

Article

An Investigation of Radial Basis Function-Finite Difference (RBF-FD) Method for Numerical Solution of Elliptic Partial Differential Equations

Suranon Yensiri ¹  and Ruth J. Skulkhu ^{1,2,*} 

¹ Department of Mathematics, Faculty of Science, Mahidol University, Bangkok 10400, Thailand; suranonmath@gmail.com

² Centre of Excellence in Mathematics, Commission on Higher Education, Ministry of Education, Si Ayutthaya Road, Bangkok 10400, Thailand

* Correspondence: ruthj.sku@mahidol.ac.th

Received: 2 September 2017; Accepted: 17 October 2017; Published: 23 October 2017

Abstract: The Radial Basis Function (RBF) method has been considered an important meshfree tool for numerical solutions of Partial Differential Equations (PDEs). For various situations, RBF with infinitely differentiable functions can provide accurate results and more flexibility in the geometry of computation domains than traditional methods such as finite difference and finite element methods. However, RBF does not suit large scale problems, and, therefore, a combination of RBF and the finite difference (RBF-FD) method was proposed because of its own strengths not only on feasibility and computational cost, but also on solution accuracy. In this study, we try the RBF-FD method on elliptic PDEs and study the effect of it on such equations with different shape parameters. Most importantly, we study the solution accuracy after additional ghost node strategy, preconditioning strategy, regularization strategy, and floating point arithmetic strategy. We have found more satisfactory accurate solutions in most situations than those from global RBF, except in the preconditioning and regularization strategies.

Keywords: numerical partial differential equations (PDEs); radial basis function-finite difference (RBF-FD) method; ghost node; preconditioning; regularization; floating point arithmetic

1. Introduction

For several decades, numerical solutions of partial differential equations (PDEs) have been studied by researchers in many different areas of science, engineering and mathematics. The common mathematical tools that are used to solve these problems are finite difference and finite element methods. These traditional numerical methods require the data to be arranged in a structured pattern and to be contained in a simply shaped region, and, therefore, these approaches does not suit multidimensional practical problems. The Radial basis function (RBF) method is viewed as an important alternative numerical technique for multidimensional problems because it contains more beneficial properties in high-order accuracy and flexibility on geometry of the computational domain than classical techniques [1–3]. The main idea of the RBF method is approximating the solution in terms of linear combination of infinitely differentiable RBF ϕ , which is the function that depends only on the distance to a center point $\underline{x}$ and shape parameter ε . More precisely, we approximate the solution as:

$$s(x, \varepsilon) = \sum_{j=1}^N \lambda_j \phi(\|x - x_j\|_2, \varepsilon), \quad (1)$$

where $x_j, j = 1, \dots, N$ are the given centers. For simplicity, we set $\phi(r, \varepsilon) = \phi(\|x - \underline{x}\|_2, \varepsilon)$. Some common RBFs in the infinitely differentiable class are listed in Table 1.

Table 1. Radial basis functions.

Name of RBF	Abbreviation	Definition
Multiquadrics	MQ	$\phi(r, \varepsilon) = \sqrt{1 + (\varepsilon r)^2}$
Inverse Multiquadrics	IMQ	$\phi(r, \varepsilon) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}}$
Inverse Quadratics	IQ	$\phi(r, \varepsilon) = \frac{1}{1 + (\varepsilon r)^2}$
Gaussians	GA	$\phi(r, \varepsilon) = e^{-(\varepsilon r)^2}$

For the case of interpolation of function $y = f(x)$, the coefficients $\{\lambda_j\}_{j=1}^N$ are determined by enforcing the following interpolation condition:

$$s(x_i, \varepsilon) = f(x_i) = y_i, \quad i = 1, 2, \dots, N.$$

This condition results in an $N \times N$ linear system:

$$[A(\varepsilon)] \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad (2)$$

where the elements of $A(\varepsilon)$ are $a_{j,k} = \phi(\|x_j - x_k\|_2, \varepsilon)$. Note that the system matrix $A(\varepsilon)$ is known to be nonsingular if the constant shape parameter ε is used and is ill-conditioned for small values of shape parameter [4,5].

The concept of RBF interpolation can also be implemented for solving elliptic PDE problems. For the global RBF method, we assume the solution in the form of Equation (1). Then, the coefficients λ_j are determined by enforcing the PDE and boundary conditions. Unlike the case of interpolation, the system matrix of this case does not guarantee to be nonsingular and is also ill-conditioned for small values of shape parameter [1,6]. Additionally, according to numerical evidence, the global RBF scheme requires high computational cost and memory requirements for large scale problems [3,7]. Therefore, the radial basis function-finite difference (RBF-FD) concept, which is a local RBF scheme, was developed by combining many benefits of the RBF method and traditional finite difference approximations. Similar to the finite difference approach, the key idea is approximating the differential operator of solutions at each interior node by using a linear combination of the function values at the neighboring node locations and then determining the FD-weights so that the approximations become exact for all the RBFs that are centered at the neighboring nodes. Straightforward algebra will then show that the FD-weights can be obtained by solving the linear system, for which the system matrix is the same as matrix A in Equation (2). The invertibility of A also implies that FD-weights can always be computed. After the calculation at all interior nodes, the approximate solution can be computed from the linear system of equations, for which the RBF-FD system matrix is sparse and therefore can be effectively inverted. Note that the RBF-FD method that we described can be efficiently and applicably implemented with large scale practical problems such as the global electric circuit within the Earth's atmosphere problem [8], steady problems in solid and fluid mechanics [9–11], chemotaxis models [12], and diffusion problems [13,14].

In the first part of this work, the numerical solution of elliptic PDEs by using the RBF-FD method with IQ, MQ, IMQ, and GA RBF is compared to the results of the global RBF scheme. It will show that the accuracy of both global and local methods does not depend on the choice of RBFs, and the RBF-FD method with small stencils can obtain the same level of accuracy as the global RBF method. However, the RBF-FD method is algebraically accurate in exchange for low computational cost and needs more additional techniques to improve the accuracy [10,14]. Thus, in the latter part of this work, we will aim to research improving the accuracy of the RBF-FD method by using different

ways to formulate the RBF-FD method. In particular, the effects of the number of nodes and stencils on the accuracy of numerical solutions of the RBF-FD method are investigated. According to [7,15], the global RBF method can improve the accuracy significantly for small values of shape parameter by using the regularization technique and extended precision floating point arithmetic to reduce the poor conditions, but how well these techniques can improve the accuracy of RBF-FD method is not yet fully explored in the literature. Therefore, the study of the RBF-FD method with these techniques will be included in this work as well.

2. Methodology

2.1. RBF Collocation Method

In this section, the global RBF scheme for solving the PDE problem called the RBF collocation method is introduced. Let $\Omega \subset \mathbb{R}^d$ be a d -dimensional domain and $\partial\Omega$ be the boundary of the domain. Consider the following elliptic PDE problem defined by:

$$\mathcal{L}u(x) = f(x) \quad \text{in } \Omega, \quad (3)$$

$$u(x) = g(x) \quad \text{on } \partial\Omega, \quad (4)$$

where $\mathcal{L}$ is a differential operator. Suppose that N is the number of center points in Ω , for which N_I points of them are the interior points. A straight RBF-based collocation method is applied by assuming the solution $u(x)$ as:

$$u(x) \approx \sum_{j=1}^N \lambda_j \phi(\|x - x_j\|_2, \epsilon).$$

Consequently, collocation with the PDE at the interior points and the boundary condition at the boundary points provide the following results:

$$\begin{aligned} \mathcal{L}u(x_i) &\approx \sum_{j=1}^N \lambda_j \mathcal{L}\phi(\|x_i - x_j\|_2, \epsilon) = f(x_i), \quad i = 1, 2, \dots, N_I, \\ u(x_i) &\approx \sum_{j=1}^N \lambda_j \phi(\|x_i - x_j\|_2, \epsilon) = g(x_i), \quad i = N_I + 1, N_I + 2, \dots, N. \end{aligned}$$

The coefficients $\{\lambda_j\}_{j=1}^N$ can be solved from the corresponding system of equations with the coefficient matrix structured as:

$$\begin{bmatrix} \mathcal{L}\phi \\ \phi \end{bmatrix} [\lambda] = \begin{bmatrix} f \\ g \end{bmatrix}.$$

2.2. RBF-FD Method

We will next present an outline of the RBF along with finite-difference (RBF-FD) formulation for solving PDE Equation (3).

First, let us look at a classical central finite difference method for approximating the derivative of function $u(x, y)$ with respect to x . Let the derivative at any grid point (i, j) of rectangular grid be written as

$$\left. \frac{\partial u}{\partial x} \right|_{(i,j)} \approx \sum_{k \in \{i-1, i, i+1\}} w_{(k,j)} u_{(k,j)}, \quad (5)$$

where $u_{(k,j)}$ is the function value at the grid point (k, j) , the unknown coefficients $w_{(k,j)}$ are computed using polynomial interpolation or Taylor series, while the set of nodes $\{(i-1, j), (i, j), (i+1, j)\}$ is a stencil in the finite difference literature. For any scattered nodes, the restriction of the structured grid is overcome by setting the approximation of the function derivative to be a linear combination of function values on scattered nodes in the stencil; however, the methodology for computing the

coefficients of finite difference formulas for any scattered points with dimensions of more than one has the problem of well-posedness for polynomial interpolation [16]. Thus, the combination of RBF and finite difference methodology (RBF-FD) is introduced to overcome the well-posedness problem.

To derive RBF-FD formulation, recall that, for a given set of distinct nodes $x_i \in \mathbb{R}^d, i = 1, 2, \dots, n$, and a corresponding function values $u(x_i), i = 1, \dots, n$, the RBF interpolation is of the form:

$$u(x) \approx \sum_{j=1}^n \lambda_j \phi(\|x - x_j\|_2, \varepsilon). \quad (6)$$

The RBF interpolation Equation (6) can alternatively be written in Lagrange form as:

$$u(x) \approx \sum_{j=1}^n \chi(\|x - x_j\|_2) u(x_j), \quad (7)$$

where $\chi(\|x - x_j\|_2)$ satisfies the following condition:

$$\chi(\|x - x_j\|_2) = \begin{cases} 1 & \text{if } x = x_j, \\ 0 & \text{if } x \neq x_j, \end{cases} \quad j = 1, \dots, n.$$

The key idea of the RBF-FD method is to approximate the linear differential operator of function $\mathcal{L}u(x)$ at each interior node as a linear combination of the function values at the neighboring node locations. For example, $\mathcal{L}u(x)$ at any node, say x_1 is approximated by an RBF interpolation Equation (7) with centres placed on the node itself and some $n - 1$ nearest neighbor nodes of x_1 , say $x_2, x_3, \dots, x_n$. These set of n nodes is called the stencil or support region for the node x_1 . For deriving RBF-FD formula at the node x_1 , we approximate $\mathcal{L}u(x_1)$ by taking the linear differential operator $\mathcal{L}$ on the both sides of the RBF interpolation Equation (7) i.e.,

$$\mathcal{L}u(x_1) \approx \sum_{j=1}^n \mathcal{L}\chi(\|x_1 - x_j\|_2) u(x_j).$$

As in finite difference methodology, this approximation can be rewritten in the form:

$$\mathcal{L}u(x_1) \approx \sum_{j=1}^n w_{(1,j)} u(x_j), \quad (8)$$

where the RBF-FD weights $w_{(1,j)} = \mathcal{L}\chi(\|x_1 - x_j\|_2), j = 1, \dots, n$.

In practice, the main difference between finite difference methodology and RBF-FD is how to compute the weights $w_{(1,j)}$ in Equation (8). While finite difference method enforces Equation (8) to be exact for polynomials $1, x, x^2, \dots, x^{n-1}$, the RBF-FD weights are computed by assuming that the approximations Equation (8) become exact for all the RBFs ϕ that are centered at each node in stencil, i.e., we assume that Equation (8) becomes exact for function $u(x) = \phi(\|x - x_k\|_2, \varepsilon), k = 1, 2, \dots, n$. In the RBF-FD case, this assumption leads to an $n \times n$ linear system:

$$\begin{bmatrix} \phi(\|x_1 - x_1\|_2, \varepsilon) & \phi(\|x_2 - x_1\|_2, \varepsilon) & \dots & \phi(\|x_n - x_1\|_2, \varepsilon) \\ \phi(\|x_1 - x_2\|_2, \varepsilon) & \phi(\|x_2 - x_2\|_2, \varepsilon) & \dots & \phi(\|x_n - x_2\|_2, \varepsilon) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|x_1 - x_n\|_2, \varepsilon) & \phi(\|x_2 - x_n\|_2, \varepsilon) & \dots & \phi(\|x_n - x_n\|_2, \varepsilon) \end{bmatrix} \begin{bmatrix} w_{(1,1)} \\ w_{(1,2)} \\ \vdots \\ w_{(1,n)} \end{bmatrix} = \begin{bmatrix} \mathcal{L}\phi(\|x_1 - x_1\|_2, \varepsilon) \\ \mathcal{L}\phi(\|x_1 - x_2\|_2, \varepsilon) \\ \vdots \\ \mathcal{L}\phi(\|x_1 - x_n\|_2, \varepsilon) \end{bmatrix}.$$

Note that the system matrix is invertible, and, therefore, the RBF-FD weight can always be calculated, while the system matrix for the finite difference case is not guaranteed to be invertible.

After we obtain RBF-FD weights, substituting the approximation Equation (8) into the elliptic PDE Equation (3) at node x_1 gives:

$$\sum_{j=1}^n w_{(1,j)} u(x_j) = f(x_1).$$

After that, we repeat this procedure at each of the interior nodes and substitute the approximations into the elliptic PDE Equation (3). Then, we obtain:

$$\sum_{j=1}^n w_{(i,j)} u(x_j) = f(x_i), \quad i = 1, 2, \dots, N_I, \quad (9)$$

where N_I is the number of interior points. After substituting the function values $u(x_j)$ in Equation (9) with boundary condition whenever x_j is the boundary point, we obtain a linear system of equations, which can be written in matrix form as:

$$Bu = F, \quad (10)$$

where u is the vector of the unknown function at all the interior nodes. Note that the matrix B is sparse, well-conditioned and can hence be effectively inverted.

3. Main Result

3.1. Solution Accuracy for Elliptic PDEs

The main result section provides better understanding on the computational process. Let $\Omega \subset \mathbb{R}^2$ be a 2-dimensional domain and $\partial\Omega$ be the boundary of Ω . The elliptic PDE that we want to investigate is presented as the following:

$$\nabla^2 u(\underline{x}) + a(x, y)u_x(\underline{x}) + b(x, y)u_y(\underline{x}) + c(x, y)u(\underline{x}) = f(\underline{x}), \quad \underline{x} \text{ in } \Omega, \quad (11)$$

$$u(\underline{x}) = g(\underline{x}), \quad \underline{x} \text{ on } \partial\Omega. \quad (12)$$

In numerical experiments, the solution $u(\underline{x})$ is known and normalized so that $\max_{\Omega} |u(\underline{x})| = 1$. Supposing that the RBF approximation is denoted by $s(\underline{x}, \varepsilon)$, then the error in a max norm is measured by:

$$\text{Max Error}(\varepsilon) = \max_{\Omega} |s(\underline{x}, \varepsilon) - u(\underline{x})|. \quad (13)$$

3.2. Results: A Comparison of a Global RBF Collocation Method and RBF-FD Method with Various RBFs

In this section, the global RBF collocation method and RBF-FD method will be first investigated with several RBFs by applying them to the following elliptic PDE:

$$\nabla^2 u - Ru_x = f(x, y), \quad (14)$$

where $R = 3$, $f(x, y)$ and the Dirichlet boundary condition are computed from the exact solution as shown in Equation (15):

$$u(x, y) = \exp(-0.5Rx) \sin(0.5Ry). \quad (15)$$

For this problem, we consider the PDE with the computational and evaluational domain of a unit disk with 394 uniformly scattered data points as shown in Figure 1. In our first experiment, we compare the results from using the implementation of global RBF scheme and RBF-FD method with stencil size = 13. The results of the computation of the global RBF collocation method and RBF-FD method, which is the local RBF scheme, are shown in Figure 2.

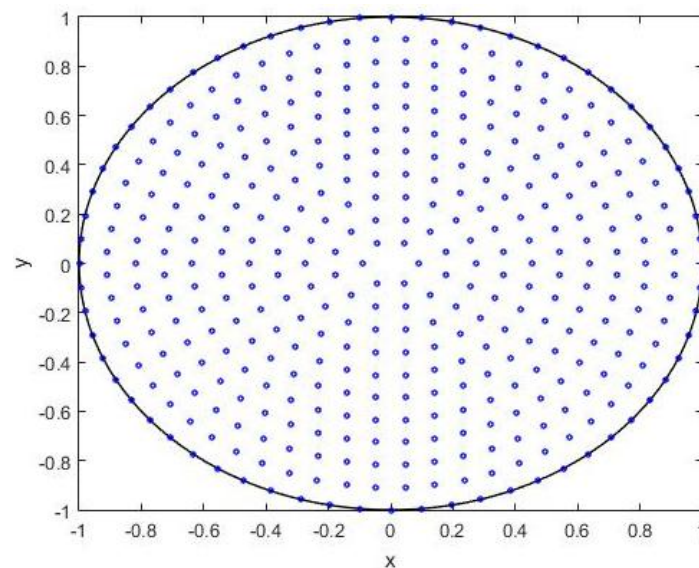


Figure 1. The computational and evaluational domain of a unit disk with 394 uniformly scattered data points.

Figure 2a shows the results of the computation of RBF collocation using IQ, IMQ, MQ, and GA radial basis functions, and Figure 2b shows the results of the computation of the RBF-FD method. Considering the global RBF scheme, all four of RBF seem to provide the same level of accuracy and fluctuated graph of max error for small values of shape parameter, while there is a little bit of difference in accuracy and smooth curve of the graph for the large values of shape parameter. The IQ, IMQ and MQ RBFs have similar behaviour of error's curves; however, the GA radial basis function has a different behaviour of error and gives the least error over the considered range of shape parameter for this problem. Note that all of them are spectrally accurate and the graph of error is oscillated since the small shape parameter makes the system matrix of global RBF collocation ill-conditioned, and, therefore, the numerical solution is computed inaccurately. This ill-conditioning problem will be more severe if more nodes are used. Due to this fact, the global RBF-scheme does not suit large scale problems.

For the RBF-FD case, all four of RBFs seem to provide the same level of accuracy for all shape parameters in the considered range. In particular, all of them have similar behaviour of error graphs and algebraic accuracy. For the large values of shape parameter, there is no significant differences of error between global and local RBF methods; however, the error of the global RBF scheme is more accurate by about two digits of accuracy than the RBF-FD method for the small values of shape parameter. Note that the graph of the error for the RBF-FD method does not oscillate for the small shape parameter because the system matrix at each node has better conditioning than the global RBF scheme for small stencil size. This fact makes the RBF-FD approach more appropriate for PDE problems with a large number of nodes.

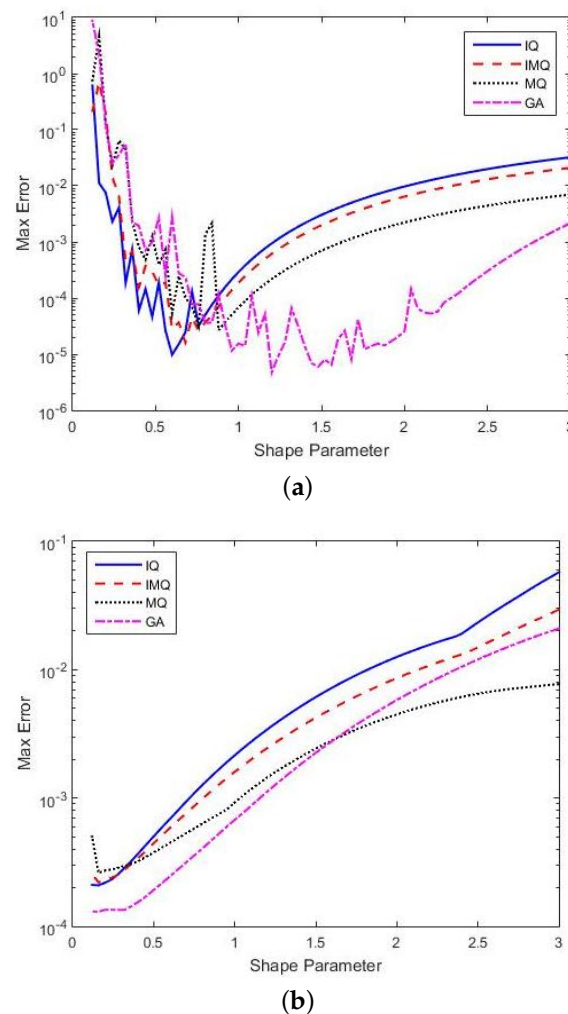


Figure 2. The graphs of max error are presented as a function of shape parameter. They compare max error of the results of global radial basis function (RBF) and radial basis function-finite difference (RBF-FD) methods. (a) global RBF scheme; (b) local RBF scheme.

3.3. Results: A Comparison of the RBF-FD Method with Different Numbers of Nodes and Stencils

From the previous section, the results show that the accuracy of RBF-FD is less than the accuracy of the global RBF scheme for the same number of nodes. However, the RBF-FD method is applicable and efficient for large scale problems. We hope that the increasing of the number of nodes will also help improve the accuracy for the case of the global RBF method, with the conditions for system matrix rapidly increased as well for the case of the RBF scheme [7,17]. Additionally, we expect that the accuracy will be more accurate whenever the stencil size increases. To ascertain this idea, we will consider the elliptic PDE given by:

$$\nabla^2 u - R(u_x + u_y) = f(x, y). \quad (16)$$

In this case, $R = 2$, $f(x, y)$ and Dirichlet boundary condition are computed from the exact solution as shown in Equation (17):

$$u(x, y) = \frac{(e^{Rxy} - 1)}{(e^R - 1)}. \quad (17)$$

For numerical experiments, we consider this problem in the computational and evaluational domain of a unit disk with uniformly scattered points. In our first experiment, the number of nodes

are varied while the stencil size is fixed equal to 25, and then the effect of the number of nodes on the accuracy is compared. As for the second, the experiment is done in the opposite direction. The stencil sizes are varied while the number of nodes is fixed equal to 394, and then the effect of the stencil sizes on the accuracy is compared.

The results of computation using IQ-RBF when the number of nodes are varied and the stencil size is fixed are shown in Figure 3a. All of the results seem to provide the same level of accuracy for large values of shape parameter, while there are improvements on accuracy for the small shape parameter. These results provide a good simple way for improving accuracy. However, we need to be careful lest the ill-conditioning problem would arise if the stencil size is too large. For the case of varying the stencil sizes and the number of nodes being fixed as shown in Figure 3b, the results show that the more nodes used in the computation, the more accurate results can be achieved regardless of the shape parameter values. These results are also similar to the case of the global RBF method. In practical problems, this strategy of improving accuracy is very effective for the RBF-FD method with small enough stencil size since the sparse RBF-FD system matrix is well-conditioned and can be effectively inverted for a large number of nodes. However, this strategy does not suit a global RBF approach because the condition number of the system matrix will grow rapidly as the number of nodes increases and this will make the numerical solution inaccurate.

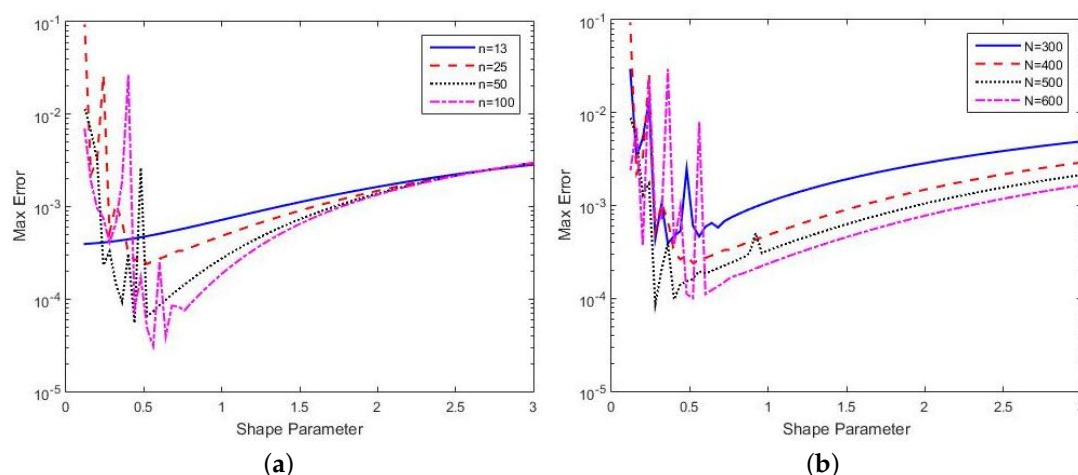


Figure 3. The graph of the max error of the RBF-FD method with different numbers of nodes and stencils. (a) varying number of stencils (n); (b) varying number of nodes (N).

3.4. Results: A Comparison of the RBF-FD Method with Strategies on PDE and Shape Parameter

In this section, the RBF-FD method with strategies on PDE and shape parameter are compared to the normal RBF-FD scheme. The elliptic PDE that we consider here is given by

$$\nabla^2 u + \exp(-(x+y))u = f(x,y), \quad (18)$$

where $f(x,y)$ and the Dirichlet boundary condition are computed from the exact solution as shown in Equation (19):

$$u(x,y) = \sinh(x^2 + y^2). \quad (19)$$

In this problem, the computations are implemented with the 395 uniformly scattered points and stencil size equal to 50.

- **Method 1 : Normal RBF-FD scheme**

As we discussed in Section 2.2, a straightforward RBF-FD method can be implemented for approximating the differential operator $\mathcal{L}$ by

$$\mathcal{L}u(x_j) \approx \sum_{i=1}^n w_{(j,i)} u(x_i),$$

for each interior node $x_j, j = 1, 2, \dots, N$. After obtaining the FD-weights, we substitute the function values $u(x_j)$ from the boundary condition, and then a system of equations as shown in Equation (10) can be formed which can be written in matrix form as

$$Bu = F,$$

where u is the vector of the unknown function value at all of the interior nodes. Note that the matrix B is sparse and well-conditioned and can hence be effectively inverted.

- **Method 2 : RBF-FD method with ghost nodes strategy**

One of the simple ways to improve accuracy of numerical solutions is applying the PDE on boundary nodes. According to the global RBF method, the error seems to be largest near the boundary of the computational domain. This problem can be fixed by getting more information there by applying the PDE at the boundary nodes [1]. In order to match the number of unknowns and equations, additional nodes, called ghost nodes, are added. The centers of these ghost nodes should be placed outside the boundary. For simplicity, we let the center points be denoted by z_j , where

$$z_j = \begin{cases} x_j, & j = 1, 2, \dots, N, \\ \text{a point outside } \Omega, & j = N + 1, N + 2, \dots, N + N_B, \end{cases}$$

where N_B is the number of boundary nodes. After that, a straightforward RBF-FD scheme is implemented by approximating the differential operator $\mathcal{L}$ by

$$\mathcal{L}u(x_j) \approx \sum_{i=1}^n w_{(j,i)} u(x_i),$$

for all node $x_j, j = 1, 2, \dots, N$. Note that the search for stencils must be applied to all nodes z_j . After obtaining the FD-weights and substituting the function values $u(x_j)$ from the boundary condition, a system of equations as shown in Equation (10) can be formed, which can be written in matrix form as

$$Bu = F,$$

where u is the vector of the unknown function value at all of the interior nodes and points outside the boundary. Note that the matrix B is sparse and well-conditioned and can hence be effectively inverted. In this problem, we use the additional 64 ghost nodes defined by $x_n = 1.1 \exp(\frac{2n\pi i}{64})$, $n = 1, 2, \dots, 64$.

- **Method 3 : RBF-FD method with preconditioning strategy**

A variation that may improve the accuracy of the numerical solution is using the different values of shape parameter on each stencil. According to numerical evidence, RBF methods usually provide better accuracy if their system matrix is ill-conditioned. For computation with double precision, the shape parameter can be selected for each stencil so that the condition number is between 10^{13} and 10^{15} [12] because the error curve of the corresponding range of the shape parameter is the smooth curve before beginning to oscillate after that, as shown in Figure 2a. The following pseudocode will be

used to select the shape parameter in this problem so that the condition number is in the desired range: $K = 0$, $K_{\min} = 1.0 \times 10^{13}$, $K_{\max} = 1.0 \times 10^{15}$, shape = initial shape, increment = 0.01

```
while  $K < K_{\min}$  or  $K > K_{\max}$  :
    form B
     $K$  = condition number of B
    if  $K < K_{\min}$ 
        shape = shape - increment
    elseif  $K > K_{\max}$ 
        shape = shape + increment.
```

Figure 4 shows the results of computation using IQ-RBF with strategies on PDE and shape parameter, compared to the normal RBF scheme. The ghost nodes strategy seems to provide the same level of accuracy for large values of shape parameter, while there is a significant improvement on accuracy for the small shape parameter. Note that using the additional ghost nodes strategy does not significantly affect the condition number of the system matrix. Therefore, for the practical problem, this approach can be implemented effectively to improve the error of numerical solution near the boundary. For the preconditioning strategy, the results show that, regardless of the initial shape parameter, this strategy can provide acceptable accurate results. Therefore, this approach is an applicable and effective strategy for choosing the shape parameter for the large scale practical problems.

After we choose the shape parameter, a normal RBF-FD method can be implemented for approximating the derivative and then forming a linear system for finding the approximate solution.

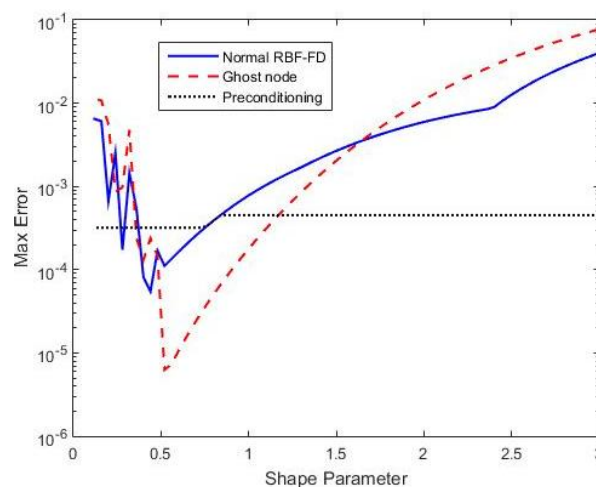


Figure 4. The graph shows the comparing of max error among the RBF-FD method with ghost nodes, the RBF-FD method with preconditioning, and the normal RBF-FD method. In the case of the RBF-FD method with preconditioning, we plot the max error against the initial shape parameter.

3.5. Results: A Comparison of RBF-FD Method with Additional Strategies for Reducing the Ill-Conditioning Problem

In the previous section, the RBF-FD method with ghost nodes and preconditioning strategies is applied for obtaining acceptable accurate results and reducing the accuracy of numerical solutions near the boundary. This section aims to improve the accuracy through another manipulation, i.e., by focusing on reducing the ill-conditioned problem for small values of shape parameters. To clarify this idea, we will investigate two alternative strategies called regularization and extended precision floating point arithmetic. The following elliptic PDE will be studied here :

$$\nabla^2 u + xu_x + yu_y - (4x^2 + y^2)u = f(x, y), \quad (20)$$

where $f(x, y)$ and the Dirichlet boundary condition are computed from the same solution as shown in Equation (21)

$$u(x, y) = \exp(-(x^2 + 0.5y^2)). \quad (21)$$

- **Strategy 1 : Regularization**

For each stencil, a simple regularization technique can be used to reduce the effects of the poor conditioning of the RBF system matrix by solving the regularized system

$$By = f \quad (22)$$

instead of the following system

$$Ax = f, \quad (23)$$

where $B = A + \mu I$. The regularization parameter μ is a small positive constant and I is an identity matrix. This approach can be used to solve effectively the ill-conditioning problem of the global RBF collocation method and give accurate results for small values of shape parameters, as shown in [15,18]. In this problem, we set $\mu = 5 \times 10^{-15}$.

- **Strategy 2 : Extended precision floating point arithmetic**

Recently, computer systems usually implement the IEEE 64-bit floating point arithmetic standard, which is referred to as double precision. According to [7], the global RBF method gains more benefit when it is implemented with extended precision, which provides more digits of accuracy than double precision. In particular, this approach also keeps the simplicity of the RBF method. In this work, the authors used the Multiprecision Computing Toolbox for Matlab (MCT) [19] to provide extended precision for the Matlab program and used quadruple precision (34 digits of precision) in this problem.

For numerical experiments, this problem is implemented with the computational and evaluational domain of a unit disk with 394 scattered data points and stencil size equal to 50. Figure 5 shows the results of computation using IQ-RBF with regularization and extended precision floating point arithmetic strategies, compared to the normal RBF-FD method. Unfortunately, the regularization seems to provide the same level of accuracy as the RBF-FD method without regularization, although this technique provides a significant improvement on accuracy compared to the global RBF method. However, for the extended precision strategy, the results show that this technique provides a significant improvement on accuracy at the same level as the global RBF method.

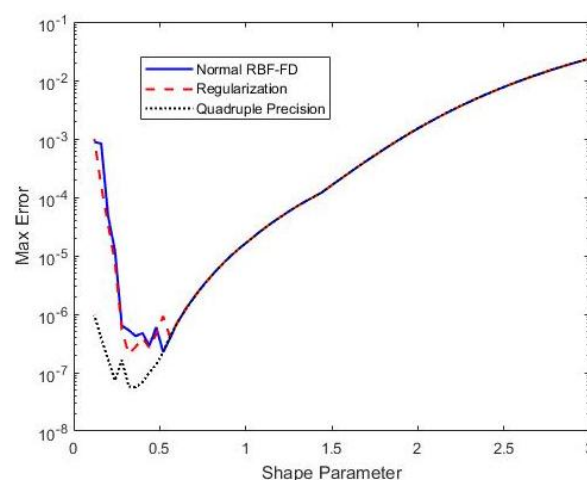


Figure 5. The graphs of max error of the RBF-FD method with regularization and extended precision strategies, compared to the normal RBF-FD method.

4. Future Work

RBF-FD method can also be easily extended to solve time-dependent PDEs by using the method of line such as the Runge–Kutta method (RK4) to solve the problem after the PDE is discretized in space with the RBF-FD method. For example, consider the diffusion problem given by

$$u_t = \nu \nabla^2 u + \gamma u^2(1 - u), \quad (24)$$

where $\nu = 0.5, \gamma = 2$, with the Dirichlet boundary and initial condition is taken from the exact solution

$$u(x, y) = \frac{1}{1 + \exp(x + y - t)}. \quad (25)$$

To attack this problem, the Laplace operator ∇^2 is discretized by the RBF-FD method as we mentioned in Section 2.2. After we obtain the results, the remaining part will be ODE, the same as the following form

$$u_t = Du + \gamma u^2(1 - u), \quad (26)$$

where D is the RBF-FD system matrix, which is obtained from discretization of the Laplace operator and u is the vector of the unknown function at all of the interior nodes. At this point, the fourth-order Runge–Kutta method (RK4) can be applied to solve the problem. The results that are implemented with the computational and evaluational domain of a unit disk with 394 scattered data points, stencil size equal to 13, and shape parameter equal to 0.6 are shown in Table 2.

Table 2. The results of solving the partial differential equation (PDE) Equation (24) with the radial basis function-finite difference (RBF-FD) method at $t = 5, 10, 15$.

t	5	10	15
Max Error	2.19×10^{-4}	2.57×10^{-4}	2.71×10^{-4}

Note that the RBF-FD method also gives reasonable numerical accuracy for time-dependent problems. However, many questions about eigenvalue stability of the RBF-FD method for time-dependent PDE problems in general have not yet been fully explored in the literature and need more study in future work.

5. Discussion

By using the RBF-FD method, we were able to solve large scale PDE problems due to the fact that RBF-FD has its own strengths related to feasibility and computational cost. We have found that accuracy can still be achieved at the same level of global RBF in the ill-conditioned system matrix for nodes. Although the RBF-FD method does not contain the same spectral accuracy as global RBF, it will give acceptable accuracy for large scale problems for which the global RBF can not be implemented. We also note that the accuracy can be improved by the increasing of nodes and stencil sizes. This strategy of improving accuracy is very effective with the RBF-FD method with small enough stencil size since the sparse RBF-FD system matrix is well-conditioned and can effectively be inverted for a large number of nodes. However, we need to be careful about an ill-conditioning problem arising if the stencil size is too large. The ghost node and preconditioning strategies are introduced to improve the accuracy of numerical solutions. Using the additional ghost node strategy does not significantly affect the condition number of the system matrix. Therefore, for the practical problem, this approach can be implemented effectively to reduce the errors of numerical solutions near boundaries. Results show that, regardless of the initial shape parameter, preconditioning strategy can provide acceptable accurate results. Improving the accuracy by reducing the ill-conditioning problem for small shape parameter is also an interesting idea. From experimental results, the extended

precision provides significant improvement in accuracy. Unfortunately, regularization strategy fails to yield better accuracy than that of the plain RBF-FD.

Table 3 shows the results of solving PDE Equation (20) by using the RBF-FD method at a shape parameter equal to 0.4 with additional different strategies, which are presented in this work. In this case, preconditioning and regularization techniques yield the same level of accuracy as the normal RBF-FD method, while the other strategies can improve accuracy by around one digit of accuracy. Note that all of these strategies can be used together as a mixed strategy for obtaining high accuracy. However, the issues about how to choose the optimal shape parameter for obtaining the most accurate results and how to solve the ill-conditioned problem for small values of shape parameter are still open problems needing further investigation.

Table 3. The results of solving the PDE Equation (20) by using the RBF-FD method at a shape parameter equal to 0.4 with different additional strategies.

Method	Max Error ($\times 10^{-7}$)
Normal RBF-FD ($N = 400, n = 25$)	42.4
Normal RBF-FD ($N = 400, n = 50$)	4.75
Normal RBF-FD ($N = 800, n = 25$)	8.80
Ghost Node ($N = 400, n = 25$, 64 ghost nodes)	5.63
Preconditioning ($N = 400, n = 25$)	40.5
Regularization ($N = 400, n = 25$)	38.1
Extended Precision ($N = 400, n = 25$)	6.34

6. Conclusions

In this paper, we have found that the combination of the Radial Basis Function and the finite difference (RBF-FD) method with either additional Ghost Node strategy or additional Extended Precision strategy can help improve the accuracy in solving the elliptic partial differential equations, while the Preconditioning and Regularization strategies are found to be of little avail for our purpose.

Acknowledgments: This research was funded by the Faculty of Science, Mahidol University, Bangkok, Thailand.

Author Contributions: Suranon Yensiri and Ruth J. Skulku designed and developed the research. Suranon Yensiri was responsible for simulating the results by the MATLAB programming (version 9.1.0.441655 (R2016b)), The MathWorks Inc., Natick, Massachusetts, United States). Both authors wrote this paper and contributed to editing and revising the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Larsson, E.; Fornberg, B. A numerical study of some radial basis function based solution methods for elliptic PDEs. *Comput. Math. Appl.* **2003**, *46*, 891–902.
2. Li, J.; Cheng, H.-D.; Chen, C.-S. A comparison of efficiency and error convergence of multiquadric collocation method and finite element method. *Eng. Anal. Bound. Elem.* **2003**, *27*, 251–257.
3. Sarra, S.A.; Kansa, E.J. *Multiquadrics Radial Basis Function Approximation Methods for The Numerical Solution to Partial Differential Equations*; Tech Science Press: Duluth, GA, USA, 2010.
4. Fasshauer, G.E. *Meshfree Approximation Methods with Matlab*; World Scientific: Singapore, 2007.
5. Micchelli, C. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constr. Approx.* **1986**, *2*, 11–22.
6. Hon, Y.C.; Schaback, R. On unsymmetric collocation by radial basis functions. *Appl. Math. Comput.* **2001**, *119*, 177–186.
7. Sarra, S.A. Radial basis function approximation methods with extended precision floating point arithmetic. *Eng. Anal. Bound. Elem.* **2011**, *35*, 68–76.
8. Bayona, V.; Flyer, N.; Lucas, G.M.; Baumgaertner, A.J.G. A 3-D RBF-FD solver for modeling the atmospheric global electric circuit with topography (GEC-RBFFD v1.0). *Geosci. Model Dev.* **2015**, *8*, 3007–3020.

9. Shan, Y.; Shu, C.; Qin, N. Multiquadric finite difference (MQ-FD) method and its application. *Adv. Appl. Math. Mech.* **2009**, *5*, 615–638.
10. Tolstykh, A.I.; Lipavskii, M.V.; Shirobokov, D.A. High-accuracy discretization methods for solid mechanics. *Arch. Mech.* **2003**, *55*, 531–553.
11. Tolstykh, A.I.; Shirobokov, D.A. On using radial basis functions in finite difference mode with applications to elasticity problems. *Comput. Mech.* **2003**, *33*, 68–79.
12. Sarra, S.A. A Local Radial Basis Function method for Advection-Diffusion-Reaction equations on complexly shaped domains. *Appl. Math. Comput.* **2012**, *218*, 9853–9865.
13. Hosseini, B.; Hashemi, R. Solution of Burgers' equation using a local-RBF meshless method. *Int. J. Comput. Meth. Eng. Sci. Mech.* **2011**, *12*, 44–58.
14. Shankar, V.; Wright, G.B.; Fogelson, A.L.; Kirby, R.M. A radial basis function (RBF) finite difference method for the simulation of reaction-diffusion equations on stationary platelets within the augmented forcing method. *Int. J. Numer. Methods Fluids* **2014**, *75*, 1–22.
15. Sarra, S.A. Regularized symmetric positive definite matrix factorizations for linear systems arising from RBF interpolation and differentiation. *Eng. Anal. Bound. Elem.* **2014**, *44*, 76–86.
16. Wright, G.B.; Fornberg, B. Scattered node compact finite difference-type formulas generated from radial basis functions. *J. Comput. Phys.* **2006**, *212*, 99–123.
17. Sarra, S.A. A numerical study of the accuracy and stability of symmetric and asymmetric RBF collocation methods for hyperbolic PDEs. *Numer. Methods Partial Differ. Equ.* **2008**, *24*, 670–686.
18. Sarra, S.A. The Matlab radial basis function toolbox. *J. Open Res. Softw.* **2017**, *5*, 8, doi:10.5334/jors.131.
19. Advanpix. Multiprecision Computing Toolbox for Matlab, Version 4.4.1.12566 for 64-bit Windows. Available online: <http://www.advanpix.com> (accessed on 17 August 2017).



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).