x}_2) \\ \vdots \\ u(\mathbf{x}_N) \end{bmatrix}.$$

The most popular radial functions are listed below [26, 29, 70]:

- Gaussian kernel (GA):

$$\psi(r) = \exp\left(-(\varepsilon r)^2\right).$$

- Multiquadric kernel (MQ):

$$\psi(r) = \left(1 + (\varepsilon r)^2.\right)^{\frac{1}{2}}$$

- Inverse multiquadric kernel (IMQ):

$$\psi(r) = \frac{1}{\left(1 + (\varepsilon r)^2\right)^{\frac{1}{2}}}.$$

- Inverse quadric kernel (IQ):

$$\psi(r) = \frac{1}{\left(1 + (\varepsilon r)^2\right)}.$$

In the present investigation, we apply the (MQ) radial function in 1- , 2-, and 3- D.

2.1.1 Shape parameter

The RBFs depend on the shape parameter, which greatly affects the accuracy and stability of the interpolation. Note that when two points are close to each other, the selecting of shape parameter can prevent producing the singular problem [48, 56]. The shape parameter is considered in two forms :

1. Fixed shape parameter
   In this case, the shape parameter is the same for all centers. In fact, the distance between the points and the centers does not affect the choice of shape parameter.
2. Variable shape parameter
   In this case, a different shape parameter is considered for each center. In fact, the distance between the points and the centers plays a role in choosing the shape parameter, for example, $\psi(x) = \sqrt{(x - x_i)^2 + \varepsilon_i^2}$, so that if the distance from one point to center is large (small), the small (large) shape parameter is considered.

There are two ways to find the optimal shape parameter:

- Test and trial
  This procedure is a completely experimental method.
- Optimal algorithms
  Certain researchers have done many studies to find the good shape parameter [56, 61]. Particularly, authors of [3, 4] succeeded in finding an optimal value of the constant and variable shape parameters for the RBF–FD method which minimize the error of approximation. Sarra [61] proposed an algorithm for choosing the optimal shape parameter in which the SVD of interpolation matrix is used.

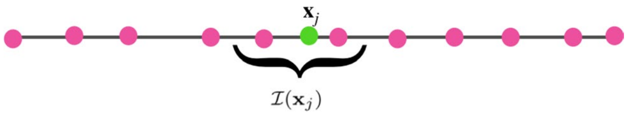


Fig. 1 One-dimensional case of considered stencil

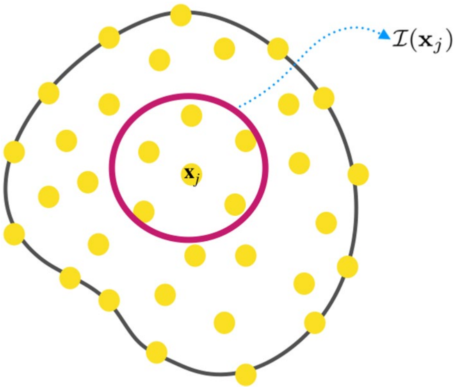


Fig. 2 Two-dimensional case of considered stencil

2.2 RBF–FD method

This section will be devoted to explaining the structure of the RBF–FD method. Figures 1 and 2 demonstrate the applied stencil in 1- and 2-D situations.

At first, we suppose that N points $\left\{\mathbf{x}_i\right\}_{i=1}^{N}$ are scattered on the computational domain $\Omega \subset \mathbb{R}^d$. The main part of RBF–FD method is that the values of unknown function, u_i $(i = 1, \dots, \left|\mathcal{I}(\mathbf{x}_j)\right|)$, are used to approximate any desired linear differential operator ($\mathcal{L}$) as below [29, 57]:

$$\mathcal{L}u(\mathbf{x}_j) \approx \sum_{i \in \mathcal{I}(\mathbf{x}_j)} w_i u(\mathbf{x}_i), \tag{2.1}$$

where $\mathcal{I}(\mathbf{x}_j)$ contains the index of the points that are in the stencil of $\mathbf{x}_j$.

In addition, the local interpolation problem is known as [29]

$$P_I(\mathbf{x}) = \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} \lambda_i \psi(\|\mathbf{x} - \mathbf{x}_i\|). \tag{2.2}$$

To get RBF–FD formulation [57], we employ the Lagrangian form of the local RBF interpolant (2.2):

$$P_l(\mathbf{x}) = \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} \psi_i^*(\mathbf{x}) u(\mathbf{x}_i), \tag{2.3}$$

in which $\psi_i^*(\mathbf{x}_j) = \delta_{ij},\ j = 1, \dots, \left|I(\mathbf{x}_j)\right|$.

A closed form for $\psi_i^*(\mathbf{x}_j)$ is presented in [30, 57] as

$$\psi_i^*(\mathbf{x}) = \frac{\det(A_i(\mathbf{x}))}{\det(A)}, \tag{2.4}$$

where $A = \psi(\left\|\mathbf{x}_j - \mathbf{x}_i\right\|)$, $i,j = 1, \dots, \left|\mathcal{I}(\mathbf{x}_j)\right|$ and A_i is achieved by inserting the ith row of A with vector:

$$B = \left[\ \psi(\left\|\mathbf{x} - \mathbf{x}_1\right\|)\ \psi(\left\|\mathbf{x} - \mathbf{x}_2\right\|)\ \dots\ \psi(\|\mathbf{x} - \mathbf{x}_{|\mathcal{I}(\mathbf{x}_j)|}\|)\right].$$

By exerting operator $\mathcal{L}$ on (2.3), the approximation of $\mathcal{L}u(\mathbf{x}_j)$ at $\mathbf{x}_j$ is obtained as [57, 71]

$$\mathcal{L}u(\mathbf{x}_j) \approx \mathcal{L}P_l(x_j) = \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} \mathcal{L}\psi_i^*(\mathbf{x}_j) u(\mathbf{x}_i). \tag{2.5}$$

Hence, from Eqs. (2.1) and (2.5), it can be concluded that $w_i = \mathcal{L}\psi_i^*(\mathbf{x}_j)$. Finally, the weights can be found by solving the linear system [29, 57]:

$$Aw^T = \left[\left(\mathcal{L}B(\mathbf{x}_i)\right)\right]^T, \tag{2.6}$$

as

$$\begin{bmatrix} \psi(\left\|\mathbf{x}_1 - \mathbf{x}_1\right\|) & \psi(\left\|\mathbf{x}_2 - \mathbf{x}_1\right\|) & \dots & \psi(\left\|\mathbf{x}_{|\mathcal{I}(\mathbf{x}_i)|} - \mathbf{x}_1\right\|) \\ \psi(\left\|\mathbf{x}_1 - \mathbf{x}_2\right\|) & \psi(\left\|\mathbf{x}_2 - \mathbf{x}_2\right\|) & \cdots & \psi(\left\|\mathbf{x}_{|\mathcal{I}(\mathbf{x}_i)|} - \mathbf{x}_2\right\|) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(\left\|\mathbf{x}_1 - \mathbf{x}_{|\mathcal{I}(\mathbf{x}_i)|}\right\|) & \psi(\left\|\mathbf{x}_2 - \mathbf{x}_{|\mathcal{I}(\mathbf{x}_i)|}\right\|) & \dots & \psi(\left\|\mathbf{x}_{|\mathcal{I}(\mathbf{x}_i)|} - \mathbf{x}_{|\mathcal{I}(\mathbf{x}_i)|}\right\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{|\mathcal{I}(\mathbf{x}_i)|} \end{bmatrix} = \begin{bmatrix} \mathcal{L}\psi(\left\|\mathbf{x} - \mathbf{x}_1\right\|)|_{\mathbf{x}=\mathbf{x}_i} \\ \mathcal{L}\psi(\left\|\mathbf{x} - \mathbf{x}_2\right\|)|_{\mathbf{x}=\mathbf{x}_i} \\ \vdots \\ \mathcal{L}\psi(\left\|\mathbf{x} - \mathbf{x}_{|\mathcal{I}(\mathbf{x}_i)|}\right\|)|_{\mathbf{x}=\mathbf{x}_i} \end{bmatrix}. \tag{2.7}$$

Now, to get the optimal shape parameter, we use the algorithm provided by Sarra [61]. The parameters required for Algorithm 1 are defined as

- $\mathbf{Q}$ denotes the interpolation matrix.
- $\boldsymbol{\sigma}_{\min}$ and $\boldsymbol{\sigma}_{\max}$, which are the smallest and largest singular values, are obtained from the singular-value decomposition (SVD).
- $\mathrm{K}_{\min} = 10^2$ and $\mathrm{K}_{\max} = 10^4$.
- $\boldsymbol{\varepsilon}_{\text{Increment}} = \dfrac{1}{N}$.

Algorithm 1: Optimal shape parameter [61]

```
Input: $\mathrm{K}_{\min}, \mathrm{K}_{\max}, \boldsymbol{\varepsilon}_{\text{Increment}}$
Output: Optimal shape parameter $\varepsilon$
function OptimalShapeParameter($\mathrm{K}_{\min}, \mathrm{K}_{\max}$)
    $\mathrm{K} = 0$;
    while $\mathrm{K} < \mathrm{K}_{\min}$ or $\mathrm{K} > \mathrm{K}_{\max}$ do
        Construct interpolation matrix $\mathbf{Q}$;
        $[U, S, V] = svd(\mathbf{Q})$;
        $\mathrm{K} = \frac{\sigma_{\max}}{\sigma_{\min}}$;
        if $\mathrm{K} < \mathrm{K}_{\min}$ then
            $\varepsilon = \varepsilon - \boldsymbol{\varepsilon}_{Increment}$;
        else
            $\varepsilon = \varepsilon + \boldsymbol{\varepsilon}_{Increment}$;
    return $\varepsilon$
```

3 Regularized long-wave equation

This section is allocated to the discretization of 1-D and 2-D RLW equations by RBF–FD procedure.

3.1 The one-dimensional RLW equation

We intend to illustrate the discretization of RBF–FD method for the 1-D case of Eq. (1.1):

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} - \mu \frac{\partial}{\partial t}\left(\frac{\partial^2 u}{\partial x^2}\right) + u\frac{\partial u}{\partial x} = 0, \quad \mathbf{x} = x \in \mathbb{R}, \quad 0 < t \le T, \tag{3.1}$$

in which $u(x, t)$ shows the horizontal velocity of fluid.

From Eq. (2.1), the approximation solution is written as

$$u(\mathbf{x}_j, t) = \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i u_i(t), \tag{3.2}$$

also, the first- and second-order derivatives are approximated as given in the following:

$$\begin{aligned} \frac{\partial u(\mathbf{x}_j, t)}{\partial x} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^x u_i(t), \\ \frac{\partial^2 u(\mathbf{x}_j, t)}{\partial x^2} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t). \end{aligned} \tag{3.3}$$

Substituting Eqs. (3.2) and (3.3) in Eq. (3.1) gives

$$\begin{aligned} &\frac{\mathrm{d}u_j(t)}{\mathrm{d}t} + \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^x u_i(t) - \mu \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t)\right) \\ &\quad + u_j(t)\left(\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^x u_i(t)\right) = 0, \quad j = 1, 2, \ldots, N. \end{aligned} \tag{3.4}$$

After collocating the points; we receive the system of ordinary differential equations (ODEs) of the form:

$$\frac{\mathrm{d}\left(\mathbf{I} - \mu \mathbf{W_{xx}}\right)\vec{\mathbf{u}}}{\mathrm{d}t} + \mathbf{W_x}\vec{\mathbf{u}}(t) + \vec{\mathbf{u}}. * \left(\mathbf{W_x}\vec{\mathbf{u}}(t)\right) = 0, \tag{3.5}$$

where

$$\mathbf{u}(t) = \left[\, u_1(t) \;\; u_2(t) \;\; \ldots \;\; u_{N-1}(t) \;\; u_N(t) \,\right]^T.$$

The structure of matrices $\mathbf{W_x}$ and $\mathbf{W_{xx}}$ depends on the number of points in each stencil ($\left|\mathcal{I}(\mathbf{x}_j)\right|$). For instance, if three points in each stencil are selected, then $\mathbf{W_x}$ and $\mathbf{W_{xx}}$ are tridiagonal matrices.

3.1.1 Time discretization

So far, the spatial derivatives of the RLW equation have been discretized by the RBF–FD method, and then, the system of ODEs (3.5) has been acquired. There are various methods for solving this equation, such as Euler, Runge–Kutta, etc. However, we preferred to use a command in MATLAB software. In the following, Algorithm 2 illustrates the full discretization of the RLW equation.

Algorithm 2: Full discretization of the 1-D RLW equation

```
Enter the required simulation parameters such as : N, $n_s$, T, $dt$, $t_{step} = [0 : dt : T]$;
Construct the differentiation matrices $\mathbf{W_x}$ and $\mathbf{W_{xx}}$;
Make the right-hand side of ODE obtained :
    $\mathrm{RHS} = @(t, u) \;\; -\mathbf{W_x}u(t) - u(t). * (\mathbf{W_x}u(t))\,;$
Construct coefficient matrix of ODE obtained :
    $\mathbf{M} = \mathbf{I} - \mu\mathbf{W}_{xx};$
Enter initial condition ($u_{init}$);
To solve the ODE, use the following command :
    $\mathrm{options} = \mathrm{odeset}('\mathrm{RelTol}', 2.3e-14, '\mathrm{AbsTol}', 1e-16, '\mathrm{Mass}', \mathbf{M});$
    $[t, w] = \mathrm{ode113}(@(t, u)\mathrm{RHS}(t, u), t_{step}, u_{init}, \mathrm{options});$
Compute the absolute error;
```

3.2 The two-dimensional RLW equation

We now focus on the 2-D case of Eq. (1.1) and apply the RBF–FD procedure to discretize it. Let the 2-D RLW equation be

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}-\mu\,\frac{\partial}{\partial t}\left(\frac{\partial^2 u}{\partial x^2}\right)-\gamma\,\frac{\partial}{\partial t}\left(\frac{\partial^2 u}{\partial y^2}\right)+u\frac{\partial u}{\partial x}+u\frac{\partial u}{\partial y}=0,\quad \mathbf{x}=(x,y)\in\Omega\subset\mathbb{R}^2,\ \ 0<t<T. \tag{3.6}$$

According to (2.1), the first- and second-order derivatives with respect to x are approximated as

$$\begin{aligned}\frac{\partial u(\mathbf{x}_j,t)}{\partial x}&=\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^x u_i(t),\\ \frac{\partial^2 u(\mathbf{x}_j,t)}{\partial x^2}&=\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t).\end{aligned} \tag{3.7}$$

In addition, the same approximations for derivatives in y direction are applied:

$$\begin{aligned}\frac{\partial u(\mathbf{x}_j,t)}{\partial y}&=\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^y u_i(t),\\ \frac{\partial^2 u(\mathbf{x}_j,t)}{\partial y^2}&=\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{yy} u_i(t).\end{aligned} \tag{3.8}$$

Substituting Eqs. (3.7)–(3.8) in (3.6) leads to

$$\begin{aligned}&\frac{\mathrm{d}u_j(t)}{\mathrm{d}t}+\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^x u_i(t)+\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^y u_i(t)\\ &-\mu\,\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t)\right)-\gamma\,\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{yy} u_i(t)\right)\\ &+u_j(t)\left(\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^x u_i(t)\right)\\ &+u_j(t)\left(\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^y u_i(t)\right)=0,\quad j=1,2,\dots,N.\end{aligned} \tag{3.9}$$

Consequently, we can attain the matrix-vector form of (3.9) as

$$\begin{aligned}&\frac{\mathrm{d}\big(\mathbf{I}-\mu\mathbf{L}_{\mathbf{xx}}-\gamma\mathbf{L}_{\mathbf{yy}}\big)\vec{\mathbf{u}}}{\mathrm{d}t}+\mathbf{L}_{\mathbf{x}}\vec{\mathbf{u}}(t)+\mathbf{L}_{\mathbf{y}}\vec{\mathbf{u}}(t)\\ &+\vec{\mathbf{u}}.*\big(\mathbf{L}_{\mathbf{x}}\vec{\mathbf{u}}(t)\big)+\vec{\mathbf{u}}.*\big(\mathbf{L}_{\mathbf{y}}\vec{\mathbf{u}}(t)\big)=0,\end{aligned} \tag{3.10}$$

in which $\mathbf{L}_{\mathbf{xx}}$ is Kronecker product $\mathbf{I}$ in $\mathbf{W}_{\mathbf{xx}}$ ($\mathbf{L}_{\mathbf{xx}}=\mathbf{I}\otimes\mathbf{W}_{\mathbf{xx}}$), $\mathbf{L}_{\mathbf{yy}}=\mathbf{W}_{\mathbf{yy}}\otimes\mathbf{I}$, $\mathbf{L}_{\mathbf{x}}=\mathbf{I}\otimes\mathbf{W}_{\mathbf{x}}$, $\mathbf{L}_{\mathbf{y}}=\mathbf{W}_{\mathbf{y}}\otimes\mathbf{I}$ and

$$\mathbf{u}(t)=\big[\,u_1(t)\ \ u_2(t)\ \ \dots\ \ u_{N^2-1}(t)\ \ u_{N^2}(t)\big]^{\,T}.$$

3.2.1 Time discretization

So far, the spatial derivatives of the RLW equation have been discretized by the RBF–FD method. Now, the fourth-order Runge–Kutta method is applied to solve the system of ODEs (3.10). Hence, we introduce Algorithm 3 for the full discretization of the RLW equation.

Algorithm 3: Full discretization of the 2-D RLW equation

```
Enter the required simulation parameters such as : N, $n_s$, T, $dt$ and $N_t$;
Construct the differentiation matrices $\mathbf{W_x}$ and $\mathbf{W_{xx}}$;
$\mathbf{L_x} = \text{sparse}(\text{kron}(\mathbf{I}, \mathbf{W_x}))$;    $\mathbf{L_y} = \text{sparse}(\text{kron}(\mathbf{W_x}, \mathbf{I}))$;
$\mathbf{L_{xx}} = \text{sparse}(\text{kron}(\mathbf{I}, \mathbf{W_{xx}}))$;    $\mathbf{L_{yy}} = \text{sparse}(\text{kron}(\mathbf{W_{xx}}, \mathbf{I}))$;
Make the right-hand side of ODE obtained :
    $\text{RHS} = @(t, u) \quad -\mathbf{L_x}u(t) - \mathbf{L_y}u(t) - u(t).*(\mathbf{L_x}u(t)) - u(t).*(\mathbf{L_y}u(t))\,;$
Construct coefficient matrix of ODE obtained :
    $\mathbf{M} = \mathbf{I} - \mu\mathbf{L_{xx}} - \gamma\mathbf{L_{yy}};$
Enter initial condition $(u_{init})$;
for $kk = 1 : N_t$ do
    $time = kk * dt$;
    $k_1 = dt\text{RHS}(time, u_{init})$;
    $k_2 = dt\text{RHS}(time + \frac{dt}{2}, u_{init} + \frac{k_1}{2})$;
    $k_3 = dt\text{RHS}(time + \frac{dt}{2}, u_{init} + \frac{k_2}{2})$;
    $k_4 = dt\text{RHS}(time + dt, u_{init} + k_3)$;
    $u_{init} = u_{init} + \frac{1}{6}(k_1 + 2(k_2 + k_3) + k_4)\,;$
    $u = \text{reshape}(u_{init}, \text{N}, \text{N})$
    Apply boundary condition;
    $u_{init} = u(:)$;
Compute the absolute error;
```

4 Extended Fisher–Kolmogrov equation

To solve 1-D and 2-D extended Fisher–Kolmogorov (EFK) equations, we employ the RBF–FD technique. First, how to apply the RBF–FD method for the 1-D case of the EFK equation is explained. Then, the attention is paid on its 2-D case.

4.1 The one-dimensional EFK equation

The one-dimensional EFK equation is

$$\frac{\partial u}{\partial t} + \gamma\frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial x^2} + g(u) = 0, \quad \mathbf{x} = x \in \Omega \subset \mathbb{R},\ 0 < t \le T, \tag{4.1}$$

which can be rewritten by changing the variable as the following system:

$$\begin{cases} v = \dfrac{\partial^2 u}{\partial x^2}, \\[2ex] \dfrac{\partial u}{\partial t} = -\gamma\dfrac{\partial^2 v}{\partial x^2} + v - g(u). \end{cases} \tag{4.2}$$

According to Eq. (2.1), second-order derivatives are approximated as below:

$$\begin{aligned} \frac{\partial^2 u(\mathbf{x}_j, t)}{\partial x^2} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t), \\ \frac{\partial^2 v(\mathbf{x}_j, t)}{\partial x^2} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} z_i^{xx} v_i(t). \end{aligned} \tag{4.3}$$

By substituting Eq. (4.3) in (4.2), we acquire

$$\begin{cases} v_j(t) = \displaystyle\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t), \\[2ex] \dfrac{\mathrm{d}u_j(t)}{\mathrm{d}t} = -\gamma \displaystyle\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} z_i^{xx} v_i(t) + v_j(t) - g(u_j(t)), \end{cases} \tag{4.4}$$

for $j = 1, 2, \ldots, N$. Finally, the matrix-vector form of (4.4) is written as

$$\begin{cases} \vec{\mathbf{v}}(t) = \mathbf{W}_{xx}\vec{\mathbf{u}}(t), \\ \dfrac{\mathrm{d}\vec{\mathbf{u}}(t)}{\mathrm{d}t} = -\gamma\mathbf{Z}_{xx}\vec{\mathbf{v}}(t) + \vec{\mathbf{v}}(t) - g(\vec{\mathbf{u}}(t)), \end{cases} \tag{4.5}$$

where

$$\mathbf{u}(t) = \left[\, u_1(t)\;\; u_2(t)\;\; \dots\;\; u_{N-1}(t)\;\; u_N(t) \right]^{\,T},$$
$$\mathbf{v}(t) = \left[\, v_1(t)\;\; v_2(t)\;\; \dots\;\; v_{N-1}(t)\;\; v_N(t) \right]^{\,T}.$$

4.1.1 Time discretization

So far, we have discretized the spatial derivatives of EFK equation by the RBF–FD method and have arrived at the system of (4.5). In the following, the full discretization of the 1-D EFK equation is introduced in Algorithm 4.

Algorithm 4: Full discretization of the 1-D EFK equation

```
Enter the required simulation parameters such as : N, $n_s$, T, $dt$
$t_{step} = [0 : dt : T]$;
Construct the differentiation matrix $\mathbf{W_{xx}}$;
Make the right-hand side of (4.5):
    $\mathrm{RHS} = @(t,s)\quad [-\gamma \mathbf{W_{xx}} s(n+1:2n) + s(n+1:2n) - g\,(s(1:n))\,; \mathbf{W_{xx}} s(1:n) - s(n+1:2n)]\,;$
Construct coefficient matrix : $\mathbf{M} = [\mathrm{eye(N)}\quad \mathrm{zeros(N)};\ \ \mathrm{zeros(N)}\quad \mathrm{zeros(N)}]$;
Enter initial condition $(y_0 = [u_{init}, v_{init}])$;
To solve the ODE obtained, use the following command :
    $\mathrm{options} = \mathrm{odese}t('\mathrm{mass}', \mathbf{M}, '\mathrm{RelTol}', 1e-14, '\mathrm{AbsTol}', 1e-8);$
    $[t, w] = \mathrm{ode15s}(@(t,s)\mathrm{RHS}(t,s), t_{step}, y_0, \mathrm{options});$
Compute the absolute error;
```

4.2 The two-dimensional EFK equation

The two-dimensional EFK equation

$$u_t + \gamma \Delta^2 u - \Delta u + g(u) = 0, \tag{4.6}$$

may be rewritten as the following system:

$$\begin{cases} v = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}, \\ \dfrac{\partial u}{\partial t} = -\gamma \left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2} \right) + v - g(u). \end{cases} \tag{4.7}$$

From Eq. (2.1), we consider the approximation of second-order derivatives with respect to x as

$$\begin{aligned} \frac{\partial^2 u(\mathbf{x}_j, t)}{\partial x^2} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t), \\ \frac{\partial^2 v(\mathbf{x}_j, t)}{\partial x^2} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} z_i^{xx} v_i(t), \end{aligned} \tag{4.8}$$

and similarly with respect to y as

$$\begin{aligned} \frac{\partial^2 u(\mathbf{x}_j, t)}{\partial y^2} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{yy} u_i(t), \\ \frac{\partial^2 v(\mathbf{x}_j, t)}{\partial y^2} &= \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} z_i^{yy} v_i(t). \end{aligned} \tag{4.9}$$

By substituting Eqs. (4.8) and (4.9) in (4.7), the following system was obtained

$$\begin{cases} v_j(t) = \displaystyle\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{xx} u_i(t) + \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} w_i^{yy} u_i(t), \\ \dfrac{\mathrm{d}u_j(t)}{\mathrm{d}t} = -\gamma \left(\displaystyle\sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} z_i^{xx} v_i(t) + \sum_{i=1}^{|\mathcal{I}(\mathbf{x}_j)|} z_i^{yy} v_i(t) \right) + v_j(t) - g(u_j(t)), \end{cases} \tag{4.10}$$

for $j = 1, 2, \dots, N$. At the end, we can write the matrix-vector form:

$$\begin{cases} \vec{\mathbf{v}}(t) = \left(\mathbf{L_{xx}} + \mathbf{L_{yy}}\right)\vec{\mathbf{u}}(t), \\ \dfrac{\mathrm{d}\vec{\mathbf{u}}(t)}{\mathrm{d}t} = -\gamma \left(\mathbf{L_{xx}} + \mathbf{L_{yy}}\right)\vec{\mathbf{v}}(t) + \vec{\mathbf{v}}(t) - g(\vec{\mathbf{u}}(t)), \end{cases} \tag{4.11}$$

in which $\mathbf{L_{xx}}$ is Kronecker product $\mathbf{I}$ in $\mathbf{W_{xx}}$, $\mathbf{L_{yy}} = \mathbf{W_{yy}} \otimes \mathbf{I}$ and

$$\mathbf{u}(t) = \left[\, u_1(t)\;\; u_2(t)\;\; \dots\;\; u_{N^2-1}(t)\;\; u_{N^2}(t) \right]^{\,T},$$
$$\mathbf{v}(t) = \left[\, v_1(t)\;\; v_2(t)\;\; \dots\;\; v_{N^2-1}(t)\;\; v_{N^2}(t) \right]^{\,T}.$$

Table 1 Comparison of the error values of the RBF–FD method with method of [64] with $d = 0.03$ for Instance 1

T	Method of [64]		RBF–FD method		
	$N = 1440, \mathrm{d}t = 0.1$		$N = 500, \left\|\mathcal{I}(\mathbf{x}_j)\right\| = 131, \mathrm{d}t = 0.1$		
	L_2	L_∞	L_2	L_∞	CPU time
1	–	–	1.0621×10^{-7}	6.8264×10^{-8}	0.16
5	0.3632×10^{-5}	0.1061×10^{-5}	5.8174×10^{-7}	2.4763×10^{-7}	0.25
10	0.7216×10^{-5}	0.2157×10^{-5}	1.2579×10^{-6}	3.5046×10^{-7}	0.32
15	1.0788×10^{-5}	0.3269×10^{-5}	1.8474×10^{-6}	3.9316×10^{-7}	0.39
20	1.4342×10^{-5}	0.4382×10^{-5}	2.3616×10^{-6}	4.5519×10^{-7}	0.46
40	–	–	1.6360×10^{-5}	1.5766×10^{-5}	0.78
60	–	–	5.3453×10^{-4}	5.2928×10^{-4}	1.04

4.2.1 Time discretization

Here, Algorithm 5 is presented to illustrate the full discretization of the 2-D EFK equation.

Algorithm 5: Full discretization of the 2-D EFK equation

```
Enter the required simulation parameters such as : N, $n_s$, T, $dt$ and $t_{step}$
Construct the differentiation matrix $\mathbf{W_{xx}}$;
$\mathbf{L_{xx}} = \mathrm{sparse}(\mathrm{kron}(\mathbf{I}, \mathbf{W_{xx}}))$;    $\mathbf{L_{yy}} = \mathrm{sparse}(\mathrm{kron}(\mathbf{W_{xx}}, \mathbf{I}))$;
Suppose $\mathbf{L} = (\mathbf{L_{xx}} + \mathbf{L_{yy}})$. Then, make the right hand side of (4.11):
    $\mathrm{RHS} = @(t,s)\ \left[-\gamma \mathbf{L}s(\mathrm{N}^2+1:2\mathrm{N}^2) + s(\mathrm{N}^2+1:2\mathrm{N}^2) - g\left(s(1:\mathrm{N}^2)\right); \mathbf{L}s(1:\mathrm{N}^2) - s(\mathrm{N}^2+1:2\mathrm{N}^2)\right];$
Construct coefficient matrix : $\mathbf{M} = [\mathrm{eye}(\mathrm{N}^2)\quad \mathrm{zeros}(\mathrm{N}^2);\ \mathrm{zeros}(\mathrm{N}^2)\quad \mathrm{zeros}(\mathrm{N}^2)]$;
Enter initial condition ($y_0 = [u_{init}, v_{init}]$);
To solve the ODE obtained, use the following command :
    $\mathrm{options} = \mathrm{odeset}('\mathrm{mass}', \mathbf{M}, '\mathrm{RelTol}', 1e-14, '\mathrm{AbsTol}', 1e-8);$
    $[t, w] = \mathrm{ode15s}(@(t,s)\mathrm{RHS}(t,s), t_{step}, y_0, \mathrm{options});$
Compute the absolute error;
```

5 Numerical results

5.1 Instance 1

As the first instance, the 1-D RLW equation (3.1) is considered with parameter $\mu = 1$. The analytic solution of this equation introduces a solitary wave as [12]

$$u(x,t) = 3d\,\mathrm{sech}^2\left(k[x - x_0 - vt]\right),$$

where $3d$ and $v = 1 + \varepsilon d$ represent the amplitude and velocity of the wave, respectively, and $k = \sqrt{\dfrac{\varepsilon d}{4\mu v}}$.

We are employing the RBF–FD method for solving mentioned model with initial condition

$$u(x,0) = 3d\,\mathrm{sech}^2\left(k[x - x_0]\right),$$

and the boundary conditions

$$u(a,t) = u(b,t) = 0,$$

in which $x_0 = 0$ and $d = 0.03$ or $d = 0.1$.

The numerical solutions and absolute errors on spatial domain $[-80, 100]$ at different final times with $\mathrm{d}t = 0.1$, $N = 500, \left|\mathcal{I}(\mathbf{x}_j)\right| = 131$ $(j = 1, \ldots, N)$ and $d = 0.1$ are plotted in Fig. 3. As shown in Fig. 3, the solitary wave moves to the right across the space interval $[-80, 100]$ in the time duration $[0, 20]$. The purpose of Table 1 is to compare the introduced

Table 2 Errors and CPU time acquired by the RBF–FD method with $d = 0.1$ at final time $T = 10$ for Instance 1

N	$\left\|\mathcal{I}(\mathbf{x}_j)\right\|$	L_2	L_∞	CPU time
50	15	9.0382×10^{-2}	6.1785×10^{-2}	0.41
100	25	2.1532×10^{-2}	1.1598×10^{-2}	0.42
200	51	1.0490×10^{-3}	4.0463×10^{-4}	0.47
300	111	4.4507×10^{-5}	1.4316×10^{-5}	0.56
400	241	4.2354×10^{-6}	3.4316×10^{-6}	0.63
500	341	6.8304×10^{-8}	1.7092×10^{-8}	0.71
600	441	5.4214×10^{-9}	1.3243×10^{-9}	0.94

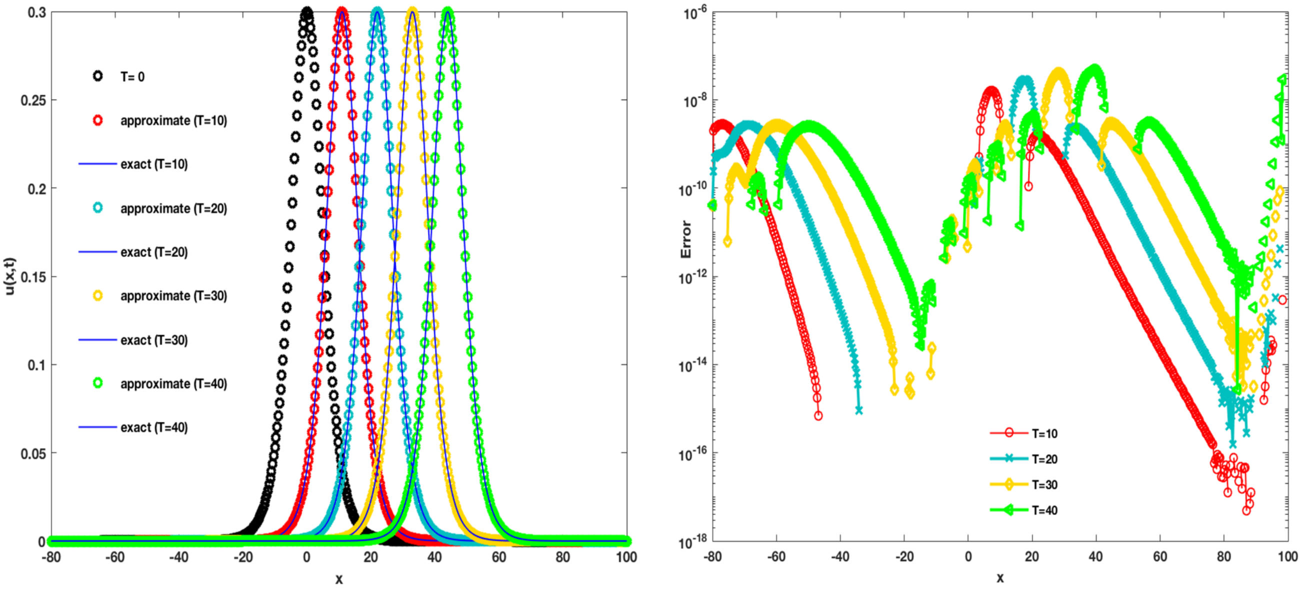


Fig. 3 Illustration of approximate and exact solutions at several final times with $N = 500, \left|\mathcal{I}(\mathbf{x}_j)\right| = 131$ $(j = 1, \dots, N)$ and $d = 0.1$ for Instance 1

method in [64] with the RBF–FD method for the parameters $d = 0.03$ and $\mathrm{d}t = 0.1$ at different final times. In Table 2, we reported the error values and CPU time acquired by RBF–FD method with the number of points N and $\left|\mathcal{I}(\mathbf{x}_j)\right|$ $(j = 1, \dots, N)$ at final time $T = 10$ for Instance 1. The behavior of numerical solution of Instance 1 with $d = 0.03$ and $\mathrm{d}t = 0.01$ at time period [0, 20] is demonstrated in Fig. 4.

5.2 Instance 2

Here, the one-dimensional RLW equation (3.1) is investigated by considering the initial condition [64]:

$$u(x, 0) = \sum_{i=1}^{2} 3\, d_i\, \mathrm{sech}^2\left(k_i(x - x_i)\right),$$

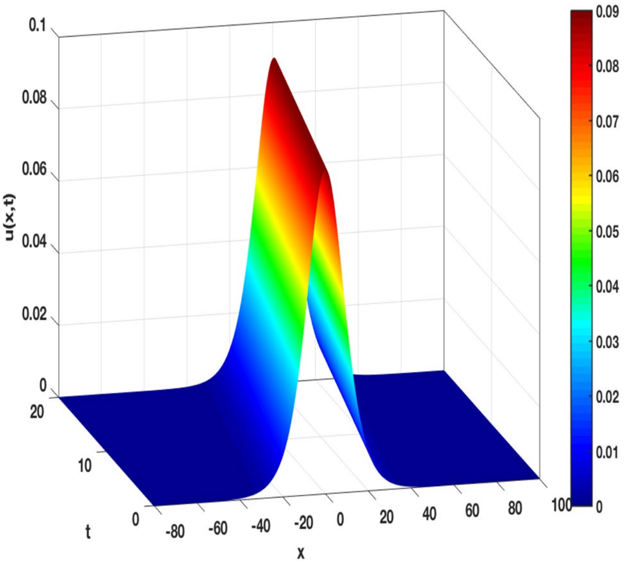


Fig. 4 Illustration of approximate solution in time interval [0, 20] with $N = 500, \left|\mathcal{I}(\mathbf{x}_j)\right| = 131$ $(j = 1, \dots, N)$ and $d = 0.03$ for Instance 1

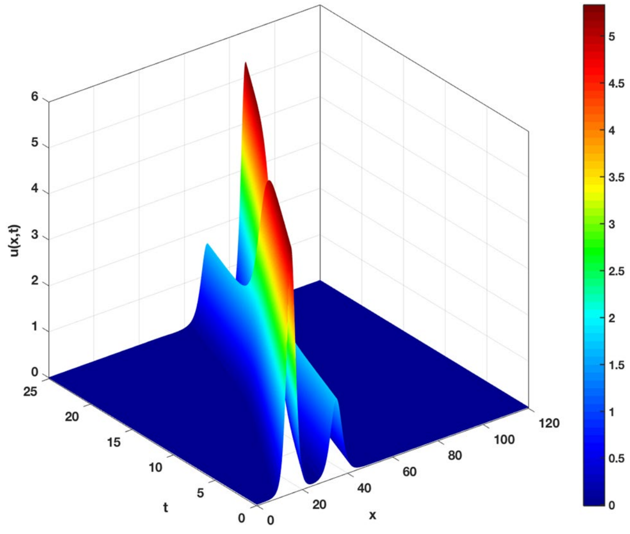


Fig. 5 Illustration of solution in time interval [0, 25] with $\mathrm{d}t = 0.01, N = 500$ and $\left|\mathcal{I}(\mathbf{x}_j)\right| = 351$ $(j = 1, \dots, N)$ for Instance 2

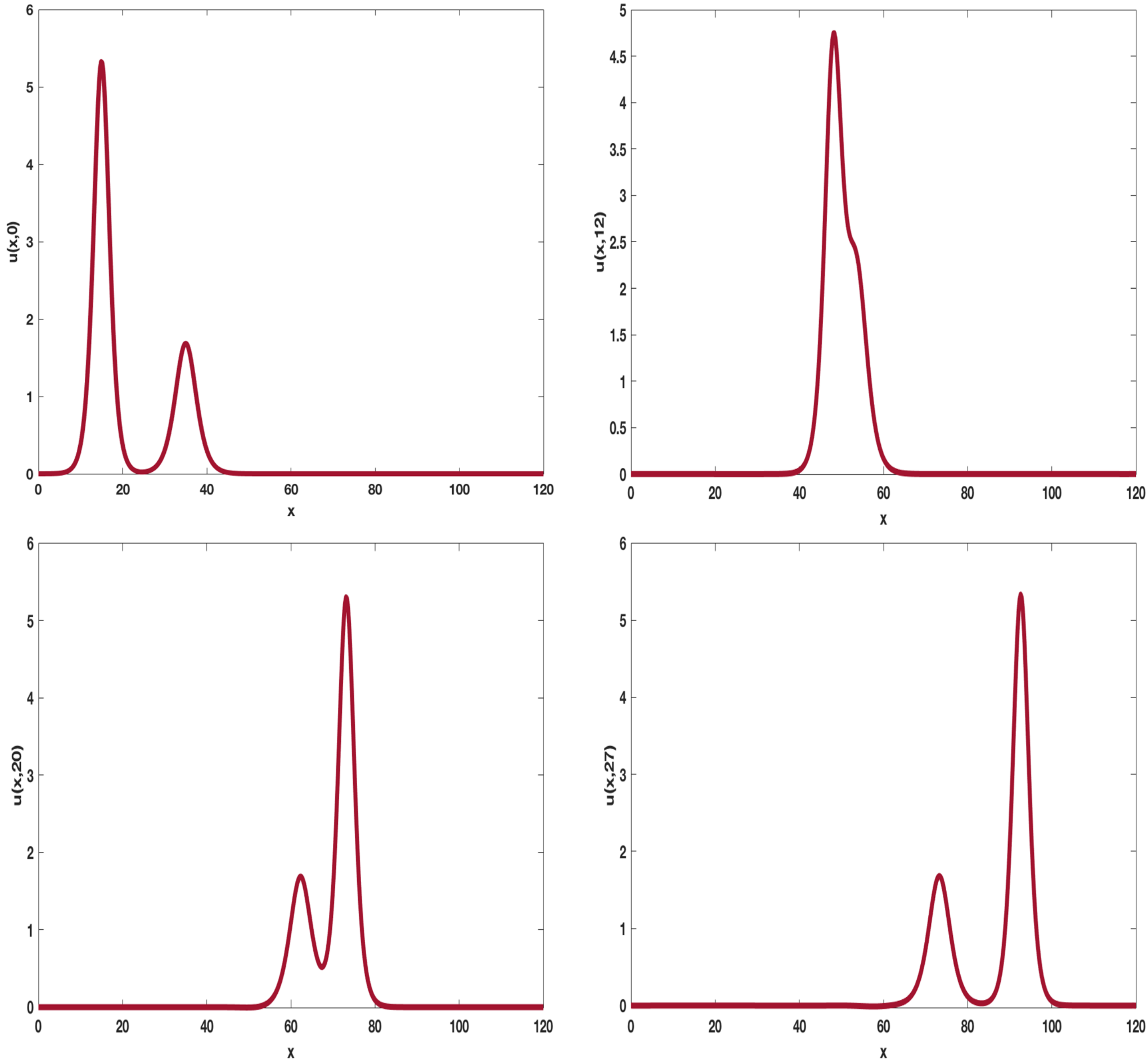


Fig. 6 Graphs of approximate solution at different final times with $\mathrm{d}t = 0.01$, $N = 500$ and $\left|\mathcal{I}(\mathbf{x}_j)\right| = 341$ $(j = 1, \ldots, N)$ for Instance 2

in which $d_i = \dfrac{4k_i^2}{1-4k_i^2}$. Moreover, its boundary conditions are assumed to be zero Dirichlet. The solution of this model presents the interaction of two solitary waves. The parameters of this example are $\varepsilon = \mu = 1$, $k_1 = 0.4$, $k_2 = 0.3$, $x_1 = 15$, and $x_2 = 35$.

We are employing the RBF–FD method to simulate the behavior of two solitary waves that are moving in opposite direction. Figure 5 presents the approximate solution in time interval [0, 25] over domain [0, 120] for Instance 2.

The approximate solutions of Instance 2 at different final times with $\mathrm{d}t = 0.01$, $N = 500$ and $\left|\mathcal{I}(\mathbf{x}_j)\right| = 341$ $(j = 1, \ldots, N)$ are indicated in Fig. 6. As shown in Fig. 6, two waves move towards each other, collide at $T = 12$ and then pass through each other.

5.3 Instance 3

In this problem, we focus on the 1-D RLW equation (3.6) which has exact solution as [17]

$$u(x, y, t) = \frac{q}{2}\mathrm{sech}^2\left[k(x + y - x_0 - y_0 - vt)\right],$$

where $q = 3(v-2)$, $k = \dfrac{\sqrt{q}}{2p}$ and $p = \sqrt{6v}$.

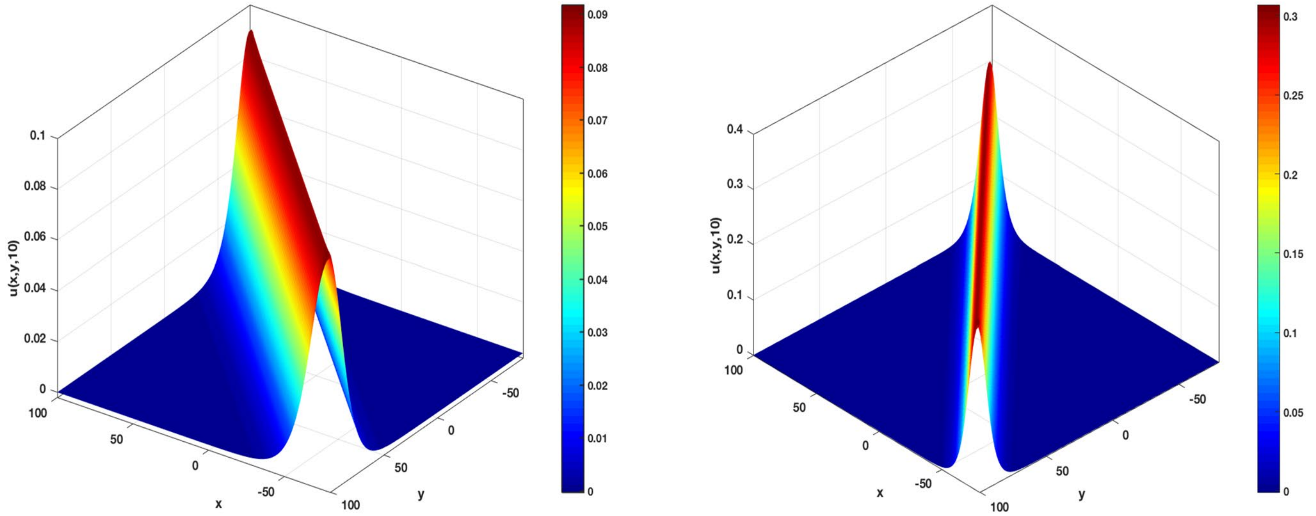


Fig. 7 Graphs of the approximate solutions at $T = 10$ with $N = 130$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 91$ $(j = 1, \dots, N)$ and $\mathrm{d}t = 0.001$ and $v = 2.2$ (right graph) and $v = 2.06$ (left graph) for Instance 3

Table 3 Comparison between the L_∞ error norms of the RBF–FD method and method of [17] with $v = 2.06$ for Instance 3

T	Method of [17]			RBF–FD method	
	$N = 540$	$N = 1080$	$N = 2160$	$(N, \left|\mathcal{I}(\mathbf{x}_j)\right|) = (80, 41)$	$(N, \left|\mathcal{I}(\mathbf{x}_j)\right|) = (90, 71)$
10	4.9044×10^{-2}	1.4938×10^{-2}	5.9543×10^{-3}	2.2811×10^{-3}	2.0083×10^{-3}
20	2.6745×10^{-2}	1.1374×10^{-2}	4.0899×10^{-3}	6.4818×10^{-3}	5.9434×10^{-3}

This model is solved by employing the RBF–FD method with various values of N and $\left|\mathcal{I}(\mathbf{x}_j)\right|$ $(j = 1, \dots, N)$ at several final times on region $\Omega = [-80, 100]^2$. Figure 7 indicates the approximate solution obtained by RBF–FD method with the parameters $T = 10$, $\mathrm{d}t = 0.001$, $N = 130$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 91$ $(j = 1, \dots, N)$, $v = 2.2$ (right graph) and $v = 2.06$ (left graph) for Instance 3.

We compare the introduced numerical results in [17] with the RBF–FD method for various values of N and $\left|\mathcal{I}(\mathbf{x}_j)\right|$ $(j = 1, \dots, N)$ at $T = 10, 20$ for Instance 3 in Table 3. From Table 3, it can be concluded that the same result of [17] can be obtained by selecting fewer points.

5.4 Instance 4

In this problem, the two-dimensional RLW equation (3.6) is considered which has exact solution as [17]

$$u(x, y, t) = \sum_{i=1}^{2} \frac{q_i}{2} \mathrm{sech}^2 \left[k_i(x + y - x_{0i} - y_{0i} - v_i t)\right],$$

where $q_i = 3(v_i - 2)$, $k_i = \dfrac{\sqrt{q_i}}{2p_i}$ and $p_i = \sqrt{6v_i}$.

We used the RBF–FD method for solving this model with various values of N and $\left|\mathcal{I}(\mathbf{x}_j)\right|$ $(j = 1, \dots, N)$ on region $\Omega = [0, 120]^2$. We compare the introduced numerical results in [17] with the RBF–FD method for various values of N and $\left|\mathcal{I}(\mathbf{x}_j)\right|$ $(j = 1, \dots, N)$ at $T = 2, 10$ for Instance 4 in Table 4. From Table 4, it can be deduced that the same results of [17] can be achieved by selecting fewer points. Figure 8 demonstrates the approximate solutions obtained from the RBF-FD method at different final times.

5.5 Instance 5

We consider the 3-D RLW equation as

$$\begin{aligned} &u_t - \Delta u_t + \nabla . u + u . \nabla u = f(x, y, z, t), \\ &\mathbf{x} = (x, y, z) \in [-1, 1]^3, \quad 0 < t < T, \end{aligned} \tag{5.1}$$

which has exact solution

$$u(x, y, z, t) = exp(-t)\ sin(2\pi x)\ sin(2\pi y)\ \sin(2\pi z). \tag{5.2}$$

By substituting solution (5.2) into Eq. (5.1), we can obtain the source function $f(x, y, z, t)$.

Using the RBF method, we obtained approximate solutions for the model mentioned. Figure 9 demonstrates the graphs of approximate solution and its absolute error with

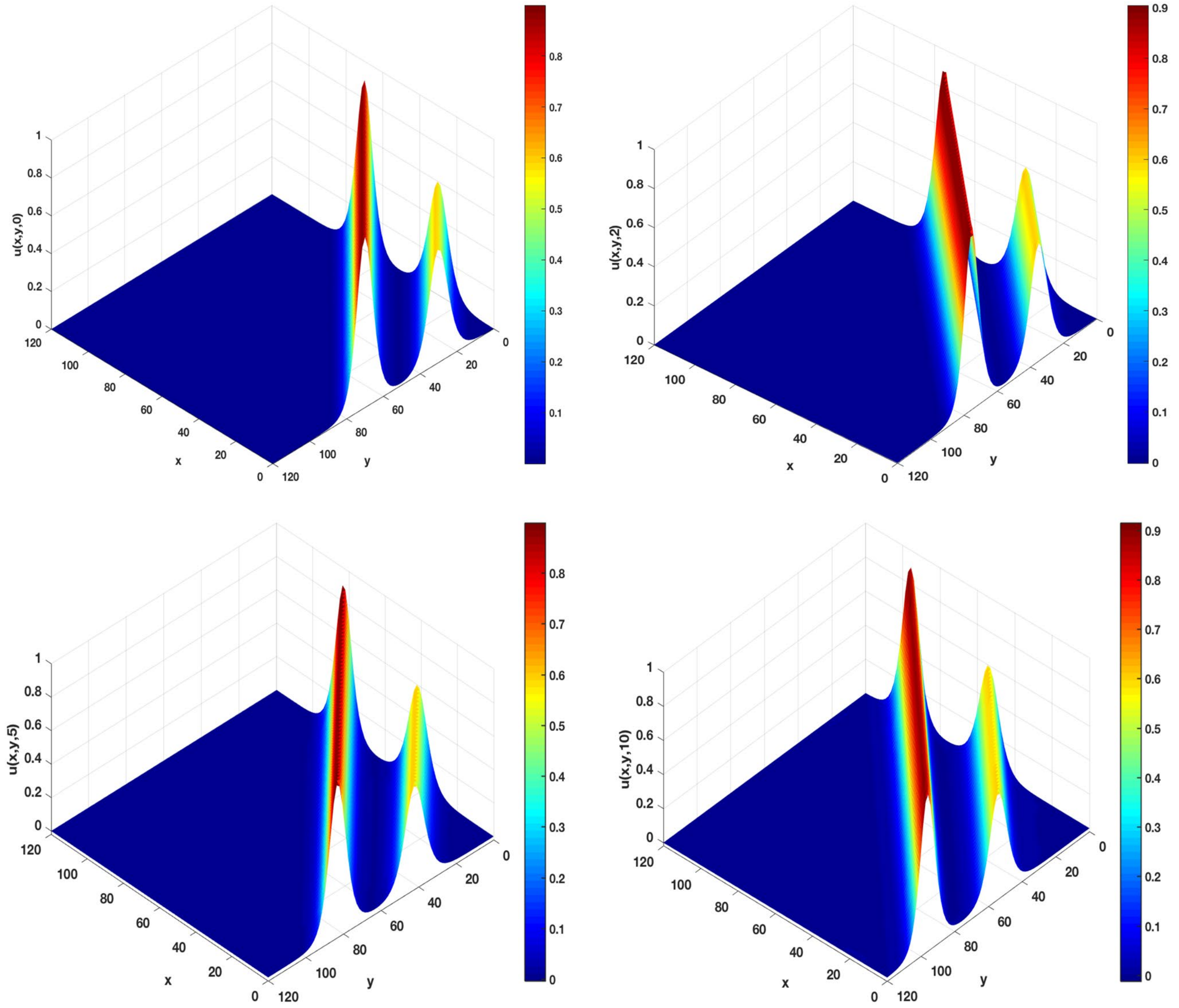


Fig. 8 Graphs of the approximate solutions at several final times with $N = 80$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 41$ $(j = 1, \ldots, N)$ and $\mathrm{d}t = 0.01$ for Instance 4

Table 4 Comparison between the L_∞ error norms of the RBF–FD method and method of [17] with $v_1 = 2.04$ and $v_2 = 2.06$ for Instance 4

T	Method of [17], dt = 0.1			RBF–FD method, dt = 0.01	
	$N = 540$	$N = 1080$	$N = 2160$	$(N, \left|\mathcal{I}(\mathbf{x}_j)\right|) = (80, 41)$	$(N, \left|\mathcal{I}(\mathbf{x}_j)\right|) = (90, 71)$
2	1.0284×10^{-2}	5.1198×10^{-3}	1.8342×10^{-3}	5.5684×10^{-4}	5.2737×10^{-4}
10	5.5711×10^{-2}	3.2873×10^{-3}	6.6394×10^{-3}	1.3871×10^{-3}	1.2591×10^{-3}

$\mathrm{d}t = 10^{-4}$, $N = 15$ and $\left|\mathcal{I}(\mathbf{x}_j)\right| = 11$ $(j = 1, \ldots, N)$ at final time $T = 1$.

5.6 Instance 6

In this problem, the 1-D EFK equation (4.1) is considered with the initial condition [13, 50, 51]

$$u(x, 0) = -\sin(\pi x), \quad x \in [-4, 4],$$

and the boundary conditions

$$u(-4, t) = u(4, t) = 0, \quad t \in [0, T].$$

We found the approximate solution of Eq. (4.1) using RBF–FD method for various values of γ. The numerical results at different final time $(T = 0, 0.05, 0.1, 0.15, 0.2)$ for

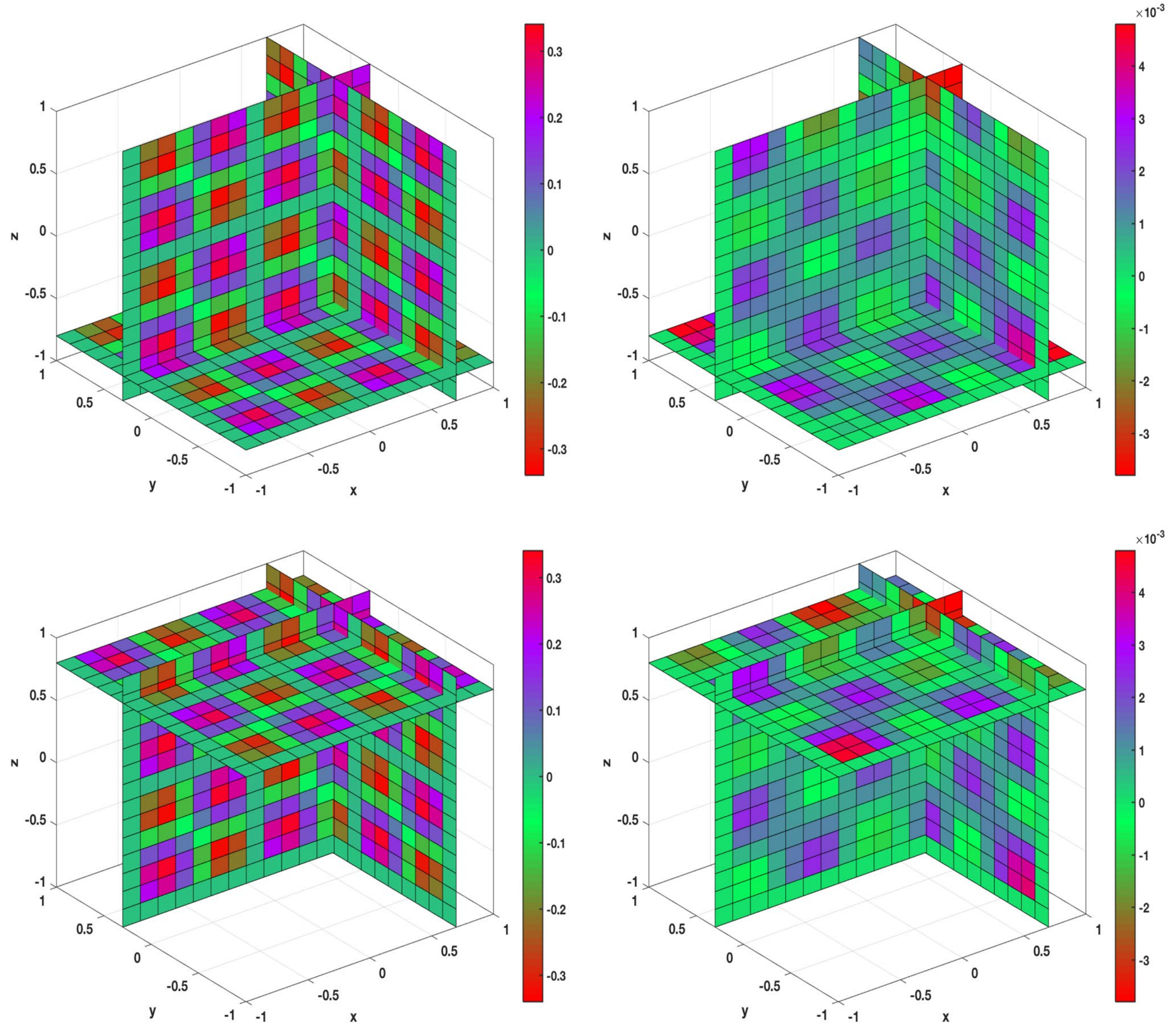


Fig. 9 Graphs of approximation solution and its absolute error applying the RBF–FD method with $N = 15$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 11$ and $\mathrm{d}t = 10^{-4}$ on region $[-1, 1]^3$ for Instance 5

$\gamma = 0, 10^{-4}, 10^{-1}$ are shown in Figs. 10 and 11. It is clear that the behavior of the solutions for $\gamma = 0, 10^{-4}$ is the same, while the numerical solution related to $\gamma = 10^{-1}$ decays fast to zero. Hence, the effect of γ is defined as the stabilization parameter for the EFK equation. These numerical results are consistent with the results obtained in [13, 50, 51].

5.7 Instance 7

We consider the two-dimensional EFK model as

$$u_t + \gamma\Delta^2 u - \Delta u + g(u) = f(x, y, t), \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in [0, T], \tag{5.3}$$

with

$$f(x, y, t) = \exp(-\mathrm{t}) \sin(2\pi \mathrm{x}) \sin(2\pi \mathrm{y})\big(64\gamma\pi^4 + 8\pi^2 \\ + \exp(-2\mathrm{t}) \sin(2\pi \mathrm{x})^2 \sin(2\pi \mathrm{y})^2 - 2\big).$$

The exact solution of Eq. (5.3) is [40]

$$u(x, y, t) = \exp(-t) \sin(2\pi x) \sin(2\pi y). \tag{5.4}$$

The initial condition can be obtained from (5.4) by substituting $t = 0$ and the boundary conditions for this problem are considered as follows:

$$u = 0, \quad \Delta u = 0.$$

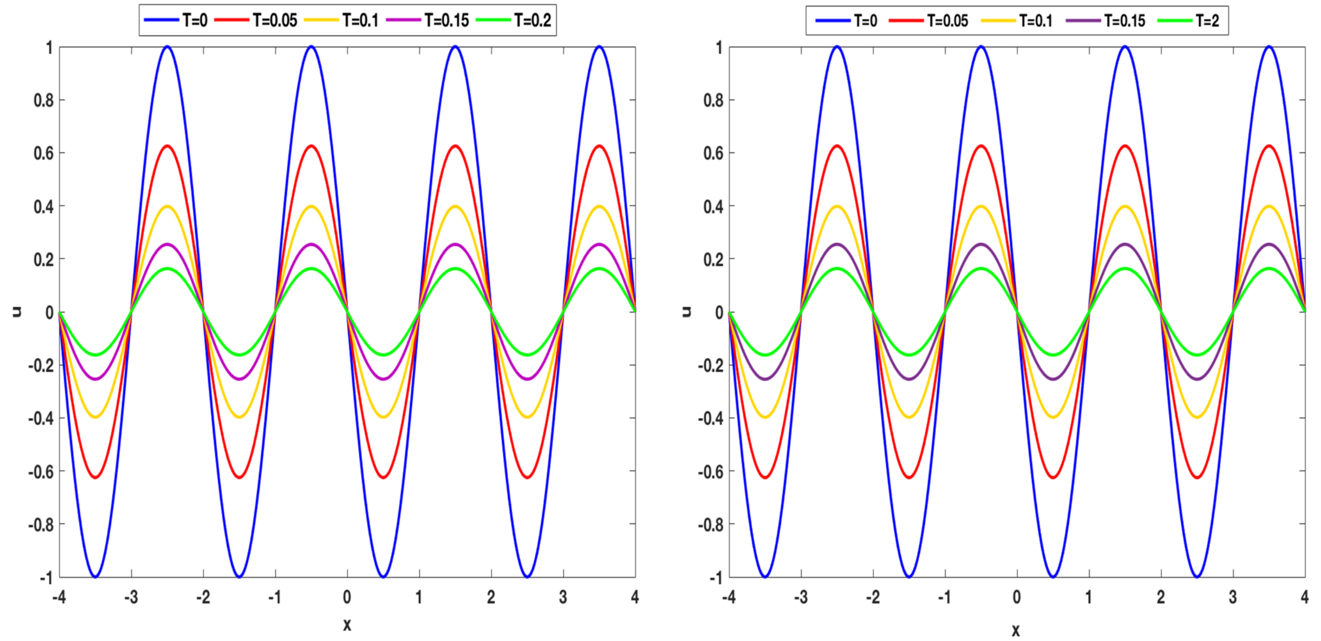


Fig. 10 Graphs of numerical solution with $\mathrm{d}t = 0.01$, $N = 400$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 55$ and $\gamma = 10^{-4}$ (right graph) and $\gamma = 0$ (left graph) on region $[-4, 4]$ at different final times for Instance 6

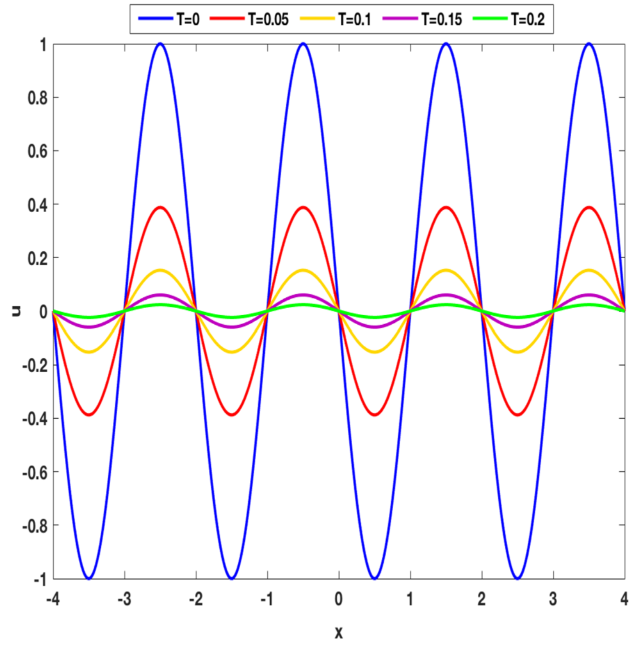


Fig. 11 Graphs of numerical solution with $\mathrm{d}t = 0.01$, $N = 400$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 55$ and $\gamma = 0.1$ on region $[-4, 4]$ at different final times for Instance 6

We solved this model with parameter $\gamma = 0.01$ using RBF–FD technique. In Table 5, the L_∞ error and the CPU time used at final time $T = 1$ are reported with considering $\mathrm{d}t = 0.1$. Moreover, the plot of numerical solution and its contour with $N = 40$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 25$ $(j = 1, \dots, N)$ at $T = 1$ is shown in Fig. 12.

5.8 Instance 8

We consider the two-dimensional EFK equation (5.3) which has the following exact solution [40]:

$$u(x, y, t) = \exp(-t)\exp\left(-\frac{(x-0.5)^2}{\eta} - \frac{(y-0.5)^2}{\eta}\right). \qquad (5.5)$$

The source term $f(x, y, t)$ is obtained by substituting exact solution (5.5) into model (5.3). Moreover, the initial and boundary conditions are acquired from exact solution (5.5).

This problem is investigated with $\gamma = \eta = 0.01$ by RBF–FD technique. The graphs of numerical solution are presented in Fig. 13. We compare the introduced numerical results in [40] with the RBF–FD method for various values of N and $\left|\mathcal{I}(\mathbf{x}_j)\right|$ $(j = 1, \dots, N)$ at $T = 1$ for Instance 8 in Table 6.

5.9 Instance 9

In this example, the three-dimensional case of model (4.6) is considered which has the exact solution:

$$u(x, y, z, t) = \exp(-t)\sin(2\pi x)\sin(2\pi y)\sin(2\pi z). \qquad (5.6)$$

Table 5 Comparison between L_∞-error and CPU Time earned by techniques introduced in [40] and RBF–FD method for Instance 7

N	$\left\|\mathcal{I}(\mathbf{x}_j)\right\|$	DLBIE method of [40]		LBIE-MK method of [40]		RBF–FD method	
		$\mathrm{d}t = 10^{-3}$		$\mathrm{d}t = 10^{-3}$		$\mathrm{d}t = 10^{-1}$	
		L_∞	CPU time	L_∞	CPU time	L_∞	CPU time
10	3	6.0212×10^{-3}	0.12	3.1204×10^{-3}	3.61	2.9918×10^{-4}	0.24
20	7	1.3589×10^{-3}	2.38	7.5035×10^{-4}	20.37	3.1051×10^{-5}	0.94
30	11	–	–	–	–	1.1755×10^{-5}	8.02
40	11	2.3295×10^{-4}	35.95	2.0228×10^{-4}	121.53	3.0037×10^{-6}	15.84
50	11	–	–	–	–	2.0485×10^{-6}	62.20
60	11	–	–	–	–	1.9801×10^{-6}	167.97

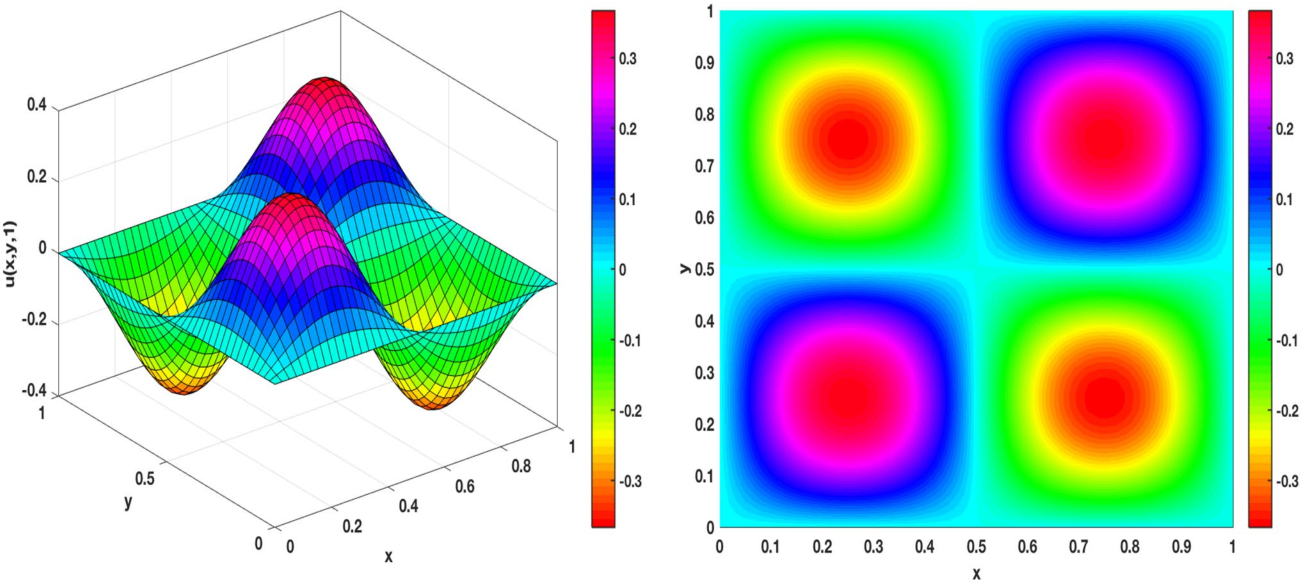


Fig. 12 Graphs of approximation solution and its contour using the RBF–FD method with $N = 40$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 25$ and $\mathrm{d}t = 0.1$ on region $[0,1] \times [0,1]$ for Instance 7

Table 6 Comparison between L_∞ error and CPU time earned by techniques introduced in [40] and RBF–FD method for Instance 8

N	$\left\|\mathcal{I}(\mathbf{x}_j)\right\|$	DLBIE method of [40]		LBIE-MK method of [40]		RBF–FD method	
		$\mathrm{d}t = 10^{-3}$		$\mathrm{d}t = 10^{-3}$		$\mathrm{d}t = 10^{-1}$	
		L_∞	CPU time	L_∞	CPU time	L_∞	CPU time
10	3	5.4336×10^{-1}	0.31	5.4469×10^{-1}	3.81	5.3902×10^{-1}	0.35
20	7	5.3776×10^{-3}	2.88	8.1634×10^{-4}	21.07	7.5489×10^{-4}	3.88
30	11	–	–	–	–	2.4014×10^{-4}	8.79
40	15	1.9977×10^{-3}	40.54	3.6261×10^{-4}	120.30	4.6914×10^{-5}	54.35
50	15	–	–	–	–	2.5926×10^{-6}	211.92
60	15	–	–	–	–	1.2951×10^{-6}	716.13

The initial and boundary conditions are acquired from the exact solution (5.6).

This instance in 3-D case is solved by RBF–FD method with $\gamma = 0.01$, $N = 14$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 7$ and $\mathrm{d}t = 0.001$ at $T = 1$.

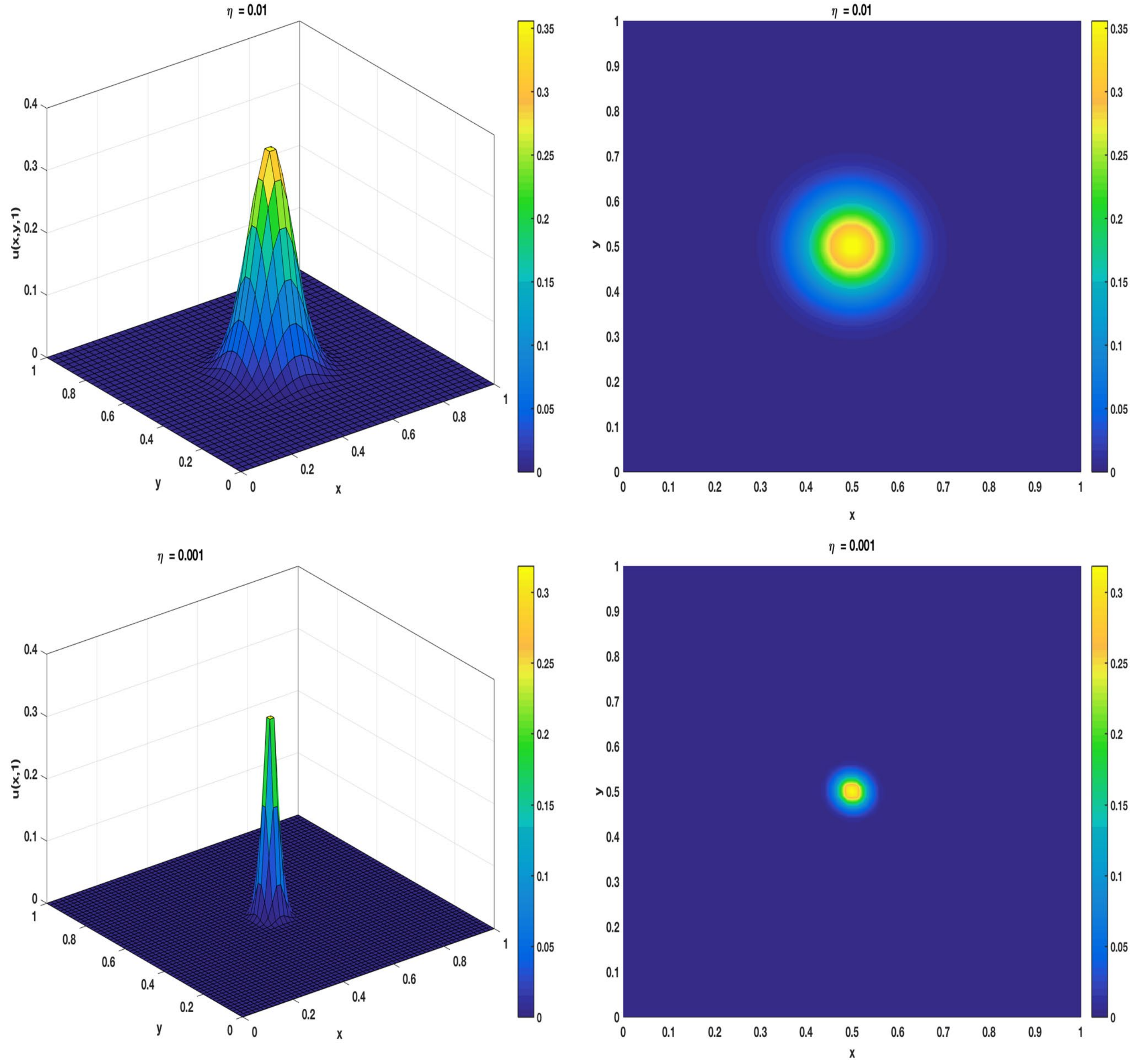


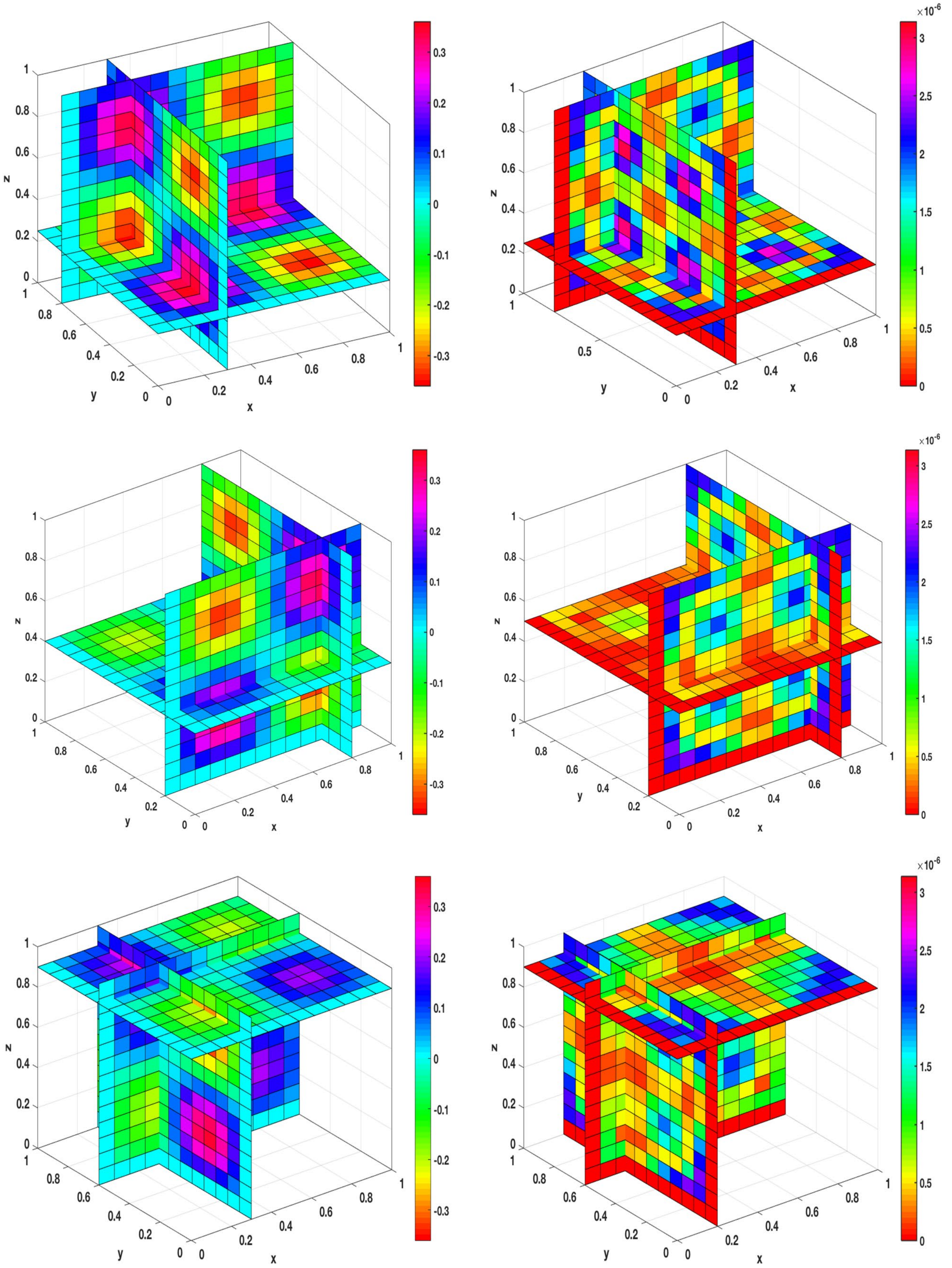
Fig. 13 Graphs of approximation solution and its contour using the RBF–FD method with $N = 40$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 25$ and $\mathrm{d}t = 0.1$ on region $[0, 1] \times [0, 1]$ for Instance 8

Figure 14 demonstrates various slices of the approximate solution and its absolute error.

6 Conclusion

This paper is devoted to solving two nonlinear equations using the RBF–FD method, which is a combination of meshless idea and finite-difference method. Unlike the finite-difference method, this method does not require any mesh generating. Rather, it is only necessary to consider a set of scattered points over the computational region of the problem. Furthermore, each differential operator of problem at a point is approximated by the values of the unknown function

Fig. 14 Graphs of approximation solution (left graph) and absolute▸ error (right graph) at $T = 1$ employing the RBF–FD method with $N = 14$, $\left|\mathcal{I}(\mathbf{x}_j)\right| = 7$ and $\mathrm{d}t = 0.001$ on region $[0, 1] \times [0, 1] \times [0, 1]$ for Instance 9


z
1
0.8
0.6
0.4
0.2
0
y
1
0.8
0.6
0.4
0.2
0
x
0
0.2
0.4
0.6
0.8
1
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
×10^-6
3
2.5
2
1.5
1
0.5
0
0.5

in the points that are in stencil of considered point. This procedure is performed for all points within the computational region. This method is examined on the one-, two-, and three-dimensional EFK equations. According to numerical results obtained for EFK equation, this method can achieve high-order accurate results compared to the recent methods. On the other hand, this method is employed to approximate the solitary waves of the one- and two-dimensional RLW equations. Numerical results show that this method can well simulate the behavior of a single solitary wave and interaction two solitary waves.

Acknowledgements The authors would like to acknowledge one of the referee for his (her) valuable comment.

References

1. Adormian G (1995) Fisher–Kolmogorov equation. Appl Math Lett 8:51–52
2. Aronson DG, Weinberger HF (1978) Multidimensional nonlinear diffusion arising in population genetics. Adv Math 30:33–67
3. Bayona V, Moscoso M, Kindelan M (2011) Optimal constant shape parameter for multiquadric based RBF-FD methods. J Comput Phys 230:7384–7399
4. Bayona V, Moscoso M, Kindelan M (2012) Optimal variable shape parameter for multiquadric based RBF-FD methods. J Comput Phys 231:2466–2481
5. Bayona V, Flyer N, Fornberg B (2019) On the role of polynomials in RBF-FD approximations: III. Behavior near domain boundaries. J Comput Phys 380:378–399
6. Benjamin TB, Bona JL, Mahony JJ (1972) Model equations for long waves in nonlinear dispersive systems. Philos Trans R Soc Lond Ser A 272:47–78
7. Bolling EF, Flyer N, Erlebacher G (2012) Solution to PDEs using radial basis function finite difference (RBF-FD) on multiple GPUs. J Comput Phys 231:7133–7151
8. Chandhini G, Sanyasiraju YVSS (2007) Local RBF-FD solutions for steady convection-diffusion problems. Int J Numer Methods Eng 72:352–378
9. Coullet P, Elphick C, Repaux D (1987) Nature of spatial chaos. Phys Rev Lett 58:431–434
10. Dag I (2000) Least squares quadratic B-spline finite element method for the regularized long wave equation. Comput Methods Appl Mech Eng 182:205–215
11. Dag I, Naci Özer M (2001) Approximation of the RLW equation by the least square cubic B-spline finite element method. Appl Math Model 25:221–231
12. Dag I, Saka B, Irk D (2004) Application of cubic B-splines for numerical solution of the RLW equation. Appl Math Comput 159:373–389
13. Danumjaya P, Pani AK (2005) Orthogonal cubic spline collocation method for the extended Fisher–Kolmogorov equation. J Comput Appl Math 174:101–117
14. Danumjaya P, Pani AK (2006) Numerical methods for the extended Fisher–Kolmogorov (EFK) equation. Int J Numer Anal Model 3:186–210
15. Dee GT, Saarloos WV (1988) Bistable systems with propagating fronts leading to pattern formation. Phys Rev Lett 60:2641–2644
16. Dehghan M, Salehi R (2011) The solitary wave solution of the two-dimensional regularized long-wave equation in fluids and plasmas. Comput Phys Commun 182:2540–2549
17. Dehghan M, Abbaszadeh M (2015) The use of interpolating element-free Galerkin technique for solving 2D generalized Benjamin–Bona–Mahony–Burgers and regularized long-wave equations on non-rectangular domains with error estimate. J Comput Appl Math 286:211–231
18. Dehghan M, Abbaszadeh M (2017) The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations. J Comput Phys 351:478–510
19. Dehghan M, Shokri A (2007) A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. Comput Math Appl 54:136–146
20. Dehghan M (2006) Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices. Math Comput Simul 71:16–30
21. Dehghan M (2005) On the solution of an initial-boundary value problem that combines Neumann and integral condition for the wave equation. Numer Methods Partial Differ Equ 21(1):24–40
22. Djidjeli K, Price WG, Twizell EH, Cao Q (2003) A linearized implicit pseudo-spectral method for some model equations the regularized long wave equations. Commun Numer Methods Eng 19:847–863
23. Driscol TA, Fornberg B (2002) Interpolation in the limit of increasingly flat radial basis functions. Comput Math Appl 43:413–422
24. Eilbeck JC, McGuire GR (1975) Numerical study of RLW equation I: numerical methods. J Comput Phys 19:43–57
25. Eilbeck JC, McGuire GR (1977) Numerical study of the regularized long-wave equation II: interaction of solitary waves. J Comput Phys 23:63–73
26. Fasshauer GE (2007) Meshfree approximation methods with MATLAB. World Scientific, New York
27. Flyer N, Lehto E, Blaise S, Wright GB, St-Cyr A (2012) A guide to RBF-generated finite differences for nonlinear transport: shallow water simulations on a sphere. J Comput Phys 231:4078–4095
28. Forenberg B, Wright G (2004) Stable computation of multiquadric interpolants for all values of the shape parameter. Comput Math Appl 16:497–523
29. Fornberg B, Lehto E (2011) Stabilization of RBF-generated finite difference methods for convective PDEs. J Comput Phys 230:2270–2285
30. Fornberg B, Wrigth G, Larsson E (2004) Some observations regarding interpolants in the limit of flat radial basis functions. Comput Math Appl 47:37–55
31. Ghiloufi A, Rouatbi A, Omrani K (2018) A new conservative fourth-order accurate difference scheme for solving a model of nonlinear dispersive equations. Math Methods Appl Sci 41:5230–5253
32. Guo BY, Cao WM (1988) The Fourier pseudospectral method with a restrain operator for the RLW equation. J Comput Phys 74:110–126
33. Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. J Geophys Res 76:1905–1915
34. Hardy RL (1975) Research results in the application of multiquadric equations to surveying and mapping problems. Surv Map 35:321–332
35. Helal MA (2002) Soliton solution of some nonlinear partial differential equations and its applications in fluid mechanics. Chaos Solitons Fractals 13(9):1917–1929
36. Hon YC, Mao XZ (1998) An efficient numerical scheme for Burgers equation. Appl Math Comput 95:37–50
37. Hon YC, Cheung KF, Mao XZ, Kansa EJ (1999) Multiquadric solution for shallow water equations. J Hydraul Eng 125:524–533
38. Hon YC, Mao XZ (1999) A radial basis function method for solving options pricing model. Financ Eng 8:31–49

39. Hornreich RM, Łuban M, Shtrikman S (1975) Critical behaviour at the onset of $\bar{k}$-space instability at the λ line. Phys Rev Lett 35:1678–1681
40. Ilati M, Dehghan M (2017) Direct local boundary integral equation method for numerical solution of extended Fisher–Kolmogorov equation. Eng Comput 34:203–213
41. Javed A, Djijdeli K, Xing JT (2014) Shape adaptive RBF-FD implicit scheme for incompressible viscous Navier–Stokes equations. Comput Fluids 89:38–52
42. Kadri T, Omrani K (2011) A second-order accurate difference scheme for an extended Fisher–Kolmogorov equation. Comput Math Appl 61:451–459
43. Kadri T, Omrani K (2018) A fourth-order accurate finite difference scheme for the extended Fisher–Kolmogorov equation. Bull Korean Math Soc 55:297–310
44. Kansa EJ (1990) Multiquadrics-A scattered data approximation scheme with applications to computational fluid dynamics- I. Surface approximations and partial derivative estimates. Comput Math Appl 9:127–145
45. Kansa EJ (1990) Multiquadrics-A scattered data approximation scheme with applications to computational fluid dynamics- II. Solutions to parabolic hyperbolic and elliptic partial differential equations. Comput Math Appl 19:147–161
46. Khiari N, Omrani K (2011) Finite difference discretization of the extended Fisher–Kolmogorov equation in two dimensions. Comput Math Appl 62:4151–4160
47. Lam L (2003) Introduction to nonlinear physics. Springer, New York
48. Liu GR, Gu YT (2005) An introduction to mesh free methods and their programming. Springer, Dordrecht, Berlin, Heidelberg, New York
49. Marcozzi M, Choi S, Chen CS (2001) On the use of boundary conditions for variational formulations arising in financial mathematics. Appl Math Comput 124:197–214
50. Mittal RC, Arora G (2010) Quintic B-spline collocation method for numerical solution of the extended Fisher–Kolmogorov equation. Int J Appl Math Mech 6:74–85
51. Mittal RC, Dahiya S (2016) A study of quintic B-spline based differential quadrature method for a class of semi-linear Fisher–Kolmogorov equations. Alexandria Eng J 55:2893–2899
52. Omrani K (2006) The convergence of fully discrete Galerkin approximations for the Benjamin–Bona–Mahony (BBM) equation. Appl Math Comput 180:614–621
53. Peletier LA, Troy WC (1995) A topological shooting method and the existence of kinks of the extended Fisher–Kolmogorov equation. Topol Methods Nonlinear Anal 6:331–355
54. Peregrine DH (1966) Calculations of the development of an undular bore. J Fluid Mech 25:321–330
55. Petras A, Ling L, Piret C, Ruuth SJ (2019) A least-squares implicit RBF-FD closest point method and applications to PDEs on moving surfaces. J Comput Phys 381:146–161
56. Rippa S (1999) An algorithm for selecting a good value for the parameter c in radial basis function interpolation. Adv Comput Math 11:193–210
57. Roque CMC, Cunha D, Shu C, Ferreira AJM (2011) A local radial basis functions—Finite differences technique for the analysis of composite plates. Eng Anal Bound Elem 35(3):363–374
58. Roshan T (2012) A Petrov–Galerkin method for solving the generalized regularized long wave (GRLW) equation. Comput Math Appl 63:943–956
59. Rouatbi A, Omrani K (2017) Two conservative difference schemes for a model of nonlinear dispersive equations. Chaos Solit Fractal 104:516–530
60. Rouatbi A, Achouri T, Omrani K (2018) High-order conservative difference scheme for a model of nonlinear dispersive equations. Comput Appl Math 37:4169–4195
61. Sarra SA (2012) A local radial basis function method for advection–diffusion–reaction equations on complexly shaped domains. Appl Math Comput 218:9853–9865
62. Shang Y, Niu P (1988) Explicit exact solutions for the RLW equation and the SRLW equation in two space dimensions. Math Appl 11:1–5
63. Shan YY, Shu CW, Qin N (2009) Multiquadric finite difference (MQ-FD) methods and its application. Adv Appl Math Mech 1:615–638
64. Shokri A, Dehghan M (2010) A meshless method using the radial basis functions for numerical solution of the regularized long wave equation. Numer Methods Partial Differ Equ 26:807–825
65. Shu CW, Ding H, Zhao N, Cao Q (2006) Numerical comparison of least square-based finite-difference (LSFD) and radial basis function-based finite-difference (RBF-FD) methods. Comput Math Appl 51:1297–1310
66. Shu CW, Ding H, Yeo KS (2003) Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 192:941–954
67. Tillenius M, Larsson E, Lehto E (2015) A scalable RBF-FD method for atmospheric flow. J Comput Phys 298:406–422
68. Tolstykh AI (2000) On using RBF-based differencing formulas for unstructured and mixed structured–unstructured grid calculation. In: Proceeding of the 16th IMACS, World Congress, Lausanne
69. Tolstykh AI, Shirobokov DA (2003) On using basis functions in a finite difference mode with applications to elasticity problems. Comput Mech 33:68–79
70. Wendland H (2005) Scattered data approximation, Cambridge monograph on applied and computational mathematics. Cambridge University Press, England
71. Wright GB, Fornberg B (2006) Scattered node compact finite difference-type formulas generated from radial basis functions. J Comput Phys 212:99–123
72. Zerroukat M, Power H, Chen CS (1992) A numerical method for heat transfer problem using collocation and radial basis functions. Int J Numer Methods Eng 42:1263–1278
73. Zheng-hong H (2002) On Cauchy problems for the RLW equation in two space dimensions. Appl Math Mech 23:169–177
74. Zhu G (1982) Experiments on director waves in nematic liquid crystals. Phys Rev Lett 49:1332–1335