The University of Southern Mississippi

The Aquila Digital Community

Dissertations

Fall 12-2016

Fast Method of Particular Solutions for Solving Partial Differential Equations

Anup Raja Lamichhane
University of Southern Mississippi

Recommended Citation

Lamichhane, Anup Raja, "Fast Method of Particular Solutions for Solving Partial Differential Equations" (2016). *Dissertations*. 876.
https://aquila.usm.edu/dissertations/876

FAST METHOD OF PARTICULAR SOLUTIONS
FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

by

Anup Raja Lamichhane

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. James Lambers, Committee Member
Associate Professor, Mathematics

Dr. Huiqing Zhu, Committee Member
Associate Professor, Mathematics

Dr. Zhaoxian Zhou, Committee Member
Associate Professor, Computing

Dr. Karen S. Coats
Dean of the Graduate School

December 2016

Published by the Graduate School

THE UNIVERSITY OF
SOUTHERN
MISSISSIPPI®

ABSTRACT

FAST METHOD OF PARTICULAR SOLUTIONS
FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

by Anup Raja Lamichhane

December 2016

Method of particular solutions (MPS) has been implemented in many science and engineering problems but obtaining the closed-form particular solutions, the selection of the good shape parameter for various radial basis functions (RBFs) and simulation of the large-scale problems are some of the challenges which need to be overcome. In this dissertation, we have used several techniques to overcome such challenges.

The closed-form particular solutions for the Matérn and Gaussian RBFs were not known yet. With the help of the symbolic computational tools, we have derived the closed-form particular solutions of the Matérn and Gaussian RBFs for the Laplace and biharmonic operators in 2D and 3D. These derived particular solutions play an important role in solving inhomogeneous problems using MPS and boundary methods such as boundary element methods or boundary meshless methods.

In this dissertation, to select the good shape parameter, various existing variable shape parameter strategies and some well-known global optimization algorithms have also been applied. These good shape parameters provide high accurate solutions in many RBFs collocation methods.

Fast method of particular solutions (FMPS) has been developed for the simulation of the large-scale problems. FMPS is based on the global version of the MPS. In this method, partial differential equations are discretized by the usual MPS and the determination of the unknown coefficients is accelerated using a fast technique. Numerical results confirm the efficiency of the proposed technique for the PDEs with a large number of computational points in both two and three dimensions. We have also solved the time fractional diffusion equations by using MPS and FMPS.

ACKNOWLEDGMENTS

I owe a deep sense of gratitude to my advisor, Prof. CS Chen, for encouraging and assisting me during this effort. I am forever indebted to him. I am also indebted to my family who inspire me to complete this work. I am very thankful to Prof. DL Young who assisted me to obtain financial support from the Department of Civil Engineering, National Taiwan University (NTU) to carry out the research work during summer 2015 at NTU.

I would like to thank Prof. A Karageorghis for his help to proofread the final version of the paper entitled “Particular solutions of Laplace and bi-harmonic operators using Matérn radial basis function”. The proofread provided by Daniel Watson for the paper entitled “The closed-form particular solutions for Laplace and biharmonic operators using a Gaussian function” is also greatly appreciated. I would also like to acknowledge my Professors, committee members and referees of all my papers for their constructive comments and suggestions. At last I would also like to thank all of my friends for their constant support and guidance.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF ILLUSTRATIONS

Figure

LIST OF ABBREVIATIONS

BPM	-	Boundary particle method
DRBF	-	Domain-type radial basis function
DRM	-	Dual reciprocity method
EFG	-	Element free Galerkin method
FDM	-	Finite difference method
FEM	-	Finite element method
FMPS	-	Fast method of particular solutions
FSM	-	Fast summation method
FVM	-	Finite volume method
GA	-	Genetic algorithm
GE	-	Gaussian elimination
GMRES	-	Generalized minimal residual method
IMQ	-	Inverse multiquadrics
LMPS	-	Localized method of particular solutions
LOOCV	-	Leave one out cross validation
LTBPM	-	Laplace transformed boundary particle method
MAE	-	Maximum absolute error
MFS	-	Method of fundamental solutions
MFS-MPS	-	Method of fundamental solutions and method of particular solutions
MLPG	-	Meshless local Petrov-Galerkin method
MLS	-	Moving least square
MPS	-	Method of particular solutions
MQ	-	Multiquadrics
NILT	-	Numerical inverse Laplace transform
PDEs	-	Partial differential equations
PS	-	Pattern search
PS-RBFs	-	Polyharmonic splines radial basis functions
RAE	-	Relative average error
RBFs	-	Radial basis functions
RKPM	-	Reproducing kernel particle method
RMSE	-	Root mean square error
SA	-	Simulated annealing
SPH	-	Smooth particle hydrodynamics

Chapter 1

Introduction

Various partial differential equations (PDEs) are used to describe physical phenomena. It is very difficult to obtain the analytical solution of most PDEs, so many numerical methods have been developed to find the approximate solutions of PDEs [62]. Generally, all of these methods can be categorized into two types, namely, mesh and meshless (meshfree) methods. In the book, "MESHFREE METHODS-Moving Beyond the Finite Element Method [66]", the author G. R. Liu described mesh and meshless method as

"Any of the open spaces or interstices between the strands of a net that is formed by connecting points in a predefined manner. In FDM, the meshes used are also often called grids; in the FVM, the meshes are called volumes or cells; and in FEM, the meshes are called elements [66]."

"The meshfree method is a method used to establish system of algebraic equations for the whole problem domain without the use of a predefined mesh for the domain discretization [66]."

The most popular and well-established numerical methods such as finite element method (FEM) [66, 69, 114], finite difference method (FDM) [56, 66], and finite volume method (FVM) [61, 66] are mesh based methods, while meshless methods, smooth particle hydrodynamics (SPH) [66, 68], element free Galerkin method (EFG) [10, 66], meshless local Petrov-Galerkin method (MLPG) [8], reproducing kernel particle method (RKPM) [71], and radial basis function (RBFs) collocation methods [6, 20–25, 33, 53, 58–60, 62] etc. are still under rapid development. Most of the meshless methods including RBFs collocation methods are easy to implement, efficient, and truly meshfree. For details of these and other meshless methods, we refer to [26, 62, 66, 67].

There are several RBFs collocation methods including Kansa method [53, 62], method of particular solutions (MPS) [23, 24, 60], and method of fundamental solutions (MFS) coupled with RBFs method such as MFS-MPS [21, 22, 33, 62]. All of these RBFs collocation methods use the radial basis functions (RBFs) to solve the PDEs. These RBFs can be formally defined as following:

Definition 1.0.1 ([35])**.** A function $\varphi : \mathbb{R}^n \to \mathbb{R}$ is called *radial* provided there exists a univariate function $\phi : [0,\infty) \to \mathbb{R}$ such that

$$\varphi(\mathbf{x}) = \phi(r), \text{ where } r = \|\mathbf{x}\|,$$

and $\|.\|$ is some norm on $\mathbb{R}^n$- usually the Euclidean norm.

In the Table 1.1, we can see the list of most commonly used RBFs.

Table 1.1: List of the most commonly used RBFs, where $c > 0$ is known as shape parameter, $n \in \mathbb{Z}^+$ and K_n is the modified Bessel function of the second kind of order n.

RBFs	$\phi(r)$
Polyharmonic splines in 2D	$r^{2n}\log(r)$
Polyharmonic splines in 2D and 3D	r^{2n-1}
Multiquadrics (MQ)	$\sqrt{r^2+c^2}$
Inverse multiquadrics (IMQ)	$\dfrac{1}{\sqrt{r^2+c^2}}$
Gaussian	$\exp(-cr^2)$
Matérn	$\phi(r) = (cr)^n K_n(cr)$

During the past two decades, RBFs have been widely applied for solving various PDEs. In 1990, Kansa [53] proposed the so-called RBFs collocation method which is also known as Kansa method for solving computational fluid dynamic problems. One of the attractions of the Kansa method is its simplicity for solving problems in high dimensions and complex geometries. Due to the popularity of the Kansa method, several other RBFs collocation methods have been proposed in the RBFs literature. Among them, MPS [23, 24] is another effective RBFs collocation method which uses the particular solutions of the given RBFs.

In recent years, the MPS [23, 24] has been developed as an alternative to the Kansa method. Each method has its own limitations. A comparison of these two methods has been given in [108]. In the next section, we present a brief review of the MPS.

1.1 Method of particular solutions (MPS)

Let us consider the following boundary value problem

$$\mathcal{L}u(\boldsymbol{x}) = f(\boldsymbol{x}), \ \boldsymbol{x} \in \Omega, \tag{1.1}$$

$$\mathcal{B}u(\boldsymbol{x}) = g(\boldsymbol{x}), \ \boldsymbol{x} \in \partial\Omega, \tag{1.2}$$

where $\mathcal{L}$ is a differential operator, $\mathcal{B}$ is a boundary differential operator, $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are known functions, Ω and $\partial\Omega$ are the interior and boundary of the computational domain, respectively. Some of the computational domains that we have used for the numerical tests are depicted in the Figures 2.3 and 3.4.

Suppose $\{\boldsymbol{x}_i\}_{i=1}^{N}$ are the interpolation points containing n_i interior points in Ω and n_b boundary points on $\partial\Omega$; i.e., $N = n_i + n_b$. Let ϕ be a given radial basis function.

1.1.1 Global method

The method of particular solution (MPS) [24] is used for the discretization of the (1.1) and (1.2). By MPS, instead of approximating the field variable u by a linear superposition of the RBFs, we assume the solution to (1.1) and (1.2) can be approximated by a linear superposition of the corresponding particular solutions of the given RBFs such as

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \Phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|), \tag{1.3}$$

where $\|\cdot\|$ is the Euclidean norm, $\{\alpha_j\}$ are the undetermined coefficients, and

$$\mathcal{L}\Phi = \phi. \tag{1.4}$$

Note that, if $\mathcal{L}$ is a more general differential operator such as

$$\mathcal{L} = k\Delta + l(\boldsymbol{x})\frac{\partial}{\partial x} + m(\boldsymbol{x})\frac{\partial}{\partial y} + n(\boldsymbol{x}), \tag{1.5}$$

where k is a constant, $l(\boldsymbol{x})$, $m(\boldsymbol{x})$ and $n(\boldsymbol{x})$ are variable coefficients, then we approximate u by a linear superposition of the particular solutions of the RBFs obtained from the Laplace or biharmonic operator. i.e.,

$$\Delta\Phi = \phi.$$

By the collocation method, from (1.1) and (1.2), we have

$$\sum_{j=1}^{N} \alpha_j \phi\left(\|\boldsymbol{x_i} - \boldsymbol{x}_j\|\right) = f(\boldsymbol{x_i}), \quad 1 \leq i \leq n_i, \tag{1.6}$$

$$\sum_{j=1}^{N} \alpha_j \Phi\left(\|\boldsymbol{x_i} - \boldsymbol{x}_j\|\right) = g(\boldsymbol{x_i}), \quad n_i + 1 \leq i \leq N. \tag{1.7}$$

From (1.6) and (1.7), we can formulate a linear system of equations

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{F}, \tag{1.8}$$

where

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{\Phi} \end{bmatrix},$$

$$\begin{aligned}
\boldsymbol{\phi} &= \left[\phi\left(\|\boldsymbol{x_i} - \boldsymbol{x}_j\|\right)\right]_{ij}, \quad 1 \le i \le n_i,\ 1 \le j \le N, \\
\boldsymbol{\Phi} &= \left[\Phi\left(\|\boldsymbol{x_k} - \boldsymbol{x}_j\|\right)\right]_{kj}, \quad n_i + 1 \le k \le N,\ 1 \le j \le N, \\
\boldsymbol{\alpha} &= [\alpha_1\ \alpha_2\ \cdots\ \alpha_N]^T, \\
\mathbf{F} &= [f(\boldsymbol{x}_1)\ \cdots\ f(\boldsymbol{x}_{n_i})\ g(\boldsymbol{x}_{n_i+1})\ \cdots\ g(\boldsymbol{x}_N)]^T.
\end{aligned}$$

For a more general form of the PDEs which involves a general differential operator $\mathcal{L}$ as in (1.5), we have [23]

$$\sum_{j=1}^{N} \alpha_j (k\phi\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right) + l(\boldsymbol{x})\Phi_x\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right) + m(\boldsymbol{x})\Phi_y\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right) + n(\boldsymbol{x})\Phi\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right)) = f(\boldsymbol{x}). \tag{1.9}$$

Once $\{\alpha_j\}$ are known, the approximate solution $\hat{u}$ can be evaluated at any point in the domain using (1.3).

1.1.2 Localized method

The localized version of the MPS is known as the localized method of particular solutions (LMPS) [106, 107]. By LMPS, the solution to (1.1) and (1.2) can be approximated by a localized formulation

$$u(\mathbf{x}_s) \approx \hat{u}(\mathbf{x}_s) = \sum_{k=1}^{n} \alpha_k^{[s]} \Phi\left(\|\boldsymbol{x}_s - \boldsymbol{x}_k^{[s]}\|\right), \tag{1.10}$$

where n is the number of nearest neighboring points $\{\mathbf{x}_k^{[s]}\}_{k=1}^{n}$ of the corresponding collocation points $\mathbf{x}_s$, $\{\alpha_k^{[s]}\}_{k=1}^{n}$ are the unknown coefficients to be determined, and Φ is as in (1.4).

From equation (1.10), using the n neighboring collocation points, we obtain the following linear system

$$\hat{\mathbf{u}}^{[s]} = \boldsymbol{\Phi}\boldsymbol{\alpha}^{[s]}, \tag{1.11}$$

where

$$\begin{aligned}
\boldsymbol{\alpha}^{[s]} &= \left[\alpha_1^{[s]}, \alpha_2^{[s]}, \cdots, \alpha_n^{[s]}\right]^T, \\
\hat{\mathbf{u}}^{[s]} &= \left[\hat{u}(\mathbf{x}_1^{[s]}), \hat{u}(\mathbf{x}_2^{[s]}), \cdots, \hat{u}(\mathbf{x}_n^{[s]})\right]^T, \\
\boldsymbol{\Phi} &= \left[\Phi\left(\|\mathbf{x}_j^{[s]} - \mathbf{x}_k^{[s]}\|\right)\right]_{jk}, \quad 1 \le j \le n,\ 1 \le k \le n.
\end{aligned}$$

It can be proved that $\boldsymbol{\Phi}$ is non-singular such that the unknown coefficients in (1.11) can be written as

$$\boldsymbol{\alpha}^{[s]} = \boldsymbol{\Phi}^{-1}\hat{\mathbf{u}}^{[s]}.$$

So, the approximate solution $\hat{\mathbf{u}}^{[s]}$ in (1.10) can be rewritten in terms of the given nodal values $\hat{\mathbf{u}}^{[s]}$ as

$$\begin{aligned}
u(\mathbf{x}_s) \approx \hat{u}(\mathbf{x}_s) &= \sum_{k=1}^{n} \alpha_k^{[s]} \Phi\left(\|\boldsymbol{x}_s - \boldsymbol{x}_k^{[s]}\|\right) \\
&= \hat{\boldsymbol{\Phi}}^{[s]} \boldsymbol{\alpha}^{[s]} \\
&= \hat{\boldsymbol{\Phi}}^{[s]} \boldsymbol{\Phi}^{-1} \hat{\mathbf{u}}^{[s]} \\
&= \boldsymbol{\Psi}^{[s]} \hat{\mathbf{u}}^{[s]},
\end{aligned} \tag{1.12}$$

where

$$\hat{\boldsymbol{\Phi}}^{[s]} = \left[\Phi\left(\|\boldsymbol{x}_s - \boldsymbol{x}_1^{[s]}\|\right), \Phi\left(\|\boldsymbol{x}_s - \boldsymbol{x}_2^{[s]}\|\right), \cdots, \Phi\left(\|\boldsymbol{x}_s - \boldsymbol{x}_n^{[s]}\|\right)\right],$$

and

$$\boldsymbol{\Psi}^{[s]} = \hat{\boldsymbol{\Phi}}^{[s]} \boldsymbol{\Phi}^{-1}.$$

Note that we have $n << N$. So if we reformulate (1.12) in terms of $\hat{u}(\mathbf{x}_j)$ at all collocation points, it has

$$\hat{\mathbf{u}}^{[s]} = \boldsymbol{\Psi}\boldsymbol{u}, \tag{1.13}$$

where $\boldsymbol{\Psi}$ is a $N \times N$ sparse matrix only having $N \times n$ nonzero elements. Substituting (1.13) into (1.1) and (1.2) yields

$$\begin{bmatrix} \mathcal{L}\boldsymbol{\Psi} \\ \mathcal{B}\boldsymbol{\Psi} \end{bmatrix} \hat{\boldsymbol{u}} = \boldsymbol{F}. \tag{1.14}$$

Then solving the above linear system, we can determine the required approximate solution.

For more information of numerical implementation using the MPS and LMPS, we refer readers to [23, 24, 107].

MPS and LMPS have been successfully implemented in many science and engineering problems such as Navier-Stokes equations [13, 14], Stokes flow problems [15], incompressible viscous flow field problems [65], linear elasticity equations [12], time fractional diffusion equations [40, 105], anisotropic elliptic problems [113], wave equations [57], inverse problem of nonhomogeneous convection-diffusion equations [51], diffusion equation with non-classical boundary [2] and nonhomogeneous Cauchy problems of elliptic PDEs [64] and so on. For details of these and other applications, we refer to [13, 14, 40, 51, 63–65, 101, 102, 105, 113].

1.2 Synopsis

Although, MPS has been implemented for solving various PDEs, obtaining the closed-form particular solutions, the selection of the good shape parameter for various RBFs, and simulations of the problems which involve large number of interpolation points have been always a daunting task. In this dissertation, several techniques have been proposed to address these issues. A brief outline of the dissertation is as follows:

- **Closed-form particular solutions:** One of the key procedures in the implementation of the MPS is to obtain the closed-form expression for the particular solutions of the corresponding RBFs. Among various types of meshless methods using RBFs, MPS, LMPS, and MFS-MPS are the meshless collocation methods which require the use of closed-form particular solutions as the basis functions in the solution process. Hence, the closed-form particular solutions become the core of these particular solutions based meshless methods. The importance of the closed-form particular solutions to the above mentioned RBF-based meshless methods is analogous to the fundamental solutions to the boundary element methods. The derivation of the particular solutions for the well-known RBFs has already been known [28, 43, 80, 97–99]. In the past, among most commonly used RBFs, only MQ and IMQ were used in the MPS due to availability of the particular solutions. Recently, the closed-form particular solutions using the Gaussian [58] and Matérn [59] have been obtained for different operators and implemented numerically to solve the boundary value problems using the MPS and LMPS. In Chapter 2, we present the closed-form particular solutions for the Laplace and biharmonic operators using the Matérn and Gaussian RBFs [58, 59]. The derived particular solutions are implemented numerically to solve boundary value problems using the MPS and LMPS in Sections 2.3 and 3.3.

- **Selection of the good shape parameter:** The accuracy and stability of the solution of the most of the RBFs collocation methods depend on the choice of the shape parameter of the RBFs. In the RBFs literature, extensive research has been done on choosing good shape parameter for better accuracy and stability of the solution. Many researchers have proposed various strategies to select the shape parameter. In this dissertation, we have used well-known Leave-one-out cross validation (LOOCV) [36, 86] to find the good constant shape parameter for the Gaussian RBFs and its corresponding closed form particular solutions in Section 3.3.

  Previously, the research was only focused on the constant shape parameter but nowadays several research are focused on the variable shape parameter. In fact, it has been shown in several works, including [16, 49, 52–54], that variable shape parameter produces more accurate results than if a constant shape parameter is used for the RBFs collocation methods like Kansa method.

  As far as the knowledge of the authors, variable shape parameter has not been used yet in the MPS, so in this research work we have implemented some popular variable shape parameter strategies in the MPS. Also, we propose some new strategies to obtain the good shape parameter and implement in the MPS. Although, the numerical examples given in the Section 3.4 only validate the proposed strategies for the MPS, it can be easily implemented in any other RBFs collocation methods. In Section 3.2, we discuss about the well-known variable shape parameter strategies and also we introduce some of the new strategies. Numerical examples to validate the existing and new strategies for the MPS are presented in Section 3.4.

- **Simulations of the large scale problems:** As we have discussed earlier, MPS has been applied for solving various science and engineering problems. These kind of challenging problems involve a large number of interpolation points. The high computational cost using traditional solvers has become an issue. In this research work, we pay special attention on how to develop a fast algorithm to alleviate the issue of high cost for solving large-scale problems using the MPS. Consequently, in Chapter 4, we present the fast method of particular solutions (FMPS) [60] where we propose to couple the MPS with fast summation method (FSM) [38] to reduce the computational time by multiplying a matrix and a vector in each step inside the iterative method. This FSM is based on the Chebyshev interpolation [38]. To demonstrate the efficiency of the proposed method, two numerical examples in 2D and 3D are given in Section 4.3.

- **Application of the FMPS:** In Chapter 5, we implement FMPS [60] for solving time fractional diffusion equations [19, 72, 81, 84, 110]. We use Laplace transform techniques [41, 78] to transform the time dependent problem into the time independent problem. Then, we implement MPS, FMPS to approximate the solution of the time independent problem in the Laplace space. Finally, we use Talbot algorithm [1] which is a numerical inverse Laplace transform (NILT) algorithm [1, 93] to retrieve the numerical solutions of the time fractional diffusion equations from the Laplace space. Section 5.1 introduces the numerical method for solving time fractional diffusion equations and numerical results are presented in Section 5.2.

Chapter 2

The closed-form particular solutions for Laplace and bi-harmonic operators using radial basis functions

Particular solutions play a critical role in solving inhomogeneous problems using boundary methods such as boundary element methods or boundary meshless methods. In the literature of the boundary element method, the Dual Reciprocity Method (DRM) [83] has been developed to avoid the domain integration. To successfully implement the DRM, a closed-form particular solution is essential. A great deal of effort has been devoted to derive the closed-form particular solution using RBFs [28, 80, 99]. In the RBFs literature, the closed-form particular solutions using MQ, polyharmonic splines, and compactly supported RBFs have been derived for the above mentioned particular solutions based meshless collocation methods [28, 43, 80, 99]. The list of the particular solutions of the most commonly used RBFs can be found in the Appendix A.

In this chapter, we present closed-form particular solutions of Matérn [59] and Gaussian [58] RBFs for the Laplace and biharmonic operator in 2D and 3D. These derived particular solutions are also essential for the implementation of the MPS, LMPS, MFS-MPS etc. for solving various types of PDEs. The role of the particular solutions in these methods is similar to that of the fundamental solutions in boundary element methods. The task of obtaining closed-form particular solutions is often non-trivial. During the past two decades, significant progress has been made in deriving closed-form particular solutions using RBFs [28, 43, 58, 80, 99]. Once the particular solutions of the given linear elliptic PDEs are available, these methods can be easily implemented. Numerical examples in 2D and 3D are given to demonstrate the effectiveness of the derived particular solutions. All of the derivations presented in this chapter are published in [58, 59].

2.1 Matérn radial basis functions

Matérn RBFs [11, 73, 94, 103] are a family of functions that are defined based on the modified Bessel functions of the second kind of different orders. Consider the Matérn RBFs

$$\phi_\nu(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}(cr)^\nu K_\nu(cr), \tag{2.1}$$

where K_ν is the modified Bessel function of the second kind (sometimes also called the modified Bessel function of the third kind, or MacDonald's function) of order $\nu > 0$ and $c > 0$. If ν is of the form $n+1/2$ where n is a nonnegative integer, then (2.1) reduces to the product of a polynomial of degree n in (cr) and $\exp(-cr)$; i.e.,

$$\phi_{n+1/2}(r) = \frac{(cr)^n \exp(-cr)}{(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!(2cr)^k}, \tag{2.2}$$

where

$$n!! = \begin{cases} n\cdot(n-2)...5\cdot 3\cdot 1, & n: \text{ positive odd integer,} \\ n\cdot(n-2)...6\cdot 4\cdot 2, & n: \text{ positive even integer,} \\ 1, & n = -1, 0. \end{cases}$$

A list of Matérn functions for various ν are given as follows:

$$\begin{aligned} &\nu = 1/2,\ \phi(r) = \exp(-cr), \\ &\nu = 1, \quad \phi(r) = crK_1(cr), \\ &\nu = 3/2,\ \phi(r) = (1+cr)\exp(-cr), \\ &\nu = 5/2,\ \phi(r) = (1+cr+c^2r^2/3)\exp(-cr). \end{aligned}$$

In the early statistical literature [73, 94], these functions were used as correlation functions and there are still many authors using them [46]. Lately, the Matérn correlation functions have attracted attention in the machine learning community [85, 91]. It has been sometimes called the Basset family, the Bessel model, the generalized Markov model, the Whittle-Matérn class, the Whittle model and the von Karman class [46]. In RBFs literature [11, 35, 79], Matérn RBFs are described as positive definite RBFs which have been considered as alternatives for other RBFs such as Gaussians, MQ, and IMQ due to the high condition number of those basis functions. Matérn RBFs were first implemented as basis functions in the context of the Kansa method [79]. Due to the unavailability of the closed-form particular solutions of the Matérn RBFs, it was not implemented as a basis function in the MPS.

We now focus on the derivation of the particular solutions using Matérn RBFs for the Laplace and biharmonic differential operators in 2D and the Laplace operator in 3D. Since the particular solution is not unique, it is important to go through the de-singularization process (e.g., Eq. (2.28)) to make sure the obtained particular solution is non-singular.

2.1.1 Particular Solutions in 2D

In this section we will derive particular solutions of the Matérn RBFs $\phi(r) = (cr)^n K_n(cr)$, $c > 0$, $n \in \mathbb{Z}^+$ for different types of differential operators in 2D. The following identities [3] related to the modified Bessel function of the second kind are useful for the derivation of particular solutions in the forthcoming sections:

$$K_{n+1}(cr) = K_{n-1}(cr) + \frac{2n}{cr}K_n(cr), \tag{2.3}$$

$$\frac{d}{dr}((cr)^n K_n(cr)) = -c(cr)^n K_{n-1}(cr), \tag{2.4}$$

$$\frac{d}{dr}K_0(cr) = -cK_1(cr). \tag{2.5}$$

Note that (2.4) can be rewritten into integral form as follows

$$\int (cr)^n K_{n-1}(cr)dr = -\frac{(cr)^n K_n(cr)}{c}. \tag{2.6}$$

Similarly, (2.5) is equivalent to the form

$$\int K_1(cr)dr = \frac{-K_0(cr)}{c}. \tag{2.7}$$

Before the derivation of particular solutions for various differential operators, we establish the following two lemmas.

Lemma 2.1.1. Let

$$I_p = \int (cr)^{n-p} K_{n-(p-1)}(cr)dr, \tag{2.8}$$

and

$$F_s = (cr)^{n-s} K_{n-s}(cr), \tag{2.9}$$

where $p = 0, 1, 2, ..., n-1$ and $s = 0, 1, 2, ..., n$. Then,

$$I_p = \frac{-1}{c}F_p + 2(n-p)I_{p+1}. \tag{2.10}$$

Proof. From (2.3), we have

$$K_{n-(p-1)}(cr) = K_{(n-p)-1}(cr) + \frac{2(n-p)}{cr}K_{n-p}(cr). \tag{2.11}$$

Then, from (2.11) and (2.6), we have

$$\begin{aligned} I_p &= \int (cr)^{n-p} K_{n-(p-1)}(cr)dr \\ &= \int (cr)^{n-p} K_{(n-p)-1}(cr)dr + 2(n-p)\int (cr)^{n-(p+1)} K_{n-p}(cr)dr \\ &= \frac{-1}{c}(cr)^{n-p} K_{n-p}(cr) + 2(n-p)\int (cr)^{n-(p+1)} K_{n-p}(cr)dr \\ &= \frac{-1}{c}F_p + 2(n-p)I_{p+1}. \end{aligned}$$

□

Lemma 2.1.2. Let I_p and F_s be denoted as in (2.8) and (2.9), respectively. Then,

$$I_{n-1} = \frac{-1}{c}\left(F_{n-1} - F_n\right). \tag{2.12}$$

Proof. Let $p = n-1$. From (2.8), we have

$$I_{n-1} = \int (cr) K_2(cr) dr. \tag{2.13}$$

From (2.3), let $n = 1$, we have

$$K_2(cr) = K_0(cr) + \frac{2}{cr} K_1(cr). \tag{2.14}$$

Substituting (2.14) into (2.13) and then applying (2.4) and (2.5), we obtain

$$\begin{aligned} I_{n-1} &= \frac{-1}{c}(cr)K_1(cr) - \frac{2}{c}K_0(cr) \\ &= \frac{-1}{c}\left(F_{n-1} - 2F_n\right). \end{aligned}$$

□

Corollary 2.1.3. Let

$$\mathfrak{I}_n = \frac{-1}{c}\int (cr)^n K_{n+1}(cr) dr, \tag{2.15}$$

and $F_j = (cr)^{n-j} K_{n-j}(cr), j = 0, 1, \cdots, n$. It can be shown that

$$\mathfrak{I}_n = \frac{n!}{c^2} \sum_{j=0}^{n} \frac{2^j}{(n-j)!} F_j. \tag{2.16}$$

Proof. For $n = 1$, from the proof of the last Lemma, (2.16) is true. Next, we assume that (2.16) is true for $n = k$, i.e.,

$$\mathfrak{I}_k = \frac{k!}{c^2} \sum_{j=0}^{k} \frac{2^j}{(k-j)!} F_k. \tag{2.17}$$

Then, for $n = k+1$ and from (2.3), we have

$$K_{k+2}(cr) = K_k(cr) + \frac{2(k+1)}{cr} K_{k+1}(cr). \tag{2.18}$$

Multiplying both sides of (2.18) by $-(cr)^{k+1}/c$ and then integrating, we obtain

$$\frac{-1}{c}\int (cr)^{k+1} K_{k+2}(cr) dr = \frac{-1}{c}\left(\int (cr)^{k+1} K_k(cr) + 2(k+1)\int (cr)^k K_{k+1}(cr) dr\right). \tag{2.19}$$

From (2.4) and (2.15), we obtain

$$\mathfrak{I}_{k+1} = \frac{1}{c^2}(cr)^{k+1} K_{k+1}(cr) - \frac{2(k+1)}{c}\int (cr)^k K_{k+1}(cr) dr. \tag{2.20}$$

Since the above relation holds for $n = k$, from (2.15) and (2.17) we have

$$\mathfrak{I}_k = \frac{-1}{c}\int (cr)^k K_{k+1}(cr)dr = \frac{k!}{c^2}\sum_{j=0}^{k}\frac{2^j}{(k-j)!}F_j. \tag{2.21}$$

Substituting (2.21) into (2.20), we obtain

$$\begin{aligned}\mathfrak{I}_{k+1} &= \frac{1}{c^2}F_0 + \frac{(k+1)!}{c^2}\sum_{j=1}^{k+1}\frac{2^j}{(k+1-j)!}F_j\\ &= \frac{(k+1)!}{c^2}\sum_{j=0}^{k+1}\frac{2^j}{(k+1-j)!}F_j.\end{aligned}$$

This shows that the relation holds for $n = k+1$. By induction, the relation holds for all $n \in \mathbb{Z}^+$. □

Particular solution of the Laplacian

Theorem 2.1.4. Let $\phi(r) = (cr)^n K_n(cr)$, and $\Delta\Phi(r) = \phi(r)$ in $2D$. Then,

$$\Phi(r) = \frac{n!}{c^2}\left(\sum_{j=0}^{n}\frac{2^j}{(n-j)!}(cr)^{n-j}K_{n-j}(cr) + 2^n\ln(cr)\right), r \neq 0, \tag{2.22}$$

and

$$\Phi(0) = \frac{n!2^n}{c^2}\left(\ln(2) - \gamma + \sum_{j=0}^{n-1}\frac{1}{2(n-j)}\right) \tag{2.23}$$

where $\gamma \simeq 0.5772156649$ is the Euler number.

Proof. Suppose

$$\Delta\Phi = (cr)^n K_n(cr), \tag{2.24}$$

where K_n is the modified Bessel function of the second kind of order n. By using the Laplace operator in terms of radial distance, we can rewrite (2.24) as

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{d\Phi}{dr}\right) = (cr)^n K_n(cr). \tag{2.25}$$

By direct integration and (2.6), we have

$$\begin{aligned}r\frac{d\Phi}{dr} &= \int c^n r^{n+1}K_n(cr)dr\\ &= \frac{1}{c}\int (cr)^{n+1}K_n(cr)dr\\ &= \frac{-1}{c^2}(cr)^{n+1}K_{n+1}(r) + c_1.\end{aligned} \tag{2.26}$$

It follows that

$$\begin{aligned}\Phi(r) &= -\int \left(c^{n-1}r^nK_{n+1}(cr)+\frac{c_1}{r}\right)dr\\ &= \frac{-1}{c}\int (cr)^nK_{n+1}(cr)dr + c_1\ln(r)+c_2\\ &= \mathcal{I}_n + c_1\ln(r)+c_2,\end{aligned}$$

where $\mathcal{I}_n$ is defined in (2.15). From Corollary 2.1.3, we obtain

$$\begin{aligned}\Phi(r) &= \frac{n!}{c^2}\sum_{j=0}^{n}\frac{2^j}{(n-j)!}(cr)^{n-j}K_{n-j}(cr)+c_1\ln(r)+c_2\\ &= \frac{2^nn!}{c^2}K_0(cr)+\frac{n!}{c^2}\sum_{j=0}^{n-1}\frac{2^j}{(n-j)!}(cr)^{n-j}K_{n-j}(cr)+c_1\ln(r)+c_2. \qquad (2.27)\end{aligned}$$

The first term in (2.27) contains the singular term $K_0(cr)$ at $r=0$. The singularity of $\Phi(r)$ at $r=0$ can be removed by properly choosing the integration constant c_1. Note that the series expansion of $K_0(cr)$ is known as

$$K_0(cr) = -\ln\left(\frac{c}{2}\right)-\ln r-\gamma+O(r^2).$$

Hence, the singularity of K_0 can be canceled by setting

$$c_1 = \frac{2^nn!}{c^2}. \qquad (2.28)$$

The integration constant c_2 in (2.27) can be chosen arbitrary. For convenience, we set

$$c_2 = \frac{2^nn!}{c^2}\ln(c).$$

Hence, $\Phi(r)$ in (2.22) is proved.

Furthermore, it is known that [3]

$$\lim_{r\to 0}(cr)^nK_n(cr) = 2^{n-1}(n-1)!.$$

It follows that

$$\begin{aligned}\lim_{r\to 0}\Phi(r) &= \lim_{r\to 0}\frac{n!}{c^2}\left(\sum_{j=0}^{n}\frac{2^j}{(n-j)!}(cr)^{n-j}K_{n-j}(cr)+2^n\ln(cr)\right)\\ &= \frac{n!}{c^2}\sum_{j=0}^{n-1}\frac{2^{j+n-j-1}}{(n-j)!}(n-j-1)!+\frac{2^nn!}{c^2}(\ln(2)-\gamma)\\ &= \frac{n!2^n}{c^2}\left(\sum_{j=0}^{n-1}\frac{1}{2(n-j)}+\ln(2)-\gamma\right),\end{aligned}$$

where γ is the Euler number. □

To solve the problem with Neumann boundary conditions, we need to find $1/r(d\Phi/dr)$. From (2.26), we have

$$r\frac{d\Phi}{dr} = -c^{n-1}r^{n+1}K_{n+1}(cr) + c_1,$$

where c_1 is given in (2.28). It follows that

$$\frac{1}{r}\frac{d\Phi}{dr} = -(cr)^{n-1}K_{n+1}(cr) + \frac{2^n n!}{(cr)^2}.$$

Furthermore,

$$\lim_{r\to 0}\frac{1}{r}\frac{d\Phi}{dr} = \frac{2^{n-2}\Gamma(n+1)}{n} = 2^{n-2}(n-1)!.$$

Particular solution of the Biharmonic operator

Theorem 2.1.5. If $\phi(r) = (cr)^n K_n(cr)$, and $\Delta^2\Phi(r) = \phi(r)$ in $2D$, then

$$\Phi(r) = \frac{n!}{c^4}\left(\sum_{j=0}^{n}\frac{2^j(j+1)}{(n-j)!}(cr)^{n-j}K_{n-j}(cr) + 2^{n-2}(cr)^2(\ln(cr)-1) + 2^n(n+1)\ln(cr)\right), \tag{2.29}$$

for $r \neq 0$, and

$$\Phi(0) = \frac{n!2^n}{c^4}\left(\sum_{j=0}^{n-1}\frac{j+1}{2(n-j)} + (n+1)(\ln(2)-\gamma)\right). \tag{2.30}$$

Proof. Let

$$\Delta^2\Phi = \Delta(\Delta\Phi) = (cr)^n K_n(cr).$$

Similar to the derivation shown in the last theorem, we obtain

$$\Phi = \frac{1}{c^4}\sum_{j=0}^{n}\sum_{i=0}^{n-j}2^{j+i}\frac{n!}{(n-j-i)!}(cr)^{n-j-i}K_{n-j-i}(cr) + \frac{2^n n!}{4c^2}r^2(\ln(cr)-1) + c_1\ln(r) + c_2.$$

Choosing $s = i + j$, we obtain

$$\frac{1}{c^4}\sum_{j=0}^{n}\sum_{s=j}^{n}2^s\frac{n!}{(n-s)!}(cr)^{n-s}K_{n-s}(cr) + \frac{2^n n!}{4c^2}r^2(\ln(cr)-1) + c_1\ln(r) + c_2.$$

We note that

$$\sum_{j=0}^{n}\sum_{s=j}^{n}a_s = \sum_{k=0}^{n}(k+1)a_k.$$

The above double sum can be reduced into a single sum which gives

$$\Phi = \frac{1}{c^4}\sum_{k=0}^{n}(k+1)2^k\frac{n!}{(n-k)!}(cr)^{n-k}K_{n-k}(cr) + \frac{2^n n!}{4c^2}r^2(\ln(cr)-1) + c_1\ln(r) + c_2.$$

Changing the dummy index k into j and selecting the integration constants as

$$c_1 = \frac{2^n(n+1)!}{c^4}, \quad c_2 = \frac{2^n(n+1)!}{c^4}\ln(c),$$

the required particular solution $\Phi(r)$ for $r \neq 0$ is

$$\Phi(r) = \frac{n!}{c^4}\left(\sum_{j=0}^{n}\frac{2^j(j+1)}{(n-j)!}(cr)^{n-j}K_{n-j}(cr) + 2^{n-2}(cr)^2(\ln(cr)-1) + 2^n(n+1)\ln(cr)\right). \tag{2.31}$$

To find the particular solution $\Phi(0)$, we can take the limit of $\Phi(r)$ at $r = 0$. After some algebraic manipulation, we obtain the required (2.30). □

Now, we will find $1/r(d\Phi/dr)$ for the Neumann boundary condition, which can be obtained by the relation (2.31), i.e.,

$$\frac{d\Phi}{dr} = \frac{1}{c^3}\sum_{j=0}^{n}\frac{n!2^j}{(n-j)!}(-1)(cr)^{n-j}K_{n-j+1}(cr) + \frac{2^n n!}{4c^2}r(2\ln(cr)-1) + \frac{2^n(n+1)!}{c^4 r},$$

$$\frac{1}{r}\frac{d\Phi}{dr} = \frac{1}{c^2}\sum_{j=0}^{n}\frac{n!2^j}{(n-j)!}(-1)(cr)^{n-j-1}K_{n-j+1}(cr) + \frac{2^n n!}{4c^2}(2\ln(cr)-1) + \frac{2^n(n+1)!}{c^4 r^2}.$$

It follows that

$$\lim_{r\to 0}\frac{1}{r}\frac{d\Phi}{dr} = \frac{2^{n-2}n!}{c^2}\left(-2\gamma + 2\ln(2) + \sum_{j=0}^{n-1}\frac{1}{(n-j)}\right).$$

2.1.2 Particular Solution for Laplacian in 3D

For the 3D case, a closed-form particular solution using the Matérn functions of integer order cannot be obtained. As a result, we consider the fractional order of Matérn function as shown in (2.2) as a basis function for the 3D case.

Theorem 2.1.6. Let

$$\phi_{n+1/2}(r) = \frac{\exp(-cr)(cr)^n}{(2n-1)!!}\sum_{k=0}^{n}\frac{(n+k)!}{k!(n-k)!(2cr)^k}.$$

Let Δ be the Laplacian in 3D. The particular solution Φ in the following expression

$$\Delta\Phi(r) = \phi_{n+1/2}(r),$$

is given by

$$\begin{aligned}\Phi(r) = &\frac{1}{c^2(2n-1)!!}\sum_{k=0}^{n}\frac{(n+k)!}{k!(n-k)!2^k}\left(\frac{\Gamma(n+3-k,cr)}{cr} - e^{-cr}(cr)^{n+1-k}\right.\\ &\left. -(n+1-k)\Gamma(n+1-k,cr) - \frac{\Gamma(n+3-k)}{cr}\right), \quad r \neq 0,\end{aligned} \tag{2.32}$$

where $n \in \mathbb{Z}^+$ and $\Gamma(a,cr)$ is an upper incomplete Gamma function [3, 7]

$$\Gamma(a,cr) = \int_{cr}^{\infty} t^{a-1} e^{-t} dt,$$

and

$$\Phi(0) = \frac{-1}{c^2(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!2^k}(n+1-k)\Gamma(n+1-k), \tag{2.33}$$

where Γ is the Gamma function [3].

Proof. Suppose

$$\Delta\Phi(r) = \frac{\exp(-cr)(cr)^n}{(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!(2cr)^k}.$$

Then, it follows that

$$\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr}\Phi(r) = \frac{\exp(-cr)(cr)^n}{(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!(2cr)^k}. \tag{2.34}$$

By integrating (2.34), we have

$$\begin{aligned} r^2\frac{d}{dr}\Phi(r) &= \frac{1}{c^2}\int \frac{\exp(-cr)(cr)^{n+2}}{(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!(2cr)^k} \\ &= \frac{1}{c^2(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!2^k} \int \exp(-cr)(cr)^{n+2-k} dr \\ &= \frac{1}{c^2(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!2^k} \frac{-\Gamma(n+3-k,cr)}{c} + c_1, \end{aligned}$$

where c_1 is an integration constant.

Repeating the above procedure and using MATHEMATICA, we have

$$\begin{aligned} \Phi(r) =& \frac{1}{c(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!2^k} \int \frac{-\Gamma(n+3-k,cr)}{(cr)^2} dr + \int \frac{c_1}{r^2} dr, \\ =& \frac{1}{c(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!2^k} \left(\frac{\Gamma(n+3-k,cr)}{c^2 r} - \frac{e^{-cr}(cr)^{n+1-k}}{c} \right. \\ & \left. -(n+1-k)\frac{\Gamma(n+1-k,cr)}{c} - \frac{c_1}{r} + c_2 \right), \end{aligned} \tag{2.35}$$

where c_2 is another arbitrary integration constant. Note that [3, 7]

$$\Gamma(n+3-k,cr) = \Gamma(n+3-k) + (cr)^{n-k}\left(\frac{(cr)^3}{k-n-3} + \frac{(cr)^4}{-k+n+4} + O(r^6) \right).$$

To cancel the singularity in (2.35), we can choose the integration constants c_1 and c_2 as follows

$$c_1 = \frac{1}{c(2n-1)!!} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!2^k} \frac{\Gamma(n+3-k)}{c^2}, \quad c_2 = 0.$$

Then the required particular solution for 3D for $r \neq 0$ is

$$\Phi(r) = \frac{1}{c^2(2n-1)!!}\sum_{k=0}^{n}\frac{(n+k)!}{k!(n-k)!2^k}\left(\frac{\Gamma(n+3-k,cr)}{cr} - e^{-cr}(cr)^{n+1-k} \right.$$
$$\left. -(n+1-k)\Gamma(n+1-k,cr) - \frac{\Gamma(n+3-k)}{cr}\right). \tag{2.36}$$

To find the particular solution $\Phi(0)$, we can take the limit of $\Phi(r)$ at $r = 0$. After some algebraic manipulation, we obtain the required (2.33). □

Now, we can derive $1/r(d\Phi/dr)$ which is needed in the numerical implementation of the Neumann boundary condition as follows:

$$r^2\frac{d\Phi}{dr} = \frac{1}{c^3}\frac{1}{(2n-1)!!}\sum_{k=0}^{n}\frac{(n+k)!}{k!(n-k)!2^k}\left(-\Gamma(n+3-k,cr)+\Gamma(n+3-k)\right),$$
$$\frac{1}{r}\frac{d\Phi}{dr} = \frac{1}{(2n-1)!!}\sum_{k=0}^{n}\frac{(n+k)!}{k!(n-k)!2^k}\left(-\frac{\Gamma(n+3-k,cr)}{(cr)^3}+\frac{\Gamma(n+3-k)}{(cr)^3}\right).$$

We know that [7]

$$\frac{-\Gamma(n+3-k,cr)}{(cr)^3}$$
$$= \left(\frac{-\Gamma(n+3-k)}{(cr)^3} + O(cr)^5\right) + (cr)^{n+3-k}\left(\frac{1}{(n+3-k)(cr)^3} - \frac{1}{(n+4-k)(cr)^2}\right.$$
$$\left. + \frac{1}{(2(n+3-k)+4)(cr)} - \frac{1}{6(n+3-k)} + \frac{cr}{24(n+3-k)+96} + O(cr)^2\right),$$

and

$$(2n-1)!! = \frac{(2n)!}{2^n n!}.$$

Therefore

$$\lim_{r\to 0}\frac{1}{r}\frac{d\Phi}{dr} = \frac{1}{(2n-1)!!}\frac{2n!}{n!2^n}\frac{1}{3} = \frac{1}{3}.$$

2.2 Gaussian radial basis functions

The Gaussian RBFs have been widely used in the area of neural networks [109]. However, the Gaussian RBFs are rarely being used for solving PDEs [74]. In the past, the closed-form particular solutions for the Laplace operator using the Gaussian as a basis function was not available due to the difficulty of the integration involving the Gaussian RBFs. In recent years, the symbolic computational tools such as MATHEMATICA and MAPLE have made the difficult integration task possible. Furthermore, the evaluations of special functions such as Bessel functions, error functions, and exponential integral functions are available as library functions or built-in functions in many computational platforms such as MATLAB, FORTRAN and C++. These special functions which often involve infinite series have been considered as the closed-form functions and can be evaluated efficiently and accurately. Motivated by the availability of these new computational tools, we re-investigate the possibility of deriving the particular solutions for various differential operators using the Gaussian RBFs.

The main goal of this study is to focus on the derivation of the closed-form particular solutions using the Gaussian RBFs [58] which are presented in Section 2.2.1. We apply the MPS for some boundary value problems in Section 2.3 and LMPS for a 3D problem in the Section 3.3 to validate the derived particular solutions. Note that the MPS is a global RBFs meshless method and LMPS is a localized meshless method which is capable of handling a large number of collocation points.

2.2.1 Particular Solutions

In this section, we will present the solution to the inhomogeneous Laplace and biharmonic operators with a Gaussian RBFs on the right hand side.

Particular Solutions for Laplace operator

Theorem 2.2.1. Let $\phi(r) = \exp(-cr^2)$, $c > 0$, and $\Delta\Phi(r) = \phi(r)$ in 2D. Then,

$$\Phi(r) = \begin{cases} \dfrac{1}{4c}\,\mathrm{Ei}(cr^2) + \dfrac{1}{2c}\log(r), & r \neq 0, \\ \dfrac{-1}{4c}(\gamma + \log(c)), & r = 0, \end{cases} \tag{2.37}$$

where

$$\mathrm{Ei}(x) = \int_x^\infty \frac{e^{-u}}{u}\,du, \tag{2.38}$$

and $\gamma \simeq 0.5772156649015328$ is the Euler-Mascheroni constant [7]. Note that $\mathrm{Ei}(x)$ is the special function known as the exponential integral function [7].

Proof. Suppose

$$\Delta\Phi = \exp(-cr^2). \tag{2.39}$$

In polar co-ordinates in 2D, for radial invariant functions we have

$$\Delta = \frac{1}{r}\frac{d}{dr}\left(r\frac{d}{dr}\right). \tag{2.40}$$

By direct integration on both sides of (2.39), we have

$$\begin{aligned} r\frac{d\Phi}{dr} &= \int r\exp(-cr^2)dr \\ &= -\frac{1}{2c}\exp(-cr^2)+C_0. \end{aligned} \tag{2.41}$$

It follows that

$$\Phi(r) = \frac{1}{4c}\operatorname{Ei}(cr^2)+C_0\log(r)+C_1, \tag{2.42}$$

where C_0 and C_1 are integration constants. Note that [7]

$$\operatorname{Ei}(cr^2) = -\gamma-\log(cr^2)+cr^2+\frac{c^2r^4}{4}+O(r^5), \tag{2.43}$$

which contains a singular term at $r=0$. By choosing the integration constant $C_0 = 1/2c$, we can de-singularize $\Phi(r)$ in (2.42). Another integration constant C_1 in (2.42) can be chosen arbitrarily. For convenience, we set $C_1 = 0$. Hence, $\Phi(r)$ in (2.37) is proved. Furthermore,

$$\begin{aligned} \lim_{r\to 0}\Phi(r) &= \lim_{r\to 0}\left(\frac{1}{4c}\operatorname{Ei}(cr^2)+\frac{1}{2c}\log(r)\right) \\ &= -\frac{1}{4c}\left(\gamma+\log(c)\right). \end{aligned}$$

□

To solve the problem with Neumann boundary condition, we need to find $1/r(d\Phi/dr)$. From (2.41), we have

$$r\frac{d\Phi}{dr} = \frac{-1}{2c}\exp(-cr^2)+\frac{1}{2c}. \tag{2.44}$$

After multiplying $1/r^2$ on both sides of (2.44), we have

$$\frac{1}{r}\frac{d\Phi}{dr} = \frac{-\exp(-cr^2)+1}{2r^2c}, \quad r\neq 0.$$

Furthermore,

$$\lim_{r\to 0}\frac{1}{r}\frac{d\Phi}{dr} = \frac{1}{2}.$$

Theorem 2.2.2. Let $\phi(r) = \exp(-cr^2)$, and $\Delta\Phi(r) = \phi(r)$ in 3D. Then,

$$\Phi(r) = \begin{cases} \dfrac{-\sqrt{\pi}}{4c^{3/2}r}\operatorname{erf}(\sqrt{c}r), & r\neq 0, \\ \dfrac{-1}{2c}, & r=0, \end{cases} \tag{2.45}$$

where $\operatorname{erf}(x)$ is the special function defined as follows [7]

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-u^2}du. \tag{2.46}$$

Proof. In the 3D case, we can use the following definition of the Laplace operator and follow a similar derivation as shown in Theorem 2.2.1. Note that in 3D

$$\Delta = \frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\right). \tag{2.47}$$

By direct integration, we have

$$\Phi(r) = \frac{-\sqrt{\pi}}{4rc^{3/2}}\operatorname{erf}(\sqrt{c}r) - \frac{C_0}{r} + C_1. \tag{2.48}$$

The error function $\operatorname{erf}(\sqrt{c}r)$ can be expanded in series as follows [7]

$$\operatorname{erf}(\sqrt{c}r) = \frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n(\sqrt{c}r)^{2n+1}}{(2n+1)n!}. \tag{2.49}$$

From (2.48) and (2.49), we have

$$\Phi(r) = \frac{-1}{2c^{3/2}}\sum_{n=0}^{\infty}\frac{(-1)^n c^{(2n+1)/2} r^{2n}}{(2n+1)n!} - \frac{C_0}{r} + C_1. \tag{2.50}$$

The first term of (2.50) does not contain any singularity. Hence, C_0 and C_1 in (2.48) can be set to equal zero. Hence, (2.45) is proved. Furthermore, from (2.50),

$$\begin{aligned}\lim_{r\to 0}\Phi(r) &= \lim_{r\to 0}\frac{-1}{2c^{3/2}}\sum_{n=0}^{\infty}\frac{(-1)^n c^{(2n+1)/2} r^{2n}}{(2n+1)n!} \\ &= \frac{-1}{2c}.\end{aligned}$$

□

To solve the problem with Neumann boundary condition, after the differentiation of (2.45) and multiplying by $1/r$, we get

$$\frac{1}{r}\frac{d\Phi}{dr} = -\frac{\sqrt{\pi}}{4c^{3/2}r}\left(\frac{2\sqrt{c}\exp(-cr^2)}{\sqrt{\pi}} - \frac{\operatorname{erf}(\sqrt{c}r)}{r^2}\right), \tag{2.51}$$

and

$$\begin{aligned}\lim_{r\to 0}\frac{1}{r}\frac{d\Phi}{dr} &= -\frac{\sqrt{\pi}}{4c^{3/2}}\lim_{r\to 0}\left(\frac{2\sqrt{c}}{\sqrt{\pi}r}\exp(-cr^2) - \frac{\operatorname{erf}(\sqrt{c}r)}{r^3}\right) \\ &= -\frac{\sqrt{\pi}}{4c^{3/2}}\lim_{r\to 0}\left(-\frac{4c^{3/2}}{3\sqrt{\pi}} + \frac{4c^{5/2}r^2}{5\sqrt{\pi}} - \frac{2c^{7/2}r^4}{7\sqrt{\pi}} + O(r^6)\right) \\ &= \frac{1}{3}.\end{aligned}$$

Particular Solutions for biharmonic operator

For the derivation of the particular solutions of the biharmonic operator in 2D, we first decompose the biharmonic operator into two Laplace operators. We then follow a similar derivation as shown in Theorem 2.2.1 to obtain the following theorem.

Theorem 2.2.3. Let $\phi(r) = \exp(-cr^2), c > 0$, and $\Delta^2\Phi(r) = \phi(r)$ in 2D. Then

$$\Phi(r) = \begin{cases} \frac{1}{8c}\left(r^2\log(r) - r^2\right) - \frac{1}{16c^2}\left(\exp(-cr^2) + \mathrm{Ei}(cr^2)\right) & \\ \quad + \frac{1}{16c} r^2\,\mathrm{Ei}(cr^2) + \frac{1}{8c^2}\log(r), & r \neq 0, \\ \frac{1}{16c^2}(-\gamma - \log(c) - 1), & r = 0, \end{cases} \tag{2.52}$$

where γ is the Euler-Mascheroni constant.

To solve the problem with Neumann boundary condition, we use

$$\frac{1}{r}\frac{d\Phi}{dr} = \frac{-1}{8c} + \frac{1}{4c}\log(r) + \frac{1}{8c}\mathrm{Ei}(cr^2) - \frac{1}{8c^2r^2}(\exp(-cr^2) - 1), \;\; r \neq 0,$$

and

$$\lim_{r\to 0}\frac{1}{r}\frac{d\Phi}{dr} = \frac{-1}{8c}(\gamma + \log(c)).$$

Similarly, we follow the proof of Theorem 2.2.2 to derive the particular solutions for biharmonic operator in 3D.

Theorem 2.2.4. Let $\phi(r) = \exp(-cr^2)$, and $\Delta^2\Phi(r) = \phi(r)$ in 3D. Then

$$\Phi(r) = \begin{cases} \frac{-\sqrt{\pi}}{4c^{3/2}}\left(\mathrm{erf}(\sqrt{c}r)\left(\frac{r}{2} + \frac{1}{4cr}\right) + \frac{1}{2\sqrt{c\pi}}\exp(-cr^2)\right), & r \neq 0, \\ \frac{-1}{4c^2}, & r = 0. \end{cases} \tag{2.53}$$

We can use

$$\frac{1}{r}\frac{d\Phi}{dr} = \frac{-\sqrt{\pi}}{4c^{3/2}}\left(\mathrm{erf}(\sqrt{c}r)\left(\frac{1}{2r} - \frac{1}{4cr^3}\right) + \frac{1}{2\sqrt{c\pi}r^2}\exp(-cr^2)\right),$$

and

$$\lim_{r\to 0}\frac{1}{r}\frac{d\Phi}{dr} = \frac{-1}{6c},$$

for the Neumann boundary condition.

2.3 Numerical results

Numerical examples are solved using MATLAB in a 16 GB memory with Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz processor. In this section, we present some numerical examples in 2D and 3D to validate the derived particular solutions for Matérn and Gaussian RBFs using MPS. To validate the derived particular solutions for the Matérn RBFs in Sections 2.1 and 2.2, four numerical examples in 2D and 3D are given. We examine the effectiveness of Matérn and Gaussian RBFs in the context of the MPS and compare the results with those obtained when using the traditional normalized MQ ($\sqrt{1+c^2r^2}$).

The Root Mean Square Error (RMSE) and the Maximum Absolute Error (MAE) are used to measure the accuracy. They are defined as follows:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{n_t}\sum_{j=1}^{n_t}(\hat{u}_j - u_j)^2}, & (2.54) \\ \text{MAE} &= \max_{1\leq j\leq n_t} |\hat{u}_j - u_j|, & (2.55) \end{aligned}$$

where n_t is the number of test points located randomly within the domain, and u_j and $\hat{u}_j$ are the exact and numerical solutions at the jth node respectively.

Example 2.3.1. We consider the following Poisson equation with Dirichlet boundary condition in 2D

$$\begin{aligned} \Delta u(x,y) &= f(x,y), \quad (x,y) \in \Omega, & (2.56) \\ u(x,y) &= g(x,y), \quad (x,y) \in \partial\Omega, & (2.57) \end{aligned}$$

where f and g are chosen according to the following exact solutions:

$$\begin{aligned} u_1(x,y) &= e^{x+y}, \ (x,y) \in \bar{\Omega}, & (2.58) \\ u_2(x,y) &= \sin(\pi x)\cos(\frac{\pi y}{2}), \ (x,y) \in \bar{\Omega}. & (2.59) \end{aligned}$$

The computational domain $\bar{\Omega} = \Omega \cup \partial\Omega$ as shown in Figure 2.1 is bounded by the curve defined by the following parametric equation:

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

where

$$\rho(\theta) = e^{\sin(\theta)}\sin^2(2\theta) + e^{\cos(\theta)}cos^2(2\theta).$$

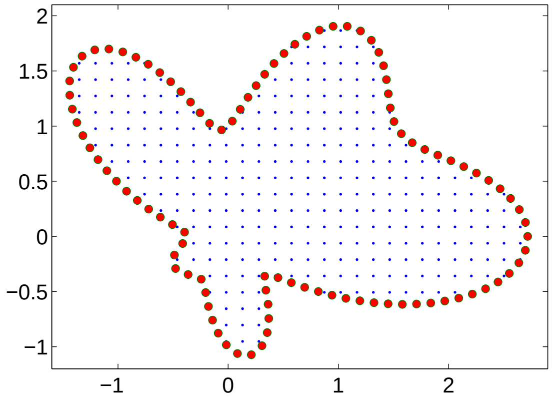

Figure 2.1: The profiles of interpolation points and boundary points of the amoeba-shaped domain

For the implementation of the MPS, we choose 414 uniformly distributed interior points, 100 boundary points and 200 randomly distributed interior points as the test points. The profile of RMSE for $u_2(x,y)$ using various orders of Matérn, Gaussian and normalized MQ RBFs versus the shape parameter is shown in Figure 2.2. We observe that the lower order Matérn RBFs are more stable but less accurate and vice versa. Matérn RBFs are more stable than normalized MQ when shape parameter becomes larger. As we see in Figure 2.2, there is no significant difference among the Matérn RBFs of various orders and the normalized MQ in terms of accuracy. Gaussian RBFs is more accurate than normalized MQ and Matérn RBFs of order 4 to 8.

The MAE in Table 2.1 for the numerical solutions of the Poisson equation with the analytical solutions $u_1(x,y)$ and $u_2(x,y)$ are obtained using Matérn RBFs of various order, Gaussian and the normalized MQ RBFs. These results show that as Matérn order increases the optimum shape parameter becomes larger which is consistent with the results shown in Figure 2.2. For the Poisson equation with the analytical solution $u_1(x,y)$, we obtained no significant difference among the Matérn RBFs of various orders, Gaussian and the normalized MQ RBFs in terms of accuracy. For $u_2(x,y)$, Table 2.1 shows that more accurate result is obtained by Gaussian RBFs compare with the other RBFs.

Example 2.3.2. In this example, we consider Poisson equation with Dirichlet boundary condition in 3D

$$\Delta u(x,y,z) = f(x,y,z), \quad (x,y,z) \in \Omega, \tag{2.60}$$

$$u(x,y,z) = g(x,y,z), \quad (x,y,z) \in \partial\Omega,, \tag{2.61}$$

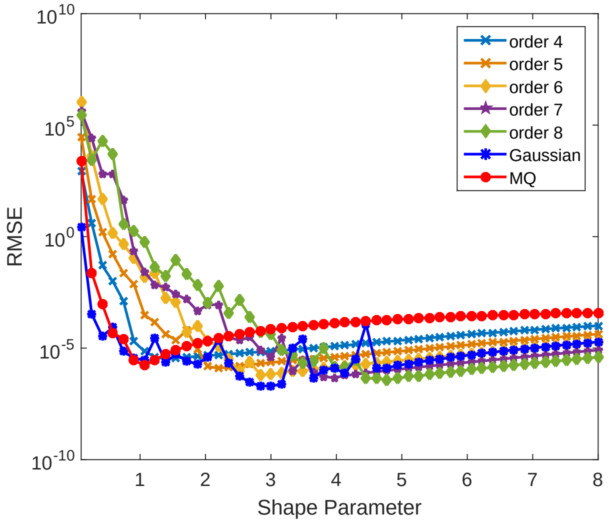

Figure 2.2: Errors versus shape parameters for $u_2(x,y)$ using various orders of Matérn, Gaussian and normalized MQ RBFs.

Table 2.1: Example 2.3.1: The optimal shape parameters and the corresponding MAE for various orders of Matérn, Gaussian and normalized MQ RBFs.

		$u_1(x,y)$		$u_2(x,y)$	
RBFs	order	shape parameter	MAE	shape parameter	MAE
Matérn	2	0.26	3.349E-04	1.87	3.819E-04
	3	0.58	6.604E-05	0.58	4.967E-05
	4	1.07	2.485E-05	1.39	1.608E-05
	5	1.87	1.657E-05	2.03	5.762E-06
	6	2.84	1.708E-05	2.84	2.571E-06
	7	3.49	1.681E-05	3.81	2.872E-06
	8	4.29	1.359E-05	4.45	1.967E-06
Gaussian		2.68	1.802E-05	2.84	5.784E-07
MQ		1.07	1.120E-05	1.07	7.759E-06

where $f(x,y,z)$ and $g(x,y,z)$ are given according to the following exact solutions:

$$u_1(x,y,z) = e^{x+y+z}, \quad (x,y,z) \in \bar{\Omega}, \tag{2.62}$$

$$u_2(x,y,z) = \cos(x)\cos(y)\cos(z), \quad (x,y,z) \in \bar{\Omega}. \tag{2.63}$$

The computational domain $\bar{\Omega} = \Omega \cup \partial\Omega$ is a bumpy sphere with the boundary $\partial\Omega$ which is defined by the following parametric equation

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

where

$$r = 1 + \frac{1}{6}\sin(7\theta)\sin(6\phi)\,.$$

The profiles of the bumpy sphere (left) and its boundary condition for $u_1(x,y,z)$(right) are shown in Figure 2.3. In the numerical implementation, we choose 1092 uniformly distributed interior points, 400 uniformly distributed surface points and 120 uniformly distributed interior points as the test points.

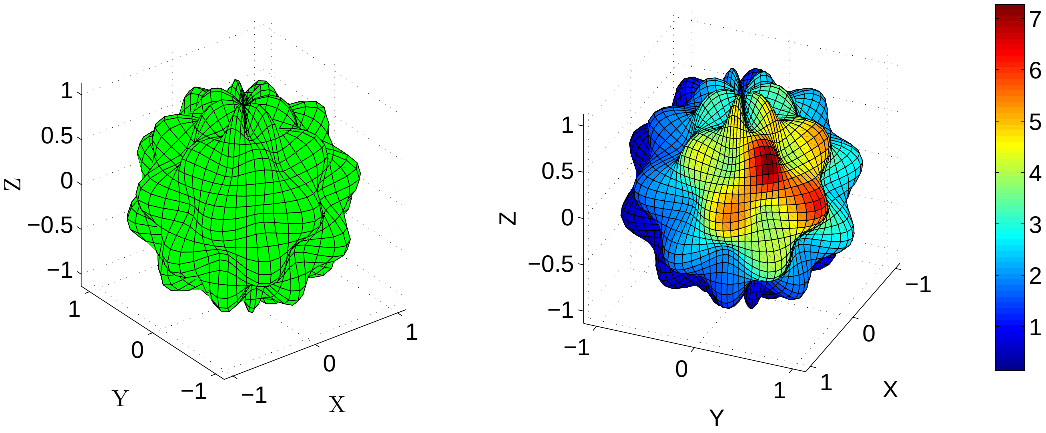

Figure 2.3: The profiles of bumpy sphere (left) and the boundary condition for $u_1(x,y,z)$ on its surface (right).

In Table 2.2, we present the MAE for the numerical solutions of the Poisson equation in 3D with two different analytical solutions $u_1(x,y,z)$ and $u_2(x,y,z)$ using Matérn RBFs of different orders, Gaussian and the normalized MQ RBFs. Figure 2.4 depicts the profile of the RMSE for $u_2(x,y,z)$ using Matérn RBFs for orders $9/2$ to $17/2$, Gaussian and the normalized MQ RBFs.

Example 2.3.3. We consider the following convection-diffusion equation:

$$\left(\Delta^2 + 2y\sin(x)\frac{\partial}{\partial x} - y\cos(x)\frac{\partial}{\partial y} + xy\right)u(x,y) = f(x,y), \quad (x,y) \in \Omega, \qquad (2.64)$$

$$u(x,y) = g_1(x,y), \quad (x,y) \in \partial\Omega, \qquad (2.65)$$

$$\Delta u(x,y) = g_2(x,y), \quad (x,y) \in \partial\Omega, \qquad (2.66)$$

Table 2.2: Example 2.3.2: MAE for different orders of Matérn, Gaussian, normalized MQ RBFs and optimal shape parameters

		$u_1(x,y,z)$		$u_2(x,y,z)$	
RBFs	order	shape parameter	MAE	shape parameter	MAE
Matérn	5/2	0.26	2.295E-03	0.34	6.238E-05
	7/2	0.90	4.610E-04	0.74	3.669E-05
	9/2	1.30	2.350E-04	1.30	7.877E-06
	11/2	1.78	1.753E-04	1.54	3.453E-06
	13/2	2.81	3.490E-05	2.73	2.116E-06
	15/2	3.77	4.261E-05	2.97	8.485E-07
	17/2	3.69	3.012E-05	3.69	7.088E-07
Gaussian		2.15	4.481E-05	1.87	4.585E-07
MQ		0.65	8.263E-05	0.73	1.625E-07

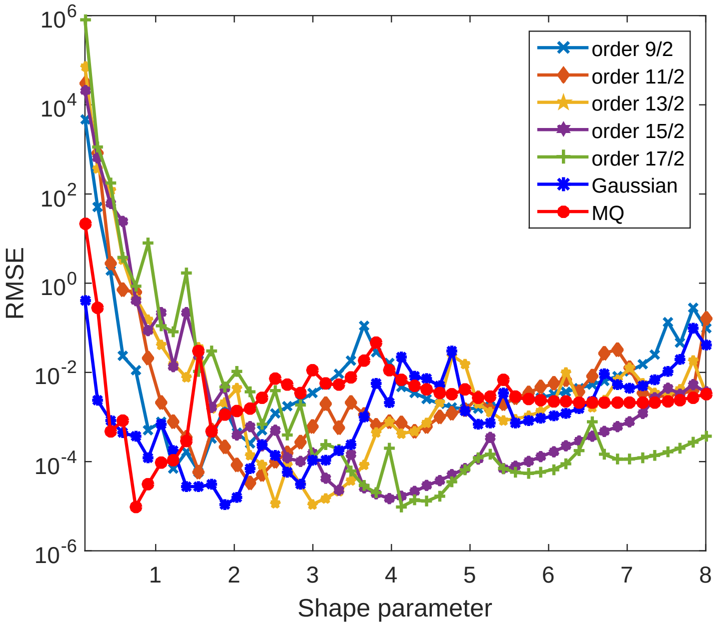

Figure 2.4: Errors versus shape parameters for $u_2(x,y,z)$ using various orders of Matérn, Gaussian and normalized MQ RBFs.

where $f(x,y), g_1(x,y)$, and $g_2(x,y)$ are given functions according to the following analytical solution:

$$u(x,y) = y\sin(x) + x\cos(y) \quad (x,y) \in \bar{\Omega}. \tag{2.67}$$

The boundary $\partial\Omega$ is defined by the following parametric equation:

$$\partial\Omega = \{(x,y) |\, x = \rho(t)\cos(t + \frac{1}{2}\sin(4t)),\; y = \rho(t)\sin(t + \frac{1}{2}\sin(4t)),\; 0 \le t < 2\pi\}$$

where

$$\rho(t) = \frac{1}{2}\left(2 + \frac{1}{2}\sin(4t)\right).$$

The computational domain Ω is a gear-shaped domain as shown in Figure 2.5.

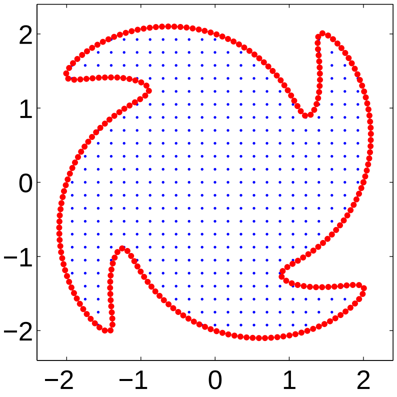

Figure 2.5: The profiles of interpolation points and boundary points of the gear-shaped domain.

In the numerical implementation, we choose 417 uniformly spaced interior points, 250 uniformly spaced boundary points and 231 randomly distributed test points. The numerical results presented in Table 2.3 are obtained using Matérn RBFs of orders 4 to 10 and Gaussian RBFs. The MAE and RMSE are consistent for various orders of Matérn RBFs. We observed that the numerical result obtained using Gaussian RBFs is more accurate than the Matérn RBFs of order 4 to 10.

Example 2.3.4. Finally, we consider the fourth order boundary value problem:

$$\left(\Delta^2 + y\cos(y)\frac{\partial}{\partial x} + \sinh(x)\frac{\partial}{\partial y} + x^2y^3\right)u(x,y) = f(x,y), \quad (x,y) \in \Omega, \tag{2.68}$$

$$u(x,y) = g_1(x,y), \quad (x,y) \in \partial\Omega, \tag{2.69}$$

$$\frac{\partial u}{\partial n}(x,y) = g_2(x,y) \cdot \mathbf{n}, \quad (x,y) \in \partial\Omega, \tag{2.70}$$

Table 2.3: Example 2.3.3: Accuracy and optimum shape parameters obtained by using various Matérn orders and Gaussian RBFs.

RBFs	order	shape parameter	MAE	shape parameter	RMSE
Matérn	4	2.32	2.813E-04	2.32	6.131E-05
	5	3.53	2.478E-04	3.13	5.838E-05
	6	3.74	1.962E-04	3.74	4.364E-05
	7	4.75	2.312E-04	4.75	6.047E-05
	8	5.96	2.568E-04	5.15	5.298E-05
	9	6.36	1.667E-04	5.76	4.085E-05
	10	4.54	1.706E-04	3.94	3.568E-05
Gaussian		1.72	3.486E-05	1.11	1.188E-05

where $f(x,y), g_1(x,y)$, and $g_2(x,y)$ are known functions according to the following analytical solution:

$$u(x,y) = \sin(\pi x)\cosh(y) - \cos(\pi x)\sinh(y), \quad (x,y) \in \bar{\Omega}. \tag{2.71}$$

The computational domain $\bar{\Omega} = \Omega \cup \partial\Omega$ is bounded by the following peanut-shaped parametric curve as shown in Figure 2.6:

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

where

$$\rho(\theta) = \cos(2\theta) + \sqrt{1.1 - \sin^2(2\theta)}.$$

In the numerical implementation, we choose different numbers of interior and boundary points. We denote by n_i the number of interior points and by n_b the number of boundary points. We choose 200 randomly distributed test points. In this example, we use the Matérn RBFs of order 6 and Gaussian RBFs. In Table 2.4, we test two sets of interior points and three sets of boundary points for each RBFs.

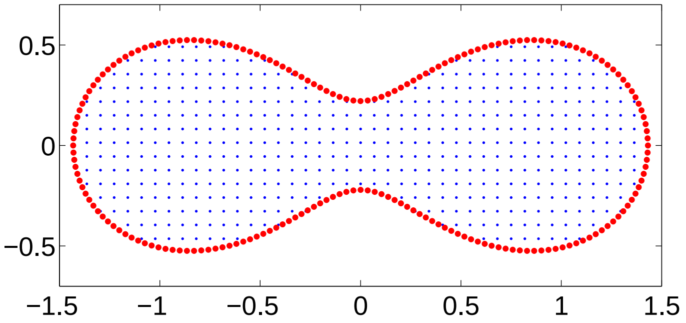

Figure 2.6: The profiles of interpolation points and boundary points of the peanut-shaped domain

Table 2.4: Example 2.3.4: Optimal shape parameter and MAE for Matérn order 6 and Gaussian RBFs.

	(n_i, n_b)	shape parameter	MAE	shape parameter	RMSE
Matérn order 6	$(493, 150)$	7.88	6.276E-04	8.18	1.489E-04
	$(493, 200)$	8.18	6.266E-04	8.18	1.447E-04
	$(493, 250)$	8.03	7.257E-04	8.79	1.755E-04
	$(614, 150)$	9.39	4.053E-04	10.15	8.433E-05
	$(614, 200)$	10.00	3.999E-04	10.30	9.315E-05
	$(614, 250)$	9.39	4.022E-04	10.45	8.949E-05
Gaussian	$(493, 150)$	1.21	1.006E-03	0.91	2.907E-04
	$(493, 200)$	1.06	8.044E-04	1.06	3.028E-04
	$(493, 250)$	1.21	8.992E-04	2.73	2.881E-04
	$(614, 150)$	2.12	9.999E-04	1.97	3.238E-04
	$(614, 200)$	0.91	8.155E-04	0.91	2.704E-04
	$(614, 250)$	0.91	6.225E-04	0.91	2.536E-04

Chapter 3

Selection of the good shape parameter

Many RBFs (Table 1.1) which are used in RBFs collocation methods [6, 20–25, 33, 53, 58–60, 62], contain a shape parameter c. As we have seen in (1.3), the solution of the given PDEs is approximated by the linear superposition of the corresponding particular solutions of the given RBFs. So, if we use one of the RBFs, such as MQ, IMQ, Gaussian, or Matérn, then these RBFs contain the shape parameter c. This shape parameter c plays an important role in obtaining high accurate solution. So, MPS with these RBFs require some strategies to obtain the good shape parameter. In this Chapter, we present some of these strategies.

In RBFs literature, many strategies have been proposed to obtain a good shape parameter. Most of these strategies can be categorized into two types: constant and variable shape parameter.

3.1 Constant shape parameter

We note that, if we consider Gaussian RBFs and its closed-form particular solutions in MPS, then $\phi(r)$ is given by

$$\phi(r) = \exp(-cr^2), \;\; c > 0. \tag{3.1}$$

Now, the solution to (1.1) and (1.2) is approximated by

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \Phi(r_j), \tag{3.2}$$

where

$$\Phi(r_j) = \Phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|),$$

are obtained depending on the differential operator in the PDEs as described in the Section 2.2.

As we can see in Figure 2.2, the accuracy of the solution (3.2) depends on the shape parameter c. Note that in (3.1), from each c, we obtain a new $\phi(r)$. So, if we fix a constant shape parameter c, then in (3.2), we obtain the basis functions which depend only on radial distance. To find the good shape parameter, many strategies have been proposed by several researchers. Since, many constant shape parameter strategies have been already implemented in MPS, we will not review here. The list of some of the strategies to obtain the good constant shape parameter is given in the Appendix B.

Among many strategies, Leave-one-out cross validation (LOOCV) is one of the strategies which has been implemented in RBFs collocation methods in [36, 58, 86]. In recent years, LOOCV has been implemented as one of the alternatives for obtaining the good shape parameter. We have briefly described LOOCV in the Appendix B. By using the derived closed-form particular solutions of Gaussian RBFs from the Chapter 2, we present the numerical results obtained by LOOCV strategy for MPS and LMPS in the Section 3.3. The numerical results presented in the Section 3.3 are published in [58].

In most of the previous MPS literature, the RBFs has always been chosen with a constant shape parameter c but in some other RBFs collocation literature such as RBFs interpolation and Kansa method, variable shape parameter has been successfully implemented. Research shows that even in many cases variable shape parameter produces more accurate results than if a constant shape parameter is used.

In this research work, we propose to use the variable shape parameter in the MPS. First of all, we implement most of the existing variable shape parameter strategies in the MPS to obtain the good variable shape parameter. In [5], Afiatdoust has used a global optimization algorithm known as genetic algorithm (GA) to obtain the variable shape parameter. Including GA, we use other well-known global optimization algorithms such as pattern search (PS) [4, 9] and simulated annealing (SA) [17, 55] to obtain the variable shape parameter. PS is a direct search method for solving optimization problems which do not need any information about the gradient of the objective function. SA is another global optimization algorithm which is motivated by an analogy to the statistical mechanics of annealing in solids. These optimization algorithms are very popular methods in optimization problems and which can be easily implemented in any other RBFs collocation methods as well.

3.2 Variable shape parameter

As we have discussed earlier, for each c in (3.1), we obtain new $\phi(r)$. So, instead of fixing a constant shape parameter c, let us obtain N different $\phi_j(r),\ \ j = 1,...,N$ corresponding to N different shape parameter $c_j,\ \ j = 1,...,N$. Then, we can modify the linear superposition of the particular solutions for the field variable u, by taking a variable shape parameter $\boldsymbol{c} = \{c_1, c_2, ..., c_N\}$ instead of using the constant shape parameter c. For example, the computational formulation of the MPS using Gaussian RBFs in (1.2) is given by

$$\begin{aligned}\sum_{j=1}^{N} \alpha_j \phi_j(||\boldsymbol{x_i}-\boldsymbol{y_j}||_2) &= f(\boldsymbol{x_i}),\ \boldsymbol{x_i} \in \Omega, \\ \sum_{j=1}^{N} \alpha_j \Phi(||\boldsymbol{x_i}-\boldsymbol{y_j}||_2) &= g(\boldsymbol{x_i}),\ \boldsymbol{x_i} \in \partial\Omega,\end{aligned} \tag{3.3}$$

where

$$\phi_j(||\boldsymbol{x_i}-\boldsymbol{y_j}||_2) = \exp(-c_j||\boldsymbol{x_i}-\boldsymbol{y_j}||_2^2) = \exp(-c_j r_j^2),$$

and $\Phi(||\boldsymbol{x_i}-\boldsymbol{y_j}||_2) = \Phi(r_j)$ are the corresponding particular solutions obtained in the Section 2.2.

The discretization in (3.3) is written in the matrix form as $\boldsymbol{A\alpha} = \boldsymbol{b}$, where $\boldsymbol{A}$ is the matrix obtained from the evaluation of the RBFs and its corresponding particular solutions, $\boldsymbol{\alpha}$ is the column vector containing the unknown coefficients and $\boldsymbol{b}$ is the column vector containing the right hand side terms.

Hence, in the matrix form of the computational formulation of the MPS by using the variable shape parameter, we have a different c_j in each column of the matrix. These c_j's are obtained by several previously developed strategies, which are given as follows:

All of the strategies given below, user most provide c_{min} and c_{max}, where

$$c_{min} = \min\{c_1, c_2, ..., c_N\} \text{ and } c_{max} = \max\{c_1, c_2, ..., c_N\}.$$

In [52], the formula

- **S1**:

$$c_j = \left[c_{min}^2 \left(\frac{c_{max}^2}{c_{min}^2} \right)^{\dfrac{j-1}{N-1}} \right]^{1/2}, \quad j = 1, 2, ..., N,$$

has been derived which gives an exponentially varying shape parameter (See Figure 3.1 (a)).

The linearly varying shape parameter formula (See Figure 3.1 (b))

- **S2**:

$$c_j = c_{min} + \left(\frac{c_{max} - c_{min}}{N-1} \right) j, \quad j = 0, 1, ..., N-1,$$

and randomly varying shape parameter formula (See Figure 3.2 (a))

- **S3**:

  $$c_j = c_{min} + (c_{max} - c_{min}) \times rand(1,N),$$

  has been introduced in [89].

  In [104], a trigonometric varying shape parameter selection formula has been also introduced as

- **S4**:

  $$c_j = c_{min} + (c_{max} - c_{min}) \times \sin(j), \quad j = 1,2,...,N.$$

- **S5**: Genetic algorithm (**GA**) has been used as a strategy in [5].

  Instead of using the S4 in the original format, we use it in the following way:

- **S6**:

  $$c_j = c_{min} + (c_{max} - c_{min}) \times \sin^2(j), \quad j = 1,2,...,N.$$

  Keeping in this format will avoid the negative real numbers for the selection of each c_j's in variable shape parameter (See Figure 3.2 (b)). Since Sara *et. al* [89] has proposed the random varying shape parameter by using the uniformly distributed random numbers as in S3, we now use the Halton quasi-random numbers to find the variable shape parameter which is done by using the MATLAB *haltonset* and the *net* built in function. This is done in the similar fashion as in S3; i.e.,

- **S7**:

  $$c_j = c_{min} + (c_{max} - c_{min}) \times net(p,N), \quad p = haltonset(1),$$

  where $p = haltonset(1)$ constructs an one-dimensional point set p of the haltonset class and $net(p,N)$ returns the first N points from the point set p of the sequence of the multi-dimensional quasi-random numbers (See Figure 3.3).

In the Figures 3.1 - 3.3, each c_j's of the variable shape parameter for each column col_n of the matrix A for different strategies has been plotted. We have used 36 RBFs centers to produce these plots with $c_{min} = 0.5$ and $c_{max} = 10$. Among the strategies, exponential and linearly varying shape parameter strategies S1 and S2, produce a monotonically increasing c_j's for the increasing numbers of the RBFs centers but the strategy obtained from the random numbers such as S3 and S7 produce randomly distributed c_j.

By numerical experiments, we observe that choosing a right c_{min} and c_{max} is also another issue to be addressed in the above mentioned strategies. So, to address the above mentioned issue, we propose to couple with some global optimizations tools. Among the global

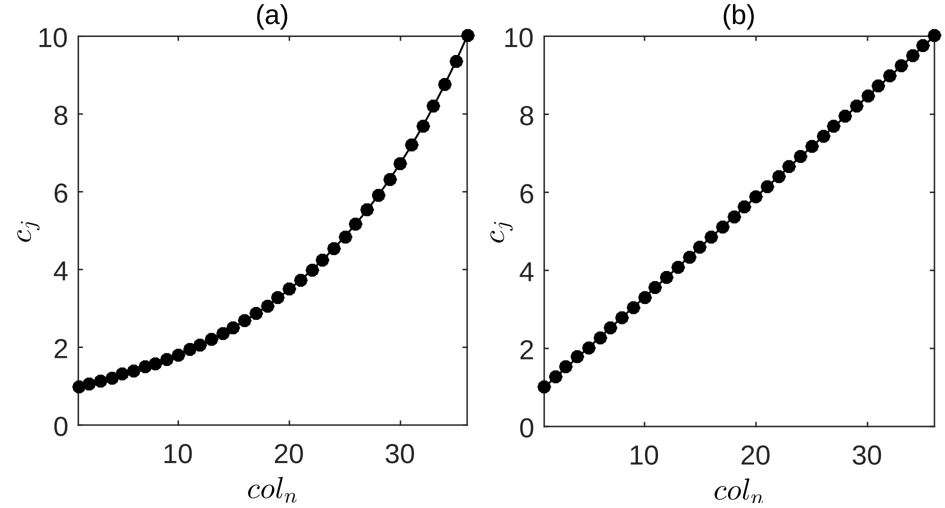

Figure 3.1: $(a),(b)$ are the figures obtained by S1 and S2 respectively for the c_j's and the column col_n of the collocation matrix A with 36 RBFs centers.

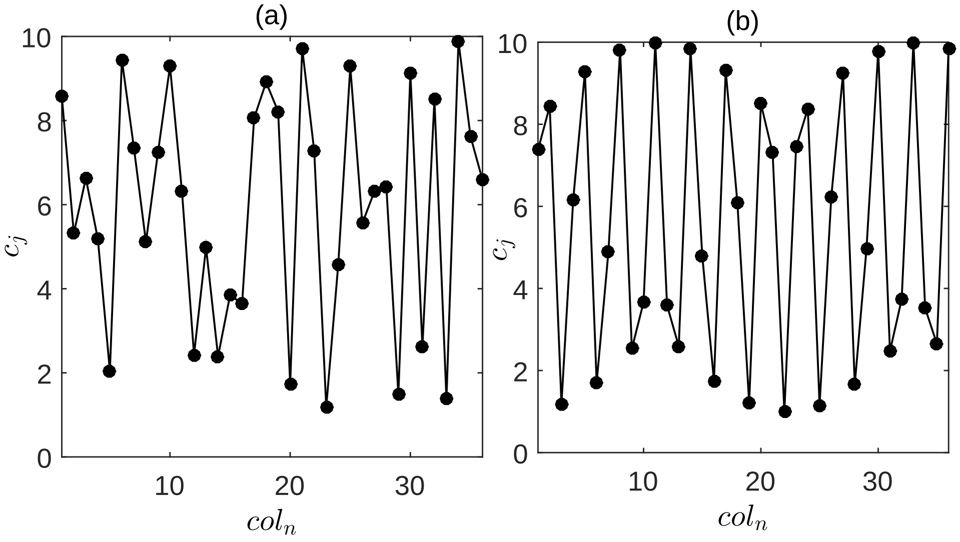

Figure 3.2: $(a),(b)$ are the figures obtained by S3 and S6 respectively for the c_j's and the column col_n of the collocation matrix A with 36 RBFs centers.

optimization tools, GA has been already proposed for the variable shape parameter selection in RBFs interpolation and Kansa method. In this work, we experiment the effectiveness of the GA in MPS for different search interval. Then, we also use other well-known global optimization tools such as Pattern search (PS) and Simulated annealing (SA). PS and SA need an initial guess of the variable shape parameter as an extra user input. So, we implement

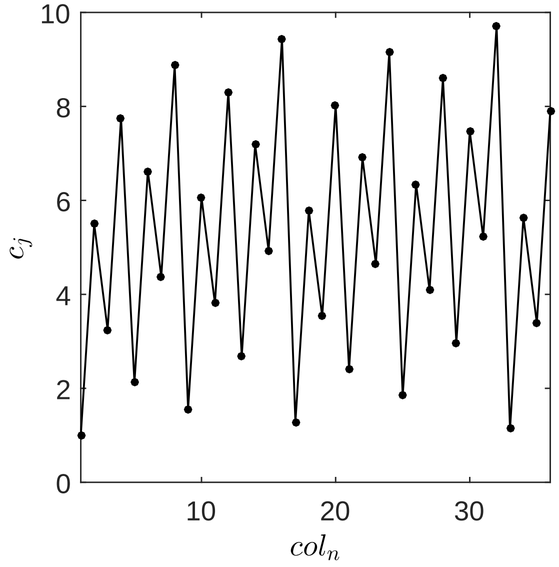

Figure 3.3: c_j's for the column col_n of the collocation matrix A produced by S7.

the variable shape parameter obtained from above strategies S1-S3, S6, S7 as an initial guess to these global optimization tools.

Numerical experiments in the Section 3.4 validate that the latter proposed idea boost the accuracy of the MPS for exponentially varying strategy for a search interval which has a very large length.

3.2.1 Pattern Search (PS)

PS is a very popular direct search method for solving optimization problems which does not need any information about the gradient of the objective function. At each iterative step, PS searches a set of points around the point which has been computed in the previous step [4, 9]. In this way this algorithm generates a sequence of points which eventually approach to an optimal point. In MATLAB, we can find the *patternsearch* built in function in the global optimization toolbox. A lot of parameters can be easily adjusted with the *psoptimset* function, which enable us to create the pattern search options structure. For details about PS, we refer reader to the direct search optimization literature [4, 9].

3.2.2 Simulated Annealing (SA)

SA is a probabilistic global optimization method which is inspired by an analogy between the physical annealing process of solids and the problem of solving optimization problems [17, 55]. In MATLAB, *simulannealbnd* function can be found in the global optimization toolbox. Similarly, we can adjust the SA parameters by *saoptimset* function which can create simulated annealing options structure.

3.2.3 Objective function

All of these optimization tools need an objective function to be minimized. We have constructed the objective function for the global optimizations by adopting the similar structure that has been used in [5], which is actually the residual error measured at couple of test points. Let us represent the objective function by the following function:

$$\mathbf{F1} = \sqrt{\frac{\sum_{i=1}^{ni_t} (\Delta\hat{u}(x_i) - f(x_i))^2}{ni_t}},$$

where ni_t is the number of the interior test points $\{x_i\}_{i=1}^{ni_t}$ in the computational domain Ω and $\hat{u}$ is the approximate solution. Once this objective function is set up, PS and SA find an optimal shape parameter which minimizes the objective function **F1**. In our numerical examples in the Section 3.4, we have calculated residual error for 100 randomly distributed test points.

3.3 Numerical results for good constant shape parameter using LOOCV

In the numerical implementation, we use the MATLAB© built-in functions `‘expint’` , `‘erf’` to compute the special functions Ei and erf respectively. To select the good constant shape parameter, the LOOCV [36, 86] strategy has been adopted. In the search algorithm using LOOCV, we have used `‘fminbnd’` to find the minimum of a function of one variable within a fixed interval. We denote [min, max] as the initial search interval for `‘fminbnd’`.

Example 3.3.1. Consider the fourth order convection-diffusion equation (2.3.3) with the analytical solution

$$u(x,y) = y\sin x + x\cos y, \quad (x,y) \in \overline{\Omega}. \tag{3.4}$$

The computational domain $\bar{\Omega}$ is bounded by the following parametric equation:

$$\{(x,y)|x = \rho(t)\cos(t + \frac{1}{2}\sin(7t)), y = \rho(t)\sin(t + \frac{1}{2}\sin(7t)), \ 0 \le \theta < 2\pi\},$$

where

$$\rho(t) = \frac{1}{2}\left(2 + \frac{1}{2}\sin(7t)\right).$$

The computational domain $\bar{\Omega}$ is a gear-shaped domain as shown in Figure 3.4. Note that $\bar{\Omega}$ is not only irregular but also contains sharp edges which presents a challenge in obtaining good numerical accuracy.

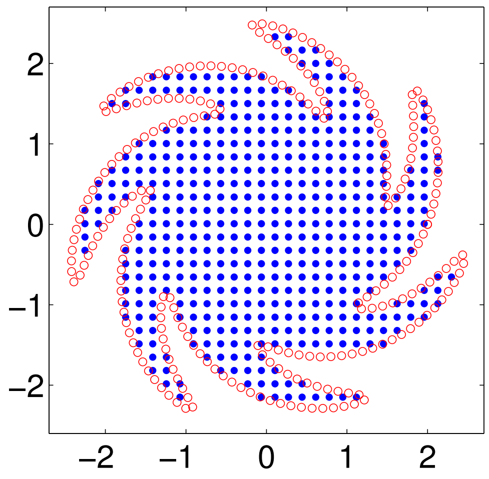

Figure 3.4: Interior points (●) and boundary points (○) of the gear-shaped domain.

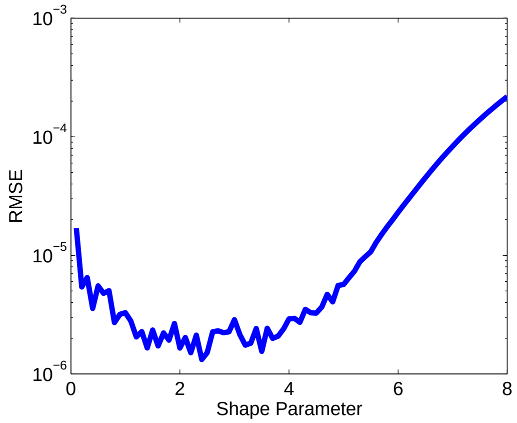

Figure 3.5: RMSE errors for different shape parameters.

In the numerical implementation, we choose 273 randomly distributed test points inside the gear-shaped domain. In Figure 3.5 we show the results of RMSE versus shape parameter using 635 interior and 250 boundary points. Notice that the good shape parameter range is between 1.5 and 4. In Table 3.1, n_i and n_b denote the number of interior and boundary points, respectively, and c_{loocv} is the shape parameter obtained by LOOCV. Using the LOOCV algorithm, the selected shape parameter is 1.792 as shown in Table 3.1. For all the results in Table 3.1, the initial search interval for LOOCV is set to [0, 5]. The number of iterations using LOOCV is 12. In this table we also show the results using various interior and boundary points. Despite the difficulty of a complicated domain with sharp edges, the numerical accuracy we obtained seems reasonable.

Table 3.1: RMSE using the Gaussian RBFs with various interior and boundary points.

n_i	n_b	c_{loocv}	RMSE
459	150	3.701	3.1687E-05
530	200	4.271	1.4274E-05
635	250	1.792	1.8478E-06

Example 3.3.2. Consider the Poisson problem (2.60) with Dirichlet boundary condition (2.61). The functions f and g are given according to the analytic solution (2.62). The computational domain is the so-called bumpy sphere (see Figure 3.6) which has been used as models for tumors. The surface of the considered domain is highly complicated. The spherical parametrization of the bumpy sphere is as follows:

$$\{(x,y,z) : x = \rho \sin\phi \cos\theta, y = \rho \sin\phi \sin\theta, z = \rho \cos\phi, \quad 0 \le \theta \le 2\pi, \ 0 \le \phi \le \pi\},$$

where

$$\rho(\phi,\theta) = 1 + \frac{1}{6}\sin(6\theta)\sin(7\phi).$$

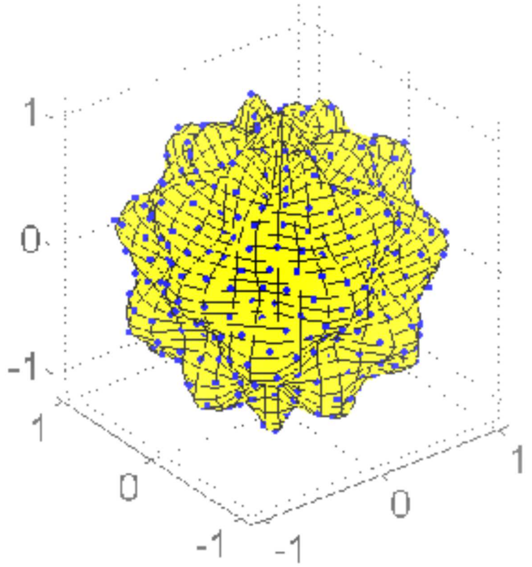

Figure 3.6: The profile of computational domain (bumpy sphere) and the uniformly distributed boundary points.

For the 3D problems, more interior and boundary points are needed. In this example, we apply the localized method of particular solutions (LMPS) [107] which is a localized meshless method to handle a large number of interpolation points. In the numerical implementation, we choose $21,672$ uniformly distributed interior points and $3,000$ uniformly distributed boundary points. For each local influence domain, we choose 35 nearest neighboring points for each RBFS center. The LOOCV algorithm is used for the selection of a good shape parameter. In Table 3.2, [min, max] denotes the initial search interval using MATLAB function `'fminbnd'`. As shown in this table, the good shape parameters are located between 0 and 1 using various initial search intervals. Despite the inconsistency in the selected shape parameters, we obtain reasonable numerical results.

Table 3.2: RMSE and the near-optimal shape parameter using the LMPS with Gaussian RBFs.

[min, max]	c_{loocv}	RMSE	[min, max]	C_{loocv}	RMSE
$[0,1]$	0.361	4.725E-04	[0, 6]	0.918	1.091E-04
$[0,2]$	0.278	1.515E-04	[0, 7]	0.354	1.040E-03
$[0,3]$	0.846	1.048E-04	[0, 8]	0.351	5.553E-05
$[0,4]$	0.913	1.411E-04	[0, 9]	0.852	4.532E-05
$[0,5]$	0.915	1.312E-04	[0,10]	0.915	2.099E-04

3.4 Numerical results for good variable shape parameter using various strategies

Numerical examples in this section are solved using the MATLAB in a desktop computer which has a 64-bit operating system and 16 GB memory with Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz processor. We use 784 interior and 116 boundary points which are uniformly distributed computational points on a regular square domain $[0,1]\times[0,1]$. Gaussian RBFs and its particular solutions have been used to solve the given PDEs by the MPS as discretized in Eq. (3.3). After the computation of the unknown coefficients α_j, all error plots are plotted for 200 randomly distributed test points. All error plots display the MAE which is defined as in (2.55).

Example 3.4.1. Consider the two dimensional linear elliptic boundary value problem (2.56) and (2.57). The $f(x,y)$ and $g(x,y)$ in (2.56) and (2.57) are obtained by the given analytical solution. We choose three different analytical solutions such as a trigonometric function

$$u_1(x,y) = \sin \pi x \sin \pi y,$$

an exponential function

$$u_2(x,y) = \exp(x+2y),$$

and a relatively flat function

$$u_3(x,y) = \frac{65}{65+(x-0.2)^2+(y+0.1)^2}.$$

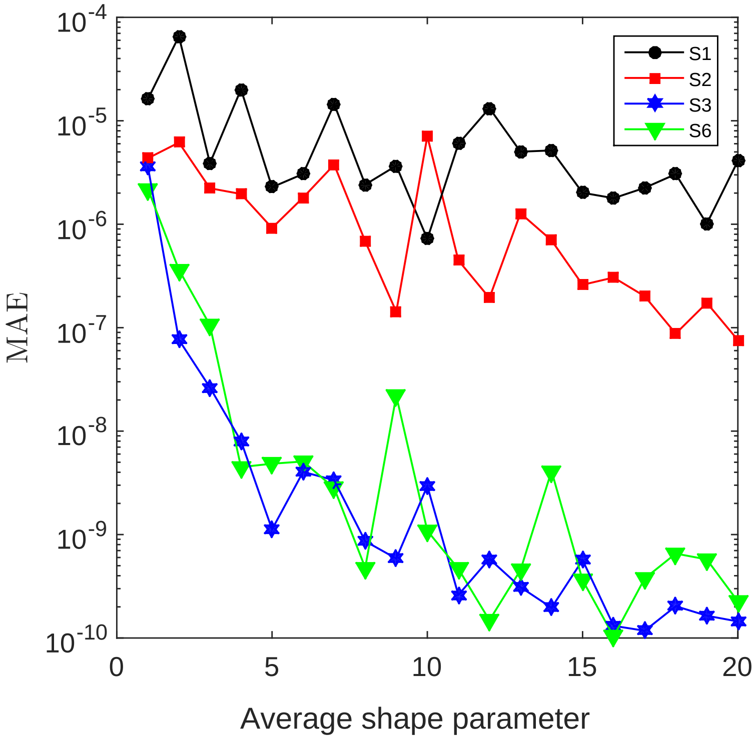

Figure 3.7: MAE obtained for $u_1(x,y)$ by different strategies at $c_{min} = 0.5$ and c_{max} varies.

Figures 3.7 - 3.9 are plotted by using a fixed c_{min} as 0.5 and varies c_{max}. Figure 3.7 shows the MAE for $u_1(x,y)$ at different c_{max}. We observed that the most accurate solution which is about $1E-10$, is obtained with $c_{min} = 0.5$ and $c_{max} = 16$. If we compare the results obtained from these strategies for the three different analytical functions u_1, u_2, u_3, we observe that S3 and S6 have better accurate results than S1 and S2. These accuracies depend on the choice of the search interval, as we can see in Figure 3.8 for $u_2(x,y)$, S3 and S6 have accuracy about $1E-10$ with $c_{min} = 0.5$ and $c_{max} = 18$ and for $u_3(x,y)$ in Figure 3.9 has more accurate results with $c_{max} = 19$.

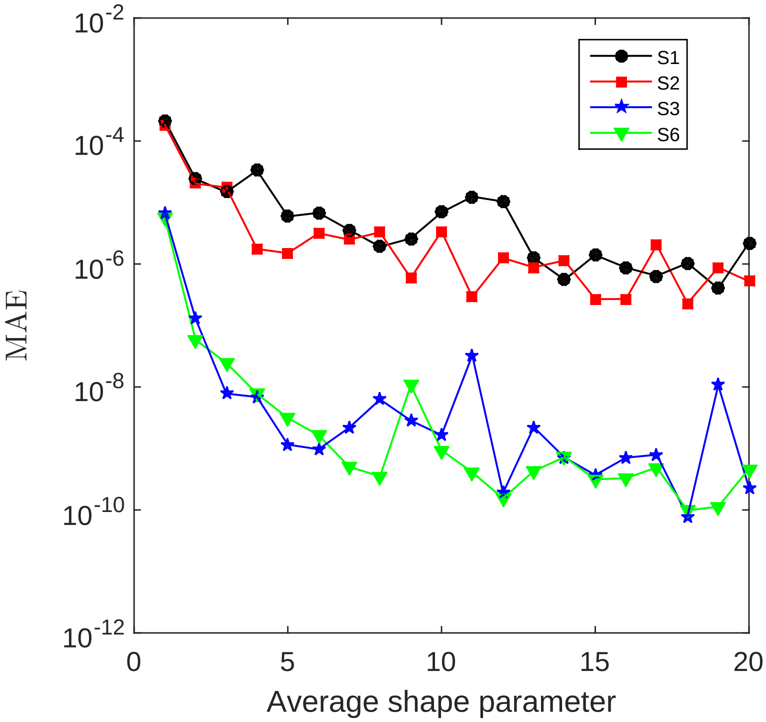

Figure 3.8: MAE obtained for $u_2(x,y)$ by different strategies at $c_{min} = 0.5$ and c_{max} varies.

Table 3.3: MAE for different strategies for different analytical functions for $c_{min} = 0.5$ and $c_{max} = 20$.

Strategies	$u_1(x,y)$	$u_2(x,y)$	$u_3(x,y)$
S1	4.534E-06	2.052E-06	9.632E-10
S2	7.383E-08	7.932E-07	3.008E-11
S3	1.219E-10	1.861E-09	5.768E-13
S6	2.236E-10	4.342E-10	1.266E-13
S7	2.504E-10	1.161E-10	3.184E-13

Table 3.3 shows that the newly proposed strategy S7 which is based on the quasi-random halton numbers, is also an effective way to compute the variable shape parameter as other strategies. It is better than the S1 and S2, so we can conclude that we do not have to stick on the uniformly random numbers such as in S3 but we can use other random number points such as the quasi-random points in S7. Also, the modified strategy S6, shows the similar accuracy along with others [104]. This concludes that if we do not want to use the negative c_j's, then we can use the modified trigonometric varying strategy as in S6. Among these

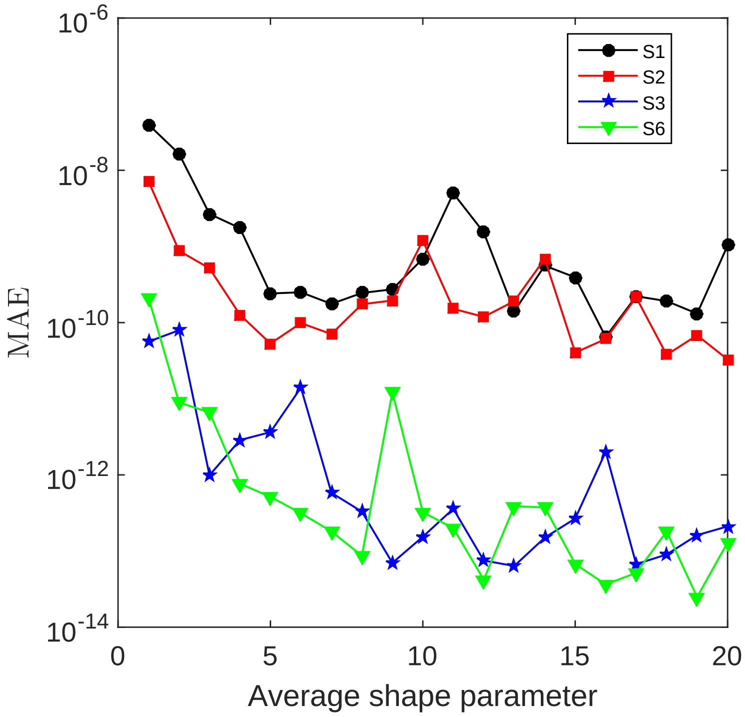

Figure 3.9: MAE obtained for $u_3(x,y)$ by different strategies at $c_{min} = 0.5$ and c_{max} varies.

three analytical functions, almost all of the strategies show less accurate results for the $u_1(x,y)$. So, we now use $u_1(x,y)$ as our test function.

Now, we will see the effectiveness of the strategy S5 for the MPS along with the other newly proposed strategies which are based on the global optimization tools such as GA, PS, and SA. Since all of these methods can be modified for efficiency and accuracy with the help of their corresponding parameters, we have chosen the following parameters for the evaluation of the required variable shape parameter in MPS:

- GA:

  $PopulationSize = 5,$

  $EliteCount = 1,$

  $Generations = 10,$

- PS:

  $MaxIter = 5,$

  $MaxFunEvals = 10,$

$SearchMethod = @MADSPositiveBasisNp1,$

$Initialguess = (1, 1, ..., 1),$

- SA:

$MaxIter = 10,$

$Initialguess = (1, 1, ..., 1),$

and all other parameters are chosen as MATLAB default parameters. These parameters are chosen to stop the searching algorithm within similar amount of time without losing its effectiveness.

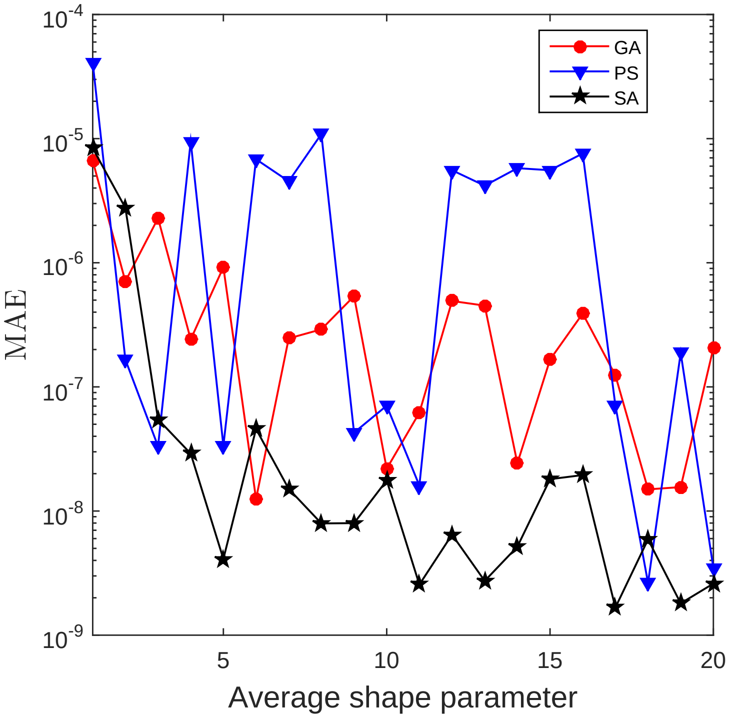

Figure 3.10: MAE obtained for $u_1(x, y)$ by different strategies at $c_{min} = 0.01$ and c_{max} varies.

Figure 3.10 has been generated by the similar structure as above; i.e., we have fixed $c_{min} = 0.01$ and let c_{max} varies. Though GA, PS and SA have less accurate results than S6 and S7, Figure 3.10 depicts that the newly proposed strategy based on SA outperformed GA. Now, instead of using the initial guess $(1, 1, ..., 1)$, we use c_j's obtained from the above strategies S1-S7 as an initial guess in PS and SA. We use $[0.01, 1000]$ as the search interval. An interval of large length is chosen to address the difficulties of choosing good search intervals in previous strategies. If we obtain stable accurate solution within this size of interval, then finding initial search interval will be an easy task. The variable shape parameters chosen by S1 and SA with S1 as an initial guess are shown in the Figure 3.11. It depicts that the variables chosen by SA slightly vary from the S1 but we can see its positive effect on the Table 3.4.

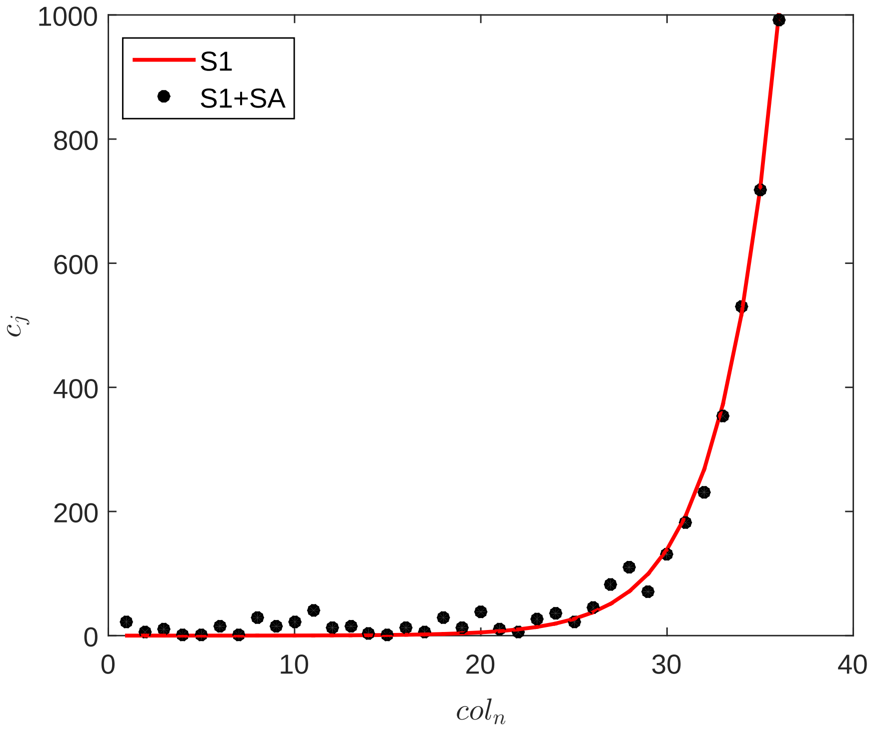

Figure 3.11: Variable shape parameters obtained for $u_1(x,y)$ by S1 and SA with S1 as an initial guess at $c_{min} = 0.01$ and $c_{max} = 1000$.

Table 3.4 shows that the combining idea does not effectively work for other strategies such as S2-S7, but it works very well for S1. In the RBFs literature, S1 is considered to be the less accurate strategy among others but if we use the c_j's obtained from S1 as an initial guess in the PS and SA, we can obtain more accurate results.

By comparing the results among the optimization tools, SA itself seems to be an effective method for the variable selection but how to guess the initial variable is a daunting task. However, if we use the S1 as an initial guess for the SA, we obtain better results as shown in the Table 3.4 and Figure 3.12. Figure 3.12 shows the error profile for S1 and SA with S1 as an initial guess for $c_{min} = 0$ and c_{max} varies up to 1000. Figure 3.13 shows the profile of the error using the S2 as an initial guess does not seem to be a good idea for higher search intervals.

Table 3.4: Maximum Absolute errors for different strategies for the $u_1(x,y)$ with $c_{min} = 0.01$ and $c_{max} = 1000$

Strategies	Alone	PS	SA
S1	9.591E-05	3.203E-07	5.532E-09
S2	5.280E-02	9.31E-02	1.427E-02
S3	1.596E-03	3.47E-03	9.747E-04
S6	5.618E-05	1.214E-05	2.096E-06
S7	1.643E-02	3.753E-03	1.406E-03

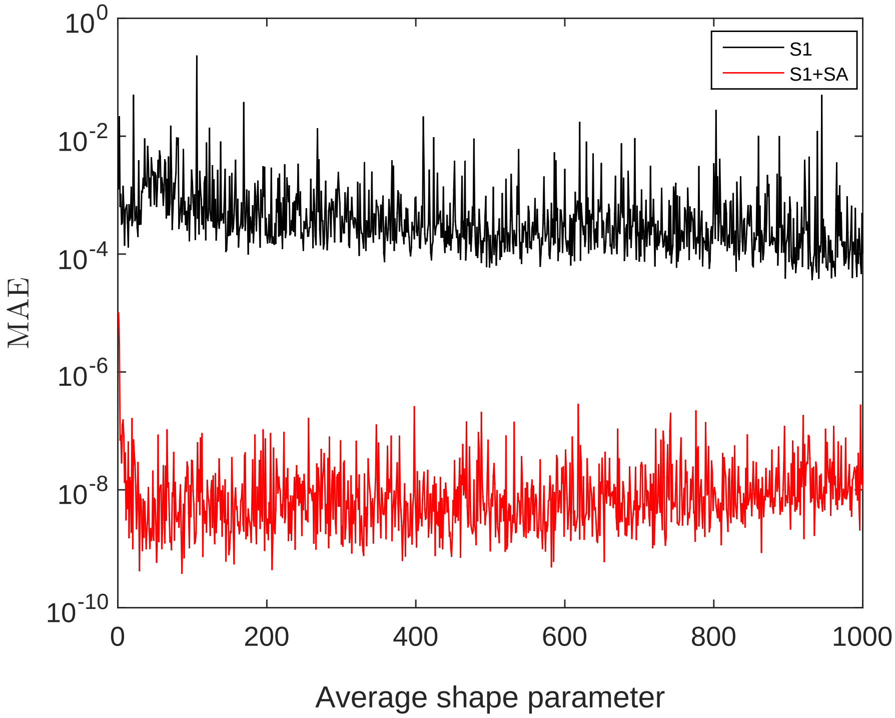

Figure 3.12: MAE obtained for $u_1(x,y)$ by S1 and SA with S1 as an initial guess at $c_{min} = 0$ and c_{max} varies up to 1000.

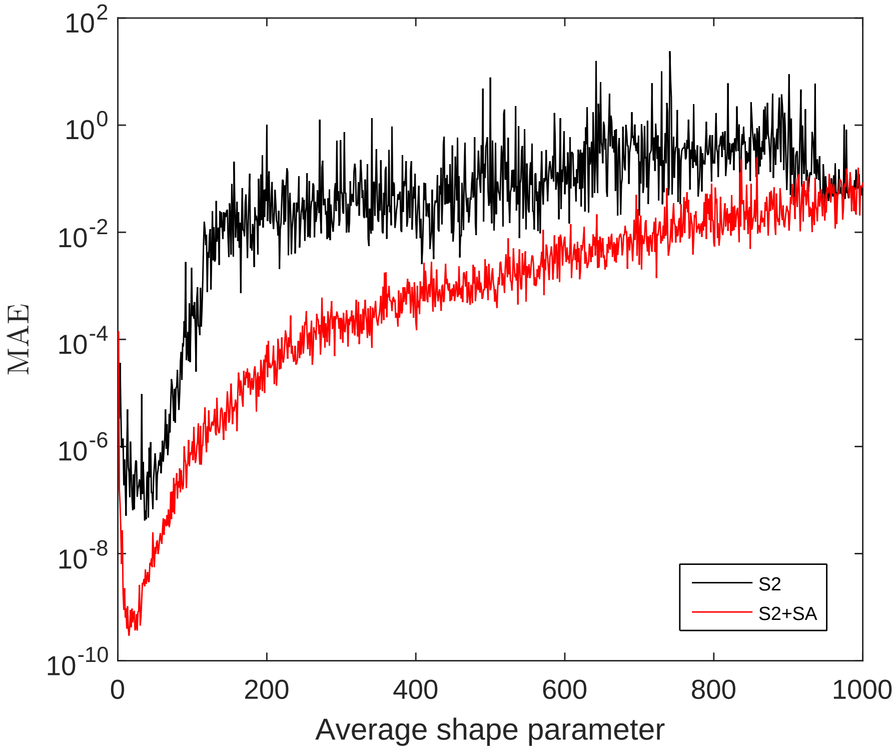

Figure 3.13: MAE obtained for $u_1(x,y)$ by S2 and SA with S2 as an initial guess at $c_{min} = 0$ and c_{max} varies up to 1000.

Chapter 4

Fast method of particular solutions using Chebyshev interpolation

The formulation of the MPS produces a full and dense matrix system which is often very ill-conditioned. Traditionally, this matrix system is solved by using direct or iterative methods. Direct methods such as Gaussian elimination require $O(N^3)$ operations for an $N \times N$ system of equations. For iterative methods, we may obtain the approximate solution in k steps with each step needing a matrix vector multiplication $O(N^2)$. In this chapter, we present the fast method of particular solutions (FMPS) [60] by coupling the FSM and the MPS as a fast algorithm for solving PDEs which require a large number of collocation points. As we shall see in the numerical results section, we have successfully solved a 2D problem using 694,541 collocation points with only 77.15 seconds of computer running time and 343,000 collocation points with 105.15 seconds for a 3D problem. Moreover, we do not compromise the accuracy for our proposed fast computation.

The structure of the chapter is as follows. In Section 4.1, we review the algorithm of fast summation method (FSM). In Section 4.2, specific algorithm has been given. To demonstrate the efficiency of the proposed method, two numerical examples in 2D and 3D are given in the Section 4.3. The research work presented in this chapter is published in [60]. In the next section, we briefly introduce a fast summation method for the matrix vector multiplication.

4.1 Fast summation method (FSM)

Consider the evaluation of the sum of the form

$$s(\boldsymbol{x}) = \sum_{j=1}^{N} b_j \kappa \left(||\boldsymbol{x} - \boldsymbol{x}_j|| \right), \tag{4.1}$$

where κ is either the RBFs or the particular solutions of the corresponding RBFs. We can evaluate the sum (4.1) in an efficient way by using the Chebyshev interpolation technique as described in [38].

From [38], let

$$P_M(\xi, \eta) = \frac{1}{M} + \frac{2}{M} \sum_{i=1}^{M-1} T_i(\xi) T_i(\eta), \tag{4.2}$$

where $\xi,\ \eta \in [-1,1]$, T_i is the first kind Chebyshev polynomial of order i.

Let H be a hypercube in D dimension which contains all the given collocation points $\{\boldsymbol{x}_i\}_{i=1}^{N}$. Consider $\{\boldsymbol{\xi}_l\}_{l=1}^{M^D}$, $\{\boldsymbol{\eta}_l\}_{l=1}^{M^D}$ be two identical sets of Chebyshev points in $[-1,1]^D$. Then by using linear transformations, we can map $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ into $\boldsymbol{\xi}_i$, $\boldsymbol{\eta}_j$, and $\boldsymbol{\xi}_l$, $\boldsymbol{\eta}_m$ in $[-1,1]^D$ into $\boldsymbol{x}_l$, $\boldsymbol{x}_m$ in H, respectively.

Instead of directly evaluating (4.1), we approximate it by using the following functional approximation

$$\kappa(||\boldsymbol{x}-\boldsymbol{y}||) = \sum_l \sum_m k(||\boldsymbol{x}_l-\boldsymbol{y}_m||)\, Q_M(\boldsymbol{\xi}_l,\boldsymbol{\xi}) Q_M(\boldsymbol{\eta}_m,\boldsymbol{\eta}), \tag{4.3}$$

where $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ are the linear transformations of $\boldsymbol{x},\boldsymbol{y}$, respectively, and in 2D,

$$Q_M(\boldsymbol{\xi},\boldsymbol{\eta}) = P_M(\xi_1,\eta_1)P_M(\xi_2,\eta_2), \quad \boldsymbol{\xi} = (\xi_1,\xi_2),\ \boldsymbol{\eta} = (\eta_1,\eta_2),$$

and in 3D,

$$Q_M(\boldsymbol{\xi},\boldsymbol{\eta}) = P_M(\xi_1,\eta_1)P_M(\xi_2,\eta_2)P_M(\xi_3,\eta_3), \quad \boldsymbol{\xi} = (\xi_1,\xi_2,\xi_3),\ \boldsymbol{\eta} = (\eta_1,\eta_2,\eta_3).$$

Note that the functional approximation (4.3) is obtained from approximating $\kappa(||\boldsymbol{x}-\boldsymbol{y}||)$ by first fixing the variable y and assuming $\kappa(||\boldsymbol{x}-\boldsymbol{y}||)$ as a function of x; i.e.,

$$\kappa(||\boldsymbol{x}-\boldsymbol{y}||) = \sum_l k(||\boldsymbol{x}_l-\boldsymbol{y}||)\, Q_M(\boldsymbol{\xi}_l,\boldsymbol{\xi}), \tag{4.4}$$

and noting that $\kappa(||\boldsymbol{x}-\boldsymbol{y}||)$ is a function of y, the interpolation formula is applied again to get Eq. (4.3).

Using (4.3), (4.1) can be separated into the product of the different sums as follows:

$$\begin{aligned} s(\boldsymbol{x}_i) &= \sum_{j=1}^{N} b_j \kappa\left(||\boldsymbol{x}_i-\boldsymbol{x}_j||\right) \\ &\approx \sum_{j=1}^{N} b_j \left(\sum_{l=1}^{M^D} \sum_{m=1}^{M^D} \kappa(||\boldsymbol{x}_l-\boldsymbol{x}_m||)\, Q_M(\boldsymbol{\xi}_l,\boldsymbol{\xi}_i) Q_M(\boldsymbol{\eta}_m,\boldsymbol{\eta}_j) \right) \\ &= \sum_{l=1}^{M^D} Q_M(\boldsymbol{\xi}_l,\boldsymbol{\xi}_i) \sum_{m=1}^{M^D} \kappa(||\boldsymbol{x}_l-\boldsymbol{x}_m||) \sum_{j=1}^{N} b_j Q_M(\boldsymbol{\eta}_m,\boldsymbol{\eta}_j). \end{aligned} \tag{4.5}$$

If we choose those M^D Chebyshev points such that $M^D << N$, then the summation can be computed efficiently. To evaluate (4.1) approximately, let us first evaluate the last term of (4.5) which gives us the weights at the Chebyshev points $\boldsymbol{x}_m$

$$W_m = \sum_{j=1}^{N} b_j Q_M(\boldsymbol{\eta}_m,\boldsymbol{\eta}_j), \ \ m = 1,\ldots,M^D,$$

and the computational cost for these Chebyshev weights is $O(M^D n)$. Then, we compute the $s(\boldsymbol{x})$ at the Chebyshev points $\boldsymbol{x}_l$ as

$$s(\boldsymbol{x}_l) = \sum_{m=1}^{M^D} W_m \kappa(||\boldsymbol{x}_l - \boldsymbol{x}_m||), \;\; l = 1, ..., M^D,$$

with computational cost $O(M^{2D})$. Eventually, we compute the $s(\boldsymbol{x})$ at the collocation points $\boldsymbol{x}_i$ as

$$s(\boldsymbol{x}_i) = \sum_{l=1}^{M^D} s(\boldsymbol{x}_l) Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i), \;\; i = 1, ..., N,$$

at the computational cost of $O(M^D N)$. Overall, this algorithm scales like $O((2 * M^D)N + (M^{2D}))$. If we choose $M << (N/(1+\sqrt{2}))^{1/D}$, then the above mentioned algorithm is faster than $O(N^2)$.

In the next section, we give a description to accelerate the MPS using this fast summation technique.

4.2 Fast method of particular solutions (FMPS)

The matrix $\mathbf{A}$ of the linear system $\mathbf{A}\boldsymbol{\alpha} = \mathbf{F}$, arising from the MPS discretization in (1.6) – (1.7), can be viewed as the formulation of two block matrices; i.e.,

$$\mathbf{A} = \begin{bmatrix} \mathbf{P_A} \\ \mathbf{Q_A} \end{bmatrix},$$

where

$$\mathbf{P_A} = \left[\phi(||\boldsymbol{x}_i - \boldsymbol{x}_j||)\right]_{1 \le i \le n_i, 1 \le j \le N}, \;\; \mathbf{Q_A} = \left[\Phi(||\boldsymbol{x}_i - \boldsymbol{x}_j||)\right]_{n_i+1 \le i \le N, 1 \le j \le N}.$$

Hence, if we want to multiply $\mathbf{A}$ by any N-dimensional column vector $\mathbf{B} = [b_1, b_2, ..., b_N]^T$, then this can be done by

$$\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} \mathbf{P_A} \cdot \mathbf{B} \\ \mathbf{Q_A} \cdot \mathbf{B} \end{bmatrix}, \tag{4.6}$$

and each row of those individual products $\mathbf{P_A} \cdot \mathbf{B}$, $\mathbf{Q_A} \cdot \mathbf{B}$ can be considered like the summation defined in (4.1) which can be computed in an efficient way by using the FSM described in the previous section.

If we use some iterative methods to solve the linear system instead of using the direct solvers like Gaussian elimination, then these fast products help us to solve the linear system faster. For this, we can use any iterative method which includes the vector multiplication with matrix $\mathbf{A}$. In this paper, we have used GMRES iterative method to find out the unknown coefficients. As we know that, in the GMRES, at each iteration, we have to multiply a matrix and an updated vector. This matrix and vector multiplication is obtained faster by the technique that we have described earlier. This kind of approach which is called FMPS, actually solves the PDEs by using MPS without explicitly computing matrix **A** which requires a minimum storage. The computational procedure for the FMPS is described as follows.

4.2.1 Algorithm

Input:

Collocation points $\{\boldsymbol{x}_i\}_{i=1}^{N}$, Chebyshev points $\boldsymbol{\xi}_l, \boldsymbol{\eta}_m$, dimension of the domain D.

Step 1: Pre-computational step

- find a hypercube H in D dimension which contains $\{\boldsymbol{x}_i\}_{i=1}^{n}$.
- by using linear transformations,

  find $\boldsymbol{\xi}_i, \boldsymbol{\eta}_j$ in $[-1,1]^D$ corresponding to $\boldsymbol{x}_i, \boldsymbol{x}_j$, respectively,

  find $\boldsymbol{x}_l, \boldsymbol{x}_m$ in H corresponding to $\boldsymbol{\xi}_l, \boldsymbol{\eta}_m$, respectively.
- compute the following matrices

$$\begin{aligned}
\mathbf{R} &= (R)_{m,j} = Q_M(\boldsymbol{\eta}_m, \boldsymbol{\eta}_j),\ m = 1,2,...,M^D,\ j = 1,2,...,N,\\
\mathbf{I_R} &= (I_R)_{l,i} = Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i),\ l = 1,2,...,M^D,\ i = 1,2,...,n_i,\\
\mathbf{B_R} &= (B_R)_{l,i} = Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i),\ l = 1,2,...,M^D,\ i = n_i+1, n_i+2,...,N,\\
\mathbf{I_K} &= (I_K)_{l,m} = \phi(||\boldsymbol{x}_l - \boldsymbol{x}_m||),\ l = 1,2,...,M^D,\ m = 1,2,...,M^D,\\
\mathbf{B_K} &= (B_K)_{l,m} = \Phi(||\boldsymbol{x}_l - \boldsymbol{x}_m||),\ l = 1,2,...,M^D,\ m = 1,2,...,M^D.
\end{aligned}$$

Step 2: Iterative step

Iterative methods converge in k steps with each step needing a matrix vector multiplication $O(N^2)$. At each iterative step k, we follow the iterative algorithm as it is except the matrix vector multiplication of the matrix **A** and the N-dimensional updated column vector **B**, which is computed by using the FSM procedure in the vectorized form as follows:

First, compute

$$\mathbf{R} \cdot \mathbf{B} = \mathbf{M} \in \mathbb{R}^{M^D \times 1},$$

and then compute

$$\begin{aligned}\mathbf{I_M} &= \mathbf{I_R}' \cdot (\mathbf{I_K} \cdot \mathbf{M}),\\ \mathbf{B_M} &= \mathbf{B_R}' \cdot (\mathbf{B_K} \cdot \mathbf{M}),\end{aligned}$$

which approximates (4.6) as

$$\mathbf{P_A} \cdot \mathbf{B} \approx \mathbf{I_M} \text{ and } \mathbf{Q_A} \cdot \mathbf{B} \approx \mathbf{B_M},$$

i.e.,

$$\mathbf{A} \cdot \mathbf{B} \approx \begin{bmatrix} \mathbf{I_M} \\ \mathbf{B_M} \end{bmatrix}.$$

The undetermined coefficient $\boldsymbol{\alpha}$ in (5.10) can be obtained by the iterative method.

Step 3: Evaluation step

Once we determine the unknown coefficient vector $\boldsymbol{\alpha}$ from step 2, we use the given n_t test points and RBFs centers to evaluate the desired solution at the test points. This is also done by multiplying the corresponding evaluation matrix with the coefficient vector $\boldsymbol{\alpha}$ using the similar FSM technique.

As we can see, in this procedure we are not computing matrix $\mathbf{A}$ explicitly. The only matrix evaluation we have to do is for the matrices described in the pre-computational step. Now, in the next section, we give some numerical examples, which validate our proposed method.

4.3 Numerical results

Numerical experiments have been done by using MATLAB$^{©}$ on a desktop computer which has a 64-bit operating system and 16 GB memory with Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz processor. The RMSE as defined in Eq. (2.54) is used to measure the accuracy of the numerical results.

We adopt the Gaussian, ϕ, as the RBFs and the corresponding particular solutions, Φ, as the basis functions for the approximation of the partial differential equation. The particular solutions of the Gaussian RBFs in (2.37) and (2.45) contain the special functions, $\mathrm{Ei}(x)$ and $\mathrm{erf}(x)$, which are costly in terms of numerical evaluation. The efficiency can be significantly improved using compiled MATLAB MEX functions. The special function exponential integral in the particular solutions of the Gaussian RBFs [58] are evaluated using the rational approximation techniques [29] with the help of the boost library [96].

Example 4.3.1. Consider the convection-diffusion-reaction equation in 2D

$$\begin{aligned}\Delta u(\boldsymbol{x})+f(\boldsymbol{x})\frac{\partial u}{\partial x}+g(\boldsymbol{x})\frac{\partial u}{\partial y}+h(\boldsymbol{x})u(\boldsymbol{x}) &= l(x,y),\ \ \boldsymbol{x}=(x,y)\in\Omega,\\ u(\boldsymbol{x}) &= m(x,y),\ \ \boldsymbol{x}=(x,y)\in\partial\Omega,\end{aligned}$$

where $f(x,y)=y\cos(y)$, $g(x,y)=\sinh(x)$, $h(x,y)=x^2+y^2$, and $l(x,y)$, $m(x,y)$ are obtained by the analytic solution

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

The above differential equation has been considered in Reference [23]. For the computational domain, we consider the standard unit square $\bar{\Omega}=\Omega\cup\partial\Omega=[0,1]^2$. To perform the test, we choose various numbers of collocation points and 1000 randomly distributed test points inside the domain to evaluate the RMSE errors. For the FSM, we employ 12×12 Chebyshev points and the shape parameter for the Gaussian RBFs is chosen as 1. To show the efficiency of the proposed method, we test the MPS using standard Gaussian elimination (GE), iterative method (GMRES), and the proposed FSM with GMRES. In Table 5.2, we show the proposed method is far more efficient than the direct Gaussian elimination and iterative method. As far as the accuracy is concerned, they all produce a similar accuracy. When the number of collocation points is increased, the computational cost using the two traditional linear solvers become too expensive. To illustrate the efficiency of the proposed method, we choose a huge number of collocation points up to one million as shown in Table 4.2. As we can see in the table, for the case of one million collocation points, the required CPU time is only 56.90 seconds which is considered to be extremely fast. Note that the tolerance of GMRES is set as 1E-05 in the above computation.

Table 4.1: RMSE and CPU time using various numbers of the collocation points and solvers in the square domain.

N	RMSE	GE	GMRES	FSM + GMRES
40^2	6.163E-05	0.32	0.34	0.05
60^2	6.637E-05	1.79	1.50	0.13
80^2	3.740E-05	6.28	4.70	0.20
100^2	2.751E-04	18.25	11.36	0.35
120^2	1.813E-04	42.86	23.29	0.40

Table 4.2: RMSE and CPU time for a large number of collocation points in the square domain using the FMPS.

N	RMSE	CPU time
250^2	1.124E-04	1.65
500^2	8.197E-05	11.04
750^2	1.709E-04	35.00
1000^2	2.964E-04	56.90

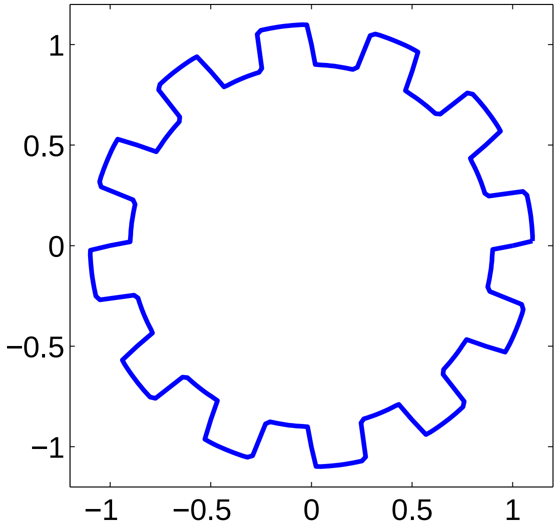

Figure 4.1: The profile of gear-shaped domain.

Next, we consider the gear-shape domain as shown in Figure 4.1. The parametric equation of the gear-shape curve can be written as

$$\{(x,y) : x = r(t)\cos t, y = r(t)\sin t, \quad 0 \leq t \leq 2\pi\},$$

where

$$r(t) = 1 + \frac{1}{10}\tanh(10\sin(12t)).$$

In this case, we choose 969 randomly distributed test points inside the domain for the evaluation of the RMSE. Table 4.3 shows the RMSE and the CPU time for various combinations of interior and boundary points using the FMPS. In this case, the tolerance of GMRES is again set as 1E-05. Table 4.3 validates the efficiency of the FMPS. Despite a large number of collocation points are used in this example, the numerical results are two orders less accurate than the results obtained in Reference [23]. Note that the large number of collocation does not necessary produce more accurate results due to the ill-conditioning of the resultant matrix and round-off errors. Furthermore, to increase the efficiency, the iterate method GMRES has been adopted in our algorithm. To increase the accuracy, we need to reduce the tolerance of stopping criteria of GMRES which will inevitably slow the computation. There is a trade-off between accuracy and efficiency in the proposed fast algorithm.

Table 4.3: RMSE and CPU time using various numbers of the collocation points for the gear-shaped domain.

(n_i, n_b)	RMSE	CPU (sec)
$(5195, 1000)$	8.860E-05	0.52
$(14552, 3000)$	8.382E-05	1.23
$(25953, 5000)$	8.851E-05	1.97
$(58623, 10000)$	9.371E-05	3.29
$(104405, 20000)$	8.907E-05	6.01
$(163297, 25000)$	9.833E-05	17.81
$(654541, 40000)$	1.591E-04	77.15

Example 4.3.2. Consider the boundary value problem (2.60)-(2.61) in 3D, where f and g are defined according to the exact solution

$$u(x,y,z) = \sin(\pi x)\sin(\pi y)\sin(\pi z).$$

Table 4.4: RMSE and CPU time for different sizes of the computational points in the unit cube.

N	RMSE	GE	GMRES	FSM+GMRES
13^3	1.804E-05	0.31	0.64	1.09
15^3	3.156E-05	0.76	1.20	1.44
17^3	6.025E-05	1.77	1.57	1.47
19^3	3.898E-05	4.18	3.08	1.66
21^3	1.217E-04	8.40	4.05	1.52
23^3	1.494E-04	17.42	6.88	1.97
25^3	3.118E-04	33.46	11.08	2.41
27^3	2.204E-04	64.39	17.64	3.04

Table 4.4 shows the accuracy and CPU time using the three techniques including the FMPS for the Poisson equation with Dirichlet boundary condition. The tolerance for the

Table 4.5: RMSE and CPU time for various numbers of the collocation points in the unit cube by FMPS.

N	RMSE	CPU (sec)
30^3	3.514E-05	7.04
40^3	5.253E-05	14.92
50^3	3.717E-05	29.96
60^3	5.219E-05	62.035
70^3	6.037E-05	105.15

GMRES is set as in the two dimensional case and 12^3 Chebyshev points in the 3D have been used for the FSM. As shown in Table 4.4, the FMPS is far more efficient than the other two traditional approaches. It is noteworthy that in Table 4.5, the FMPS needs only 105.15 seconds to solve the given problem with $70^3 = 343,000$ collocation points.

Chapter 5

Solving time fractional diffusion equations by the FMPS

In this chapter, FMPS [60] is implemented for the numerical simulation of the time fractional diffusion equations. We use the Laplace transform technique to obtain the corresponding time-independent inhomogeneous equation in Laplace space and then implement FMPS [60] to solve this Laplace transformed problem. Finally, Talbot algorithm [1, 93] which is a numerical inverse Laplace transform (NILT) is implemented to retrieve the numerical solutions of the time fractional diffusion equations from the Laplace space. Fractional order differential equations [84] have been recently proved to be valuable tools for the modeling of many phenomena in engineering, physics, fluid mechanics, viscoelasticity, mathematical biology, electrochemistry and others [48, 84]. Various fractional order differential equations have been solved recently including space-time fractional partial differential equations [19, 72, 81, 110], fractional order two-point boundary value problem [37], the fractional Kdv equation [30], and fractional partial differential equations fluid mechanics [82].

Fractional kinetic equations such as fractional diffusion equation, fractional advection-diffusion equation, fractional Fokker-Plank equation, fractional cable equation etc., are recognized as useful approaches for the description of transport dynamics in complex systems including systems exhibiting Hamiltonian chaos, disordered medium, plasma, and fluid turbulence, underground water pollution, dynamics of protein molecules, motions under the influence of optical tweezers, reactions in complex systems and more [18, 75, 76, 111].

Fractional diffusion equation [84] is considered as a recent alternative model to describe anomalous diffusion phenomena [75] in a wide range of engineering and physics fields [48] such as electron transportation [90], seepage [45], magnetic plasma [31], dissipation [95], and turbulence [92]. Compared with normal diffusion phenomena, anomalous diffusion exhibits the striking characteristics of the long-range interaction and history dependency [44].

In most of the meshless numerical schemes that have been used to solve the fractional diffusion equations, FDM is used for temporal discretization and different meshless techniques are applied to find the numerical solution of the time fractional diffusion equations. Wen Chen et. al [27] makes the first attempt to apply the Kansa method in the solution of the time fractional diffusion equations in which the MQ and thin plate splines serve as the RBFs. In [27], FDM scheme is used to discretize time fractional derivative and Kansa method is used for spatial derivatives. In [70], Q. Liu et al, presented an implicit meshless approach based on RBFs for the numerical simulation of time fractional diffusion equations in which locally supported MQ are used to construct the meshless shape function based on the local interpolation domains. Another implicit meshless approach based on the moving least squares (MLS) approximation is also used for numerical simulation of fractional advection-diffusion equations in [112]. Recently, along with the FDM scheme, MPS has also been used to solve the constant and variable order fractional diffusion models [40] and time-fractional diffusion equations with a non-local boundary condition [105]. All of these researchers used FDM as temporal discretization and used different shape functions and meshless strong-forms to obtain the discrete system of equations. Ge Hong-Xia et. al [50] use EFG method based on the MLS approximation. They used Galerkin Weak form to obtain the discrete equations, and the essential boundary conditions are enforced by the penalty method.

In order to overcome the drawback of FDM for temporal discretization, Zhuo-Jia Fu et. al [42] use a boundary meshless method which does not use the FDM scheme to discretize the time. Instead of discretizing the time, they used Laplace transform technique known as Laplace transformed boundary particle method (LTBPM) to obtain the corresponding time independent inhomogeneous equation in the Laplace space. After transforming the time dependent problems into time-independent problems, a truly boundary-only meshless boundary particle method (BPM) is used to solve Laplace-transformed problem. Finally, numerical inverse Laplace transform (NILT) is implemented to retrieve the numerical solutions of time fractional diffusion equations from the corresponding BPM solutions. In comparison with finite difference discretization, the LTBPM introduces Laplace transform and NILT algorithm to deal with time fractional derivation term.

We note that LTBPM does not use temporal discretization but it requires boundary-only meshless discretization of inhomogeneous problem. In this dissertation, we try to overcome this drawback by using truly meshless method, MPS and its fast version FMPS [60] which is useful for the large number of collocation points. We use similar approach in LTBPM but instead of using BPM which is a boundary-only meshless method, we use MPS and FMPS [60]. Numerical results suggest that MPS has comparable accuracy with LTBPM but FMPS [60] with large number of collocation points produces more accurate solutions.

The rest of the chapter is organized as follows. Section 5.1 introduces the numerical method for solving time fractional diffusion equations followed by numerical results and discussion in Section 5.2.

5.1 Methodology

5.1.1 Time fractional diffusion equation

In the current work, we have considered the following time fractional diffusion equations in a bounded domain Ω with the boundary $\partial\Omega$,

$$\frac{\partial^\alpha u(\mathbf{x},t)}{\partial t^\alpha} = \Delta u(\mathbf{x},t) + f(\mathbf{x},t),\ \ 0<\alpha<1, \mathbf{x}\in\Omega,\ \ t\in(0,T), \tag{5.1}$$

with boundary condition

$$u(\mathbf{x},t) \ \ = \ \ g(\mathbf{x},t),\ \ \mathbf{x}\in\partial\Omega,\ \ t\in(0,T), \tag{5.2}$$

and initial condition

$$u(\mathbf{x},0) = u_0(x),\ \ \mathbf{x}\in\Omega, \tag{5.3}$$

where Δ is the Laplace operator, $f(\mathbf{x},t),\ g(\mathbf{x},t)$, and $u_0(\mathbf{x})$ are known functions, $\mathbf{x}=(x,y)$ for 2D problem and $\mathbf{x}=(x,y,z)$ for 3D problem, T the total time to be considered, and $\frac{\partial^\alpha}{\partial t^\alpha}$ the Caputo fractional derivative of order α with respect to t defined by [84]

$$\frac{\partial^\alpha u(\mathbf{x},t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{\partial u(\mathbf{x},\eta)}{\partial\eta}\frac{d\eta}{(t-\eta)^\alpha},\ 0<\alpha<1. \tag{5.4}$$

5.1.2 Numerical method

In this section, we describe the numerical procedure for solving the time fractional diffusion equations by using Laplace transform and MPS/FMPS [60]. As we mentioned earlier, there are three steps to follow:

Step 1: convert the time fractional diffusion problems from time domain to Laplace domain,

Step 2: use MPS/FMPS to obtain the solution in Laplace domain,

Step 3: invert the solution into original domain by numerical inverse Laplace transform.

Step 1:

Applying the Laplace transform to (5.1) - (5.3) produces

$$\begin{aligned} \mathcal{R}\tilde{u}(\mathbf{x},s) &= \tilde{F}(\mathbf{x},s),\ 0<\alpha<1,\ \mathbf{x}\in\Omega, \\ \tilde{u}(\mathbf{x},s) &= \tilde{g}(\mathbf{x},s),\ \mathbf{x}\in\partial\Omega, \end{aligned} \tag{5.5}$$

where $\mathcal{R}=\Delta-s^\alpha, \tilde{F}(\mathbf{x},s) = -\tilde{f}(\mathbf{x},s) - s^{\alpha-1}u_0(x)$. This is the Laplace transformed problem of the time fractional diffusion equations.

Step 2

Once the problem is transformed into the Laplace domain, we use MPS/FMPS to solve the Laplace transformed time-independent inhomogeneous problems obtained from Step 1.

Method of particular solutions (MPS): Let us consider the Laplace transformed problem from (5.5). By the MPS as in (1.3) for each s, we assume the solution to (5.5) is approximated by

$$\tilde{u}(\mathbf{x},s) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \Phi(\|\boldsymbol{x}-\boldsymbol{x}_j\|), \tag{5.6}$$

where $\|\cdot\|$ is the Euclidean norm, $\{\alpha_j\}$ are the undetermined coefficients, and

$$\Delta\Phi = \phi. \tag{5.7}$$

By the collocation method, from (5.5), we have

$$\sum_{j=1}^{N} \alpha_j \phi\left(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\right) - s^{\alpha} \sum_{j=1}^{N} \alpha_j \Phi\left(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\right) = \tilde{F}(\boldsymbol{x}_i,s), \quad 1 \le i \le n_i, \tag{5.8}$$

$$\sum_{j=1}^{N} \alpha_j \Phi\left(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\right) = \tilde{g}(\boldsymbol{x}_i,s), \quad n_i+1 \le i \le N. \tag{5.9}$$

From (5.8) – (5.9), we can formulate a linear system of equations

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{F}, \tag{5.10}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_{11}} \\ \mathbf{A_{21}} \end{bmatrix},$$

$$\begin{aligned}
\mathbf{A_{11}} &= \left[\phi\left(\|\boldsymbol{x_i}-\boldsymbol{x}_j\|\right) - s^{\alpha}\Phi\left(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\right)\right]_{ij}, \quad 1 \le i \le n_i, 1 \le j \le N,\\
\mathbf{A_{21}} &= \left[\Phi\left(\|\boldsymbol{x}_k-\boldsymbol{x}_j\|\right)\right]_{kj}, \quad n_i+1 \le k \le N, 1 \le j \le N,\\
\boldsymbol{\alpha} &= [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_N]^T,\\
\mathbf{F} &= [\tilde{F}(\boldsymbol{x}_1,s) \ \cdots \ \tilde{F}(\boldsymbol{x}_{n_i},s) \ \tilde{g}(\boldsymbol{x}_{n_i+1},s) \ \cdots \ \tilde{g}(\boldsymbol{x}_N,s)]^T.
\end{aligned}$$

Once we solved the system of equations (5.10), we can determine the unknown coefficients $\{\alpha_j\}_{j=1}^{N}$ and then using (5.6) we can approximate the solution in the Laplace domain for each s.

If we implement the MPS using conditionally positive definite polyharmonic splines radial basis functions (PS-RBFs) for solving the Laplace transformed problem (5.5), then we assume that the solution to (5.5) is approximated by the following formulation:

$$\tilde{u}(\mathbf{x},s) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \Phi(\|\boldsymbol{x}-\boldsymbol{x}_j\|) + \sum_{d=1}^{q} \alpha_{N+d} p_d(\boldsymbol{x}), \ \boldsymbol{x} \in \Omega, \tag{5.11}$$

where $p_d(\boldsymbol{x}) = x^d y^{q-d}, \quad 1 \le d \le q,\ q$ is the number of additional polynomial basis functions, $\Delta\Phi(r) = \phi(r)$ (see Appendix A.1), and $\phi(r) = r^{2m}\ln(r),\ m \in \mathbb{N}$.

By the collocation method, for $1 \le i \le n_i$, from (5.5) we have,

$$\left[\sum_{j=1}^{N} \alpha_j \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) + \sum_{d=1}^{q} \alpha_{N+d} q_d(\boldsymbol{x}_i)\right] - s^{\alpha}\left[\sum_{j=1}^{N} \alpha_j \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) + \sum_{d=1}^{q} \alpha_{N+d} P_d(\boldsymbol{x}_i)\right] = \tilde{F}(\boldsymbol{x}_i, s),$$
$$\sum_{j=1}^{N} \alpha_j \left[\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - s^{\alpha}\Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)\right] + \sum_{d=1}^{q} \alpha_{N+d}\left[q_d(\boldsymbol{x}_i) - s^{\alpha} P_d(\boldsymbol{x}_i)\right] = \tilde{F}(\boldsymbol{x}_i, s), \tag{5.12}$$

where $\Delta p_d(\boldsymbol{x}) = q_d(\boldsymbol{x}),\ 1 \le d \le q$. Also, for $n_i + 1 \le i \le N$ in (5.5) we have,

$$\sum_{j=1}^{N} \alpha_j \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) + \sum_{d=1}^{q} \alpha_{N+d} P_d(\boldsymbol{x}_i) = \tilde{g}(\boldsymbol{x}_i, s). \tag{5.13}$$

Since there are 'q' additional degrees of freedoms, the standard polynomial insolvency constraints must be applied. Thus, for $1 \le d \le q$, we have,

$$\sum_{j=1}^{n_i} \alpha_j q_d(\boldsymbol{x}_j) = 0, \tag{5.14}$$
$$\sum_{j=n_i+1} \alpha_j p_d(\boldsymbol{x}_j) = 0.$$

Then (5.12) – (5.14) produce the linear system

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{F}, \tag{5.15}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \\ \mathbf{A_{31}} & \mathbf{0} \end{bmatrix}, \tag{5.16}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F_1} \\ \mathbf{F_2} \\ \mathbf{0} \end{bmatrix}, \tag{5.17}$$

$$\mathbf{A_{11}} = \left[\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) - s^{\alpha}\Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)\right]_{ij}, 1 \le i \le n_i,\ 1 \le j \le N,$$
$$\mathbf{A_{21}} = \left[\Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)\right]_{ij}, n_i + 1 \le i \le N,\ 1 \le j \le N,$$
$$\mathbf{A_{12}} = q_d(\boldsymbol{x}_i) - s^{\alpha} p_d(\boldsymbol{x}_i), 1 \le i \le N_i,\ 1 \le d \le q,$$
$$\mathbf{A_{22}} = p_d(\boldsymbol{x}_i), n_i + 1 \le i \le N,\ 1 \le d \le q,$$
$$\mathbf{A_{31}} = \begin{bmatrix} \mathbf{A_{12}} \\ \mathbf{A_{22}} \end{bmatrix}^T.$$

Once we solve the system (5.10) or (5.15) depending on which RBFs we choose, we determine the unknown coefficients $\{\alpha_j\}_{j=1}^N$ and then using the corresponding equation (5.6) or (5.11) we approximate the solution in the Laplace domain for each s.

Fast method of particular solutions (FMPS): If we implement the FMPS using PS-RBFs, then we note that from the Chapter 4, we need to multiply the matrix **A** in (5.16) with an updated vector **B** by using FSM inside each iteration of any iterative method. This can be done in the following way:

Suppose **A** is a block matrix with the following block structure

$$A = \left[\begin{array}{c|c} \mathbf{A_{11}} & \mathbf{A_{12}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \\ \hline \mathbf{A_{31}} & \mathbf{0} \end{array}\right]. \tag{5.18}$$

The matrix multiplication of the matrix **A**, with a vector

$$\mathbf{B} = \left[\begin{array}{c} \mathbf{b_1} \\ \mathbf{b_2} \end{array}\right],$$

can be obtained in each row by

$$\mathbf{A_{11}b_1} + \mathbf{A_{12}b_2},$$

$$\mathbf{A_{21}b_1} + \mathbf{A_{22}b_2},$$

$$\mathbf{A_{31}b_1}.$$

The product of the upper left side of the block matrix with $\mathbf{b_1}$ is obtained by using the FSM technique in each iteration of the iterative method. Remaining products are computed in a usual way.

In fact, we can adopt a different approach which is described as follows:

For the polynomial function, we can use similar functional approximation as in (4.3) by applying the similar interpolation formula as in (4.4); i.e.,

$$p_d(\boldsymbol{x}) = \sum_{d=1}^{n} p_d(\boldsymbol{x}_l) Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}). \tag{5.19}$$

If we implement functional approximation (4.3), (5.19), to find the product of the forms

$$s(\boldsymbol{x}_i) = \sum_{j=1}^{N} \kappa\left(||\boldsymbol{x}_i - \boldsymbol{x}_j||\right) b_j + \sum_{d=1}^{q} p_d(\boldsymbol{x}) b_{N+d}, \tag{5.20}$$

then as in (4.5),

$$
\begin{aligned}
s(\boldsymbol{x}_i) &= \sum_{j=1}^{N} \kappa\left(||\boldsymbol{x}_i - \boldsymbol{x}_j||\right) b_j + \sum_{d=1}^{q} p_d(\boldsymbol{x}_i) b_{N+d}, \\
&\approx \sum_{j=1}^{N} b_j \left(\sum_{l=1}^{M^D} \sum_{m=1}^{M^D} \kappa(||\boldsymbol{x}_l - \boldsymbol{x}_m||) Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i) Q_M(\boldsymbol{\eta}_m, \boldsymbol{\eta}_j) \right) + \sum_{d=1}^{q} b_{N+d} \left(\sum_{d=1}^{q} p_d(\boldsymbol{x}_i) Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i) \right), \\
&= \left[\sum_{l=1}^{M^D} Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i) \sum_{m=1}^{M^D} \kappa(||\boldsymbol{x}_l - \boldsymbol{x}_m||) \sum_{j=1}^{N} b_j Q_M(\boldsymbol{\eta}_m, \boldsymbol{\eta}_j) \right] + \sum_{d=1}^{q} b_{N+d} \left[\sum_{d=1}^{q} p_d(\boldsymbol{x}_i) Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i) \right].
\end{aligned}
\tag{5.21}
$$

The first sum in (5.21) is computed efficiently by following the Algorithm 4.2.1. The second sum in 5.21 is computed by using direct multiplication but we note that during the computation of the first sum, the matrices $\mathbf{I_R}$, $\mathbf{B_R}$ for $Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}_i)$ had already been computed.

In this way, we obtain the product in an efficient way and this product will be used in each iteration of the iteration method which results the solution efficiently. If we need accurate solution, then we must iterate the algorithm with larger tolerance.

This Step replaces the BPM in [42] which uses the domain-only meshless discretization.

5.1.3 Step 3:

Once the approximate solution of $\tilde{u}(\mathbf{x}, s)$ is found in the Laplace domain from the Step 2, we need to invert back to the original time domain which is achieved by applying numerical inverse Laplace transform schemes. There are many schemes in the literature such as Gaver-Stehfest, Euler, and Talbot algorithm [1, 93].

All of these algorithms can be combined into a single inversion procedure with different scaling constants as

$$
u(x,t) = \frac{1}{t} \sum_{\nu=1}^{n_s} \operatorname{Re}\left(W_\nu u(\mathbf{x}, s_\nu / t)\right), \tag{5.22}
$$

where n_s is the number of terms in each algorithm, W_ν and s_ν are exterior and interior scaling constants respectively. The list of these parameters on each algorithm is given below:

- Gaver-Stehfest Algorithm

$$
\begin{aligned}
&n_s = 2M, \quad M \in \mathbb{Z}^+, \\
&s_\nu = \nu \ln(2), \\
&W_\nu = (-1)^{M+\nu} \ln(2) \sum_{k=[(\nu+1)/2]}^{\min(\nu, M)} \frac{k^{M+1}}{M!} \binom{M}{k} \binom{2k}{k} \binom{k}{\nu - k},
\end{aligned}
$$

where $[x]$ is the greatest integer less than or equal to x.

- Euler Algorithm

$$\begin{aligned} n_s &= 2M, \quad M \in \ \mathbb{Z}^+, \\ s_\nu &= \frac{M \ln(10)}{3} + \pi i \nu, \\ W_\nu &= (-1)^\nu 10^{M/3} \xi_\nu, \end{aligned}$$

where,

$$\xi_\nu = \begin{cases} \dfrac{1}{2}, & \nu = 0, \\ 1, & 1 \leq \nu \leq M, \\ \dfrac{1}{2^M}, & \nu = 2M, \\ \xi_{2M-\nu} = \xi_{2M-\nu+1} + 2^{-M}\binom{M}{\nu}, & 0 < \nu < M. \end{cases}$$

- Talbot Algorithm

$$\begin{aligned} n_s &= M - 1, \quad M \in \ \mathbb{Z}^+, \\ s_\nu &= \begin{cases} \dfrac{2M}{5}, & \nu = 0, \\ \dfrac{2\nu\pi}{5}\left(\cot(\nu\pi/M) + i\right), & 0 < \nu < M, \end{cases} \\ W_\nu &= \begin{cases} \dfrac{2}{10} e^{s_0}, & \nu = 0, \\ \dfrac{2}{5}\left[1 + i(\nu\pi/M)\left(1 + [\cot(\nu\pi/M)]^2\right) - i\cot(\nu\pi/M)\right] e^{s_\nu}, & 0 < \nu < M. \end{cases} \end{aligned}$$

The accuracy of these algorithm depends on the choice of the number n_s of the terms in (5.22). As n_s increases, the accuracy improves first, but then round-off errors become dominant and finally the accuracy declines. This is a rather common phenomenon in practical numerical computation. The optimal n_s has a significant impact on the quality of the final solution of our proposed method. According to [1], $j-$ significant digits of accuracy can be obtained by setting $M = [1.1j]$ in Gaver-Stehfest and $M = [1.7j]$ in Euler and Talbot algorithms. In this dissertation, we have used Talbot algorithm to invert the solution from the Laplace space.

5.2 Numerical results

First, we numerically solve a time fractional advection-diffusion equation using MPS and compare the result with some of the well-known methods. After figuring out the supremacy of our proposed method compared with the others, we use the fast version of the MPS, FMPS to deal with problem with large number of collocation points. We compare the results by using RMSE, MAE, Relative Average Error (RAE). RMSE and MAE are defined as in (2.54), (2.55) respectively and RAE is defined follows:

$$\mathrm{RAE} = \sqrt{\frac{\sum_{j=1}^{n_t}(\hat{u}_j - u_j)^2}{\sum_{j=1}^{n_t}(u(j))^2}}, \tag{5.23}$$

where $n_t, \hat{u}_j$ and u_j are the number of test points, approximate solution, and exact solution, respectively.

Example 5.2.1. Let us consider the time fractional diffusion equation with the Dirichlet boundary conditions in a unit square domain $\Omega = [0,1] \times [0,1]$. This problem has been solved with LTBPM in [42] and compared with other well-known techniques. So, let us introduce the similar problem,

$$\frac{\partial^\alpha u(\boldsymbol{x},t)}{\partial t^\alpha} = \Delta u(\boldsymbol{x},t) + \left[\frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} - 2t^2\right]\exp(x+y), \ \ \boldsymbol{x} \in (0,1)\times(0,1), \tag{5.24}$$

with boundary condition

$$u(\boldsymbol{x},t) = t^2 \exp(x+y),$$

and initial condition

$$u(\boldsymbol{x},0) = 0.$$

We choose the fractional parameter $\alpha = 0.85$ as in example [42]. The analytical solution is $u(\boldsymbol{x},t) = t^2 e^{x+y}$. The Laplace transform problem for (5.24) is given in the Appendix C.

In the table 5.1, we have compared the accuracy obtained by MPS with other methods for $T = 1$ with $\Delta h = 0.2$, where Δh is the length of the discretization of the collocation points in the square domain. We use the similar computational points as in [42], just to compare with LTBPM and other methods. Table 5.1 and Figure 5.1 are obtained by using Talbot inversion algorithm with $M = 64$ in MPS at $T = 1$. The Matérn RBFs with order 3 at the shape parameter 1.81 has been used as a basis for MPS to obtain the result in Table 5.1.

Table 5.1: Comparison of the RAE and MAE of MPS with LTBPM and DRBF

	RAE	MAE
MPS (Matérn 3)	8.228e-07	3.358e-04
LTBPM	5.596E-05	4.135E-04
DRBF ($\Delta t = 0.1$)	2.073E-03	1.234E-02
DRBF ($\Delta t = 0.004$)	5.116E-5	3.046E-04

In [42], LTBPM has RAE around $1E-06$ and $1E-07$ for $\Delta h = 0.1$ and $\Delta h = 0.05$ respectively. Figure 5.1 shows the RAE for different shape parameters by the MPS. As we see in the Figure 5.1, for each size of the domains, MPS has better accuracy than LTBPM. All of the RAE for different computational points, Matérn RBFs has similar results. As we increase the computational points, the optimal shape parameter shifted to the right and accuracy improved more. Also near the optimal shape parameter, Matérn has sudden jump.

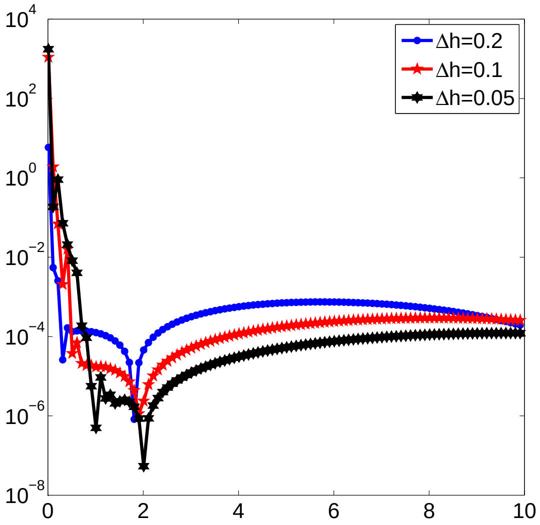

Figure 5.1: RAE obtained by Matérn RBFs with order 3 at different shape parameters with different computational points.

The RMSE, RAE, and MAE obtained by several RBFs including MQ, IMQ, Gaussian, Matérn RBFs with order 2 and 3 are shown in the Table 5.2. The Talbot inversion algorithm has been used at $T = 1$, for the number of computational points obtained from $\Delta h = 0.05$. The optimal RMSE and RAE are obtained at the similar shape parameter for each RBFs but the optimal MAE is obtained at different shape parameter. Among all of these RBFs, Matérn RBFs with order 3 has better accuracy.

Table 5.2: RMSE, RAE, and MAE obtained by different RBFs at $\Delta h = 0.05$

RBFs	RMSE (ep)	RAE (ep)	MAE(ep)
MQ	4.661E-08 (3.535)	1.427E-08 (3.535)	7.453E-06 (1.876)
IMQ	1.056E-06 (1.699)	3.232E-07 (1.699)	1.364E-05 (1.701)
Gaussian	1.289E-06 (1.045)	3.947E-07 (1.045)	2.350E-04 (4.739)
Matérn (2)	4.391E-08 (1.654)	1.344E-08 (1.654)	1.332E-05 (0.985)
Matérn (3)	9.656E-09 (2.02)	2.956E-09 (2.02)	2.034E-06 (1.326)

Now, we show the numerical results for the higher number of computational points. To solve the problem with higher number of points, we use the FMPS with PS as a RBFs. We have used $\Delta h = 0.01$ for the discretization of the collocation points in the square domain. Figures 5.2 - 5.4 depict the RAE, MAE and RMSE errors for $5, 9, 13$ orders of the PS-RBFs at each iteration. Figure 5.5 shows the computational time used by various orders of the PS-RBFs at each iteration. In FMPS, we have chosen 144 Chebyshev points. Table 5.3 illustrates the RMSE, RAE, MAE and the computational time used by MPS using various orders of the PS-RBFs. As we see from the Table 5.3 and Figures 5.2 - 5.4 that to achieve the similar accuracy FMPS is highly efficient as compared to MPS. Now, we increase the number of computational points and observe more numerical results using FMPS.

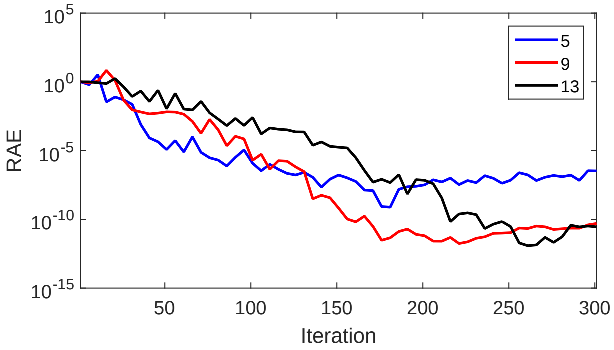

Figure 5.2: RAE obtained by different polynomial order at each iteration

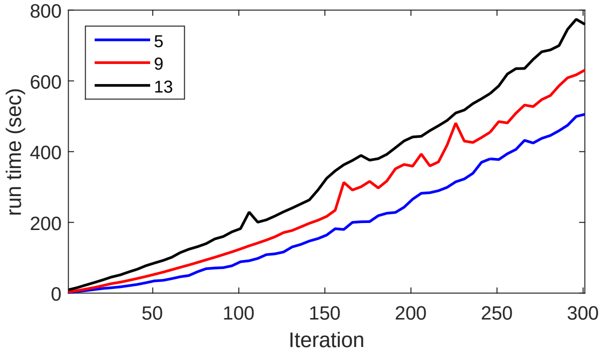

Figure 5.5: Computational time taken by different polynomial order at each iteration.

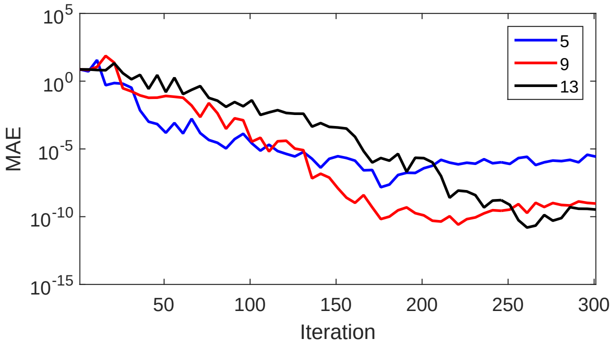

Figure 5.3: MAE obtained by different polynomial order at each iteration

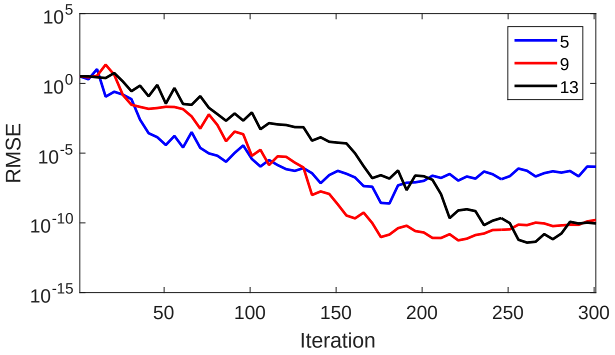

Figure 5.4: RMSE obtained by different polynomial order at each iteration

Table 5.3: RMSE, RAE, MAE, and the computational time obtained by MPS using different polyharmonic orders

Order	RMSE	RAE	MAE	time (sec)
5	2.645407E-06	8.254937E-07	6.413589E-06	2713
9	1.598312E-09	4.987499E-10	6.976534E-09	2782
13	5.768313E-10	1.799990E-10	5.281207E-09	2827

Chapter 6

Conclusion and Future works

As we have discussed in our introductory Chapter 1, the three different issues in MPS: a) obtaining the closed-form particular solutions, b) the selection of the good shape parameter for various RBFs, and c) simulations of the problems which involve large number of interpolation points, have been thoroughly investigated. These proposed techniques for resolving each issues have been validated by the numerical results sections. Now, we draw a conclusion on each issues in the next section.

6.1 Conclusion

- **Obtaining the closed-form particular solutions:** In this study, we focus on the derivation of the closed-form particular solutions of the Laplace and biharmonic differential operators using Matérn and Gaussian RBFs in the context of the MPS and LMPS. Numerical examples presented in each chapters of this dissertation demonstrate the effectiveness of the derived particular solutions. The obtained closed-form particular solutions using the Gaussian RBFs are particularly useful for the LMPS. Since the LMPS is a localized method in which only a small number of neighboring points is required in the solution process, we need to use the RBFs with high convergence rate. To our knowledge, only MQ and inverse MQ are effective for the LMPS. The derived particular solutions using the Gaussian RBFs will allow the practitioners to have one more choice for the implementation of the LMPS. One drawback of our derived particular solutions is the use of the special functions erf and Ei which is more computational intensive than the standard functions such as sine, exponential and polynomial. With the increasing computing power of a modern computer, the above mentioned issue is less of a problem. These derived particular solutions have also been implemented to solve time fractional differential equations.

- **Selection of the good shape parameter for various RBFs:** To select a good constant shape parameter of the Gaussian RBFs, we apply the LOOCV algorithm. The numerical results in Section 3.3 show that derived particular solutions are effective for solving boundary value problems in irregular domains with sharp edges. Existing variable shape parameter strategies work well for the MPS such as other RBFs collocation methods. The modified and newly proposed techniques produce the comparable results with the existing one. The proposed idea of modifying the trigonometric varying shape parameter to avoid the negative c_j's does not lose its effectiveness in terms of accuracy. We conclude that, we do not have to stick on the uniformly distributed random numbers but we can use other random numbers like quasi-random numbers. The global optimization tool SA works well than PS and GA. The idea of using the existing shape parameter strategies as an initial guess for PS and SA works very well for the exponentially varying shape parameter strategy but it does not perform better for other strategies for a large size of the search interval.

- **Simulations of the problems which involve large number of interpolation points:** In this dissertation, we propose to couple the FSM with GMRES to speed up the computational efficiency of the MPS. FMPS has been proven to be a very effective alternative for solving the PDEs when a large number of computational points is required for both two and three dimensional problems. The numerical results in Section 4.3 show the proposed method is highly efficient. The difficulty of global RBFs collocation methods such as the MPS for solving large-scale elliptic PDEs has been alleviated. Also, time fractional diffusion equations have been solved with large number of collocation points by using the Laplace transform techniques. To our knowledge, the proposed algorithm is the first MPS fast method which is a RBFs-based method for solving PDEs.

6.2 Future works

Our main focus in this dissertation is on the derivation of some techniques to improve the MPS for solving more challenging science and engineering problems. In this section, we list some of the other extensions of this work.

- In this dissertation, the derivation of the closed-form particular solutions of the Matérn and Gaussian RBFs have been implemented in MPS, LMPS, and FMPS for solving various PDEs. These derived closed-form particular solutions can be applied in other numerical methods including MFS-MPS and DRM for solving challenging science and engineering problems. These proposed methods can be extended to axisymmetric problems [87, 88]. Also, the derivation of the closed-form particular solutions of the higher order differential operator can be another subject of future research project.

- In Chapter 3, we observed that, the global optimization tools SA, PS, and GA are coupled with the existing variable shape parameter strategies which help us to obtain the good shape parameter. Since the work in Chapter 3 is an introduction for newly proposed ideas, we have not thoroughly investigated PS and SA and left it as a future work. There are still other global optimization tools including particle swarm optimization which can also be applied to obtain the good constant and variable shape parameter.

- Solving large-scale problem using global MPS was a difficult task. The development of the FMPS has shown an alternative way to simulate the large-scale problems. In FMPS, we have used Chebyshev polynomials to approximate the given RBFs or its corresponding particular solutions. Instead of using the Chebyshev polynomials, we can try other effective interpolation scheme to develop the similar fast method. It would be interesting to compare the performance of the proposed approach which is a global method to localized RBFs collocation methods for solving large-scale problems. Further improvement in the accuracy of FMPS will be the subject of our another future research.

- In Chapter 4, we have only coupled the FSM with MPS but these approach can be implemented at any RBFs collocation methods including MFS, and MFS-MPS.

- Although we have successfully solved the time fractional diffusion equations by FMPS, there are still more challenging real-life problems which need to solve efficiently. In future, we will continue our research to solve other science and engineering problems using FMPS.

Appendix A

List of the closed-form particular solutions for various RBFs

A.1 Closed-form particular solutions for polyharmonic splines:

A.1.1 In 2D [28]

$$\Delta\Phi = r^{2n-1} \qquad \Longrightarrow \Phi = \frac{1}{(2n+1)^2} r^{2n+1},$$

$$\Delta^2\Phi = r^{2n-1} \qquad \Longrightarrow \Phi = \frac{1}{(2n+1)^2(2n+3)^2} r^{2n+3},$$

$$\Delta\Phi = r^{2n}\ln r \qquad \Longrightarrow \Phi = \frac{r^{2n+2}}{4(n+1)^2}\left(\ln r - \frac{1}{n+1}\right),$$

$$\Delta^2\Phi = r^{2n}\ln r \qquad \Longrightarrow \Phi = \frac{r^{2n+4}}{16(n+1)^2(n+2)^2}\left[\ln r - \frac{2n+3}{(n+1)(n+2)}\right].$$

A.1.2 In 3D [28]

$$\Delta\Phi = r^{2n-1} \qquad \Longrightarrow \Phi = \frac{r^{2n+1}}{(2n+1)(2n+2)},$$

$$\Delta^2\Phi = r^{2n-1} \qquad \Longrightarrow \Phi = \frac{r^{2n+3}}{(2n+1)(2n+2)(2n+3)(2n+4)}.$$

A.2 Closed-form particular solutions for multiquadrics:

A.2.1 In 2D [77, 106]

$$\Delta\Phi = \sqrt{r^2+c^2} \quad \Longrightarrow \Phi = \frac{4c^2+r^2}{9}\sqrt{r^2+c^2} - \frac{c^3}{3}\ln\left(c+\sqrt{r^2+c^2}\right),$$

$$\Delta^2\Phi = \sqrt{r^2+c^2} \quad \Longrightarrow \Phi = \frac{2c^2}{45}(r^2+c^2)^{3/2} - \frac{7c^4}{60}\sqrt{r^2+c^2} + \frac{2c^2-5r^2}{60}c^3\ln\left(c+\sqrt{r^2+c^2}\right)$$

$$+\frac{1}{225}(r^2+c^2)^{5/2} + \frac{c^3r^2}{12},$$

$$\Delta\Phi = \sqrt{1+c^2r^2} \quad \Longrightarrow \Phi = \frac{1}{9c^2}\left[(4+c^2r^2)\sqrt{1+c^2r^2} - 3\ln\left(1+\sqrt{1+c^2r^2}\right)\right],$$

$$\Delta^2\Phi = \sqrt{1+c^2r^2} \quad \Longrightarrow \Phi = \frac{2-5c^2r^2}{60c^4}\ln\left(1+\sqrt{1+c^2r^2}\right) + \frac{\sqrt{1+c^2r^2}}{900c^4}(4c^4r^4+48c^2r^2-61)$$

$$+\frac{2c^2r^2+1}{24c^4}.$$

A.2.2 In 3D [77, 106]

$$\Delta\Phi = \sqrt{r^2+c^2} \quad \Longrightarrow \Phi = \begin{cases} \dfrac{2r^2+5c^2}{24}\sqrt{r^2+c^2} + \dfrac{c^4}{8r}\ln\left(\dfrac{r+\sqrt{r^2+c^2}}{c}\right), & r > 0, \\[2ex] \dfrac{c^3}{3}, & r = 0, \end{cases}$$

$$\Delta^2\Phi = \sqrt{r^2+c^2} \quad \Longrightarrow \Phi = \begin{cases} \dfrac{c^4(6r^2-c^2)}{96r}\ln\left(\dfrac{r+\sqrt{r^2+c^2}}{c}\right) & \\[2ex] +\dfrac{\sqrt{r^2+c^2}}{1440}(4r^4+28c^2r^2-81c^4), & r > 0, \\[2ex] \dfrac{-c^5}{15}, & r = 0, \end{cases}$$

$$\Delta\Phi = \sqrt{1+c^2r^2} \quad \Longrightarrow \Phi = \begin{cases} \dfrac{5+2c^2r^2}{24c^2}\sqrt{1+c^2r^2} + \dfrac{\sinh^{-1}(cr)}{8c^3r}, & r > 0, \\[2ex] \dfrac{1}{3c^2}, & r = 0, \end{cases}$$

$$\Delta^2\Phi = \sqrt{1+c^2r^2} \quad \Longrightarrow \Phi = \begin{cases} \dfrac{1+\sqrt{c^2r^2}}{1440c^4}(4c^4r^4+28c^2r^2-81) & \\[2ex] +\dfrac{\sinh^{-1}(cr)}{96c^5r}(6c^2r^2-1), & r > 0, \\[2ex] \dfrac{-1}{15c^4}, & r = 0. \end{cases}$$

A.3 Closed-form particular solutions for inverse multiquadrics:

A.3.1 In 2D [77, 106]

$$\Delta\Phi = \frac{1}{\sqrt{r^2+c^2}} \quad \Longrightarrow \Phi = \sqrt{r^2+c^2} - c\ln\left(c+\sqrt{r^2+c^2}\right),$$

$$\Delta^2\Phi = \frac{1}{\sqrt{r^2+c^2}} \quad \Longrightarrow \Phi = \frac{4r^2-11c^2}{36}\sqrt{r^2+c^2} + \frac{c(2c^2-3r^2)}{12}\ln\left(c+\sqrt{r^2+c^2}\right)$$

$$+\frac{c^3\ln(2c)}{6} + \frac{r^2c}{4},$$

$$\Delta\Phi = \frac{1}{\sqrt{1+c^2r^2}} \quad \Longrightarrow \Phi = \frac{1}{c^2}\left[\sqrt{1+c^2r^2} - \ln\left(1+\sqrt{1+c^2r^2}\right)\right],$$

$$\Delta^2\Phi = \frac{1}{\sqrt{1+c^2r^2}} \quad \Longrightarrow \Phi = \frac{\sqrt{1+c^2r^2}}{36c^4}(4c^2r^2-11) + \frac{2-3c^2r^2}{12c^4}\ln\left(1+\sqrt{1+c^2r^2}\right)$$

$$+\frac{r^2}{4c^2}.$$

A.3.2 In 3D [77, 106]

$$\Delta\Phi = \frac{1}{\sqrt{r^2+c^2}} \qquad \Longrightarrow \Phi = \begin{cases} \frac{1}{2}\sqrt{r^2+c^2} + \frac{c^2}{2r}\ln\left(\frac{r+\sqrt{r^2+c^2}}{c}\right), & r>0, \\ \frac{c}{2} & r=0, \end{cases}$$

$$\Delta^2\Phi = \frac{1}{\sqrt{r^2+c^2}} \qquad \Longrightarrow \Phi = \begin{cases} \frac{c^2(4r^2-c^2)}{16r}\ln\left(\frac{r+\sqrt{r^2+c^2}}{c}\right) & \\ \quad +\frac{2r^2-13c^2}{48}\sqrt{r^2+c^2}, & r>0, \\ \frac{-c^3}{3}, & r=0, \end{cases}$$

$$\Delta\Phi = \frac{1}{\sqrt{1+c^2r^2}} \qquad \Longrightarrow \Phi = \begin{cases} \frac{1+c^2r^2}{2c^2} + \frac{\sinh^{-1}(cr)}{2c^3r}, & r>0, \\ \frac{1}{c^2} & r=0, \end{cases}$$

$$\Delta^2\Phi = \frac{1}{\sqrt{1+c^2r^2}} \qquad \Longrightarrow \Phi = \begin{cases} \frac{\sqrt{1+c^2r^2}}{48c^4}(2c^2r^2-13) & \\ +\frac{\sinh^{-1}(cr)}{16c^5r}(4c^2r^2-1), & r>0, \\ \frac{-1}{3c^4}, & r=0. \end{cases}$$

Appendix B

Shape parameter strategies

B.1 List of some of the constant shape parameter strategies

- R. Hardy [47]

  $$c = 0.815d, \text{ where } d = \frac{1}{N}\sum_{i=1}^{N} d_i,$$

  d_i is the distance from the i^{th} center to its nearest neighbor and N is the number of collocation points,

- R. Franke [39]

  $$c = (1.25D)/\sqrt{N},$$

  where D is the diameter of the smallest circle containing all data points,

- G. E. Fasshauer [34]

  $$c = 2/\sqrt{N},$$

- **Leave one out cross validation (LOOCV)** [36, 86],

- **Golden section search algorithm** [100],

- **Genetic Algorithm** [32].

B.2 Leave-one-out cross validation (LOOCV)

We briefly define the LOOCV method for finding the shape parameter as in [36].

Suppose,

$$\boldsymbol{x}^{[k]} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{k-1}, \boldsymbol{x}_{k+1}, ..., \boldsymbol{x}_N]^T$$

is the vector containing all of the given points except $\boldsymbol{x}_k$. Similarly, we can define $f^{[k]}, P_f^{[k]}, \alpha^{[k]}$ as the vectors containing functional values, approximation and undetermined coefficients removing the k^{th} term respectively. Then, the following pseudo code for LOOCV given in [36] computes the good shape parameter for RBFs approximation:

Algorithm

```
Fix $c$
For $k = 1, ..., N$
Let
    $P_f^{[k]}(\boldsymbol{x}) = \sum_{j=1}^{N-1} \boldsymbol{\alpha}_j^{[k]} \phi \left( \|\boldsymbol{x} - \boldsymbol{x}_j^{[k]}\| \right)$
Compute the error estimator at the $k^{th}$ data point
    $e_k = |f(\boldsymbol{x}_k) - P_f^{[k]}(\boldsymbol{x}_k)|$
end
Form the cost vector $e = [e_1, ..., e_N]^T$
The optimal $c$ is given by minimizing $||e||$
```

Similar procedure works for the MPS except the interpolation matrix will be obtained from the MPS discretization. The implementation of the LOOCV algorithm presented above is very expensive. In [86], Rippa has shown that the algorithm can be simplified to a single formula by using the following formula for the cost vector:

$$e_k = \frac{\alpha_k}{B_{kk}^{-1}},$$

where α_k is the k^{th} coefficient in the expansion of the interpolant P_f based on the full data set, and B_{kk}^{-1} is the k^{th} diagonal element of the inverse of the corresponding interpolation matrix. In [36], Fasshauer has given a MATLAB implementation of the LOOCV which is presented as follows:

- Construction of the cost function:

```
% CostEps(.)  is a cost function which computes the norm of the cost
vector
%c-shape parameter
%rbf-radial basis function
%DM-Distance Matrix between all given data points
%rhs-functional values at given data points
%
function ceps=CostEps(c,rbf,DM,rhs)
A=rbf(c,DM);% Constructs the interpolation matrix
invA=pinv(A);% Find the pseudoinverse of the interpolation matrix
```

```
errorvector=(invA*rhs)./diag(invA);%Computes the cost vector by using
Rippa’s formula
ceps=norm(errorvector);% Norm of the cost vector
```

- Minimization of the cost function

  ```
  % The output of this function is the good shape parameter
  c=fminbnd((c) CostEps(c,rbf,DM,rhs),minc,maxc);
  ```

  where `minc` and `maxc` define the end points of the search interval for the good shape parameter.

Appendix C

LAPLACE TRANSFORMED PROBLEM

C.1 Laplace transforms

Definition C.1.1 (Laplace transform [42]). Let $f(t)$ be a casual time domain function as an independent variable $t \geq 0$ and $\tilde{f}(s)$ denotes its image in Laplace space domain. Laplace transform is defined by

$$\tilde{f}(s) = L(f(t)) = \int_0^\infty f(t)e^{-st}dt, \tag{C.1}$$

where s is Laplace transform parameter, and unless otherwise specified the quantities in Laplace space domain are denoted by an over tilde.

Note that

$$L(t^p) = \frac{\Gamma p+1}{s^{p+1}}, \quad p > -1.$$

Definition C.1.2. Laplace transform of the Caputo fractional derivative [84] can be written as

$$L\left(\frac{\partial^\alpha f(t)}{\partial t^\alpha}\right) = s^\alpha \tilde{f}(s) - s^{\alpha-1}f(0),\ 0 < \alpha < 1,\ t \geq 0. \tag{C.2}$$

C.2 Laplace transform of the time fractional diffusion equation

C.2.1 Laplace transform of the Eq. (5.24)

Suppose

$$Lu(\boldsymbol{x},t) = \tilde{u}(\boldsymbol{x},s),$$

$$\begin{aligned}
\mathcal{L}\left(\frac{\partial^\alpha u(\boldsymbol{x},t)}{\partial t^\alpha}\right) &= \mathcal{L}\left(\Delta u(\boldsymbol{x},t) + \left[\frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} - 2t^2\right]\exp(x+y)\right), \\
s^\alpha \tilde{u}(\boldsymbol{x},s) - s^{\alpha-1}\tilde{u}(\boldsymbol{x},0) &= \Delta\tilde{u}(\boldsymbol{x},s) + \left[\frac{2}{\Gamma(3-\alpha)} \cdot \frac{\Gamma(3-\alpha)}{s^{(3-\alpha)}} - 2\frac{\Gamma(3)}{s^3}\right]\exp(x+y), \\
(\Delta - s^\alpha)\tilde{u}(\boldsymbol{x},s) &= -\left[\frac{2}{s^{3-\alpha}} - \frac{4}{s^3}\right]\exp(x+y).
\end{aligned}$$

BIBLIOGRAPHY

[1] J. Abate and W. Whitt. A unified framework for numerical inverting Laplace transforms. *INFORMS Journal on Computing*, 18(4), Fall 2006.

[2] S. Abbasbandy, H.R. Ghehsareh, M.S. Alhuthali, and H.H. Alsulami. Comparison of meshless local weak and strong forms based on particular solutions for a non-classical 2-d diffusion model. *Engineering Analysis with Boundary Elements*, 39:121–128, 2014.

[3] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1965.

[4] M.A. Abramson, C. Audet, J.E. Dennis JR, and S.L. Digabel. Orthomads: A deterministic MADS instance with orthogonal directions. *SIAM J. OPTIM.*, 20(2):948–966, 2009.

[5] F. Afiatdoust and M. Esmaeilbeigi. Optimal variable shape parameters using genetic algorithm for radial basis function approximation. *Ain Shams Engineering Journal*, 6(639-647), 2015.

[6] C.J.S. Alves and C.S. Chen. Approximating functions and solutions of non homogeneous partial differential equations using the method of fundamental solutions. *Advances in Computational Mathematics*, pages 125–142, 2005.

[7] G. Arfken. *Mathematical Methods for Physicists*. Orlando FL: Academic Press, 1985.

[8] S.N. Atluri and T.-L. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in Computational mechanics. *Computational Mechanics*, 22:117–127, 1998.

[9] C. Audet and J.E. Dennis JR. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. OPTIM.*, 17(1):188–217, 2006.

[10] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *Int. J. Numerical Methods Eng.*, 37:229–256, 1994.

[11] M. Bozzini, M. Rossini, and R. Schaback. Generalized whittle- matérn and Polyharmonic kernels. *Advances in Computational Mathematics*, 39:129–141, 2013.

[12] C.A. Bustamante, H. Power, W. Florez, and C.Y. Hang. The global approximate particular solution meshless method for two-dimensional linear elasticity problems. *International Journal of Computer Mathematics*, 90(5):978–993, 2013.

[13] C.A. Bustamante, H. Power, and W.F. Florez. A global meshless collocation particular solution method for solving the two-dimensional Navier-Stokes system of equations. *Computers and Mathematics with Applications*, 65:1939–1955, 2013.

[14] C.A. Bustamante, H. Power, and W.F. Florez. Schwarz alternating domain decomposition approach for the solution of two-dimensional Navier-Stokes flow problems by the method of approximate particular solutions. *Numerical Methods for Partial Differential Equations*, 31(3):777–797, 2015.

[15] C.A. Bustamante, H. Power, Y.H. Sua, and W. Florez. A global meshless collocation particular solution method (integrated radial basis function) for two-dimensional Stokes flow problems. *Applied Mathematical Modelling*, 37:4538–4547, 2013.

[16] R.E. Carlson and Foley T.A. The parameter {R2} in multiquadric interpolation. *Computers & Mathematics with Applications*, 21(9):29 – 42, 1991.

[17] V. Cerny. Thermodynamical approach to the traveling salesman problems: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51, 1985.

[18] A. Chechkin, V.Y. Gonchar, J. Kflafter, and R. Metzler. Fundamental of lévy fight processess. In Kalmykov Y.P. Coffey, W.T., editor, *Fractals, Diffusion, and Relaxation in Disordered Complex Systems*, volume 133. Advances in Chemical Physics, Part B,, 2006.

[19] Chang-ming Chen, F. Lui, and Burrage K. Finite difference methods and fourier analysis for the fractional reaction-subdiffusion equation. *Appl. Math Comput.*, 2008.

[20] C.S. Chen, C.A. Brebbia, and H. Power. Dual reciprocity method using compactly supported radial basis functions. *Comm. Num. Meth. Eng.*, 15:137–150, 1999.

[21] C.S. Chen, C.M Fan, and Jeanette Monroe. The method of fundamental solutions for solving elliptic PDEs with variable coefficients. In C.S. Chen, A. Karageorghis, and Y.S. Smyrlis, editors, *The Method of Fundamental Solutions – A Meshless Method*, pages 75–105. Dynamic System Inc., 2008.

[22] C.S. Chen, C.M Fan, and P.H. Wen. The method of fundamental solutions for solving elliptic PDEs with variable coefficients. In C.S. Chen, A. Karageorghis, and Y.S. Smyrlis, editors, *The Method of Fundamental Solutions – A Meshless Method*, pages 75–105. Dynamic System Inc., 2008.

[23] 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.

[24] 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.

[25] 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.

[26] W. Chen, Z.J. Fu, and C.S. Chen, editors. *Recent Advances in Radial Basis Function Collocation Methods*. Springer, 2014.

[27] W. Chen, L. Ye, and Sun H. Fractional diffusion equations by the Kansa method. *Computers and Mathematics with Applications*, 59:1614–1620, 2010.

[28] 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.

[29] W.J. Cody and H.C. Thacher, Jr. Rational Chebyshev approximations for the exponential integral $E_1(x)$. *Math. Comp.*, 22:641–649, 1968.

[30] L. Debnath and D. Bhatta. Solutions to few linear fractional inhomogeneous partial differential equations in fluid mechanics. *Frac. Calc. Appl. Anal.*, 2004.

[31] D. Del-Castillo-Negrete, B.A. Carreras, and V.E. Lynch. Front dynamics in reaction-diffusion systems with levy flights: a fractional diffusion approach. *Physical Review Letters*, 2003.

[32] M. Esmaeilbeigi and M. M. Hosseini. A new approach based on the genetic algorithm for finding a good shape parameter in solving partial differential equations by kansa's method. *Applied Mathematics and Computation*, 249:419–428, December 2014.

[33] C.M. Fan, C.S. Chen, and J. Monroe. The method of fundamental solutions for solving convection-diffusion equations with variable coefficients. *Advances in Applied Mathematics and Mechanics*, 1(2):215–230, April 2009.

[34] G. Fasshauer. Newton iteration with multiquadrics for the solution of nonlinear PDEs. *Computers Math. Applic.*, 43(3-5):423–438, 2002.

[35] G.E. Fasshauer. *Meshfree Approximation Methods with MATLAB*. World Scientific, 2007.

[36] G.E. Fasshauer and J.G. Zhang. On choosing optimal shape parameters for RBF approximation. *Numerical Algorithms*, 45:345–368, 2007.

[37] G.J. Fix and J.P. Roof. Least squares finite-element solution of a fractional order two-points boundary value problem. *Comput. Math. Appl.*, 2004.

[38] W. Fong and E. Darve. The black-box fast multipole method. *Journal of Computational Physics*, 228:8712–8725, 2009.

[39] R. Franke. *A Critical Comparison of Some Methods for Interpolation of Scattered Data*. Defense Technical Informational Center, 1979.

[40] Z. Fu, W. Chen, and L. Ling. Method of approximate particular solutions for constant- and variable-order fractional diffusion models. *Engineering Analysis with Boundary Elements*, 57:37–46, 2015.

[41] Z.J. Fu, W. Chen, and Hai-Tian Yang. Boundary particle method for Laplace transformed time fractional diffusion equations. *Computational Physics*, 235:52–66, 2013.

[42] Z.J. Fu, W. Chen, and H.T. Yang. Boundary particle method for Laplace transformed time fractional diffusion equations. *J. Comput. Physics*, 2013.

[43] M.A. Golberg and C.S. Chen. The method of fundamental solutions for potential, Helmholtz and diffusion problems. In M.A. Golberg, editor, *Boundary Integral Methods: Numerical and Mathematical Aspects*, pages 103–176. WIT Press, 1998.

[44] R. Gorenflo, Y. Luchko, and F. Mainardi. Wright functions as scale-invariant solutions of the diffusion-wave equation. *Journal of Computational and Applied Mathematics*, 2000.

[45] R. Gorenflo, F. Mainardi, D. Moretti, G. Pagnini, and P. Paradisi. Discrete random walk models for space-time fractional diffusion. *ChemicalPhysics*, 2007.

[46] P. Guttorp and Geniting T. Studies in the history of probability and statistics XLIX on the Matérn correlation family. *Biometrika*, 93:989–995, 2006.

[47] R.L. Hardy. Multiquadric equations of Topography and other irregular surfaces. *Journal of Geophysical Research*, 176:1905–1915, 1971.

[48] R. Hilfer. *Application of Fractional Calculus in Physics*. World Scientific, Singapore, 2000.

[49] Y.C. Hon and X.Z. Mao. An efficient numerical scheme for Burgers' equation. *Appl. Math. Comput.*, 95(1):37–50, 1998.

[50] G. Hong-Xia, L. Yong-Qing, and C. Rong-Jun. Element-free Galerkin (EFG) method for analysis of the time-fractional partial-differential equations. *Chin. Phys.*, 2012.

[51] T. Jiang, M. Li, and Chen C. S. The method of particular solutions for solving inverse problems of a nonhomogeneous convection-diffusion equation with variable coefficients. *Numerical heat transfer, Part A*, 61(5):338–352, 2012.

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

[53] 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.

[54] E.J. Kansa and R.E. Carlson. Improved accuracy of multiquadric interpolation using variable shape parameters. *Computers Math. Applic.*, 24(12):99–120, 1992.

[55] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vechhi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[56] M. Kleiber and A. Borkowski. *Handbook of Computational Solid Mechanics*. Springer-Verlag, Berlin, 1998.

[57] L.H. Kuo, M.H. Gu, D.L. Young, and C.Y. Lin. Domain type kernel-based meshless methods for solving wave equations. *CMC: Computers, Materials & Continua*, 33(3):213–228, 2013.

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

[59] A.R. Lamichhane and C.S. Chen. Particular solutions of Laplace and bi-harmonic operators using Matérn radial basis functions. *International Journal of Computer Mathematics*, https://doi.org/10.1080/00207160.2015.1127359, 2016.

[60] A.R. Lamichhane, D.L. Young, and C.S. Chen. Fast method of approximate particular solutions using Chebychev interpolation. *Eng Anal Bound Elem*, 64:290–294, 2016.

[61] R. J. Le Veque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, New York, 2002.

[62] J. Li and Chen Y. *Computational Partial Differential Equations Using MATLAB*. Chapman & Hall/CRC, har/com edition, 2008.

[63] M. Li, G. Amazzar, A. Naji, and C. S. Chen. Solving biharmonic equation using the localized method of approximate particular solutions. *International Journal of Computer Mathematics*, DOI:10.1080/00207160.2013.862525, 2014.

[64] M. Li, W. Chen, and C.H. Tsai. A regularization method for the approximate particular solution of nonhomogeneous Cauchy problems of elliptic partial differential equations with variable coefficients. *Engineering Analysis with Boundary Elements*, 36:274–280, 2012.

[65] C.Y. Lin, M. Gu, D. Young, J. Sladek, and V. Sladek. The localized method of approximate particular solutions for solving two-dimensional incompressible viscous flow field. *Engineering Analysis with Boundary Elements*, 57(23-36), 2015.

[66] G.R. Liu. *Meshfree Methods: Moving Beyond the Finite Element Methods*. CRC Press, second edition, 2009.

[67] G.R. Liu and Y. T. Gu. *An Introduction to Meshfree Methods and Their Programming*. Springer, 2005th edition, 2005.

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

[69] G.R. Liu and S.S. Quek. *The Finite Element Method: A Practical Course*. Butterworth Heimemann, Oxford, 2002.

[70] Q Liu., Y. T. Gu., P. Zhuang, F. Liu, and Y. F. Nie. An implicit RBF meshless approach for time fractional diffusion equations. *Computational Mechanics*, 2011.

[71] W. K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko. Reproducing kernel particle methods for structural dynamics. *Int. J. for Numerical methods in engineering*, 38:1655–1679, 1995.

[72] F. Lui, P. Zhuang, V. Anh, I. Turner, and K. Burrage. Stability and convergence of the difference methods for the space-time fractional advection diffusion equation. *Appl. Math Comput.*, 2007.

[73] Mate´rn. *Spatial Variation*. Number 36 in Lecture notes in statistics. Springer-V, Berlin, 1986.

[74] M. McCourt. Using Gaussian eigenfunctions to solve boundary value problems. *Adv. Appl. Math. Mech*, 5(4):569–594, 2013.

[75] R. Metzler and J. Klafter. The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Physics Reports*, 2000.

[76] R. Metzler and J. Klafter. The restaurant at the end of the random walk: Recent developments in the description of anomalous transport by fractional dynamics. *J. Phys. A. Math Gen.*, 2004.

[77] J. Monroe. *Hybrid Meshless Method for Numerical Solution of Partial Differential Equations*. PhD thesis, University of Southern Mississippi, 2014.

[78] 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.

[79] C.T. Mouat. *Fast Algorithms and Preconditioning Techniques for Fitting Radial Basis Functions*. PhD thesis, Mathematics Department, University of Canterbury, Christchurch, New Zealand, 2001.

[80] 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.

[81] Diego A. Murio. Implicit finite difference approximation for time fractional diffusion equations. *Comput. Math. Appl.*, 2008.

[82] Momani S. Odibat, Z. The variational iteration method: An efficient scheme for handling fractional partial differential equations in fluid mechanics. *Comput. Math. Appl.*, 2009.

[83] P.W. Partridge, C.A. Brebbia, and L.C. Wrobel. *The Dual Reciprocity Boundary Element Method*. CMP/Elsevier, 1992.

[84] I. Podlubny. *Fractional Differential Equations*. Academic Press, San Diego, 1999.

[85] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[86] 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.

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

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

[89] S. A. Sarra and D. Sturgill. A random variable shape parameter strategy for radial basis approximation methods. *Engineering Analysis with Boundary Elements*, 33:1239–1245, 2009.

[90] H. Scher and E.W. Montroll. Anomalous transit-time dispersion in amorphous solids. *Phys. Rev.*, 1975.

[91] M. Seeger. Gaussian processes for machine learning. *Int. J. Neural Syst.*, 14:69–106, 2004.

[92] I.M. Sokolov, J. Klafter, and A. Blumen. Ballastic versus diffusive pair dispersion in the richardson regime. *Phys. Rev. E*, 2000.

[93] H. Stehfest. Algorithm 368: numerical inversion of Laplace transform. *Commun. ACM*, 13:47–49, 1970.

[94] M. Stein. *Interpolation of Spatial Data: Some theory for Kriging*. Springer-Verlag, 1999.

[95] T.L. Szabo and J. Wu. A model for longitudinal and shear wave propagation in viscoelastic media. *Journal of Acoustical Society of America*, 2000.

[96] BOOST C++ library. http://www.boost.org.

[97] C.C. Tsai. Automatic particular solutions of arbitrary high-order splines associated with polyharmonic and poly-Helmholtz equations. *Eng Anal Bound Elem*, 35(7):925–34, 2011.

[98] C.C. Tsai. Generalized polyharmonic multiquadrics. *Eng Anal Bound Elem*, 50(0):239–48, 2015.

[99] C.C. Tsai, A. H-D. Cheng, and C.S. Chen. Particular solutions of splines and monomials for polyharmonic and products of Helmholtz operators. *Engineering Analysis with Boundary Elements*, 33:514–521, 2009.

[100] C.H. 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.

[101] 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.

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

[103] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

[104] S. Xiang, K. Wang, Y. Ai, Y. Sha, and H. Shi. Trigonometric variable shape parameter and exponent strategy for generalized multiquadric radial basis function approximation. *Applied Mathematical Modelling*, 36:1931–1938, 2012.

[105] L. Yan and F. Yang. The method of approximate particular solutions for the time-fractional diffusion equation with a non-local boundary condition. *Computers and Mathematics with Applications*, 70:254–264, 2015.

[106] G. Yao. *Local Radial Basis Function Methods for Solving Partial Differential Equations*. PhD thesis, University of Southern Mississippi, 2010.

[107] 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, 2011.

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

[109] P.V. Yee and Simon Haykin. *Regularized Radial Basis Functional Networks: Theory and Applications*. John Wiley & Sons, Inc. New York, 2001.

[110] Z. Yunying, Li. Changpin, and Z. Zhengang. A note on the finite element method for the space-fractional advection diffusion equation. *Comput. Math. Appl.*, 2010.

[111] G. Zaslavsky. Chao, fractional kinetics, and anomalous transport. *Phys. Rep.*, 2002.

[112] P. Zhauang, T. Gu, Y, F. Liu, I. Turner, and P.K. D.V. Yarlagadda. Time-dependent fractional advection-diffusion equations by an implicit MLS meshless method. *Int. J. Numer. Meth. Engng*, 2011.

[113] H. Zhu. The method of approximate particular solutions for solving anisotropic elliptic problems. *Engineering Analysis with Boundary Elements*, 40:123–127, 2014.

[114] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. Butterworth Heimemann, Oxford, 5th edition, 2000.