Fall 2019

Localized Method of Approximate Particular Solutions for Solving Fourth-Order PDEs

Lionel Amuzu
University of Southern Mississippi

LOCALIZED METHOD OF APPROXIMATE PARTICULAR SOLUTIONS FOR SOLVING FOURTH-ORDER PDES

by

Lionel Elikem Amuzu

A Thesis
Submitted to the Graduate School,
the College of Arts and Sciences
and the School of Mathematics and Natural Sciences
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Masters of Science

Approved by:

Dr. C.S. Chen, Committee Chair
Dr. James Lambers
Dr. Huiqing Zhu

Dr. C.S. Chen	Dr. Bernd Schroeder	Dr. Karen S. Coats
Committee Chair	Director of School	Dean of the Graduate School

December 2019

ABSTRACT

In the past, dealing with fourth-order partial differential equations using the Local Method was not reliable due to difficulties in solving them directly. An approach such as splitting these equations into two Poisson differential equations was adopted to alleviate such challenges. However, this has a limitation since it is only applicable to Dirichlet and Laplace boundary conditions. In this paper, we solve fourth-order PDEs directly using the LMAPS. The improvement on the accuracy of this Method was as a result of the proposed distribution of boundary conditions to alternating boundary points. And, also the use of suitable shape parameter; calculated using LOOCV(Leave-One-Out-Cross-Validation) Algorithm [23]. The effectiveness of this Method was evident when we compared the results from two numerical examples.

ACKNOWLEDGMENTS

I want to express my gratitude towards my supervisor, Dr. C.S Chen, for the time he invested in supervising my thesis. I have learned a lot during this time, and I look forward to learning more from you.

I also want to thank my committee, Dr. Lambers and Dr. Zhu, whom I have studied a lot from in class and during my research. I appreciate working with you.

I also want to acknowledge my mates in the MathZone.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

LIST OF TABLES

Table

LIST OF ABBREVIATIONS

AME	-	Absolute Maximum Error
IMQ	-	Inverse Multiquadrics
LOOCV	-	Leave-One-Out-Cross-Validation
LMAPS	-	Localized Method of Approximate Particular Solutions
MAPS	-	Method of Approximate Particular Solutions
MQ	-	Multiquadrics
PDE	-	Partial Differential Equation
RBF	-	Radial Basis Function
RMSE	-	Root Mean Square Error

NOTATION AND GLOSSARY

General Usage and Terminology

The usage of various notations in this paper are relatively common in mathematics and computing. The techniques employed in this research may have vast applications, but this focused on PDEs. In several cases, these fields tend to use different preferred notation to indicate the same idea. In this paper, some symbols have been used in published literature aside from the standard terminology.

For the sets of real numbers, we used $\mathbb{R}$. Also, capital boldfaced greek letters, e.g.,$\boldsymbol{\Theta}$, $\boldsymbol{\Xi}$ and $\boldsymbol{\Pi}$ are vectors and $\mathcal{P}_{\boldsymbol{\Theta}}, \mathcal{P}_{\boldsymbol{\Xi}}$ and $\mathcal{P}_{\boldsymbol{\Pi}}$ denote matrices. Functions that in lower case roman letters such as f, g, h are real-valued functions. The Calligraphic letters $\mathcal{L}$ and $\mathcal{B}$ denote partial differential operators. Lower case roman letter, i, j, k and n are indices of a vector or matrix.

u is a function of two variables x and y defined on some domain Ω of $\mathbb{R}^2$. At (x, y), the Laplacian of u is defined by

$$\Delta(u) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

Also

$$\Delta^2(u) = \Delta \cdot \Delta(u) = \frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4}$$

The solution to a PDE is not unique, therefore to know one solution of a PDE, it is necessary to impose boundary or initial conditions. In the domain Ω, $\partial\Omega$ denotes the boundary.

$||\cdot||$ represents Euclidean norm, but vectors and matrices are closed in square brackets, $[\cdot]$.

Chapter 1

BACKGROUND

1.1 Introduction

During the past decades, the method of particular solution(MAPS) has gained its popularity as applied to various partial differential equations[4, 11, 30], which intends to solve problems in physics, engineering, etc. This range of problems is most likely associated with higher-order equations such as fourth-order PDEs. Many strategies have been developed and implemented to solve fourth-order PDEs. Some of which are; approximation of biharmonic equation using the finite difference scheme, applied to general irregular planar domains by Ben-Artzi *et al.* [2]. Ben-Artzi *et al.* approximated $\Delta^2\Phi$ at a grid point by interpolating the data on the (irregular) stencil by a polynomial of degree six. Yao *et al.* [30] compared three meshless methods. Method of particular solution(MPS), method of fundamental solutions (MFS-MPS) and the Kansas method and concluded that the MFS-MPS outperformed the MPS and Kansa's method

Upon all these achievements, there are still difficulties in solving fourth-order PDEs using local methods. One of the challenges has been with the approximation of fourth-order derivative using a small number of source points. Various decoupling techniques proposed by researchers to split the fourth-order PDEs into two decoupled second-order PDEs [20, 28, 1] avoided this problem. Though, this strategy is only possible in applying the Dirichlet and Laplace boundary conditions.

There is a high accuracy guaranteed when using RBFs. However, the choice of a shape parameter c that reduces the approximation error has been a severe problem for researchers. There have been various strategies for use [19, 16, 3]. As c approaches zero, the resultant matrix gets ill-conditioned, whereas the error minimizes. This interdependence is called the uncertainty principle. Rippa [23] implemented the LOOCV to find a suitable shape parameter that does provide not only an accurate approximation but also a well-conditioned matrix.

In fourth-order PDEs, there is a need for two boundary conditions. We impose these

boundary conditions in an alternating sense on different boundary points, and this makes the resultant matrix square, as such, improves the accuracy and stability of our results. In [1, 20, 28], a system of the non-squared matrix had to be solved.

Chapter 2, we briefly introduce meshless methods, the Definition of RBFs, and some commonly used RBFs. In Section 2.2, we present the LOOCV and how to use it to determine the optimal shape parameter. In Section 2.3, we show how to implement the variable shape parameter. Then in section 2.4, we introduce the decoupling approach.

This thesis is organized as follows: Chapter 3 is the formulation of LMAPS and 4 demonstrate our numerical implementation, examples and results. Finally, in chapter 5 is the conclusion of the method.

1.2 Literature Review

The LMAPS have been in use for the first time in 2011 by Yao *et al.* [29] when the difficulty of using globally applying MAPS was identified. Yao *et al.* realized that the global method results in a large dense matrix as such, restricting the number of collocation points. The larger the number of collocation points the denser the matrix and the extremely difficult it becomes to solve large-scale science and engineering problems. Hence, they develop the LMAPs that allows the use of small neighboring points (Local point) which are a subset of the collocation points to approximate the solution of PDEs. Due to the difficulty of solving fourth-order PDEs directly, in [20] Li *et al.* Implement the technique of splitting the biharmonic equation into two Poisson equation after which the LMAPS was applied to the Poisson equations to approximate the solution.In [28] Yang et al. used local Kansas's method to the Berger equation using RBF. To succeed in solving higher order differential equations using localized RBF methods without difficulty; Yang et al. split the given equation into two second-order partial differential equations. They also used the LOOCV to find a desirable shape parameter of MQ and Matern RBFs. In the paper [17] the local Kansas method and the LMAPS to solve eigenvalue problem on the L-shape and irregular domain. Li et at [20]. present the LMAPS, in solving a 2D biharmonic equation in a bounded region. The scheme focus on decoupling the biharmonic problem into two Poisson equations, and then the LMAPS is applied to each Poisson equation to solve for numerical results. Chang *et al.* [4] solves fourth-order PDEs using two second-order closed-form Particular solutions. For Chang *et al.* to implement high degree polynomials which in the normal case are for ill-conditioning and extremely unstable. They adopt a multi-scale strategy to minimize the large condition number of the resultant matrix. Chen *et al.* in [6] they implemented the

MAPS using RBFs where elliptic PDE with variable coefficient was solved. Chen *et al.* used the dual reciprocity boundary element method (DRBEM) and their MAPS to arrange the elliptic PDE into a Poisson-type equation. Which made the PDE easier to be solved.

1.3 Aim of this Study

The goal of this research is to solve fourth-order PDEs with constant coefficient directly without splitting the PDE into decoupled second-order PDE and be able to produce a stable and accurate numerical result. We should achieve our purpose when we consider the following.

- Assign boundary conditions on alternating boundary point, making a square matrix.
- Use the LOOCV to find a suitable shape parameter instead of guessing
- Employ variable shape parameter over the constant shape parameter.

These three points when implemented properly should guarantee an excellent result.

Chapter 2

MESHLESS METHOD

Recently, methods such as the LMAPS, Kansas methods, MLS methods and radial basis functions are widely used to solve different kinds of problems in the fields of engineering and science, these methods are categorized as mesh-less methods. Unlike mesh methods like the FDM and FEM that require extensive meshing henceforth tedious and time consuming, mesh-less methods are simple, yet guarantees accurate results and does not require re-meshing. In [21], A mesh is defined as any of the open spaces or interstices between the strands of a net that is formed by connecting nodes in a predefined manner. whilst The meshfree method is used to establish a system of algebraic equations for the whole problem domain without the use of a predefined mesh or uses easily generable meshes in a much more flexible or "freer" manner.

In this charter we briefly introduce RBFs and list of some closed form particular solutions.

2.1 Radial Basis Functions

RBFs are used for approximating unknown functions with known data. They can be used to approximate solutions of partial differential equations with certain initial and boundary conditions. These approximations are generally multivariate functions, however, reduced to a scalar function, $\phi(r)$. Where r the Euclidean norm $||X||_2$, defined as a radial distance between the collocation points and centers. The recovery functions from meshless data is then obtained by the linear combination of radial basis function, ϕ,

$$u(X_i) = \sum_{k=1}^{N} a_i \phi(||X_i - X_k||_2) \quad 1 \leq i \leq m$$

The unknown multivariate function, u, is obtained by solving the linear system for the weight $\{a_k\}_i^N$ of the center, with a finite number of collocation points, $\{X_i\}_1^M$. The nodes $\{X_k\}_i^N$ are called centers.

It is important to choose the number of collocation points M to be greater than or equal to the number of centers N. However, to ensure easy solvability of the linear system we choose $M = N$. Note: $X = (x, y), \quad r = ||X - X_k||_2 = \sqrt{(x - x_k)^2 + (y - y_k)^2}$

Definition 2.1.1. A radial basics function ϕ on $[0\infty)$ defined in [5] is positive definite on $\mathbb{R}^d$, if for all choices of sets $X := \{x_1, ..., x_N\}$ of finitely many points $x_1, ..., x_N \in \mathbb{R}^d$ and arbitrary in the symmetric $N \times N$ matrices $\phi(||X_i - X_k||_2)$ are positive definite.

For a system of equations containing rbf to be solved, it needs to satisfy the above definition. In table 2.1 are some few types of RBFs mostly used by scientists and engineers.

Table 2.1: Commonly used RBFs

Name	$\phi(r)$
Gaussian	$\exp(-cr^2)$
Multiquadrics	$\sqrt{r^2 + c^2}$
Inverse multiquadrics	$\frac{1}{\sqrt{r^2+c^2}}$
Conical	r^{2n-1}
Thin plate spline	$r^2 \log r$
Matern	$(cr)^n K_n(cr)$

Previous researches have shown how the use of RBFs improved accuracy. Despite RBFs great performance, its accuracy depends on the shape parameter c.There are various techniques[9, 14] for finding a suitable shape parameter. In the next section we explained, the leave-one-out-cross-validation, the technique we used to determine the optimal shape parameter.

2.2 Loocv

Many RBf users have successfully found a way to reduce the condition number of the coefficient matrix when solving system linear equations [10, 12, 13]. However, there is always an approximation error associated with the value of the shape parameter. It is crucial to find the optimal shape parameter in order to minimize the error. The standard technique

commonly used is the method of cross-validation. wiki: "Cross-validation is a model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set". In other words, the method of cross-validation is used to estimate how accurately a predictive model can perform. The particular case of Cross-validation we implemented in this paper is the LOOCV. LOOCV "uses a single observation from the original sample as the validation data, and the remaining observations as the training data". The procedure is repeated such that each observation in the sample is used once as the validation data. The resultant vector is then used to find the optimal c. Let

$$x^{[k]} = [x_1, x_2, ..., x_{k-1}, x_{k+1}, ..., x_N]^{[T]},$$

be the datasetes with the validation point x_k removed. The radial basis function interpolant $\hat{u}^{[k]}$ to f is given by

$$\hat{u}^{[k]}(x) = \sum_{i=1}^{N-1} a_i^{[k]} \phi\left(||x - x_i||_2\right)$$

where

$$\hat{u}^{[k]}(x_i) = f_i, \quad i \in [1,k) \cup (k,N]$$

Then the error e_k is calculated as difference in the approxiamtion and the exact function value at x_k,

$$e_k = f(x_k) - \hat{u}^{[k]}$$

The accuracy of the overall datasets is then determined by the norm of the error vector $e = [e_1, e_2, ..., e_N]^{[T]}$ which is the cost function. It is obtained by removing, in turn, one of the datasets and comparing the approximated value with the known value at the removed points. Minimizing the cost function $||e||$ gives the optimal c.

Rippa [23] proposed a very simple and efficient way to implement the LOOCV algorithm, calculating the error using the formula

$$e_k = \frac{a_k}{A_k^{-1}} \tag{2.1}$$

Where a_k is the k^{th} coefficient of $\hat{u}$ for all datasetes and A_k^{-1} is the k^{th} diagonal element of the inverse of the corresponding interpolation matrix. We can find a suitable shape parameter by using the MatLab function `fminbnd` to minimize the cost function for c obtained from (2.1).

The implementation of the idea on how to determine the cost function for c is as follows.

```
1 function ceps=costEps(ep,parsol,D,DM)
2 rhs = IMQ(ep, D);
3 mq = MQ(ep, DM);
4 A = parsol(ep, DM, mq);
5 invA = inv(A);
6 errorvector = (A\rhs)./diag(invA);
7 ceps=norm(errorvector);
8 return
```

Where `parsol` is the particular solution (fourth-order in our case) based on the shape parameter (ep), a vector of distances D and a distant matrix DM, which is the pairwise distance between collocation points. Furthermore, IMQ and MQ are RBFs; inline-five, the function `inv` finds the inverse of the square Matrix (A). Inline-six, we have computed all entries of the error vector at a goal. The `diag` and `norm` functions find the diagonal entries of matrix and the Euclidean norm of a vector, respectively.

Now to determine the optimal shape parameter by using function `fminbnd` to find the minimum of `costEPs`.

$$ep_{min} = \text{fminbnd}(@(\text{ep})\ \text{costEPs}(\text{ep}, \text{parsol}, \text{D}, \text{DM}), \text{c}_{\min}, \text{c}_{\max}) \tag{2.2}$$

where c_{min}, c_{max} is the initial search guessed interval for the optimal shape parameter.

2.3 Variable shape parameter

A variable shape parameter [24] strategy refers to uses a possibly different value of the shape parameter at each center. The theoretical complexity of the variable shape parameter is rather too difficult to explain [25]. However, there is no doubt it has proven to be effective [15, 27] when implemented. One importance of using the Variable shape parameter is creating distinct elements in the sparse matrix as such, and it reduces the condition number. Now that the optimal shape parameter is known from the previous section, it is possible to now construct the Variable shape parameter by using random shape strategy in [24]

$$ep_j = ep_{min} + (ep_{max} - ep_{min})\ \text{rand}(1, \text{N}) \tag{2.3}$$

Note: ep_{min} is the optimal shape parameter obtained in the previous section and $ep_{max} = ep_{min} + \delta \quad (\delta \in +\mathbb{R})$. `rand` in MATLAB returns an array of random numbers. `rand(1,N)` in equation (2.3) generates a `1-by-N` vector.

Instead of using `rand` equation 2.3 we used `haltonset` and `net` both MATLAB functions.The former constructs a quasi random point set from Halton sequence whilst the later generates quasi-random point set. So the new equation becomes

$$ep_j = ep_{min} + (ep_{max} - ep_{min})\ \text{net}(\text{p}, \text{ns}) \tag{2.4}$$

where $p = halt onset$. Readers should check MATLAB Documentation for more information of these functions. There are few other shape parameter strategies [24, 25, 27] such as;

1. Trigonometry shape parameter strategy
2. Exponential shape parameter strategy

2.4 Preliminary

Let us consider the boundary value problem of biharmonic equation

$$\Delta^2 u(x,y) = f(x,y) \qquad x,y \in \Omega \tag{2.5}$$
$$\Delta u(x,y) = g(x,y) \qquad x,y \in \partial\Omega_L \tag{2.6}$$
$$u(x,y) = h(x,y) \qquad x,y \in \partial\Omega_D \tag{2.7}$$

Where Ω is a two-dimensional domain bounded by a surface $\partial\Omega$ which consist of two parts, $\partial\Omega = \partial\Omega_L \cup \partial\Omega_D$ and $\partial\Omega_L \cap \partial\Omega_D = \emptyset$.

In order to avoid the difficulty in solving the biharmonic equation directly, implementing a scheme in [20] is necessary to split (2.5)–(2.7) into two Poisson equations by substituting an intermediate function $v = \Delta u$ as shown below:

$$\Delta v(x,y) = f(x,y) \qquad x,y \in \Omega$$
$$v(x,y) = g(x,y) \qquad x,y \in \partial\Omega_L$$

and

$$\Delta u(x,y) = v(x,y) \qquad x,y \in \Omega$$
$$v(x,y) = h(x,y) \qquad x,y \in \partial\Omega_D$$

This scheme of splitting the fourth-order differential equation into two-second order may be reliable yet inconvenient due to the restriction on boundary conditions.

2.5 Paricular Solutions

Methods such as LMAPS, MAPS, and MFS require the use of closed-form particular solutions as a basis function to approximate the solution to PDEs. Most recent researchers do not have to derive their particular solution before use since they are available, [22, 18, 4, 8, 26]. Below are the lists of RBFs and their second and fourth-order particular solution we implemented in this paper.

The Particulars of 2D Laplace differential operator [22]

$$\Delta\Gamma(r) = \phi(r)$$
$$\Delta\Gamma(r) = \frac{1}{r}\frac{d}{dr}\left(r\frac{d\Upsilon''(r)}{dr}\right)$$

- $\phi(r) = \sqrt{r^2+c^2}$,

$$\Gamma(r) = \frac{4c^2+r^2}{9}\sqrt{r^2+c^2} - \frac{c^3}{3}\ln(c+\sqrt{r^2+c^2}) \tag{2.8}$$

- $\phi(r) = \frac{1}{\sqrt{r^2+c^2}}$,

$$\Gamma(r) = \sqrt{r^2+c^2} - \ln\left(1+\sqrt{r^2+c^2}\right)$$

- $\phi(r) = \sqrt{1+c^2r^2}$,

$$\Gamma(r) = \frac{1}{9c^2}\left((4+c^2r^2)\sqrt{1+c^2r^2} - 3\ln(1+\sqrt{1+c^2r^2})\right)$$

- $\phi(r) = \frac{1}{\sqrt{1+c^2r^2}}$,

$$\Gamma(r) = \frac{1}{c^2}\left(\sqrt{1+c^2r^2} - \ln(1+\sqrt{1+c^2r^2})\right) \tag{2.9}$$

The Particulars of 2D Biharmonic differential operator [22]

$$\Delta^2\Upsilon(r) = \Delta\Delta\Upsilon(r) = \phi(r)$$
$$\Delta^2\Upsilon(r) = \frac{1}{r}\frac{d}{dr}\left(r\frac{d}{dr}\left(\frac{1}{r}\frac{d}{dr}\left(r\frac{d\Upsilon(r)}{dr}\right)\right)\right)$$

- $\phi(r) = \sqrt{r^2+c^2}$,

$$\Upsilon(r) = \frac{2c^2}{45}(r^2+c^2)^{\frac{3}{2}} - \frac{7c^4}{60}\sqrt{r^2+c^2} + \frac{2c^2-5r^2}{60}c^3\ln(c+\sqrt{r^2+c^2})\cdot\cdot\cdot + \frac{1}{225}(r^2+c^2)^{\frac{5}{2}} + \frac{c^3r^2}{12} \tag{2.10}$$

- $\phi(r) = \frac{1}{\sqrt{r^2+c^2}}$,

$$\Upsilon(r) = \frac{4r^2-11c^2}{36}\sqrt{r^2+c^2} + \frac{c(2c^2-3r^2)}{12}\ln(c+\sqrt{r^2+c^2}) + \frac{c^3\ln(2c)}{6} + \frac{r^2c}{4}$$

- $\phi(r) = \frac{1}{\sqrt{1+c^2r^2}}$,

$$\Upsilon(r) = \frac{\sqrt{1+c^2r^2}}{36c^4}(4c^2r^2-11) + \frac{2-3c^2r^2}{12c^4}\ln(1+\sqrt{1+c^2r^2}) + \frac{r^2}{4c^2} \tag{2.11}$$

Chapter 3

LOCALIZED METHOD OF APPROXIMATE PARTICULAR SOLUTIONS

3.1 Formulation

In this section, we introduce a meshless collocation method [7], LMAPS. By directly implementing the particular solution of the RBF. Let us consider a fourth-order partial differential equation as follows:

$$\mathcal{L}u(x,y) = f(x,y) \qquad x,y \in \Omega \tag{3.1}$$

$$\mathcal{B}_1 u(x,y) = g(x,y) \qquad x,y \in \partial\Omega \tag{3.2}$$

$$\mathcal{B}_2 u(x,y) = h(x,y) \qquad x,y \in \partial\Omega \tag{3.3}$$

where $\mathcal{L} = \Delta^2 + \alpha\Delta + \beta$, $\alpha, \beta \in \mathbb{R}$, is a linear differential operator, Ω is a bounded and closed nonempty domain with boundary $\partial\Omega$. We consider two sets of collocation points, the interior and boundary points.

Let $\{(x_i, y_i)\}_1^{n_i}$ be the interior points and $\{(x_i, y_i)\}_{n_i+1}^{n_i+n_b}$ be the boundary points. Note that $N = n_i + n_b$ is the total number of collocation points. For each $x_i \in \Omega_i$, using KD-tree algorithm we find n_s nearest neighboring points to form a local domain Ω_i, such that $\Omega_i \cap \Omega_s \neq \emptyset$ for some $\Omega_s, s = 1,2,3...n_s$.
The idea is to reformulate (3.1) as:

$$\mathcal{L}u(x,y) = \sum_{k=1}^{N} a_k \phi(x,y) \tag{3.4}$$

The fourth-order particular solution $\Upsilon(x,y)$ is obtained by repeatedly integrating the radial basis function $\phi(x,y)$

$$\mathcal{L}\Upsilon(r) = \phi(r) \tag{3.5}$$

Using the LMAPS, we approximate the exact solution as the linear combination of the fourth-order particular solutions $\{\Upsilon(r_k)\}$ at each of the local points.

$$u(x_s,y_s) \simeq \hat{u}(x_s,y_s) = \sum_{k=1}^{n_s} \alpha_k \Upsilon(r_k), \quad s = 1,2,3...n_s, \tag{3.6}$$

where $\{\alpha_k\}_{k=1}^{n_s}$ are the undetermined coefficients and r is the distance matrix define as the Euclidean norm that is;

$$r = ||(x_s,y_s) - (x_i^{[s]}, y_i^{[s]})|| \tag{3.7}$$

and it the distance r between the collocation point (x_s,y_s) and the local (source) point $(x_i^{[s]}, y_i^{[s]})$.
From (3.6), we have

$$\alpha^{[s]} = (\Upsilon^{[s]})^{-1} \hat{u}^{[s]} \tag{3.8}$$

$$\hat{u}^{[s]} = \left[\hat{u}(x_1,y_1)^{[s]}, \hat{u}(x_2,y_2)^{[s]}, \ldots, \hat{u}(x_{n_s},y_{n_s})^{[s]}\right]^T, \quad a^{[s]} = [a_1,a_2,\ldots,a_{n_s}]^T$$

From (3.5),(3.6), and (3.8), we can reformulate (3.1) as follows:

$$\begin{aligned} \mathcal{L}\hat{u}(x_s,y_s) &= \sum_{k=1}^{n_s} \alpha_k \mathcal{L}\Upsilon(r) && = f(x_s,y_s). \\ &= \sum_{k=1}^{n_s} \alpha_k \phi(r) \\ &= \Theta^{[s]} \alpha^{[s]} \\ &= \Theta^{[s]} (\Upsilon^{[s]})^{-1} \hat{u}^{[s]} \\ &= \mathcal{P}_\Theta \hat{u}^{[s]} && = f(x_s,y_s) \end{aligned} \tag{3.9, 3.10}$$

Now, we impose (3.6) on the boundary condition (3.2)

$$\begin{aligned} \mathcal{B}_1\hat{u}(x_s,y_s) &= \sum_{k=1}^{n_s} \alpha_k \mathcal{B}_1 \Upsilon(r) && = g(x_s,y_s) \\ &= \Xi^{[s]} \alpha^{[s]} \\ &= \Xi^{[s]} (\Upsilon^{[s]})^{-1} \hat{u}^{[s]} \\ &= \mathcal{P}_\Xi \hat{u}^{[s]} && = g(x_s,y_s) \end{aligned} \tag{3.11}$$

Like we did for the first boundary condition, we formulate the second boundary condition

(3.3)

$$\begin{aligned}\mathcal{B}_2\hat{u}(x_s,y_s) = \sum_{k=1}^{n_s} \alpha_k \mathcal{B}_2\Upsilon(r) &= h(x_s,y_s)\\ &= \Pi^{[s]}\alpha^{[s]}\\ &= \Pi^{[s]}(\Upsilon^{[s]})^{-1}\hat{u}^{[s]}\\ &= \mathcal{P}_\Pi\hat{u}^{[s]} = h(x_s,y_s)\end{aligned} \tag{3.12}$$

where

$$\begin{aligned}\mathcal{P}_\Theta &= \Theta^{[s]}(\Upsilon^{[s]})^{-1}\\ \mathcal{P}_\Xi &= \Xi^{[s]}(\Upsilon^{[s]})^{-1}\\ \mathcal{P}_\Pi &= \Pi^{[s]}(\Upsilon^{[s]})^{-1}\end{aligned}$$

$\Theta^{[s]}$, $\Xi^{[s]}$ and $\Pi^{[s]}$ are $n_s \times 1$ vectors.

Finally, by collocating all points (boundary and interior points) using (3.10)–(3.12), we obtain a sparse linear system of equations. For simplicity, we use X_i instead of (x_i, y_i) in the sparse system shown below.

$$\begin{matrix}\\ (ni)\times N\\ \\ \\ \\ \\ \frac{nb}{2}\times N\\ \\ \\ \frac{nb}{2}\times N\\ \\ \\ \\ \end{matrix}\!\!\begin{bmatrix} \mathcal{P}_\Theta(X_1) \\ \\ \mathcal{P}_\Theta(X_2) \\ \vdots \\ \mathcal{P}_\Theta(X_{n_i}) \\ \\ \mathcal{P}_\Xi(X_{n_{i+1}}) \\ \vdots \\ \mathcal{P}_\Xi(X_{n_i+\frac{nb}{2}}) \\ \mathcal{P}_\Pi(X_{n_i+\frac{nb}{2}+1}) \\ \vdots \\ \mathcal{P}_\Pi(X_{n_i+nb}) \end{bmatrix} \begin{bmatrix} \hat{u}(X_1) \\ \\ \hat{u}(X_2) \\ \vdots \\ \hat{u}(X_{n_i}) \\ \\ \hat{u}(X_{n_i+1}) \\ \vdots \\ \hat{u}(X_{n_i+\frac{nb}{2}}) \\ \hat{u}(X_{n_i+(\frac{nb}{2}+1)}) \\ \vdots \\ \hat{u}(X_{n_i+nb}) \end{bmatrix} = \begin{bmatrix} f(X_1) \\ \\ f(X_2) \\ \vdots \\ f(X_{n_i}) \\ \\ g(X_{n_i+1}) \\ \vdots \\ g(X_{n_i+\frac{nb}{2}}) \\ h(X_{n_i+(\frac{nb}{2}+1)}) \\ \vdots \\ h(X_{n_i+nb}) \end{bmatrix}. \tag{3.13}$$

We generalize the sparse linear system of equation in (3.13) as

$$\mathcal{P}u = \mathcal{F}, \qquad \texttt{size}(\mathcal{P}) = N \times N \tag{3.14}$$

The solution can be obtained by simply multiplying both sides of equation (3.14) by $\mathcal{P}^{-1}$ to yield;

$$u = \mathcal{F}\mathcal{P}^{-1} \tag{3.15}$$

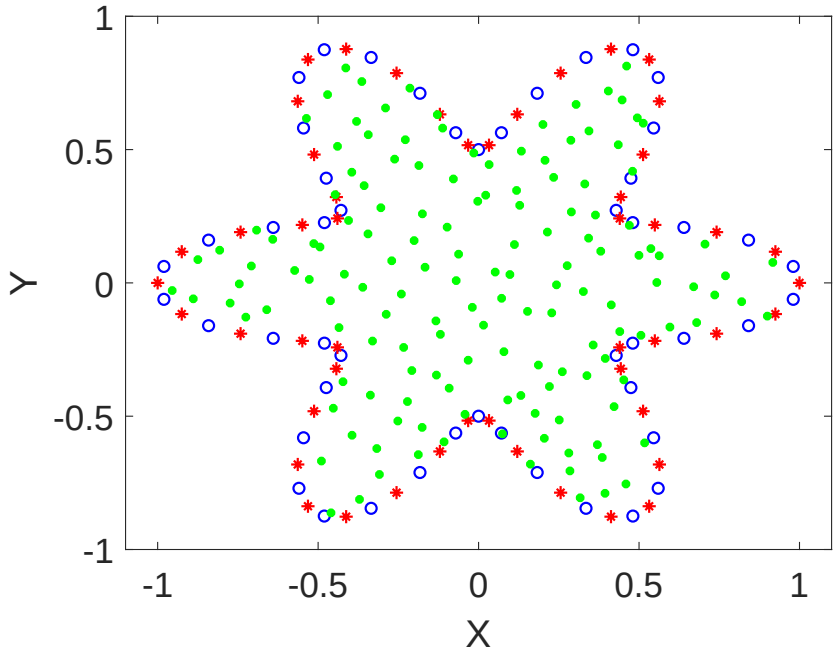

Figure 3.1: The computational domain showing interior points (●) and alternating boundary points (● and ○)
in a star-shaped domain.

The purpose of our study is to improve the efficiency of this method; thus, in our numerical implementation, we impose different boundary conditions on the alternating boundary points, as shown in Figure 3.1 by doing so, resulting in a sparse square matrix in ((3.13)). Hence we obtain stable and accurate results.

It is important to note that in the traditional approach, the two boundary conditions are imposed on each boundary point, therefore, the resulting sparse matrix is non-square with size (N+nb, N). Also, the constant shape parameter was used instead of the variable shape parameter. Using the LMAPS, the system of equation is ill-conditioned; thus, the results tend to be inaccurate and unstable.

Chapter 4

NUMERICAL IMPLEMENTATION

The total number of interior points n_i we considered in all examples is far too much for the implementation of LMAPS. As such, the KD-tree algorithm is used to construct a binary tree for the collocation points. Afterward, the `Knnsearch` function is used to search for n_s nearest neighboring points from the binary tree of a collocation point. The n_s points are called local nodes. To illustrate the effectiveness of the method, we considered three numerical examples in 2D. We used two RBFs (MQ and IMQ) with their particular solutions, listed in chapter 2.

The accuracy is measured by the root mean squared error and the absolute maximum error

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\hat{u}(x_j,y_j)-u(x_j,y_j))^2}, \tag{4.1}$$

$$AME = \max_{1\le j\le N}|\hat{u}(x_j,y_j)-u(x_j,y_j)|, \tag{4.2}$$

where u and $\hat{u}$ are exact and approximate solutions, respectively.

For all the numerical implementation, we choose ni number of interior points and n_b number of boundary points. We generate uniformly partitioned interior points using Halton quasi-random set generator, `haltonset` [5, 11]. The uniform distribution of the boundary points also helps in well-conditioning of the resulting matrix.

For the examples in the next section, we consider three irregular domains in our numerical implementation. The parametric equation for these domains is as follows.

$$\partial\Omega = \{(x,y)|x = r(\theta)\cos(\theta), y = r(\theta)\sin(\theta), 0 \le \theta \le 2\pi\}$$

.

1. Star-shaped domain:$r(\theta) = \frac{1}{2}(1+\cos^2(3\theta))$.

2. Peanut-shaped domain: $r(\theta) = 0.3\sqrt{\cos(2\theta) + \sqrt{1.1 - \sin^2(2\theta)}}$.

3. Amoeba-shaped domain: $r(\theta) = e^{\sin(\theta)}(\sin^2(2\theta)) + e^{\cos(\theta)}(\cos^2(2\theta))$.

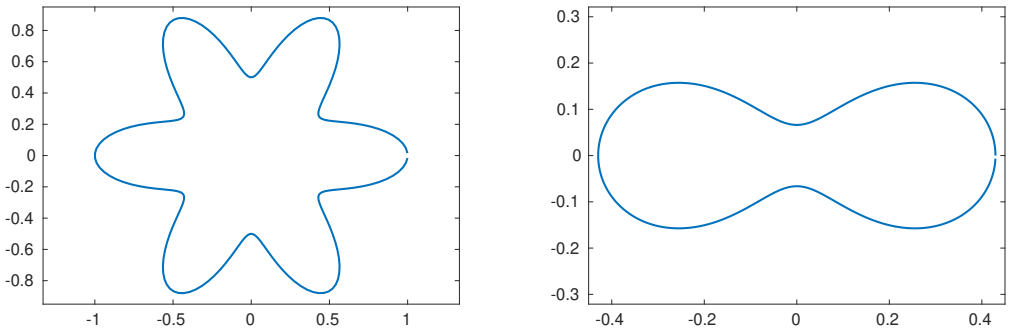

Figure 4.1: The profiles of three irregular domains. (a) Star-shaped (b) Peanut-shaped .

4.1 Numerical Examples and Results

Example 1. In this example, we consider a forth order partial differential equation, with Dirichlet and Neumann boundary conditions. ,

$$(\Delta^2 + \Delta - \lambda)u(x,y) = f(x,y), \qquad (x,y) \in \Omega, \tag{4.3}$$

$$\frac{\partial}{\partial n}u(x,y) = g(x,y), \qquad (x,y) \in \partial\Omega, \tag{4.4}$$

$$u(x,y) = h(x,y), \qquad (x,y) \in \partial\Omega, \tag{4.5}$$

The exact solution is $u(x,y) = e^{(x+y)}$. The irregular star-shape shown in Figure 4.1(a). bounds the computational domain Ω . $f(x,y)$ and $g(x,y)$ are determined based on the exact solution.

In the numerical implementation, we considered the radial basis function, $\phi(r) = \frac{1}{\sqrt{1+c^2r^2}}$. Meaning based on $\Delta^2\Upsilon(r) = \phi(r)$; the fourth-order particular solution is that in (2.11) and based on $\Delta\Upsilon(r) = \phi(r)$; the second-order particular solution is that in (2.9).

We chose 40 local points throughout the implementation of this example and started with $\lambda = 2$.

The profile of the exact solution is shown in Figure 4.2

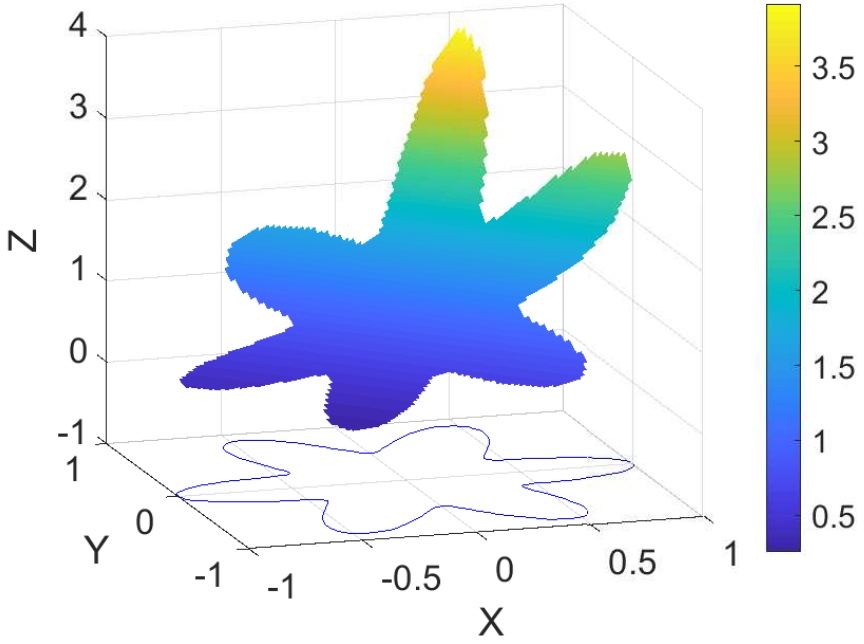

Figure 4.2: Profile of the exact solution.

Fourth-order PDEs have two boundary conditions. So we impose the boundary conditions on each boundary point (non-alternating boundary point) as previously implemented in the traditional approach, as such resulting in a sparse matrix that is non-square. On the new technique, we imposed the two boundary conditions one after the other on alternating boundary points by insignificantly shifting the boundary points(alternating boundary points). This time the resultant sparse matrix is square.

In figure 4.3 below, we compare three different accuracies — the non-alternating boundary point scheme with a constant shape parameter, which is the traditional method. And the non-alternating boundary point scheme with a variable shape parameter which is the 'first stage' improved traditional method. The alternating boundary point approach is the second stage improvement where we imposed the boundary conditions one after the other on alternating boundary points and also implemented a variable shape parameter.

As such, the RMSEs for alternating points with the variable shape parameter is more accurate and stable as compared to the non-alternating points with the variable shape parameter. That of the non-alternating with constant shape parameter has a massive error, as shown in figure 4.3.

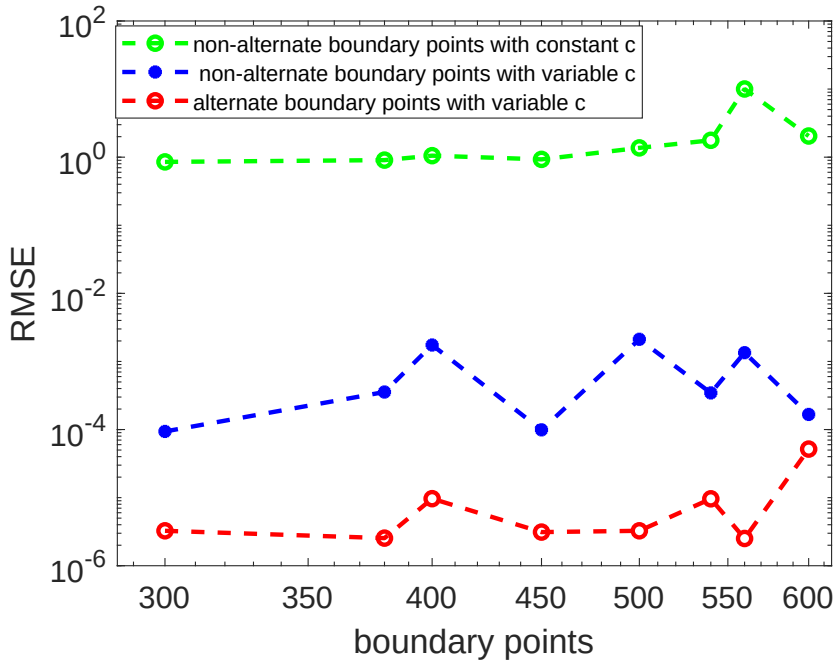

Figure 4.3: Example 1: Illustrating differences in accuracy for alternating and non-alternating boundary points. ni=6000.

How accurate and stable the solution of most RBFs meshless methods depends on the selection of the shape parameter. There are several techniques to choose the shape parameter, but the one applied in this paper is the Leave-one-out cross-validation [23]. This strategy is used to find a fixed shape parameter for $\phi(r)$ with its corresponding particular solution by implementing the algorithm in section 2.2. With a search interval $[min, max]$, it was possible to find the optimal shape parameter by minimizing the cost-function by using `fminbnd` function. It is important to note that the LOOCV depends on the search interval; therefore, if not well-chosen, there is no guarantee of getting an optimal shape parameter. So we had to adjust $[min, max]$ through some multiple testing to get a suitable shape parameter. It is still a research topic on how to get a proper search interval.

Earlier, there has been an emphasis on the use of fixed shape parameter, but now there has been a switch to the use of variable shape parameter for more accuracy [15]. As shown in Table 4.1.

Again In Table 4.1, ep_{min} is the optimal shape parameter we calculated using LOOCV. $ep_{min} = c$ is the constant shape parameter, and it is the lower bound for the variable shape parameter. The upper bound, $ep_{max} = ep_{min} + \delta \quad (\delta \in +\mathbb{R})$. For instants in the first-row second column of the Table $ep_{min} = 0.640$ and $ep_{max} = 0.640 + 1$, that is $[0.640, 1.640]$.

Table 4.1: Example 1:Comparing the accuracy using constant and variable shape parameter

	variable shape parameter		**constant shape parameter**	
search Interval	$[ep_{min}, ep_{max}]$	**RMSE**	c	**RMSE**
$[0,1]$	$[0.640, 1.640]$	4.968E-05	0.640	9.449E-04
$[0,3]$	$[1.635, 2.635]$	9.863E-06	1.635	1.587E-04
$[0,5]$	$[2.767, 3.767]$	2.380E-06	2.767	1.536E-04
$[0,7]$	$[3.148, 4.148]$	2.721E-06	3.148	1.464E-04
$[0,9]$	$[3.297, 4.297]$	2.272E-06	3.297	2.574E-02
$[0,11]$	$[3.420, 4.420]$	7.595E-05	3.420	1.232E-04

The shape parameter also depends on how good we choose the search interval. For the case of the variable shape parameter, it is clear from the table that too little, $[0,1]$ or too big $[0,11]$ search intervals result in RMSEs 4.968E-05 and $7.595E-05$, respectively which are not accurate as compared to the others. The more precise and stable results came from the search interval $[0,3]$ to $[0,9]$.

For the case of the constant shape parameter, all the search intervals produce a shape parameter that yields relatively good accuracy, except for the search interval $[0,9]$ with shape parameter 3.297, the accuracy is $2.574E-02$, which is not consistent with the others.

We conclude that for the variable shape parameter, the RMSE is more accurate and stable, while RMSE of the constant shape parameter is not as good and not stable, as shown in Table 4.1.

For testing for different delta in $[epmin, epmax]$ where epmin= c and $ep_{max} = ep_{min} + \delta$. We introduce another strategy to construct the variable shape parameter. Here $ep_{min} = c - \delta_1$ and $ep_{max} = c + \delta_2$, where $\delta = \delta_1 + \delta_2$. It is important to take $ep_{min} = max\{c - \delta_1, 0\}$ since we do not want the shape parameter to be less than zero. c is the optimal shape parameter obtained from LOOCV; it is the constant shape parameter. In this case the variable shape parameter is given as $[c - \delta_1, c + \delta_2]$ same as $[epmin, epmax]$. If $\delta_1 = 0$ then $ep_{min} = c$.

For the search interval $[0,5]$ $c = 2.767$ and the $RMSE = 1.536E-04$. In all cases where $\delta_2 \geq \delta_1$, the value $ep_{max} \geq c$ and the accuracy becomes excellent and stable. However, as $\delta_2 < \delta_1$, $ep_{max} < c$, the efficiency reduces. It is advisable to keep ep_{max} not lesser or too higher than the optimal shape parameter c. This analysis of result is shown in Table 4.2

Table 4.2: Example 1: The accuracy using a variable shape parameter with interval $[0,5]$ and $n_s = 40$ taking $n_b = 400$ and $n_i = 6000$.

$\delta = 1.72$		$c = 2.767$	
δ_1	δ_2	$[c-\delta_1, c+\delta_2]$	RMSE
0	1.72	$[2.767, 4.487]$	2.620E-06
0.72	1	$[2.047, 3.047]$	2.453E-06
0.86	0.86	$[1.907, 2.767]$	8.320E-06
1	0.72	$[1.767, 2.481]$	1.039E-05
1.40	0.32	$[1.367, 1.687]$	1.454E-05

In Table 4.3, we show the RMSE for different λ and interior points but kept a fixed $n_b = 400$ boundary points. The numerical accuracy should always be valid no matter the value of λ.

Table 4.3: Example 1: RMSE for different λ using 40 local nodes and 400 boundary points.

ni	5000	6000	9000
$\lambda = 10$	2.527E-06	9.915E-06	7.198E-06
$\lambda = 10^2$	2.624E-06	1.102E-05	7.311E-06
$\lambda = 10^3$	5.089E-06	6.422E-06	8.903E-06

Table 4.4 shows more numerical results considering 40 local points and a fixed, variable shape parameter for different interior and boundary points.

For 6000 and 8000 interior points, the various number of boundary points from 250 to 850 produces excellent and stable numerical results. However, as we increase the number of interior points to 9000; the RMSE is not as good and stable for boundary points 250 to 600. Increasing the number of boundary points from 750 through to 850, we obtained better and stable accuracies. One can say that for a large number of interior points, there is a need to take large boundary points.

It is possible to calculate the number of boundary points depending on the number of interior points; this technique will be part of our future work.

Table 4.4: Example 1: The accuracy using a variable shape parameter with interval $[0,3]$ and $n_s = 40$ for various boundary and interior points.

$ni = 6000$		$ni = 8000$		$ni = 10000$	
nb	RMSE	nb	RMSE	nb	RMSE
250	1.960E-06	250	3.620E-06	350	1.387E-05
400	9.863E-06	350	4.522E-06	400	1.336E-05
450	2.685E-06	450	3.251E-06	450	1.602E-05
500	3.623E-06	550	3.950E-06	550	5.104E-06
600	4.741E-06	600	5.134E-06	600	4.152E-05
650	3.462E-06	700	4.805E-06	750	6.655E-06
750	3.778E-06	750	4.988E-06	800	9.339E-06
850	5.177E-06	800	3.560E-06	850	7.674E-06

Example 2. In this example, we consider the following bi-harmonic problem, with Dirichlet and Laplace boundary conditions.

$$\Delta^2 u(x,y) = f(x,y), \qquad (x,y) \in \Omega \tag{4.6}$$

$$\Delta u(x,y) = g(x,y), \qquad (x,y) \in \partial\Omega \tag{4.7}$$

$$u(x,y) = h(x,y), \qquad (x,y) \in \partial\Omega \tag{4.8}$$

Again $f(x,y), g(x,y)$ and $xh(x,y)$ are determined using the exact solution below;

$$u(x,y) = y\sin(x) + x\cos(y)$$

The exact solution is $u(x,y) = e^{(x+y)}$. The irregular Peanut-shaped domain represented in Figure 4.1(b) bounds the computational domain Ω . $f(x,y)$ and $g(x,y)$ are determined based on the exact solution.

In the numerical implementation, we considered the radial basis function, $\phi(r) = \sqrt{c^2 + r^2}$. Thus based on $\Delta^2\Upsilon(r) = \phi(r)$; the fourth-order particular solution is that in (2.10). We chose 45 local points throughout the implementation of this example.

In Section 3.1 we reformulated the first boundary condition (3.2) using (3.6) to yield $\Xi^{[s]}(\Upsilon^{[s]})^{-1}\hat{u}^{[s]}$. where $\Xi^{[s]} = \mathcal{B}_1\Upsilon(r)$. For this Example, $\mathcal{B}_1 = \Delta$ therefore $\Xi^{[s]} = \Delta\Upsilon(r)$ and

$$\Delta\Upsilon(r) = \Gamma(r)$$

where $\Upsilon(r)$ is fourth-order particular solution (2.10) and $\Gamma(r)$ is the second order particular solution(2.8).

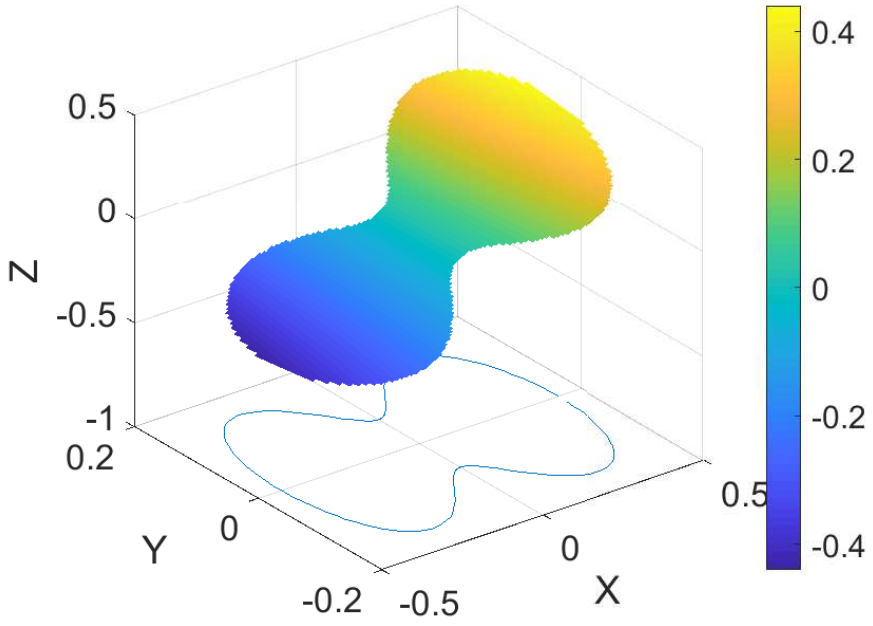

Figure 4.4: Profile of the exact solution.

In Table 4.5, we made the comparison of the numerical accuracy for one set and two sets of boundary points that are non-alternating and alternating, respectively. We choose 400 boundary points, different interior points, and 45 local points. Again the optimal shape parameter is obtained using the LOOCV with the initial search interval of $[3,5]$. We used

Table 4.5: Example 2:Comparing the accuracy using alternating and non-alternating boundary points.

	Alternating boundary points		**Non-Alternating boundary points**	
ni	**RMSE**	**MAX ERROR**	**RMSE**	**MAX ERROR**
5000	7.991E-07	2.296E-06	6.784E-05	2.320E-04
7000	9.699E-07	2.987E-06	7.348E-05	2.311E-04
9000	1.882E-06	6.045E-06	1.072E-04	3.026E-04
12000	2.245E-06	5.421E-06	6.359E-05	1.484E-04
14000	4.160E-06	1..101E-05	2.841E-04	1.090E-03
15000	4.715E-06	1.141E-05	4.734E-05	2.003E-04

We used the variable shape parameter instead of the constant shape parameter for all numerical results in this table. Once again, implementing the alternating boundary points produces more accurate and stable results with various interior points as compared to the non-alternating boundary points.

In Figure 1, we also compared RMSEs for variable and constant shape parameters where we choose 10000 interior points and different boundary points for computation. For all boundary points considered in the graph At nb=400, the RMSE taking the variable and constant shape parameters is 1.790E-06 and 7.162E-05, respectively. For nb=700, the RMSEs are 8.737E-06 and 3.413E-04, respectively. In all cases for the various number of boundary points, the use of the variable shape parameter has proven to be more accurate and stable.

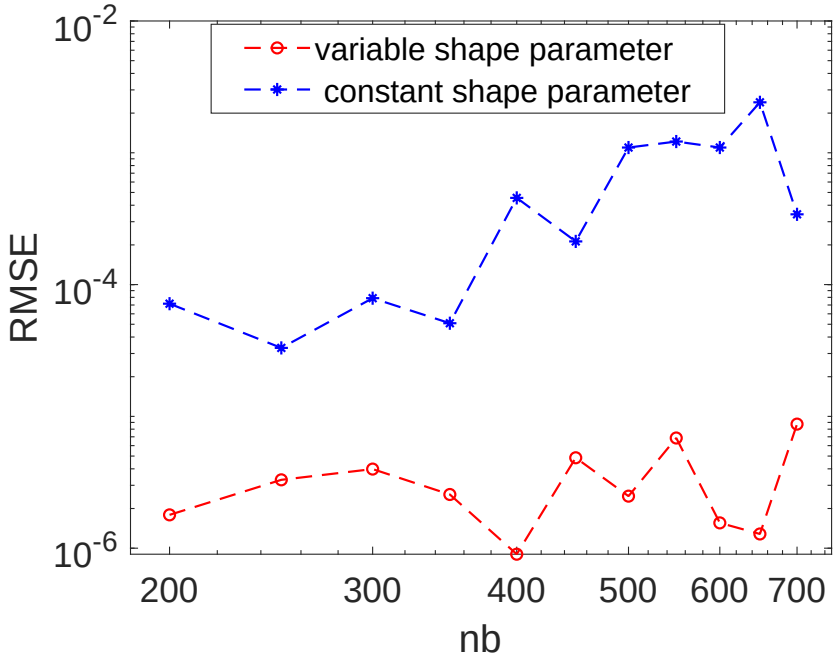

Figure 4.5: Example 2: The comparison of accuracy using variable and constant shape parameter. ni=10000.

Chapter 5

CONCLUSIONS AND REMARKS

The difficulty of solving fourth-order PDE has led to the splitting of fourth-order PDEs into two decoupled second-order PDEs. This decoupling technique focuses only on PDEs with the Dirichlet and Laplace boundary conditions. This study aims to solve fourth-order PDEs directly without splitting into two second-order PDEs. To enhance the performance of this technique, we further investigate our objectives already mentioned in chapter 1.

- Without splitting fourth-order PDEs using LMAPS, we have successfully solved fourth-order PDEs with constant coefficients directly. The method has no restriction on a distinct boundary condition. In numerical examples one and two, we impose the Dirichlet, Neumann and the Dirichlet, Laplace boundary conditions.
- We also made sure to assign each boundary condition on alternating boundary points, instead of imposing the two boundary conditions at the same time on a boundary point. In doing so, the resultant matrix is square. In essence, the accuracy significantly improved.
- In the selection of a suitable (fixed) shape parameter, we adopt the LOOCV algorithm, then extend the fixed shape parameter to the variable shape parameter. In all numerical computations, the variable shape parameter outperforms the constant shape parameter as expected.

We are expecting to improve upon the method in our future research.

- One crucial thing to look at is finding a strategy to determine the search interval for obtaining the optimal shape parameter.
- To achieve a more accurate and stable result, we will consider how to determine the number of boundary points to use for a given number of interior points.
- Instead of alternating points, we will introduce a fictitious boundary outside the main domain.

- We will extend this method to the 3-dimensional domain, time-dependent problem, and fourth-order PDEs with variable coefficient.
- We will also compare our approach to Kansa's method and MPS-Kansa method.

BIBLIOGRAPHY

[1] Matania Ben-Artzi, I. Chorev, J.-P. Croisille, and Dalia Fishelov. A compact difference scheme for the biharmonic equation in planar irregular domains. *SIAM Journal on Numerical Analysis*, 47(4):3087–3108, 2009.

[2] S. Britt, Semyon Tsynkov, and Eli Turkel. A compact fourth order scheme for the Helmholtz equation in polar coordinates. *Journal of Scientific Computing*, 45(1-3):26–47, 2010.

[3] Damian Brown, Leevan Ling, Edward Kansa, and Jermy Levesley. On approximate cardinal preconditioning methods for solving PDEs with radial basis functions. *Engineering Analysis with Boundary Elements*, 29(4):343–353, 2005.

[4] Wanru Chang, C. S. Chen, and Wen Li. Solving fourth order differential equations using particular solutions of Helmholtz-type equations. *Applied Mathematics Letters*, 86:179–185, 2018.

[5] C. S. Chen, Y. C. Hon, and R. A. Schaback. Scientific computing with radial basis functions. *Usm*, 2005.

[6] Ching-Shyang Chen, Chia-Ming Fan, and P. H. Wen. The method of approximate particular solutions for solving elliptic problems with variable coefficients. *International Journal of Computational Methods*, 8(03):545–559, 2011.

[7] Ching-Shyang Chen, Chia-Ming Fan, and P. H. Wen. The method of approximate particular solutions for solving certain partial differential equations. *Numerical Methods for Partial Differential Equations*, 28(2):506–522, 2012.

[8] A. H.-D. Cheng. Particular solutions of Laplacian, Helmholtz-type, and polyharmonic operators involving higher order radial basis functions. *Engineering Analysis with Boundary Elements*, 24(7-8):531–538, 2000.

[9] A. H.-D. Cheng. Multiquadric and its shape parameter—a numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Engineering analysis with boundary elements*, 36(2):220–239, 2012.

[10] Tobin A. Driscoll and Bengt Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Computers & Mathematics with Applications*, 43(3-5):413–422, 2002.

[11] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary. *Computers & Mathematics with Applications*, 43(3-5):439–455, 2002.

[12] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers & Mathematics with Applications*, 48(5-6):853–867, sep 2004.

[13] Bengt Fornberg and Julia Zuev. The Runge phenomenon and spatially variable shape parameters in RBF interpolation. *Computers & Mathematics with Applications*, 54(3):379–398, 2007.

[14] C.-S. Huang, C.-F. Lee, and A. H.-D. Cheng. Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method. *Engineering Analysis with Boundary Elements*, 31(7):614–623, 2007.

[15] E. J. Kansa and R. E. Carlson. Improved accuracy of multiquadric interpolation using variable shape parameters. *Computers & Mathematics with Applications*, 24(12):99–120, 1992.

[16] E. J. Kansa and Y. C. Hon. Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Computers & Mathematics with applications*, 39(7-8):123–137, 2000.

[17] Amy M. Kern. Localized Meshless Methods With Radial Basis Functions For Eigenvalue Problems. *The Aquila Digital Community*, 2013.

[18] A. R. Lamichhane and C. S. Chen. The closed-form particular solutions for Laplace and biharmonic operators using a Gaussian function. *Applied Mathematics Letters*, 46:50–56, aug 2015.

[19] Elisabeth Larsson and Bengt Fornberg. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Computers & Mathematics with Applications*, 49(1):103–130, 2005.

[20] Ming Li, Ghizlane Amazzar, Ahmed Naji, and C. S. Chen. Solving biharmonic equation using the localized method of approximate particular solutions. *International Journal of Computer Mathematics*, 91(8):1790–1801, 2014.

[21] Gui-Rong Liu. *Meshfree methods: moving beyond the finite element method*. CRC press, 2009.

[22] Jeanette Marie Monroe. Hybrid Meshless Method for Numerical Solution of Partial Differential Equations. *University of southern mississippi*, 2014.

[23] Shmuel Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2/3):193–210, 1999.

[24] Scott A. Sarra and Derek Sturgill. A random variable shape parameter strategy for radial basis function approximation methods. *Engineering Analysis with Boundary Elements*, 33(11):1239–1245, nov 2009.

[25] Derek Joseph Sturgill. *Variable Shape Parameter Strategies In Radial Basis Funchtion Methods*. PhD thesis, Marshall University Libraries, 2009.

[26] Chia-Cheng Tsai, A. H.-D. Cheng, and Ching-Shyang Chen. Particular solutions of splines and monomials for polyharmonic and products of Helmholtz operators. *Engineering Analysis with Boundary Elements*, 33(4):514–521, 2009.

[27] Song Xiang, Ke ming Wang, Yan ting Ai, Yun dong Sha, and Hong Shi. Trigonometric variable shape parameter and exponent strategy for generalized multiquadric radial basis function approximation. *Applied Mathematical Modelling*, 36(5):1931–1938, may 2012.

[28] Jingyu Yang, Xiaofeng Liu, and P. H. Wen. The local Kansa's method for solving Berger equation. *Engineering Analysis with Boundary Elements.*, 57:16–22, aug 2015.

[29] Guangming Yao, Joseph Kolibal, and Ching-Shyang Chen. A localized approach for the method of approximate particular solutions. *Computers & Mathematics with Applications*, 61(9):2376–2387, 2011.

[30] Guangming Yao, C. H. Tsai, and Wen Chen. The comparison of three meshless methods using radial basis functions for solving fourth-order partial differential equations. *Engineering Analysis with Boundary Elements*, 34(7):625–631, jul 2010.