Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(11): 27542–27572.
DOI: 10.3934/math.20231409
Received: 08 June 2023
Revised: 17 August 2023
Accepted: 27 August 2023
Published: 27 September 2023

Research article

Lagrange radial basis function collocation method for boundary value problems in 1D

Kawther Al Arfaj[1,*] **and Jeremy Levesly**[2]

[1] Department of Mathematics, College of Science, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia

[2] School of Computing and Mathematical Sciences, Leicester University, University Rd, Leicester, UK

* **Correspondence:** Email: kkalarfaj@kfu.edu.sa.

Abstract: This paper introduces the Lagrange collocation method with radial basis functions (LRBF) as a novel approach to solving 1D partial differential equations. Our method addresses the trade-off principle, which is a key challenge in standard RBF collocation methods, by maintaining the accuracy and convergence of the numerical solution, while improving the stability and efficiency. We prove the existence and uniqueness of the numerical solution for specific differential operators, such as the Laplacian operator, and for positive definite RBFs. Additionally, we introduce a perturbation into the main matrix, thereby developing the perturbed LRBF method (PLRBF); this allows for the application of Cholesky decomposition, which significantly reduces the condition number of the matrix to its square root, resulting in the CPLRBF method. In return, this enables us to choose a large value for the shape parameter without compromising stability and accuracy, provided that the perturbation is carefully selected. By doing so, highly accurate solutions can be achieved at an early level, significantly reducing central processing unit (CPU) time. Furthermore, to overcome stagnation issues in the RBF collocation method, we combine LRBF and CPLRBF with multilevel techniques and obtain the Multilevel PLRBF (MuCPLRBF) technique. We illustrate the stability, accuracy, convergence, and efficiency of the presented methods in numerical experiments with a 1D Poisson equation. Although our approach is presented for 1D, we expect to be able to extend it to higher dimensions in future work.

Keywords: positive definite radial basis functions; Lagrange RBF collocation method (LRBF); Multilevel Perturbed LRBF method (MuPLRBF); Ill-conditioning; Cholesky decomposition
Mathematics Subject Classification: 65F05, 65M30, 65N22

1. Introduction

Partial differential equations (PDEs) arise in various fields of science and engineering as mathematical models to describe complex physical phenomena. In many applications, an accurate and efficient numerical solution to PDEs is crucial. Among the numerous numerical techniques available for solving PDEs, radial basis function (RBF) methods, particularly collocation methods, have received significant attention in recent years [1–3].

RBF methods were first introduced by Hardy in the 1970s as a tool for spatial interpolation [4–6]. However, it wasn't until the late 1980s and early 1990s that RBFs began to gain traction as a numerical technique for solving PDEs, see; [7–9]. Pioneering work by Kansa, Micchelli, Madych, and Nelson in the early 1990s laid the foundations for the RBF collocation method, which has since become a popular approach to solving PDEs. After that, in the 2000s, many papers had been published to show the effectiveness of the RBF collocation method when applied to a wide range of applications [10–12]. An interesting modification of the RBF collocation method was introduced in [13], in which the authors numerically solve a Poisson equation with Dirichlet conditions through multinode Shepard interpolants by collocation. Furthermore, in [14], the Shepard multinode method was used to numerically estimate electric scalar potentials via collocation. RBF-based collocation methods still attract significant interest due to their various advantages, such as high accuracy, flexibility in handling complex geometries as a meshless method, and ease of implementation. One of the well-known element-free Galerkin methods was introduced in [15]. However, despite their potential benefits, these methods also come with a set of challenges, one of the main ones being the so-called "trade-off principle", which implies that an increase in the accuracy of the approximation is typically accompanied by a decrease in stability. This trade-off arises due to the ill-conditioning and high condition numbers of the matrix, which can impact the stability and accuracy of the numerical solution. Moreover, stagnation issues in the standard collocation RBF method where ill-conditioning of the problem and the stability issue are managed by adapting the shape parameter to the corresponding level, but where the error stagnates and does not improve-can further limit the effectiveness of these methods.

Several attempts have been made to tackle the trade-off principle in RBF collocation methods, such as using preconditioning techniques [16], employing different types of RBFs [17], or optimizing the shape parameter [18]. Despite these efforts, the trade-off principle remains a fundamental challenge in RBF collocation methods.

This paper introduces a novel numerical method, the Lagrange collocation method with radial basis functions (LRBF), for solving one-dimensional PDEs. Our method is designed to address the primary challenges associated with standard RBF-based collocation methods, especially the trade-off principle, while maintaining the accuracy and convergence properties of the solution. To accomplish this, we focus on specific differential operators such as the Laplacian operator and positive definite RBF functions. Then, we introduce a perturbation to the main matrix, leading to the development of the perturbed LRBF method (PLRBF). This perturbation allows us to apply Cholesky decomposition, significantly reducing the matrix's condition number to its square root. As a result, we can select a large value for the shape parameter without sacrificing either stability or accuracy, as long as the perturbation is chosen carefully. This approach enables us to obtain highly accurate solutions at early levels, thus greatly reducing the computational time and improving the overall efficiency of the method. Furthermore, we establish the existence and uniqueness of the numerical solution to our model.

To tackle the stagnation issues commonly observed in standard RBF collocation methods, we integrate the LRBF and PLRBF approaches with multilevel techniques, culminating in the Multilevel PLRBF method (MuPLRBF). Then, we demonstrate our main findings through numerical experiments in 1D. Although we initially focus on 1D problems, we anticipate that our approach can be extended to higher dimensions, which will be explored in future work.

The remainder of this paper is organized as follows. Section 2 presents the LRBF method, providing an overview of its key concepts and motivation. Section 3 delves deeper into the theoretical foundation and main characteristics of the method, including the existence and uniqueness of the numerical solution for positive definite RBFs. In Section 4, we demonstrate how applying the Cholesky decomposition to the LRBF method significantly improves stability by reducing the condition number of the matrix.

Section 5 introduces the PLRBF method, a novel approach that incorporates a perturbation into the main matrix, thereby further enhancing stability and enabling the use of Cholesky decomposition. Section 6 details the development of the MuLRBF and MuPLRBF methods, which combine the LRBF and PLRBF methods with multilevel techniques to address the stagnation issues commonly encountered in standard RBF collocation methods.

In Section 7, we describe a series of numerical experiments performed to validate our methods and discuss the findings related to accuracy, stability, and convergence. We compare the performance of our proposed methods with that of traditional RBF collocation methods and provide insights into their respective strengths and weaknesses. Finally, in Section 8, we conclude the paper by summarizing our key contributions and offering suggestions for future research directions, including potential extensions to higher-dimensional problems.

2. Lagrange RBF collocation method in 1D

Consider the following boundary value problem (BVP) for the target function $U(x)$, in the one-dimensional space $\Omega =]0,1[$.

$$\begin{cases} \mathcal{L}U(x) = f(x), & x \in \Omega, \\ U(0) = u_0, & U(1) = u_1, \end{cases} \tag{2.1}$$

where $\mathcal{L}$ is a linear differential operator and $f(x)$ is a given function. To obtain a general domain $[a, b]$ from the considered one, we use the transformation $y = (b-a)x+a$. Therefore, for the rest of this paper, without a loss of generality, we consider $\Omega =]0,1[$, and we define the nodes $x_i \in [0,1]$ as follows:

$$x_i = \frac{i-1}{n-1}, \quad i = 1, 2, \cdots, n,$$

where n is the number of nodes distributed on $[0,1]$. Note that $x_1 = 0$ and $x_n = 1$.

Summarizing the Lagrange RBF collocation method, we define the numerical solution $\widehat{U}$ to the problem (2.1) as follows:

$$\begin{cases} \widehat{U}(x) = \displaystyle\sum_{i=1}^{n} \beta_i \ell_i(x), \\ \ell_i(x) = \displaystyle\sum_{j=1}^{n} \alpha_{i,j}\phi(x - x_j), \qquad x \in [0,1], \\ \mathcal{L}\ell_i(x_k) = \delta_{ik}, \qquad\qquad\quad i,k = 1,2,...,n, \end{cases} \tag{2.2}$$

where β_i, $i = 1,2,\dots,n$ are unknown constant coefficients to be determined, ℓ_i, $i = 1,2,\cdots,n$ are linear combinations of RBFs $\phi(x-x_j)$, named $\mathcal{L}$-Lagrange, such that $\mathcal{L}\ell_i$ are Lagrange functions, δ_{ik} is the usual Kronecker delta symbol, and $\alpha_{i,j}$, $i,j = 1,2,\dots,n$ are constant coefficients to be determined. In addition, the numerical solution $\widehat{U}$ must satisfy the BVP system (2.1) at the nodes $(x_i)_{1\le i\le n}$. Hence, we get the following discrete system:

$$\begin{cases} \displaystyle\sum_{i=1}^{n} \beta_i \mathcal{L}\ell_i(x_k) = f(x_k), \quad k = 2,\cdots,n-1, \\ \displaystyle\sum_{i=1}^{n} \beta_i \ell_i(0) = u_0, \quad \sum_{i=1}^{n} \beta_i \ell_i(1) = u_1. \end{cases} \tag{2.3}$$

We write $f_k := f(x_k)$.

2.1. Determine the $\mathcal{L}$-Lagrange functions

We note that the $\mathcal{L}$-Lagrange functions $(\ell_i)_{1\le i\le n}$ depend on the differential operator $\mathcal{L}$ and the chosen grid $(x_i)_{1\le i\le n}$, but not on the numerical solution, and each Lagrange function $(\mathcal{L}\ell_i)_{1\le i\le n}$ satisfies the Lagrange conditions $\mathcal{L}\ell_i(x_k) = \delta_{ik}, i,k = 1,2,...,n$. The term Lagrange is usually applied to polynomials, but is also commonly used in the RBF literature; see [12].

Thus, the construction of the $\mathcal{L}$-Lagrange functions can be treated completely independently from the above. In order to determine the unknown constant coefficients $\alpha_{i,j}$, $i,j = 1,2,\dots,n$, we make use of the second equation in the system (2.2) by applying the linear operator $\mathcal{L}$ on both sides. We then have the following:

$$\mathcal{L}\ell_i(x) = \sum_{j=1}^{n} \alpha_{i,j}\mathcal{L}\phi(x - x_j), \quad x \in [0,1], \quad i = 1,2,\dots,n. \tag{2.4}$$

Using the Eq (2.4) at the set of nodes x_k, and the third equation in the system (2.2), we obtain the following:

$$\mathcal{L}\ell_i(x_k) = \sum_{j=1}^{n} \alpha_{i,j}\mathcal{L}\phi(x_k - x_j) = \delta_{ik}, \quad i,k = 1,2,\dots,n. \tag{2.5}$$

The relations (2.5) form the main linear systems for the coefficients $\alpha_{i,j}$. These systems can be rewritten as follows:

$$\mathbf{A}\boldsymbol{\alpha_i} = \boldsymbol{\delta_i}, \quad i = 1,2,\dots,n, \tag{2.6}$$

where the components are given by the following:

$$\mathbf{A} = \begin{pmatrix} \mathcal{L}\phi(x_1 - x_1) & \mathcal{L}\phi(x_1 - x_2) & . & . & \mathcal{L}\phi(x_1 - x_n) \\ \mathcal{L}\phi(x_2 - x_1) & \mathcal{L}\phi(x_2 - x_2) & . & . & \mathcal{L}\phi(x_2 - x_n) \\ . & . & . & . & . \\ . & . & . & . & . \\ \mathcal{L}\phi(x_n - x_1) & \mathcal{L}\phi(x_n - x_2) & . & . & \mathcal{L}\phi(x_n - x_n) \end{pmatrix},$$

$$\boldsymbol{\alpha_i} = \begin{pmatrix} \alpha_{i,1} \\ \alpha_{i,2} \\ . \\ . \\ \alpha_{i,n} \end{pmatrix} \text{ and } \boldsymbol{\delta_i} = \begin{pmatrix} \delta_{i1} \\ \delta_{i2} \\ . \\ . \\ \delta_{in} \end{pmatrix}. \tag{2.7}$$

As shown in Theorem 3.2, the matrix **A** is symmetric and a negative definite for positive definite RBF functions and for a certain differential operator $\mathcal{L} = \gamma D_x^2$ with $\gamma > 0$. Thus, we have a unique solution to the main linear system (2.6), and which can be formally written as follows:

$$\boldsymbol{\alpha_i} = \mathbf{A}^{-1}\boldsymbol{\delta_i},\ i = 1, 2, \dots, n. \tag{2.8}$$

The solutions to Eq (2.6) provide the coefficients $\alpha_{i,j}$, as described in (2.8), which determine the $\mathcal{L}$-Lagrange functions ℓ_i.

We recall that since the matrix **A** is independent of the numerical solution, and the right-hand side is the unit vector, we can solve the above linear system once and for all ℓ_i outside the main program, thus improving the efficiency.

2.2. The numerical solution

Applying the Kronecker delta condition from (2.2) to the first equation in (2.3), we obtain the following:

$$\sum_{i=1}^{n} \beta_i \delta_{ik} = f_k, \quad k = 2, \dots, n-1,$$

This results in the following:

$$\beta_k = f_k, \quad k = 2, \dots, n-1. \tag{2.9}$$

Therefore the numerical solution $\widehat{U}(x)$ in (2.2) can be expressed as follows:

$$\widehat{U}(x) = \beta_1 \ell_1(x) + \beta_n \ell_n(x) \ + \sum_{i=2}^{n-1} f_i \ell_i(x).$$

Hence, to determine the approximate solution $\widehat{U}(x)$, we need to determinate the coefficients β_1,β_n.

Insertion of $\mathcal{L}$-Lagrange functions ℓ_i into the boundary conditions of the system (2.3) leads to the following two equations in β_1 and β_n:

$$\beta_1 \ell_1(0) + \beta_n \ell_n(0) = u_0 - \sum_{i=2}^{n-1} f_i \ell_i(0),$$

$$\beta_1\ell_1(1)+\beta_n\ell_n(1)=u_1-\sum_{i=2}^{n-1} f_i\ell_i(1),$$

which can be presented in matrix form as follows:

$$\begin{pmatrix} \ell_1(0) & \ell_n(0) \\ \ell_1(1) & \ell_n(1) \end{pmatrix}\begin{pmatrix} \beta_1 \\ \beta_n \end{pmatrix}=\begin{pmatrix} u_0-\sum\limits_{i=2}^{n-1} f_i\ell_i(0) \\ u_1-\sum\limits_{i=2}^{n-1} f_i\ell_i(1) \end{pmatrix}. \tag{2.10}$$

Thus, we have a system with a 2×2 matrix, which can be easily solved to determine the coefficients β_1, β_n.

This, combined with (2.9), gives all β. The fact that the $\mathcal{L}$-Lagrange functions were determined separately in section 2.1 means that the desired numerical solution $\widehat{U}(x)$ in (2.2) has been established.

3. Well-posedness & main characteristics

So far, we have introduced the LRBF method in 1D for general radial basis functions and general differential operators. In this section, we focus on positive definite RBF functions and the second-order linear differential operator of the form $\mathcal{L}=\gamma D_x^2$, with $\gamma>0$ and D_x^2 representing the second derivative. For this class, we demonstrate the existence and uniqueness of our solution and show how to improve stability and efficiency without losing accuracy.

To prove the existence and uniqueness of the solution to the system (2.6), it is sufficient to prove that the matrix $\mathbf{A}$ is symmetric and either positive definite or negative definite. In fact, if the matrix is symmetric and either positive or negative definite, then it is invertible, which implies the existence and uniqueness of the solution.

Definition 3.1. *(Positive definite matrix) An $N\cdot N$ symmetric real matrix A is said to be positive semi-definite if its associated quadratic form is non-negative, i.e.,*

$$\sum_{j=1}^{N}\sum_{k=1}^{N}\lambda_j\lambda_k A_{jk}\geq 0,$$

for all vectors $\boldsymbol{\lambda}=(\lambda_1,\ldots,\lambda_N)^T\in\mathbb{R}^N$. If the only vector $\boldsymbol{\lambda}$ that turns the above quadratic form into an equality is the zero vector, then A is called positive definite.

Definition 3.2. *(Positive definite function) A real valued continuous function $\phi:\mathbb{R}^d\to\mathbb{R}$ is said to be positive semi-definite on $\mathbb{R}^d$ if, and only if, it is even and*

$$\sum_{j=1}^{N}\sum_{k=1}^{N}\lambda_j\lambda_k\phi\left(\mathbf{x}_j-\mathbf{x}_k\right)\geq 0,$$

for any N pairwise different points $\mathbf{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}\subseteq\mathbb{R}^d$, and $\boldsymbol{\lambda}=(\lambda_1,\ldots,\lambda_N)^T\in\mathbb{R}^N$. The function ϕ is strictly positive definite on $\mathbb{R}^d$ if the only vector $\boldsymbol{\lambda}$ that turns the above into an equality is the zero vector.

Note: It is worth noting the two well-known classes of positive definite RBFs, namely the Gaussian RBF $\phi(r) = \exp(-\frac{r^2}{c^2})$ and the family of Inverse multi-quadric $\phi(r) = (r^2 + c^2)^{-\beta}$, where $\beta \geq 0$, and c is a positive constant. In this paper, we focus on studying the Gaussian RBF and the Inverse multi-quadric, where $\beta = 1/2$, i.e., $\phi(r) = \frac{1}{\sqrt{r^2+c^2}}$.

To prove our main theorem on the well-posedness of our solution, we introduce the concept of Fourier transformation and make use of the Bochner Theorem 3.1.

Definition 3.3. *Let $f \in L^2(\mathbb{R}^d)$, where the Fourier transform $\mathcal{F}$ is defined as follows:*

$$\mathcal{F}(f)(\xi) = \int_{\mathbb{R}^d} f(\mathbf{x}) e^{-2\pi i \mathbf{x}\cdot\xi} d\mathbf{x}, \quad (\mathbf{x} \in \mathbb{R}^d)$$

and the inverse Fourier transform is defined by the following:

$$\mathcal{F}^{-1}(f)(\mathbf{x}) = \int_{\mathbb{R}^d} f(\xi) e^{2\pi i \mathbf{x}\cdot\xi} d\xi, \quad (\xi \in \mathbb{R}^d).$$

Theorem 3.1. *(Bochner, ([19], p70)) $\phi : \mathbb{R}^d \to \mathbb{R}$ is positive definite if, and only if, ϕ is bounded and its Fourier transform $\mathcal{F}(\phi)$, which is nonnegative and nonvanishing.*

Now, we are in a position to prove our theorem.

Theorem 3.2. *(Existence & uniqueness) For a positive definite radial basis function $\phi : \mathbb{R}^d \to \mathbb{R}$ and a differential operator $\mathcal{L} = \sum_{i,j=1}^{d} \gamma_{ij} \partial_{x_i} \partial_{x_j}$ with $\Gamma = (\gamma_{ij})_{1 \leq i,j \leq d}$ a symmetric and positive definite matrix, we define the matrix $\mathbf{A} \in \mathbb{R}^{n \cdot n}$ as follows:*

$$\mathbf{A} = \begin{pmatrix} \mathcal{L}\phi(\mathbf{x}_1 - \mathbf{x}_1) & \mathcal{L}\phi(\mathbf{x}_1 - \mathbf{x}_2) & . & . & \mathcal{L}\phi(\mathbf{x}_1 - \mathbf{x}_n) \\ \mathcal{L}\phi(\mathbf{x}_2 - \mathbf{x}_1) & \mathcal{L}\phi(\mathbf{x}_2 - \mathbf{x}_2) & . & . & \mathcal{L}\phi(\mathbf{x}_2 - \mathbf{x}_n) \\ . & . & . & . & . \\ . & . & . & . & . \\ \mathcal{L}\phi(\mathbf{x}_n - \mathbf{x}_1) & \mathcal{L}\phi(\mathbf{x}_n - \mathbf{x}_2) & . & . & \mathcal{L}\phi(\mathbf{x}_n - \mathbf{x}_n) \end{pmatrix},$$

where $(\mathbf{x}_k)_{k=1}^n$ are distinct points with $\mathbf{x}_k \in \mathbb{R}^d$. The matrix $\mathbf{A}$ is symmetric and negative definite. If Γ is a negative definite matrix, then the matrix $\mathbf{A}$ is positive definite.

Proof. First, using the fact that ϕ is an RBF and $\mathcal{L}$ is a symmetric differential operator with a direct calculation, we conclude that $\mathcal{L}\phi(\mathbf{x}_i - \mathbf{x}_j) = \mathcal{L}\phi(\mathbf{x}_j - \mathbf{x}_i)$, proving the symmetry of the matrix $\mathbf{A}$.

Now, we prove that the matrix A is negative definite, i.e.,

$$\lambda^T \mathbf{A} \lambda = \sum_{i,j=1}^{n} \lambda_i \lambda_j \mathcal{L}\phi(\mathbf{x}_i - \mathbf{x}_j) < 0,$$

for all nonzero vectors $\lambda \in \mathbb{R}^n$.

Using the Fourier transform $\mathcal{F}$ and a straightforward calculation, it can be shown that

$$\mathcal{F}(\mathcal{L}\phi)(\xi) = -4\pi^2 \xi^T \Gamma \xi \mathcal{F}(\phi)(\xi).$$

Applying the Bochner Theorem 3.1 to the positive definite RBF ϕ, we have $\mathcal{F}(\phi)(\xi) \geq 0$. Additionally, Γ is a positive definite matrix, i.e., $\xi^T \Gamma \xi > 0$.

Combining the last two facts, we clearly see that

$$\mathcal{F}(\mathcal{L}\phi)(\xi) \leq 0.$$

Now, applying the Bochner theorem from the other side (i.e., using the fact that the Fourier transform of $(\mathcal{L}\phi)$ is nonpositive and nonvanishing), it follows that the function $(\mathcal{L}\phi)$ is negative definite. This gives us the following:

$$\sum_{i,j=1}^{n} \lambda_i \lambda_j \mathcal{L}\phi(\mathbf{x}_i - \mathbf{x}_j) < 0,$$

which proves that $\mathbf{A}$ is negative definite. Similarly, if Γ is a negative definite matrix, then the matrix $\mathbf{A}$ is positive definite.

In particular, in the one-dimensional case, the differential operator takes the form $\mathcal{L} = \gamma D_x^2$ with $\gamma > 0$.

The above theorem proves the existence and uniqueness of the $\mathcal{L}$-Lagrange functions, and thus, the existence and uniqueness of our numerical solution.

3.1. Main characteristics of the LRBF method

We summarize several noteworthy characteristics of the linear system (2.6) that play a crucial role in enhancing the efficiency and stability of our novel LRBF method. In our numerical experiments in Section 7, we analyse the stability, accuracy, and efficiency of several methods, including RBF, confirming our findings below.

(1) **Stability:** The matrix $\mathbf{A}$ is symmetric and negative/positive definite (depending on the sign of γ) which allows for the use of different linear solvers such as LDL^T decomposition, in particular Cholesky decomposition. Using Cholesky decomposition, we show in Section 4 that the condition number of the main problem reduces to its square root (i.e., the condition numbers stay almost all the time under 10^{10}). However, numerical experiments show that this matrix is numerically unstable and often prevents the use of Cholesky decomposition. Thus, further thoughts and techniques are required to deal with this stability problem. To address this issue, we explore additional techniques and present a solution in Section 5.

(2) **Efficiency:** The fact that the $\mathcal{L}$-Lagrange functions are independent of the numerical solution allows for a calculation of the coefficients outside of the main program once and for all, saving valuable time during the calculations. In addition, the structure of the matrix $\mathbf{A}$ and the right-hand side $\boldsymbol{\delta}_i$ significantly improve the calculation speed in the following ways:

a) The matrix $\mathbf{A}$ is symmetric negative/positive definite, thus allowing for the use of a faster linear solver.

b) The right-hand side is the most trivial to calculate as it is the unit vector with 1 at element i and 0 elsewhere.

4. Improved stability of the LRBF method by applying Cholesky decomposition

In this section, the focus is on improving the condition number of the matrix $\mathbf{A}$ given in (2.6), which is negative/positive definite (depending on the sign of γ), as shown in Theorem 3.2. First, we briefly introduce the well-known Cholesky decomposition [20] and highlight its ability to significantly improve our problem's condition number. Next, we show that despite promising theoretical results, our system cannot confirm them numerically, as we encounter a numerically unstable matrix.

4.1. Recap: Cholesky decomposition

For a given symmetric and positive definite matrix $\mathbf{A}$, the Cholesky decomposition can be represented as follows:

$$\mathbf{A} = LL^T,$$

where L is a lower triangular matrix with positive diagonal elements.

Now, recalling the form of our main problem with a symmetric positive definite matrix $\mathbf{A}$,

$$\mathbf{A}\alpha = b. \tag{4.1}$$

Applying the Cholesky decomposition of $\mathbf{A} = LL^T$, we obtain the following:

$$LL^T\alpha = b, \tag{4.2}$$

which is equivalent to the linear system

$$\begin{cases} Ly & = b, \\ L^T\alpha & = y. \end{cases} \tag{4.3}$$

The formal expression for α can be written as follows:

$$\alpha = (L^T)^{-1}y = (L^T)^{-1}L^{-1}b.$$

In case $\mathbf{A}$ is negative definitive (i.e., the Cholesky decomposition becomes $\mathbf{A} = -LL^T$), we have the above system (4.3) with a minus on the first equation.

Additionally, we will make use of the following proposition, which can be proven through a direct calculation.

Proposition 4.1. *(Improved stability) For a symmetric positive definite matrix* $\mathbf{A}$ *with a Cholesky decomposition* $\mathbf{A} = LL^T$*, or in case* $\mathbf{A}$ *is negative definite, (*$A = -LL^T$*), we have*

$$Cond(\mathbf{A}) = Cond(L)^2, \tag{4.4}$$

where $Cond(\mathbf{A}) = ||\mathbf{A}||_2||\mathbf{A}^{-1}||_2$.

Proof. First, let us remark that, $||A||_2 = ||L||_2^2$. Indeed,

$$||A||_2 = \sqrt{\lambda_{\max}(A^TA)} = \sqrt{\lambda_{\max}^2(LL^T)} = \lambda_{\max}\left(LL^T\right) = ||L||_2^2.$$

Similarly,

$$||A^{-1}||_2 = ||(L^{-1})^T||_2^2 = ||(L^{-1})||_2^2.$$

Thus,

$$Cond(A) = \|A\|_2 \|A^{-1}\|_2 = \left(\|L\|_2 \|L^{-1}\|_2\right)^2 = Cond(L)^2.$$

Remark: The benefit of the Cholesky decomposition method lies in its ability to transform the main problem (4.1), which involves a matrix with a large condition number, into two linear systems given by (4.3). The matrices in these systems have much smaller condition numbers (the square root of the original, as shown above), which is expected to yield a more stable solution to our numerical problem. However, as demonstrated in the following section, the original matrix becomes numerically unstable beyond a certain level or for a large shape parameter, and does not even permit the application of Cholesky decomposition.

4.2. Numerical studies-Part I

In this section, we present our initial numerical studies, focusing on the stability and condition number of the matrix $\mathbf{A}$ for the Laplacian operator ($\mathcal{L} = \Delta$). We define equidistant nodes on the interval $[0, 1]$ and utilize two well-known positive definite RBF functions – the Gaussian function $\phi(r) = e^{(r/c)^2}$ and the inverse multi-quadric $\phi(r) = \frac{1}{\sqrt{r^2+c^2}}$, where $c > 0$ – is the shape parameter. In our experiments, we systematically explore various levels i and shape parameters C by setting $h_i = 2^{-i}$ and $c_i = C \cdot h_i$, where $C > 0$. We consider a comprehensive range of values, with i ranging from 1 to 9 and C given by 2^j for $-3 \leq j \leq 8$. To maintain the clarity and conciseness of our presentation, we have elected to display a subset of our results, specifically for levels up to 6 and C values of $1, 8$ and 32. This allows us to effectively illustrate the key findings of our study without overwhelming the reader with excessive data. Based on Theorem 3.2, we expect that the matrix $\mathbf{A}$ is negative definite, with all eigenvalues being strictly negative, thereby allowing for the Cholesky decomposition of $-\mathbf{A}$ at all costs. However, our experiments reveal that this property only holds for either small shape parameters, c, or at the first two levels. The primary observations from our experiments and Tables 1–3 can be summarized as follows:

- In line with expectations, the condition number of the original problem experiences a significant increase up to level 6.
- Beyond a specific level (e.g., level 4 and $C = 8$ for Gaussian), our matrix becomes numerically unstable and no longer permits the use of Cholesky decomposition. This instability is dependent on both the level and the shape parameter, C.
- A detailed analysis of our data indicates that the main issue lies in the rapid convergence of matrix entries to their diagonal counterparts, resulting in an almost singular matrix. This phenomenon is demonstrated in the Max EV column, where the maximum eigenvalue of the matrix approaches 0, signifying a numerically singular matrix.

Table 1. The condition number of the matrix **A** and the minimum/maximum eigenvalue of the matrix **A** using Gaussian RBF and IMQ RBF with $C = 1$.

	Gaussian				IMQ			
Level	Cond(**A**)	Max EV	Min EV	Cholesky Y/N?	Cond(**A**)	Max EV	Min EV	Cholesky Y/N?
1	3.55E+00	-3.3E+00	-1.2E+01	Y	1.76E+00	-5.4E+00	-9.6E+00	Y
2	7.13E+00	-6.7E+00	-4.8E+01	Y	2.41E+00	-3.2E+01	-7.7E+01	Y
3	1.78E+01	-1.1E+01	-1.9E+02	Y	3.59E+00	-1.7E+02	-6.2E+02	Y
4	5.09E+01	-1.5E+01	-7.7E+02	Y	5.26E+00	-9.5E+02	-5.0E+03	Y
5	1.36E+02	-2.3E+01	-3.1E+03	Y	6.84E+00	-5.9E+03	-4.0E+04	Y
6	2.69E+02	-4.6E+01	-1.2E+04	Y	7.81E+00	-4.1E+04	-3.2E+05	Y

Table 2. The condition number of the matrix **A** and the minimum/maximum eigenvalue of the matrix **A** using Gaussian RBF and IMQ RBF with $C = 8$.

	Gaussian				IMQ			
Level	Cond(**A**)	Max EV	Min EV	Cholesky Y/N?	Cond(**A**)	Max EV	Min EV	Cholesky Y/N?
1	2.86E+03	-1.2E-04	-3.5E-01	Y	5.03E+02	-8.5E-05	-4.3E-02	Y
2	2.54E+06	-8.3E-07	-2.1E+00	Y	2.29E+04	-2.2E-05	-4.9E-01	Y
3	2.10E+11	-5.2E-11	-1.1E+01	Y	1.33E+06	-3.8E-06	-5.0E+00	Y
4	5.51E+17	5.0E-16	-5.9E+01	N	2.33E+07	-2.0E-06	-4.7E+01	Y
5	9.53E+16	1.6E-14	-2.9E+02	N	7.95E+07	-5.5E-06	-4.4E+02	Y
6	1.17E+18	1.8E-13	-1.3E+03	N	1.12E+08	-3.4E-05	-3.8E+03	Y

Table 3. The condition number of the matrix **A** and the minimum/maximum eigenvalue of the matrix **A** using Gaussian RBF and IMQ RBF with $C = 32$.

	Gaussian				IMQ			
Level	Cond(**A**)	Max EV	Min EV	Cholesky Y/N?	Cond(**A**)	Max EV	Min EV	Cholesky Y/N?
1	7.83E+05	-3.0E-08	-2.3E-02	Y	1.31E+05	-5.6E-09	-7.3E-04	Y
2	1.98E+11	-7.8E-13	-1.5E-01	Y	1.38E+09	-7.0E-12	-9.6E-03	Y
3	1.01E+17	6.3E-17	-1.1E+00	N	2.01E+15	-6.4E-17	-1.3E-01	Y
4	1.47E+18	3.9E-16	-7.4E+00	N	5.13E+17	1.9E-17	-1.8E+00	N
5	3.31E+17	4.3E-15	-4.3E+01	N	6.10E+17	5.9E-16	-2.0E+01	N
6	2.56E+19	2.1E-14	-2.3E+02	N	1.72E+18	1.2E-14	-1.8E+02	N

Our numerical results highlight the challenges associated with the original matrix, despite its symmetric and positive definite nature. In certain cases, its numerical instability and near-singularity prevent the effective use of Cholesky decomposition, necessitating the exploration of alternative approaches to address these issues.

5. Perturbed $\mathcal{L}$-Lagrange functions and Perturbed LRBF method

In the standard LRBF method, we observed stability issues primarily arising due to matrix entries converging to their diagonal when the shape parameter is sufficiently large, and the distance between the points decreases. We introduce a perturbation technique and develop the perturbed LRBF method (PLRBF) to overcome this challenge. Our numerical experiments confirm that the new PLRBF method is more stable and, surprisingly, exhibits superior accuracy, especially when applying the multilevel technique.

First, we introduce a general perturbation to the diagonal entries of the matrix and then demonstrate simple yet very useful properties for the perturbed matrix when the original matrix is either positive or negative definite. We recall the key aspect of our approach is the application of Cholesky decomposition, which, together with the perturbation techniques, significantly improves the numerical stability of the method.

Definition 5.1. *(Perturbed matrix) For a small $\varepsilon \geq 0$ and a matrix $\mathbf{A} \in \mathbb{R}^{n \cdot n}$, we define a sequence of matrices of diagonal perturbation as follows:*

$$\mathbf{A}^{\varepsilon} = sgn(\lambda_{max})\mathbf{A} + \varepsilon I_n,$$

where $I_n = diag(1, 1, \ldots, 1)$ is the identity matrix and λ_{max} is the largest eigenvalue of the matrix $\mathbf{A}$.

Proposition 5.1. *(Properties of the perturbed matrix) For either a symmetric positive or negative definite matrix $\mathbf{A}$, the perturbed matrix $\mathbf{A}^{\varepsilon}$ as defined above satisfies the following:*

(1) For $\varepsilon \to 0$ the matrix $\mathbf{A}^{\varepsilon}$ converges uniformly to $sgn(\lambda)\mathbf{A}$.
(2) $\mathbf{A}^{\varepsilon}$ is symmetric and positive definite.
(3) When A is positive definite, we have the following:

$$cond(\mathbf{A}^{\varepsilon}) = \frac{\lambda_{max} + \varepsilon}{\lambda_{min} + \varepsilon}.$$

(4) When A is negative definite, we have the following:

$$cond(\mathbf{A}^{\varepsilon}) = \frac{|\lambda_{min}| + \varepsilon}{|\lambda_{max}| + \varepsilon}.$$

(5) $\mathbf{A}^{\varepsilon}$ is always as good or better conditioned than $\mathbf{A}$, i.e.,

$$cond(\mathbf{A}^{\varepsilon}) \leq cond(\mathbf{A}).$$

Proof. A direct calculation.

5.1. Perturbed LRBF method

Based on the perturbed matrix $\mathbf{A}^{\varepsilon}$, we define our perturbed $\mathcal{L}$-Lagrange functions as follows:

$$\ell_i^{\varepsilon}(x) = \sum_{j=1}^{n} \alpha_{i,j}^{\varepsilon}\phi(x - x_j), \quad x \in [0, 1], \tag{5.1}$$

where $\alpha^{\varepsilon}_{i,j}$ are the solution of the following perturbed system:

$$\mathbf{A}^{\varepsilon}\boldsymbol{\alpha}^{\varepsilon}_{i} = \boldsymbol{\delta}_{i}, \quad i = 1, 2, ..., n, \tag{5.2}$$

where $\boldsymbol{\alpha_i}^{\varepsilon} = \left(\alpha^{\varepsilon}_{i,1}, \ldots, \alpha^{\varepsilon}_{i,n}\right)^{T}$, and $\mathbf{A}^{\varepsilon}$ is the followingperturbation of the original matrix $\mathbf{A}$ as defined in (2.6).

This allows us to define our Perturbed LRBF (PLRBF) solution as follows:

$$U^{\varepsilon}(x) = \sum_{i=1}^{n} \beta_i \ell^{\varepsilon}_{i}(x). \tag{5.3}$$

Remark: By applying the Cholesky decomposition to the system (5.1), we determine the coefficients $\boldsymbol{\alpha}^{\varepsilon}_{i}$ and, thus, the perturbed $\mathcal{L}$-Lagrange functions ℓ^{ε}_{i} and the Cholesky PLRBF (CPLRBF) solution U^{ε}. A detailed implementation of the (perturbed) $\mathcal{L}$-Lagrange functions and the CPLRBF method is captured in Algorithms 1 and 2. Additionally, we note that U^{ε} converges uniformly to U when $\varepsilon \to 0$.

Algorithm 1: Algorithm of Perturbed $\mathcal{L}$-Lagrange functions

```
Input:
  • Choose the max level $n$, $\varepsilon$, and $C$.
  • Choose the RBFs $\phi(x)$.
  • Choose discrete $n_k = 2^k + 1$, $k = 1, \cdots, n$ points in the interval $[0, 1]$.
Step 1:
  • for $k = 1, \cdots, n$
      – Generate the grids $\mathbf{X}_k = (x^k_i)_{1\le i\le n_k}$, with $h_k = \frac{1}{2^k}$,
      – Set the shape parameters $c_k = C \cdot h_k$,
  • end for
Step 2:
  • for $k = 1, \ldots, n$ do
      – Calculate matrix $\mathbf{A}^{\varepsilon,k}$, and the unit vectors $(\boldsymbol{\delta}^k_i)_{1\le i\le n_k}$, as described in equation (2.7).
      –   for $i = 1, \ldots, n_k$ do
            ∗ If $\mathbf{A}^{\varepsilon,k}$ is positive definite,
              use Cholesky decomposition; solve the linear system $\mathbf{A}^{\varepsilon,k}\boldsymbol{\alpha}^{\varepsilon,k}_i = \boldsymbol{\delta}^k_i$;
              CPLRBF is ready;
            ∗ else
              solve the linear system $\mathbf{A}^{\varepsilon,k}\boldsymbol{\alpha}^{\varepsilon,k}_i = \boldsymbol{\delta}^k_i$;
              PLRBF is ready
      –   end for
  • end for
Output: Save and store all coefficients $(\boldsymbol{\alpha}^{\varepsilon,k}_i)_{1\le i\le n_k}$ for the expansions of the $\mathcal{L}$-Lagrange functions
  $\ell^{\varepsilon,k}_i$ in an Excel file for $k = 1, \cdots, n$.
```

Algorithm 2: Algorithm for the PLRBF collocation method

```
Input:
  • Choose the level $k$, and $C$.
  • Choose the RBFs $\phi(x)$.
  • Choose discrete $n$ points in the interval $[0, 1]$.
Step 1: Generate the grid $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, with $h = \frac{1}{2^k}$, and the shape parameters $c = C \cdot h$.
Step 2: Call the corresponding coefficients $(\boldsymbol{\alpha}_i^\varepsilon)_{1\le i\le n}$ for the $\mathcal{L}$-Lagrange functions $\ell_i^\varepsilon,\ i = 1, \cdots, n$,
already calculated in Algorithm 1. Either for PLRBF or CPLRBF
Step 3: Compute the coefficients $(\beta_i)_{2\le i\le n-1}$
  • for $i = 2, \ldots, n-1$ do
      – $\beta_i = f(x_i)$;
  • end for
Step 4: Compute the left- and the right-hand sides of the linear system given in (2.10).
      $(LHS)_{i,j} = \ell_i^\varepsilon(x_j),\ i, j = 1, n.$
      $(RHS)_j = g(x_j) - \sum_{i=2}^{n-1} \beta_i \ell_i^\varepsilon(x_j),\ j = 1, n.$
Step 5: Solve the linear system (2.10) to determinate $\beta_1$,$\beta_n$.
Output: Compute the solution $\widehat{U^\varepsilon}$.
      $\widehat{U^\varepsilon}(x) = \sum_{i=1}^{n} \beta_i \ell_i^\varepsilon(x)$;   (C)PLRBF-approximation solution.
```

5.2. Numerical studies-Part II

In this section, we investigate the condition number of the matrix $\mathbf{A}^\varepsilon$ for the Laplacian operator. To do so, we perform a series of experiments analogous to those in Part I, with $\varepsilon = 10^{-i}$ for $6 \le i \le 15$. From Theorem 3.2, we expect that the matrix $\mathbf{A}^\varepsilon$ will exhibit positive definiteness, resulting in strictly positive eigenvalues and enabling the Cholesky decomposition of $\mathbf{A}^\varepsilon$ at all $\cdot$. Our experiments validate this hypothesis in most cases, with the choice of ε proving to be a primary determinant of stability, while the influence of the level and shape parameter, c, is less pronounced than with the original matrix $\mathbf{A}$.

As in Part I, to maintain clarity and conciseness in our presentation, we have elected to display a subset of our results, specifically for levels up to 6, C values of 1, 8 and 32, and $\varepsilon = 10^{-6}, 10^{-15}$. This enables us to effectively convey the key findings of our study without overwhelming the reader with excessive data.

Upon analysing our data, we arrived at the following conclusions, confirmed with Tables 4–9:

- As shown in Proposition 5.1, the condition number of the perturbed matrix $\mathbf{A}^\varepsilon$ is either equal or smaller to the condition number of the original matrix $\mathbf{A}$ (i.e., the problem with $\mathbf{A}^\varepsilon$ is better conditioned).

- Our approach successfully mitigates the instability issues associated with the LRBF matrix, as the matrix $\mathbf{A}^{\varepsilon}$ demonstrates numerical stability, thereby allowing for Cholesky decomposition in the majority of cases.
- The choice of ε plays a key role in the system's stability. However, remarkably, in this regard, our experiments encountered only minimal limitations in this regard. For all $\varepsilon = 10^{-12}$ to 10^{-6}, no stability issues were observed for all levels up to 9 and all C values as defined in Part I.

Table 4. The condition number of the matrix $\mathbf{A}^{\varepsilon}$ and the minimum/maximum eigenvalue of the matrix $\mathbf{A}^{\varepsilon}$ using Gaussian RBF and IMQ RBF with $C = 1$ and $\varepsilon = 10^{-6}$.

Gaussian					IMQ			
Level	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?
1	3.55E+00	1.2E+01	3.3E+00	Y	1.76E+00	9.6E+00	5.4E+00	Y
2	7.13E+00	4.8E+01	6.7E+00	Y	2.41E+00	7.7E+01	3.2E+01	Y
3	1.78E+01	1.9E+02	1.1E+01	Y	3.59E+00	6.2E+02	1.7E+02	Y
4	5.09E+01	7.7E+02	1.5E+01	Y	5.26E+00	5.0E+03	9.5E+02	Y
5	1.36E+02	3.1E+03	2.3E+01	Y	6.84E+00	4.0E+04	5.9E+03	Y
6	2.69E+02	1.2E+04	4.6E+01	Y	7.81E+00	3.2E+05	4.1E+04	Y

Table 5. The condition number of the matrix $\mathbf{A}^{\varepsilon}$ and the minimum/maximum eigenvalue of the matrix $\mathbf{A}^{\varepsilon}$ using Gaussian RBF and IMQ RBF with $C = 8$ and $\varepsilon = 10^{-6}$.

Gaussian					IMQ			
Level	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?
1	2.84E+03	3.5E-01	1.2E-04	Y	4.97E+02	4.3E-02	8.6E-05	Y
2	1.15E+06	2.1E+00	1.8E-06	Y	2.18E+04	4.9E-01	2.3E-05	Y
3	1.09E+07	1.1E+01	1.0E-06	Y	1.05E+06	5.0E+00	4.8E-06	Y
4	5.90E+07	5.9E+01	1.0E-06	Y	1.56E+07	4.7E+01	3.0E-06	Y
5	2.89E+08	2.9E+02	1.0E-06	Y	6.74E+07	4.4E+02	6.5E-06	Y
6	1.26E+09	1.3E+03	1.0E-06	Y	1.09E+08	3.8E+03	3.5E-05	Y

Table 6. The condition number of the matrix $\mathbf{A}^{\varepsilon}$ and the minimum/maximum eigenvalue of the matrix $\mathbf{A}^{\varepsilon}$ using Gaussian RBF and IMQ RBF with $C = 32$ and $\varepsilon = 10^{-6}$.

Gaussian					IMQ			
Level	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?
1	2.27E+04	2.3E-02	1.0E-06	Y	7.25E+02	7.3E-04	1.0E-06	Y
2	1.54E+05	1.5E-01	1.0E-06	Y	9.60E+03	9.6E-03	1.0E-06	Y
3	1.08E+06	1.1E+00	1.0E-06	Y	1.33E+05	1.3E-01	1.0E-06	Y
4	7.44E+06	7.4E+00	1.0E-06	Y	1.77E+06	1.8E+00	1.0E-06	Y
5	4.25E+07	4.3E+01	1.0E-06	Y	1.97E+07	2.0E+01	1.0E-06	Y
6	2.29E+08	2.3E+02	1.0E-06	Y	1.82E+08	1.8E+02	1.0E-06	Y

Table 7. The condition number of the matrix $\mathbf{A}^{\varepsilon}$ and the minimum/maximum eigenvalue of the matrix $\mathbf{A}^{\varepsilon}$ using Gaussian RBF and IMQ RBF with $C = 1$ and $\varepsilon = 10^{-15}$.

Gaussian					IMQ			
Level	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?
1	3.55E+00	1.2E+01	3.3E+00	Y	1.76E+00	9.6E+00	5.4E+00	Y
2	7.13E+00	4.8E+01	6.7E+00	Y	2.41E+00	7.7E+01	3.2E+01	Y
3	1.78E+01	1.9E+02	1.1E+01	Y	3.59E+00	6.2E+02	1.7E+02	Y
4	5.09E+01	7.7E+02	1.5E+01	Y	5.26E+00	5.0E+03	9.5E+02	Y
5	1.36E+02	3.1E+03	2.3E+01	Y	6.84E+00	4.0E+04	5.9E+03	Y
6	2.69E+02	1.2E+04	4.6E+01	Y	7.81E+00	3.2E+05	4.1E+04	Y

Table 8. The condition number of the matrix $\mathbf{A}^{\varepsilon}$ and the minimum/maximum eigenvalue of the matrix $\mathbf{A}^{\varepsilon}$ using Gaussian RBF and IMQ RBF with $C = 8$ and $\varepsilon = 10^{-15}$.

Gaussian					IMQ			
Level	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?
1	2.86E+03	3.5E-01	1.2E-04	Y	5.03E+02	4.3E-02	8.5E-05	Y
2	2.54E+06	2.1E+00	8.3E-07	Y	2.29E+04	4.9E-01	2.2E-05	Y
3	2.10E+11	1.1E+01	5.2E-11	Y	1.33E+06	5.0E+00	3.8E-06	Y
4	3.52E+16	5.9E+01	1.2E-15	Y	2.33E+07	4.7E+01	2.0E-06	Y
5	9.53E+16	2.9E+02	-1.6E-14	N	7.95E+07	4.4E+02	5.5E-06	Y
6	1.17E+18	1.3E+03	-1.8E-13	N	1.12E+08	3.8E+03	3.4E-05	Y

Table 9. The condition number of the matrix $\mathbf{A}^{\varepsilon}$ and the minimum/maximum eigenvalue of the matrix $\mathbf{A}^{\varepsilon}$ using Gaussian RBF and IMQ RBF with $C = 32$ and $\varepsilon = 10^{-15}$.

Gaussian					IMQ			
Level	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?	Cond($\mathbf{A}^{\varepsilon}$)	Max EV	Min EV	Cholesky Y/N?
1	7.83E+05	2.3E-02	3.0E-08	Y	1.31E+05	7.3E-04	5.6E-09	Y
2	1.98E+11	1.5E-01	7.8E-13	Y	1.38E+09	9.6E-03	7.0E-12	Y
3	1.10E+15	1.1E+00	1.0E-15	Y	1.25E+14	1.3E-01	1.1E-15	Y
4	8.41E+15	7.4E+00	8.1E-17	Y	1.82E+15	1.8E+00	9.8E-16	Y
5	1.13E+18	4.3E+01	-9.6E-16	N	1.25E+19	2.0E+01	5.0E-16	Y
6	4.37E+18	2.3E+02	-2.2E-14	N	7.65E+17	1.8E+02	-1.2E-14	N

Our numerical experiments demonstrate the effectiveness of introducing a perturbed matrix $\mathbf{A}^{\varepsilon}$ in addressing the stability issues associated with the LRBF matrix. By carefully selecting the value of ε (e.g., $10^{-13} \leq \varepsilon \leq 10^{-6}$), we could consistently obtain numerically stable solutions and improved condition numbers compared to the original matrix $\mathbf{A}$.

6. The multilevel full grid PLRBF collocation method in one dimension

This section combines the well-established multilevel approach with our LRBF and PLRBF methods, resulting in the Multilevel LRBF (MuLRBF) and Multilevel PLRBF (MuPLRBF) methods. Our primary objective is to tackle the known stagnation issues in the standard collocation RBF method. By capitalizing on the perturbed matrix's robustness and utilizing the $\mathcal{L}$-Lagrange functions's precalculable nature, we anticipate that our MuPLRBF method will be more stable, efficient, and accurate.

Initially, we will develop the MuLRBF method for the BVP (2.1), followed by a comprehensive presentation of its associated algorithm, which is also valid for MuPLRBF-ensuring a solid foundation for implementing and analysing the MuLRBF/MuPLRBF/MuCPLRBF methods in the following section.

Let $\{\mathbf{X}_k\}_{1\leq k\leq m}$ be a strictly increasing sequence of nested full grids in the interval $[0, 1]$, each with $2^k + 1$ equally spaced nodes.

The approximation $\Delta\widehat{U}_1$ on the first grid $\mathbf{X}_1$ is constructed in such a way that satisfies the system

$$\begin{cases} \mathcal{L}(\Delta\widehat{U}_1)(x) = f(x), \ \ x \in]0, 1[, , \\ \Delta\widehat{U}_1(0) = u_0, \quad \Delta\widehat{U}_1(1) = u_1. \end{cases} \tag{6.1}$$

We define the approximation $\widehat{U}_1(x) := \Delta\widehat{U}_1(x)$. Now, for the grids $\mathbf{X}_k$, $k = 2, \ldots, m$, the approximations $\Delta\widehat{U}_k(x)$ satisfy the residual system, which takes the following form:

$$\begin{cases} \mathcal{L}(\Delta\widehat{U}_k)(x) = f(x) - \mathcal{L}\widehat{U}_{k-1}(x), \ \ x \in]0, 1[, \\ \Delta\widehat{U}_k(0) = U_0 - \widehat{U}_{k-1}(0), \quad \Delta\widehat{U}_k(1) = U_1 - \widehat{U}_{k-1}(1), \end{cases} \tag{6.2}$$

where $\widehat{U}_k(x) = \widehat{U}_{k-1}(x) + \Delta\widehat{U}_k(x)$. The final approximate solution $\widehat{U}_m$ is the sum of the numerical solution $\widehat{U}_1$ and the defect terms $\Delta\widehat{U}_k, k = 2,\ldots,m,$, i.e.,

$$\widehat{U}_m(x) = \widehat{U}_{m-1}(x) + \Delta\widehat{U}_m(x) = ... = \widehat{U}_1(x) + \sum_{k=2}^{m} \Delta\widehat{U}_k(x).$$

Since $\widehat{U}_1(x) := \Delta\widehat{U}_1(x)$, then

$$\widehat{U}_m(x) = \sum_{k=1}^{m} \Delta\widehat{U}_k(x). \tag{6.3}$$

The residual system at level k is written at the nodes $(x_i^k)_{1\leq i\leq n_k}$ as

$$\begin{cases} \mathcal{L}(\Delta\widehat{U}_k)(x_i^k) = f(x_i^k) - \mathcal{L}\widehat{U}_{k-1}(x_i^k), \quad x_i^k \in X_k\cap]0,1[, \\ \Delta\widehat{U}_k(0) = u_0 - \widehat{U}_{k-1}(0), \quad \Delta\widehat{U}_k(1) = u_1 - \widehat{U}_{k-1}(1). \end{cases} \tag{6.4}$$

The approximation $\Delta\widehat{U}_k$ is expressed as follows:

$$\Delta\widehat{U}_k(x) = \sum_{i=1}^{n_k} \beta_i^k \ell_i^k(x), \quad x \in [0,1], \tag{6.5}$$

where n_k is the number of nodes in the k^{th} grid and β_i^k are unknown coefficients. In the k^{th} level, the Lagrange's functions $\ell_i^k(x)$ are precalculated as described in Section 2, and can be given as follows:

$$\ell_i^k(x) = \sum_{j=1}^{n_k} \alpha_{i,j}^k \phi(x - x_j^k), \quad x \in [0,1], \tag{6.6}$$

where $\alpha_{i,j}^k$ are the constant coefficients calculated in Section 2.

By making use of (6.5) and (6.3), we rewrite the first equation in the system(6.4) as follows:

$$\mathcal{L}(\Delta\widehat{U}_k)(x_j^k) = \sum_{i=1}^{n_k} \beta_i^k \mathcal{L}\ell_i^k(x_j^k) = f(x_j^k) - \sum_{r=1}^{k-1} \mathcal{L}(\Delta\widehat{U}_r)(x_j^k), \quad j = 2,\ldots,n_k - 1. \tag{6.7}$$

Using the fact that $\mathcal{L}\ell_i^k(x_j^k) = \delta_{ij}$, we obtain the following:

$$\beta_i^k = f(x_i^k) - \sum_{r=1}^{k-1} \mathcal{L}(\Delta\widehat{U}_r)(x_i^k), \; i = 2,\ldots,n_k - 1. \tag{6.8}$$

Now, it remains to determine the coefficients β_1^k and $\beta_{n_k}^k$ to obtain the expression of the approximate solution, as given in (6.5). Using the boundary conditions expressed in the system (6.2), we obtain the following:

$$\Delta\widehat{U}_k(0) = \sum_{i=1}^{n_k} \beta_i^k \ell_i^k(0) = U_0 - \widehat{U}_{k-1}(0) := v_1^k,$$

$$\Delta\widehat{U}_k(1) = \sum_{i=1}^{n_k} \beta_i^k \ell_i^k(1) = U_1 - \widehat{U}_{k-1}(1) := v_2^k,$$

Then,

$$\beta_1^k \ell_1^k(0) + \beta_{n_k}^k \ell_{n_k}^k(0) = v_1^k - \sum_{i=2}^{n_k-1} \beta_i^k \ell_i^k(0),$$

$$\beta_1^k \ell_1^k(1) + \beta_{n_k}^k \ell_{n_k}^k(1) = v_2^k - \sum_{i=2}^{n_k-1} \beta_i^k \ell_i^k(1),$$

which can be written in the following matrix form

$$\begin{pmatrix} \ell_1^k(0) & \ell_{n_k}^k(0) \\ \ell_1^k(1) & \ell_{n_k}^k(1) \end{pmatrix} \begin{pmatrix} \beta_0^k \\ \beta_{n_k}^k \end{pmatrix} = \begin{pmatrix} v_1^k - \sum_{i=2}^{n_k-1} \beta_i^k \ell_i^k(0) \\ v_2^k - \sum_{i=2}^{n_k-1} \beta_i^k \ell_i^k(1) \end{pmatrix}.$$

The solution to this system and (6.8) give the coefficients β_i^k, $i = 1, 2, \cdots, n_k$, which determine the approximation $\Delta\widehat{U}_k(x)$ in the *kth* level. Similarly, we calculate the Multilevel PLRBF (MuPLRBF) solution and the Multilevel CPLRBF (MuCPLRBF) solution, as described in Algorithm 3.

Algorithm 3: Multilevel PLRBF (MuPLRBF) method

```
Input:
  • Choose $Tol$, $r$ = start level, $n$ = end level, $C$, $\widehat{U}_{r-1} = 0$, $Error_0 = 1$.
  • Choose the RBFs $\phi(x)$.
  • Choose discrete $n_k = 2^k + 1$, $k = r, \cdots, n$ points in the interval $[0, 1]$.
Step 1:
  • for $k = r, \cdots, n$
    – Generate the grids $\mathbf{X}_k = (x_i^k)_{1 \le i \le n_k}$, with $h_k = \frac{1}{2^k}$.
    – Set the shape parameters $c_k = C \cdot h_k$.
  • end for
Step 2: Compute a reference solution $\widehat{U}^{\varepsilon}_{\text{ref}}$ (e.g.: The exact solution if it is known, or a numerical solution on a very fine grid).
Step 3: Call and store the corresponding coefficients $(\alpha_i^{\varepsilon,k})_{1 \le i \le n_k}$ for $\mathcal{L}$-Lagrange functions $\ell_i^{\varepsilon,k}$, $i = 1, \cdots, n_k$; $k, = r, \cdots, n$, already calculated in Algorithm 1.
Step 4:
  • Set $j = 1$, $k = r$.
  • While $Error^{\varepsilon}_{j-1} \ge Tol$, and $k \le n$
    – Using the PLRBF collocation method, we solve
      $\begin{cases} \mathcal{L}\Delta\widehat{U}^{\varepsilon}_k(x) = f(x) - \mathcal{L}\widehat{U}^{\varepsilon}_{k-1}(x), & x \in \mathbf{X}_k \cap \Omega, \\ \Delta\widehat{U}^{\varepsilon}_k(x) = g(x) - \widehat{U}^{\varepsilon}_{k-1}(x), & x \in \mathbf{X}_k \cap \partial\Omega, \end{cases}$
    – Set $\widehat{U}^{\varepsilon}_k(x) = \widehat{U}^{\varepsilon}_{k-1}(x) + \Delta\widehat{U}^{\varepsilon}_k(x)$, $Error^{\varepsilon}_j = \max_{x \in \mathbf{X}_k} |\widehat{U}^{\varepsilon}_k(x) - \widehat{U}^{\varepsilon}_{\text{ref}}(x)|$.
    – If $Error^{\varepsilon}_j < Tol$,
      write (The level at which the $Tol$ was reached is k); $m = k$;
    – end if
    – If $k = n$,
      write (max level reached with a numerical error of $Error^{\varepsilon}_j$); $m = k$; break;
    – end if
      $k = k + 1$; $j = j + 1$;
  • end while
Output: $\widehat{U}^{\varepsilon}_k(x) = \sum_{k=1}^{m} \Delta\widehat{U}^{\varepsilon}_k(x)$
```

7. Numerical experiments

In this section, we discuss the numerical experiments conducted to evaluate the performance of our new methods. Our analysis encompasses seven distinct examples to comprehensively assess the methods' accuracy, convergence, and stability. While considering numerous scenarios in our experiments, we have carefully selected a representative subset to include in this paper to maintain clarity and conciseness. First, we will provide a brief overview of the experiments and the parameters under investigation. Following this, we delve into the results of one example in detail, focusing on the aspects of accuracy, convergence, and stability. Finally, we summarize our findings and highlight the key takeaways from these numerical experiments.

7.1. Overview of the experiments

These are the parameters we considered for our experiments:

- Example: BVP in 1D with known exact solution, including trigonometric functions.
- Methods: Six methods, namely RBF, MuRBF, LRBF, MuLRBF, CPLRBF, and MuCPLRBF. In the CPLRBF and MuCPLRBF methods, we consider different values for ε; $\varepsilon = 10^{-i}$ for $6 \leq i \leq 13$, resulting in a total of 20 methods.
- Levels: Nine levels with $h_i = 2^{-i}$, with $1 \leq i \leq 9$.
- Shape parameters: The constant C takes the values $C_j = 2^j$ with $-3 \leq j \leq 8$, and the adaptive shape parameter for each level i is $c_{i,j} = C_j \cdot h_i$.
- RBF functions: Gaussian RBF $\phi(r) = \exp(-\frac{r^2}{c^2})$.
- Norms: Max & RMS Norm defined as $Max\, u = \max_i |u(x_i)|$ and $RMS\ u = \sqrt{\sum_i^N u(x_i)^2/N}$, with x_i being the nodes on the interval $[0, 1]$.
- Tools: Matlab and Excel for data analysis (per example, we had 8424 data points).
- Notation: CPLRF & MuCPLRBF will be written as CPL & MuCPL in this section.
- A set of points (t), presenting 64,000 equally spaced points on $[0, 1]$.

7.2. Presentation of results for a representative example:

Now, we select the particular example below to illustrate the key performance attributes of our methods, allowing us to discuss the aspects of accuracy, convergence, and stability.

Let the target function U satisfy the following BVP:

$$\nabla^2 U(x) = -\frac{\sinh(x)}{(1+\cosh x)^2}, \quad x \in]0,1[\ ,$$
$$U(0) = 0, \quad U(1) = 0.$$

The exact solution to this problem as follows:

$$U(x) = \frac{\sinh(x)}{1+\cosh x}.$$

We solved this BVP using the parameters described in Subsection 7.1.

Our analysis revealed that the results strongly depend on the shape parameter. To make this clear, we present the analysis across three regions, defined by the C, and provide key takeaways for each region in terms of convergence, stability and accuracy. Parallel to these observations, we delve into the realm of computational efficiency, which is an oft-neglected, yet crucial aspect of numerical methods. Here, a salient finding is the CPLRBF method's pronounced efficiency from the third level onwards, where it consistently demanded nearly half the computation time of its standard RBF counterpart. This trend, as expected, also finds a reflection in the multilevel variants: the MuCPLRBF, in a race against time, invariably outpaces the MuRBF.

(1) Convergence, Stability and Accuracy

In the following, we show how the shape parameter, C, influences convergence, stability, and accuracy. We've categorised our analysis into three regions based on the value of C.

(a) Non-convergence region $C \leq c_0$:

In this region, there is neither convergence nor accuracy, as observed in Figure 1. In our representative example, c_o was equal to 1 (i.e., for $C \leq 1$ the methods were not able to deliver any meaningful solution). Although the condition number appears nearly ideal, as observed in Figure 2, this is primarily because the matrix converges to the identity matrix, which does not yield any meaningful insights into the PDE solution.

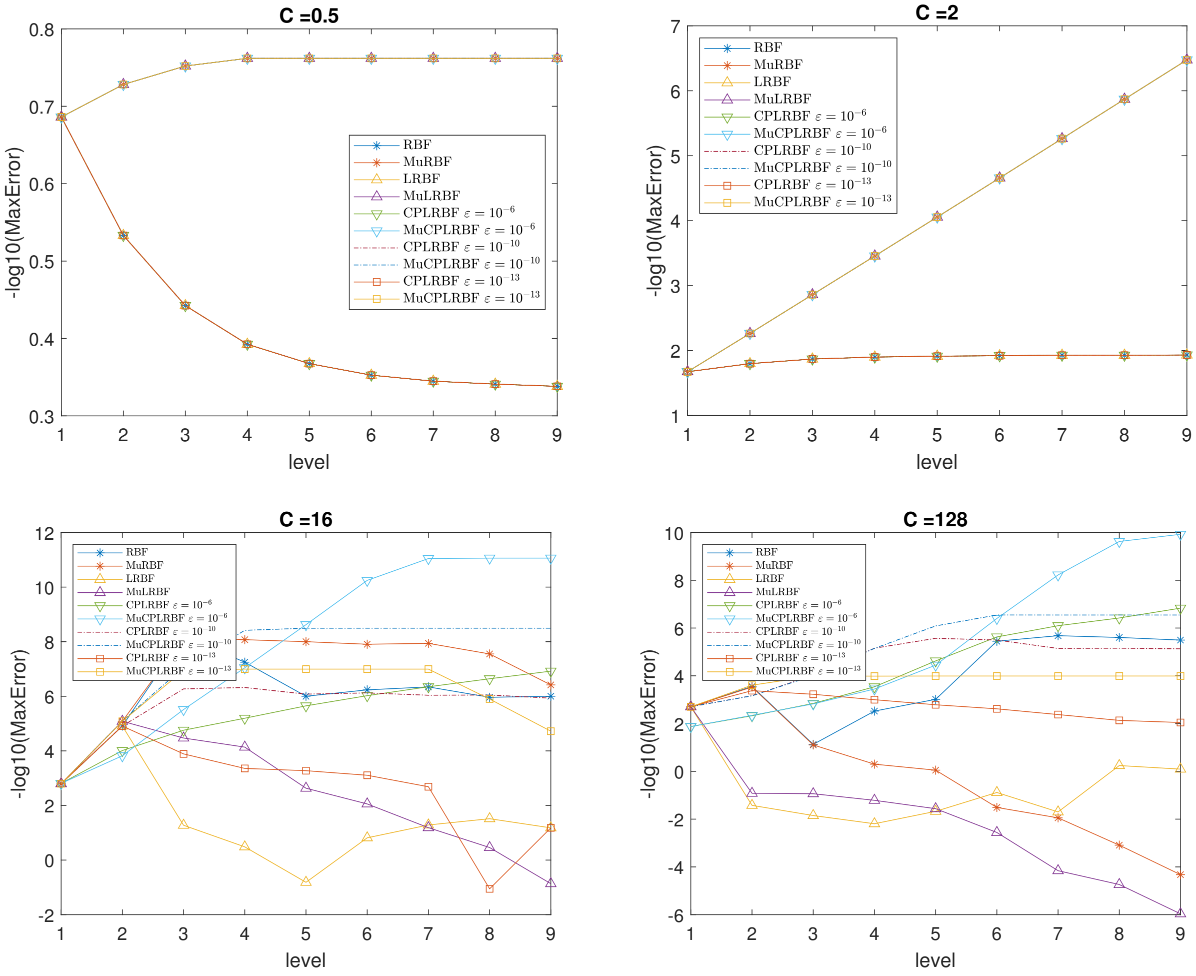

Figure 1. The Max Norm errors evaluated at the set of points t using the RBF, LRBF, CPLRBF, MuRBF, MuLRBF, and MuCPLRBF methods for different values of ε, C, and levels.

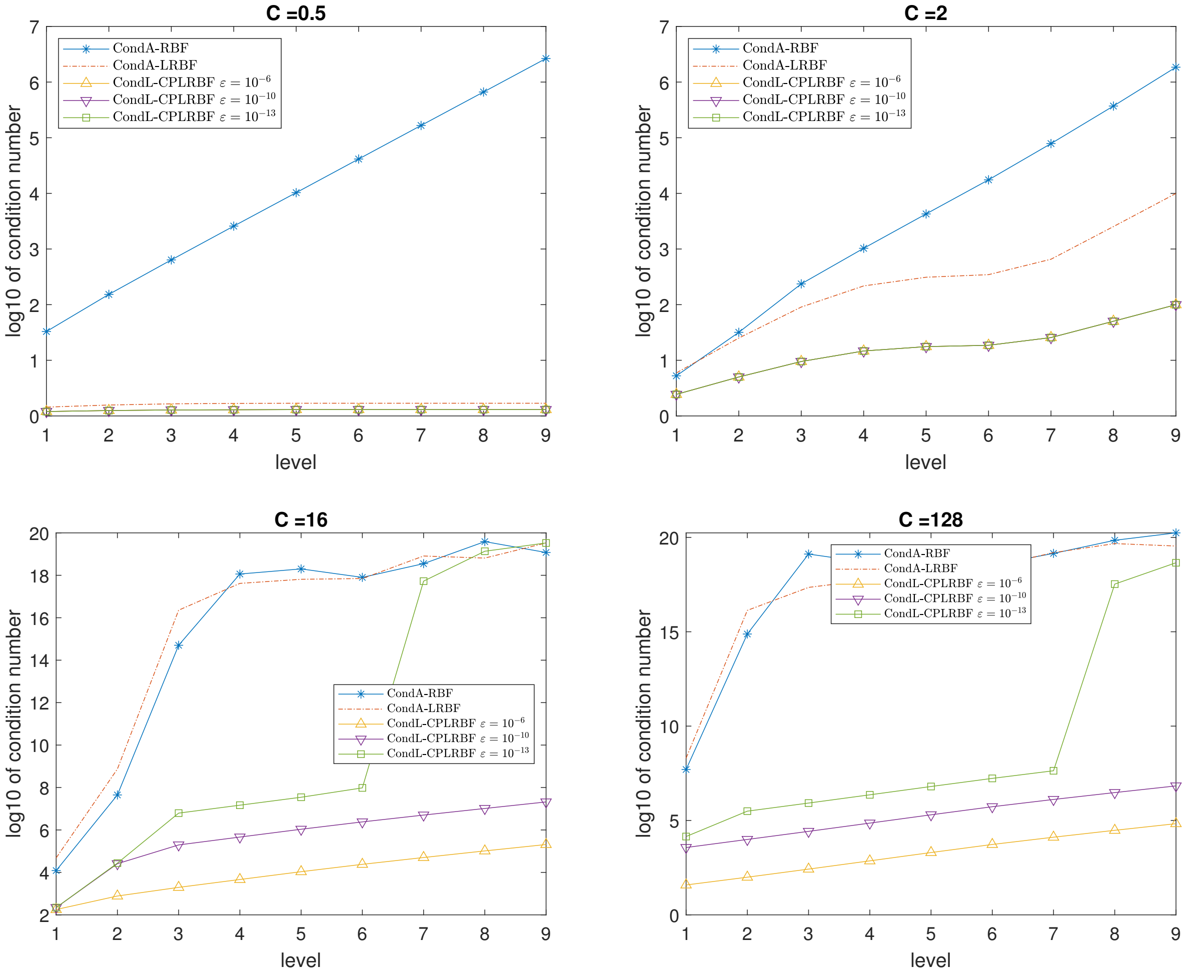

Figure 2. The condition number of the RBF, LRBF, and CPLRBF[I] methods for different values of ε, C, and levels.

(b) Convergence region $c_0 < C < c_1$:

This region represents the optimal performance for the multilevel methods. As demonstrated in Figure 1, for $C = 2$ and Table 10, the multilevel method effectively overcomes stagnation issues, thereby achieving higher accuracy levels. As we have convergence in this region and the fact that all the multilevel methods achieve the same accuracy level, we observe that the choice of ε does not significantly impact the accuracy and stability of the CPLRBF and the MuCPLRBF methods, as confirmed in Table 11.

In terms of convergence, our analysis and Tables 12 and 13 strongly suggest that the experimental order of convergence (EOC) for the multilevel methods is equal to the constant C chosen to define the shape parameter at each level for both the maximum and the RMS norm. This finding highlights the relationship between the shape parameter and the convergence behaviour of the multilevel methods, further emphasizing the importance of selecting an appropriate constant C to achieve optimal convergence rates, i.e., we assume

[I] The CPLRBF method is treated as the standard PLRBF method as soon as Cholesky is not possible, i.e., when the matrix for the PLRBF system shows numerical instability, the linear system is solved directly and not through a Cholesky decomposition, resulting in showing a higher condition number for the CPLRBF method as observed in Figure 2 in case $\varepsilon = 10^{-13}$ and for $C = 10$ & $C = 128$.

that

$$Error_{Mu} = O(h^{EOC}). \tag{7.1}$$

To show this, we calculated the EOC for our presented multilevel methods as follows:

$$EOC = \frac{\log\left(\frac{Error_n}{Error_{n+1}}\right)}{\log(2)}.$$

In terms of stability, the LRBF method displays stability behaviour similar to that of the standard RBF collocation method, with dependencies on both the shape parameter and the level. Perturbing the LRBF method slightly improves the stability, but dependencies still persist. In comparison, the CPLRBF method greatly surpasses the other methods in terms of stability and condition number. Our experiments reveal that the perturbed matrix is numerically stable, and the Cholesky decomposition's condition number supports our theoretical outcome of reducing the original condition number to its square root, as confirmed in Table 14.

Table 10. The smallest achieved errors for all considered methods using Gaussian RBF presented in ascending order based on the max norm evaluated at the set of points t, incl. their corresponding RMS norm, the condition number, CPU time, the level, and the constant C for the shape parameter.

Min	Max Norm	RMSN	Cond	Time	Level	C	Method
1	3.58E-13	9.24E-14	2.92E05	1.39E-01	9	8	MuCPLRBF, $\varepsilon = 10^{-6}$
2	5.07E-13	2.74E-13	6.53E14	5.32E-01	9	4	MuRBF
3	6.03E-13	3.01E-13	2.15E07	1.41E-01	9	4	MuCPLRBF, $\varepsilon = 10^{-10}$
4	6.15E-13	3.13E-13	1.16E07	1.40E-01	9	4	MuCPLRBF, $\varepsilon = 10^{-9}$
5	6.76E-13	2.97E-13	2.51E07	1.39E-01	9	4	MuCPLRBF, $\varepsilon = 10^{-11}$
6	6.84E-13	2.95E-13	6.44E14	1.43E-01	9	4	MuLRBF
7	7.34E-13	3.89E-13	4.08E06	1.41E-01	9	4	MuCPLRBF, $\varepsilon = 10^{-8}$
8	7.51E-13	2.94E-13	2.54E07	1.39E-01	9	4	MuCPLRBF, $\varepsilon = 10^{-13}$
9	7.64E-13	3.06E-13	2.54E07	1.41E-01	9	4	MuCPLRBF, $\varepsilon = 10^{-12}$
10	8.08E-13	1.94E-13	2.29E05	2.46E-02	7	8	MuCPLRBF, $\varepsilon = 10^{-7}$
11	6.16E-09	2.92E-09	5.05E14	9.63E-03	3	16	RBF
12	2.45E-08	8.96E-09	2.06E06	4.54E-02	9	16	CPLRBF, $\varepsilon = 10^{-8}$
13	2.65E-08	5.46E-09	4.59E05	4.91E-02	9	32	CPLRBF, $\varepsilon = 10^{-7}$
14	7.67E-08	1.52E-08	1.45E05	4.63E-02	9	32	CPLRBF, $\varepsilon = 10^{-6}$
15	1.02E-07	2.19E-08	7.59E05	2.26E-02	6	16	CPLRBF, $\varepsilon = 10^{-9}$
16	4.39E-07	1.05E-07	7.26E06	1.74E-02	7	8	CPLRBF, $\varepsilon = 10^{-10}$
17	1.92E-06	4.73E-07	6.23E05	9.62E-03	3	16	CPLRBF, $\varepsilon = 10^{-11}$
18	1.95E-06	1.04E-06	2.10E11	9.61E-03	3	8	LRBF
19	1.96E-06	1.07E-06	4.58E05	9.20E-03	3	8	CPLRBF, $\varepsilon = 10^{-13}$
20	2.05E-06	1.06E-06	4.54E05	9.47E-03	3	8	CPLRBF, $\varepsilon = 10^{-12}$

Table 11. The max norm errors evaluated at the set points t for the CPLRBF and MuCPLRBF methods using Gaussian RBF, $\varepsilon = 10^{-i}, 6 \leq i \leq 13$ with $C = 4$ for different values of level.

Maximum Norm								
Level	CPL $\varepsilon = 10^{-6}$	CPL $\varepsilon = 10^{-7}$	CPL $\varepsilon = 10^{-8}$	CPL $\varepsilon = 10^{-9}$	CPL $\varepsilon = 10^{-10}$	CPL $\varepsilon = 10^{-11}$	CPL $\varepsilon = 10^{-12}$	CPL $\varepsilon = 10^{-13}$
1	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03
2	1.28E-03	1.28E-03	1.28E-03	1.28E-03	1.28E-03	1.28E-03	1.28E-03	1.28E-03
3	3.34E-04	3.29E-04	3.29E-04	3.29E-04	3.29E-04	3.29E-04	3.29E-04	3.29E-04
4	1.68E-04	1.30E-04	1.10E-04	1.06E-04	1.06E-04	1.06E-04	1.06E-04	1.06E-04
5	1.30E-04	1.02E-04	8.35E-05	7.13E-05	6.42E-05	6.17E-05	6.13E-05	6.13E-05
6	1.05E-04	8.48E-05	7.13E-05	6.27E-05	5.75E-05	5.49E-05	5.42E-05	5.40E-05
7	8.91E-05	7.36E-05	6.36E-05	5.74E-05	5.39E-05	5.25E-05	5.23E-05	5.22E-05
8	7.77E-05	6.59E-05	5.85E-05	5.41E-05	5.20E-05	5.14E-05	5.13E-05	5.13E-05
9	6.94E-05	6.05E-05	5.50E-05	5.21E-05	5.11E-05	5.09E-05	5.09E-05	5.09E-05
Level	MuCPL $\varepsilon = 10^{-6}$	MuCPL $\varepsilon = 10^{-7}$	MuCPL $\varepsilon = 10^{-8}$	MuCPL $\varepsilon = 10^{-9}$	MuCPL $\varepsilon = 10^{-10}$	MuCPL $\varepsilon = 10^{-11}$	MuCPL $\varepsilon = 10^{-12}$	MuCPL $\varepsilon = 10^{-13}$
1	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03	4.03E-03
2	4.27E-04	4.26E-04	4.26E-04	4.26E-04	4.26E-04	4.26E-04	4.26E-04	4.26E-04
3	2.78E-05	2.78E-05	2.78E-05	2.78E-05	2.78E-05	2.78E-05	2.78E-05	2.78E-05
4	1.62E-06	1.37E-06	1.33E-06	1.33E-06	1.33E-06	1.33E-06	1.33E-06	1.33E-06
5	1.09E-07	8.46E-08	7.46E-08	6.62E-08	6.33E-08	6.29E-08	6.29E-08	6.29E-08
6	6.95E-09	5.10E-09	4.25E-09	3.66E-09	3.39E-09	3.28E-09	3.27E-09	3.26E-09
7	4.25E-10	2.97E-10	2.38E-10	2.00E-10	1.83E-10	1.76E-10	1.76E-10	1.75E-10
8	2.51E-11	1.68E-11	1.31E-11	1.08E-11	9.85E-12	9.52E-12	9.52E-12	9.43E-12
9	1.46E-12	9.42E-13	7.34E-13	6.15E-13	6.03E-13	6.76E-13	7.64E-13	7.51E-13

Table 12. The max norm EOC for the multilevel methods evaluated at the set of points t using Gaussian RBF with $\varepsilon = 10^{-13}$ for different C values and levels.

Maximum Norm								
$C \rightarrow$	$C = 1$		$C = 2$		$C = 4$		$C = 8$	
Level ↓	MuCPL	EOC	MuCPL	EOC	MuCPL	EOC	MuCPL	EOC
1	6.67E-02	-	2.12E-02	-	4.03E-03	-	4.41E-04	-
2	2.83E-02	1.24	5.45E-03	1.96	4.27E-04	3.24	1.24E-05	5.16
3	1.11E-02	1.35	1.38E-03	1.98	2.78E-05	3.94	4.85E-07	4.67
4	4.29E-03	1.37	3.50E-04	1.98	1.62E-06	4.10	1.75E-08	4.79
5	1.65E-03	1.38	8.80E-05	1.99	1.09E-07	3.90	5.38E-10	5.02
6	7.31E-04	1.18	2.20E-05	2.00	6.95E-09	3.97	1.46E-11	5.20
7	6.08E-04	0.27	5.46E-06	2.01	4.25E-10	4.03	6.08E-13	4.59
8	5.79E-04	0.07	1.35E-06	2.01	2.51E-11	4.08	3.59E-13	0.76
9	5.72E-04	0.02	3.35E-07	2.01	1.46E-12	4.11	3.58E-13	0.00

Table 13. The RMS norm EOC for the multilevel methods evaluated at the set of points t using Gaussian RBF with $\varepsilon = 10^{-13}$ for different C values and levels.

RMS Norm								
$C \rightarrow$	$C = 1$		$C = 2$		$C = 4$		$C = 8$	
Level ↓	MuCPL	EOC	MuCPL	EOC	MuCPL	EOC	MuCPL	EOC
1	3.78E-02	-	1.35E-02	-	2.82E-03	-	2.96E-04	-
2	1.40E-02	1.43	2.94E-03	2.20	2.58E-04	3.45	6.52E-06	5.50
3	5.00E-03	1.49	6.87E-04	2.10	1.59E-05	4.02	2.26E-07	4.85
4	1.90E-03	1.39	1.66E-04	2.05	8.98E-07	4.15	8.26E-09	4.78
5	8.38E-04	1.18	4.08E-05	2.03	5.91E-08	3.93	2.51E-10	5.04
6	4.99E-04	0.75	1.00E-05	2.02	3.74E-09	3.98	6.78E-12	5.21
7	4.07E-04	0.30	2.48E-06	2.02	2.28E-10	4.04	1.82E-13	5.22
8	3.83E-04	0.08	6.13E-07	2.02	1.35E-11	4.08	9.23E-14	0.98
9	3.78E-04	0.02	1.51E-07	2.02	7.75E-13	4.12	9.24E-14	0.00

Table 14. The condition numbers for the RBF, LRBF, PLRBF, and CPLRBF methods using Gaussian RBF with $C = 256$, and $\varepsilon = 10^{-10}$ for nine levels.

Cond(**A**) and Cond($\mathbf{A}^{\varepsilon}$)				Cond number Cholesky	
Level	RBF	LRBF	PLRBF $\varepsilon = 10^{-10}$	(CPLRBF)^2 $\varepsilon = 10^{-10}$	CPLRBF $\varepsilon = 10^{-10}$
1	8.05E08	3.22E09	3.66E06	3.66E06	1.91E03
2	1.95E17	1.83E17	2.44E07	2.44E07	4.94E03
3	6.50E20	7.57E17	1.76E08	1.76E08	1.33E04
4	1.36E19	5.96E17	1.33E09	1.33E09	3.64E04
5	8.04E18	5.11E18	1.02E10	1.02E10	1.01E05
6	6.10E18	1.30E19	7.87E10	7.87E10	2.81E05
7	1.06E19	1.38E19	5.71E11	5.71E11	7.56E05
8	4.37E20	2.91E19	3.38E12	3.37E12	1.84E06
9	1.06E20	1.67E20	1.82E13	1.81E13	4.26E06

(c) Partial convergence region $C > c_1$: Here, standard methods do not demonstrate convergence, accuracy, or stability. As observed in Figure 1 for $C = 16$ and $C = 128$, the errors for the standard methods either dramatically increase after a certain level or achieve a significantly lower accuracy level, as seen with the RBF method. Moreover, for $C = 16$ and $C = 128$, Figure 2 clearly shows that the condition number becomes excessively high after level 3, highlighting the ill-conditioning of the system. In contrast, as observed in Figure 2, our MuCPLRBF method maintains stability and achieves a good level of accuracy, as illustrated in Figure 1, albeit with some stagnation. In Table 15, we list the condition number for CPLRBF, respectively, noting the interesting fact that the condition number increases with the same factor as the ε decreases. Overall, this suggests that the MuCPLRBF method can deliver stable and accurate solutions at an early level, significantly reducing the computational time when ε is chosen carefully. Larger ε values lead to more stable and more accurate solutions, provided that ε remains sufficiently small (e.g., for $C > 16$, $\varepsilon = 10^{-6}$ delivers the most stable and accurate solution). It appears that the trade-off principle becomes more of a balance between the shape parameter and ε, that is, there is no stability issue for the perturbed methods, and the accuracy, even though not as strong as the standard methods, still depends on the shape parameter, C.

Table 15. The condition number for the CPLRBF method using Gaussian RBF, $\varepsilon = 10^{-i}, 6 \leq i \leq 13$ with $C = 128$ for different values of level.

	Condition number for CPLRBF							
Level	CPL $\varepsilon = 10^{-6}$	CPL $\varepsilon = 10^{-7}$	CPL $\varepsilon = 10^{-8}$	CPL $\varepsilon = 10^{-9}$	CPL $\varepsilon = 10^{-10}$	CPL $\varepsilon = 10^{-11}$	CPL $\varepsilon = 10^{-12}$	CPL $\varepsilon = 10^{-13}$
1	3.83E01	1.21E02	3.83E02	1.21E03	3.69E03	9.21E03	1.33E+04	1.41E+04
2	9.88E01	3.12E02	9.88E02	3.12E03	9.88E03	3.12E04	9.88E+04	3.12E+05
3	2.65E02	8.38E02	2.65E03	8.38E03	2.65E04	8.38E04	2.65E+05	8.38E+05
4	7.26E02	2.29E03	7.26E03	2.29E04	7.26E04	2.29E05	7.26E+05	2.29E+06
5	2.00E03	6.32E03	2.00E04	6.32E04	2.00E05	6.32E05	2.00E+06	6.32E+06
6	5.36E03	1.70E04	5.36E04	1.70E05	5.36E05	1.70E06	5.36E+06	1.70E+07
7	1.30E04	4.11E04	1.30E05	4.11E05	1.30E06	4.11E06	1.31E+07	4.31E+07
8	3.01E04	9.52E04	3.01E05	9.52E05	3.01E06	9.55E06	3.09E+07	3.36E+17
9	6.78E04	2.14E05	6.78E05	2.14E06	6.79E06	2.17E07	7.61E+07	4.54E+18

(2) Efficiency: Our detailed efficiency analysis shows that the CPLRBF method had a pronounced computational advantage over the standard RBF method, particularly from the third level onward. One of the primary reasons for this efficiency is the pre-calculation of the Lagrange functions outside of the main program once and for all, which greatly conserves computational time. As illustrated in Table 16, the CPLRBF method consistently operates in nearly half the time as the standard RBF. This same efficiency pattern was mirrored in their multilevel variants: MuCPLRBF consistently outperformed MuRBF, as confirmed in Table 17. As expected, this efficiency trend was observed to be consistent, regardless of the shape parameter choice, underscoring our proposed method's robustness. Such consistent time savings, when combined with the ability of the MuCPLRBF method to function optimally even at high C values, underpins a significant advantage of our proposed method. It implies that the MuCPLRBF method can deliver high-level accuracy at a fraction of the computational time, making it a formidable solution for time-intensive problems. Additionally, this can be seen in Table 10, where the MuCPLRBF with $\varepsilon = 10^{-7}$ achieved the best possible accuracy for $C = 8$ and level = 7 at a record time (i.e., 25 times faster than the standard MuRBF for the same accuracy level).

Table 16. The CPU time for RBF, LRBF, and CPLRBF methods using Gaussian RBF with $\varepsilon = 10^{-6}$, and $C = 8$ for different levels.

CPU time						
Method →	RBF		LRBF		CPLRBF	
Level ↓	Max-error	Time	Max-error	Time	Max-error	Time
1	4.43E-04	9.91E-03	4.43E-04	9.16E-03	4.41E-04	9.47E-03
2	6.25E-05	9.85E-03	6.25E-05	9.03E-03	6.46E-05	9.94E-03
3	1.87E-06	1.11E-02	1.95E-06	9.61E-03	2.90E-05	1.09E-02
4	6.88E-08	1.41E-02	2.77E-02	1.03E-02	1.15E-05	1.01E-02
5	4.07E-08	1.73E-02	4.45E-02	1.09E-02	6.30E-06	1.08E-02
6	5.19E-07	2.51E-02	1.64E-01	1.30E-02	3.52E-06	1.29E-02
7	2.82E-07	4.40E-02	4.14E-03	1.84E-02	2.11E-06	1.73E-02
8	4.17E-06	1.02E-01	1.95E-01	5.53E-02	1.31E-06	4.70E-02
9	1.26E-05	2.39E-01	7.09E-02	4.91E-02	8.40E-07	4.68E-02

Table 17. The CPU time for RBF, LRBF, and CPLRBF methods using Gaussian RBF with $\varepsilon = 10^{-6}$, and $C = 8$ for different levels.

CPU time						
Method →	MuRBF		MuLRBF		MuCPLRBF	
Level ↓	Max-error	Time	Max-error	Time	Max-error	Time
1	4.43E-04	9.15E-03	4.43E-04	9.36E-03	4.41E-04	9.15E-03
2	7.28E-06	9.70E-03	7.28E-06	9.35E-03	1.24E-05	9.65E-03
3	5.20E-08	1.07E-02	5.22E-08	9.91E-03	4.85E-07	9.82E-03
4	1.41E-10	1.24E-02	3.37E-08	1.09E-02	1.75E-08	1.07E-02
5	6.92E-11	1.47E-02	9.03E-08	1.17E-02	5.38E-10	1.16E-02
6	1.27E-10	2.33E-02	4.56E-07	1.53E-02	1.46E-11	1.49E-02
7	1.23E-10	5.01E-02	5.22E-07	2.48E-02	6.08E-13	2.44E-02
8	1.41E-09	1.54E-01	3.02E-05	5.49E-02	3.59E-13	5.32E-02
9	8.87E-08	5.41E-01	1.60E-03	1.41E-01	3.58E-13	1.39E-01

Remark: In our study, we also looked at and analysed four other examples including: polynomial, exponential, log, and trigonometric functions. In all these cases, we established similar results to those detailed in the presented example, thus confirming the summary of our findings in the following section.

7.3. Summary of our findings

Our investigation has explored various RBF-based methods and their respective properties, focusing on accuracy, stability, and convergence. Here, we summarize our key findings and the main advantages as well as limitations of our CPLRBF method:

(1) Enhanced Stability with Perturbation: The CPLRBF method significantly improves stability over the standard RBF and LRBF collocation methods, largely credited to the use of a perturbed

Lagrange matrix and the implementation of Cholesky decomposition.

(2) Optimized Performance in the Convergence Region: Operating within the shape parameter range $c_0 < C < c_1$, the MuCPLRBF method showcases its superiority in terms of stability and condition number. It achieves an EOC that matches C, paralleling the accuracy of traditional methods, though with augmented efficiency.

(3) Efficiency from Early Stages: From level 3 onwards, the MuCPLRBF method stands out, not just in terms of matching the accuracy of standard RBF, but also in its computational swiftness, thereby requiring notably shorter durations to deliver an accurate solution.

(4) Reliability in High C Values: With rising values of C, while many methods falter, the MuCPLRBF method remains steadfast, delivering solutions with consistent accuracy.

(5) Promising Scalability: While the current focus is on one-dimensional problems, the MuCPLRBF method is not just a solution for the present; it is a promising avenue for tackling higher-dimensional challenges in future research, retaining its advantages in stability and efficiency.

(6) Parameter Interplay Complexity: A deep understanding of the interplay between the shape parameter, C, and the perturbation parameter, ε, is still elusive. While our findings suggest a nuanced balance, a comprehensive understanding of their direct relationship might be a limitation of the current method and certainly serves as an exciting avenue for future research.

8. Conclusions and remarks

In this paper, we have introduced novel numerical methods, the Lagrange collocation method with radial basis functions (LRBF) and the perturbed version (PLRBF) for solving 1D PDEs. We presented an in-depth analysis of various RBF-based methods, with a particular emphasis on the introduction of perturbation and multilevel techniques to address the challenges associated with the standard RBF collocation method. Our findings not only elucidate the properties of these methods, but also provide valuable insights for future research and applications.

We found that incorporating perturbation and developing the CPLRBF method significantly improves stability when applying the Cholesky decomposition compared to standard RBF and LRBF collocation methods. The choice of the perturbation parameter, ε, shape parameter, C, and level all play crucial roles in determining stability and convergence. However, a clear relationship between these parameters is still pending and presents one of the main limitations of our method.

Three distinct regions were identified based on the chosen shape parameter, C, each with different convergence, accuracy, and stability characteristics. Our numerical experiments in 1D confirmed our main findings and showcased the potential of our proposed methods, particularly the MuCPLRBF method, in delivering accurate solutions at an early level, thereby significantly reducing computational time.

In conclusion, our study highlights the importance of adopting perturbed (thus Cholesky) and multilevel techniques to address challenges in the standard RBF collocation method.

While this paper has focused on one-dimensional problems, we anticipate that our approach can be extended to higher dimensionalities, a topic we plan to explore in future work.

We would expect more challenges around choosing the perturbed parameter ε and its dependency on the shape parameter C. Additionally, while we've emphasised the benefits of pre-calculating the

Lagrange functions, the associated computational overhead, particularly for more extensive and complex systems in higher dimensions, is a limitation that warrants further investigation. The results presented in this paper open up new avenues for research and application of RBF-based methods in solving PDEs, offering improved stability and efficiency while maintaining high accuracy and convergence level.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors thank Ruslan Davidchack for proofreading the paper and for the insightful discussions and suggestions on presenting the numerical results.

Funding

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Project No. GRANT4010].

References

1. B. Fornberg, N. Flyer, Solving PDEs with radial basis functions, *Acta Numer.*, **24** (2015), 215–258. https://doi.org/10.1017/S0962492914000130
2. M. Nawaz Khan, I. Ahmad, H. Ahmad, A radial basis function collocation method for space-dependent inverse heat problems, *J. Appl. Comput. Mech.*, **6** (2020), 1187–1199. https://doi.org/10.22055/JACM.2020.32999.2123
3. C. S. Chen, A. Karageorghis, F. Dou, A novel RBF collocation method using fictitious centres, *Appl. Math. Lett.*, **101** (2020), 106069. https://doi.org/10.1016/j.aml.2019.106069
4. R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.*, **76** (1971), 1905–1915. https://doi.org/10.1029/JB076i008p01905
5. W. R. Madych, Miscellaneous error bounds for multiquadric and related interpolators, *Comput. Math. Appl.*, **24** (1992), 121–138. https://doi.org/10.1016/0898-1221(92)90175-H
6. S. Hubbert, T. M. Morton, Lp-error estimates for radial basis function interpolation on the sphere, *J. Approx. Theory*, **129** (2004), 58–77. https://doi.org/10.1016/j.jat.2004.04.006
7. E. J. Kansa, Multiquadrics–A scattered data approximation scheme with applications to computational fluid-dynamics–I surface approximations and partial derivative estimates, *Comput. Math. Appl.*, **19** (1990), 127–145. https://doi.org/10.1016/0898-1221(90)90270-T
8. C. A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, *Constr. Approx.*, **2** (1986), 11–22. https://doi.org/10.1007/BF01893414

9. E. J. Kansa, 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** (1990), 147–161. https://doi.org/10.1016/0898-1221(90)90271-K

10. H. Y. Hu, Z. C. Li, A. H. D. Cheng, Radial basis collocation methods for elliptic boundary value problems, *Comput. Math. Appl.*, **50** (2005), 289–320. https://doi.org/10.1016/j.camwa.2004.02.014

11. W. Du Toit, *Radial basis function interpolation*, Stellenbosch: Stellenbosch University, 2008.

12. M. D. Buhmann, Radial basis functions, *Acta Numer.*, **9** (2000), 1–38. https://doi.org/10.1017/S0962492900000015 .

13. F. Dell'Accio, F. Di Tommaso, O. Nouisser, N. Siar, Solving Poisson equation with Dirichlet conditions through multinode Shepard operators, *Comput. Math. Appl.*, **98** (2021), 254–260. https://doi.org/10.1016/j.camwa.2021.07.021

14. F. Dell'Accio, F. Di Tommaso, G. Ala, E. Francomano, Electric scalar potential estimations for non-invasive brain activity detection through multinode Shepard method, In: *IEEE 21st mediterranean electrotechnical conference (MELECON)*, Italy, 2022, 1264–1268. https://doi.org/10.1109/MELECON53508.2022.9842881

15. X. Li, S. Li, Meshless Galerkin analysis of the generalized Stokes problem, *Comput. Math. Appl.*, **144** (2023), 164–181. https://doi.org/10.1016/j.camwa.2023.05.027

16. G. E. Fasshauer, *Meshfree approximation methods with matlab*, World Scientific, 2007. https://doi.org/10.1142/6437

17. R. Schaback, *A practical guide to radial basis functions*, 2007.

18. A. C. Faul, *Iterative techniques for radial basis function interpolation*, University of Cambridge, 2000. *https://doi.org/10.17863/CAM.74829*

19. H. Wendland, *Scattered data approximation*, Cambridge University Press, 2010. https://doi.org/10.1017/CBO9780511617539

20. N. J. Higham, *Functions of matrices: theory and computation*, Society for Industrial and Applied Mathematics, 2008.

21. W. Chen, Z. J. Fu, C. S. Chen, *Recent advances in radial basis function collocation methods*, Heidelberg: Springer, 2014. https://doi.org/10.1007/978-3-642-39572-7