Received: January 7, 2025; **Accepted:** June 3, 2025; **Published:** June 3, 2025.
DOI. 10.30473/coam.2025.73348.1281
Winter-Spring (2025) Vol. 10, No. 1, (193-215)

Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

Mesh-Free RBF-FD Method with Polyharmonic Splines and Polynomials for High-Dimensional PDEs and Financial Option Pricing

Narges Hosseinzadeh, Elyas Shivanian ✉, Saeid Abbasbandy

Department of Applied Mathematics, Imam Khomeini International University, Qazvin, Iran.

✉ **Correspondence:**
Elyas Shivanian
E-mail:
shivanian@sci.ikiu.ac.ir

How to Cite
Hosseinzadeh, N., Shivanian, E., Abbasbandy, S. (2025). "Mesh-free RBF-FD method with polyharmonic splines and polynomials for high-dimensional PDEs and financial option pricing", Control and Optimization in Applied Mathematics, 10(1): 193-215, doi: 10.30473/coam.2025.73348.1281

Abstract. This study employs the radial basis function-generated finite difference (RBF-FD) method to address high-dimensional elliptic differential equations under Dirichlet boundary conditions. The method utilizes polyharmonic spline functions (PHSs) combined with polynomials for approximation. A notable benefit of this approach is that PHSs do not require a shape parameter, simplifying implementation and enhancing numerical stability. The proposed method offers several advantages, including high accuracy, rapid computation, and adaptability to complex geometries and irregular node arrangements. It is particularly effective for high-dimensional problems, providing a mesh-free alternative that scales efficiently with increased complexity. Beyond scientific computing, the method is also applied to financial option pricing, where integro-differential equations are transformed into a series of second-order elliptic partial differential equations (PDEs). Numerical experiments demonstrate that the proposed algorithm significantly outperforms existing RBF-based approaches in both accuracy and efficiency. These strengths make it a robust tool for solving a wide range of PDEs in both regular and irregular domains.

Keywords. Radial basis functions, Polyharmonic splines, High-dimensional partial differential equations, Finite difference method, Option pricing.

MSC. 65M06;91G60.

©2025 by the authors. Lisensee PNU, Tehran, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY4.0) (http:/creativecommons.org/licenses/by/4.0)

1 Introduction

Partial differential equations are extensively used in various scientific fields, including physics, biology, and economics. While analytical solutions are available for simpler problems that arise in straightforward geometric domains, many complex issues do not have closed-form solutions. Consequently, researchers have concentrated on developing numerical methods to approximate solutions to partial differential equations (PDEs). The field of numerical analysis has made significant advancements in this area, leading to the creation of various techniques, including a recent approach that utilizes radial basis functions. This method approximates the weights of finite differences for local domains, making it effective with scattered nodes and adaptable to any desired spatial dimension [1, 2, 3, 4, 24]. The radial basis function-generated finite difference (RBF-FD) method is a versatile technique applicable in various contexts, particularly for addressing large-scale real-world problems. It has been used in numerous scenarios, such as simulating fluid flow and heat transfer in complex geometries, which include aircraft wings and turbine blades. Additionally, the RBF-FD method has been employed to model seismic wave propagation and earthquake dynamics, helping researchers gain a better understanding of the behavior of the Earth's crust.

This technique has proven effective for image denoising, providing precise and efficient solutions for image analysis. It has also been applied in financial modeling to predict stock prices, analyze market trends, and optimize investment strategies. The RBF-FD method is particularly advantageous in scenarios involving higher dimensions, complex geometries, and irregular or unstructured node distributions. The use of RBFs in these techniques facilitates the interpolation and approximation of scattered data, which is essential for their effectiveness [7, 9, 10, 14, 16, 17, 20, 22].
In conclusion, the RBF-FD method is a valuable tool for approximating the weights of finite differences in local domains. Various RBFs can be incorporated into the framework of this method. Initially, infinitely smooth RBFs were commonly utilized in several studies; however, they have certain limitations, such as the need for symbolic computation to find the optimal shape parameter. Recent research has explored the benefits of using polynomials and polyharmonic splines within RBF-FD methods, effectively overcoming these limitations [8, 11, 18, 19]. For example, RBFs with shape parameters, such as multiquadrics or inverse quadratics, often necessitate expensive and time-consuming processes to determine their optimal values. The proposed method tackles this challenge. Moreover, the combination of RBF-FD methods with characteristics such as sparsity, geometric flexibility, and ease of implementation is preserved [6, 5, 12, 21].

In this research, we propose an improved RBF-FD method that integrates polyharmonic splines with polynomial augmentation to construct accurate local differential operators. The proposed framework eliminates the need for shape parameter optimization, one of the main computational challenges in classical RBF methods [17, 21] thus enhancing numerical stability and simplifying implementation. Moreover, the method is particularly effective in high-dimensional settings and irregular domains, leveraging sparsity and local support properties [18, 22]. It is applied to solve partial integro-differential equations (PIDEs) arising in financial option pricing under jump-diffusion models [19]. Our numerical results demonstrate superior accuracy, stability, and computational efficiency compared to existing RBF-based schemes [23]. These advancements collectively represent a significant contribution to meshfree numerical techniques for solving complex scientific and financial problems.

Recent advancements have demonstrated the versatility of local meshless methods, particularly RBF-FD schemes, in solving highly nonlinear and complex PDEs across various scientific domains. For instance, innovative applications of localized RBF-FD approaches to heat transfer, structural mechanics, and multi-physics problems have been reported in the latest literature. These studies not only improve numerical stability but also offer enhanced accuracy for irregular domains and discontinuous coefficients. Our present work builds upon these ideas by extending the meshless RBF-FD technique to financial models involving jump-diffusion dynamics. Recent studies have further advanced meshless methods by integrating local adaptive refinement strategies, improved stabilization techniques, and hybrid approaches combining RBFs with finite elements. For instance, localized RBF-FD schemes have demonstrated remarkable performance in handling highly nonlinear, time-dependent PDEs and multiphysics problems [13, 15, 25, 26]. These developments highlight the increasing relevance of mesh-free methods in real-world applications, particularly in computational finance, where models often feature complex stochastic dynamics, jumps, and discontinuities.

Our present contribution fits within this growing body of work by proposing a robust, stable, and efficient RBF-FD framework that specifically addresses the challenges of high-dimensional financial models with jump-diffusion dynamics. By eliminating the need for shape parameter optimization and leveraging polyharmonic splines, our method achieves both high accuracy and numerical stability without sacrificing computational efficiency.

The structure of the paper is organized as follows. Section 2 introduces the fundamental concepts of the RBF-FD method and polyharmonic splines. Section 3 formulates the mathematical model for European option pricing under jump-diffusion dynamics. Section 4 presents the discretization scheme using an implicit-explicit strategy. Section 5 validates the proposed approach through various numerical experiments, and Section 6 concludes with a summary of the results and potential avenues for future research.

2 Some Basic RBF Relations

Radial functions are defined as follows:

Definition 1. A function $\Theta : \mathbb{R}^n \to \mathbb{R}$ is called a radial function if there exists a function $\theta : [0, \infty) \to \mathbb{R}$ such that $\Theta(x) = \theta(\|x\|)$.

Radial functions can be classified into two main categories: infinitely smooth functions and piecewise smooth functions, some common RBFs are presented in Table 1. Although various norms can be employed, this study used the standard Euclidean norm.

Consider a set of distinct points $\{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^n$ and a RBF θ, the associated radial function θ, generates functions $\theta_i : \mathbb{R}^n \to \mathbb{R}$, defined as $\theta_i(x) = \Theta(x, x_i) = \theta(\|x - x_i\|)$ for $i = 1, 2, \cdots, N$. This $\theta_i : \mathbb{R}^n \to \mathbb{R}$ exhibit radial symmetry around point x_i and are linearly independent over $\mathbb{R}^n$, thus forming a basis for a space and earning the name RBFs. Identifying interpolants within this space is known as RBF interpolation. Given a set of function values $\{f_1, f_2, \cdots, f_N\} \subset \mathbb{R}$ corresponding to the points in $\{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^n$, we examine interpolants of the following structure:

$$P_f(x) = \sum_{i=1}^{N} c_i \theta(\|x - x_i\|), \tag{1}$$

for some RBF θ, and some set of weights $\{c_1, c_2, \cdots, c_N\} \subset \mathbb{R}$. For a given θ, to find P_f we just need to find the weights $\{c_1, c_2, \cdots, c_N\}$, that satisfy the linear system (2).

$$\begin{pmatrix} \theta(\|x_1 - x_1\|) & \theta(\|x_1 - x_2\|) & \cdots & \theta(\|x_1 - x_N\|) \\ \theta(\|x_2 - x_1\|) & \theta(\|x_2 - x_2\|) & \cdots & \theta(\|x_2 - x_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \theta(\|x_N - x_1\|) & \theta(\|x_N - x_2\|) & \cdots & \theta(\|x_N - x_N\|) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}. \tag{2}$$

Table 1: Some common RBFs

Infinitely Smooth		
Multiquadric	(MQ)	$\theta(r) = \sqrt{1 + (\epsilon r)^2}$
Gaussian	(GA)	$\theta(r) = e^{-(\epsilon r)^2}$
Inverse Multiquadric	(IMQ)	$\theta(r) = \left(\sqrt{1 + (\epsilon r)^2}\right)^{-1}$
Inverse Quadratic	(IQ)	$\theta(r) = \left(1 + (\epsilon r)^2\right)^{-1}$
Polyharmonic Splines		
Thin-Plate Spline	(TS)	$\theta(r) = r^2 \log r$
Natural Cubic Spline	(NCS)	$\theta(r) = r^3$
Even PHS	$(EPHS)$	$\theta(r) = r^{2l} \log r$
Odd PHS	$(OPHS)$	$\theta(r) = r^{2l+1}$

2.1 Polyharmonic Spline (PHSs)

Our main goal is to use RBFs to approximate local differential operators instead of integral operators, which tend to be global. The performance of local RBF interpolation has improved, making it well-suited for our needs. However, this improvement comes at the expense of sacrificing spectral convergence rates. Therefore, it is sensible to explore alternatives, such as polyharmonic splines, since spectral convergence rates are a key reason for choosing RBFs that are infinitely smooth. PHSs are defined by

$$\theta(r) = r^{2l+1}, \qquad \text{or} \qquad \theta(r) = r^{2l}\log r, \qquad l \in \mathbb{N}, \tag{3}$$

the exponent represents the degree of the polyharmonic splines. In contrast to infinitely smooth RBFs, PHSs only achieve polynomial convergence rates, even for global interpolation, and do not require a shape parameter. Additionally, PHSs do not guarantee non-singular interpolation matrices; instead, they are conditionally positive definite.

Definition 2. A real-valued continuous even function f is called a conditionally positive definite of order s on $\mathbb{R}^n$ if

$$\sum_{j=1}^{N}\sum_{k=1}^{N} c_j c_k f(x_j - x_k) \geq 0, \tag{4}$$

For any distinct points $\{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^n$ and nonzero $c \subset \mathbb{R}^N$ satisfying

$$\sum_{j=1}^{N} c_j p(x_j) = 0,$$

for all polynomials p of degree less than or equal to $s-1$.

The PHSs $\theta(r) = r^{2l+1}$ and $\theta(r) = r^{2l}\log r$ are conditionally positive definite (CPD) of order $l+1$. A fundamental property of CPD RBFs of this order is that their interpolation matrices will be nonsingular when a basis of polynomials up to degree l in incorporated into the interpolant. This holds provided the polynomials are not linearly dependent on chosen interpolation points and the necessary additional conditions are satisfied.

2.2 Radial Basis Function Finite Differences

Consider a spatial domain $\Omega \subset \mathbb{R}$ and set of distinct RBF collocation points $A = \{x_1, x_2, \ldots, x_n\}$ in Ω. Let $\{x_1, x_2, \ldots, x_N\} \subset A$ denote a subset that includes point x_e along with its $N-1$ nearest neighbors, thereby forming a stencil centered at x_e, where $N \ll n$. In the RBF-FD approach, any differential operator L acts on function $u(x)$ evaluated at point x_j. This operator is approximated using a linear weighted combination of the function values of u at points in the stencil. Specifically, this approximation can be expressed as:

$$Lu(x_e) \approx \sum_{i=1}^{N} w_i u(x_i), \tag{5}$$

where w_i and n are differential wights and stencil sizes respectively. For each node $x(j) \in \Omega$ the weights are computed on each local support. Usually, these nodes are considered equidistant, and their weights are calculated using interpolation polynomials.

Consider $s(x)$ as a RBF interpolant that approximates function $u(x)$ at specific interpolation points in x_i, the representation of $s(x)$ can be expressed as follows:

$$s(x) = \sum_{j=1}^{N} \lambda_j \theta(\|x - x_j\|) + \sum_{j=1}^{M} \alpha_j p_j(x), \tag{6}$$

where $\|.\|$ is the Euclidian norm and $\{p_1, p_2, \ldots, p_M\}$ is a basis of polynomial space up to degree l where M is the dimension of the polynomial space up to degree l, and d is the dimension of space, then $M = \begin{pmatrix} l+d \\ l \end{pmatrix}$. The coefficients λ_j and α_j are evaluated by imposing the following conditions:

$$\begin{aligned} &s(x_i) = u(x_i), \qquad i = 1, 2, \ldots, n, \\ &\sum_{j=1}^{N} \lambda_j p_k(x_j) = 0, \quad k = 1, 2, \ldots, M, \end{aligned} \tag{7}$$

imposing conditions (7) on $s(x)$ gives a linear system:

$$\begin{pmatrix} \Theta & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} u \\ 0 \end{pmatrix}. \tag{8}$$

The components of the above system are as follows:

$$\Theta = \begin{pmatrix} \theta(\|x_1 - x_1\|) & \theta(\|x_1 - x_2\|) & \cdots & \theta(\|x_1 - x_N\|) \\ \theta(\|x_2 - x_1\|) & \theta(\|x_2 - x_2\|) & \cdots & \theta(\|x_2 - x_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \theta(\|x_N - x_1\|) & \theta(\|x_N - x_2\|) & \cdots & \theta(\|x_N - x_N\|) \end{pmatrix}, \tag{9}$$

$$P = \begin{pmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_M(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_M(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_N) & p_2(x_N) & \cdots & p_M(x_N) \end{pmatrix}, \tag{10}$$

$$u = [u_1, u_2, \cdots, u_N]^T, \tag{11}$$

now we proceed to find the unknown weights according to the system (8)

$$\begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} \Theta & P \\ P^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} u \\ 0 \end{pmatrix}, \tag{12}$$

by applying differential operator L to interpolant (6), we obtain

$$Ls(x)|_{x=x_e} = \sum_{i=1}^{n} \lambda_i L\theta(\|x - x_i\|)|_{x=x_e} + \sum_{j=1}^{m} \alpha_j Lp_j(x)|_{x=x_e}, \tag{13}$$

$$= \begin{pmatrix} b & d \end{pmatrix} \begin{pmatrix} c \\ \alpha \end{pmatrix}, \tag{14}$$

$$= \left(\begin{pmatrix} b & d \end{pmatrix} \begin{pmatrix} \Theta & P \\ P^T & 0 \end{pmatrix}^{-1} \right) \begin{pmatrix} u \\ 0 \end{pmatrix}, \tag{15}$$

the row vector resulting from multiplication of the expressions in parentheses is the same vector of weights for the approximation $Lu(x_e) \approx \sum_{i=1}^{N} w_i u(x_i)$, where $w \in \mathbb{R}^{1\times N}$ and $v \in \mathbb{R}^{1\times M}$. It should be noted that the final weights were discarded because they were ultimately multiplied by zero. By transposing, the weights were calculated by solving the following linear system:

$$\begin{pmatrix} \Theta & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} w^T \\ v^T \end{pmatrix} = \begin{pmatrix} b^T \\ d^T \end{pmatrix}, \tag{16}$$

where

$$b = \Big(L\theta(\|x - x_1\|)|_{x=x_e} \quad L\theta(\|x - x_2\|)|_{x=x_e} \quad \cdots \quad L\theta(\|x - x_N\|)|_{x=x_e} \Big)_{1\times N}, \tag{17}$$

$$d = \Big(Lp_1(x)|_{x=x_e} \quad Lp_2(x)|_{x=x_e} \quad \cdots \quad Lp_M(x)|_{x=x_e} \Big)_{1\times M}. \tag{18}$$

The final linear system (10) is significantly smaller than the global RBF collocation system. Any interpolation method can produce finite difference weights for differential operators, and RBF interpolation is no exception. This approach is called RBF-FD.

Algorithm: Addressing a PDE using the RBF-FD Method

1. **Node Generation:** Generate a set of scattered nodes $\{x_i\}_{i=1}^{N}$ over the computational domain Ω.

2. **Stencil Construction:** For each evaluation point x_e, identify a local stencil consisting of its N_e nearest neighbors.

3. **Local Interpolation:** Construct an RBF interpolant augmented with polynomials over each stencil:
$$s(x) = \sum_{j=1}^{N_e} \lambda_j \theta(\|x - x_j\|) + \sum_{k=1}^{M} \alpha_k p_k(x).$$

4. **Weight Computation:** Apply the differential operator $\mathcal{L}$ to $s(x)$ at $x = x_e$ to obtain RBF-FD weights.

5. **Discretization:** Replace the derivatives in the PDE using the computed weights at all nodes.

6. **System Assembly:** Assemble the global sparse linear system from all local discretizations.

7. **Apply Boundary/Initial Conditions:** Incorporate the relevant boundary or initial conditions.
8. **Solve:** Solve the resulting algebraic system for the unknown function values.

The following steps outline the process for solving a PDE using the RBF-FD method:

1. Establish a set of nodes for the PDE.
2. Create a stencil for each evaluation point, consisting of the neighboring nodes.
3. Develop an approximation for each differential operator by linearly combining the values of the unknown functions at the scattered nodes within the stencil.
4. Calculate the weights or differencing coefficients for each stencil.
5. Substitute the approximations from Step 4 for the derivatives at each node in the PDE to form the corresponding final system.
6. Solve the final system to obtain the desired results.

The effectiveness of the RBF-FD methods can be influenced by various factors, including the algorithms used to determine the nodes for each stencil. In this study, we utilized the k-nearest neighbor (KNN) algorithm implemented in MATLAB.

The following items are essential for the efficiency of the proposed method:

* The number of nodes and the stencil size (represented as n in this paper).
* The type of PHS radial function (represented as $\theta(r)$ in this paper).
* The highest polynomial degree included for approximation (represented as HPD in this paper).

We investigated these three aspects in numerical experiments. In this study, we used the root mean square (RMS) error to demonstrate the efficiency of the method.

$$RMS_{error} = \sqrt{\frac{\sum_{i=1}^{N}(u_{ex}(x_i) - u_{ap}(x_i))^2}{N}}, \tag{19}$$

where $u_{ex}(x_i)$ and $u_{ap}(x_i)$ are the exact and approximate solutions respectively on points x_i, and N is the number of total points.

2.3 Theoretical Considerations on Convergence

Although the primary focus of this work is on the practical implementation and numerical performance of the proposed RBF-FD method, it is important to briefly highlight its theoretical convergence foundation. The method leverages polyharmonic spline RBFs augmented with polynomials to construct local differential approximations. This combination has been shown in prior studies to offer both flexibility and accuracy without requiring shape parameter optimization [6, 7, 21].

From a theoretical standpoint, the interpolation matrix arising from the local RBF-FD formulation is guaranteed to be non-singular provided that a complete polynomial basis up to degree ℓ is included and the node distribution within each stencil satisfies mild geometric requirements. This ensures the well-posedness of the local interpolation problem. Furthermore, under these conditions, the method exhibits algebraic convergence rates, where the order of convergence depends on the polynomial degree and the spatial dimension.

These theoretical results, as discussed in [7, 9, 10], establish a solid foundation for the reliability of the RBF-FD method. While we refrain from presenting formal proofs in this paper, our numerical results in Section 5 demonstrate consistent convergence behavior and accuracy improvements as the number of nodes increases, which are in line with the theoretical expectations. This alignment provides additional confidence in the robustness of the proposed approach for solving elliptic and integro-differential equations.

3 The Mathematical Model

This part of the paper offers an extensive exploration of the challenges associated with valuing European options within the framework of jump-diffusion models. We assume that the price dynamics of a stock or an underlying financial asset adhere to the behavior described by an exponential jump-diffusion model. This model is expressed mathematically as:

$$S_t = S_0 e^{rt+X_t}, \tag{20}$$

where S_0 is the initial price of the stock at time $t = 0$, r signifies the risk-free interest rate, and $(X_t)_{t\geq 0}$ is a Levy process incorporating both diffusion and jump components. The Levy process $(X_t)_{t\geq 0}$ is formally defined as follows:

$$X_t := at + \sigma W_t + \sum_{i=1}^{N_t} G_i. \tag{21}$$

In this representation, a and $\sigma > 0$ are constants, with $(W_t)_{t\geq 0}$ denoting a standard Brownian motion and $(N_t)_{t\geq 0}$ representing a Poisson counting process. Additionally, the random variables G_i are independent and identically distributed, following a Gaussian distribution in the context of the Merton jump-diffusion model.

To calculate the price of a European option under these jump-diffusion settings, denoted as $V(S,t)$, one must solve a partial integro-differential equation. This computation is formalized in the theorem presented below:

Theorem 1. Let the Levy process $(X_t)_{t\geq 0}$ be characterized by the Levy triplet (σ^2, γ, ν), where $\sigma > 0$, $\gamma \in \mathbb{R}$, and ν is the Levy measure. If the following condition is satisfied:

$$\sigma > 0 \quad or \quad \exists \beta \in (0,2) \quad s.t \quad \liminf_{\epsilon \to 0} \epsilon^{-\beta} \int_{-\epsilon}^{\epsilon} |x|^2 \nu(dx) > 0. \tag{22}$$

Then the valuation of a European option with a payoff function $Z(S_T)$ is given by $V(S,t)$, where

$$V : [0,\infty) \times [0,T] \to \mathbb{R},$$

$$(S,t) \to V(S,t) = e^{-r(T-t)}\mathbb{E}[Z(S_T) \mid S_t = S],$$

is a function that is continuous on $[0,\infty) \times [0,T]$, twice differentiable with respect to S, and once differentiable with respect to t on the open interval $(0,\infty) \times (0,T)$. This function satisfies the PIDE given below:

$$\begin{aligned} -\frac{\partial V}{\partial t}(S,t) &= \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}(S,t) + rS\frac{\partial V}{\partial S}(S,t) - rV(S,t) \\ &+ \int_{-\infty}^{\infty} \left[V(Se^x,t) - V(S,t) - S(e^x-1)\frac{\partial V}{\partial S}(S,t) \right] \nu(dx), \end{aligned} \tag{23}$$

defined on $(0,\infty) \times [0,T]$, with the terminal condition:

$$V(S,T) = Z(S) \quad \text{for all} \quad S > 0.$$

By applying a transformation of variables to the PIDE given in equation (23), the problem is reformulated. Let:

$$\tau = T - t, \quad x = \ln\left(\frac{S}{S_0}\right), \quad \text{and} \quad u(x,\tau) = V(S_0 e^x, T-\tau).$$

Under these new variables, $u(x,\tau)$ becomes the solution to a PIDE with constant coefficients:

$$\begin{aligned} \frac{\partial u}{\partial \tau}(x,\tau) &= \frac{1}{2}\sigma^2 \frac{\partial^2 u}{\partial x^2}(x,\tau) + \left(r - \frac{\sigma^2}{2} - \lambda\zeta\right)\frac{\partial u}{\partial x}(x,\tau) - (r+\lambda)u(x,\tau) \\ &+ \lambda \int_{-\infty}^{\infty} u(y,\tau) f(y-x) dy, \end{aligned} \tag{24}$$

with the initial condition:

$$u(x,0) = g(x) \quad \text{for all} \quad x \in \mathbb{R},$$

where $g(x) = \max(K - S_0 e^x, 0)$, $\zeta = \int_{-\infty}^{\infty}(e^x - 1) f(x) dx$, and $f(x)$ is the probability density function of the jump distribution. The parameter λ denotes the intensity of jumps.

The asymptotic properties of European put options are described as follows:

$$\lim_{x\to-\infty}[u(x,\tau) - (Ke^{-r\tau} - S_0 e^x)] = 0 \quad \text{and} \quad \lim_{x\to\infty} u(x,\tau) = 0, \tag{25}$$

where K is the strike price of the option. The PIDE presented in Equation (23) represents the evolution of the price $V(S,t)$ of a European put option under a jump-diffusion process, capturing both continuous market fluctuations (via Brownian motion) and sudden, discontinuous jumps (via a Poisson process). The first three terms on the right-hand side of the equation correspond to the diffusion part of the asset dynamics, akin to the classical Black-Scholes model. The final integral term incorporates the jump component, which accounts for large price movements that occur unexpectedly, such as market crashes or significant macroeconomic news. This term models the expectation of option value changes due to jumps and is weighted by the Lévy measure $\nu(dx)$, which describes the distribution and frequency of jumps.

This model is crucial for capturing the realistic behavior of financial markets, where asset prices do not always follow smooth paths. Therefore, using this PIDE enables a more accurate and robust valuation of financial derivatives in markets with discontinuities. Equation (24), resulting from a logarithmic change of variables, further simplifies the problem to allow for efficient numerical approximation, especially in unbounded domains.

4 Implicit-Explicit Finite Difference Discretization

We consider the following problem and we will implement our suggested method to solve the initial valued PIDE:

$$\frac{\partial u}{\partial \tau}(x,\tau) = Lu(x,\tau), \quad (x,\tau) \in \Gamma \times (0,T], \tag{26}$$

$$u(x,\tau) = h(x,\tau), \quad x \in \mathbb{R}\backslash\Gamma, \tag{27}$$

$$u(x,0) = g(x), \quad x \in \Gamma, \tag{28}$$

where L is the integro-differential operator, $\Gamma = (-X, X)$, X is a positive number, and $h(x,\tau) = max\big(0, Ke^{-r\tau} - S_0e^x\big)$ and $g(x) = max\big(K - S_0e^x, 0\big)$.
We define the differential operator D, and integral operator I, as

$$Du(x,\tau) = \frac{1}{2}\sigma^2\frac{\partial^2 u}{\partial x^2}(x,\tau) + \big(r - \frac{\sigma^2}{2} - \lambda\zeta\big)\frac{\partial u}{\partial x}(x,\tau), \tag{29}$$

$$Iu(x,\tau) = \lambda\int_{-\infty}^{\infty} u(y,\tau)f(y-x)dy, \tag{30}$$

to estimate the integral term numerically, we can split the integral into two components: one over Γ and another over $\mathbb{R}\backslash\Gamma$ is obtained by

$$R(x,\tau,X) = \int_{\mathbb{R}\backslash\Gamma}\big(Ke^{-r\tau} - S_0e^y\big)^+ f(y-x)dy, \tag{31}$$

in the Merton model, the jump density function is equal $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\dfrac{(x-\mu_j)^2}{2\sigma_j^2}}$, therefore the integral discussed can be written as

$$R(x,\tau,X) = Ke^{-r\tau}\Theta\Big(-\frac{x+X+\mu_j}{\sigma_j}\Big) - S_0e^{x+\mu_j+\dfrac{\sigma_j^2}{2}}\Theta\Big(-\frac{x+X+\mu_j+\sigma_j^2}{\sigma_j}\Big), \tag{32}$$

where

$$\Theta(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-\dfrac{x^2}{2}}dx, \tag{33}$$

to numerically approximate the integral term on the interval $(-X, X)$, we establish a uniform grid on the bounded domain $(-X, X) \times [0, T]$. Using positive integers N and M, we define $\Delta\tau = \frac{T}{N}$ and $\Delta x = \frac{2X}{M}$.
We set $\tau_n = n\Delta\tau$ for $n = 0, 1, 2, ..., N$ and $x_m = -X + m\Delta x$ for $m = 0, 1, ..., M$. We represent u_m^n as $u(x_m, \tau_n)$ and $f_{m,j}$ as $f(x_j - x_m)$. The integral term over the region $(-X, X)$ can be estimated using the composite trapezoidal rule as follows:

$$\int_{\Gamma} u(y,\tau_n)f(y-x_m)dy \approx \frac{\Delta x}{2}\Big(u_0^n f_{m,0} + 2\sum_{k=1}^{M-1} u_k^n f_{m,k} + u_M^n f_{m,M}\Big). \tag{34}$$

We use approximation at $\tau = \tau_n$ and $\tau = \tau_{n+1}$ by

$$u^n(x) := u(\tau_n, x), \quad u^{n+1}(x) := u(\tau_{n+1}, x), \tag{35}$$

now, we utilize finite difference discretization (details in [23])

$$\frac{u^{n+1}(x) - u^n(x)}{\Delta\tau} = D\left(\frac{u^{n+1}(x) + u^n(x)}{2}\right) + I\big(u^n(x)\big), \tag{36}$$

D and I are differential operators and integral operators respectively in Equations (32) and (33).

$$u^{n+1}(x) - u^n(x) = \Delta\tau\left(\frac{1}{2}D\big(u^{n+1}(x) - u^n(x)\big)\right) + \Delta\tau I\big(u^n(x)\big), \tag{37}$$

we arrive at the following recursive equation

$$u^{n+1}(x) - \frac{1}{2}\Delta\tau D u^{n+1}(x) = u^n(x) + \frac{\Delta\tau}{2} D u^n(x) + \Delta\tau I(u^n(x)), \tag{38}$$

after re-arranging the terms, the elliptic PDE (39)-(40) is obtained.

$$\begin{cases} u^{n+1}(x) - \dfrac{\Delta\tau}{2} D u^{n+1}(x) = u^n(x) + \dfrac{\Delta\tau}{2} D u^n(x) + \Delta\tau I(u^n)(x), \\ u(0,x) = \max\big(0, K - S_0 e^x\big), \\ u(\tau,x) = \max\big(0, K - e^{(-r\tau)} - S_0 e^x\big), \end{cases} \tag{39}$$

where

$$L[\cdot] := [\cdot] - \frac{\Delta\tau}{2} D[\cdot], \quad f^n := u^n + \frac{\Delta\tau}{2} D u^n + \Delta\tau I(u^n), \tag{40}$$

and g^n denotes the related boundary condition at t_n.

$$\begin{cases} L u^n = f^n & \text{in} \quad \Gamma, \\ u^n = g^n & \text{on} \quad \partial\Gamma. \end{cases} \tag{41}$$

We arrive at the recursive Equation (41). We obtain a solution by solving a sequence of second-order elliptic PDEs using the RBF-FD method.

5 Numerical results

All numerical experiments were performed using MATLAB 2017a on a computer with a 3.60 GHz CPU and 8.00 GB of RAM. In all the numerical experiments presented below, we used 101 local points in each stencil, unless stated otherwise. In this section, we examine the desired method for the five issues. These issues have already been solved in [23] using the shape parameter, the method discussed in this paper solves these problems with shorter time and higher accuracy. To report the calculation error, we use the RMS error, which is taken from the CPU time (in seconds), to show the effectiveness of the method in terms of calculation time. The specifications of the system used are as follows: Scripts were run using MATLAB 2017a on a computer with a 3.60 GHz CPU and 8.00 GB of RAM.

5.1 The First Problem

We investigate the following PDE at the first:

$$\begin{cases} \nabla^2(x,y) = \dfrac{-5}{4}\pi^2 \cos(\dfrac{\pi y}{2}) \sin(\pi x), & (x,y) \in L = [0,1] \times [0,1], \\ u(x,y) = \sin(\pi x), & (x,y) \in L_1, \\ u(x,y) = 0, & (x,y) \in L_2. \end{cases} \tag{42}$$

An analytic solution is given by

$$u(x,y) = \sin(\pi x) \cos(\frac{\pi y}{2}), \tag{43}$$

$$L_1 = \{(x,y) : 0 \leq x \leq 1, y = 0\}, \qquad \text{and} \qquad L_2 = \partial L \backslash L_1.$$

The RBF is used $\theta(r) = r^7$, $n = 101$ points for neighbor nodes, and the degree of the polynomial used is 7. We use Halton points for unstructured nodes. The result of various numbers of nodes (N) is brought in Table 2 and the distribution of structured and unstructured nodes is shown (as they are observed in Figure 1).

Table 2: RMS errors and CPU times for the first problem with different node types and densities.

	$RMS - error$		$CPU - Time$	
N(total nodes)	Structured	Halton	Structured	Halton
N=121	$3.470523e-01$	$2.782551e-01$	0.46	0.52
N=441	$8.691045e-01$	$4.054324e-02$	1.27	1.30
N=961	$3.856241e-02$	$2.886017e-02$	2.76	2.97
N=5041	$7.020331e-04$	$5.115574e-04$	21.99	63.74

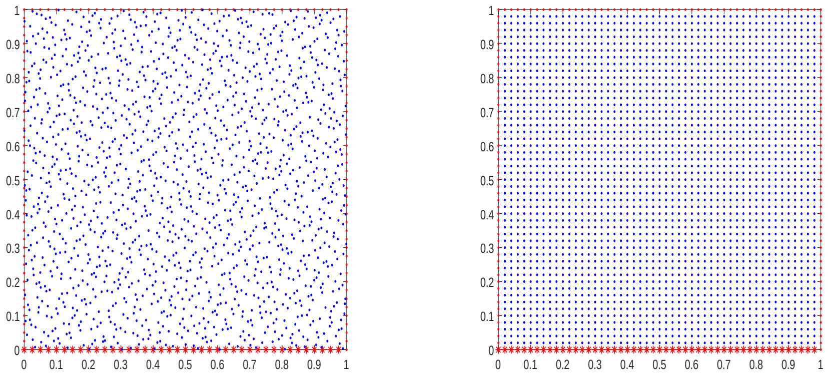


Figure 1: Structured vs. Halton node distributions for the first problem, showing differences in regularity and spatial coverage.

5.2 The Second Problem

For the second problem, we perform the method on modified Helmholtz PDE as:

$$\begin{cases} \nabla^2 u(x,y) - u(x,y) = f(x,y), & (x,y) \in \Gamma, \\ u(x,y) = h(x,y), & (x,y) \in \partial\Gamma. \end{cases} \tag{44}$$

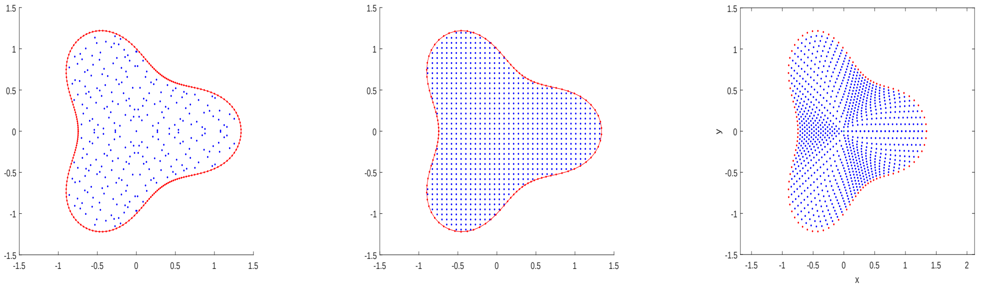
Let f and h denote known functions derived from the analytical solution The analytical solution is given by $u(x,y) = x\cos(\pi y) + y\sin(\pi x)$, where Γ_1 and Γ_2 are considered as domains, with their boundaries defined by the following parametric relations:

$$\partial\Gamma_i = \{(x,y) : x = r_i\cos(\alpha), y = r_i\sin(\alpha), -\pi \le \alpha \le \pi\},$$

where $r_1 = \frac{1}{81}(100 - 10\cos(9\alpha))$ and $r_2 = \left(\cos(3\alpha) + \sqrt{2 - \sin^2(3\alpha)}\right)^{\frac{1}{3}}$. Domains with structured and unstructured are in Figure 2. The RMS error for the second problem on Γ_2 is given in Table 3.

Table 3: RMS errors and times for the second problem on domain Γ_2, comparing node strategies.

	$RMS - error$		$CPU - Time$	
N(number of nodes)	Structured	Halton	Structured	Halton
N=100	$6.756743e-01$	$1.013648e-02$	0.24	0.64
N=200	$3.961037e-02$	$2.734398e-05$	0.28	1.03
N=300	$2.825933e-04$	$8.009564e-06$	0.34	1.46
N=400	$1.772352e-05$	$2.890093e-06$	0.45	2.09
N=500	$3.270082e-07$	$6.429890e-07$	2.96	2.77



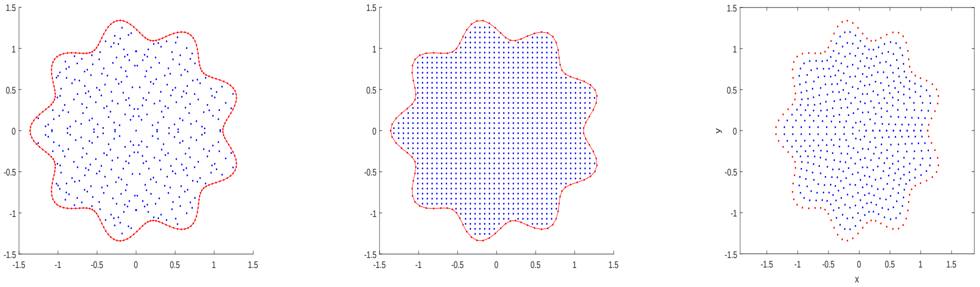


Figure 2: Structured and unstructured node sets for the second problem on irregular domains.

5.3 The Third Problem

We are examining a problem involving multiple boundary conditions as follows:

$$\begin{cases} \nabla^2 u(x,y) = 0, & (x,y) \in [-1,1] \times [-1,1], \\ u(x,y) = \dfrac{1}{5}\sin(3\pi y), & (x,y) \in L_2, \\ u(x,y) = \sin^4(\pi x), & (x,y) \in L_4, \\ u(x,y) = 0, & (x,y) \in L_1 \cup L_3 \cup L_5, \end{cases} \tag{45}$$

and their bounds are

$$\begin{aligned} L_1 &= \{(x,y) : -1 \le x \le 1, y = -1\}, \\ L_2 &= \{(x,y) : -1 \le y \le 1, x = +1\}, \\ L_3 &= \{(x,y) : 0 \le x \le 1, y = +1\}, \\ L_4 &= \{(x,y) : -1 \le x \le 0, y = 1\}, \\ L_5 &= \{(x,y) : -1 \le y \le 1, x = -1\}, \end{aligned}$$

as given in Figure 3.
An analytical solution to this problem is not available but we know $u(0,0) = 0.049595$ and we approximate $u(x,y)$ at $(0,0)$ on bounds for both structured and unstructured distributions as it is visible in Table 4.

5.4 The Fourth Problem

The fourth problem that we consider is a three-dimensional Poisson PDE follows:

$$\begin{cases} \nabla^2 u(x,y,z) & = f(x,y,z), \qquad (x,y,z) \in L, \\ u(x,y,z) & = h(x,y,z), \qquad (x,y,z) \in \partial L, \end{cases} \tag{46}$$

functions f and h are obtained by inserting the analytical solution $u(x,y)$ in the equation L_1, L_2 which are the unit cubic and the unit sphere, respectively (see Figure 4). Analytical solution is

Table 4: Approximation of solution at (0,0) in the third problem using different node types

	$RMS - error$		$CPU - Time$	
N(points)	Structured	Halton	Structured	Halton
N=121	$2.457101e - 02$	$5.000000e - 02$	1.40	1.53
N=361	$2.039124e - 02$	$1.298321e - 02$	2.10	2.19
N=1849	$1.066230e - 05$	$5.935990e - 04$	8.66	9.48

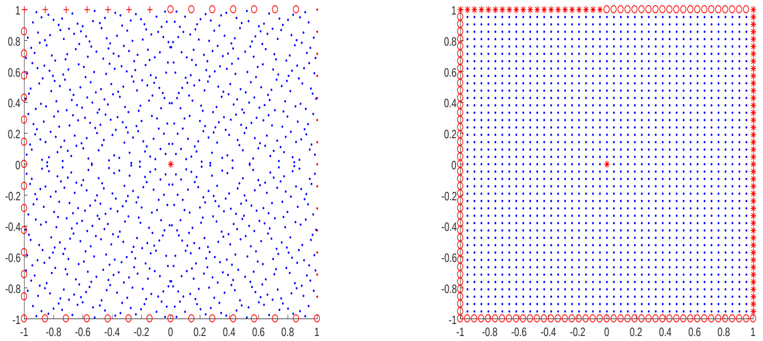


Figure 3: Node distributions for the third problem, used to solve Laplace's equation with mixed boundary conditions.

$$u(x,y,z) = x^3 + x^2 + y^2 - 2z^2. \tag{47}$$

Tables 5 and 6 represent the results of this test problem.

Table 5: RMS errors and CPU times for the fourth problem on a unit cube

	$RMS - error$		$CPU - Time$	
Total points	Structured	Halton	Structured	Halton
125	$2.340348e - 16$	$8.112420e - 15$	1.17	1.49
343	$4.349299e - 16$	$4.931915e - 16$	1.48	1.64
1331	$5.435784e - 16$	$3.130752e - 16$	1.68	1.93

Table 6: RMS errors for solving the Poisson equation on a unit sphere

	$RMS - error$		$CPU - Time$	
Total points	Structured	Halton	Structured	Halton
971	$9.976154e - 16$	$1.778201e - 15$	0.50	0.67
2105	$5.709501e - 16$	$2.320423e - 15$	0.99	0.89

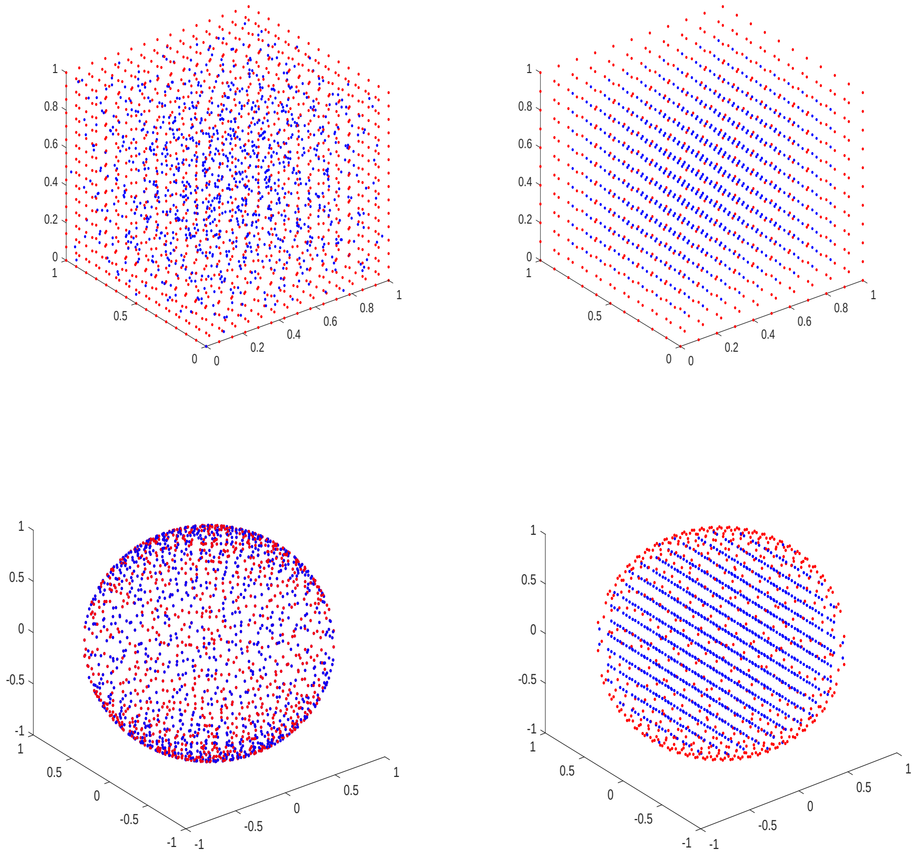


Figure 4: Structured and Halton nodes in 3D for the unit cube (top) and unit sphere (bottom).

5.5 The Fifth Problem

As a final problem, we examine the pricing of the European option under a jump-diffusion model. We employed the even PHS function along with a polynomial as the RBF, using a polynomial degree of three. The exact values are 9.285418 at $(S = 90)$, 3.149026 at $(S = 100)$, and 1.401186 at $(S = 110)$. The parameters utilized in the simulations are as follows:

$$\sigma_J = 0.45; \lambda = 0.10; K = 100; T = 0.25; r = 0.05; \mu_J = -0.90; \sigma = 0.15; S_0 = 100.$$

We present the numerical results obtained from the proposed method in Table 7, which demonstrate greater accuracy and speed compared to the values reported in the study by [23].

Table 7: European put option prices and maximum absolute errors under Merton's model computed via RBF-FD with polyharmonic splines (including $r^2 \log r$. Results at $S = 100$ and $S = 110$.

	$S = 100$		$S = 110$	
N	Value	Max-Error	Value	Max-Error
201	3.139797	$9.2280e-03$	1.398764	$2.4213e-03$
301	3.149595	$5.3921e-04$	1.400379	$8.0645e-04$
401	3.148036	$1.8920e-04$	1.400625	$5.6001e-04$

5.6 Economic Interpretation and Financial Implications

The precise and efficient valuation of financial derivatives, especially European options within jump-diffusion frameworks, is a fundamental aspect of quantitative finance. Conventional models like Black-Scholes do not account for discontinuities or sudden jumps observed in real asset prices. Merton's jump-diffusion model addresses this limitation by incorporating abrupt price changes, thereby better reflecting actual market behavior. The resulting partial integro-differential equations (PIDEs) derived from such models are considerably more complex to solve than the classic Black-Scholes PDE. In this context, the proposed Radial Basis Function-Finite Difference (RBF-FD) method represents a significant advancement. It offers practitioners and analysts the following benefits:

- **Enhanced Pricing Accuracy:** As demonstrated in Table 7, the RBF-FD approach produces highly precise option prices across various stock levels S, with errors around 10^{-4} even using a relatively small number of nodes. This level of accuracy is vital for effective risk management and portfolio hedging.

- **Reduced Computational Time:** Tables 4-7 reveal that this method attains lower root mean square (RMS) errors with less CPU time compared to existing RBF techniques that involve shape parameter optimization. This efficiency facilitates faster algorithmic trading and quicker scenario analysis under evolving market conditions.

- **Handling Complex Financial Instruments:** Its ability to efficiently solve high-dimensional PDEs makes the RBF-FD method suitable for pricing multi-asset options, credit derivatives, and structured products—areas where many traditional numerical methods face scalability issues.
- **Support for Risk Management:** The flexible framework can be adapted to compute Greeks, such as Delta, Vega, and Theta, which are essential for dynamic hedging. Future research may focus on directly calculating these sensitivities within the RBF-FD context.
- **Facilitating Real-Time Risk Assessment and Compliance:** With increasingly stringent regulatory standards (e.g., FRTB, Basel III) demanding real-time or near-real-time exposure calculations, the RBF-FD method can significantly boost a financial institution's ability to maintain compliance and competitiveness.

While the current study does not employ fractional derivatives directly, our mesh-free RBF-FD framework is highly adaptable and can be extended to solve fractional PDEs that arise in models incorporating long-range memory. Fractional derivatives are widely used in finance to model processes where future dynamics depend not only on the current state but also on historical values—capturing the so-called memory effect. This is particularly relevant in modeling volatility clustering, heavy-tailed distributions, and mean-reverting behaviors. Incorporating fractional-order terms into jump-diffusion models would enhance their capacity to reflect real market dynamics. Due to its flexibility in handling irregular domains and scattered data, our RBF-FD method is well-suited for such extensions, opening avenues for future research in robust financial modeling.

5.7 Stability and Noise Sensitivity Tests

To provide a more comprehensive evaluation of the proposed RBF-FD method, we conducted additional numerical experiments to assess both the *stability under varying stencil sizes* and *robustness to noisy data*. These experiments offer further insight into the performance of the method in realistic and challenging computational scenarios.

5.7.1 Stability with Varying Stencil Sizes

In this test, we applied the method to the first benchmark problem (Section 5.1) using different stencil sizes, namely $N = 51, 101, 151$, while keeping the number of total nodes fixed. Table 8 presents the root mean square errors and CPU times for each configuration. The results indicate that increasing the stencil size slightly improves accuracy while maintaining numerical stability.

Table 8: RMS errors and CPU times for different stencil sizes

Stencil Size (N)	RMS Error	CPU Time (s)
51	$5.22e-02$	0.38
101	$2.78e-02$	0.52
151	$2.55e-02$	0.87

5.7.2 Robustness to Noisy Boundary Data

We further tested the sensitivity of the method to noisy input by adding Gaussian noise to the boundary conditions of the third problem (Section 5.3). Specifically, noise with zero mean and variances $\sigma^2 = 10^{-2}, 10^{-4}, 10^{-6}$ was added to the prescribed boundary data. The approximate solution at the center point (0,0) was then compared to the noise-free case. The deviations are shown in Table 9. Results indicate that the method remains robust under mild to moderate noise levels, demonstrating its potential for practical applications involving imperfect data.

Table 9: Effect of boundary noise on computed value at $(0, 0)$

Noise Variance σ^2	Approximate $u(0,0)$	Absolute Deviation
0	0.04959	$-$
10^{-6}	0.04963	$8.06e-05$
10^{-4}	0.04986	$2.72e-04$
10^{-2}	0.05232	$2.73e-03$

6 Conclusion

This study employed the radial basis function-generated finite difference (RBF-FD) method to solve high-dimensional elliptic differential equations with Dirichlet boundary conditions, utilizing a combination of polyharmonic spline functions and polynomials for approximation. A key advantage of this approach is that polyharmonic spline functions (PHSs) eliminate the need for complex parameter tuning, simplifying implementation. Additionally, the study extended its scope to financial mathematics by transforming parabolic partial integro-differential equations into a series of second-order elliptic partial differential equations (PDEs), subsequently solved using the proposed method. The algorithm proved effective in accurately and efficiently solving high-dimensional PDEs across both regular and irregular domains without requiring calculation of the shape parameter. It also holds potential for application to nonlinear problems and more complex scenarios, such as biharmonic equations, with other RBFs serving as alternatives for approximation. Despite the demonstrated robustness, the RBF-FD method has several limitations. Its accuracy can be sensitive to highly irregular node distributions, especially in very high dimensions or extremely distorted computational domains. While PHSs enhance stability and simplicity, they may not achieve the spectral convergence rates of infinitely smooth RBFs. Future research could focus on adaptive node refinement techniques and exploring hybrid RBF frameworks that combine the benefits of both smooth and non-smooth basis functions to overcome these challenges.

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published paper.

Funding

The authors conducted this research without any funding, grants, or support.

Competing Interests

The authors declare that they have no competing interests relevant to the content of this paper.

Authors' Contributions

The main text of manuscript is collectively written by the authors.

References

[1] Abd-Elhameed, W.M., Atta, A.G., Youssri, Y.H. (2022). "Shifted fifth-kind Chebyshev polynomials Galerkin based procedure for treating fractional diffusion-wave equation", *International Journal of Modern Physics C*, 33(8), 2250102, doi:10.1142/S0129183122501026.

[2] Atta, A.G., Abd-Elhameed, W.M., Moatimid, G.M., Youssri, Y.H. (2022). "A fast Galerkin approach for solving the fractional Rayleigh–Stokes problem via sixth-kind Chebyshev polynomials", *Mathematics*, 10(11), 1843, doi:10.3390/math10111843.

[3] Atta, A.G., Abd-Elhameed, W.M., Moatimid, G.M., Youssri, Y.H. (2022). "Modal shifted fifth-kind Chebyshev Tau integral approach for solving heat conduction equation", *Fractal and Fractional*, 6(11), 619, doi:10.3390/fractalfract6110619.

[4] Atta, A.G., Youssri, Y.H. (2022). "Advanced shifted first-kind Chebyshev collocation approach for solving the nonlinear time-fractional partial integro-differential equation with a weakly singular kernel". *Computational and Applied Mathematics*, 41(8), 381, doi:10.1007/s40314-022-01904-1.

[5] Bayona, V., Flyer, N., Fornberg, B. (2019). "On the role of polynomials in RBF-FD approximations: III. Behavior near domain boundaries", *Journal of Computational Physics*, 380, 378-399, doi:10.1016/j.jcp.2018.12.036.

[6] Bayona, V., Flyer, N., Fornberg, B., Barnett, G.A. (2016). "On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy", *Journal of Computational Physics*, 321, 21-38, doi:10.1016/j.jcp.2016.05.005.

[7] Fasshauer, G.E. (2007). "Meshfree Approximation Methods with MATLAB, volume 44, *World Scientific Publishers*, Singapore, doi:10.1142/9789812706331.

[8] Fasshauer, G.E., Zhang, J.G. (2007). "On choosing "optimal" shape parameters for RBF approximation", *Numerical Algorithms*, 45, 345-368, doi:10.1007/s11075-007-9072-8.

[9] Fornberg, B., Flyer, N. (2015)."A primer on radial basis functions with applications to the geosciences", *SIAM, Philadelphia*, doi:10.1137/1.9781611974041.

[10] Fornberg, B., Flyer, N. (2015). "Solving PDEs with radial basis functions", *Acta Numerica*, 24, 215-258, doi:10.1017/S096249291500005X.

[11] Fornberg, B., Piret, C. (2008). "On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere", *Journal of Computational Physics*, 227, 2758-2780, doi:10.1016/j.jcp.2007.11.013.

[12] Fornberg, B., Bayona, V., Flyer, N., Barnett, G.A. (2017). "On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs", *Journal of Computational Physics*, 332, 257-273, doi:10.1016/j.jcp.2016.12.041.

[13] Haghighi, D., Abbasbandy, S., Shivanian, E. (2023). "Study of the Fragile Points Method for solving two-dimensional linear and nonlinear wave equations on complex and cracked domains", *Engineering Analysis with Boundary Elements*, 146, 44-55, doi:10.1016/j.enganabound.2022.09.036.

[14] Hardy, R.L. (1971). "Multiquadric equations of topography and other irregular surfaces", *Journal of Geophysical Research*, 76, 1905-1915, doi:10.1029/JB076i008p01905.

[15] Hosseinzadeh, N., Shivanian, E., Fairooz, M.Z., Chegini, T.G. (2025). "A robust RBF-FD technique combined with polynomial enhancements for valuing European options in jump-diffusion frameworks", *International Journal of Dynamics and Control*, 13, 212, 10.1007/s40435-025-01722-6.

[16] Kansa, E.J. (1990). "Multiquadrics—A scattered data approximation scheme with applications to computational fluid dynamics I: Surface approximations and partial derivative estimates", *Computers & Mathematics with Applications*, 19, 127-145, doi:10.1016/0898-1221(90)90270-T.

[17] Kansa, E.J. (1990). "Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations", *Computers & Mathematics with Applications*, 19, 147-161. doi:10.1016/0898-1221(90)90271-K.

[18] Moscoso, M., Bayona, V., Kindelan, M. (2011). "Optimal constant shape parameter for multiquadric based RBF-FD method", *Journal of Computational Physics*, 230, 7384-7399, doi:10.1016/j.jcp.2011.06.025.

[19] Moscoso, M., Bayona, V., Kindelan, M. (2012). "Optimal variable shape parameter for multiquadric based RBF-FD method", *Journal of Computational Physics*, 231, 2466-2481, doi:10.1016/j.jcp.2011.12.036.

[20] Phys, J.C. (2006). "Scattered node compact finite difference-type formulas generated from radial basis functions", *Engineering Analysis with Boundary Elements*, 212, 99-123, doi:10.1016/j.jcp.2005.05.030.

[21] Rahimi, A., Shivanian, E. (2023). "An efficient RBF-FD method using polyharmonic splines alongside polynomials for the numerical solution of two-dimensional PDEs held on irregular domains and subject to Dirichlet and Robin boundary conditions", *International Journal of Nonlinear Analysis and Applications*, (1), doi:10.22075/IJNAA.2023.28181.3826.

[22] Schaback, R. (1995). "Error estimates and condition numbers for radial basis function interpolants", *Advances in Computational Mathematics*, 3, 251-264, doi:10.1007/BF02123482.

[23] Shirzadi, M., Dehghan, M., Foroush Bastani, A. (2021). "A trustable shape parameter in the kernel-based collocation method with application to pricing financial options", *Engineering Analysis with Boundary Elements*, 126, 108-117, doi:10.1016/j.enganabound.2021.02.005.

[24] Shivanian, E., Khodabandehlo, H.R., Abbasbandy, S. (2022). "Numerical solution of nonlinear delay differential equations of fractional variable-order using a novel shifted Jacobi operational matrix", *Engineering with Computers*, 38, 204-219, doi:10.1007/s00366-020-01114-1.

[25] Shivanian, E., Jafarabadi, A., Chegini, T.G., Dinmohammadi, A. (2025). "Analysis of a time-dependent source function for the heat equation with nonlocal boundary conditions through a local meshless procedure", *Computational & Applied Mathematics*, 44, 282, doi:10.1007/s40314-025-03246-3.

[26] Shivanian, E., Hajimohammadi, Z., Baharifard, F., Parand, K., Kazemi, R. (2023). "A novel learning approach for different profile shapes of convecting–radiating fins based on shifted Gegenbauer LSSVM", *New Mathematics and Natural Computation*, 19 (1), 195-215, doi:10.1142/S1793005723500060.