Summer 2017

Numerical Solution of Partial Differential Equations Using Polynomial Particular Solutions

Thir R. Dangal
University of Southern Mississippi

NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS USING POLYNOMIAL PARTICULAR SOLUTIONS

by

Thir Raj Dangal

A Dissertation
Submitted to the Graduate School
and the Department of Mathematics
at The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved:

Dr. Ching-Shyang Chen, Committee Chair
Professor, Mathematics

Dr. Haiyan Tian, Committee Member
Associate Professor, Mathematics

Dr. Huiqing Zhu, Committee Member
Associate Professor, Mathematics

Dr. Zhifu Xie, Committee Member
Professor, Mathematics

Dr. Bernd Schroeder
Chair, Department of Mathematics

Dr. Karen S. Coats
Dean of the Graduate School

August 2017

Published by the Graduate School

THE UNIVERSITY OF
SOUTHERN
MISSISSIPPI®

ABSTRACT

NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS
USING POLYNOMIAL PARTICULAR SOLUTIONS

by Thir Raj Dangal

August 2017

Polynomial particular solutions have been obtained for certain types of partial differential operators without convection terms. In this dissertation, a closed-form particular solution for more general partial differential operators with constant coefficients has been derived for polynomial basis functions. The newly derived particular solutions are further coupled with the method of particular solutions (MPS) for numerically solving a large class of elliptic partial differential equations. In contrast to the use of Chebyshev polynomial basis functions, the proposed approach is more flexible in selecting the collocation points inside the domain. Polynomial basis functions are well-known for yielding ill-conditioned systems when their order becomes large. The multiple scale technique is applied to circumvent the difficulty of ill-conditioning.

The derived polynomial particular solutions are also applied in the localized method of particular solutions to solve large-scale problems. Many numerical experiments have been performed to show the effectiveness of the particular solutions on this algorithm.

As another part of the dissertation, a modified method of particular solutions (MPS) has been used for solving nonlinear Poisson-type problems defined on different geometries. Polyharmonic splines are used as the basis functions so that no shape parameter is needed in the solution process. The MPS is also applied to compute the sizes of critical domains of different shapes for a quenching problem. These sizes are compared with the sizes of critical domains obtained from some other numerical methods. Numerical examples are presented to show the efficiency and accuracy of the method.

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my committee chair Dr. C.S. Chen for providing the patient guidance, encouragement and advice throughout my time as his student. I am very happy to get him as my supervisor because he cared so much about my work and responded to my questions and queries so promptly.

I would like to extend my sincere thanks to the committee members Dr. Haiyan Tian, Dr. Huiqing Zhu and Dr. Zhifu Xie for their constructive feedback in my work. My sincere thanks also goes to Dr. Bernd Schroeder for his constant support and guidance to improve my dissertation. I would like to acknowledge my friends at the University of Southern Mississippi for their unconditional help.

Last, but not the least, I thank my wife for her support and encouragement throughout the dissertation period. I appreciate her effort.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF ILLUSTRATIONS

Figure

LIST OF ABBREVIATIONS

BEM	-	Boundary Element Method
CS-RBF	-	Compactly Supported Radial Basis Function
FDM	-	Finite Difference Method
FEM	-	Finite Element Method
FVM	-	Finite Volume Method
Inv MQ	-	Inverse Multiquadric
LMPS	-	Localized Method of Particular Solutions
MAE	-	Maximum Absolute Error
MFS	-	Method of Fundamental Solutions
MPS	-	Method of Particular Solutions
MQ	-	Multiquadric
PDE	-	Partial Differential Equation
RBF	-	Radial Basis Functions
Rel Err	-	Relative Error
RMSE	-	Root-Mean-Square Error
TPS	-	Thin-Plate Spline

Chapter 1

INTRODUCTION

1.1 Background

A partial differential equation (PDE) is a differential equation which involves an unknown function of several variables and its partial derivatives. There is a wide range of applications of PDEs in science and engineering. PDEs can be used to describe many real-world phenomena such as sound, heat, electrostatics, electrodynamics, fluid flow, elasticity, or quantum mechanics.

Analytical solutions of most of the PDEs cannot be obtained trivially and many of them do not have analytical solutions. Thus, numerical methods are developed to approximate the solutions of PDEs [48]. Some methods are mesh-based methods such as the finite element method (FEM) [7, 8, 34, 50, 59, 64], finite difference method (FDM) [49, 72], finite volume method (FVM) [28, 46] and the boundary element method (BEM) [38, 41, 60, 69]. In the mesh-based methods, mesh generations for the domain are essential. But the generations of mesh for irregular domains with high dimensions are very difficult. It is one of the biggest limitations of mesh-based methods. To overcome this difficulty, meshless methods [5, 14, 15, 30, 39, 40, 55] are used to solve PDEs.

Meshless methods are computational methods which do the approximation entirely in terms of the computational nodes. They do not need mesh connectivity. Therefore, such methods can easily overcome the difficulty of mesh generations of the domain which is essential in mesh-based methods. There are many kinds of meshless methods in the literature. These methods are more applicable to high dimensional and moving boundary problems. Some of the numerical meshless techniques are well developed and widely applied to academic, industrial and engineering problems. Some such methods are the smooth particle hydrodynamics (SPH) [30, 54], the element free Galerkin method [5] , the reproducing kernel particle method [55], Kansa's method [39, 40] and the method of approximate particular solutions (MPS) [14, 15].

Meshless methods are classified as boundary-type and domain-type methods. The boundary-type methods are used to solve homogeneous PDEs and the domain-type methods are applied to solve non-homogeneous PDEs. The boundary-type methods require only the boundary to be discretized whereas the domain-type methods require the whole domain

to be discretized. Some examples of boundary-type meshless methods are the method of fundamental solutions (MFS) [26, 43], the Trefftz Method [19, 20] and the boundary knot method (BKM) [21]. Some of the domain-type methods are Kansa's method [39, 40], the MPS [14, 15] and smooth particle hydrodynamics [30].

The MPS is a radial basis function based domain-type meshless method. The method has been used to solve many engineering problems such as the Navier-Stokes problem [87], the wave propagation problem [79] and the time-fractional diffusion problem [81]. The MPS can work on regular and irregular domains of high dimensions. However, there are some obvious weaknesses of the MPS such as the very ill-conditioned collocation matrix, the indeterminate shape parameter of the radial basis functions and the difficulty in deriving the closed-form particular solutions for the differential operators [22, 61, 63, 82].

The size of the globally dense matrix in the MPS grows with the increase of computational points. It causes the condition number of the resultant matrix to become usually very high.

The determination of the optimal shape parameter of the radial basis functions to get the most accurate approximation is an outstanding research topic [29, 36, 65, 76, 80].

Furthermore, the MPS requires the particular solutions of the basis function for the given differential operator in the solution process. Although the particular solutions are not unique, it is always a complicated task to find the particular solution for the given differential operator. Much work has been done to find particular solutions [14, 15, 18, 31, 32].

This dissertation will address these specific issues within the method. The following improvements have been made in the method:

- Polynomial basis functions are applied in the MPS instead of radial basis function to avoid the need for an indeterminate shape parameter. The change has made the method more accurate and easy to implement.

- The closed-form particular solution using the standard polynomial basis function of order s $\{x^{i-j}y^j\}, 0 \leq i \leq s, 0 \leq j \leq i$, under a general linear differential operator has been derived. It is a great improvement in the method. With these closed-form particular solutions, the method is open to solve PDEs with linear differential operators of any order.

- It is known that polynomial basis functions are notoriously unstable when the order of the polynomial basis becomes higher. As a result, the polynomial basis functions are not ideal for a global approach since the resultant matrix is extremely ill-conditioned when the order of the polynomial basis is getting higher. Hence, our derived closed-

form particular solution is useless without the proper treatment of the matrix resulting from our formulation. There are various types of preconditioners in the literature. In this paper, we adopt the so-called multiple scale technique [51, 52] which is a pre-conditioning technique to reduce the condition number of the resultant matrix of the MPS. As we shall see in our presented numerical results, the multiple scale technique is very effective for the reduction of the condition number of our formulated matrix system and thus allows our proposed algorithm to successfully solve various kinds of boundary value problems.

The derived polynomial particular solutions have been coupled with the MPS to solve a large class of partial differential equations. The numerical results are produced using traditional MPS and the MPS based on polynomial basis functions. The comparison of the results from the two methods has shown that the new approach is far better than the traditional MPS in terms of accuracy, stability and ease of implementation.

The method of particular solution is not suitable for large-scale problem because of the highly ill-conditioned resultant matrix. There are many mesh-based localized numerical methods such as the finite element method [50] and the finite difference method [49] to solve large-scale problems. But these methods are difficult to implement for the problems defined on higher dimensional irregular geometries. Therefore, localized meshless methods [25, 45, 68, 70, 78, 83] are preferred over the mesh-based methods because of their ability to solve problems defined on regular and irregular domains in higher dimensions. In this dissertation, the localized method of particular solution (LMPS) [83] has been modified by replacing the radial basis function by polynomial basis functions. The advantages of choosing polynomial basis functions over radial basis functions are already explained above. Various types of partial differential equations have been solved using this modified algorithm to see the efficacy of the polynomial particular solution in the localized approach.

1.2 Outline

In this dissertation, closed-form polynomial particular solutions for general second order partial differential operators with constant coefficients have been derived. The derived particular solutions are applied on the MPS and the LMPS to solve various types of partial differential equations. Using these methods, we can solve both small-scale and large-scale problems defined on regular and irregular domains with high dimensions.

In Chapter 2, we derive closed-form particular solutions for general second order partial differential operators with constant coefficients. Furthermore, brief explanation of the MPS and the LMPS based on radial basis functions are given. We have experienced ill

conditioning problems with our new approach when higher-order polynomial basis functions are employed. So, we have applied a pre-conditioning technique, called the multiple scale technique to reduce the condition number of the resultant matrix. In the next portion of Chapter 2, the multiple scale technique is presented. At the end of Chapter 2, the Houbolt method, a third order finite difference time marching scheme is explained. The method has been applied to solve some linear time dependent problems.

In Chapter 3, we explain the method of particular solutions based on polynomial basis functions. As a numerical experiment, the method is employed to solve PDEs with constant coefficients. The closed-form particular solutions we derived in Chapter 2 works only for PDEs with constant coefficients. They do not work for Poisson problem and PDEs with variable coefficients. In one portion of Chapter 3, a new approach is proposed to solve Poisson problem and PDEs with variable coefficients. The new algorithm successfully solved Poisson problem and PDEs with variable coefficients. To prove the further effectiveness of the method, we have solved some nonlinear, time-dependent and higher dimensional problems. The numerical results are illustrated by using tables and figures. We have used the root mean squared error (RMSE), root mean squared error of the derivative with respect to x (RMSEx), maximum absolute error (MAE), average absolute error (Ave Abs Err) and the relative error (Rel Err) to compute the numerical accuracy of the algorithms.

In Chapter 4, the localized method of particular solutions based on polynomial basis functions is presented. This new scheme is applied to solve large-scale problems. The numerical results for PDEs with constant and variable coefficients are presented. The computational domains are regular and irregular in different dimensions. In one part of Chapter 4, some near-singular problems are solved using this approach.

In Chapter 5, we explain a new approach for the MPS [84]. In this approach, the basis function is formed by combining polyharmonic splines and polynomial basis functions. We have used this modified method to compute the sizes of critical domains for a quenching problem and compared the sizes of the domains with the sizes obtained from some other numerical methods. The results show that the proposed scheme has excellent outcomes.

The conclusion and future work are explained in Chapter 6.

Chapter 2

MESHLESS METHODS

2.1 Basis Functions

A set of functions in a function space is called a basis if the functions are linearly independent and every function in the function space is a linear combination of the functions in the set. The functions in this set are called basis functions. In numerical methods and in approximation theory, basis functions are an important tool to approximate a given function and its derivatives. Furthermore, basis functions are used in many numerical models to obtain approximate solutions of partial differential equations. In this dissertation, a large class of partial differential equations is solved by numerical methods using radial basis functions and polynomial basis functions.

2.1.1 Radial Basis Functions

In recent research, radial basis functions [10] have many applications in science and engineering problems such as multivariate function approximation, surface reconstruction, computer graphics, numerical solution of partial differential equations etc. Since radial basis functions are univariate continuous functions, they can easily handle scatter data interpolation problems in very high dimensions. Therefore, radial basis functions have been a good alternative tool for scatter data interpolation problems. There are many kinds of radial basis functions in the literature. Some are globally supported which are good to use for small-scale problems. For large-scale problems, such functions are not feasible to use because of the dense and ill-conditioned interpolation matrix. To control this difficulty, compactly supported radial basis functions(CSRBF) [10, 16, 17], the Local Multiquadric Approximation [45], and the Local RBF Collocation Method [68] have been developed.

Definition 2.1.1. Let $\mathbb{R}^s$ be the s-dimensional Euclidean space. A function $\Phi : \mathbb{R}^s \to \mathbb{R}$ is said to be radial if there exists a univariate function $\phi : [0,\infty) \to \mathbb{R}$ such that

$$\Phi(\mathbf{x}) = \phi(r),$$

where $r = \|\mathbf{x}\|$ is the Euclidean norm.

Some of the commonly used globally supported RBFs are listed in Table 2.1.

Table 2.1: Commonly Used RBFs

RBFs	Formulations
Linear	$\phi(r) = r$
Cubic	$\phi(r) = r^3$
Multiquadric(MQ)	$\phi(r) = \sqrt{r^2 + c^2}$, $c > 0$
Inverse Multiquadric (IMQ)	$\phi(r) = 1/\sqrt{r^2 + c^2}$, $c > 0$
Matern	$\phi(r) = K_\nu(r) r^\nu$, $\nu > 0$
Gaussian	$\phi(r) = e^{-cr^2}$, $c > 0$
Thin Plate Spline (TPS)	$\phi(r) = r^2 ln(r)$
Polyharmonic Splines in 2D	$\phi(r) = r^{2n} ln(r)$, $n \in \mathbb{N}$
Polyharmonic Splines in 3D	$\phi(r) = r^{2n-1}$, $n \in \mathbb{N}$

2.2 Polynomial Particular Solutions

The derivation of particular solutions has played a key role in solving various types of differential equations. In general, for a given differential equation, if the particular solution and the homogenous solution can be obtained, the problem is considered to be solved. However, it is a challenge to obtain a particular solution and the homogeneous solution is not always available. It is well-known that the particular solution of a given differential equation is not unique and there are numerous ways to find a particular solution [3, 15, 22, 32, 63, 77] for various differential operators and basis functions.

Consider the following partial differential equation

$$Lu_p(x,y) = f(x,y)$$

where L is a given linear differential operator with constant coefficients and $f(x,y)$ is a given function. For a general function $f(x,y)$, the closed-form particular solution $u_p(x,y)$ is difficult, if not impossible, to obtain. Consequently, the approximate particular solution is often needed. Over the past two decades, many numerical methods have been proposed for the approximation of the particular solution [14, 15, 18, 31, 32]. In recent years, radial basis functions (RBFs) have been successfully employed for the construction of approximate particular solutions. Due to the rapid development in this area, the method of particular solutions has been established [14, 15, 79] in the context of RBFs and has been applied for solving a large class of partial differential equations in science and engineering. Despite the success of the use of RBFs, there are still challenges such as the determination of the shape parameter of RBFs and the difficulty in deriving closed-form particular solutions [22, 61, 63, 82]. As a result, Chebyshev polynomial functions have been adopted as an

alternative to alleviate some of these difficulties [15, 18, 77]. These approaches have been proven to be highly accurate. However, the solution procedure is quite tedious and the closed-form particular solutions are only available for some specific differential operators. One of the disadvantages of using Chebyshev polynomials as the basis functions is the requirement that the forcing term of the differential equation should be smoothly extendable to the exterior of the domain for the case of non-rectangular domains. As such, the collocation points can be selected to be at the specific Gauss-Lobatto points.

In this dissertation, the closed-form particular solution using the standard polynomial basis functions of order s $\{x^{i-j}y^j\}, 0 \leq i \leq s, 0 \leq j \leq i$, under a general linear differential operator have been derived. Coupling with the MPS using the newly derived particular solution, a large class of partial differential equations, have been simulated. One of the clear advantages of the proposed approach using the standard polynomial basis over the Chebyshev polynomial basis is that the collocation points can be distributed arbitrarily inside the computational domain without the need for fictitious collocation points outside the domain. Hence, the proposed method is more widely applicable. Furthermore, the proposed method can be easily coupled with the MPS which allows us to solve more general types of partial differential equations.

It is known that polynomial basis functions are notoriously unstable when the order of the polynomial basis becomes higher. As a result, polynomial basis functions are not ideal for a global approach since the resultant matrix is extremely ill-conditioned when the order of the polynomial basis is getting higher. Hence, our derived closed-form particular solution is useless without proper treatment of the matrix resulting from our formulation. There are various types of preconditioners in the literature. In this paper, we adopt the so-called multiple scale technique [51, 52] which is a pre-conditioning technique to reduce the condition number of the resultant matrix of the MPS. As we shall see in our presented numerical results, the multiple scale technique is very effective for the reduction of the condition number of our formulated matrix system and thus allows our proposed algorithm to successfully solve various kinds of boundary value problems.

Let us consider a polynomial basis and find the particular solutions for the basis functions for the general partial differential operators. For simplicity, let us consider the 2D case. We choose the following sequence of the polynomials

$$\mathbf{p}_l(x,y) = x^{i-j}y^j, \quad 0 \leq j \leq i, 0 \leq i \leq s, 1 \leq l \leq w, \quad (x,y) \in \mathbb{R}^2.$$

This sequence forms a basis for $\mathbf{P}_s^d$, the set of d-variate polynomials of degree $\leq s$, with $d = 2$. Here, $w = (s+1)(s+2)/2$ is the dimension of the polynomial space $\mathbf{P}_s^d$.

Let us see one simple example to find the particular solution for such kind of basis

function. Consider a partial differential equation of the form:

$$(L-3I)\mathbf{u_p} := \left(\Delta + \frac{\partial}{\partial x} + \frac{\partial}{\partial y} - 3I\right)\mathbf{u_p} \;=\; x^2y^2, \tag{2.1}$$

where

$$L \;=\; \left(\Delta + \frac{\partial}{\partial x} + \frac{\partial}{\partial y}\right) \tag{2.2}$$

and I is an identity operator. Then, we have

$$-3\left(I - \frac{1}{3}L\right)\mathbf{u_p} = x^2y^2. \tag{2.3}$$

We see that

$$\left(I - \frac{1}{3^5}L^5\right)x^2y^2 = x^2y^2.$$

It follows that

$$\left(I - \frac{1}{3}L\right)\left(I + \frac{1}{3}L + \frac{1}{3^2}L^2 + \frac{1}{3^3}L^3 + \frac{1}{3^4}L^4\right)x^2y^2 = x^2y^2. \tag{2.4}$$

Then, from (2.3) and (2.4), we have

$$-3\mathbf{u_p} \;=\; \left(I + \frac{1}{3}L + \frac{1}{3^2}L^2 + \frac{1}{3^3}L^3 + \frac{1}{3^4}L^4\right)x^2y^2. \tag{2.5}$$

Then, the particular solution of equation (2.1) can be written as

$$\mathbf{u_p} \;=\; -\frac{1}{3}x^2y^2 - \frac{2}{9}x^2y - \frac{2}{9}xy^2 - \frac{8}{27}x^2 - \frac{8}{27}y^2 - \frac{8}{27}xy - \frac{4}{9}x - \frac{4}{9}y - \frac{56}{81}.$$

Similarly, we can find the particular solution for the polynomial functions for any general differential operator.

Theorem 2.2.1. Consider a general form second order linear partial differential equation in two variables with constant coefficients:

$$a_1\frac{\partial^2 u_p}{\partial x^2} + a_2\frac{\partial^2 u_p}{\partial x\partial y} + a_3\frac{\partial^2 u_p}{\partial y^2} + a_4\frac{\partial u_p}{\partial x} + a_5\frac{\partial u_p}{\partial y} + a_6u_p = x^my^n, \tag{2.6}$$

where $\{a_i\}_{i=1}^6$ are real constants, $a_6 \neq 0$ and m and n are positive integers. Then the polynomial particular solution of (2.6) is given by

$$u_p = \frac{1}{a_6}\sum_{k=0}^{\mathcal{N}}\left(\frac{-1}{a_6}\right)^k \mathcal{L}^k(x^my^n) \tag{2.7}$$

where $\mathcal{N} = m+n$ and

$$\mathcal{L} = a_1\frac{\partial^2}{\partial x^2} + a_2\frac{\partial^2}{\partial x\partial y} + a_3\frac{\partial^2}{\partial y^2} + a_4\frac{\partial}{\partial x} + a_5\frac{\partial}{\partial y}.$$

Proof. Equation (2.6) can be written as

$$(\mathcal{L}+a_6 I)\,u_p = x^m y^n, \tag{2.8}$$

which implies

$$\left(I+\frac{\mathcal{L}}{a_6}\right)(a_6 u_p) = x^m y^n. \tag{2.9}$$

Since $\mathcal{L}$ is a differential operator containing various partial derivatives, it is clear that $\mathcal{L}^{m+n+1}(x^m y^n) = 0$. Hence, the following identity is always true.

$$\left(I+\left(\frac{\mathcal{L}}{a_6}\right)^{\mathcal{N}+1}\right) x^m y^n \;\; = \;\; x^m y^n, \tag{2.10}$$

where $\mathcal{N} = m+n$. By direct algebraic factorization, we have

$$I+\left(\frac{\mathcal{L}}{a_6}\right)^{\mathcal{N}+1} = \left(I+\frac{\mathcal{L}}{a_6}\right)\sum_{k=0}^{\mathcal{N}}\left(\frac{-1}{a_6}\right)^k \mathcal{L}^k. \tag{2.11}$$

From (2.10) and (2.11), we have

$$\left(I+\frac{\mathcal{L}}{a_6}\right)\sum_{k=0}^{\mathcal{N}}\left(\frac{-1}{a_6}\right)^k \mathcal{L}^k(x^m y^n) = x^m y^n. \tag{2.12}$$

Comparing (2.9) and (2.12), it follows that

$$a_6 u_p = \sum_{k=0}^{\mathcal{N}}\left(\frac{-1}{a_6}\right)^k \mathcal{L}^k(x^m y^n).$$

Consequently, the particular solution u_p for the above general differential operator is given by

$$u_p = \frac{1}{a_6}\sum_{k=0}^{\mathcal{N}}\left(\frac{-1}{a_6}\right)^k \mathcal{L}^k(x^m y^n). \tag{2.13}$$

□

The following algorithm is presented to find the particular solution of the basis function $x^m y^n$ for the above operator.

Algorithm 1

Step 1: Let $p(x,y) = x^m y^n$, m,n: nonnegative integers.

Step 2: Let partsol = 0 and $\mathcal{L} = a_1\frac{\partial^2}{\partial x^2} + a_2\frac{\partial^2}{\partial x \partial y} + a_3\frac{\partial^2}{\partial y^2} + a_4\frac{\partial}{\partial x} + a_5\frac{\partial}{\partial y}$.

for $k = 1,2,...,m+n$

term $= \mathcal{L}p$

partsol = partsol + $\left(\frac{-1}{a_6}\right)^k$ * term

$p =$ term

end

Step 3: The required particular solution of $x^m y^n$ for the operator $\mathcal{L}$ is given by $\frac{1}{a_6}$(partsol + $x^m y^n$).

2.3 The Method of Particular Solutions

To review the formulation of the method of particular solutions, we consider a boundary value problem: for given $f(x)$ and $g(x)$, find $u(x)$ from

$$\mathcal{L}u(x) = f(x), x \in \Omega \tag{2.14}$$

$$\mathcal{B}u(x) = g(x), x \in \Gamma, \tag{2.15}$$

where $\mathcal{L}$ is a differential operator, $\mathcal{B}$ is a boundary differential operator and Ω is a bounded domain with boundary Γ.

By the method of particular solutions, we approximate

$$\hat{u}(x) = \sum_{i=1}^{n} a_i \Phi(\|x - x_i\|), \ x \in \Omega, \tag{2.16}$$

where $\Phi(\|x - x_i\|)$ is obtained by analytically solving

$$\mathcal{L}\Phi(\|x - x_i\|) = \phi(\|x - x_i\|).$$

Then,

$$f(x) \approx \mathcal{L}\hat{u}(x) = \sum_{i=1}^{n} a_i \phi(\|x - x_i\|), x \in \Omega, \tag{2.17}$$

where $\phi(\|x - x_i\|)$ is a radial basis function. Also,

$$g(x) \approx \hat{u}(x) = \sum_{i=1}^{n} a_i \mathcal{B}\Phi(\|x - x_i\|), \ x \in \Gamma. \tag{2.18}$$

Let us take n_i interior points in Ω and n_b boundary points in Γ such that $n = n_i + n_b$ is the total number of collocation points. For given interior points $\{x_i\}_{i=1}^{n_i}$, we have

$$f(x_i) = \sum_{j=1}^{n} a_j \phi(\|x_i - x_j\|), \;\; i = 1,2,...,n_i. \tag{2.19}$$

Similarly, for the boundary points $\{x_i\}_{i=n_i+1}^{n}$, we have

$$g(x_i) = \sum_{j=1}^{n} a_j \mathcal{B}\Phi(\|x_i - x_j\|), i = n_i + 1,...,n. \tag{2.20}$$

Equations (2.19) and (2.20) can be solved for $\{a_j\}_{j=1}^{n}$. Then (2.16) together with the computed weighted coefficients give the approximate values $\{\hat{u}_i\}_{i=1}^{n}$.

2.4 Localized Method of Particular Solutions

The MPS is not suitable for large-scale problems because of the dense and ill-conditioned coefficient matrix for the resulting system of equations. There are many mesh-based localized numerical methods such as the finite elements method, and the finite difference method to solve large-scale problems. But these methods are difficult to implement for the problems defined on higher dimensional irregular geometries. Therefore, localized meshless methods [25, 45, 68, 70, 78, 83] are preferred over mesh-based methods because of their ability to solve problems defined on regular and irregular domains in higher dimensions. These localized meshless methods have utilized radial basis functions in their solution process. In this section, we will have a brief review of the localized method of particular solutions.

Let us consider the boundary value problem (2.14) – (2.15). Let $\{\mathbf{x}_i\}_{i=1}^{N_i}$ be interior points in Ω, let $\{\mathbf{x}_i\}_{i=N_i+1}^{N_i+N_b}$ be boundary points on Ω and let $N = N_i + N_b$. For each $\mathbf{x}_i$, we choose n nearest points $\mathbf{x}_k^{[i]}$, $k = 1,2,...,n$, to form a local domain Ω_i such that $\Omega_i \cap \Omega_j \neq \phi$ for some $j \neq i$ and $[\mathbf{x}_i]_{i=1}^{N} = \bigcup_i \Omega_i$. In the LMPS, we approximate the solution at each point $\mathbf{x}_j^{[i]} \in \Omega_i$, $j = 1,2,...,n$, by

$$u(\mathbf{x}_j^{[i]}) \simeq \hat{u}(\mathbf{x}_j^{[i]}) = \sum_{k=1}^{n} \alpha_k \Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_k^{[i]}\right\|\right), \; j = 1,2,...,n, \tag{2.21}$$

where $\Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_k^{[i]}\right\|\right)$ is obtained by analytically solving

$$\mathcal{L}\Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_k^{[i]}\right\|\right) = \phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_k^{[i]}\right\|\right).$$

Let Φ_n be the matrix in the system of equations (2.21), we have

$$\alpha = \Phi_n^{-1}\hat{\mathbf{u}}_n \tag{2.22}$$

where

$$\hat{\mathbf{u}}_n = \left[\hat{u}(\mathbf{x}_1^{[i]}), \hat{u}(\mathbf{x}_2^{[i]}), \ldots, \hat{u}(\mathbf{x}_n^{[i]})\right]^T, \alpha = \left[\alpha_1, \alpha_2, \ldots, \alpha_n\right]^T.$$

Then, applying (2.22) to (2.21), we get

$$\hat{u}(\mathbf{x}_j^{[i]}) = \sum_{k=1}^{n} \alpha_k \Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_k^{[i]}\right\|\right) = \hat{\Phi}_n \alpha = \hat{\Phi}_n \Phi_n^{-1} \hat{\mathbf{u}}^{[i]} = \Psi^{[i]} \hat{\mathbf{u}}^{[i]} \tag{2.23}$$

where

$$\hat{\Phi}_n(\mathbf{x}_j^{[i]}) = \left[\Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_1^{[i]}\right\|\right), \Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_2^{[i]}\right\|\right), ..., \Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_n^{[i]}\right\|\right)\right]$$

and

$$\Psi^{[i]} = \hat{\Phi}_n \Phi_n^{-1} = [\psi_1, \psi_2, ..., \psi_n].$$

Now, we want to express equation (2.23) in the global sense. Let $\hat{\mathbf{u}} = [\hat{u}(\mathbf{x}_1), \hat{u}(\mathbf{x}_2), \ldots, \hat{u}(\mathbf{x}_N)]^T$. The global form of equation (2.23) is

$$\hat{u}(\mathbf{x}_j^{[i]}) = \Psi \hat{\mathbf{u}}, \tag{2.24}$$

where the $N \times 1$ vector Ψ is obtained by padding $N - n$ zeros in $\Psi^{[i]}$ based on the mapping between $\hat{\mathbf{u}}^{[i]}$ and $\hat{\mathbf{u}}$. The above scheme can be used to solve the problem (2.14)-(2.15). Applying (2.21) to (2.14), We have

$$\begin{aligned}
f(\mathbf{x}_j^{[i]}) &= \mathcal{L}\hat{u}(\mathbf{x}_j^{[i]}) \\
&= \sum_{k=1}^{n} \alpha_k \mathcal{L}\Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_k^{[i]}\right\|\right) \\
&= \mathcal{L}\hat{\Phi}_n \alpha \\
&= \mathcal{L}\hat{\Phi}_n \Phi_n^{-1} \hat{\mathbf{u}}^{[i]} \\
&= \Lambda^{[i]} \hat{\mathbf{u}}^{[i]} \\
&= \Lambda(\mathbf{x}_j^{[i]}) \hat{\mathbf{u}}
\end{aligned}$$

where

$$\Lambda^{[i]} = \mathcal{L}\hat{\Phi}_n \Phi_n^{-1},$$

and $\Lambda(\mathbf{x}_j^{[i]})$ is obtained by padding $N - n$ zeros in $\Lambda^{[i]}$. Thus,

$$f(\mathbf{x}_j^{[i]}) = \Lambda(\mathbf{x}_j^{[i]}) \hat{\mathbf{u}}, \;\; 1 \le i \le N_i. \tag{2.25}$$

Similarly, applying (2.21) to (2.15), we get

$$\begin{aligned} g(\mathbf{x}_j^{[i]}) &= \mathcal{B}\hat{u}(\mathbf{x}_j^{[i]}) \\ &= \sum_{k=1}^{n} \alpha_k \mathcal{B}\Phi\left(\left\|\mathbf{x}_j^{[i]} - \mathbf{x}_k^{[i]}\right\|\right) \\ &= \mathcal{B}\hat{\Phi}_n \alpha \\ &= \mathcal{B}\hat{\Phi}_n \Phi_n^{-1} \hat{\mathbf{u}}^{[i]} \\ &= \Upsilon^{[i]} \hat{\mathbf{u}}^{[i]} \\ &= \Upsilon(\mathbf{x}_j^{[i]}) \hat{\mathbf{u}} \end{aligned}$$

where

$$\Upsilon^{[i]} = \mathcal{B}\hat{\Phi}_n \Phi_n^{-1},$$

and $\Upsilon(\mathbf{x}_j^{[i]})$ is obtained by padding $N-n$ zeros in $\Upsilon^{[i]}$. Thus,

$$g(\mathbf{x}_j^{[i]}) = \Upsilon(\mathbf{x}_j^{[i]})\hat{\mathbf{u}}, \;\; N_i + 1 \leq i \leq N. \tag{2.26}$$

Finally, equations (2.25) and (2.26) form a sparse system of equations. The approximate solution at each point can be obtained by solving equations (2.25) and (2.26).

2.5 Multiple Scale Technique

High order polynomials are notorious in numerical interpolation due to the severe ill conditioning of the resulting matrix. The MPS using polynomials as basis functions has experienced the same difficulty and a special treatment of the resultant matrix system (2.25) – (2.26) is required. To alleviate this difficulty, a multiple scale technique [51, 52] is applied to reduce the condition number of the resulting matrix. Let $w = (s+1)(s+2)/2$. Equations (2.25) – (2.26) can be written in the matrix form

$$\mathbf{A}\mathbf{c} = \mathbf{b}, \tag{2.27}$$

where

$$\mathbf{A} = \begin{bmatrix} \left[x^{i-j}y^j\right]_{n_i \times w} \\ \\ \left[\mathcal{B}u_p^{ij}\right]_{n_b \times w} \end{bmatrix}, \mathbf{c} = \begin{bmatrix} a_{00} \\ a_{10} \\ \vdots \\ a_{ss} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} f(x_1, y_1) \\ \vdots \\ f(x_{n_i}, y_{n_i}) \\ g(x_{n_{i+1}}, y_{ni+1}) \\ \vdots \\ g(x_n, y_n) \end{bmatrix}.$$

Let

$$\mathbf{A} = [A_1\, A_2\, \cdots\, A_w] \text{ and } R_k = \|A_k\|_2,\ k = 1,2,\cdots,w,$$

where A_k is the k^{th} column of matrix $\mathbf{A}$. In the multiple scale technique, the linear system (5.18) is equivalent to

$$\widetilde{\mathbf{A}}\,\widetilde{\mathbf{c}} = \mathbf{b}, \tag{2.28}$$

where

$$\widetilde{\mathbf{A}} = \left[\frac{A_1}{R_1}\ \frac{A_2}{R_2}\ \cdots\ \frac{A_w}{R_w}\right]$$

and

$$\begin{aligned}\widetilde{\mathbf{c}} &= [\widetilde{a}_{00}\,\widetilde{a}_{10}\,\cdots\,\widetilde{a}_{ss}]^T \\ &= [a_{00}R_1\, a_{10}R_2\,\cdots\, a_{ss}R_w]^T. \end{aligned} \tag{2.29}$$

Note that $\widetilde{A}$ is now better conditioned due to the reduction of round-off errors. Once (2.28) is solved, $\{a_{ij}\}$ in $\mathbf{c}$ can be recovered from $\widetilde{\mathbf{c}}$ in (2.29); i.e.,

$$a_{00} = \frac{\widetilde{a}_{00}}{R_1},\ a_{10} = \frac{\widetilde{a}_{10}}{R_2},\ \cdots,\ a_{ss} = \frac{\widetilde{a}_{ss}}{R_w}.$$

2.6 Houbolt Method

For time-dependent PDEs, there are many schemes to discretize the time domain. One of them is the Houbolt finite difference scheme [35, 73, 86]. The method is obtained by solving the Taylor's series expansion as follows:

$$u^n \approx u^{n+1} - (\delta t)u_t^{n+1} + \frac{(\delta t)^2}{2}u_{tt}^{n+1} - \frac{(\delta t)^3}{6}u_{ttt}^{n+1}, \tag{2.30}$$

$$u^{n-1} \approx u^{n+1} - (2\delta t)u_t^{n+1} + \frac{(2\delta t)^2}{2}u_{tt}^{n+1} - \frac{(2\delta t)^3}{6}u_{ttt}^{n+1}, \tag{2.31}$$

$$u^{n-2} \approx u^{n+1} - (3\delta t)u_t^{n+1} + \frac{(3\delta t)^2}{2}u_{tt}^{n+1} - \frac{(3\delta t)^3}{6}u_{ttt}^{n+1}, \tag{2.32}$$

where $u^n = u(x,t_n), \delta t = t^{n+1} - t^n, u_t^{n+1} = \partial u^{n+1}/\partial t, u_{tt}^{n+1} = \partial^2 u^{n+1}/\partial t^2, u_{ttt}^{n+1} = \partial^3 u^{n+1}/\partial t^3$. We can solve equations (2.30), (2.31) and (2.32) to get

$$u_t^{n+1} \approx \frac{1}{6\delta t}(11u^{n+1} - 18u^n + 9u^{n-1} - 2u^{n-2}), \tag{2.33}$$

$$u_{tt}^{n+1} \approx \frac{1}{\delta t^2}(2u^{n+1} - 5u^n + 4u^{n-1} - u^{n-2}). \tag{2.34}$$

In this dissertation, the Houbolt method is utilized to discretize the time domain and change the problem into a series of time-independent problems. Then, the MPS with polynomial basis functions is employed to solve the problems. Time-dependent problems such as the diffusion equation and the wave equation are solved in this dissertation.

Chapter 3

METHOD OF PARTICULAR SOLUTIONS BASED ON POLYNOMIAL BASIS FUNCTIONS

3.1 Methodology

In Chapter 2, we discussed the particular solutions for differential operators using polynomial basis functions. In this section, we will use the derived particular solutions to formulate the MPS based on polynomial basis functions.

Let us consider the boundary value problem (2.14) – (2.15). To discretize the given partial differential equation, we employ the MPS using a polynomial basis functions. In the MPS, we assume that the solution of (2.14)–(2.15) can be represented by

$$u(x,y) \simeq \hat{u}(x,y) = \sum_{i=0}^{s} \sum_{j=0}^{i} a_{ij} u_p^{ij}(x,y), \tag{3.1}$$

where

$$\mathcal{L} u_p^{ij}(x,y) = x^{i-j} y^j,\ 0 \le j \le i, 0 \le i \le s. \tag{3.2}$$

Let $\{(x_i,y_i)\}_{i=1}^{n_i}$ be the set of interior points in the domain Ω, let $\{(x_i,y_i)\}_{i=n_i+1}^{n}$ be the boundary points on Γ, and let $n = n_i + n_b$. Applying (3.1) to (2.14), we obtain

$$\sum_{i=0}^{s} \sum_{j=0}^{i} a_{ij} \mathcal{L} u_p^{ij}(x_k,y_k) = f(x_k,y_k),\ k = 1,2,...,n_i. \tag{3.3}$$

From (3.2), the above equation becomes

$$\sum_{i=0}^{s} \sum_{j=0}^{i} a_{ij} x_k^{i-j} y_k^j = f(x_k,y_k),\ k = 1,2,...,n_i. \tag{3.4}$$

In the MPS, the governing differential equation (2.14) has been transformed to a simple data interpolation problem as shown in (3.4). Imposing (3.1) to satisfy the boundary condition (2.15), we obtain

$$\sum_{i=0}^{s} \sum_{j=0}^{i} a_{ij} \mathcal{B} u_p^{ij}(x_k,y_k) = g(x_k,y_k),\ k = n_i+1, n_i+2, ..., n. \tag{3.5}$$

To ensure that the system of equations (3.4) – (3.5) is solvable, the total number of collocation points n has to be larger than $(s+1)(s+2)/2$. The least squares method will be adopted to solve the above system. Once the undetermined coefficients

$$\{a_{ij}\} = \{a_{00}, a_{10}, a_{11}, a_{20}, a_{21}, a_{22}, \cdots, a_{ss}\}$$

are determined, the approximate solution $\hat{u}$ can be obtained from (3.1).

3.2 Numerical Results for Linear PDEs with constant coefficients

For the numerical experiments, we have considered several domains including regular and irregular geometries. The parametric equation of the boundary $\partial\Omega$ is defined as follows:

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

where

- $$r(\theta) = e^{\sin\theta}\sin^2(2\theta) + e^{\cos\theta}\cos^2(2\theta),\ \sigma(\theta) = \theta \tag{3.6}$$

  is the amoeba-like boundary.

- $$r(\theta) = \left(\cos(4\theta) + \sqrt{\frac{18}{5} - \sin^2(4\theta)}\right)^{1/3},\ \sigma(\theta) = \theta \tag{3.7}$$

  is the cassini-shaped domain.

- $$r(\theta) = 2 + \frac{1}{2}\sin(6\theta),\ \sigma(\theta) = \theta + \frac{1}{2}\sin(6\theta) \tag{3.8}$$

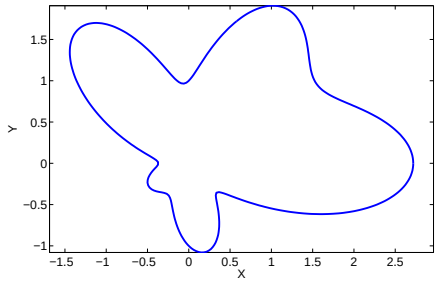
  is the gear-shaped domain.

- $$r(\theta) = 1 + \cos^2(4\theta),\ \sigma(\theta) = \theta \tag{3.9}$$

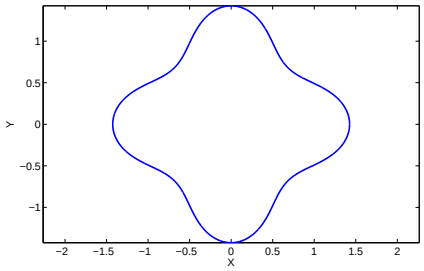
  is the star-shaped domain.

We have considered two more domains which do not have parametric representations. The domains are the L-shaped domain and the corner-shaped domain. The profiles of the domains are given in Figure 3.1.

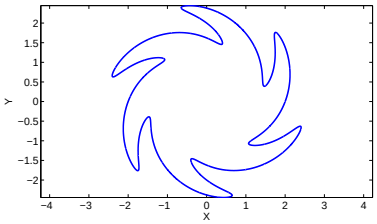
The root-mean-squared error (*RMSE*), the root-mean-squared error of the derivative with respect to x (*RMSEx*), the average absolute error (Ave Abs Error), the maximum



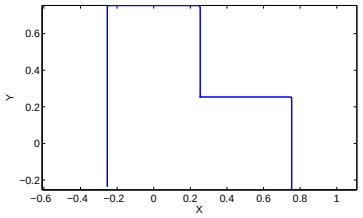
(a) Amoeba-Like



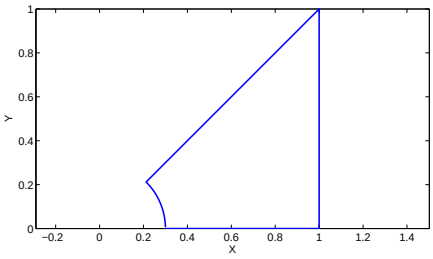
(b) Cassini-Shaped



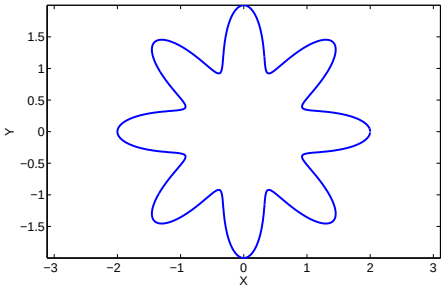
(c) Gear-Shaped



(d) L-Shaped



(e) Corner Shaped



(f) Star-Shaped

Figure 3.1: The profiles of the computational domains.

absolute error (MAE) and the relative error ($Rel\ Err$) are used to measure the accuracy of the solutions. They are defined as follows

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{j=1}^{n_t}\left(\hat{u}_j - u_j\right)^2},$$

$$RMSEx = \sqrt{\frac{1}{n_t}\sum_{j=1}^{n_t}\left(\frac{\partial \hat{u}_j}{\partial x} - \frac{\partial u_j}{\partial x}\right)^2},$$

$$Ave\ Abs\ Err = \frac{1}{n_t}\sum_{j=1}^{n_t}|\hat{u}_j - u_j|,$$

$$MAE = \max_{1 \le j \le n_t} |\hat{u}_j - u_j|$$

and

$$Rel\ Err = \sqrt{\frac{\sum_{j=1}^{n_t} \left(\hat{u}_j - u_j\right)^2}{\sum_{j=1}^{n_t} u_j^2}},$$

where n_t is the number of test points in the domain and $\hat{u}_j$ and u_j are the approximate solution and exact solution at the j^{th} test point respectively.

Example 3.2.1. In this example, we consider the following differential equation in the unit square.

$$\Delta u(x,y) - \frac{\partial^2 u(x,y)}{\partial x \partial y} + \frac{\partial u(x,y)}{\partial y} - u(x,y) = f(x,y),\ (x,y) \in \Omega, \tag{3.10}$$

$$u(x,y) = g(x,y),\ (x,y) \in \partial\Omega, \tag{3.11}$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y) = e^{2x}\cos(y),\ (x,y) \in \overline{\Omega}.$$

The number of interior points, boundary points and test points are 400, 108 and 230 respectively. In Figure 3.2, we show the condition number and RMSE for various orders of the polynomial basis functions with and without the use of the multiple scale technique. From these figures, we clearly see that the multiple scale technique plays an important role in the reduction of the condition number of the collocation matrix. We also observe the improvement of the accuracy when using higher order polynomial bases. One important feature of the proposed algorithm is the numerical stability. When the order of polynomial basis becomes higher, the numerical accuracy remains stable. Without implementing the multiple scale technique, the proposed approach would fail due to an extremely high condition number.

Table 3.1 shows the RMSE and maximum errors for various sets of interior and boundary points with a polynomial basis of order 11. From this table, we observe that increasing the number of interior and boundary points does not contribute to increasing the accuracy. In contrast, Figure 3.2 shows that increasing the order of the polynomial basis functions significantly improves the accuracy. For a polynomial basis with order 11, there are (11+1)(11+2)/2=78 basis functions. Hence, the minimum number of interior and boundary points should be at least 78. From the first row in Table 3.1, it is shown that we can achieve good accuracy using only 80 interior points and 20 boundary points. Since the computational cost is relatively low for such a small number of collocation points, we will double this number in the numerical implementation.

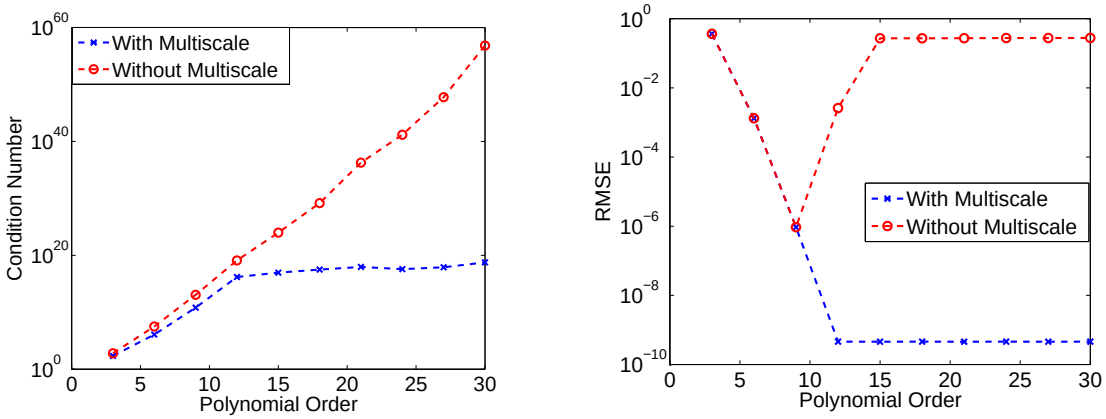


Figure 3.2: Example 3.2.1: The profiles of condition numbers and RMSE with and without using multiple scale technique.

Table 3.1: Example 3.2.1: The RMSE and Maximum errors for different numbers of interior and boundary points with polynomial basis of order 11.

(n_i, n_b)	RMSE	MAE
$(81,20)$	$5.265e-08$	$4.530e-07$
$(121,28)$	$6.405e-08$	$1.760e-07$
$(169,64)$	$5.749e-10$	$2.389e-09$
$(361,88)$	$1.666e-09$	$5.702e-09$
$(576,116)$	$8.325e-09$	$2.497e-08$
$(1089,316)$	$1.468e-08$	$4.912e-08$
$(1444,556)$	$6.749e-09$	$2.658e-08$
$(2304,636)$	$2.056e-08$	$7.318e-08$

Since the multiple scale technique is essential in overcoming the ill-conditioning of the resultant matrix and improves the numerical accuracy, we will continue to use such a technique in the remaining examples in this section.

Example 3.2.2. Let us consider the following Helmholtz problem

$$\Delta u(x,y) + u(x,y) = f(x,y),\ (x,y) \in \Omega, \tag{3.12}$$

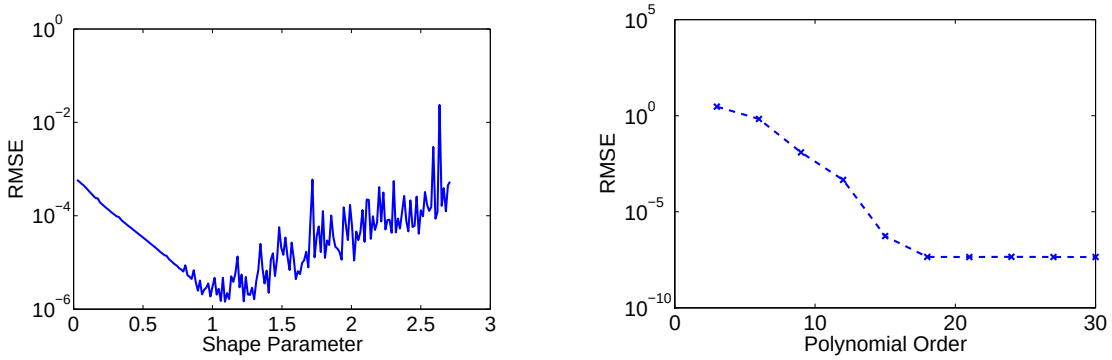
$$u(x,y) = g(x,y),\ (x,y) \in \partial\Omega, \tag{3.13}$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y) = \sin(2x)\cos(2y),\ (x,y) \in \overline{\Omega}.$$

The domain is the amoeba-like domain as shown in Figure 3.1a.

For the numerical approximation, the number of interior points, boundary points, and test points are 412, 100 and 257 respectively. As we have mentioned in the previous example, the number of collocation points depends on the order of the polynomial basis functions. For simplicity, we choose the same number of collocation points for the case of polynomial order equal to 30. To demonstrate the effectiveness of the proposed algorithm, we make a comparison with the MPS using the MQ (Multiquadric) radial basis function. In Figure 3.3, we observe that our proposed approach is not only more accurate but also more stable than the MPS using MQ. As shown in Figure 3.3b, the higher order of the polynomials does not cause any problem in stability due to the use of the multiple scale scheme. On the other hand, the uncertainty of the shape parameter as shown in Figure 3.3a is an additional challenge for the MPS using radial basis functions.



(a) RBF (b) Polynomial

Figure 3.3: Example 3.2.2: (a) RMSE versus the shape parameter using RBF. (b) RMSE versus the polynomial order using polynomial basis.

Example 3.2.3. Let us consider the following mixed boundary value problem

$$\Delta u(x,y) + u(x,y) = f(x,y), \ (x,y) \in \Omega, \tag{3.14}$$

$$u(x,y) = g(x,y), \ (x,y) \in \partial\Omega^D, \tag{3.15}$$

$$\frac{\partial u}{\partial \mathbf{n}} = h(x,y), \ (x,y) \in \partial\Omega^N, \tag{3.16}$$

where $\mathbf{n}$ is the unit outward normal vector, and $f(x,y)$, $g(x,y)$, and $h(x,y)$ are given based on the following analytical solution

$$u(x,y) = e^{2x+2y}, \ (x,y) \in \overline{\Omega}.$$

The boundaries $\partial\Omega^D$ and $\partial\Omega^N$ denote the boundaries on which the Dirichlet and Neumann conditions are applied respectively such that $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N, \partial\Omega^D \cap \partial\Omega^N = \emptyset$. As shown in Figure 3.4, $\partial\Omega^N$ is located in the fourth quadrant; i.e., $3\pi/2 \leq \theta < 2\pi$.

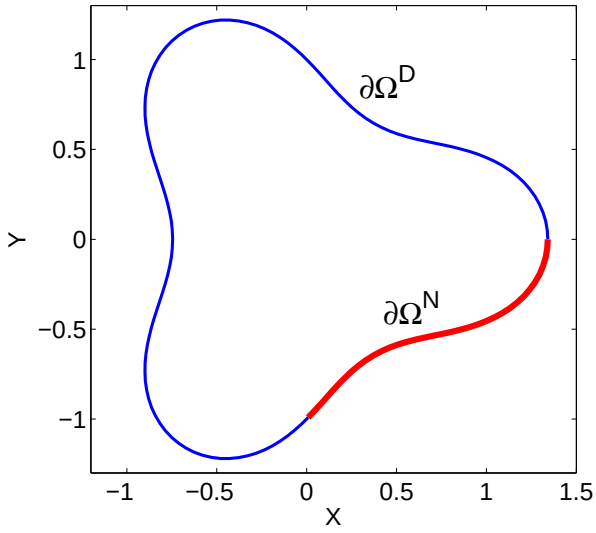


Figure 3.4: Example 3.2.3: The profile of the domain and its boundary conditions.

In the numerical implementation, we choose 545 uniformly distributed interior points, 116 points on the Dirichlet boundary, 34 points on the Neumann boundary, and 350 randomly selected test points inside the domain.

We observed similar results in accuracy and stability as in the previous example. Due to the mixed boundary conditions, the accuracy is even more sensitive to the shape parameter in the case of the MPS using MQ as shown in Figure 3.5a. On the other hand, as shown in Figure 3.5b, not only is the accuracy using a polynomial basis much better than the MPS using MQ but so is the stability, which is extremely important in the numerical computation.

Example 3.2.4. In this example, we perform numerical tests on various geometric domains as shown in Figure 3.1. We consider the following partial differential equation.

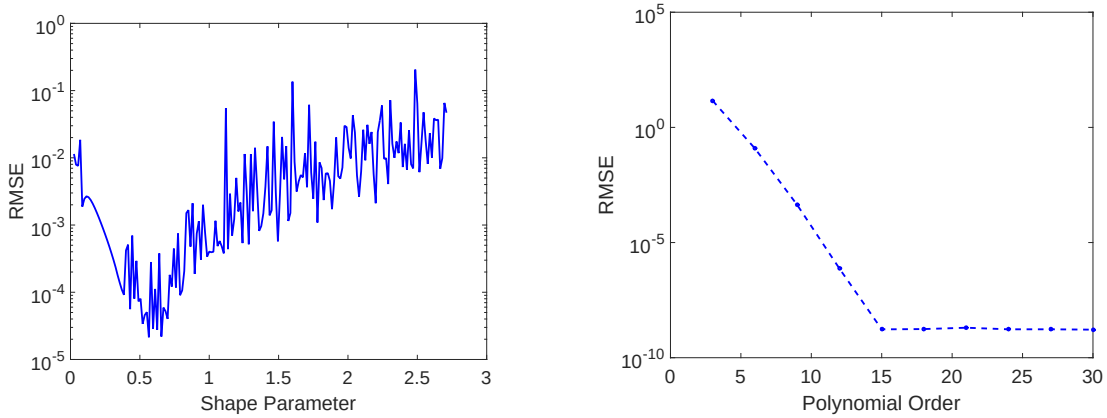
$$2\frac{\partial^2 u(x,y)}{\partial x^2}+3\frac{\partial^2 u(x,y)}{\partial x\partial y}+3\frac{\partial^2 u(x,y)}{\partial y^2}+7u(x,y) = f(x,y),\ (x,y)\in\Omega, \quad (3.17)$$

$$u(x,y) = g(x,y),\ (x,y)\in\partial\Omega, \quad (3.18)$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y)=e^{2x+3y},\ (x,y)\in\overline{\Omega}.$$

In Table 3.2, we present results for five different irregular domains as shown in Figure 3.1. Since the areas of these five domains are quite different, the maximum forcing terms



(a) RBF (b) Polynomial

Figure 3.5: Example 3.2.3: (a) RMSE versus the shape parameter of the MQ. (b) RMSE versus the polynomial order.

and exact solutions have a large discrepancy. Hence, it is more appropriate to use relative error to measure the accuracy of our computation. Overall, from Table 3.2, it appears that the smoothness of the boundary has little impact on the numerical accuracy. The order of polynomial remains a dominating factor for performance on the proposed algorithm.

Table 3.2: Example 3.2.4: RMSE and relative error (Rel Err) for different computational domains.

Domains	n_i	n_b	RMSE	Rel Err	Polynomial Order
Amoeba	294	100	2.072e-06	6.974e-09	18
Cassini	313	100	6.430e-08	4.892e-09	17
Gear-Shaped	296	100	5.952e-05	1.562e-07	21
L-shaped	300	96	2.641e-10	6.443e-11	13
Corner-Shaped	330	90	8.867e-08	2.979e-09	13

Example 3.2.5. In this example, we consider the following fourth order boundary value problem from [82].

$$(\Delta^2 - 100)u(x,y) = f(x,y), \ (x,y) \in \Omega, \tag{3.19}$$

$$u(x,y) = g(x,y), \ (x,y) \in \partial\Omega, \tag{3.20}$$

$$\Delta u(x,y) = h(x,y), \ (x,y) \in \partial\Omega, \tag{3.21}$$

where $f(x,y), g(x,y)$ and $h(x,y)$ are given based on the following analytical solution

$$u(x,y) = \sin(\pi x)\cosh(y) - \cos(\pi x)\sinh(y),\ (x,y) \in \overline{\Omega}.$$

The computational domain is the Cassini (three) as shown in Figure 3.4.

In order to compare our results to those of [82], we choose the same number of collocation points in both approaches. To measure the numerical accuracy, we choose 230 random test points to compute the errors. In Table 5.3, we observe that the proposed algorithm is far more accurate than the results obtained in [82] where polyharmonic splines of order 3 were used. In this example, the proposed algorithm is also very effective for solving fourth order partial differential equations in an irregular domain.

Table 3.3: Example 3.2.5: Comparison of RMSE and RMSEx with polynomial basis functions and polyharmonic splines.

	Polyharmonic Splines ($r^6 \ln r$)		Polynomial Basis Functions		
(n_i, n_b)	RMSE	RMSEx	RMSE	RMSEx	order
(126,80)	2.440e-05	8.585e-05	8.382e-11	2.968e-10	22
(208,140)	5.887e-06	3.482e-05	5.441e-12	1.539e-11	23
(374,200)	1.643e-06	1.710e-05	3.605e-13	1.295e-12	23

3.3 Numerical Results for Poisson-type problems and PDEs with variable coefficients

In Section 3.2, we solved partial differential equations with constant coefficients. Theorem 2.2.1 does not work if the coefficients of the PDEs are variable or $a_6 = 0$. Therefore, we were not able to solve PDEs with variable coefficients and Poisson problem. In this section, a new scheme is presented to solve such problems using the particular solutions obtained from Theorem 2.2.1.

If $\mathcal{L}$ is an operator with variable coefficients, we can rewrite equations (2.14) as follows:

$$\mathcal{L}'u(x,y) + u(x,y) + \mathcal{L}u(x,y) \quad = \quad \mathcal{L}'u(x,y) + u(x,y) + f(x,y),$$

or

$$(\mathcal{L}' + 1)u(x,y) \quad = \quad (\mathcal{L}' - \mathcal{L} + 1)u(x,y) + f(x,y), \tag{3.22}$$

where $\mathcal{L}'$ is a differential operator with constant coefficients. Then the system of equations (2.14) and (2.15) takes the form:

$$\begin{aligned}(\mathcal{L}'+1)u(x,y) &= (\mathcal{L}'-\mathcal{L}+1)u(x,y)+f(x,y),\ (x,y)\in\Omega &(3.23)\\ u(x,y) &= g(x,y),\ (x,y)\in\Gamma. &(3.24)\end{aligned}$$

Now, using the MPS, we approximate the solution to the problem (3.23)-(3.24) by

$$\mathbf{u}(x,y)\approx\hat{\mathbf{u}}(x,y)=\sum_{i=0}^{s}\sum_{j=0}^{i}a_{ij}u_p^{ij}(x,y), \tag{3.25}$$

where

$$(\mathcal{L}'+1)u_p^{ij}(x,y)=x^{i-j}y^j,\ 0\le j\le i, 0\le i\le s.$$

Then, using the approximation in (3.25) for $\mathbf{u}(x,y)$, equations (3.23) and (3.24) form a linear system of equations in the following way:

$$\sum_{i=0}^{s}\sum_{j=0}^{i}a_{ij}(\mathcal{L}'+1)u_p^{ij}(x_k,y_k)=\sum_{i=0}^{s}\sum_{j=0}^{i}a_{ij}(\mathcal{L}'-\mathcal{L}+1)u_p^{ij}(x_k,y_k)+f(x_k,y_k),\ k=1,2,...,n_i.$$

It follows that

$$\sum_{i=0}^{s}\sum_{j=0}^{i}a_{ij}x_k^{i-j}y_k^j=\sum_{i=0}^{s}\sum_{j=0}^{i}a_{ij}(\mathcal{L}'-\mathcal{L}+1)u_p^{ij}(x_k,y_k)+f(x_k,y_k),\ k=1,2,...,n_i \tag{3.26}$$

and

$$\sum_{i=0}^{s}\sum_{j=0}^{i}a_{ij}\mathcal{B}u_p^{ij}(x_k,y_k)=g(x_k,y_k),\ k=n_i+1,n_i+2,...,n. \tag{3.27}$$

Equations (3.26) and (3.27) can be solved to find the undetermined coefficients

$$\{a_{ij}\}=\{a_{00},a_{10},a_{11},a_{20},a_{21},a_{22},\cdots,a_{ss}\}.$$

The approximate solutions can be obtained from equation (3.25) using the calculated values of $\{a_{ij}\}$.

In Section 3.2, we noticed that multiple scale technique is essential to control condition number of the resulting matrix and improve the numerical accuracies when using the polynomial basis functions. Therefore, in this section, all the numerical results are produced using multiple scale technique.

Example 3.3.1. Let us consider a Poisson equation with mixed boundary conditions:

$$\Delta u(x,y) = f(x,y),\ (x,y) \in \Omega, \tag{3.28}$$

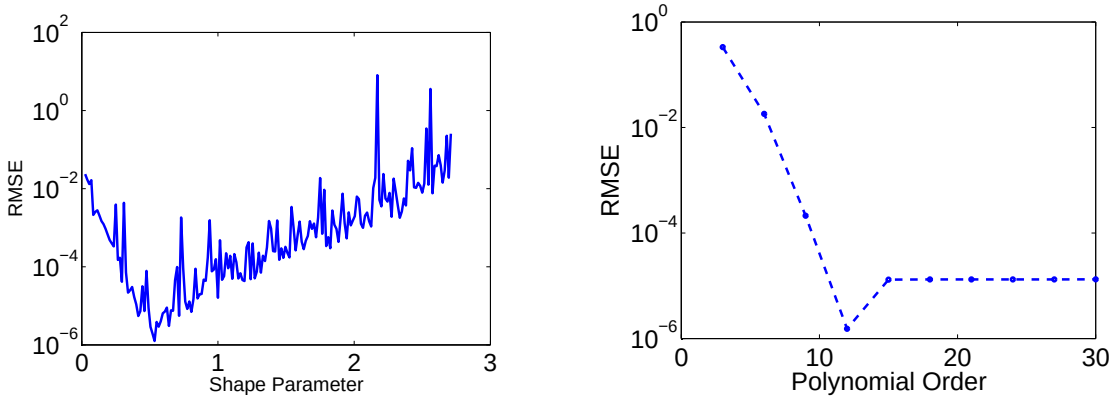
$$u(x,y) = g(x,y),\ (x,y) \in \partial\Omega^D, \tag{3.29}$$

$$\frac{\partial u}{\partial \mathbf{n}} = h(x,y),\ (x,y) \in \partial\Omega^N, \tag{3.30}$$

where $\mathbf{n}$ is the unit outward normal vector, $f(x,y)$, $g(x,y)$, and $h(x,y)$ are given based on the following analytical solution

$$u(x,y) = \sin(4x)\cos(4y),\ (x,y) \in \overline{\Omega}.$$

The boundaries $\partial\Omega^D$ and $\partial\Omega^N$ denote the boundaries on which the Dirichlet and Neumann conditions are applied respectively such that $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N$, $\partial\Omega^D \cap \partial\Omega^N = \emptyset$. We have chosen a unit square to be the domain. The number of interior points, boundary points and test points are 625, 200, and 400 respectively. We have taken 150 points on the Dirichlet boundary and 50 points on the Neumann boundary. The test points are randomly distributed inside the domain. The problem is solved using the MPS based on radial basis function MQ and polynomial basis functions. Figure 3.6a shows the effect of indeterminate shape parameter on the RMSE. In Figure 3.6b, we can see that the RMSE is very stable for the higher order polynomial basis functions. It shows that our approach work for Poisson's problem very effectively.



(a) RBF (b) Polynomial

Figure 3.6: Example 3.3.1: (a) RMSE versus the shape parameter using RBF MQ. (b) RMSE versus the polynomial order using polynomial basis.

Example 3.3.2. Let us consider the following partial differential equation with variable coefficients.

$$\begin{aligned} \Delta u(x,y) + x^2 y \frac{\partial u(x,y)}{\partial x} - y\cos(y)\frac{\partial u(x,y)}{\partial y} &= f(x,y),\ (x,y) \in \Omega, \\ u(x,y) &= g(x,y),\ (x,y) \in \partial\Omega, \end{aligned}$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y) = \cos(\pi x) - \cos(\pi y/2),\ (x,y) \in \overline{\Omega}.$$

In this example, we have compared the numerical accuracy of the MPS based on RBF MQ and the polynomial basis functions on various computational domains. Cassini, Amoeba and Gear-Shaped domains are considered for the numerical experiment. The number of interior points taken on Cassini, Amoeba and Gear-Shaped domains are 429, 438 and 434 respectively. 150 boundary points are taken on each domain. The number of test points are 376, 365, 374 respectively. Since the domains are of different sizes, there is a large difference in the maximum forcing terms and exact solutions among these domains. Hence, for a fair comparison, we have used the relative error to observe the effectiveness of our algorithm on these domains. Figure 3.7a shows the profiles of the relative errors for different values of the shape parameter of MQ for three computational domains. Similarly, Figure 3.7b shows the comparison of the relative error of the method defined on three different domains for different polynomial orders. The results are consistent with the geometrical complexity of the computational domains. In this experiment, we can see more stable and accurate results from our approach in comparison to the MPS using RBF MQ.

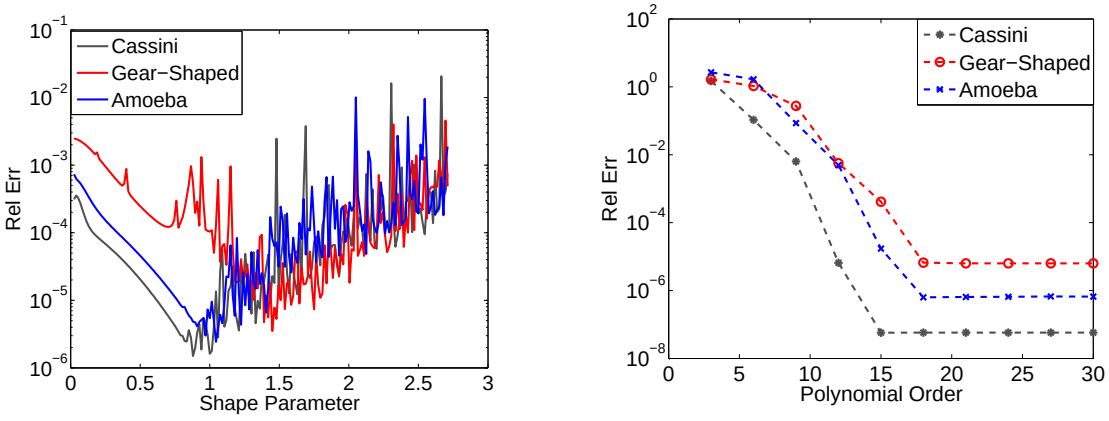
Example 3.3.3. Let us consider the following fourth-order boundary value problem with Dirichlet and Neumann boundary condition.

$$\begin{aligned} \Delta^2 u(x,y) + y\cos(y)\frac{\partial u(x,y)}{\partial x} + \sinh(x)\frac{\partial u(x,y)}{\partial y} + x^2 y^3 u(x,y) &= f(x,y),\ (x,y) \in \Omega, \\ u(x,y) &= g(x,y),\ (x,y) \in \partial\Omega, \\ \frac{\partial u}{\partial \mathbf{n}} &= h(x,y),\ (x,y) \in \partial\Omega^N, \end{aligned}$$

where $\mathbf{n}$ is the unit outward normal vector, $f(x,y)$, $g(x,y)$, and $h(x,y)$ are given based on the following analytical solution

$$u(x,y) = y\sin(x) + x\cos(y),\ (x,y) \in \overline{\Omega}.$$

The boundaries $\partial\Omega^D$ and $\partial\Omega^N$ denote the boundaries on which the Dirichlet and Neumann conditions are applied respectively such that $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N, \partial\Omega^D \cap \partial\Omega^N = \emptyset$.



(a) RBF (b) Polynomial

Figure 3.7: Example 3.3.2: (a) Rel Err versus the shape parameter using RBF MQ. (b) Rel Err versus the polynomial order using polynomial basis.

The problem is defined on Cassini (Three). The number of interior points and test points are 545 and 350 respectively. We have taken 120 points on the Dirichlet boundary and 30 points on the Neumann boundary. The profile of RMSE and RMSEx with different polynomial order is given in Figure (3.8). This result shows the efficacy of our approach to approximate the solutions and the derivative of the solutions of the fourth-order boundary value problems.

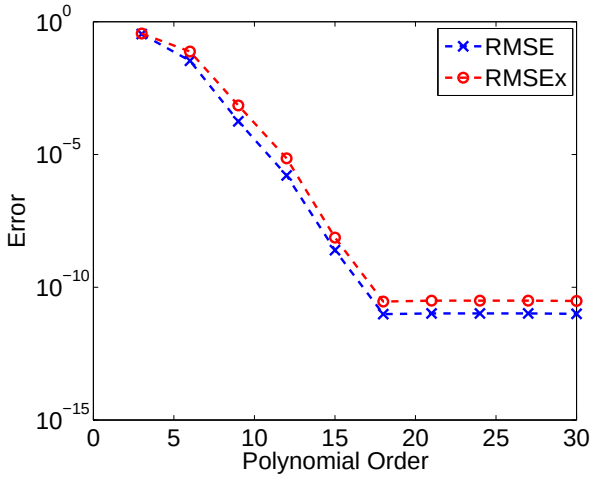


Figure 3.8: Example 3.3.3: RMSE and RMSEx versus the polynomial order.

Example 3.3.4. In this example, we solve a fourth-order boundary value problem with

Dirichlet and Laplacian boundary condition. Consider the following problem.

$$\begin{aligned}\Delta^2 u(x,y) + 2y\sin(x)\frac{\partial u(x,y)}{\partial x} - y\cos(x)\frac{\partial u(x,y)}{\partial y} + xyu(x,y) &= f(x,y),\ (x,y)\in\Omega,\\ u(x,y) &= g(x,y),\ (x,y)\in\partial\Omega,\\ \Delta u(x,y) &= h(x,y),\ (x,y)\in\partial\Omega,\end{aligned}$$

where $f(x,y), g(x,y)$ and $h(x,y)$ are given based on the following analytical solution

$$u(x,y) = \sin(\pi x)\cosh(y) - \cos(\pi x)\sinh(y),\ (x,y)\in\overline{\Omega}.$$

An amoeba-shaped domain is considered for numerical computation. The number of interior points, boundary points and test points are 468, 150 and 386 respectively. The profile of RMSE and RMSEx with different polynomial order is given in Figure (3.9). The results on

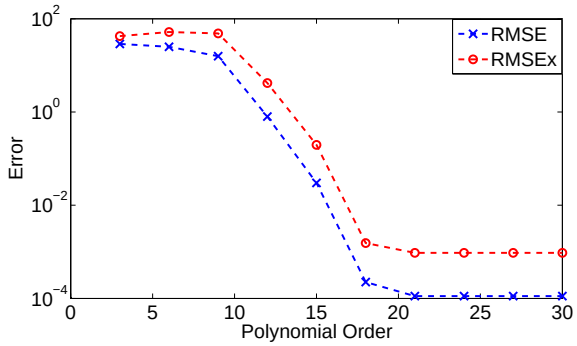


Figure 3.9: Example 3.3.4: RMSE and RMSEx versus the polynomial order.

the figure show that our algorithm is very effective for solving fourth order boundary value problem with Dirichlet and Laplacian boundary conditions.

3.3.1 Cauchy-Navier Equation of Elasticity

In this section, we solve a Cauchy-Navier equation of elasticity subject to a Dirichlet boundary condition [33]. The problem is as follows:

$$\mu\Delta u_1 + \frac{\mu}{1-2\nu}\left(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_2}{\partial xy}\right) = f_1(x,y),\ (x,y)\in\Omega, \tag{3.31}$$

$$\frac{\mu}{1-2\nu}\left(\frac{\partial^2 u_1}{\partial x\partial y} + \frac{\partial^2 u_2}{\partial y^2}\right) + \mu\Delta u_2 = f_2(x,y),\ (x,y)\in\Omega \tag{3.32}$$

with the Dirichlet boundary conditions

$$u_1(x,y) = g_1(x,y),\ (x,y)\in\partial\Omega, \tag{3.33}$$

$$u_2(x,y) = g_2(x,y),\ (x,y)\in\partial\Omega, \tag{3.34}$$

where the constants $\mu \geq 0$ and $\nu \in [0, 1/2]$ are called shear modulus and Poisson's ratio. Let us see the formulation of the problem using our approach. we can rewrite equations (3.31) and (3.32) as follows:

$$2\mu\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial^2 u_1}{\partial x^2}+\mu\frac{\partial^2 u_1}{\partial y^2}+\left(\frac{\mu}{1-2\nu}\right)\frac{\partial^2 u_2}{\partial xy}=f_1(x,y),\ \ (x,y)\in\Omega, \tag{3.35}$$

$$\left(\frac{\mu}{1-2\nu}\right)\frac{\partial^2 u_1}{\partial xy}+2\mu\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial^2 u_2}{\partial y^2}+\mu\frac{\partial^2 u_2}{\partial y^2}=f_2(x,y),\ \ (x,y)\in\Omega \tag{3.36}$$

Furthermore,

$$\begin{aligned}\left(2\mu\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial^2 u_1}{\partial x^2}+\mu\frac{\partial^2 u_1}{\partial y^2}+u_1(x,y)\right) &- u_1(x,y)\\ +\left(\left(\frac{\mu}{1-2\nu}\right)\frac{\partial^2 u_2}{\partial xy}+u_2(x,y)\right)-u_2(x,y) &= f_1(x,y),\ \ (x,y)\in\Omega, \quad (3.37)\\ \left(\frac{\mu}{1-2\nu}\right)\frac{\partial^2 u_1}{\partial xy}+2\mu\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial^2 u_2}{\partial y^2}+\mu\frac{\partial^2 u_2}{\partial y^2} &= f_2(x,y),\ \ (x,y)\in\Omega. \quad (3.38)\end{aligned}$$

Let

$$L_{11} := 2\mu\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial^2}{\partial x^2}+\mu\frac{\partial^2}{\partial y^2}+1 \tag{3.39}$$

$$L_{12} := \left(\frac{\mu}{1-2\nu}\right)\frac{\partial^2}{\partial xy}+1 \tag{3.40}$$

$$L_{21} := \left(\frac{\mu}{1-2\nu}\right)\frac{\partial^2}{\partial xy} \tag{3.41}$$

$$L_{22} := 2\mu\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial^2}{\partial y^2}+\mu\frac{\partial^2}{\partial y^2}. \tag{3.42}$$

Then, equations (3.37) and (3.38) become

$$\begin{aligned}L_{11}u_1(x,y)-u_1(x,y)+L_{12}u_2(x,y)-u_2(x,y) &= f_1(x,y),\ \ (x,y)\in\Omega, \quad (3.43)\\ L_{21}u_1(x,y)+L_{22}u_1(x,y) &= f_2(x,y),\ \ (x,y)\in\Omega, \quad (3.44)\end{aligned}$$

subject to

$$\begin{aligned}u_1(x,y) &= g_1(x,y),\ \ (x,y)\in\partial\Omega, \quad (3.45)\\ u_2(x,y) &= g_2(x,y),\ \ (x,y)\in\partial\Omega. \quad (3.46)\end{aligned}$$

Let us construct particular solutions $u_{p1}^{(ij)}$ and $u_{p2}^{(ij)}$ such that

$$L_{11}u_{p1}^{(ij)}=x^{i-j}y^j,\ \ L_{21}u_{p2}^{(ij)}=x^{i-j}y^j,\ \ 0\leq i\leq s,\ 0\leq j\leq i. \tag{3.47}$$

By the MPS, we approximate the solutions $u_1(x,y)$ and $u_2(x,y)$ by

$$\hat{u}_1(x,y) = \sum_{i=0}^{s}\sum_{j=0}^{i} a_{ij}u_{p1}^{ij}(x,y), \tag{3.48}$$

$$\hat{u}_2(x,y) = \sum_{i=0}^{s}\sum_{j=0}^{i} b_{ij}u_{p2}^{ij}(x,y). \tag{3.49}$$

Let $\{(x_k,y_k)\}_{k=1}^{n_i}$ be the set of interior points in Ω and $\{(x_k,y_k)\}_{k=n_i+1}^{n}$ be the boundary points in $\partial\Omega$. Then from equations (3.43) and (3.44), we have

$$\sum_{i=0}^{s}\sum_{j=0}^{i} a_{ij}x_k^{i-j}y_k^j - u_{p1}^{ij}(x_k,y_k) + \sum_{i=0}^{s}\sum_{j=0}^{i} b_{ij}x_k^{i-j}y_k^j - u_{p2}^{ij}(x_k,y_k) = f_1(x_k,y_k), \tag{3.50}$$

$$\sum_{i=0}^{s}\sum_{j=0}^{i} a_{ij}L_{21}u_{p1}^{ij}(x_k,y_k) + \sum_{i=0}^{s}\sum_{j=0}^{i} b_{ij}L_{22}u_{p2}^{ij}(x_k,y_k) = f_2(x_k,y_k), \tag{3.51}$$

for $1 \le k \le n_i$.

For the Dirichlet boundary condition, we have

$$\sum_{i=0}^{s}\sum_{j=0}^{i} a_{ij}u_{p1}^{ij}(x_k,y_k) = g_1(x_k,y_k), \tag{3.52}$$

$$\sum_{i=0}^{s}\sum_{j=0}^{i} b_{ij}u_{p2}^{ij}(x_k,y_k) = g_2(x_k,y_k), \tag{3.53}$$

for $n_i+1 \le k \le n$. Equations (3.50) – (3.53) can be solved to evaluate $\{a_{00},a_{10},a_{11},...,a_{ss}\}$ and $\{b_{00},b_{10},b_{11},...,b_{ss}\}$. These weighting coefficients can be applied in equations (3.48) and (3.49) to obtain the values of approximate solutions.

Example 3.3.5. As a numerical experiment, we solve the above problem with $\mu = 1$ and $\nu = 0.3$.The exact solutions considered for this problems are

$$u_1(x,y) = e^{x+y}, \ (x,y) \in \bar{\Omega}, \tag{3.54}$$

$$u_2(x,y) = \sin(x+y), \ (x,y) \in \bar{\Omega}. \tag{3.55}$$

The domain is a unit square with uniform interior and boundary points and random test points. The number of interior points, boundary points, and test points are 361, 88, 324 respectively. The profile of relative error with different polynomial order for both solutions is given in Figure 3.10.

Table 3.4 shows the numerical accuracy of the problem defined on various kinds of domains.

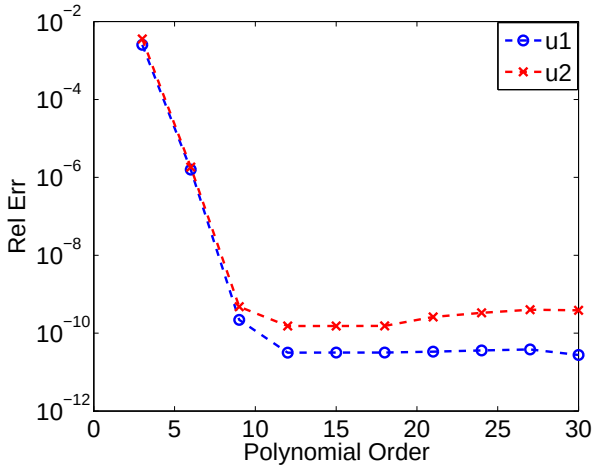


Figure 3.10: Example 3.3.5: Relative error versus the polynomial order.

Table 3.4: Example 3.3.5: Relative error for different computational domains.

Domains	n_i	n_b	Rel Err(u_1)	Rel Err(u_2)	Polynomial Order
Amoeba	294	100	7.966e-10	9.839e-10	15
Cassini	313	100	3.172e-12	4.354e-12	15
Gear-Shaped	296	100	5.372e-10	2.977e-09	15
L-shaped	300	96	2.161e-11	4.216e-11	9
Star-Shaped	317	100	1.689e-10	7.034e-10	15
Corner-Shaped	330	90	1.360e-10	3.633e-10	9

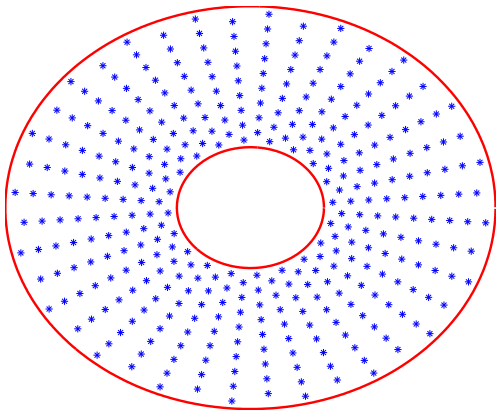


Figure 3.11: Example 3.3.5: The profile of annular domain.

Moreover, the numerical accuracy of our method is compared with that of Kansa-radial basis function method [56]. The annular domain considered in [56] is given in Figure 3.11. The relative error obtained from Kansa-radial basis function method and the MPS with polynomial basis functions are given in Table 3.5. The comparison shows that the results obtained from the MPS is much accurate than that obtained from the Kansa-radial basis function method.

Table 3.5: Example 3.3.5: Comparison of Rel Err for the Kansa-radial basis function method and the MPS with polynomial basis functions.

	Kansa Method		The MPS with polynomial basis functions		
(n_i, n_b)	Rel Err(u_1)	Rel Err(u_2)	Rel Err(u_1)	Rel Err(u_2)	order
(6240,160)	1.368e-05	5.191e-05	1.342e-09	3.888e-09	12
(9800,200)	1.212e-05	1.736e-05	1.186e-09	3.813e-09	12
(14160,240)	2.877e-06	8.399e-06	1.660e-09	5.256e-09	12
(19320,280)	3.262e-06	1.173e-05	1.187e-09	3.814e-09	12
(25280,320)	4.089e-06	6.032e-06	1.016e-08	3.251e-08	12

In this section, we solved various kinds of boundary value problems with variable coefficients using our algorithm. The numerical accuracies of the solutions and the derivative of the solutions of these problems look very stable and accurate for higher order polynomial basis functions.

3.4 Nonlinear Problems

In this section, we have employed our approach to solve nonlinear partial differential equations. Three numerical examples are solved on regular and irregular domains to show the effectiveness of the proposed scheme. Since the problems we considered are nonlinear, we used the Picard method to carry out the iteration [13]. The iteration is summarized as follows:

Algorithm 2

Step 1: Construct a sequence $\left\{\mathbf{u}^{(i)} : i \in \mathbb{N} \cup \{0\}\right\}$ such that
$\mathbf{u}^{(0)} = 0$ and
$\Delta\mathbf{u}^{(i+1)} = \mathbf{f}(\mathbf{u}^{(i)})$ *in* Ω,
$\mathbf{u}^{(i+1)} = \mathbf{g}$ *on* $\partial\Omega$.

Step 2: At each iteration, compute $\mathbf{u}^{(i+1)}$ using modified MPS.

Step 3: If $\left|\mathbf{u}^{(i+1)} - \mathbf{u}^{(i)}\right| < \varepsilon,\ \varepsilon > 0,$
stop,
end.

Step 4: $\mathbf{u}^{(i+1)}$ at the final iteration is the required approximate solution.

For all the numerical results, we have chosen the tolerance $\varepsilon = 10^{-4}$ in Algorithm 2 to ensure the accuracy of the solution.

Example 3.4.1. Let us consider the nonlinear Poisson problem with Dirichlet boundary condition:

$$\Delta u(x,y) = u^2 + 6x - x^6 - 4x^4y - 4x^2y^2,\ (x,y) \in \Omega, \tag{3.56}$$

$$u(x,y) = g(x,y),\ (x,y) \in \partial\Omega, \tag{3.57}$$

where $g(x,y)$ is given based on the following exact solution

$$u_{exact}(x,y) = x^3 + 2xy.$$

In this example, we have tested our method on the Poisson-type boundary value problem which includes the space variables x and y in the source term. The computational domain is the Cassini as shown in Figure 3.1b. The number of interior points and boundary points are 460 and 150 respectively. In Figure 3.12, we observed that the numerical accuracy improves with higher order of polynomial basis functions. As in previous examples, the numerical accuracy remains stable when higher order polynomial basis functions are taken. We have compared the numerical results for several sets of collocation points on the same domain. Table 3.6 shows the RMSE and MAE for various sets of collocation points with polynomial order 15. We see that the numerical accuracy does not improve with an increase in the number of collocation points. We need only 5 iterations to achieve excellent result. Thus, from this experiment, we observe that our algorithm is very efficient and accurate to solve nonlinear problems as well.

Example 3.4.2. In this example, we test our algorithm on a Helmholtz nonlinear problem

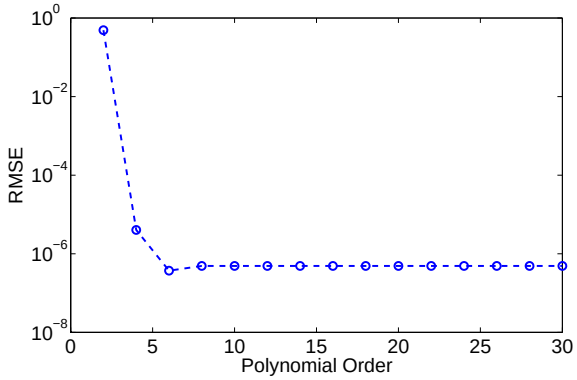


Figure 3.12: Example 3.4.1: RMSE versus the polynomial order.

Table 3.6: Example 3.4.1: The RMSE and Maximum absolute error with a various set of collocation points.

(n_i, n_b)	RMSE	MAE	# Iterations
$(313, 100)$	$4.92e-07$	$1.39e-06$	5
$(460, 150)$	$4.90e-07$	$1.39e-06$	5
$(1108, 250)$	$4.90e-07$	$1.39e-06$	5
$(1973, 450)$	$4.90e-07$	$1.39e-06$	5
$(3529, 600)$	$4.89e-07$	$1.39e-06$	5

with mixed boundary conditions. Let us consider the problem:

$$\Delta u(x,y) + u(x,y) = 4u^3 + \frac{1}{(4+x+y)}, \ (x,y) \in \Omega, \tag{3.58}$$

$$u(x,y) = g(x,y), \ (x,y) \in \partial\Omega^D, \tag{3.59}$$

$$\frac{\partial u}{\partial \mathbf{n}} = h(x,y), \ (x,y) \in \partial\Omega^N, \tag{3.60}$$

where $\mathbf{n}$ is the unit outward normal vector, $f(x,y)$, $g(x,y)$, and $h(x,y)$ are given based on the following analytical solution

$$u_{exact}(x,y) = \frac{1}{(4+x+y)}, \ (x,y) \in \overline{\Omega}.$$

The domain considered is a Cassini (three). The number of interior points, Dirichlet's and Neumann's boundary points are 545, 80 and 70 respectively. The RMSEs of the solution with different polynomial orders have been presented in Figure 3.13. The behavior of the approximate solutions to this problem is very similar to that of our previous examples. The

numerical accuracy improves with the order of the basis functions and eventually the error remains stable when the order of basis functions increases.

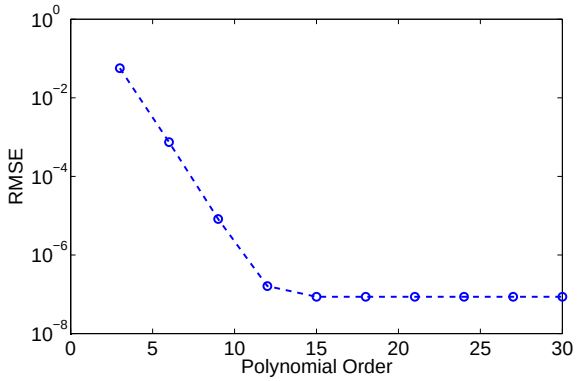


Figure 3.13: Example 3.4.2: RMSE versus the polynomial order.

Example 3.4.3. Let us consider the modified Helmholtz problem with a Dirichlet boundary condition:

$$\Delta u(x,y) - u(x,y) = 4u^3 - \frac{1}{(4+x+y)}, \quad (x,y) \in \Omega, \tag{3.61}$$

$$u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega, \tag{3.62}$$

where $g(x,y)$ is given based on the following analytical solution

$$u_{exact}(x,y) = \frac{1}{(4+x+y)}.$$

In Figure 3.14, we see the error for different orders of polynomial basis functions in various computational domains. The number of points in Cassini, gear-shape domain, star-shape domain, and amoeba are 429, 434, 438, 438 respectively. The number of boundary points on each domain is 150. We can see that the numerical accuracy is getting better when the order of the basis functions is increasing. Furthermore, the numerical results are consistent with the geometrical complexity of the domains. Table 3.7 shows the RMSE and maximum absolute error for the above problem defined on various domains when polynomial basis functions of order 10 are employed. We can see excellent numerical results with a small number of iterations. This proves that our algorithm is very efficient to solve such problems on irregular domains. We note that the number of iterations is higher for complicated domains.

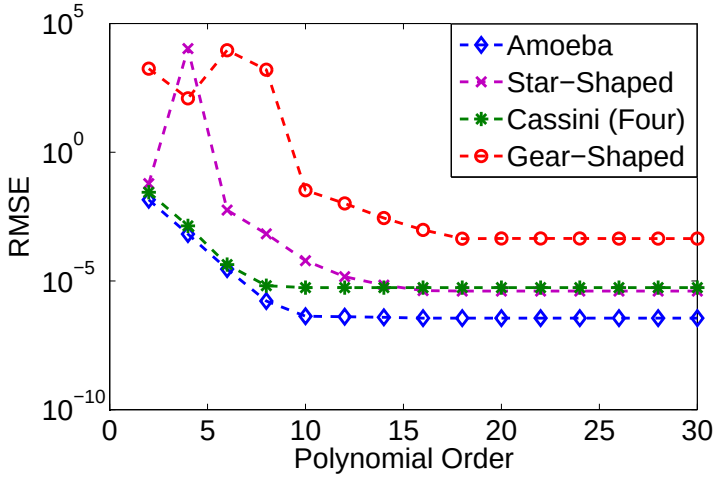


Figure 3.14: Example 3.5.3: RMSE versus the polynomial order.

Table 3.7: Example 3.4.3: The RMSE and Maximum error with different domains.

Domain	(n_i, n_b)	# Iterations	RMSE	MAE
Amoeba	$(438, 150)$	4	$4.26e-07$	$1.03e-06$
Star-Shaped	$(438, 150)$	5	$6.11e-05$	$2.80e-04$
Cassini (Four)	$(429, 150)$	4	$5.48e-06$	$1.15e-05$
Gear-Shaped	$(434, 150)$	33	$3.31e-02$	$1.88e-01$

3.5 Time-Dependent Problems

In this section, we solve time-dependent problems using the MPS with polynomial basis functions. For the time discretization, we have used the Houbolt method, a third order finite difference time marching scheme. The Houbolt method transforms the time-dependent problem into a series of inhomogeneous modified Helmholtz equations, which are solved by our numerical approach. Three numerical examples are presented to demonstrate the effectiveness of the proposed scheme.

3.5.1 Reaction-Diffusion Equations

Example 3.5.1. Consider the parabolic equation

$$\begin{aligned}
\frac{\partial u}{\partial t}(x,y,t) &= \Delta u(x,y,t) + (2\cos(t) - \sin(t))\sin(x)\sin(y) \quad (x,y) \in \Omega, t \geq 0, \\
u(x,y,t) &= \sin(x)\sin(y)\cos(t), \quad (x,y) \in \partial\Omega, t \geq 0, \\
u(x,y,0) &= \sin(x)\sin(y), \quad (x,y) \in \Omega.
\end{aligned}$$

where the analytical solution is given by

$$u(x,y) = \sin(x)\sin(y)\cos(t).$$

The domain is the amoeba-like domain as shown in Figure 3.1a. The numerical results are observed with terminal time $t = 10$. The number of interior points, boundary points, and test points are 316, 100 and 234 respectively. In Figure 3.15, we have compared the RMSE

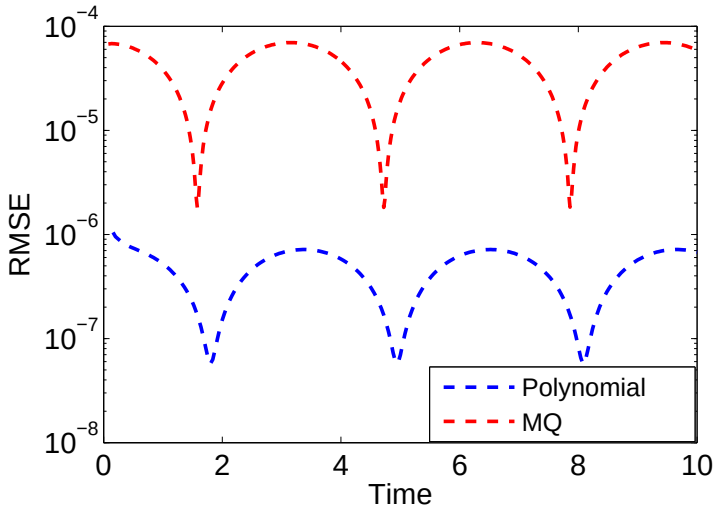


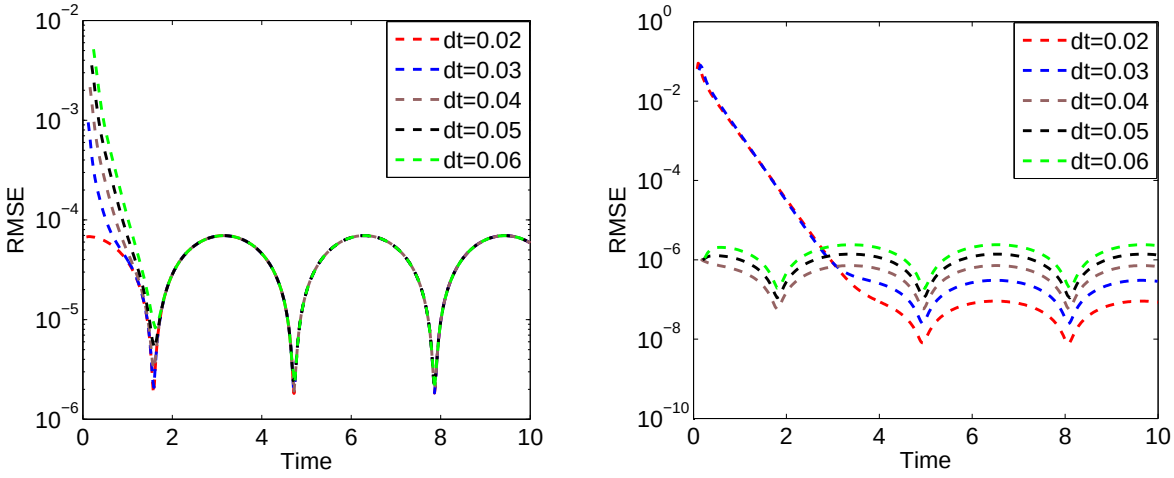
Figure 3.15: Example 3.5.1: The profile of the RMSE vs Time using radial basis function MQ and polynomial basis functions of order 20.

obtained by the MPS using radial basis function MQ and using polynomial basis functions of order 20. The time step chosen for the MPS with polynomial basis functions and radial basis function are 0.04 and 0.02 respectively. We can see that the MPS with polynomial basis functions is far better than the traditional MPS with radial basis function MQ.

The problem is also solved by using the MPS based on radial basis function MQ and based on polynomial basis functions for various time steps. Figure 3.16 shows that our approach gives a better result than that obtained from the MPS with radial basis function MQ.

Furthermore, we have computed the RMSE for the problem with various computational domains. Table 3.8 shows very nice results for the problem defined on various irregular domains. The numerical accuracy and the polynomial orders are consistent with the complexity of the domains.

Example 3.5.2. In this example, we consider another time-dependent problem defined on a



(a) RBF (b) Polynomial

Figure 3.16: Example 3.5.1: The profile of the RMSE vs Time for different time steps.

Table 3.8: Example 3.5.1: RMSE for different computational domains.

Domains	n_i	n_b	RMSE	Polynomial Order
Amoeba	316	100	5.191e-07	20
Cassini	336	100	6.904e-08	20
Gear-Shaped	475	170	6.745e-07	25
L-shaped	437	140	3.267e-09	19
Star-Shaped	317	120	2.942e-07	19

square domain. Consider the parabolic equation

$$\begin{aligned}
\frac{\partial u}{\partial t}(x,y,t) &= \Delta u(x,y,t) + (2\sin(t)+\cos(t))\sin(x)\sin(y) \quad (x,y)\in\Omega, t\geq 0,\\
u(x,y,t) &= \sin(x)\sin(y)\sin(t), \ (x,y)\in\partial\Omega, t\geq 0,\\
u(x,y,0) &= 0, \quad (x,y)\in\Omega.
\end{aligned}$$

where the analytical solution is given by

$$u(x,y) = \sin(x)\sin(y)\sin(t).$$

The computational domain is given by $[-2,2]\times[-2,2]$. The numerical results were observed with terminal time $t = 10$. The number of interior points, boundary points and the test points are 400, 156 and 529 respectively. The time step chosen for the MPS with polynomial basis functions and radial basis function are 0.004 and 0.02 respectively. In

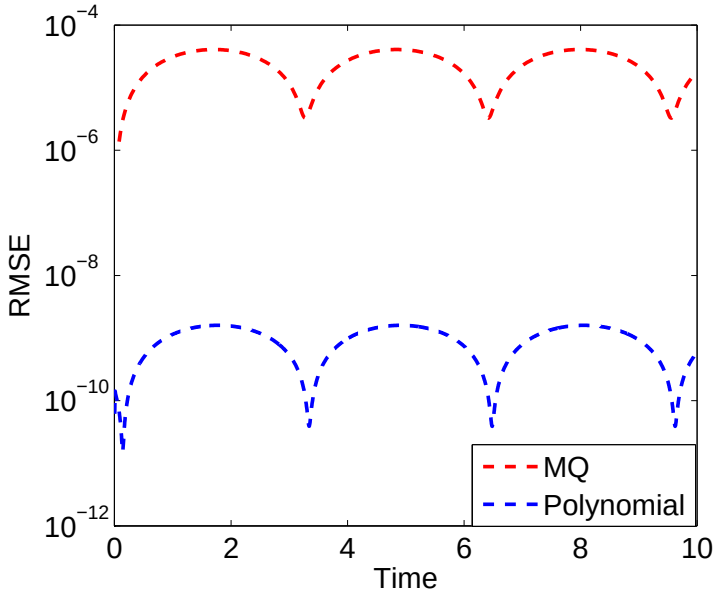


Figure 3.17: Example 3.5.2: The profile of the RMSE vs Time using radial basis function MQ and polynomial basis functions of order 21.

Figure 3.17, we see that the RMSE with polynomial basis functions is far better than that with radial basis function MQ. Polynomial basis functions of order 21 were employed in the computation.

3.5.2 Wave Equation

Example 3.5.3. Consider the wave equation

$$\begin{aligned}
\frac{\partial^2 u}{\partial t^2}(x,y,t) &= \Delta u(x,y,t) + f(x,y,t) \quad (x,y) \in \Omega, t > 0, \\
u(x,y,t) &= x(1-x)y(1-y)\cos(t), \quad (x,y) \in \partial\Omega, t > 0, \\
u(x,y,0) &= x(1-x)y(1-y), \quad (x,y) \in \Omega, \\
\frac{\partial u}{\partial t}(x,y,0) &= 0, \quad (x,y) \in \Omega,
\end{aligned}$$

where the analytical solution is given by

$$u(x,y) = x(1-x)y(1-y)\cos(t)$$

and

$$f(x,y,t) = (2x(1-x) + 2y(1-y) - x(1-x)y(1-y))\cos(t).$$

The computational domain is given by $[0,1] \times [0,1]$. The number of interior points, boundary points and the test points are 400, 156 and 361 respectively. We chose $dt = 0.1$. In

Figure 3.18, we have shown the profiles of RMSE versus times for various terminal time. Polynomial basis functions of order 19 were employed in the solution process.

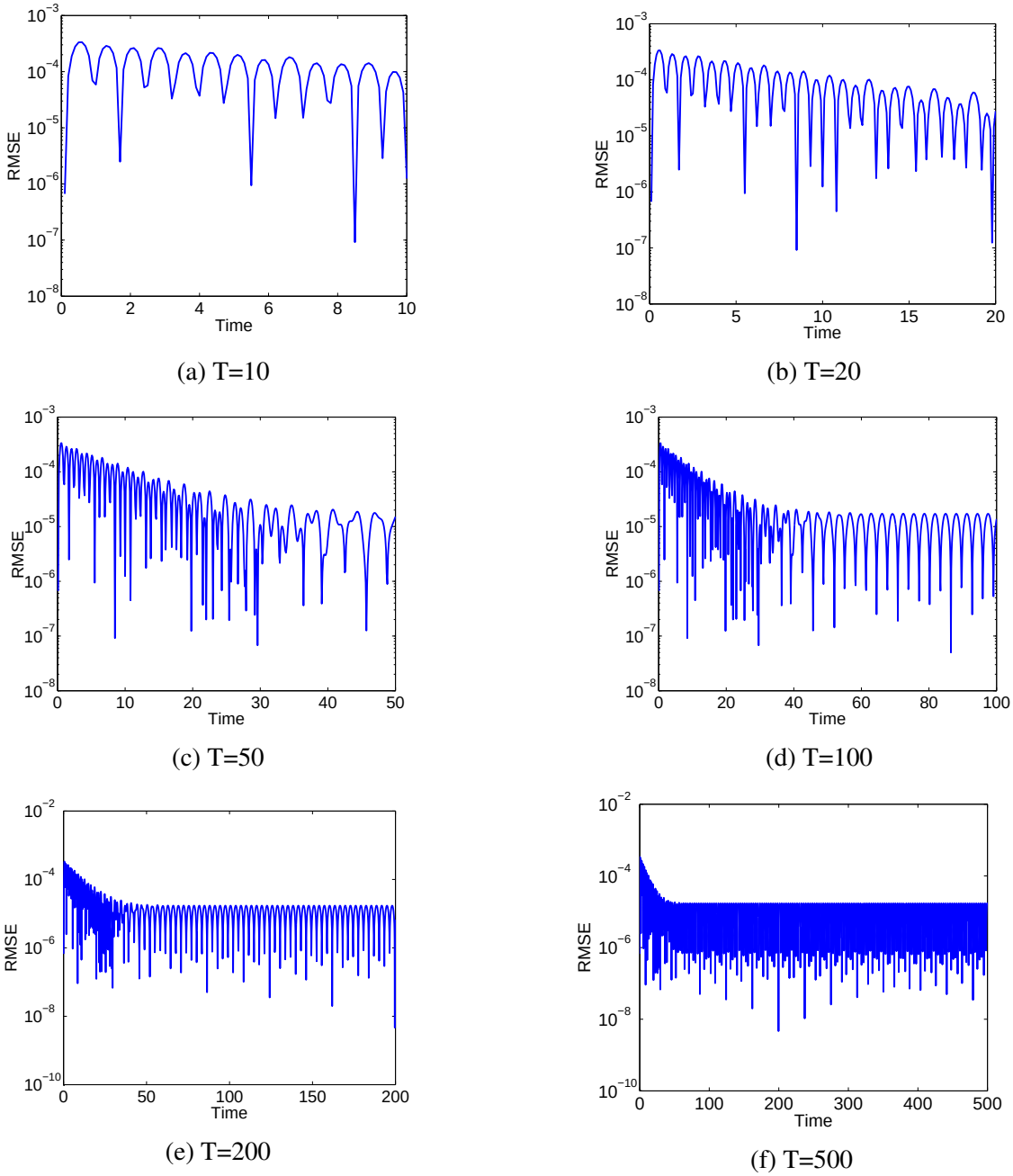
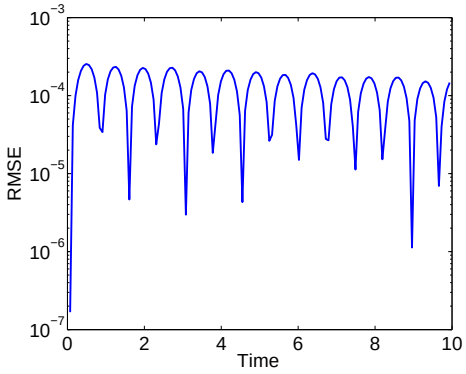


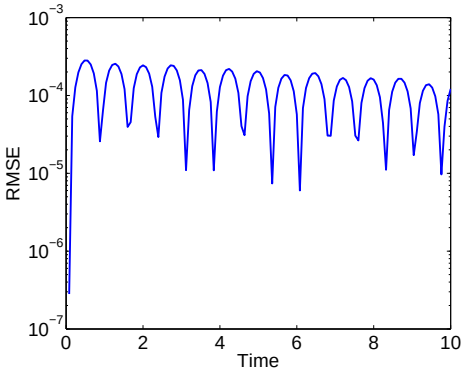
Figure 3.18: Example: 3.5.3: RMSE versus time.

The problem is also solved for various time steps with polynomial basis functions of order 24. Figure 3.19 shows the profiles of the RMSE versus time for various time steps. We observed that the higher order polynomial basis functions were necessary to obtain

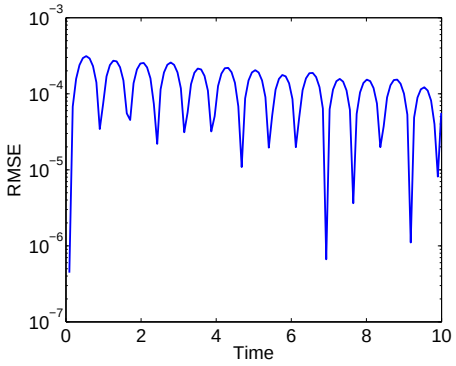
acceptable results for smaller time steps.



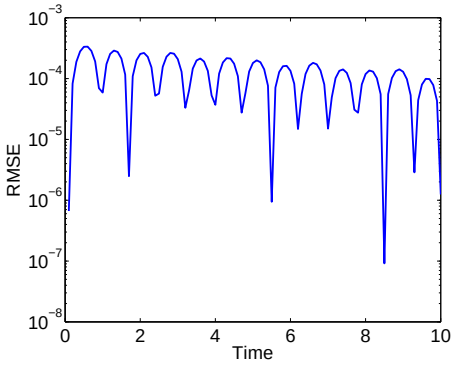
(a) dt=0.07



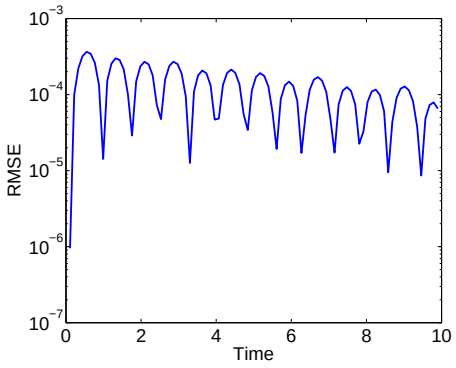
(b) dt=0.08



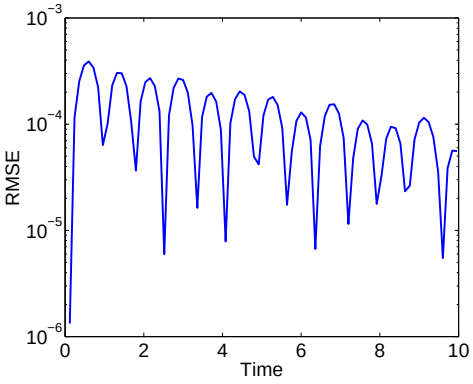
(c) dt=0.09



(d) dt=0.1



(e) dt=0.11



(f) dt=0.12

Figure 3.19: Example: 3.5.3: RMSE versus time.

3.6 3D Boundary Value Problems

In this section, we are solving some 3-D boundary value problems using the MPS with polynomial basis functions. The determination of the particular solutions for the differential operator is very similar to the 2-D case. Therefore, the way of solving three-dimensional problems is as easy as the 2-D problems. We have presented two numerical examples to show the efficacy of the proposed scheme in 3D problems.

Example 3.6.1. Let us consider the following 3-D Poisson problem.

$$\begin{aligned} \Delta u(x,y,z) &= f(x,y,z), \ (x,y,z) \in \Omega, \\ u(x,y,z) &= g(x,y,z), \ (x,y,z) \in \partial\Omega, \end{aligned}$$

where $f(x,y,z)$ and $g(x,y,z)$ are given based on the following analytical solution

$$u(x,y) = \cos(3x)\cos(3y)\cos(3z), \ (x,y,z) \in \overline{\Omega}.$$

We consider the computational domain as the Stanford Bunny whose boundary data points are available at the website of the Stanford Computer Graphics Laboratory [1]. Since the scale of the original Bunny is too small, we magnify the size of the Bunny by 10 as the computational domain in our numerical simulation. The number of interior points, boundary points, and test points are 2345, 453, 690 respectively. The profile of the computational domain is given in Figure 3.20. We solve this problem using the MPS based on radial basis

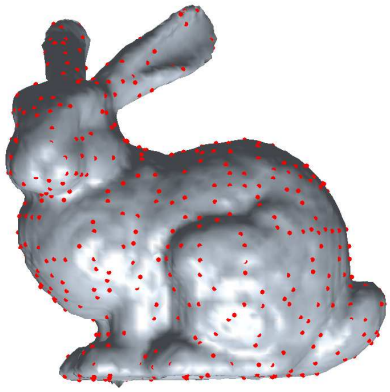
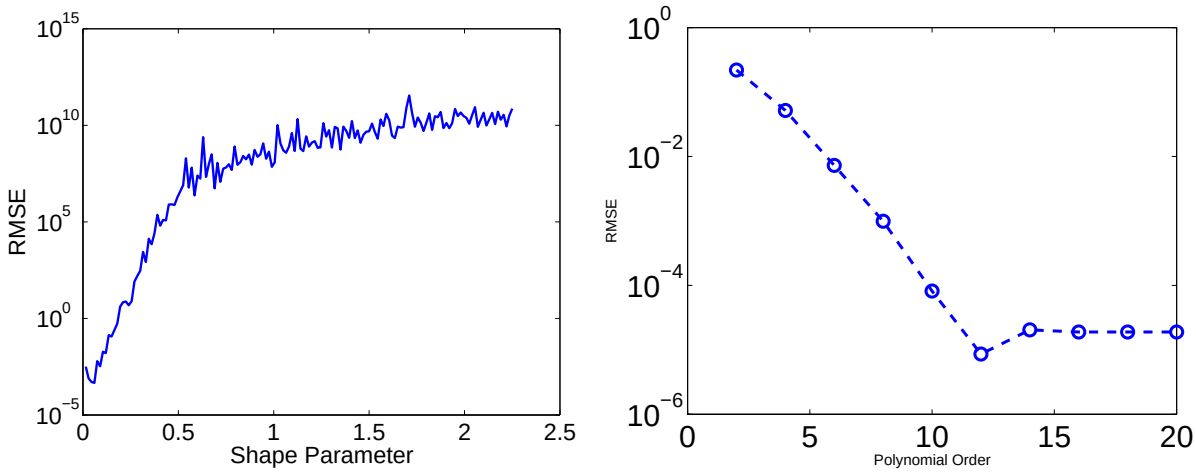


Figure 3.20: Example 3.6.1: The profile of the Stanford Bunny with boundary points.

function MQ and based on polynomial basis functions. In Figure 3.21a, the RMSE of the solution is fluctuating heavily with the shape parameter. But, in Figure 3.21b, we can see

that the results are stable when the order of polynomial basis functions increases. The results obtained from the MPS with polynomial basis functions are far better and stable than those of the MPS with radial basis function MQ.



(a) RBF (b) Polynomial

Figure 3.21: Example 3.6.1: (a) RMSE versus the shape parameter of the MQ. (b) RMSE versus the polynomial order.

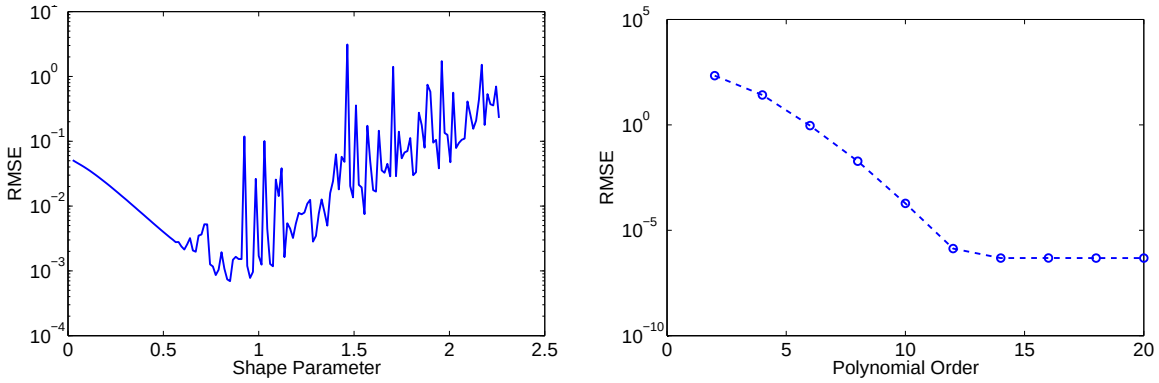
Example 3.6.2. Let us consider the following 3-D Helmholtz problem.

$$\begin{aligned}(\Delta+\lambda^2)u(x,y,z) &= f(x,y,z),\ (x,y,z)\in\Omega,\\ u(x,y,z) &= g(x,y,z),\ (x,y,z)\in\partial\Omega,\end{aligned}$$

where $f(x,y,z)$ and $g(x,y,z)$ are given based on the following analytical solution

$$u(x,y) = e^{2x+2y+2z},\ (x,y,z)\in\overline{\Omega}.$$

The domain considered here is a unit cube. The number of interior points, boundary points, and test points are 1728, 488, 729 respectively. The numerical accuracy of the problem is compared with the MPS using MQ and polynomial basis functions for $\lambda = 1$. Figure 3.22 shows the comparison of the numerical results obtained from the MPS with radial basis function MQ and with polynomial basis functions of different orders. The figure clearly shows that the MPS with polynomial basis functions can perform better than the MPS with radial basis function MQ. As a further experiment, for the same domain with the given collocation points, we have observed the numerical accuracy of our proposed method for various values of λ with polynomial basis functions of order 12 in Table 3.9. We observed that the accuracy improves with the increase of the wavelength λ.



(a) RBF (b) Polynomial

Figure 3.22: Example 3.6.2: (a) RMSE versus the shape parameter of the MQ. (b) RMSE versus the polynomial order.

Table 3.9: Example 3.6.2: RMSE and MAE for different values of wavelengths.

λ	RMSE	MAE
1	1.35e-06	5.96e-06
10	1.75e-04	4.81e-04
50	8.24e-07	5.81e-06
100	6.65e-07	5.85e-06
500	6.27e-07	5.91e-06
1000	6.13e-07	5.54e-06

Chapter 4

LOCALIZED METHOD OF PARTICULAR SOLUTIONS BASED ON POLYNOMIAL BASIS FUNCTIONS

4.1 Introduction

The global approach of the method of particular solutions (MPS) [14, 15, 79] is capable of solving a large class of partial differential equations in science and engineering. This method has employed radial basis functions (RBFs) for the construction of particular solutions for the differential equations. Challenges faced when using this method include the indeterminate shape parameter of the radial basis functions, the difficulty of finding the particular solutions for the differential operator, and a dense and ill-conditioned interpolation matrix. In the previous chapter, we discussed the global-type method of particular solutions using polynomial basis functions. The replacement of radial basis function with polynomial basis functions has made the method more accurate and simpler. The method of particular solutions based on polynomial basis functions is still not suitable for the large-scale problem because the matrix formed in the collocation process becomes highly ill-conditioned. There are many mesh-based localized numerical methods such as the finite element method [50] and the finite difference method [49] to solve large-scale problems. These methods are difficult to implement for problems defined on higher dimensional domains with irregular geometries. Therefore, localized meshless methods [25, 45, 68, 70, 78, 83] are preferred over mesh-based methods because of their ability to solve problems defined on regular and irregular domains in higher dimensions. These localized meshless methods have utilized radial basis functions in their solutions process.

Most of the radial basis functions contain shape parameters. Finding the best shape parameter of the RBF has been an open research problem [29, 36, 65, 76, 80]. To avoid the difficulty of determining a good shape parameter of the RBFs, we propose the localized method of particular solutions based on polynomial basis functions. The main attractions of the proposed approach are:

- Closed-form particular solutions for the general partial differential equations with constant coefficients based on polynomial basis functions are available.
- The shape parameter is avoided.

Replacement of the radial basis function by the polynomial basis functions has made the method simpler and easy to implement with very nice numerical results.

4.2 Methodology

Let us consider the boundary value problem (2.14) – (2.15). Let $\{\mathbf{x}_i\}_{i=1}^{N_i}$ be the interior points in Ω and $\{\mathbf{x}_i\}_{i=N_i+1}^{N_i+N_b}$ be the boundary points on Ω and $N = N_i + N_b$. For each $\mathbf{x}_i$, we choose n nearest points $\mathbf{x}_k^{[i]},\ k = 1,2,...,n$, to form a local domain Ω_i such that $\Omega_i \cap \Omega_j \neq \phi$ for some $j \neq i$ and $[\mathbf{x}_i]_{i=1}^{N} = \bigcup_i \Omega_i$. In the LMPS, we approximate the solution at each point $\mathbf{x}_j^{[i]} \in \Omega_i,\ j = 1,2,...,n$, by

$$u(\mathbf{x}_j^{[i]}) \simeq \hat{u}(\mathbf{x}_j^{[i]}) = \sum_{l=0}^{s} \sum_{m=0}^{l} a_{lm}^{[i]} U_{lm}(\mathbf{x}_j^{[i]}),\ j = 1,2,...,n, \tag{4.1}$$

where

$$\mathcal{L}U_{lm}(\mathbf{x}_j^{[i]}) = x^{l-m}y^m,\ 0 \leq m \leq l, 0 \leq l \leq s. \tag{4.2}$$

The system of equations in (4.1) can be written in matrix form as follows:

$$\begin{bmatrix} \hat{u}(\mathbf{x}_1^{[i]}) \\ \hat{u}(\mathbf{x}_2^{[i]}) \\ \vdots \\ \hat{u}(\mathbf{x}_n^{[i]}) \end{bmatrix} = \begin{bmatrix} U_{00}(\mathbf{x}_1^{[i]}) & U_{10}(\mathbf{x}_1^{[i]}) & U_{11}(\mathbf{x}_1^{[i]}) & \dots & U_{ss}(\mathbf{x}_1^{[i]}) \\ U_{00}(\mathbf{x}_2^{[i]}) & U_{10}(\mathbf{x}_2^{[i]}) & U_{11}(\mathbf{x}_2^{[i]}) & \dots & U_{ss}(\mathbf{x}_2^{[i]}) \\ \vdots & & & & \\ U_{00}(\mathbf{x}_n^{[i]}) & U_{10}(\mathbf{x}_n^{[i]}) & U_{11}(\mathbf{x}_n^{[i]}) & \dots & U_{ss}(\mathbf{x}_n^{[i]}) \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ \vdots \\ a_{ss} \end{bmatrix} \tag{4.3}$$

Let $\Phi_{n,s}$ be the matrix in equation (4.3). From (4.3), we have

$$\mathbf{a}_{lm}^{[i]} = \Phi_{n,s}^{-1} \hat{\mathbf{u}}^{[i]} \tag{4.4}$$

where

$$\hat{\mathbf{u}}^{[i]} = \left[\hat{u}(\mathbf{x}_1^{[i]}), \hat{u}(\mathbf{x}_2^{[i]}), \dots, \hat{u}(\mathbf{x}_n^{[i]})\right]^T, \mathbf{a}_{lm}^{[i]} = \left[a_{00}, a_{10}, \dots, a_{ss}\right]^T.$$

Then, applying (4.4) to (4.1), we get

$$\hat{u}(\mathbf{x}_j^{[i]}) = U_{lm}^{[i]} \mathbf{a}_{lm}^{[i]} = U_{lm}^{[i]} \Phi_{n,s}^{-1} \hat{\mathbf{u}}^{[i]} = \Psi^{[i]} \hat{\mathbf{u}}^{[i]} \tag{4.5}$$

where

$$U_{lm}^{[i]} = [U_{00}(\mathbf{x}_j^{[i]}), U_{10}(\mathbf{x}_j^{[i]}), ..., U_{ss}(\mathbf{x}_j^{[i]})]$$

and

$$\Psi^{[i]} = U_{lm}^{[i]} \Phi_{n,s}^{-1} = [\psi_1, \psi_2, ..., \psi_n].$$

Now, we want to express equation (4.5) in the global sense. Let $\hat{\mathbf{u}} = [\hat{u}(\mathbf{x}_1), \hat{u}(\mathbf{x}_2), \ldots, \hat{u}(\mathbf{x}_N)]^T$. The global form of equation (4.5) is

$$\hat{u}(\mathbf{x}_j^{[i]}) = \Psi\hat{\mathbf{u}}, \tag{4.6}$$

where the $1 \times N$ vector Ψ is obtained by padding $N-n$ zeros in $\Psi^{[i]}$ based on the mapping between $\hat{\mathbf{u}}^{[i]}$ and $\hat{\mathbf{u}}$. The above scheme can be used to solve the problem (2.14)-(2.15). Applying (4.1) to (2.14), we have

$$\begin{aligned} f(\mathbf{x}_j^{[i]}) &= \mathcal{L}\hat{u}(\mathbf{x}_j^{[i]}) \\ &= \sum_{l=0}^{s}\sum_{m=0}^{l} a_{lm}^{[i]}\mathcal{L}U_{lm}(\mathbf{x}_j^{[i]}) \\ &= \mathcal{L}U_{lm}^{[i]}\mathbf{a}_{lm}^{[i]} \\ &= \mathcal{L}U_{lm}^{[i]}\Phi_{n,s}^{-1}\hat{\mathbf{u}}^{[i]} \\ &= \Lambda^{[i]}\hat{\mathbf{u}}^{[i]} \\ &= \Lambda(\mathbf{x}_j^{[i]})\hat{\mathbf{u}} \end{aligned}$$

where

$$\Lambda^{[i]} = \mathcal{L}U_{lm}^{[i]}\Phi_{n,s}^{-1},$$

and $\Lambda(\mathbf{x}_j^{[i]})$ is obtained by padding $N-n$ zeros in $\Lambda^{[i]}$. Thus,

$$f(\mathbf{x}_j^{[i]}) = \Lambda(\mathbf{x}_j^{[i]})\hat{\mathbf{u}}, \;\; 1 \le i \le N_i. \tag{4.7}$$

Similarly, applying (4.1) to (2.15), we get

$$\begin{aligned} g(\mathbf{x}_j^{[i]}) &= \mathcal{B}\hat{u}(\mathbf{x}_j^{[i]}) \\ &= \sum_{l=0}^{s}\sum_{m=0}^{l} a_{lm}^{[i]}\mathcal{B}U_{lm}(\mathbf{x}_j^{[i]}) \\ &= \mathcal{B}U_{lm}^{[i]}\mathbf{a}_{lm}^{[i]} \\ &= \mathcal{B}U_{lm}^{[i]}\Phi_{n,s}^{-1}\hat{\mathbf{u}}^{[i]} \\ &= \Upsilon^{[i]}\hat{\mathbf{u}}^{[i]} \\ &= \Upsilon(\mathbf{x}_j^{[i]})\hat{\mathbf{u}} \end{aligned}$$

where

$$\Upsilon^{[i]} = \mathcal{B}U_{lm}^{[i]}\Phi_{n,s}^{-1},$$

and $\Upsilon(\mathbf{x}_j^{[i]})$ is obtained by padding $N-n$ zeros in $\Upsilon^{[i]}$. Thus,

$$g(\mathbf{x}_j^{[i]}) = \Upsilon(\mathbf{x}_j^{[i]})\hat{\mathbf{u}}, \;\; N_i + 1 \le i \le N. \tag{4.8}$$

Finally, equations (4.7) and (4.8) form a sparse system of equations. The approximate solutions at each point can be obtained by solving equations (4.7) and (4.8).

4.3 Numerical Results

For the numerical implementation of the local approach of the MPS, we have adopted the kd-tree search algorithm [71] to find the nearest neighbor points of each collocation point. The kd-tree or k dimensional tree is a data structure used to organize the points in k dimensional space. Regular and irregular domains are considered in 2D and 3D spaces.

The resultant matrix sometimes becomes singular for the uniform set of collocation points. Thus, the collocation points and test points are chosen randomly throughout the problem domain.

Throughout all numerical experiments, the size of the local domain is same as the number of polynomial basis functions applied in the method.

In the local approach of the MPS, the size of the local domain is small. Thus, we do not need high order polynomial basis functions to approximate the solution. As a result, the interpolation matrix is small and well-conditioned. Therefore, multiple scale technique is not required to reduce the condition number of the matrix.

Example 4.3.1. Let us consider the following Helmholtz problem.

$$\begin{aligned} (\Delta+\lambda^2)u(x,y) &= f(x,y),\ (x,y)\in\Omega, \\ u(x,y) &= g(x,y),\ (x,y)\in\partial\Omega, \end{aligned}$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y) = e^{3x+4y},\ (x,y)\in\overline{\Omega}.$$

In this example, we consider a unit square domain with randomly chosen collocation points. We have observed the numerical accuracy of the method with different sets of collocation points with various orders of polynomial basis functions. The results in Table 4.1 show that the RMSE improves slightly with an increase in the collocation points. But the accuracy improves rapidly with an increase in the order of the basis functions. As in the global approach with the MPS, the numerical accuracy heavily depends upon the order of the polynomial basis functions.

Furthermore, the RMSE and MAE of the solutions with different wavelengths have been presented in Table 4.2. We have taken 20000 randomly chosen interior points and 200 boundary points as our collocation points in the unit square domain. Polynomial basis

Table 4.1: Example 4.3.1: The RMSE for various sets of interior and boundary points with different polynomial orders.

(n_i, n_b)	RMSE	Order	Time
$(20000, 200)$	1.427e-02	3	1.79
	4.931e-04	4	3.16
	2.362e-05	5	6.21
$(40000, 400)$	4.612e-03	3	4.17
	2.118e-04	4	6.91
	2.327e-06	5	13.83
$(60000, 600)$	3.999e-03	3	6.98
	2.526e-04	4	11.08
	6.688e-07	5	20.66

Table 4.2: Example 4.3.1: RMSE and MAE for different wavelength.

λ	RMSE	MAE
1	2.362e-05	4.267e-04
10	7.314e-04	1.846e-03
50	8.866e-05	3.355e-04
100	7.752e-06	4.214e-05
500	4.806e-08	2.432e-06
1000	7.616e-08	9.283e-06

functions of order 5 have been applied in the solution process. It is clear to see that the results are getting better for higher wavelengths.

Example 4.3.2. Let us consider a partial differential equation containing convection terms.

$$\begin{aligned} \Delta u(x,y) + 2\frac{\partial^2}{\partial x \partial y} u(x,y) - 4\frac{\partial}{\partial y} u(x,y) + 8u(x,y) &= f(x,y),\ (x,y) \in \Omega, \\ u(x,y) &= g(x,y),\ (x,y) \in \partial\Omega, \end{aligned}$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y) = \cos(3x) - \cos(3y),\ (x,y) \in \overline{\Omega}.$$

In this example, we solve the above problem defined in various domains. Table 4.3 lists the RMSE and Average Abs Error for different computational domains. Basis functions of order 5 are employed for all the computations. The results show that our proposed approach

Table 4.3: Example 4.3.2: RMSE and average absolute error for different computational domains.

Domains	n_i	n_b	RMSE	Ave Abs Err
Amoeba	18614	301	2.765e-05	1.450e-05
Cassini	18738	301	6.276e-06	3.534e-06
Gear-Shaped(5 Gear)	19530	301	1.380e-04	4.712e-05
Star-Shaped	18746	301	1.198e-05	7.339e-06
L-shaped	19360	200	3.040e-07	2.089e-07

can solve large-scale problems very effectively for complicated domains. The accuracy is consistent with the complexity of the problem domains.

Example 4.3.3. Let us consider the following Poisson's problem.

$$\begin{aligned} \Delta u(x,y) &= f(x,y),\ (x,y) \in \Omega, \\ u(x,y) &= g(x,y),\ (x,y) \in \partial\Omega, \end{aligned}$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y) = \frac{1.25 + \cos(5.4y + 2.7)}{6(1 + (3x + 0.5)^2)}.$$

We choose Ω to be the Star-Shape domain as shown in Figure 3.1f. Table 4.4 shows the numerical accuracy of our approach with various set of collocation points. The method produces nice results for various densities of the collocation points.

Table 4.4: Example 4.3.3: The RMSE and average absolute error for different number of interior and boundary points with polynomial basis of order 6.

(n_i, n_b)	RMSE	Ave Abs Err
$(1165, 300)$	$4.63e-03$	$1.91e-03$
$(4673, 300)$	$5.64e-05$	$2.84e-05$
$(10465, 300)$	$1.52e-04$	$2.92e-05$
$(18629, 300)$	$2.79e-05$	$1.47e-05$
$(29063, 300)$	$6.37e-05$	$3.37e-05$
$(41905, 300)$	$4.53e-05$	$1.81e-05$
$(57093, 300)$	$1.32e-05$	$5.16e-06$
$(74553, 300)$	$4.59e-04$	$1.28e-04$
$(116501, 300)$	$2.68e-04$	$6.11e-05$

Example 4.3.4. Let us consider a problem with variable coefficients

$$\begin{aligned} \Delta u(x,y) + y\cos(y)u_x(x,y) + \sinh(x)u_y(x,y) + 10xyu(x,y) &= f(x,y), \ (x,y) \in \Omega, \\ u(x,y) &= g(x,y), \ (x,y) \in \partial\Omega, \\ \frac{\partial u(x,y)}{\partial \mathbf{n}} &= h(x,y), \ (x,y) \in \partial\Omega, \end{aligned}$$

where $\mathbf{n}$ is the unit outward normal vector, $f(x,y)$, $g(x,y)$, and $h(x,y)$ are given based on the following analytical solution

$$u(x,y) = \sin(\pi x)\cosh(y) - \cos(\pi x)\sinh(y).$$

In this example, we have solved the problem using polynomial basis functions of different orders. The domain considered in this problem is a unit square. The number of interior points and boundary points are 40,000 and 236 respectively. Table 4.5 shows the numerical accuracy of our approach for different orders of basis functions. We can see that the numerical accuracy is better for higher order basis functions.

Table 4.5: Example 4.3.4: The RMSE for different polynomial orders.

Domain	RMSE	Order	Time
Unit Square	1.88e-05	3	6.61
	1.99e-07	4	10.88
	2.37e-09	5	19.33
	5.28e-10	6	30.65

Example 4.3.5. In this example, we choose the following modified Helmholtz problem [83]:

$$\begin{aligned} (\Delta - 100)u(x,y) &= f(x,y), \ (x,y) \in \Omega, \\ u(x,y) &= g(x,y), \ (x,y) \in \partial\Omega, \end{aligned}$$

where $f(x,y)$ and $g(x,y)$ are given based on the following analytical solution

$$u(x,y) = \sin\frac{\pi x}{6}\sin\frac{7\pi x}{4}\sin\frac{3\pi y}{4}\sin\frac{5\pi y}{4}, \ (x,y) \in \overline{\Omega}.$$

The domain is the star-shaped domain as shown in Figure 3.1f. The collocation points are randomly chosen. We have compared the numerical results obtained from the LMPS using radial basis function MQ and the LMPS using polynomial basis functions. In Table 4.6,

the comparison of RMSE obtained from the LMPS with RBF MQ and polynomial basis functions is presented. The result shows that the RMSE obtained from the LMPS with polynomial basis functions is slightly better than that from the LMPS with RBF MQ. It is well known that the LMPS with RBF performs better for uniform sets of collocation points. But, for the non-uniform set of collocation points, we see that the LMPS with polynomial basis functions gives better results than the LMPS with RBF MQ.

Table 4.6: Example 4.3.5: Comparison of RMSE using MPS with RBF MQ and polynomial basis functions for various set of collocation points.

	RBF MQ	Polynomial Basis Functions	
(n_i, n_b)	RMSE	RMSE	order
(384,80)	3.35e-03	1.12e-01	3
(6600,300)	1.17e-02	6.03e-03	3
(20161,900)	5.70e-02	2.03e-03	3

Furthermore, in Table 4.7, we present the RMSE for different sizes of local domain. We have taken 6600 randomly chosen interior points and 300 boundary points in the same domain. We see that the LMPS using polynomial basis functions gives better results than the LMPS with RBF MQ. We observed improvement of the accuracy in our approach with an increase of the polynomial order.

Table 4.7: Example 4.3.5: Comparison of RMSE using MPS with RBF MQ and polynomial basis functions for various sizes of local domains.

	RBF MQ	Polynomial Basis Functions	
n	RMSE	RMSE	order
10	1.17e-02	6.03e-03	3
15	1.32e-01	4.08e-04	4
21	1.64e-02	1.01e-04	5
28	2.01e-02	1.34e-05	6

Example 4.3.6. In this example, we solve the Cauchy-Navier equation of elasticity [33] using the local scheme. We have solved the problem (3.31) – (3.32) for a Dirichlet boundary condition with $\mu = 1$ and $\nu = 0.3$. The exact solutions considered for this problem are

$$u_1(x,y) = e^{x+y}, \ (x,y) \in \bar{\Omega}, \tag{4.9}$$

$$u_2(x,y) = \sin(x+y), \ (x,y) \in \bar{\Omega}. \tag{4.10}$$

We have tested the accuracy of the method on various types of computational domains. Since the domains we considered are of various sizes, we have computed relative errors to observe and compare the numerical accuracies of the method among the domains. Table 4.8 shows that the local scheme can successfully solve the problem defined in many different irregular domains. We noticed that the numerical accuracy of the method depends upon the geometrical complexity of the computational domain.

Table 4.8: Example 4.3.6: Relative error for different computational domains.

Domains	n_i	n_b	Rel Err(u_1)	Rel Err(u_2)	Polynomial Order
Amoeba	87873	300	7.199e-08	5.720e-07	5
Cassini	88377	300	1.630e-08	2.823e-008	5
Gear-Shaped	89442	300	7.092e-07	3.064e-06	5
Star-Shaped	84972	300	5.374e-08	1.380e-07	5

4.4 3D Boundary Value Problems

Example 4.4.1. Let us consider the following 3-D Poisson problem.

$$\begin{aligned} \Delta u(x,y,z) &= f(x,y,z),\ (x,y,z) \in \Omega, \\ u(x,y,z) &= g(x,y,z),\ (x,y,z) \in \partial\Omega, \end{aligned}$$

where $f(x,y,z)$ and $g(x,y,z)$ are given based on the following analytical solution

$$u(x,y) = \cos(x)\cos(2y)\cos(3z),\ (x,y,z) \in \overline{\Omega}.$$

A Stanford Bunny is the domain with $n_i = 2345$ and $n_b = 453$. As in example 3.6.1, we magnify the size of the Bunny by 10 as the computational domain in our numerical simulation. Table 4.9 shows the numerical accuracy of the problem with different orders of polynomial basis functions. The accuracy of the method is improving with an increase of the order of the basis functions. The number of neighboring points in the local domain for orders 3, 4, 5, and 6 are 20, 35, 56, and 84 respectively.

Example 4.4.2. Let us consider the following 3-D Helmholtz problem.

$$\begin{aligned} (\Delta + \lambda^2) u(x,y,z) &= f(x,y,z),\ (x,y,z) \in \Omega, \\ u(x,y,z) &= g(x,y,z),\ (x,y,z) \in \partial\Omega, \end{aligned}$$

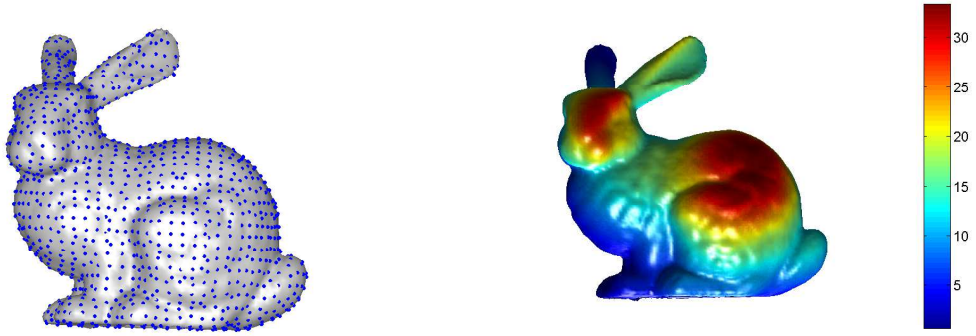
Table 4.9: Example 4.4.1: RMSE and average absolute error for various polynomial orders.

Order	RMSE	MAE
3	3.10e-02	2.14e-01
4	1.07e-02	6.91e-02
5	3.08e-04	2.10e-03
6	2.99e-05	5.108e-04

where $f(x,y,z)$ and $g(x,y,z)$ are given based on the following analytical solution

$$u(x,y) = e^{x+2y+3z},\ (x,y,z) \in \overline{\Omega}.$$

A Stanford Bunny is considered to be our domain. As in example 4.4.1, we magnify the size of the Bunny by 10 as the computational domain in our numerical simulation. The profile of the Stanford Bunny with the distribution of the boundary points is shown in Figure 4.1a. Since the maximum of the exact solution occurs on the boundary, we also plot the profile of the exact solution on the surface of the Bunny as shown in Figure 4.1b. The main challenge of this problem is due to its complicated three-dimensional geometry.



(a) RBF (b) Polynomial

Figure 4.1: Example 4.4.2: Profiles of (a) Stanford Bunny and its boundary points (b) exact solution.

In the numerical implementation, we first select $N_i = 5000$ and $N_b = 453$, with polynomial order 3. Table 4.10 shows the numerical accuracy for various values of wavelength λ. As we can see, the result for low wavelength is poor. To improve the accuracy, we can either increase the order of the polynomials or the number of collocation points.

Table 4.10: Example 4.4.2: RMSE and MAE for different value of wavelength λ using order = 3, N_i =5000, N_b = 453.

λ	RMSE	MAE
1	9.130e-02	1.384e+00
10	3.291e-01	2.178e+00
100	8.110e-03	2.071e-01
500	1.279e-04	4.075e-03
1000	4.654e-04	3.278e-02

In Table 4.11, we observe that the increase of the number of boundary points to 8171 and of the number of interior points up to 20000 using the same order of polynomial does not improve the accuracy as we have expected. On the other hand, as shown in Table 4.12, the increase of the polynomial basis order has significantly improved the accuracy. Note that the number of neighboring points in the local domain for orders 3, 4, 5, and 6 are 20, 35, 56, and 84 respectively. Hence, in the 3D case, we would not recommend to use a polynomial basis beyond order 5. Otherwise, the computational cost becomes too high. In Table 4.13, we show the results of using a large number of boundary and interior points and polynomial order 5. In these tables, we have demonstrated that the proposed localized method can handle a large number of collocation points in a complicated irregular domain in 3D with good accuracy.

Table 4.11: Example 4.4.2: RMSE and MAE for different number of interior points using order = 3, N_b = 8171.

N_i	RMSE	MAE
5000	6.919e-01	1.972e+01
10000	8.474e-02	1.315e+00
15000	6.596e-02	7.773e-01
20000	3.224e-02	4.590e-01

4.5 Near-Singular Problems

In this section, we test our algorithm on near-singular problems.

Table 4.12: Example 4.4.2: RMSE and MAE for different number of polynomial orders using N_i = 5000, N_b = 453.

order	RMSE	MAE
3	9.130e-02	1.384e+01
4	5.534e-03	9.521e-02
5	3.386e-04	6.607e-03
6	2.004e-04	4.858e-03

Table 4.13: Example 4.4.2: RMSE and MAE for different number of polynomial orders using order =5, N_i = 20000, N_b = 8171.

λ	RMSE	MAE
1	2.927e-04	2.898e-02
10	2.259e-04	1.652e-02
100	7.572e-07	3.793e-04
1000	1.589e-07	2.107e-05

Example 4.5.1. Let us consider the following Poisson problem [85]

$$\begin{aligned} \Delta u(x,y) &= f(x,y), \ (x,y) \in \Omega, \\ u(x,y) &= g(x,y), \ (x,y) \in \partial\Omega, \end{aligned}$$

defined in a unit squared domain, where a is a constant and

$$f(x,y) = \frac{4a^2 - 3a\sqrt{x^2+y^2} + x^2 + y^2}{(\sqrt{x^2+y^2} - a)^3},$$
$$g(x,y) = \frac{x^2+y^2}{\sqrt{x^2+y^2} - a}.$$

The analytical solution for this problem is given by

$$u(x,y) = \frac{x^2+y^2}{\sqrt{x^2+y^2} - a}. \tag{4.11}$$

We can see that the forcing term $f(x,y)$ contains the singularity at (1,1) when $a = \sqrt{2}$. Therefore, it is difficult to approximate the solutions near the boundary when the values of a are sufficiently close to $\sqrt{2}$. The profile of the exact solution when $a = 1.45$ is given in Figure 4.2. For the numerical experiment, we have taken 25,281 interior points and 640 boundary points. The RMSE obtained from our approach is compared with that from LMPS

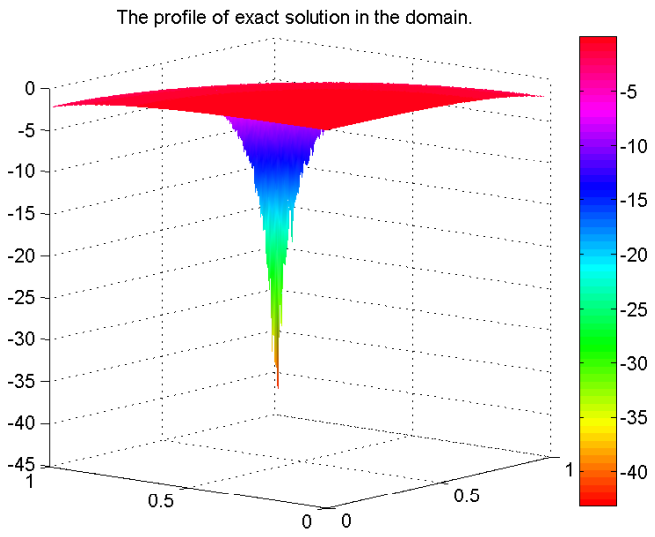


Figure 4.2: Example 4.5.1: The profile of analytical solution in the domain

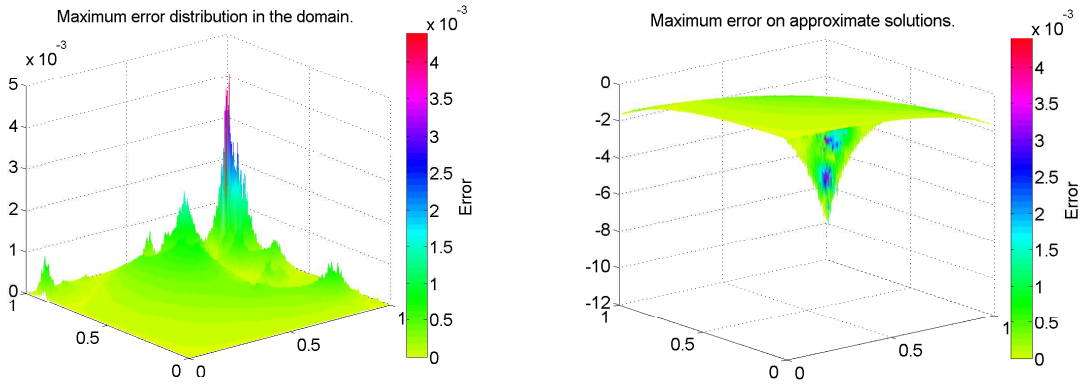
using RBF MQ [85] in Table 4.14. The size of the local domain in LMPS using RBF MQ was 5. We have solved this problem for different values of a near $\sqrt{2}$.

Table 4.14: Example 4.3.5: Comparison of RMSE using MPS with RBF MQ and polynomial basis functions for various values of a.

	RBF MQ	Polynomial Basis Functions	
a	RMSE	RMSE	order
1.43	5.64e-03	8.03e-02	3
1.44	2.31e-03	1.47e-02	3
1.45	1.27e-03	1.17e-02	3
1.46	8.12e-04	8.86e-03	3
1.47	5.69e-04	6.29e-03	3
1.48	4.24e-04	4.41e-03	3
1.49	3.29e-04	3.12e-03	3
1.50	2.65e-04	2.25e-03	3
1.60	4.04e-05	3.04e-04	3

In Table 4.14, we can see the effect of the singularity at one corner of the boundary. As a approaches $\sqrt{2}$, the numerical accuracy of the problem is getting worse. For $a = 1.6$, we have presented the error distribution on the surface of the exact solution and on the surface of the domain. Figure 4.3 shows the profiles of maximum error on the computational domain and on the surface of the approximate solution. In both of the figures, we can see that the

error is maximal near the singular point (1,1).



(a) Maximum Error on Domain

(b) Maximum Error on Approximate Solution

Figure 4.3: Example 4.5.1: (a) Maximum error distribution in the computational domain. (b) Maximum errors on approximate solutions.

Example 4.5.2. In this example, we solve a Helmholtz-type near singular problem [75]. Consider a modified Helmholtz problem

$$\begin{aligned}(\Delta - 10)u(x,y) &= f(x,y),\ (x,y) \in \Omega,\\ u(x,y) &= g(x,y),\ (x,y) \in \partial\Omega,\end{aligned}$$

defined in a domain $[-0.5,0.5] \times [-0.5,0.5]$, where the function $f(x)$ and $g(x)$ are given by

$$f(x,y) = \frac{4a^2 - 3ar + r^2}{(a-r)^3} - \frac{10r^2}{a-r},$$
$$g(x,y) = \frac{r^2}{a-r}$$

with

$$r = \sqrt{(x+0.5)^2 + (y+0.5)^2}. \tag{4.12}$$

The analytical solution to this problem is given by (4.11). The forcing term has a singularity at (0.5,0.5) when $a = \sqrt{2}$. The number of interior points and boundary points are 400 and 136 respectively.

The results in Table 4.15 follow a similar pattern as in Table 4.14. The numerical error deteriorates as a gets closer to $\sqrt{2}$. In Table 4.16, we investigate the numerical accuracy of

Table 4.15: Example 4.5.2: RMSE and MAE for various values of a.

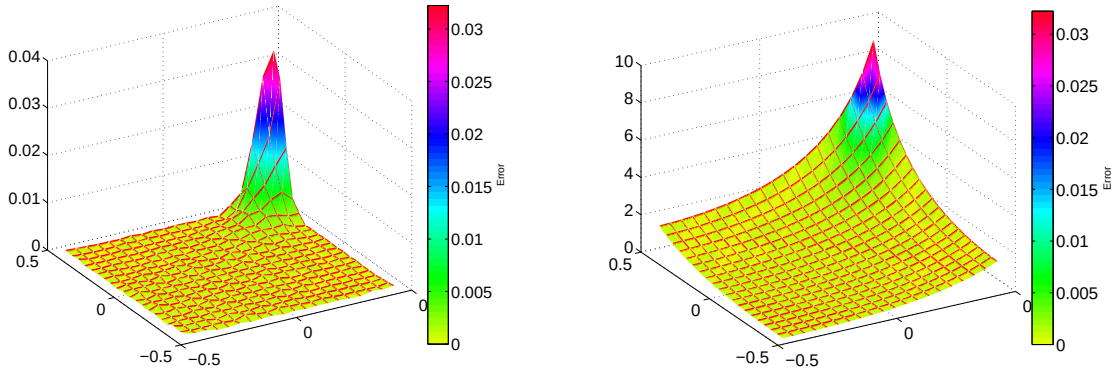
a	RMSE	MAE
1.45	4.15e-01	4.46e+00
1.5	2.12e-02	2.26e-01
1.6	6.80e-04	7.37e-03
1.7	7.54e-05	8.17e-04
1.8	1.77e-05	1.60e-04
1.9	6.67e-06	4.34e-05

the method with various orders of polynomial basis functions for $a = 1.6$. The accuracy improves with the increase of the order of basis functions. In the global approach of our algorithm, the accuracy remains stable after some order of the basis function. We can see the similar behavior of the polynomial order in the local approach. The improvements of the error for orders 4,5 and 6 are not remarkable.

Table 4.16: Example 4.5.2: RMSE and MAE for various Polynomial Order.

Order	RMSE	MAE
3	1.33e-02	4.58e-02
4	3.63e-04	1.96e-03
5	3.07e-04	4.61e-03
6	6.80e-04	7.37e-03

Figure 4.4 illustrates the profile of the maximum error of the solution in the domain and on the surface of the numerical solution for $a = 1.55$. As in Figure 4.4, the error seems maximal at the corner $(0.5, 0.5)$ of the boundary. As a final experiment in this problem, we have presented the RMSE and MAE for various sets of collocation points on the same domain in Table 4.17. We have chosen $a = 1.6$ and the order of the polynomial basis functions is 6.



(a) Maximum Error on Domain

(b) Maximum Error on Approximate Solution

Figure 4.4: Example 4.5.2: (a) Maximum error distribution in the computational domain. (b) Maximum errors on approximate solutions.

Table 4.17: Example 4.5.2: The RMSE and Maximum errors for different numbers of interior and boundary points with polynomial basis of order 6.

(n_i, n_b)	RMSE	MAE
$(400, 136)$	$6.80e-04$	$7.37e-03$
$(1089, 236)$	$1.24e-04$	$6.01e-04$
$(2025, 316)$	$8.47e-04$	$6.56e-03$
$(4225, 404)$	$3.58e-05$	$1.80e-04$
$(7225, 404)$	$3.22e-05$	$1.30e-04$
$(11025, 404)$	$7.45e-05$	$2.81e-04$
$(21025, 404)$	$7.34e-05$	$2.64e-04$
$(61504, 404)$	$2.39e-05$	$1.07e-04$

Chapter 5

THE MODIFIED MPS FOR FINDING CRITICAL DOMAINS FOR QUENCHING PROBLEMS

Yao *et al.* [84] has modified the traditional MPS by using polyharmonic splines coupled with polynomial basis functions. In this chapter, we have employed this modified method to solve some Poisson-type nonlinear problems and to compute critial domains for quenching problems.

5.1 Introduction

Nonlinear partial differential equation is one of the most active fields of recent research. Most real world problems, including gas dynamics, fluid mechanics, elasticity, relativity and much more, are modeled by nonlinear partial differential equations. In general, the exact solutions of many such problems are not available. Therefore, numerical approximation has to be done to find the solutions of the problems. There are many numerical schemes to solve nonlinear problems. Some of them are the finite element method (FEM) [50], the finite difference method (FDM) [49], the boundary element method (BEM) [41] and meshless methods [4, 15, 40]. The mesh-based methods like the FEM and the FDM require extensive work of mesh generation on the computational domain, requiring large computational resources to solve the problem. Furthermore, the generation of the mesh for an irregular domain is non-trivial. The radial basis function (RBF) collocation methods are meshless methods which are able to overcome these drawbacks. We have used the MPS [14, 15, 79], one of the RBF collocation methods, to solve some nonlinear Poisson-type problems defined on regular and irregular domains. In addition, the MPS is also applied to compute the size of the critical domains for the quenching problems [12]. The MPS is modified to make the method more efficient.

Consider the following initial boundary value problem

$$\Delta u - \frac{\partial u}{\partial t} = -f(t,u) \text{ in } \Omega, \tag{5.1}$$

$$u(\mathbf{x},\mathbf{0}) = u_0(\mathbf{x}) \text{ on } \bar{\mathbf{D}}, \tag{5.2}$$

$$u(\mathbf{x},\mathbf{t}) = 0 \text{ on } \mathbf{S}, \tag{5.3}$$

where $\mathbf{D} \in \mathbb{R}^m$ is a bounded convex domain, $\Omega = \mathbf{D} \times (0, \mathbf{T})$, $\mathbf{S} = \partial\mathbf{D} \times (0, \mathbf{T})$, and $\mathbf{T} \leq \infty$ such that

$$\lim_{u \to c^-} f(t, u) = \infty,$$

for some positive constant c. This kind of problem was first studied by Kawarada [42] in 1975. According to him, The solution u is said to quench if there exists a finite time $\mathbf{T}$ such that

$$\sup \left\{ |u_t(\mathbf{x}, t)| : \mathbf{x} \in \bar{\mathbf{D}} \right\} \to \infty \text{ as } t \to \mathbf{T}^-. \tag{5.4}$$

The time at which the quenching occurs is called the quenching time. When u is an increasing function of t, a necessary condition for (5.4) is

$$\max \left\{ u(\mathbf{x}, t) : \mathbf{x} \in \bar{\mathbf{D}} \right\} \to c^- \text{ as } t \to \mathbf{T}^-. \tag{5.5}$$

The point in Euclidian space where the solution u reaches c is called quenching point. For the one-dimensional case, Acker and Kawohl [2] proved that the origin is the only quenching point for the problem (5.1) – (5.3) when the domain is a ball with center at the origin. Also, Deng and Levine [24] extended the results from balls to a convex domain $\mathbf{D}$ with smooth boundary $\partial\mathbf{D}$. They showed that the quenching points are in a compact subset of $\mathbf{D}$. In 1994, Chan and Ke [12] studied the critical domains and developed a method to find the size of such domain for the problem (5.1) – (5.3) defined on a domain with piecewise smooth boundary. A critical domain $\mathbf{D}^*$ is a domain such that the solution exists for all domain $\mathbf{D}$ at all time when $\mathbf{D} \subset \mathbf{D}^*$ and the quenching always occurs at finite time $\mathbf{T}$ in $\mathbf{D} \supseteq \mathbf{D}^*$. With $u_0 \equiv 0$, Chan and Ke [12] proved that a unique critical domain exists for each shape of domain for the problem (5.1) – (5.3). This means each distinct shape of a domain has distinct critical domain size. The size of the domain is specified by area in 2D and by volume for 3D or higher. In 2007, Tian [74] applied a numerical method using Delta-shaped basis function to compute the critical domains for quenching problems and compared the results with the results already established with some other computational methods.

Let $\mathbf{D} \subseteq \mathbb{R}^m$ and $\mathbf{D}_1 \subseteq \mathbf{R}^m$. They will have the same shape if there exists $\mathbf{x_0} \in \mathbf{D} \cap \mathbf{D_1}$ and a positive constant λ such that

$$\mathbf{D}_1 = \{\mathbf{y} : \mathbf{y} = \mathbf{x_0} + \lambda(\mathbf{x} - \mathbf{x_0})\} \quad \text{for} \quad \mathbf{x} \in \mathbf{D}. \tag{5.6}$$

If the problem (5.1) – (5.3) is defined in the domain $\mathbf{D}_1$ with known shape and x_0 is at the origin, we can transform this problem into a problem defined in a different sized domain $\mathbf{D}$ having the same shape in the following way:

$$\Delta u_\lambda - \frac{\partial u_\lambda}{\partial t} = -\lambda^2 f(\lambda^2 t, u_\lambda) \text{ in } \Omega, \tag{5.7}$$

$$u_\lambda = 0 \text{ on } \bar{\mathbf{D}} \cup \mathbf{S}, \tag{5.8}$$

where (cf. Chan and Ke [12]),

$$\lambda = \left(\frac{\text{size of } \mathbf{D}_1}{\text{size of } \mathbf{D}} \right)^{1/m}. \tag{5.9}$$

From (5.9), we see that the size of the critical domain is determined by λ. The modified MPS has been used to compute λ.

In Section 5.2, a brief review for the MPS for solving nonlinear Poisson-type problems with Dirichlet's boundary condition is given. In Section 5.3, an algorithm for the computation of quenching problems is given. In Section 5.4, some numerical examples for solving nonlinear Poisson-type problems are given, and critical domains for quenching problems are computed and compared to the results already established by some other numerical methods.

5.2 The Method of Particular Solutions (MPS)

In this section, we give a brief review of the MPS [14, 15, 79] using polyharmonic splines. For simplicity, let us consider the following Poisson problem in 2D

$$\Delta u(x,y) = f(x,y), (x,y) \in \Omega, \tag{5.10}$$

$$u(x,y) = g(x,y), (x,y) \in \partial\Omega, \tag{5.11}$$

where $\Omega \subseteq \mathbb{R}^2$ is a bounded and closed domain with boundary Ω, f and g are known functions.

Let ϕ be a radial basis function (RBF) and let $\{p_l\}_{l=1}^{w}$ be a basis of $\mathbf{P}_m$, the set of two dimensional polynomials of degree $\leq m$ with $w = (m+1)(m+2)/2$. Let $\{(x_i,y_i)\}_{i=1}^{n}$ be a set of pairwise distinct interpolation points with $\{(x_i,y_i)\}_{i=1}^{n_i} \subseteq \Omega$ and $\{(x_i,y_i)\}_{i=n_i+1}^{n} \subseteq \partial\Omega$ such that $n = n_i + n_b$.

The MPS has been widely used in the context of RBFs with shape parameter such as MQ, inverse MQ, Gaussian. Recently, Yao *et al.* [84] have extended the MPS using polyharmonic splines with augmented polynomials as the basis function as follows

$$u(x,y) \simeq \hat{u}(x,y) = \sum_{j=1}^{n} a_i \Phi(r) + \sum_{l=1}^{w} a_{n+l} p_l(x,y), \ \ (x,y) \in \Omega, \tag{5.12}$$

where $r = \|(x,y) - (x_j,y_j)\|$ and

$$\Delta\Phi(r) = r^{2k} \ln r, \ \ k \in \mathbb{Z}^+. \tag{5.13}$$

Note that Φ in (5.13) can be easily obtained through repeated integration in the polar coordinates. It follows that

$$\Phi(r) = \frac{r^{2k+2}}{4(k+1)^2} \ln r - \frac{r^{2k+2}}{4(k+1)^3}. \tag{5.14}$$

Furthermore,

$$f(x,y) \approx \Delta\hat{u}(x,y) = \sum_{j=1}^{n} a_j r^{2n} \ln r + \sum_{l=1}^{w} a_{n+l} q_l(x,y),\ (x,y) \in \Omega, \tag{5.15}$$

where

$$\Delta p_l(x,y) = q_l(x,y),\ \ l = 1,2,...,w.$$

In addition, the following augmented equation needs to be imposed [27]

$$\sum_{j=1}^{n} a_j p_l(x,y) = 0,\ \ l = 1,2,...,w. \tag{5.16}$$

Using (5.12) and (5.16), we can establish the following matrix system

$$\begin{bmatrix} \hat{\mathbf{u}} \\ \mathbf{0}_w \end{bmatrix} = \begin{bmatrix} A_\Phi & \mathbf{P}_{nw} \\ \mathbf{P}_{nw}^T & \mathbf{0}_{ww} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n+w} \end{bmatrix}, \tag{5.17}$$

where $\mathbf{0}_w$ is the zero matrix of order $w \times 1$, $\hat{\mathbf{u}} = [\hat{u}(x_1,y_1), \cdots, \hat{u}(x_n,y_n)]^T$, $A_\Phi = [\Phi(r_{ij})]_{1 \le i,j \le n}$ and $\mathbf{P}_{il} = p_l(x_i,y_i),\ \ i = 1,2,...,n$. From (5.17), we have

$$\mathbf{a} = \begin{bmatrix} A_\Phi & \mathbf{P}_{nw} \\ \mathbf{P}_{nw}^T & \mathbf{0}_{ww} \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{u}} \\ \mathbf{0}_w \end{bmatrix}, \tag{5.18}$$

where $\mathbf{a} = [a_1, a_2, \cdots, a_{n+w}]^T$. By collocating the interior points, we have

$$f(x_i,y_i) = \sum_{j=1}^{n} a_j r^{2k} \ln r + \sum_{l=1}^{w} a_{n+l} q_l(x_i,y_i),\ \ i = 1,2,...,n_i. \tag{5.19}$$

with the additional augmented equations

$$\sum_{j=1}^{n_i} a_j q_l(x_j,y_j) = 0,\ \ l = 1,2,...,w. \tag{5.20}$$

Similarly, by collocating the boundary points, we have

$$g(x_i,y_i) = \sum_{j=1}^{n} a_j \Phi(r_{ij}) + \sum_{l=1}^{w} a_{n+l} p_l(x_i,y_i),\ \ i = n_i+1,...,n. \tag{5.21}$$

with additional augmented conditions

$$\sum_{j=n_i+1}^{n} a_j p_l(x_j,y_j) = 0,\ \ l = 1,2,...,w. \tag{5.22}$$

From (5.19)–(5.22), we have the following block matrix system

$$\begin{bmatrix} \phi_{n_i n} & \mathbf{Q}_{n_i w} \\ \Phi_{n_b n} & \mathbf{P}_{n_b w} \\ [\mathbf{Q}_{n_i w}^T, \mathbf{P}_{n_b w}^T] & \mathbf{0}_{ww} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n+w} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{n_i} \\ \mathbf{g}_{n_b} \\ \mathbf{0}_w \end{bmatrix}. \tag{5.23}$$

Using (5.23), (5.18) becomes

$$\begin{bmatrix} \phi_{n_i n} & \mathbf{Q}_{n_i w} \\ \Phi_{n_b n} & \mathbf{P}_{n_b w} \\ [\mathbf{Q}_{n_i w}^T, \mathbf{P}_{n_b w}^T] & \mathbf{0}_{ww} \end{bmatrix} \begin{bmatrix} A_\Phi & \mathbf{P}_{nw} \\ \mathbf{P}_{nw}^T & \mathbf{0}_{ww} \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{u}} \\ \mathbf{0}_{ww} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{n_i} \\ \mathbf{g}_{n_b} \\ \mathbf{0}_w \end{bmatrix}. \tag{5.24}$$

We can rewrite equation (5.24) as

$$\begin{bmatrix} \phi_{n_i n_i} & \phi_{n_i n_b} & \mathbf{Q}_{n_i w} \\ \Phi_{n_b n_i} & \Phi_{n_b n_b} & \mathbf{P}_{n_b w} \\ \mathbf{Q}_{n_i w}^T & \mathbf{P}_{n_b w}^T & \mathbf{0}_{ww} \end{bmatrix} \begin{bmatrix} \Phi_{n_i n_i} & \Phi_{n_i n_b} & P_{n_i w} \\ \Phi_{n_b n_i} & \Phi_{n_b n_b} & P_{n_b w} \\ P_{n_i w}^T & P_{n_b w}^T & \mathbf{0}_{ww} \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{u}}_{n_i} \\ \hat{\mathbf{u}}_{n_b} \\ \mathbf{0}_w \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{n_i} \\ \mathbf{g}_{n_b} \\ \mathbf{0}_w \end{bmatrix}. \tag{5.25}$$

Let

$$A = \begin{bmatrix} \phi_{n_i n_i} & \phi_{n_i n_b} & \mathbf{Q}_{n_i w} \\ \Phi_{n_b n_i} & \Phi_{n_b n_b} & \mathbf{P}_{n_b w} \\ \mathbf{Q}_{n_i w}^T & \mathbf{P}_{n_b w}^T & \mathbf{0}_{ww} \end{bmatrix}, \quad B = \begin{bmatrix} \Phi_{n_i n_i} & \Phi_{n_i n_b} & P_{n_i w} \\ \Phi_{n_b n_i} & \Phi_{n_b n_b} & P_{n_b w} \\ P_{n_i w}^T & P_{n_b w}^T & \mathbf{0}_{ww} \end{bmatrix}.$$

Since the matrices A and B have the same second row, Lemma 1 in the Appendix implies that the second rows of the matrices AB^{-1} and BB^{-1} are equal. Thus, the product matrix AB^{-1} takes the form

$$AB^{-1} = \begin{bmatrix} \mathbf{C}_{n_i n_i} & \mathbf{C}_{n_i n_b} & \mathbf{C}_{n_i w} \\ \mathbf{0}_{n_b n_i} & \mathbf{I}_{n_b n_b} & \mathbf{0}_{n_b w} \\ \mathbf{C}_{w n_i} & \mathbf{C}_{w n_b} & \mathbf{C}_{ww} \end{bmatrix}. \tag{5.26}$$

Consequently, from (5.25) and (5.26), we have

$$\begin{bmatrix} \mathbf{C}_{n_i n_i} & \mathbf{C}_{n_i n_b} & \mathbf{C}_{n_i w} \\ \mathbf{0}_{n_b n_i} & \mathbf{I}_{n_b n_b} & \mathbf{0}_{n_b w} \\ \mathbf{C}_{w n_i} & \mathbf{C}_{w n_b} & \mathbf{C}_{ww} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_{n_i} \\ \hat{\mathbf{u}}_{n_b} \\ \mathbf{0}_w \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{n_i} \\ \mathbf{g}_{n_b} \\ \mathbf{0}_w \end{bmatrix}. \tag{5.27}$$

Note that (5.18) can be further reduced to the following matrix system

$$\begin{bmatrix} \mathbf{C}_{n_i n_i} & \mathbf{C}_{n_i n_b} \\ \mathbf{0}_{n_b n_i} & \mathbf{I}_{n_b n_b} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_{n_i} \\ \hat{\mathbf{u}}_{n_b} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{n_i} \\ \mathbf{g}_{n_b} \end{bmatrix}. \tag{5.28}$$

Equation (5.28) implies that

$$\mathbf{C}_{n_i n_i} \hat{\mathbf{u}}_{n_i} + \mathbf{C}_{n_i n_b} \hat{\mathbf{u}}_{n_b} = \mathbf{f}_{n_i}, \tag{5.29}$$

$$\hat{\mathbf{u}}_{n_b} = \mathbf{g}_{n_b}. \tag{5.30}$$

Substituting (5.30) into (5.29), we can further reduce to the following system

$$\hat{\mathbf{u}}_{n_i} = \mathbf{C}_{n_i n_i}^{-1}(\mathbf{f}_{n_i} - \mathbf{C}_{n_i n_b}\mathbf{g}_{nb}). \tag{5.31}$$

It is obvious that solving (5.31) is more efficient than solving (5.27).

The above-mentioned MPS can be extended to more general equations containing variable coefficients or three-dimensional cases. We refer readers to [15] for more details.

5.3 Critical Domains for Quenching Problems

In this section, we apply the MPS to find the critical domain for the quenching problem (5.7) – (5.8). We suppose $f(u) = 1/(1-u)$. Then, we have,

$$\Delta u - u_t = -\frac{\gamma}{1-u} \text{ in } \Omega, \tag{5.32}$$

$$u = 0 \text{ on } \bar{\mathbf{D}} \cup S, \tag{5.33}$$

where $\gamma = \lambda^2$. The critical value of γ is obtained from the steady-state form of the above problem. We have,

$$\Delta U = -\frac{\gamma}{1-U} \text{ in } \mathbf{D}, \tag{5.34}$$

$$U = 0 \text{ on } \partial\mathbf{D}. \tag{5.35}$$

Once we compute the critical value γ^* of γ, we can use equation (5.9) to compute critical size of the considered domain. Here, we present an algorithm to compute the critical value γ^*.

Algorithm 3

Step 1: The upper bound γ_{upper} for γ^* is obtained from the solution of the problem
$\Delta\xi = -1$ in $\mathbf{D}$,
$\xi = 0$ on $\partial\mathbf{D}$.
We have, $\gamma_{upper} = 1/max_{\bar{\mathbf{D}}}\xi$ and choose $\gamma_{lower} = 0$.
Estimate γ^* with $\gamma^{(1)} = (\gamma_{upper} - \gamma_{lower})/2$.

Step 2: For each $\gamma^{(k)}$, Compute $\left\{U^{(i)}\right\}$ defined by
$U^{(0)} = 0$ *on* $\bar{\mathbf{D}}$,
and for $i \geq 1$,
$\Delta U^{(i)} = -\frac{\gamma^k}{1-U^{(i-1)}}$ *in* $\mathbf{D}$,
$U^{(i)} = 0$ *on* $\partial\mathbf{D}$.
For each $\gamma^{(k)}$, Compute $\left\{U^{(i)}\right\}$ by the modified MPS.
If $\left\{U^{(i)}\right\}$ converges, the corresponding $\gamma^{(k)}$ is the lower bound for γ^*; otherwise, it is an upper bound.

Step 3: If $\left|\gamma^{(k-1)} - \gamma^{(k-2)}\right| < \varepsilon,\ \varepsilon > 0,$
(a given tolerance), then $\gamma^{(k-1)}$ is the approximate critical value.

Step 4: Update $\gamma^{(k-1)}$ in the following way:
If $\left\{U^{(i)}\right\}$ in Step 2 converges,
$\gamma^{(k)} = \gamma^{(k-1)} + \frac{1}{2}\left|\gamma^{(k-1)} - \gamma^{(k-2)}\right|$;
otherwise
$\gamma^{(k)} = \gamma^{(k-1)} - \frac{1}{2}\left|\gamma^{(k-1)} - \gamma^{(k-2)}\right|$.

Step 5: Repeat Steps $2-4$ until the tolerance is reached.

As a numerical experiment, the critical values are computed by using the modified MPS with the above algorithm for the problem (5.32) – (5.33) defined on different domains of given shape. Furthermore, using the relation (5.9), the critical sizes of the domains are calculated and compared with the sizes obtained from other numerical methods.

5.4 Numerical Results

To demonstrate the effectiveness of the modified MPS, we consider three numerical examples of Poisson-type nonlinear problems with regular and irregular domains. The domains we considered are squares, amoeba-like, and peanut-like domains. Furthermore, critical sizes of the domains are computed for quenching problems defined on regular and irregular domains and we compare the results with results obtained from other computational methods. The

parametric equation of the boundary $\partial\Omega$ is defined as follows

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

where

$$r(\theta) = e^{\sin\theta}\sin^2(2\theta) + e^{\cos\theta}\cos^2(2\theta) \tag{5.36}$$

is the amoeba-like boundary and

$$r(\theta) = \sqrt{\cos(2\theta) + \sqrt{1.1 - \sin^2(2\theta)}} \tag{5.37}$$

is the peanut-like boundary. The profiles of the amoeba-like and peanut-like domains are shown in Figure 5.1.

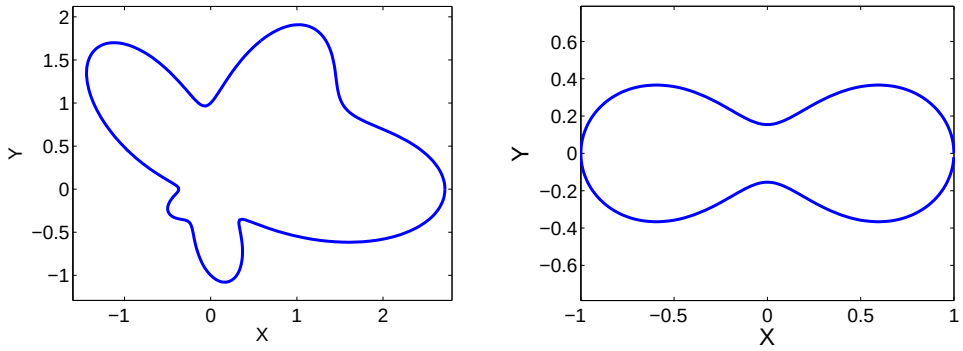


Figure 5.1: The profiles of amoeba-like and peanut-like domains.

Through all numerical examples except Example 5.4.5 in this section, we have chosen the tolerance $\varepsilon = 10^{-7}$ in Algorithm 2 to ensure the accuracy of the solution. We choose polyharmonic splines of order 2 as the radial basis function plus a polynomial basis up to degree 2; i.e., $\{1, x, y, x^2, xy, y^2\}$. If necessary, higher accuracy can be achieved using high order polyharmonic splines. We find polyharmonic splines of order 2 to be sufficient for solving nonlinear problems in this section. The boundary and interior points are selected uniformly through the domain.

Example 5.4.1. Consider the following nonlinear Poisson problem with Dirichlet boundary condition:

$$\Delta u(x,y) = 3u^2, \quad (x,y) \in \Omega, \tag{5.38}$$

$$u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega, \tag{5.39}$$

where $g(x,y)$ is given based on the following analytical solution

$$u_{exact}(x,y) = \frac{4}{(3+x+y)^2}.$$

In Table 5.1, we show the numerical results for three different domains. From the table, we observe that the accuracy is quite high. The algorithm is also very efficient since only a small number of iterations is required for all these three domains. The reasons that the unit square domain has high accuracy could be because the domain is more regular and the area of the domain is smaller than the other two domains.

Table 5.1: Example 5.4.1: The RMSE and Maximum error with different domains.

Domain	(n_i, n_b)	# Iterations	RMSE	MAE
Unit Square	$(841,236)$	6	$4.378e-08$	$1.233e-07$
Amoeba	$(861,300)$	15	$5.220e-07$	$8.406e-06$
Peanut	$(832,290)$	15	$1.204e-06$	$2.265e-05$

Example 5.4.2. Let us consider the following nonlinear Poisson type problem:

$$\Delta u(x,y) = u^3 - \frac{5}{2} - \left(1 - \frac{x^2}{4} - y^2\right)^3, \ (x,y) \in \Omega, \tag{5.40}$$

$$u(x,y) = g(x,y), \ (x,y) \in \partial\Omega, \tag{5.41}$$

where $g(x,y)$ is given based on the following exact solution

$$u_{exact}(x,y) = 1 - \frac{x^2}{4} - y^2.$$

In this example, we have tested our method for the Poisson-type nonlinear PDE which includes the space variables x and y in the source term. The numerical results are presented with the same domains as in previous examples. In Table 5.2, we observe that fewer of iterations are required to obtain excellent accuracy. Therefore, the method can be a good alternative to solve such problems accurately and efficiently.

Example 5.4.3. In this example, we consider a Poisson-Boltzmann Equation [6], a typical example of Poisson-type nonlinear equations. This equation has been widely applied in physical problems including bio-molecular processes and electro-static interactions between colloidal particles. Here, we solve this equation defined on a square domain to observe the efficiency and accuracy of the present method. Consider the problem

$$\nabla \cdot (\varepsilon \nabla u) = \kappa \sinh(u) + f, \ \ (x,y) \in \Omega, \tag{5.42}$$

Table 5.2: Example 5.4.2: The RMSE and Maximum error with different domains.

Domain	(n_i, n_b)	# Iterations	RMSE	MAE
Unit Square	$(841, 236)$	7	$2.987e-10$	$6.224e-10$
Amoeba	$(861, 300)$	15	$3.132e-09$	$9.966e-09$
Peanut	$(832, 290)$	8	$1.937e-09$	$3.827e-09$

with boundary condition

$$u(x,y) \quad = \quad g(x,y),\ (x,y) \in \partial\Omega. \tag{5.43}$$

Here, ε and κ are known field functions. In this experiment, we let $\varepsilon = 1$ and $\kappa = 1$. The function $g(x,y)$ is given based on the following exact solution, $u_{exact}(x,y) = x^2 + y^2 + e^x \cos(y)$ and the source term $f = 4 - \sinh(x^2 + y^2 + e^x \cos(y))$. We have generated different sets of interior points and boundary points on a square domain $D = [-1,1] \times [-1,1]$. In Table 5.3, various number of interior and boundary points are chosen and excellent results have been observed. The improvement of the accuracy is consistent with the increasing number of interior and boundary points. The number of iterations for various number of interior and boundary points in Table 5.3 is 26.

Table 5.3: Example 5.4.3: The RMSE and Maximum error with different set of collocation points.

(n_i, n_b)	RMSE	MAX
$(361, 88)$	$3.516e-05$	$1.049e-04$
$(576, 116)$	$1.526e-05$	$4.830e-05$
$(1089, 316)$	$4.778e-06$	$1.646e-05$
$(1444, 476)$	$2.831e-06$	$9.953e-06$
$(3364, 596)$	$5.766e-07$	$2.084e-06$

Example 5.4.4. In this example, we extend the MPS for solving nonlinear problems to compute critical domains of quenching problems using Algorithm 2. Let us consider the rectangular domain $D = [0,a] \times [0,b]$. The ratio a/b determines the shape of the rectangle. Using the relation (5.9), we have,

$$\text{Critical size of the domain} = \gamma^* \cdot (\text{Area of } D). \tag{5.44}$$

In Table 5.4, we show the results of the critical size computation for rectangular domains with different ratios a/b and compare our results with Chan and Ke [12], who had adopted the finite difference method (FDM). We have taken 841 interior points and 236 boundary points. The results of both approaches are very close to each other. Since the computational domain is rectangular, the FDM can be easily applied. However, for an irregular domain, the FDM will have difficulty in implementing the solution algorithm. On the other hand, the numerical procedure for the proposed MPS for the irregular domain is the same as the rectangular domain. This is one of the attractive features of the MPS for solving nonlinear or quenching problems.

Table 5.4: Example 5.4.4: Critical size of domains for rectangles with the different ratio a/b.

Ratio a/b	FDM	MPS
0.125	18.80540	18.81603
0.250	9.67221	9.67679
0.375	6.85011	6.85462
0.500	5.59863	5.60066
0.625	4.96792	4.97582
0.750	4.64531	4.64751
0.875	4.49641	4.50236
1.000	4.45375	4.46474

Example 5.4.5. In this example, the critical domain sizes of quenching problems are produced for three different domain shapes including regular and irregular geometries and compared with Chan [11] and Chan and Ke [12], Tian [74]. Chan [11] devised a computational method using Green's functions whereas Chan and Ke [12] used the finite difference method and Tian [74] applied a numerical method using Delta-shaped basis functions to compute the sizes of the critical domains.

In this numerical experiment, we have considered three domains, namely rectangle, ellipse, and peanut. The rectangular domain is defined as $[0,0.5]\times[0,1]$ whereas the elliptic domain is defined as

$$D=\left\{(x,y):\frac{x^2}{a^2}+\frac{y^2}{b^2}<1\right\} \tag{5.45}$$

where $a=0.4575$ and $b=0.3$. The peanut-like domain we use in this example is different from the one we considered in the examples above. The domain is defined in polar form as

$$r(\theta)=\frac{1}{4}(1+\cos^2(\theta)),\ 0\leq\theta\leq 2\pi.$$

To make the comparison, we choose the domain the same as taken by previous authors in [11, 12, 74]. The profile of the peanut-like domain in this example is shown in Figure 5.2. We have chosen 729 interior points and 208 boundary points of the rectangular domain and 858 interior points and 300 boundary points for the elliptical domain. Similarly, the interior points and boundary points taken for the peanut-like domain are 780 and 300 respectively. Furthermore, the areas of the rectangle, ellipse and peanut-like domains are 0.5, 0.4312, and 0.4663 respectively. As shown in Table 5.5, we notice that the critical size of three domains using the modified MPS is close to the results obtained by other methods. The computer running time for rectangle, ellipse, and peanut-like domains are 4.44, 6.59, and 6.61 seconds respectively. There is no report of the numerical efficiency of previous work.

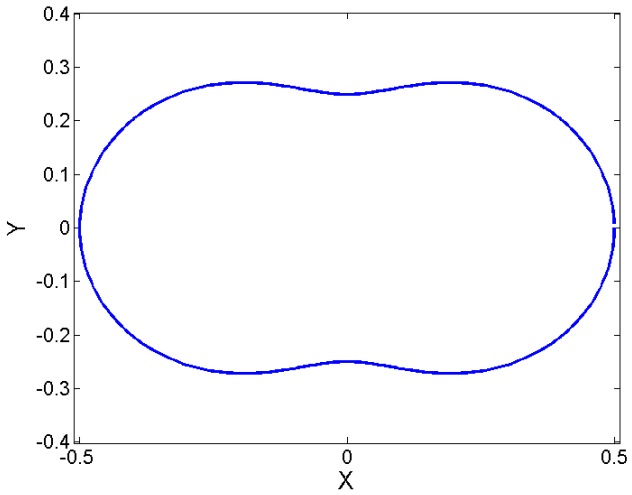


Figure 5.2: Example 5.4.5: The profile of peanut-like domain.

Table 5.5: Example 5.4.5: Critical size of domains for different geometries.

Domains	Delta-shaped function	MPS	FDM	Green's function
Rectangle	5.599	5.601	5.599	–
Ellipse	4.460	4.466	–	4.460
Peanut	5.052	5.053	–	–

Appendix

Lemma 1. Let A, B and C be three matrices of same size in block matrix form with 3 row partitions and 3 column partitions such that the sub-matrices on the second row of the matrix

A are equal to the corresponding sub-matrices of B. Let

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ A_{21} & A_{22} & A_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

$$\text{and} \qquad C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}.$$

Then the sub-matrices on the second row of the product matrix AC are equal to the corresponding sub-matrices of the product matrix BC.

Chapter 6

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

In this dissertation, polynomial particular solutions for general linear differential operators with constant coefficients have been derived. This is a further improvement of the previous works [18, 32] where the differential operator contains no convective terms. Instead of confining the collocation points on the Gauss-Lobatto points in a rectangular domain, we are allowed to choose the collocation points in an arbitrary fashion using our proposed algorithm. A multiple scale technique is required to alleviate the ill-conditioning problem. Because the polynomial particular solutions are available, the MPS can be easily employed to solve various types of elliptic partial differential equations. Another advantage of the proposed algorithm is that there is no parameter to be adjusted and the algorithm is very stable and highly accurate. Once the particular solutions are generated and stored in a table, the computation is very efficient.

The particular solutions can not be obtained if the differential operator has variable coefficients or $a_6 = 0$. Therefore, Poisson's problem and PDEs with variable coefficients can not be solved using this approach. The method has been modified to solve such problems. We have presented many numerical examples solved by the new scheme. The numerical stability and efficiency of the method have been preserved even after modification. With this new modification, we are able to solve a large class of partial differential equations. We have tested the method on solving nonlinear, time-dependent, and 3-D problems. The effectiveness of the method for solving such problems is very promising.

To solve large-scale problems, the localized version of the MPS based on polynomial basis function is devised. The method has been employed to solve Poisson's problem, PDEs with constant coefficients, and PDEs with variable coefficients. Furthermore, 3-D problems defined on regular and complicated domains are successfully solved by this new algorithm. Additionally, the method is applied to solve some near-singular problems. Numerical results show that the proposed algorithm is highly stable and accurate.

The numerical outcomes have proven that the particular solutions derived in this dissertation have made these two methods easy to implement and highly efficient to solve a large class of partial differential equations.

As another part of the dissertation, the traditional MPS has been modified for and applied to solving nonlinear Poisson-type problems. The numerical results indicate that the method can be an attractive alternative to other traditional methods. Furthermore, the method is further used to compute the critical domains of quenching problems and compared to the results of some other established numerical methods such as the finite difference method. The comparisons show that the modified MPS can effectively solve nonlinear singular problems. Unlike the FDM, it can solve such problems more efficiently defined on irregular domains with higher dimensions. The simplicity of the implementation is another attraction of this method, apart from the numerical accuracy and efficiency.

6.2 Future Works

The derivation of particular solutions for general partial differential operators is an important contribution to the field of numerical methods. We have done extensive work to test these particular solutions on two different numerical methods to see their effectiveness in solving various partial differential equations. The derived particular solutions have made the MPS and the LMPS more accurate and simple. These solutions have opened up future research topics:

- The MPS is applied to solve a limited number of nonlinear and time-dependent problems. We are planning to solve more challenging problems such as quenching problem [12], the Burger's equation [47], and the Fisher's equation [9, 44].

- For nonlinear problems, we have applied Picard's iteration in the solution process. For Picard's iteration, the forcing term should be Lipschitz continuous and the Lipschitz constant should be strictly less than 1 for the sequence to converge. We will use the Newton's iteration method instead of Picard's iteration in the MPS to solve nonlinear problems.

- To prove the further applicability of our new schemes, we are collaborating with Dr. Ji Lin to solve more engineering problems such as the inverse Cauchy Problem [53] and plate bending vibration problems [37] .

- The LMPS with polynomial basis functions has successfully solved many PDEs including near-singular problems. We have not solved nonlinear and time-dependent problems using this algorithm. We plan to solve such problems in the future.

- The MPS and LMPS will be used to solve more challenging problems in 3-D with complicated domains. The method will also be extended to solve axisymmetric

problems [66, 67].

Appendix A

POLYNOMIAL PARTICULAR SOLUTIONS

A.1 Poisson Type Problems

The particular solutions of polynomial basis functions for the Laplacian operator for 2D and 3D cases [23, 32] are as follows:

A particular solution of

$$\Delta\phi = x^n y^m, \quad m \geq 0, n \geq 0, \tag{A.1}$$

is given by

$$\phi(x,y) = \begin{cases} \sum_{k=1}^{[\frac{n+2}{2}]} (-1)^{k+1} \frac{m!n!x^{m+2k}y^{n-2k+2}}{(m+2k)!(n-2k+2)!} & m \geq n, \\ \sum_{k=1}^{[\frac{m+2}{2}]} (-1)^{k+1} \frac{m!n!x^{m-2k+2}y^{n+2k}}{(m-2k+2)!(n+2k)!} & m < n. \end{cases} \tag{A.2}$$

Similarly, the particular solution of

$$\Delta\phi = x^l y^m z^n, \quad m \geq 0, n \geq 0, \tag{A.3}$$

is given by

$$\phi(x,y) = \sum_{i=1}^{[\frac{m}{2}]+[\frac{n}{2}]+1} x^{l+2i} \sum_{j=max\{1,i-[\frac{n}{2}]\}}^{min\{i,[\frac{m}{2}]+1\}} a_{ij} y^{m-2j+2} z^{n-2i+2j} \tag{A.4}$$

where

$$\begin{aligned} a_{ij} = \frac{-1}{(l+2i)(l+2i-1)} [&(m-2j+4)(m-2j+3)a_{i-1,j-1} \\ &+(n-2i+2j+2)(n-2i+2j+1)a_{i-1,j}], \end{aligned} \tag{A.5}$$

$$a_{11} = a_{i0} = a_{0j} = 0. \tag{A.6}$$

A.2 Helmholtz Type Problems

The particular solutions of polynomial basis functions for the Helmholtz type operator for 2D and 3D cases [62, 32] are as follows: For $\varepsilon = \pm 1$,the particular solution of

$$\Delta\phi + \varepsilon\lambda^2\phi = x^n y^m, \;\; m,n \geq 0 \tag{A.7}$$

is given by

$$\phi(x,y) = \sum_{0}^{[\frac{m}{2}]}\sum_{0}^{[\frac{n}{2}]} \frac{\varepsilon(-\varepsilon)^{k+l}(k+l)!m!n!x^{m-2k}y^{n-2l}}{\lambda^{2k+2l+2}k!l!(m-2k)!(n-2l)!}. \tag{A.8}$$

where $[t]$ is the largest integer that is less than or equal to t.

Similarly, for $\varepsilon = \pm 1$, the particular solution of

$$\Delta\phi + \varepsilon\lambda^2\phi = x^p y^p z^r, \tag{A.9}$$

is given by

$$\phi(x,y,z) = \sum_{j=0}^{[\frac{p}{2}]}\sum_{k=0}^{[\frac{q}{2}]}\sum_{l=0}^{[\frac{r}{2}]} \frac{\varepsilon(-\varepsilon)^{k+l}(j+k+l)!p!q!r!x^{p-2j}y^{q-2k}z^{r-2l}}{\lambda^{2j+2k+2l+2}j!k!l!(p-2j)!(q-2k)!(r-2l)!} \tag{A.10}$$

where p,q and r are the positive integers and $[t]$ is the largest integer that is less than or equal to t.

BIBLIOGRAPHY

[1] http://graphics.stanford.edu/data/3dscanrep/.

[2] A. Acker and B. Kawohl. Remarks on quenching. *Nonlinear Analysis*, 13:53–61, 1989.

[3] K.E. Atkinson. The numerical evaluation of particular solutions for Poisson's equation. 5:319–338, 1985.

[4] T. Belytschko, Y. Krongauz, D.J. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering, special issue*, 139:3–47, 1996.

[5] T. Belytschko, Y. Lu, and L. Gu. Element free galerkin method. *Int. J. Num. Eng.*, 37:229–256, 1994.

[6] G.C. Bourantas and V.N. Burganos. An implicit meshless scheme for the solution of transient non-linear Poisson-type equations. *Eng Anal Bound Elem,*, 37:1117–1126, 2013.

[7] D. Braess. *Finite Elements. Theory, Fast Solvers and Applications in Solid Mechanics*. Cambridge University Press, 2001.

[8] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods, Second edition*. Springer, 2002.

[9] N.F. Britton. *Reaction-Diffusion Equations and Their Applications to Biology*. Academic Press, 1986.

[10] M.D. Buhmann. *Radial Basis Functions, Theory and Implementations*. Cambridge University Press, 2003.

[11] C. Chan. Computation of the critical domain for quenching in an elliptic plate. *Neural Parallel Sci. Comput.,*, 1:153–162, 1993.

[12] C.Y. Chan and L. Ke. Parabolic quenching for nonsmooth convex domains. *J Math Anal Appl,*, 116:52–65, 1994.

[13] C.S. Chen. The method of fundamental solutions for non-linear thermal explosions. *Comm. Numer. Methods Engrg.*, 11(8):675–681, 1995.

[14] C.S. Chen, C.M. Fan, and P.H. Wen. The method of particular solutions for solving elliptic problems with variable coefficients. *The International Journal for Numerical Methods in Biomedical Engineering*, 8:545–559, 2011.

[15] C.S. Chen, C.M. Fan, and P.H. Wen. The method of particular solutions for solving certain partial differential equations. *Numerical Methods for Partial Differential Equations*, 28:506–522, 2012.

[16] C.S. Chen, M.A. Golberg, M. Ganesh, and A.H.-D. Cheng. Multilevel compact radial functions based computational schemes for some elliptic problems. *Computers and Mathematics with Application*, 43:359–378, 2002.

[17] C.S. Chen, M.A. Golberg, and R. Schaback. Recent developments of the dual reciprocity method using compactly supported radial basis functions. In Y.F. Rashed, editor, *Transformation of Domain Effects to the Boundary*, pages 183–225. WIT Press, 2003.

[18] C.S. Chen, S.W. Lee, and C.-S. Huang. Derivation of particular solution using Chebyshev polynomial based functions. *International Journal of Computational Methods*, 4(1):15–32, 2007.

[19] J.T. Chen, Y.T. Lee, S.R. Yu, and S.C. Shieh. Equivalence between the Trefftz method and the method of fundamental solution for the annular Green's function using the addition theorem and image concept. *Engineering Analysis with Boundary Elements*, 33(5):678–688, 2009.

[20] J.T. Chen, C.S. Wu, Y.T. Lee, and K.H. Chen. On the equivalence of the Trefftz method and method of fundamental solutions for Laplace and biharmonic equations. *Computers and Mathematics with Applications*, 53(6):851–879, 2007.

[21] W. Chen. Symmetric boundary knot method. *Engineering Analysis with Boundary Elements*, 26:489–494, 2002.

[22] A.H.-D. Cheng. Particular solutions of Laplacian, Helmholtz-type, and polyharmonic operators involving higher order radial basis functions. *Eng. Analy. Boundary Elements*, 24:531–538, 2000.

[23] A.H.-D. Cheng, O. Lafe, and S. Grilli. Dual reciprocity BEM based on global interpolation functions. *Eng. Anal. Boundary Elements*, 13:303–311, 1994.

[24] K. Deng and H.A. Lavine. On the blow up of u_t at quenching. *Proc. Amer. Math. Soc.,*, 106:1049–1056, 1989.

[25] E. Divo and A.J. Kassab. An efficient localized rbf meshless method for fluid flow and conjugate hear transfer. *ASME Journal of Heat Transfer*, 129:124–136, 2007.

[26] G. Fairweather and A. Karageorghis. The method of fundamental solution for elliptic boundary value problems. *Advances in Computatonal Mathematics*, 9:69–95, 1998.

[27] G. Fasshauer. *Meshfree Approximation Methods with MATLAB. Interdisciplinary Mathematical Science*. World Scientific Publishing Company,, 2007.

[28] G. Fasshauer and G. Liu. *Meshfree methods: moving beyond the finite element method.* CR Press: Boca Raton,, 2007.

[29] GregoryE. Fasshauer and JackG. Zhang. On choosing optimal shape parameters for rbf approximation. *Numerical Algorithms*, 45(1-4):345–368, 2007.

[30] R.A. Gingold and J. J. Moraghan. Smoothed particle hydrodynamics: theory and application to non spherical stars. *Man. Not. Astrou. soc.*, 181:375–389, 1977.

[31] M.A. Golberg, C.S. Chen, and Y.F. Rashed. The annihilator method for computing particular solutions to partial differential equations. *Eng. Analy. Boundary Elements*, 23:275–279, 1999.

[32] M.A Golberg, A.S. Muleshkov, C.S. Chen, and A.H.-D. Cheng. Polynomial particular solutions for certain kind of partial differential operators. *Numerical Methods for Partial Differential Equations*, 19:112–133, 2003.

[33] F. Hartmann. *Elastostatics, Progress in Boundary Element Methods.* Pentech Press, London, pp. 84-167, 1981.

[34] Y.C. Hon. Typhoon surge in Tolo Harbour of Hong Kong - an approach using finite element method with quadrilateral elements and parallel processing technique. *Chinese J. Num. Math. Appl.*, 15(4):21–33, 1993.

[35] J. Houbolt. A recurrence matrix solution for the dynamic response of elastic aircraft. *Journal of the Aeronautical Sciences,*, 17:540–550, 1950.

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

[37] T.J.R. Hughes, R.L. Taylor, and W. Kanoknukulchai. A simple and efficient finite element for plate bending. *International Journal for Numerical methods in Engineering.*, 11:1529–1543, 1977.

[38] R.L. Johnston, G. Fairweather, and A. Karageorghis. An adaptive indirect boundary element method with applications. In M. Tanaka and C.A. Brebbia, editors, *Boundary Elements VIII*, pages 587–597. Springer-Verlag, Tokyo, 1987.

[39] E.J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - I. *Comput. Math. Applic.*, 19(8/9):127–145, 1990.

[40] E.J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - II. *Comput. Math. Applic.*, 19(8/9):147–161, 1990.

[41] J.J. Kasab, S. Karur, and P. Ramachandran. Quasi linear boundary element method for nonlinear Poisson-type problems. *Eng Anal Bound Elem,*, 15:277–282, 1995.

[42] H. Kawarada. On Solutions of Initial-Boundary Problem for $u_t = u_{xx} + \frac{1}{1-u}$.. *Publ. Res.Inst. Math.*, 10:729–736, 1975.

[43] V.D. Kupradze and M.A. Aleksidze. The method of functional equations for the approximate solution of certain boundary value problems. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 4:82–126, 1964.

[44] S. Langdon. *Domain Embedding Boundary Integral Equation Methods and Parabolic PDEs*. PhD thesis, University of Bath, 1999.

[45] C. K. Lee, X. Liu, and S. C. Fan. Local multiquadric approximation for solving boundary value problems. *Computational Mechanics*, 30:396–409, 2003.

[46] R. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[47] J. Li, Y.C. Hon, and C.S. Chen. Numerical comparisons of two meshless methods using radial basis functions. *Eng. Anal. Boundary Elements*, 26:205–225, 2002.

[48] Jichun Li and Yi-Tung Chen. *Computational Partial Differential Equations Using MATLAB*. Chapman & Hall/CRC, 2008.

[49] Z. Li, C.V. Pao, and Z.A. Qiao. A Finite Difference Method and Analysis for 2D Nonlinear Poisson-Boltzmann Equations. *J Sci Comput,*, 30:61–81, 2007.

[50] Y.P. Lin and T. Zhang. Finite element methods for nonlinear Sobolev equations with nonlinear boundary conditions. *J Math Anal Appl,*, 165:180–191, 1992.

[51] C. Liu. A highly accurate MCTM for inverse Cauchy problems of Laplace equations in arbitrary plane domains. *CMES*, 35:91–111, 2008.

[52] C. Liu. A multiple-scale Trefftz method for an incomplete Cauchy problem of biharmonic equaion. *Eng Anal Bound Elem,*, 37:1445–1446, 2013.

[53] C. S. Liu and D.L. Young. A multiple-scale Pascal polynomial for 2D Stokes and inverse Cauchy-Stokes problems. *Journal of Computational Physics*, 312:1–13, 2016.

[54] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Partical Method*. World Sci., River Edge, NJ, 2003.

[55] W.K. Liu, S. Jun, and Y.F. Zhang. Reproducing kernel particle methods. *Int. J. Numer. Methd. Fluids*, 20:1081–1106, 1995.

[56] Xiao-Yan Liu, Andreas Karageorghis, and C.S. Chen. A Kansa-Radial Basis Function Method for Elliptic Boundary Value Problems in Annular Domains. *J Sci Comput*, 65:1240–1269, 2015.

[57] T. May. *Boundary Conditions in the Numerical Integration of Hyperbolic Equations*. PhD thesis, University of Reading, Reading, May 1978.

[58] I. McCausland. *Introduction to Optimal Control*. John Wiley & Sons, 1969.

[59] J.M. Melenk and I. Babuska. The Partition of Unity Finite Element Method: Basic Theory and Applications. *Comput. Meths. Appl. Mech. Engrg.*, 139:289–314, 1996.

[60] G.L. Moridis and D.L. Reddell. The Laplace transform boundary element (LTBE) method for the solution of diffusion-type equations. In C.A. Brebbia, editor, *Boundary Elements XIII*, pages 83–97. Springer-Verlag, Berlin, 1991.

[61] A. Muleshkov, M. Golberg, and C.S. Chen. Particular solutions of the multi-helmholtz type equaion. *Eng Anal Bound Elem,*, 31(7):624–30, 2007.

[62] A.S. Muleshkov, C.S. Chen, M.A. Golberg, and A.H-D. Cheng. Analytic particular solutions for inhomogeneous Helmholtz-type equations. In S.N. Atluri and F.W. Brust, editors, *Advances in Computational Engineering & Sciences*, pages 27–32. Tech Science Press, 2000.

[63] A.S. Muleshkov, M.A. Golberg, and C.S. Chen. Particular solutions of Helmholtz-type operators using higher order polyharmonic splines. *Comp. Mech.*, 23:411–419, 1999.

[64] J. N. Reddy. *An Introduction to the Finite Element Method*. McGraw-Hill, Inc., 2006.

[65] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11:193–210, 1999.

[66] B. Sarler. Axisymmetric augmented thin plate splines. *Engineering analysis with boundary elements*, 21:81–85, 1998.

[67] B. Sarler, N. Jelic, I. Kovacevic, M. Lakner, and J. Perko. Axisymmetric multiquadrics. *Engineering analysis with boundary elements*, 30:137–142, 2006.

[68] B. Sarler and R. Vertnik. Meshfree explicit local radial basis function collocation method for diffusion problems. *Computers and Mathematics with Applications*, 21:1269–1282, 2006.

[69] R. Schaback. Convergence analysis of methods for solving general equations. In A. Kassab, C.A. Brebbia, E. Divo, and D. Poljak, editors, *Boundary Elements XXVII*, pages 17–24. WITPress, Southampton, 2005.

[70] C. Shu, H. Ding, and K.S. Yeo. Local radial basis function-based differential quadrature method and its application to solve two dimensional incompressible Navier-Stokes equations. *Computer Methods and Applied Mechanics Engineering*, 192:941–954, 2003.

[71] Steven Skiena. *The Algorithm Design Manual, 2nd Edition*. Springer, 2008.

[72] G.D. Smith. *Numerical Solution of partial differential equations, Finite Difference Methods*. Oxford University Preox (3rd ed.), 1985.

[73] A. Soroushian and J. Farjoodi. A unified starting procedure for the Houbolt method. *Communications in Numerical Methods in Engineering,*, 24(1):1–13, 2008.

[74] H. Tian. Computations of critical domains for quenching problems by delta-shaped basis functions. *Neural, Parallel and Scientific Computations*, 15:501–514, 2007.

[75] H.Y. Tian, S. Reutskiy, and C.S. Chen. A new basis function for approximation and the solutions of partial differential equations. *In Proceedings of International Conference on Computational and Experimental Engineering and Sciences*, pages 851–857, 2007.

[76] C. Tsai, J. Kolibal, and M. Li. The golden section search algorithm for finding a good shape parameter for meshless collocation methods. *Eng Anal Bound Elem,*, 34:738–746, 2010.

[77] C.C. Tsai. Particular solutions of Chebyshev polynomials for polyharmonic and poly-Helmholtz equations. *CMES: Comput. Model. Eng. Sci.*, 27(3):151–162, 2008.

[78] R. Vertnik and B. Sarler. Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *International Journal of Numerical Methods for Heat and Fluid Flow*, 16:617–640, 2006.

[79] P.H. Wen and C.S. Chen. The method of particular solutions for solving scalar wave equations. *The International Journal for Numerical Methods in Biomedical Engineering*, 26:1878–1889, 2010.

[80] J. Wertz, E.J. Kansa, and L. Ling. The role of the multiquadric shape parameters in solving elliptic partial differential equations. *Computers & Mathematics with Applications*, 51(8):1335 – 1348, 2006.

[81] L. Yan and F. Yang. The method of approximate particular solutiions for the time fractional diffusion equation with a non-local boundary condition. *Computers and Mathematics with Applications*, 70(3):354–364, 2015.

[82] G. Yao, C.S. Chen, and C.C. Tsai. A revisit on the derivation of the particular solution for the differential operator $\Delta^2 - \lambda^2$. *Advances in Applied Mathematics and Mechanics*, 1:750–768, 2009.

[83] G. Yao, J. Kolibal, and C.S. Chen. A localized approach for the method of approximate particular solutions. *Comput. Math. Appl.*, 61(9):2376–2387, May 2011.

[84] G. Yao, H. Zheng, and C.S. Chen. A modified method of approximate particular solutions for solving partial differential equations. *Submitted*, 2016.

[85] Guangming Yao, C.S. Chen, Wen Li, and D. L. Young. The localized method of approximated particular solutions for near-singular two- and three-dimensional problems. *Computers and Mathematics with Applications*, 70:2883–2894, 2015.

[86] D. Young, M.H. Gu, and C. Fan. The time-marching method of fundamental solutions for wave equations. *Eng Anal Bound Elem,*, 33(12):1411–1425, 2008.

[87] X. Zhang, M. Chen, C.S. Chen, and Z. Li. Localized method of approximate particular solutions for solving unsteady Navier-Stokes problem. *Applied Mathematical Modelling*, 40(3):2265–2273, 2016.