

Fall 2017

## Radial Basis Function Differential Quadrature Method for the Numerical Solution of Partial Differential Equations

Daniel Watson  
*University of Southern Mississippi*

Follow this and additional works at: <https://aquila.usm.edu/dissertations>



Part of the [Numerical Analysis and Computation Commons](#), and the [Partial Differential Equations Commons](#)

---

### Recommended Citation

Watson, Daniel, "Radial Basis Function Differential Quadrature Method for the Numerical Solution of Partial Differential Equations" (2017). *Dissertations*. 1468.  
<https://aquila.usm.edu/dissertations/1468>


This Dissertation is brought to you for free and open access by The Aquila Digital Community. It has been accepted for inclusion in Dissertations by an authorized administrator of The Aquila Digital Community. For more information, please contact [aquilastaff@usm.edu](mailto:aquilastaff@usm.edu).

Fall 2017

# Radial Basis Function Differential Quadrature Method for the Numerical Solution of Partial Differential Equations

Daniel Watson

Follow this and additional works at: <https://aquila.usm.edu/dissertations>

 Part of the [Numerical Analysis and Computation Commons](#), and the [Partial Differential Equations Commons](#)

---

The University of Southern Mississippi

RADIAL BASIS FUNCTION DIFFERENTIAL QUADRATURE METHOD FOR THE  
NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

by

Daniel Wade Watson

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

December 2017

RADIAL BASIS FUNCTION DIFFERENTIAL QUADRATURE METHOD FOR THE  
NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

by

Daniel Wade Watson

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

Approved:

---

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

---

Dr. Haiyan Tian, Committee Member  
Professor, Mathematics

---

Dr. Zhifu Xie, Committee Member  
Professor, Mathematics

---

Dr. Huiqing Zhu, Committee Member  
Associate Professor, Mathematics

---

Dr. Karen S. Coats  
Dean of the Graduate School

December 2017

COPYRIGHT BY

DANIEL WADE WATSON

2017

## ABSTRACT

### RADIAL BASIS FUNCTION DIFFERENTIAL QUADRATURE METHOD FOR THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

by Daniel Wade Watson

December 2017

In the numerical solution of partial differential equations (PDEs), there is a need for solving large scale problems. The Radial Basis Function Differential Quadrature (RBF-DQ) method and local RBF-DQ method are applied for the solutions of boundary value problems in annular domains governed by the Poisson equation, inhomogeneous biharmonic equation, and the inhomogeneous Cauchy-Navier equations of elasticity. By choosing the collocation points properly, linear systems can be obtained so that the coefficient matrices have block circulant structures. The resulting systems can be efficiently solved using matrix decomposition algorithms (MDAs) and fast Fourier transforms (FFTs). For the local RBF-DQ method, the MDAs used are modified to account for the sparsity of the arrays involved in the discretization. An adjusted Fasshauer estimate is used to obtain a good shape parameter value in the applied radial basis functions (RBFs) for the global RBF-DQ method while the leave-one-out cross validation (LOOCV) algorithm is employed for the local RBF-DQ method using a sample of local influence domains. A modification of the kdtree algorithm is used to select the nearest centers for each local domain. In several numerical experiments, it is shown that the proposed algorithms are capable of solving large scale problems while maintaining high accuracy.

## ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. C.S. Chen. He is an amazing mathematician who makes the most difficult concepts appear simple. His passion of scientific computing and problem solving has inspired me beyond measure. Without his continuous support, this work would not have come to fruition.

I would also like to extend my sincere thanks to Dr. Karageorghis for sharing his expert knowledge in the field. Though we have never officially met, our email correspondence has been very valuable. I am also deeply grateful to my committee members Drs. Tian, Zhu, and Xie for their continuous support, encouragement, and useful feedback in this work. I am very fortunate to have support of my committee members.

I would like to express special thanks to my family. Without the unwavering support, encouragement, and sacrifice of my beautiful wife, Melinda, this would not have been possible. Thank you to my wonderful children Linda Rose, Isaac, and Eli, for your understanding and encouragement. I hope this shows you that you can do anything that you put your mind to.

Finally, my thanks goes to all of my friends at USM, my research group members Thir R Dungal, Lei-Hsin Kuo, Balaram K Ghimire, and Anup R Lamichhane who were always ready to help me out at any time. I especially thank Shannon Ryle. We have walked through this journey together and I would not have made it without you my friend.

# TABLE OF CONTENTS

<b>ABSTRACT</b> . . . . .	iii
<b>ACKNOWLEDGMENTS</b> . . . . .	i
<b>LIST OF ILLUSTRATIONS</b> . . . . .	iv
<b>LIST OF TABLES</b> . . . . .	v
<b>LIST OF ABBREVIATIONS</b> . . . . .	vii
<b>NOTATION AND GLOSSARY</b> . . . . .	viii
<b>1 INTRODUCTION</b> . . . . .	<b>1</b>
1.1 Background	1
1.2 Literature Review	2
1.3 Synopsis	4
<b>2 STATE-OF-THE-ART IN MESHLESS METHOD</b> . . . . .	<b>6</b>
2.1 Radial Basis Function (RBF) Interpolation	6
2.2 The Collocation (Kansa) Method	7
2.3 The Method of Particular Solutions	8
2.4 Radial Basis Function Differential Quadrature Method	9
2.5 Local Radial Basis Function Differential Quadrature Method	11
2.6 Shape Parameter	13
<b>3 MATRIX DECOMPOSITION ALGORITHM</b> . . . . .	<b>18</b>
3.1 Matrix Decomposition	18
3.2 Circulant Matrices	22
3.3 Properties of Circulant Matrices	24
<b>4 Global and Local RBF-DQ MDA</b> . . . . .	<b>31</b>
4.1 RBF-DQ MDA	31
4.2 Circulant Matrix Transform	45
4.3 Matrix Decomposition Algorithm for Cauchy-Navier	49
4.4 LRBF-DQ MDA	51
<b>5 NUMERICAL RESULTS</b> . . . . .	<b>57</b>
5.1 Poisson Equation RBF-DQ	58



5.2	Biharmonic Equation RBF-DQ	63
5.3	Cauchy-Navier RBF-DQ	67
5.4	Poisson Equation LRBF-DQ	69
5.5	Biharmonic Equation LRBF-DQ	73
5.6	Cauchy-Navier LRBF-DQ	76
<b>6</b>	<b>CONCLUSIONS AND FUTURE WORKS</b>	<b>79</b>
6.1	Conclusions	79
6.2	Future Works	81
	<b>INDEX</b>	<b>82</b>
	<b>BIBLIOGRAPHY</b>	<b>82</b>

# LIST OF ILLUSTRATIONS

## Figure

2.1	Choosing $r_0$ in each local influence domain. . . . .	17
3.1	Annulus Domain . . . . .	19
3.2	Discretization of the annular domain with (a) no rotation of the collocation points and (b) with rotation of the collocation points. The crosses (+) denote the collocation points. . . . .	20
3.3	Distribution of collocation points on domain . . . . .	21
4.1	Distribution of collocation points in local domains. The four neighboring points for the two centers (X) are highlighted. . . . .	53
5.1	Example 1: Profile of the Exact Solution. . . . .	58
5.2	Example 1: Profile of Relative Error. . . . .	59
5.3	Example 1, Multiquadric (MQ): Maximum relative error vs. shape parameter, $M = 100, N = 100$ . . . . .	60
5.4	Example 1, Inverse Multiquadric (IMQ): Maximum relative error vs. shape parameter, $M = 100, N = 100$ . . . . .	60
5.5	Example 1, Gaussian (GA): Maximum relative error vs. shape parameter, $M = 100, N = 100$ . . . . .	60
5.6	Example 1, Gaussian (GA): Maximum relative error versus shape parameter with $M = 100, N = 100$ . . . . .	61
5.7	Example 1, Poisson Dirichlet problem: Maximum relative error versus $m_x$ using MQ. . . . .	61
5.8	Example 2: Profile of Relative Error. . . . .	64
5.9	Example 2, First Biharmonic Problem: Maximum relative error versus shape parameter with $M = 100, N = 100$ using MQ. . . . .	65
5.10	Example 3: Profile of the Exact Solutions, $u_1$ and $u_2$ , respectively. . . . .	68
5.11	Example 3, Cauchy Navier Dirichlet problem: Maximum relative error versus shape parameter with $M = 100, N = 100$ using MQ RBF. . . . .	69
5.12	Example 4, Poisson Dirichlet problem: Maximum relative error versus dimensionless shape parameter with $M = N = 100$ using MQ RBF. . . . .	71
5.13	Example 5: First Biharmonic problem: Error versus shape parameter for the cases $M = N = 80, 200, n = 30$ using MQ RBF. . . . .	74
5.14	Example 6: Cauchy Navier problem. Maximum relative error versus shape parameter for $M = N = 100, n = 9$ using MQ RBF. . . . .	77

## LIST OF TABLES

### Table

2.1	Globally supported RBFs. . . . .	7
5.1	Example 1, Optimal Shape Parameter $c$ and Max Rel Error for MQ, IMQ, and Gaussian RBFs . . . . .	59
5.2	Example 1, Poisson Dirichlet Problem: Max Rel Error and shape parameter using MQ. . . . .	62
5.3	Example 1, Poisson Dirichlet problem: Max Rel Error and shape parameter using MQ up to 1 million points . . . . .	62
5.4	Example 1, Poisson Dirichlet problem: Max Rel Error with $c = 90$ using Gaussian RBF. . . . .	62
5.5	Example 1, Poisson Neumann problem: Max Rel Error and shape parameter using MQ RBF. . . . .	63
5.6	Example 1, Poisson Dirichlet problem: Comparison of Errors, $E$ , and CPU times for full RBF-DQ and RBF-DQ MDA solutions for various $M = N$ with initial search interval $[0, 8]$ . . . . .	64
5.7	Example 2, First Biharmonic Problem: Max Rel Error and shape parameter using MQ. . . . .	65
5.8	Example 2, First Biharmonic problem: Max Rel Error with $c = 60$ using Gaussian RBF. . . . .	66
5.9	Example 2, Second Biharmonic problem: Max Rel Error and shape parameter using MQ RBF. . . . .	66
5.10	Example 2, Second Biharmonic problem: Max Rel Error for $c = 60$ using Gaussian RBF. . . . .	66
5.11	Example 3, Cauchy Navier Dirichlet problem: Max Rel Error and shape parameter using MQ RBF. . . . .	67
5.12	Example 3, Cauchy Navier Neumann/Dirichlet problem: Max Rel Error and shape parameter using MQ RBF. . . . .	68
5.13	Example 3, Cauchy Navier Dirichlet problem: Max Rel Error with $c = 90$ using Gaussian RBF. . . . .	68
5.14	Example 3, Cauchy Navier Neumann/Dirichlet problem: Max Rel Error with $c = 90$ using Gaussian RBF. . . . .	69
5.15	Example 4, Poisson Dirichlet problem: Max Rel Error and optimal shape parameter with various search intervals of LOOCV for $M = N = 200$ using MQ RBF. . . . .	70
5.16	Example 4, Poisson Dirichlet problem: Max Rel Error and optimal shape parameter via LOOCV for $n = 9$ using MQ RBF. . . . .	70

5.17	Example 4, Poisson Neumann problem: Max Rel Error and optimal shape parameter via LOOCV for $n = 9$ using MQ RBF. . . . .	71
5.18	Example 4, Poisson Dirichlet problem $M = N = 200$ : Max Rel Error for dimensionless shape parameter using MQ RBF. . . . .	72
5.19	Example 4, Poisson Dirichlet problem: Max Rel Error with $c = MN/1000$ using MQ RBF. . . . .	72
5.20	Example 4, Poisson Neumann problem: Max Rel Error with $c = MN/1000$ using MQ RBF. . . . .	73
5.21	Example 4, Poisson Dirichlet problem: Influence Domain for $M=N=500$ using MQ RBF. . . . .	73
5.22	Example 5, First Biharmonic problem: Max Rel Error and optimal shape parameter via LOOCV for $M = N = 80$ using MQ RBF. . . . .	75
5.23	Example 5, Second Biharmonic problem: Max Rel Error and optimal shape parameter via LOOCV for $M = N = 80$ using MQ RBF. . . . .	75
5.24	Example 5, First Biharmonic problem: Max Rel Error versus shape parameter with various LOOCV initial search intervals for $M = N = 200$ using MQ RBF. . . . .	75
5.25	Example 5, Second Biharmonic problem: Max Rel Error versus shape parameter with various LOOCV initial search intervals for $M = N = 200$ using MQ RBF. . . . .	76
5.26	Example 5, Second Biharmonic Problem: Shape parameters and corresponding errors obtained using dimensionless shape parameter. . . . .	76
5.27	Example 6, Cauchy Navier Dirichlet problem: Shape parameters and corresponding errors obtained with various search intervals of LOOCV; $M = N = 100$ , $n = 9$ using MQ RBF. . . . .	77
5.28	Example 6, Cauchy Navier Dirichlet problem: Shape parameters and corresponding errors obtained using LOOCV with local domain size $n = 9$ using MQ RBF. . . . .	77
5.29	Example 6, Cauchy Navier Neumann problem: Shape parameters and corresponding errors obtained using various search intervals of LOOCV; $M = N = 100$ , $n = 9$ using MQ RBF. . . . .	78
5.30	Example 6, Cauchy Navier Neumann problem: Shape parameters and corresponding errors obtained using dimensionless shape parameter using MQ RBF. . . . .	78

## LIST OF ABBREVIATIONS

<b>BEM</b>	-	Boundary Element Method
<b>CFD</b>	-	Computational Fluid Dynamics
<b>CS-RBF</b>	-	Compactly Supported Radial Basis Function
<b>DEM</b>	-	Discrete Element Method
<b>DQ</b>	-	Differential Quadrature
<b>FDM</b>	-	Finite Difference Method
<b>FEM</b>	-	Finite Element Method
<b>FVM</b>	-	Finite Volume Method
<b>IMQ</b>	-	Inverse Multiquadric
<b>LOOCV</b>	-	Leave-one-out Cross Validation
<b>LRBF-DQ</b>	-	Local Radial Basis Function Differential Quadrature
<b>MAE</b>	-	Maximum Absolute Error
<b>MDA</b>	-	Matrix Decomposition Algorithm
<b>MFS</b>	-	Method of Fundamental Solutions
<b>MPS</b>	-	Method of Particular Solutions
<b>MQ</b>	-	Multiquadric
<b>NMQ</b>	-	Normal Multiquadric
<b>PDE</b>	-	Partial Differential Equation
<b>RBF</b>	-	Radial Basis Functions
<b>RBFCM</b>	-	Radial Basis Function Collocation Method
<b>RBF-DQ</b>	-	Radial Basis Function Differential Quadrature
<b>RMSE</b>	-	Root-Mean-Square Error
<b>SVD</b>	-	Singular Value Decomposition
<b>TPS</b>	-	Thin Plate Spline

# NOTATION AND GLOSSARY

## General Usage and Terminology

The notation used in this dissertation represents fairly standard mathematical and computational usage. Standard fonts are used to denote sets of numbers:  $\mathbb{R}$  for the field of real numbers,  $\mathbb{C}$  for the complex field,  $\mathbb{Z}$  for the integers, and  $\mathbb{Q}$  for the rationals. Functions in boldface type represent vector valued functions. The caligraphic letters  $\mathcal{L}$  and  $\mathcal{B}$  denote partial differential operators and the capital letters,  $A, B, \dots$  are used to denote matrices.

It is common to denote  $u$  as a function of several variables,  $u(x, y, t)$ . Functions are generally defined on some region  $\Omega$  of  $\mathbb{R}^2$ . A partial differential equation (PDE) involves one or more partial derivatives of an unknown function of several variables. If all partial derivatives of  $u = u(x, y)$  are continuous in a region  $\Omega$  of  $\mathbb{R}^2$ , the Laplacian of  $u$  is,

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

Generally, norms are represented by using double pairs of lines, i.e.,  $\|\cdot\|$ , and the absolute value of numbers is denoted using a single pairs of lines, i.e.,  $|\cdot|$ . The determinant of the matrix is denoted by single pairs of lines around the matrices.

## Chapter 1

### INTRODUCTION

#### 1.1 Background

Partial Differential Equations (PDEs) are mathematical equations that are significant in modeling physical phenomena that occur in nature. Applications can be found in physics, engineering, mathematics, and finance. Examples include modeling mechanical vibration, heat, sound vibration, elasticity, and fluid dynamics, just to name a few. Although PDEs have a wide range of applications to real world problems in science and engineering, the majority of PDEs do not have analytical solutions. It is, therefore, important to be able to obtain an accurate solution numerically.

The advancements of high-speed computers have made it possible to find numerical solutions to complex PDEs while minimizing the time it requires to perform the computations. Many computational methods have been developed and implemented to successfully approximate solutions. Traditionally, mesh methods such as the finite difference method (FDM), finite element method (FEM), and boundary element method (BEM) have been used [4, 43, 44]. These methods require a mesh to connect nodes inside the computational domain or on the boundary. Complications of these methods include a slow rate of convergence, spatial dependence, instability, low accuracy, and difficulty of implementation in complex geometries. However, meshless approximation techniques using radial basis functions (RBFs) have been developed over the last several decades. These techniques are easy to implement, highly accurate, and truly meshless, which avoids troublesome mesh generation for high-dimensional problems.

Edward Kansa [26, 27] first introduced the original radial basis function collocation method (RBFCM) in 1990 also known as the Kansa method. While the mesh based methods require every node to be connected to its neighboring nodes, the nodes in the RBFCM has no nodal connectivity and thus can easily handle complex geometries. Additional advantages include stability with a high rate of convergence, spatial independence, and infinitely differentiable. Many RBFs have been used in the meshless literature including multiquadric (MQ), inverse multiquadric (IMQ), Gaussian, and thin plate splines being the most commonly used globally supported RBFs [7, 26, 27, 18]. Polynomials and trigonometric functions can also be used as basis functions [36].

## 1.2 Literature Review

Over recent decades, meshless methods using RBFs have evolved, and a number of different techniques have been developed for solving partial differential equations, such as the Kansa method, method of approximate particular solutions, method of fundamental solutions, and singular boundary method, just to name a few. In addition to these well known methods, the radial basis function differential quadrature method (RBF-DQ) is a significant and effective method. Bellman [1] first proposed the differential quadrature (DQ) method when searching for a method that only required a few grid points in order to obtain accurate numerical results. Borrowing from the integral quadrature where an integral on a closed domain is approximated by a linear combination of functional values at all nodes, the differential quadrature approximates a derivative of a function with respect to a coordinate at a node using a linear combination in the whole domain in that coordinate direction. Wu and Shu [54] combined this differential quadrature method with radial basis functions and referred to it as the RBF-DQ method. This method is simple to implement, ensures non-singularity, and is appropriate for both linear and non-linear problems.

While this method along with other methods have been proven to be effective, a significant disadvantage to these methods is that they are not able to handle the solution of problems on a large scale. In this case, the matrices can become dense and are often poorly conditioned. Not only is stability an issue, but the computational cost and memory allocation become very high thereby rendering the method ineffective. Furthermore, the solution is very sensitive to the choice of the shape parameter in the RBFs. To circumvent these issues, a number of efforts have been made in the literature including a domain decomposition introduced by Main-Duy and Tran-Cong [40], a multi-grid approach in [50], compact support radial basis functions by Chen *et al.* [9], the greedy algorithm [46, 24], extended precision arithmetic [25], the improved truncated singular value decomposition method [38], and local methods such as the local Kansa method [35] and local Method of Approximate Particular Solutions [50].

In the local approach, instead of solving the problem using all of the collocation points, a local influence domain is created for each point where only a small number of neighboring points is used to approximate the solution. By repeatedly solving for the coefficients using the small local domains, the resultant coefficient matrix becomes sparse, which allows for the use of a large number of collocation points. Also, since a low number of collocation points is used, the shape parameter naturally will become more stable. While the shape parameter for each local domain theoretically is different, this difference will not be significant, and using one shape parameter for all local domains yields acceptable results. While this method



does not achieve the same accuracy as the global method, it allows a large scale problem to be solved while still resulting in an acceptable approximation [5, 6]. As an alternative to these methods, a matrix decomposition algorithm (MDA) can be used to resolve the issues that arise with large scale problems. This MDA has the ability of reducing the solution of an algebraic problem to the solution of a set of independent systems of lower dimension. Further, by appropriately choosing the collocation points lying on concentric circles in the domain, linear systems can be obtained in which the coefficient matrices will have block circulant structures. By utilizing the properties of circulant matrices, additional computational time savings can be realized.

Another disadvantage to RBF methods is that the determination of an optimal shape parameter is often problem-dependent and continues to be an area of research. While the leave-one-out cross validation (LOOCV) method is effective in finding a sub-optimal shape parameter, the computational time is an issue and becomes a burden for large scale problems. In this dissertation, a modification of the Fausshauer estimate is explored, which can be effective in estimating a good value for the shape parameter without requiring a significant computational cost. An additional parameter that can affect the outcome is the selection of the radial basis function. Different RBFs will also be considered to determine which is most effective.

In this dissertation, a matrix decomposition algorithm (MDA) is implemented in order to reduce the dense collocation matrix to a diagonal matrix. This will in effect reduce the solution of the PDE to the solution of a set of independent systems of lower dimension. The decomposition can be implemented efficiently using fast Fourier Transforms (FFTs). Since the global structure is maintained in this scheme, the same high accuracy is able to be achieved in the numerical results. Such MDAs have been implemented in a variety of methods, including the Kansa method and the method of fundamental solutions (MFS) [29, 30]. MDA leads to substantial savings in computational cost and memory and can be further coupled with the local method to further improve time and memory savings.

Finally, this dissertation will:

- address the problem of solving large scale problems by implementing a matrix decomposition algorithm.
- examine the attractive properties of circulant matrices and explore how these properties can contribute to computational savings.
- explore the selection of the radial basis function and the effect it has on the accuracy of the solution.

- test methods for selecting an appropriate shape parameter that minimizes the error of the RBF-DQ method.
- explore the pairing of the local radial basis function differential quadrature method with the matrix decomposition algorithm.

### 1.3 Synopsis

The purpose of this study is to pair a matrix decomposition algorithm with the radial basis function differential quadrature method for solving elliptic PDEs:

$$\begin{aligned}\mathcal{L}u &= f(x,y), & (x,y) \in \Omega, \\ \mathcal{B}u &= g(x,y), & (x,y) \in \partial\Omega,\end{aligned}\tag{1.1}$$

where  $\mathcal{L}$  and  $\mathcal{B}$  are linear partial differential operators,  $\Omega$  is a bounded domain in  $\mathbb{R}^2$ , and  $\partial\Omega$  is the boundary of the domain  $\Omega$ .

In this dissertation, a numerical scheme Radial Basis Function Differential Quadrature method (RBF-DQ) is combined with a matrix decomposition algorithm (MDA) for solving various types of PDEs, which is applied both globally and locally. This numerical scheme is direct, less ill-posed, easy to implement, and highly accurate. Also, this can be easily extended to higher order elliptic PDEs and in higher dimensions as well.

Chapter 2 begins with a review of state-of-the-art meshless methods focusing particularly on the methods and techniques used in this dissertation. The RBF collocation method, known as Kansa method, is introduced. The method of particular solutions (MPS) is also briefly discussed. The RBF-DQ method is then introduced along with the local scheme. A discussion of the techniques for choosing an appropriate shape parameter follows, including the Leave-one-out cross validation (LOOCV) and other techniques, such as the adjusted Faussbauer estimate and dimensionless shape parameter.

Chapter 3 introduces the matrix decomposition algorithm (MDA) in detail. The relationship between similar matrices and their eigenvalues is first introduced followed by a strategy for choosing collocation points, which is necessary in order for a circulant structure to be achieved. Circulant matrices are then defined along with important properties and theorems that is necessary for the matrix decomposition algorithm. Once the circulant structure is established, the fast Fourier transform is then applied to decompose the matrices.

In Chapter 4, the RBF-DQ method is combined with the MDA to solve partial differential equations along using both the global and local scheme. Examples include the Poisson equation, Biharmonic equation, and the Cauchy-Navier problem of elasticity. Details will be

given to transform a matrix with a non-circulant structure to one that is circulant as required for the Cauchy-Navier problem.

Chapter 5 consists of numerical results utilizing the global and local RBF-DQ MDA schemes. Various numerical experiments are performed to test the numerical accuracy of the proposed methods on the problems governed by the Poisson equation, Biharmonic equation, and the Cauchy Navier problem with different types of boundary conditions including the Dirichlet and Neumann boundary conditions. The time, memory, accuracy, and selection of the shape parameter are of particular interest.

Conclusions from the numerical results and possible future works are listed in Chapter 6.

## Chapter 2

### STATE-OF-THE-ART IN MESHLESS METHOD

#### 2.1 Radial Basis Function (RBF) Interpolation

Radial basis functions have been widely used in applications for solving a variety of science and engineering problems including function interpolation and solving PDEs. The RBF collocation method for interpolation was created by making an extension of the piecewise polynomial interpolation using a function of Euclidean distance which is defined as follows [4]:

**Definition 2.1.1.** Given a set of  $N$  distinct data points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ , and corresponding data values  $f_1, \dots, f_N$ , the RBF interpolant is given by

$$u(x) = \sum_{i=1}^N \alpha_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|)$$

where  $\varphi$  is some radial basis function,  $\|\cdot\|$  represents the Euclidean norm, and coefficients  $\{\alpha_i\}_{i=1}^N$  are determined by using the interpolation condition  $u(\mathbf{x}_i) = f_i$ ,  $i = 1, 2, \dots, N$ , which leads to the following linear system:

$$\mathbf{A}\alpha = \mathbf{f},$$

where the entries of  $\mathbf{A}$  are  $A_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ ,  $i, j = 1, 2, \dots, N$ ,  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$  and  $\mathbf{f} = [f_1, f_2, \dots, f_N]^T$ .

There are a number of globally supported radial basis functions that are widely used in the literature including those in Table 2.1.

In this chapter, several basic meshless methods are briefly introduced, such as the radial basis function collocation method (RBF-CM), also referred to as the Kansa method, the method of particular solutions (MPS), the RBF Differential Quadrature method (RBF-DQ), and the local RBF-DQ method (LRBF-DQ). The RBF-DQ and LRBF-DQ are the main focus of the chapter. The intent is to combine both the global and local approaches with a matrix decomposition algorithm in the remaining chapters, so that this method can be used for solving large scale problems.

Table 2.1: Globally supported RBFs.

Types of basis function	$\varphi(r), (r \geq 0)$
Multiquadric (MQ)	$\sqrt{r^2 + c^2}$
Normal Multiquadric (NMQ)	$\sqrt{1 + (rc)^2}$
Inverse multiquadric (IMQ)	$\frac{1}{\sqrt{r^2 + c^2}}$
Thin plate spline (TPS)	$r^2 \ln(r)$
Gaussian	$e^{-cr^2}$
Conical	$r^{2n-1}$

## 2.2 The Collocation (Kansa) Method

The Kansa method, the original radial basis function collocation method (RBFCM), is a well-known meshless method. First developed by Kansa [26] in 1990, the method uses RBFs to solve both linear and nonlinear PDEs. The Kansa method has become very popular due to its simplicity and effectiveness. The method has been applied to physics, material science, and a variety of engineering problems [4]. The novelty of the Kansa method is that it does not require any mesh, making it flexible in the geometrical sense. The only geometric property used is the distance between points in the computational domain. This allows the extension to higher order dimensions without the method increasing in difficulty. While the Kansa method has a high-order accuracy, issues still remain such as stability for complex time dependent problems [34].

To demonstrate the Kansa method, consider the following boundary value problem:

$$\mathcal{L}u(\mathbf{x}) = f(\mathbf{x}), \mathbf{x} \in \Omega, \quad (2.1)$$

$$\mathcal{B}u(\mathbf{x}) = g(\mathbf{x}), \mathbf{x} \in \partial\Omega, \quad (2.2)$$

where  $\mathcal{L}$ ,  $\mathcal{B}$  are differential operators,  $f$  and  $g$  are known functions,  $\Omega$  is a computational domain, and  $\partial\Omega$  is the boundary of  $\Omega$ . Let  $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{n_i}$  be  $n_i$  distinct interior collocation points in  $\Omega$  and  $\{(\mathbf{x}_i, g(\mathbf{x}_i))\}_{i=n_i+1}^N$  boundary points such that  $n_b$  are the number of boundary nodes and  $N = n_i + n_b$ .

The collocation technique reduces the boundary value problem to a discrete problem by imposing finitely many conditions [10]:

$$\mathcal{L}u(\mathbf{x}_j) = f(\mathbf{x}_j), j = 1, 2, \dots, n_i, \quad (2.3)$$

$$\mathcal{B}u(\mathbf{x}_j) = g(\mathbf{x}_j), j = n_i + 1, n_i + 2, \dots, N. \quad (2.4)$$

In the Kansa method, the primary idea is to approximate the solution  $u$  by a linear combination of the RBFs,

$$\hat{u}(\mathbf{x}) = \sum_{i=1}^N \alpha_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|), \quad (2.5)$$

where  $\{\alpha_i\}_{i=1}^N$  are undetermined coefficients. Then we have

$$\mathcal{L}\hat{u}(\mathbf{x}) = \sum_{i=1}^N \alpha_i \mathcal{L}\varphi(\|\mathbf{x} - \mathbf{x}_i\|), \quad (2.6)$$

and

$$\mathcal{B}\hat{u}(\mathbf{x}) = \sum_{i=1}^N \alpha_i \mathcal{B}\varphi(\|\mathbf{x} - \mathbf{x}_i\|). \quad (2.7)$$

From (2.1) and (2.2), we have

$$\sum_{i=1}^N \alpha_i \mathcal{L}\varphi(\|\mathbf{x}_j - \mathbf{x}_i\|) = f(\mathbf{x}_j), j = 1, 2, \dots, n_i, \quad (2.8)$$

and

$$\sum_{i=1}^N \alpha_i \mathcal{B}\varphi(\|\mathbf{x}_j - \mathbf{x}_i\|) = g(\mathbf{x}_j), j = n_i + 1, \dots, N. \quad (2.9)$$

This is a square linear system for which the coefficients  $\{\alpha_i\}_{i=1}^N$  can be determined. There are a number of different RBFs that can be used in the Kansa method, although MQ is one of the most widely adopted.

### 2.3 The Method of Particular Solutions

Chen *et al.* [7, 8] proposed a method to solve inhomogeneous problems (2.1)-(2.2) without the requirement of finding the homogeneous solution, known as the method of particular solutions (MPS). A particular solution of a boundary value problem is defined as a function  $u_p$  which satisfies the inhomogeneous differential equation

$$\mathcal{L}u_p(\mathbf{x}) = f(\mathbf{x}), \mathbf{x} \in \Omega. \quad (2.10)$$

It is not a requirement for the particular solution to satisfy the boundary condition (2.2). This would imply that the particular solution does not have uniqueness. In order to apply the MPS, a solution,  $\Phi(r)$ , must be known in advance for the differential equation:

$$\mathcal{L}\Phi(r) = \varphi(r), \quad (2.11)$$

where  $\varphi(r)$  is a chosen radial basis function.

We can construct a particular solution  $u_p(\mathbf{x})$  in (2.10) by interpolating the forcing term  $f(\mathbf{x})$  using RBFs,  $\varphi(r)$ ,

$$f(\mathbf{x}) = \sum_{i=1}^{n_i} \alpha_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|). \quad (2.12)$$

By (2.11), an approximate particular solution of (2.10) can be written as

$$\hat{u}_p(\mathbf{x}) = \sum_{i=1}^{n_i} \alpha_i \Phi(\|\mathbf{x} - \mathbf{x}_i\|). \quad (2.13)$$

The coefficients  $\{\alpha_i\}_{i=1}^N$  can then be determined by the same collocation technique described in the previous section.

Since the differential operator and the RBFs are radially invariant, the solutions  $\Phi(r)$  must also be radially invariant. Thus, the MPS is a special kind of RBF collocation method. Unlike the Kansa method using  $\varphi(r)$ , the MPS uses  $\Phi(r)$ . This is an important tool for evaluating solutions of PDEs. The primary difference between the MPS and Kansa method is the utilization of the RBF,  $\varphi(r)$ , as a basis for the Kansa method, while the MPS uses the solution of (2.11) as the basis function. While it is necessary to find the closed form of  $\Phi$  for different operators, deriving  $\Phi(r)$  for higher order differential operators has proved to be a challenge. Also, this closed form depends on the dimension of the domain space.

## 2.4 Radial Basis Function Differential Quadrature Method

In this section, the radial basis function differential quadrature (RBF-DQ) method is presented. The differential quadrature (DQ) method is a numerical discretization technique that approximates the derivative of a function  $f(\mathbf{x})$  with respect to a coordinate using a linear combination of function values in the domain of that coordinate direction as shown in [1]. The idea of the DQ method came from the integral quadrature, where the integral over a closed domain is approximated by a linear combination of function values at all nodes, such as the well known Trapezoidal Rule and Simpson's Rule that students learn in an introductory Calculus course. A variety of methods have been developed based on the DQ method, including the polynomial-based differential quadrature (PDQ) and the Fourier-expansion-based differential quadrature (FDQ) [51, 49]. In the PDQ, the coefficients are determined by a simple algebraic formulation or a recurrence relationship that is independent of the selection of the nodal points. In a similar approach, the FDQ uses the Fourier series expansion to approximate the function. While the PDQ and FDQ methods are able to obtain accurate results using only a small number of grid points, they are mesh-based methods. Moreover, the distribution of the nodes has limitations, and the mesh must be clustered close to the boundary limiting its usefulness.

In 2002, Wu and Shu [54] applied radial basis functions (RBF) to the DQ method to take advantage of the naturally meshless method that RBFs provide. In the RBF-DQ method, the coefficients are determined using radial basis functions. This has been found to be very easy to implement while ensuring that the collocation matrix is non-singular. Since the functional values at each node represent the dependent variables, this method can also be suitable for non-linear problems.

In the RBF-DQ method, the primary idea for solving an nonhomogeneous problem (2.1)-(2.2) is to approximate the derivative of a function  $u(\mathbf{x})$  using a linear combination of the functional values at given nodes. Consider the following example:

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

The approximation for the derivatives of  $u(\mathbf{x})$  at the  $i^{th}$  node for this method can be written as,

$$\begin{aligned} \frac{\partial u(\mathbf{x}_i)}{\partial x} &= \sum_{j=1}^N a_{ij} u(\mathbf{x}_j) \\ \frac{\partial^2 u(\mathbf{x}_i)}{\partial x^2} &= \sum_{j=1}^N b_{ij} u(\mathbf{x}_j) \\ \frac{\partial^2 u(\mathbf{x}_i)}{\partial y^2} &= \sum_{j=1}^N c_{ij} u(\mathbf{x}_j), \end{aligned} \quad (2.14)$$

where  $N$  is the total number of collocation points used for approximating the derivatives at each node  $\mathbf{x}_i$ , and  $a_{ij}, b_{ij}, c_{ij}$  are the coefficients of  $\frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}$ , respectively. Determining the coefficients  $a_{ij}, b_{ij}, c_{ij}$  can be achieved using radial basis functions. Consider the first order derivative:

$$\frac{\partial \varphi_k(\mathbf{x}_i)}{\partial x} = \sum_{j=1}^N a_{ij} \varphi_k(\mathbf{x}_j), \quad (2.15)$$

where  $\varphi_k(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{x}_k\|)$ . Equation (2.15) can be written in matrix form as

$$\begin{bmatrix} \frac{\partial \varphi_1(\mathbf{x}_i)}{\partial x} \\ \frac{\partial \varphi_2(\mathbf{x}_i)}{\partial x} \\ \vdots \\ \frac{\partial \varphi_N(\mathbf{x}_i)}{\partial x} \end{bmatrix} = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \varphi_1(\mathbf{x}_2) & \dots & \varphi_1(\mathbf{x}_N) \\ \varphi_2(\mathbf{x}_1) & \varphi_2(\mathbf{x}_2) & \dots & \varphi_2(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_N(\mathbf{x}_1) & \varphi_N(\mathbf{x}_2) & \dots & \varphi_N(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{iN} \end{bmatrix}, \quad (2.16)$$

or in vector form,

$$\frac{\partial \varphi(\mathbf{x}_i)}{\partial x} = G a_i. \quad (2.17)$$

The coefficient vector  $a_i$  can then be found,

$$a_i = G^{-1} \frac{\partial \varphi(\mathbf{x}_i)}{\partial x}. \quad (2.18)$$



The RBF-DQ method inherits the attractive quality from  $\varphi(\|\mathbf{x} - \mathbf{x}_j\|)$  that  $\mathbf{G}$  will always be invertible, so there is no question whether or not  $\mathbf{G}$  is singular. Also, it is important to observe that the matrix  $G$  is independent of  $\mathbf{x}_i$  so the coefficients at each center can easily be found by simply changing the vector  $\frac{\partial \varphi(\mathbf{x}_i)}{\partial \mathbf{x}}$ . Also, since  $G$  is fixed, it is only necessary to compute this collocation matrix once and then use it to solve for the remaining coefficients,  $b_{ij}$  and  $c_{ij}$ . Once the coefficients have been computed using Eq. (2.14)-(2.18), it is now possible to solve for  $u$ ,

$$\sum_{j=1}^N a_{ij}u(\mathbf{x}_j) + \sum_{j=1}^N b_{ij}u(\mathbf{x}_j) + \sum_{j=1}^N c_{ij}u(\mathbf{x}_j) = f(\mathbf{x}_i). \quad (2.19)$$

By rearranging this can be written as

$$\sum_{j=1}^N (a_{ij} + b_{ij} + c_{ij})u(\mathbf{x}_j) = f(\mathbf{x}_i) \quad (2.20)$$

or in vector form,

$$Au = f, \quad (2.21)$$

where

$$A = \begin{pmatrix} a_{11} + b_{11} + c_{11} & a_{12} + b_{12} + c_{12} & \cdots & a_{1N} + b_{1N} + c_{1N} \\ a_{21} + b_{21} + c_{21} & a_{22} + b_{22} + c_{22} & \cdots & a_{2N} + b_{2N} + c_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} + b_{N1} + c_{N1} & a_{N2} + b_{N2} + c_{N2} & \cdots & a_{NN} + b_{NN} + c_{NN} \end{pmatrix}.$$

By solving the above system,  $u$  can now be found.

The coefficients are dependent only on the chosen RBFs, which means that it is necessary to compute them only once. Furthermore, the RBF-DQ scheme does not require a mesh and is not sensitive to the spatial dimension, making it a multi-dimensional meshless method for solving PDEs.

## 2.5 Local Radial Basis Function Differential Quadrature Method

In the global RBF-DQ method, the RBFs are used to obtain the coefficients so that the derivatives of a function  $f(x)$  can be written as a linear combination of the functional values at the predetermined nodes. This numerical scheme is simple and effective. However, the approximation can become unstable as the number of collocation points become large resulting in dense matrices. This leads to ill-conditioning and sensitivity to the shape parameters in the RBF formulation. We can alleviate these issues by using a localized formulation. A number of RBF collocation methods have been localized, including the Kansa

method and MAPS [55, 56]. In these local methods, since only neighboring collocation points are needed, it does not have the typical ill-conditioning that comes with large, dense matrix systems. Additionally, the shape parameter of the RBF does not vary much, which is beneficial in choosing a good value. Another advantage of the local approach is that the accuracy is not compromised by the computational efficiency. Unlike the global approach where it is required to work with a dense matrix, the local approach results in a sparse matrix which can be solved efficiently.

Shu *et al.* [50] proposed a local approach to the RBF-DQ method, known as the local radial basis function differential quadrature (LRBF-DQ) method. The derivatives of a function  $u(\mathbf{x})$  can be approximated at a point  $\mathbf{x}_i$  using a linear combination of the functional values  $u(\mathbf{x}_j)$  within the local domain  $\Omega_i = \{\mathbf{x}_j\}_{j=1}^n$ . The union of all local influence domains covers the global domain.

The approximation for  $\frac{\partial u_k(\mathbf{x}_i)}{\partial \mathbf{x}}$  at the  $i^{\text{th}}$  node can be written as:

$$\frac{\partial u_k(\mathbf{x}_i)}{\partial x} = \sum_{j=1}^n a_{ij} u(\mathbf{x}_j). \quad (2.22)$$

The corresponding coefficients  $a_{ij}$  can be achieved as follows:

$$\frac{\partial \varphi_k(\mathbf{x}_i)}{\partial x} = \sum_{j=1}^n a_{ij} \varphi_i(\mathbf{x}_j), \quad (2.23)$$

where  $\varphi_i(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{x}_i\|)$ . Assume that within each  $\Omega_i$ , using the collocation method, the resulting linear system is obtained

$$\begin{bmatrix} \frac{\partial \varphi_1(\mathbf{x}_i)}{\partial x} \\ \frac{\partial \varphi_2(\mathbf{x}_i)}{\partial x} \\ \vdots \\ \frac{\partial \varphi_n(\mathbf{x}_i)}{\partial x} \end{bmatrix} = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \varphi_1(\mathbf{x}_2) & \dots & \varphi_1(\mathbf{x}_n) \\ \varphi_2(\mathbf{x}_1) & \varphi_2(\mathbf{x}_2) & \dots & \varphi_2(\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_n(\mathbf{x}_1) & \varphi_n(\mathbf{x}_2) & \dots & \varphi_n(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix}. \quad (2.24)$$

or in vector form,

$$\frac{\partial \varphi(\mathbf{x}_i)}{\partial x} = G a_i. \quad (2.25)$$

Note that  $G$  is a different matrix for each local domain. Since  $G$  is non-singular, the inverse matrix can be calculated assuming that all of the nodal points inside  $\Omega_i$  are distinct. The unknown coefficients in (2.25) can be written as

$$a_i = G^{-1} \frac{\partial \varphi(\mathbf{x}_i)}{\partial x}, \quad (2.26)$$

where

$$a_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix}$$

and

$$\frac{\partial \phi(\mathbf{x}_i)}{\partial x} = \begin{bmatrix} \frac{\partial \hat{\phi}_1(\mathbf{x}_i)}{\partial x} \\ \frac{\partial \hat{\phi}_2(\mathbf{x}_i)}{\partial x} \\ \vdots \\ \frac{\partial \hat{\phi}_n(\mathbf{x}_i)}{\partial x} \end{bmatrix}.$$

The vectors  $a_i, i = 1, \dots, N$ , contain all of the required coefficients for the approximations  $\{u(\mathbf{x}_i)\}_{i=1}^N$  in the global system. However, it is necessary to distribute  $a_i$  appropriately in the global matrix. This can be accomplished by padding  $a_i$  with zero entries at the appropriate positions yielding the vector  $\alpha$  [57]. For example, suppose that  $N = 100, n = 3$ , and  $\Omega_i = \{x_{18}, x_{26}, x_{32}\}$ . Then 97 zeros are inserted into the  $n$ -vector,  $a_i$ , given in (2.26). This will pad the vector at all positions except 18, 26, and 32, as shown in

$$\alpha(\mathbf{x}) = [0, 0, \dots, \underbrace{a_1}_{18th}, 0, 0, \underbrace{a_2}_{26th}, 0, 0, 0, \underbrace{a_3}_{32nd}, 0, \dots, \underbrace{0}_{100th}]. \quad (2.27)$$

In (2.27) there are 17 zeros before  $a_1$  and 68 zeros after  $a_3$ . This zero padding keeps track of the original position at each local point so that  $\alpha$  can be easily obtained from  $a_i$ . This can be repeated to solve for any of the remaining coefficients as necessary, such as  $b_{ij}, c_{ij}$ . The assembly of all equations for the  $N$  centers  $\mathbf{x}_i$  will yield the  $N \times N$  system

$$Au = f, \quad (2.28)$$

where the global matrix  $A$  is sparse. The above sparse system can now be solved to approximate  $u(\mathbf{x})$ .

This local method is effective for solving large-scale problems. By pairing this method with the matrix decomposition algorithm, additional computational savings can be realized as will be demonstrated in Chapter 5.

## 2.6 Shape Parameter

In the previous sections, the global and local RBF-DQ schemes were described in detail. It is important to note that this scheme is appropriate only if the collocation matrix,  $G$ , is invertible in (2.17) so that the coefficients  $a_i$  can be obtained. Schoenberg [47] proved

the invertibility of the collocation matrix for positive definite RBFs, Gaussian, and Inverse Multiquadric (IMQ). Multiquadric (MQ) is only conditionally positive definite. However, it is conditionally positive definite of order one, which Micchelli established the invertibility of  $G$  for this class of RBFs [42]. He further proved the invertibility of the collocation matrix for conditionally positive definite functions of order  $m$  given certain requirements, such as polyharmonic splines. The interpolation matrix of this class of RBFs can be singular even for non-trivial sets of distinct centers. To guarantee invertibility, it is necessary to add polynomial terms to the RBF interpolation problem.

Among the RBFs, the multiquadric (MQ) is perhaps the most popular that is used in applications. This is due to its high convergence rate and accuracy. Perhaps a disadvantage that the MQ has, compared to the polyharmonic splines, is that the accuracy depends on the correct choice of the shape parameter. This same disadvantage applies to the inverse multiquadric (IMQ) and the Gaussian (GA) RBF as well. Finding the appropriate shape parameter that minimizes the approximation error is not trivial and is an ongoing research problem. In 2001, Main-Duy and Tran-Cong [40] claimed that the shape parameter relates to the grid distance. However, other researchers such as Zhang *et al.* in [58] argue that the optimal shape parameter is dependent upon the problem itself. In 2003, Lee *et al.* [35] claimed that the numerical solution is less sensitive to the selection of the shape parameter in the local collocation methods than the global methods. A number of different techniques have been developed throughout RBF literature for finding an optimal shape parameter. Of course, the simplest strategy is the trial and error method where the best value of the shape parameter is found by performing a number of experiments with different shape parameter values. This method is certainly not sophisticated and can only be used with certainty if we know the original function. In one of the earliest papers on the topic of RBFs, Hardy [20] suggested the value  $c = 1/(0.815d)$ , where  $d = 1/N \sum_{j=1}^N d_j$  and  $d_j$  is the distance of the  $j^{th}$  data value from its nearest neighbor. Franke, in 1982, [19] recommended an alternative value of  $c = 0.8\sqrt{N}/D$ , where  $D$  is the diameter of the smallest circle containing the data points  $\{\mathbf{x}\}_{j=1}^N$ , for the MQ RBF. Both approaches yielded satisfactory approximations. However, at that time, only single precision computers were used. Today, double precision computation has caused these shape parameter estimates to become obsolete.

More recently, Fasshauer [16] proposed the value  $c = 2\sqrt{N}$  for interpolation in a regular domain two dimensional problem. Fasshauer later suggested a "safe" shape parameter based on Schaback's uncertainty principle [15], which states that it is impossible to achieve both good stability and small errors simultaneously. As the shape parameter approaches zero, the coefficient matrix's condition number grows exponentially. In this strategy, Fasshauer used the smallest value of the shape parameter without MATLAB issuing a warning of

ill-conditioning. While this strategy will not result in an optimal shape parameter, it does at least guarantee that the system will not be near-singular. Rippa [45] used the Leave-One-Out-Cross-Validation (LOOCV) method to find a sub-optimal shape parameter that results in an accurate approximation while maintaining matrix stability. LOOCV is a technique that has been used extensively in RBF literature. In a large scale global scheme, LOOCV becomes ineffective due to the great computational burden it causes. However, it can still present usefulness in a local scheme, such as the local RBF-DQ. The next sections these techniques for calculating the shape parameter in more detail.

### 2.6.1 Fasshauer Estimate

For a large scale problem, the LOOCV is not a practical tactic for finding an optimal shape parameter in a global scheme. Therefore, a more sophisticated method for the shape parameter is desired that will produce high accuracy while being able to handle a large number of collocation points. After extensive experimentation, Karageorghis *et al.* [29] observed that the value  $c = \sqrt[4]{n}/5$ , where  $n$  is the number of collocation points, yielded the most accurate results when using the normalized MQ RBF. Taking this formula a step further, we can allow the denominator to be adjustable,  $c = \sqrt[4]{n}/m_x$ , where  $m_x$  depends on the density of the collocation points. As we will observe in the numerical results, as the number of collocation points increases, the proper value of  $m_x$  decreases. Not only does this formula yield accurate results, it is computationally simple and works well for large scale problems as will be shown in Chapter 5, in addition to a detailed description of how  $m_x$  is chosen in the formula.

### 2.6.2 Leave-One-Out Cross Validation

Cross validation is a statistical technique that tests the accuracy of a method by separating given data into two sections. The first section of the data set is used to calculate an approximation of the data. The error is then measured by finding the difference between the approximation and the actual data in the second section of the data. Leave-one-out cross validation (LOOCV) uses  $n - 1$  of the available  $n$  data points for the approximation. The remaining data point is then used to calculate the error. This procedure is then repeated for each data point, and the set of errors resulting from the  $n$  procedures are used to estimate the relative accuracy. This can be accomplished without the need of computing all of the partial interpolants.

In order to use LOOCV to calculate the optimal shape parameter, first, define the vector of data points with the  $k^{th}$  data point removed  $\mathbf{x}^{[k]} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \dots, \mathbf{x}_n]^T$ . The

approximation of the deleted point  $\mathbf{x}_k$  can then be found,

$$\hat{u}^{[k]}(\mathbf{x}) = \sum_{i=1}^{n-1} \alpha_i \varphi_i(\mathbf{x}^{[k]}). \quad (2.29)$$

The error between the approximation and the actual value at  $\mathbf{x}_k$  can be calculated as

$$e_k = |u(\mathbf{x}_k) - \hat{u}^{[k]}(\mathbf{x}_k)|. \quad (2.30)$$

The accuracy for the entire data can then be found as the norm of the vector of errors  $\mathbf{e} = [e_1, e_2, \dots, e_n]^T$  by removing each of the data points and comparing the approximation with the exact value at the removed data point. The norm of the error vector  $\|\mathbf{e}\|$  is the cost function of the shape parameter. We can then choose the appropriate shape parameter that minimizes  $\|\mathbf{e}\|$ .

This procedure would not be efficient if these  $n$  linear systems had to be solved and would, in fact, increase the computational complexity of the RBF-DQ algorithm. However, this can be done efficiently without the need of calculating the error at every point. Rippa [45] showed that the error can be calculated using the formula

$$e_k = \frac{\alpha_k}{A_k^{-1}}, \quad (2.31)$$

where  $\alpha_k$  is the  $k^{th}$  coefficient of  $\hat{u}$  for the entire data set and  $A_k^{-1}$  is the  $k^{th}$  diagonal element of the inverse of the corresponding interpolation matrix. Once  $\mathbf{e}$  is calculated using the above formula, the relative minimum of the cost function for  $c$  can be found using the MATLAB function **fminbnd**. The calling sequence for the cost function is

$$c = \text{fminbnd}(@(\text{costeps}(c, rbf, DM, rhs), \text{minc}, \text{maxc})), \quad (2.32)$$

where  $DM$  is the distance matrix of  $A$ ,  $\text{minc}$ ,  $\text{maxc}$  is the interval used to search for the sub-optimal shape parameter  $c$  and  $rhs$  represents the right-hand side of the equation  $Ax = b$ . While this technique provides good results, the shape parameter is not necessarily optimal because of its dependence on the chosen search interval ( $\text{minc}$ ,  $\text{maxc}$ ). The LOOCV is shown to be effective for the local RBF-DQ method, however, for the global case of large scale problems, this algorithm proves to be computationally complex. Thus, it is necessary to present other techniques for finding an effective shape parameter.

### 2.6.3 Dimensionless Shape Parameter

While using a single shape parameter value for each local domain is acceptable, a second approach for calculating the shape parameter involves choosing a fixed value and multiplying

by the maximal distance in each local domain, allowing a tailored shape parameter for each domain. In [35], the authors found the choice of the shape parameter of MQ in the local collocation method to be less sensitive than in the global method. The optimal shape parameter seems to depend on the density of nodes, the number of nodes, and the functional value of the nodes in the influence domain. In the LRBF-DQ method, the number of nodes in the local domain is fixed. Assigning different values of the shape parameter for each local domain proves to be very difficult. In [53], the author adopted a dimensionless shape parameter for a local collocation method to reduce this difficulty. This technique produced excellent results using MQ. Let

$$r_0 = \max_{1 \leq j \leq n} \|\mathbf{x}_0 - \mathbf{x}_j\|, \quad (2.33)$$

where  $r_0$  is the maximum distance from the center  $\mathbf{x}_0$  to all nodal points among all local domains  $\Omega_i, i = 1, 2, \dots, N$  as shown in Figure 2.1.

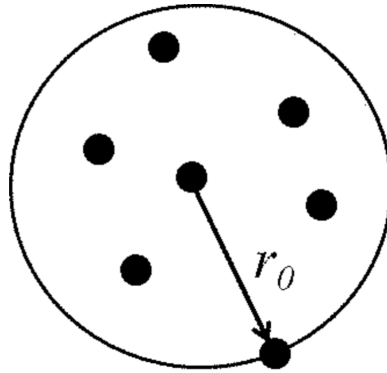


Figure 2.1: Choosing  $r_0$  in each local influence domain.

By using the dimensionless shape parameter, the MQ RBF is changed to the following form

$$\varphi(r) = \sqrt{r^2 + (cr_0)^2}. \quad (2.34)$$

When  $r_0$  is small, a large value for  $c$  is chosen. Overall, the desire is to select the value  $c$  such that  $cr_0$  is a fixed number for various values of  $r_0$ . This approach is also appropriate for other RBFs, such as the normalized MQ (NMQ). The NMQ RBF would then be converted to the form

$$\varphi(r) = \sqrt{1 + (cr_0)^2 r^2}, \quad (2.35)$$

Choosing the dimensionless shape parameter  $c$  becomes less critical due to the wider range of values that will result in an accurate estimation. This makes it easier to choose a shape parameter without sacrificing accuracy. A comparison of the dimensionless shape parameter with the LOOCV for the LRBF-DQ method will be presented in Chapter 5.

## Chapter 3

### MATRIX DECOMPOSITION ALGORITHM

In this section, the matrix decomposition algorithm (MDA) is presented. An MDA [3] is a direct method that can be used to reduce the solution of an algebraic problem into the solution of a set of independent systems having lower dimension, leading to significant memory and computational cost savings. For large scale problems, it is necessary to diagonalize the coefficient matrix to alleviate the computational burden. The fact that the coefficient matrix, as well as the right hand side matrix, are circulant yields the ability to use a Fast Fourier Transform to complete the diagonalization. MDAs have been successfully applied in a variety of RBF techniques [28, 30, 31, 39].

#### 3.1 Matrix Decomposition

Two matrices having the same eigenvalues are known as similar matrices. The eigenvalues of an  $n \times n$  matrix  $A$  are the zeros of its characteristic polynomial  $P(x) = \det(A - xI)$ . An  $n \times n$  matrix  $B$  such that  $B = S^{-1}AS$ , where  $S$  is a nonsingular  $n \times n$  matrix, will have the same characteristic polynomial as  $A$ . This implies that  $A$  and  $B$  have the same eigenvalues, in other words,  $A$  and  $B$  are similar. The computational time for solving a matrix equation, such as  $Ax = b$ , can be reduced if we can diagonalize the matrix  $A$  due to the sparsity of a diagonal matrix.

If  $A$  is diagonalizable, instead of using the full matrix to solve the equation

$$Ax = b, \quad (3.1)$$

$A$  can be diagonalized, and (3.1) can be rewritten as

$$(S^{-1}AS)(S^{-1}x) = S^{-1}b \quad (3.2)$$

$$\hat{A}\hat{x} = \hat{b}, \quad (3.3)$$

where  $\hat{A} = S^{-1}AS$ ,  $\hat{x} = S^{-1}x$ ,  $\hat{b} = S^{-1}b$ . The new equation (3.3) can now be solved, and the solution to the original equation (3.1) can be recovered:

$$\hat{x} = \hat{A}^{-1}\hat{b} \quad (3.4)$$



$$x = S\hat{x}. \quad (3.5)$$

As the size of matrix  $A$  gets large, significant computational time can be saved using the diagonalization process. As mentioned earlier, it is necessary to possess a diagonalizable matrix in order for this process to work. In this dissertation, the matrices that are being used are circulant. The next sections will demonstrate how to construct a circulant matrix and will prove that all circulant matrices are diagonalizable. Details for finding the matrix  $S$  will also be provided.

### 3.1.1 The domain

The domain that is being studied for the elliptic boundary value problems in this dissertation is the annulus  $\Omega$  defined by

$$\Omega = \{\mathbf{x} \in \mathbb{R}^2 : \gamma_1 < \|\mathbf{x}\| < \gamma_2\}. \quad (3.6)$$

The boundary is  $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ ,  $\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$ , where  $\partial\Omega_1 = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| = \gamma_1\}$  and  $\partial\Omega_2 = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| = \gamma_2\}$ . The outward unit normal vector to the boundary is denoted by  $\mathbf{n} = (n_x, n_y)$ .

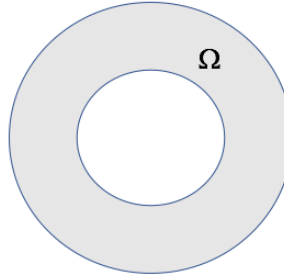


Figure 3.1: Annulus Domain

### 3.1.2 Distribution of Collocation Points

The collocation points  $\{\mathbf{x}_{mn}\}_{m=1,n=1}^{M,N}$  will be placed on concentric circles in the domain  $\Omega$ . First, define the  $M$  angles on each concentric circle

$$\theta_m = \frac{2\pi(m-1)}{M}, \quad m = 1, \dots, M, \quad (3.7)$$

and the radii for the  $N$  concentric circles

$$r_n = \gamma_1 + (\gamma_2 - \gamma_1) \frac{n-1}{N-1}, \quad n = 1, \dots, N. \quad (3.8)$$

For the two-dimensional case,  $\{\mathbf{x}_{mn}\}_{m=1,n=1}^{M,N} = \{(x_{mn}, y_{mn})\}_{m=1,n=1}^{M,N}$  are then defined as follows:

$$x_{mn} = r_n \cos\left(\theta_m + \frac{2\pi\alpha_n}{M}\right), \quad y_{mn} = r_n \sin\left(\theta_m + \frac{2\pi\alpha_n}{M}\right), \quad m = 1, \dots, M, \quad n = 1, \dots, N. \quad (3.9)$$

The parameters  $\{\alpha_n\}_{n=1}^N \in [-1/2, 1/2]$  correspond to rotations of the collocation points and may be used to produce more uniform distributions. Typical distributions of collocation points without rotation ( $\alpha_n = 0, n = 1, \dots, n$ ) and with rotation are given in Figure 3.2.



*Figure 3.2:* Discretization of the annular domain with (a) no rotation of the collocation points and (b) with rotation of the collocation points. The crosses (+) denote the collocation points.

With the distribution of collocation points, the RBF interpolation matrix can then be constructed. First define  $\varphi_{im}(r_{jn}) = \varphi(\|\mathbf{x}_{jn} - \mathbf{x}_{im}\|)$ , where  $i, j$  are the concentric circle index and  $m, n$  are the collocation point index on each concentric circle. Then the interpolation matrix of each  $i^{th}, j^{th}$  concentric circle is

$$A_{ij} = \begin{pmatrix} \varphi_{i1}(r_{j1}) & \varphi_{i1}(r_{j2}) & \cdots & \varphi_{i1}(r_{jM}) \\ \varphi_{i2}(r_{j1}) & \varphi_{i2}(r_{j2}) & \cdots & \varphi_{i2}(r_{jM}) \\ \vdots & \vdots & & \vdots \\ \varphi_{iM}(r_{j1}) & \varphi_{iM}(r_{j2}) & \cdots & \varphi_{iM}(r_{jM}) \end{pmatrix}, \quad i, j = 1, \dots, N. \quad (3.10)$$

To study the effect the distribution of points will have on the interpolation matrix,

consider the following example for  $A_{12}$  where  $M = 4, N = 4$ :

$$A_{12} = \begin{pmatrix} \varphi_{11}(r_{21}) & \varphi_{11}(r_{22}) & \varphi_{11}(r_{23}) & \varphi_{11}(r_{24}) \\ \varphi_{12}(r_{21}) & \varphi_{12}(r_{22}) & \varphi_{12}(r_{23}) & \varphi_{12}(r_{24}) \\ \varphi_{13}(r_{21}) & \varphi_{13}(r_{22}) & \varphi_{13}(r_{23}) & \varphi_{13}(r_{24}) \\ \varphi_{14}(r_{21}) & \varphi_{14}(r_{22}) & \varphi_{14}(r_{23}) & \varphi_{14}(r_{24}) \end{pmatrix}. \quad (3.11)$$

The distribution of the collocation points are shown in Figure 3.1.2. By examining the

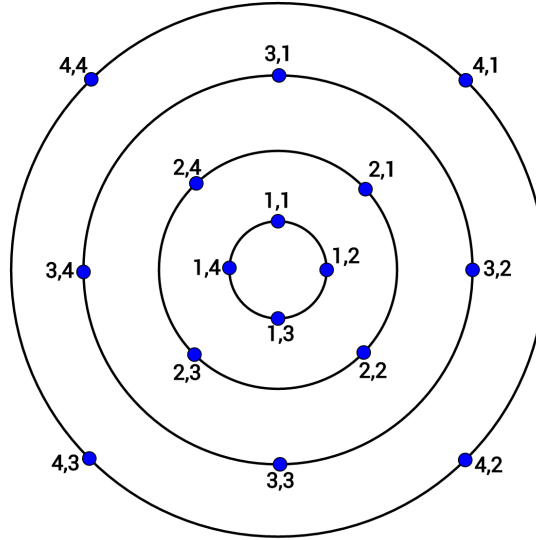


Figure 3.3: Distribution of collocation points on domain

geometry of this distribution, it can be observed that many of the distances between each point are the same. This allows  $A_{12}$  to be rewritten as

$$A_{12} = \begin{pmatrix} \varphi_{11}(r_{21}) & \varphi_{11}(r_{22}) & \varphi_{11}(r_{23}) & \varphi_{11}(r_{24}) \\ \varphi_{11}(r_{24}) & \varphi_{11}(r_{21}) & \varphi_{11}(r_{22}) & \varphi_{11}(r_{23}) \\ \varphi_{11}(r_{23}) & \varphi_{11}(r_{24}) & \varphi_{11}(r_{21}) & \varphi_{11}(r_{22}) \\ \varphi_{11}(r_{22}) & \varphi_{11}(r_{23}) & \varphi_{11}(r_{24}) & \varphi_{11}(r_{21}) \end{pmatrix}. \quad (3.12)$$

Notice that every row is a right cyclic shift of the previous row. This same structure is common for all  $A_{ij}$ , which is a special type of matrix known as a circulant matrix. The interpolation matrix now consists of circulant submatrices

$$G = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{pmatrix}. \quad (3.13)$$

In the next section, the circulant matrix will be discussed in further detail and its importance in the matrix decomposition algorithm.

### 3.2 Circulant Matrices

A circulant matrix occurs when every row of the matrix is a right cyclic shift of the previous row:

$$C_M = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_M \\ c_M & c_1 & c_2 & \dots & c_{M-1} \\ c_{M-1} & c_M & c_1 & \dots & c_{M-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_2 & c_3 & c_4 & \dots & c_1 \end{bmatrix} \quad (3.14)$$

Circulant matrices and their properties have been well documented in [13]. One advantage of circulant matrices is that they can be completely described by the first row of the matrix due to its cyclic permutations of the row. By strategically choosing the collocation points so that they lie on concentric circles, the collocation matrix will then consist of circulant submatrices. These circulant matrices come with special inherent properties that can be utilized in simplifying the process of the numerical solution to the partial differential equation.

First, a circulant matrix can be diagonalized by a discrete Fourier transform. Let  $\omega = e^{2\pi i/M}$ ,  $M \geq 1$ ,  $i^2 = -1$ . Note that  $\omega$  is a primitive  $M^{\text{th}}$  root of unity, meaning  $\omega^M = 1$ . We can then construct the Fourier matrix of order  $M$ :

$$U_M = \frac{1}{\sqrt{M}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{M-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(M-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega^{M-1} & \omega^{2(M-1)} & \dots & \omega^{(M-1)(M-1)} \end{pmatrix}. \quad (3.15)$$

**Theorem 1.** The Discrete Fourier matrix,  $U_M$ , is unitary, i.e.  $U_M U_M^* = I_M$ , where  $U_M^*$  is the conjugate transpose of  $U_M$ .

**Proof.** Consider an element  $u_{ij}$  of  $U_M U_M^*$ ,

$$u_{ij} = \sum_{k=0}^{M-1} \frac{1}{\sqrt{M}} \omega^{ki} \frac{1}{\sqrt{M}} \omega^{-kj} = \frac{1}{M} \sum_{k=0}^{M-1} \omega^{k(i-j)} = \frac{1}{M} \left( \frac{1 - \omega^{M(i-j)}}{1 - \omega^{i-j}} \right). \quad (3.16)$$

For  $i = j$ ,

$$u_{ij} = \frac{1}{M} \left( 1 + \omega^{(i-j)} + \omega^{2(i-j)} + \dots + \omega^{(M-1)(i-j)} \right) = 1 \quad (3.17)$$

For  $i \neq j$ ,

$$u_{ij} = \frac{1}{M} \left( \frac{1 - \omega^{M(i-j)}}{1 - \omega^{i-j}} \right) = 0 \quad (3.18)$$

Therefore,  $U_M U_M^* = I_M$ .  $\square$

The eigenvalues and eigenvectors of a circulant matrix can be found by the following equation

$$\begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_M \\ c_M & c_1 & c_2 & \dots & c_{M-1} \\ c_{M-1} & c_M & c_1 & \dots & c_{M-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_2 & c_3 & c_4 & \dots & c_1 \end{bmatrix} \begin{bmatrix} 1 \\ \omega^i \\ \omega^{2i} \\ \vdots \\ \omega^{(M-1)i} \end{bmatrix} = \lambda_i \begin{bmatrix} 1 \\ \omega^i \\ \omega^{2i} \\ \vdots \\ \omega^{(M-1)i} \end{bmatrix}, \quad (3.19)$$

where  $i = 0, \dots, M-1$ .

**Definition 3.2.1.**  $\lambda_i = c_1 + c_2 \omega^i + \dots + c_M \omega^{(M-1)i}$  is an eigenvalue of  $C_M$ .

$x_i = [1, \omega^i, \omega^{2i}, \dots, \omega^{(M-1)i}]$  is an eigenvector of  $C_M$ .

Note that the columns of  $\{x_i\}_{i=0}^{M-1}$  of the discrete Fourier matrix,  $U_M$ , are the eigenvectors of  $C_M$  and are independent of the entries  $c_1, \dots, c_M$ . Hence,  $U_M$  can now be used to diagonalize  $C_M$  along with  $\Lambda$ , which is the diagonal matrix consisting of the eigenvalues  $\{\lambda_i\}_{i=0}^{M-1}$  as the diagonal entries

$$C_M = U_M^* \Lambda U_M. \quad (3.20)$$

Since  $U_M$  is unitary,  $\Lambda$  can be written as

$$\Lambda = U_M C_M U_M^*. \quad (3.21)$$

In MATLAB, the diagonalization process can be achieved using the functions *fft* (fast Fourier Transform) and *ifft* (inverse fast Fourier Transform). The fast Fourier Transform is a very efficient algorithm that is used to compute the discrete Fourier Transform by factorizing the DFT matrix into a product of sparse factors. Using these commands not only adds savings in computational cost, but it also provides considerable savings in storage since the eigenvalues can be found using only the first row of the matrix  $[c_1, c_2, \dots, c_n]$ .

To apply the diagonalization to each submatrix in the interpolation matrix (3.13), the tensor product can be applied:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \dots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix}, \quad (3.22)$$

where  $A$  and  $B$  are  $m \times n$  and  $p \times q$  matrices, respectively. The resultant matrix will be  $mp \times nq$ . By multiplying  $G$ , consisting of circulant submatrices, by  $I_N \otimes U_M$  on the left hand side and  $I_N \otimes U_M^*$  on the right hand side, each circulant block will be diagonalized

$$\begin{aligned}
\Lambda &= (I_N \otimes U_M)G(I_N \otimes U_M^*) \\
&= \begin{pmatrix} U_M & 0 & \cdots & 0 \\ 0 & U_M & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & U_M \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MN} \end{pmatrix} \begin{pmatrix} U_M^* & 0 & \cdots & 0 \\ 0 & U_M^* & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & U_M^* \end{pmatrix} \\
&= \begin{pmatrix} \Lambda_{11} & \Lambda_{12} & \cdots & \Lambda_{1N} \\ \Lambda_{21} & \Lambda_{22} & \cdots & \Lambda_{2N} \\ \vdots & \vdots & & \vdots \\ \Lambda_{M1} & \Lambda_{M2} & \cdots & \Lambda_{MN} \end{pmatrix}
\end{aligned} \tag{3.23}$$

$\Lambda$  is a sparse matrix consisting of diagonal submatrices.

### 3.3 Properties of Circulant Matrices

The following theorems will present additional properties of circulant matrices that are necessary for the implementation of the matrix decomposition algorithm in the RBF-DQ method.

**Theorem 2.** *Consider the system*

$$BA = C, \tag{3.24}$$

where the  $MN \times MN$  matrices  $B$  and  $C$  are block circulant, each consisting of  $N^2$  circulant submatrices  $B_{n_1, n_2}, C_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , respectively, each of order  $M$ . Assume that the matrix  $B$  is nonsingular. Then the  $MN \times MN$  matrix  $A$  will also be block circulant consisting of  $N^2$  circulant submatrices  $A_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ .

**Proof.** It is first necessary to prove that if an  $MN \times MN$  matrix  $B$  is block circulant then so is its inverse. Since  $B$  is block circulant,  $B$  can be rewritten as

$$B = (I_N \otimes U_M^*)D(I_N \otimes U_M), \tag{3.25}$$

where the matrix  $D$  is block diagonal and nonsingular, consisting of  $N^2$  diagonal submatrices  $D_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ . The inverse of the matrix  $B$  will be

$$B^{-1} = (I_N \otimes U_M^*)D^{-1}(I_N \otimes U_M). \tag{3.26}$$

Next, it must be proven that since  $D$  is block diagonal, so is  $D^{-1}$ . Let  $D^{-1}$  consist of  $N^2$  submatrices  $\hat{D}_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ . Then

$$\begin{pmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,N} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,N} \\ \vdots & \vdots & & \vdots \\ D_{N,1} & D_{N,2} & \cdots & D_{N,N} \end{pmatrix} \begin{pmatrix} \hat{D}_{1,1} & \hat{D}_{1,2} & \cdots & \hat{D}_{1,N} \\ \hat{D}_{2,1} & \hat{D}_{2,2} & \cdots & \hat{D}_{2,N} \\ \vdots & \vdots & & \vdots \\ \hat{D}_{N,1} & \hat{D}_{N,2} & \cdots & \hat{D}_{N,N} \end{pmatrix} = \begin{pmatrix} I_M & 0 & \cdots & 0 \\ 0 & I_M & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & I_M \end{pmatrix}, \quad (3.27)$$

where each diagonal submatrix  $D_{n_1, n_2} = \text{diag}(d_1^{n_1, n_2}, d_2^{n_1, n_2}, \dots, d_M^{n_1, n_2})$ . Assume that each submatrix  $\hat{D}_{n_1, n_2} = (\hat{d}_{m_1, m_2}^{n_1, n_2})_{m_1, m_2=1}^M$ , is full.

Consider the system created by multiplying  $D$  by the first column of  $D^{-1}$

$$\begin{pmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,N} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,N} \\ \vdots & \vdots & & \vdots \\ D_{N,1} & D_{N,2} & \cdots & D_{N,N} \end{pmatrix} \begin{pmatrix} \hat{D}_{1,1} \\ \hat{D}_{2,1} \\ \vdots \\ \hat{D}_{N,1} \end{pmatrix} = \begin{pmatrix} I_M \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (3.28)$$

This system can be broken to  $M$  independent systems of order  $N$

$$\begin{pmatrix} d_1^{1,1} & d_1^{1,2} & \cdots & d_1^{1,N} \\ d_1^{2,1} & d_1^{2,2} & \cdots & d_1^{2,N} \\ \vdots & \vdots & & \vdots \\ d_1^{N,1} & d_1^{N,2} & \cdots & d_1^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{1,1}^{1,1} & \hat{d}_{1,2}^{1,1} & \cdots & \hat{d}_{1,M}^{1,1} \\ \hat{d}_{1,1}^{2,1} & \hat{d}_{1,2}^{2,1} & \cdots & \hat{d}_{1,M}^{2,1} \\ \vdots & \vdots & & \vdots \\ \hat{d}_{1,1}^{N,1} & \hat{d}_{1,2}^{N,1} & \cdots & \hat{d}_{1,M}^{N,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$\begin{pmatrix} d_2^{1,1} & d_2^{1,2} & \cdots & d_2^{1,N} \\ d_2^{2,1} & d_2^{2,2} & \cdots & d_2^{2,N} \\ \vdots & \vdots & & \vdots \\ d_2^{N,1} & d_2^{N,2} & \cdots & d_2^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{2,1}^{1,1} & \hat{d}_{2,2}^{1,1} & \cdots & \hat{d}_{2,M}^{1,1} \\ \hat{d}_{2,1}^{2,1} & \hat{d}_{2,2}^{2,1} & \cdots & \hat{d}_{2,M}^{2,1} \\ \vdots & \vdots & & \vdots \\ \hat{d}_{2,1}^{N,1} & \hat{d}_{2,2}^{N,1} & \cdots & \hat{d}_{2,M}^{N,1} \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$

up to

$$\begin{pmatrix} d_M^{1,1} & d_M^{1,2} & \cdots & d_M^{1,N} \\ d_M^{2,1} & d_M^{2,2} & \cdots & d_M^{2,N} \\ \vdots & \vdots & & \vdots \\ d_M^{N,1} & d_M^{N,2} & \cdots & d_M^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{M,1}^{1,1} & \hat{d}_{M,2}^{1,1} & \cdots & \hat{d}_{M,M}^{1,1} \\ \hat{d}_{M,1}^{2,1} & \hat{d}_{M,2}^{2,1} & \cdots & \hat{d}_{M,M}^{2,1} \\ \vdots & \vdots & & \vdots \\ \hat{d}_{M,1}^{N,1} & \hat{d}_{M,2}^{N,1} & \cdots & \hat{d}_{M,M}^{N,1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$

Similarly, the system

$$\begin{pmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,N} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,N} \\ \vdots & \vdots & & \vdots \\ D_{N,1} & D_{N,2} & \cdots & D_{N,N} \end{pmatrix} \begin{pmatrix} \hat{D}_{1,2} \\ \hat{D}_{2,2} \\ \vdots \\ \hat{D}_{N,2} \end{pmatrix} = \begin{pmatrix} 0 \\ I_M \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (3.29)$$

is equivalent to

$$\begin{pmatrix} d_1^{1,1} & d_1^{1,2} & \cdots & d_1^{1,N} \\ d_1^{2,1} & d_1^{2,2} & \cdots & d_1^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_1^{N,1} & d_1^{N,2} & \cdots & d_1^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{1,1}^{1,2} & \hat{d}_{1,2}^{1,2} & \cdots & \hat{d}_{1,M}^{1,2} \\ \hat{d}_{1,1}^{2,2} & \hat{d}_{1,2}^{2,2} & \cdots & \hat{d}_{1,M}^{2,2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{1,1}^{N,2} & \hat{d}_{1,2}^{N,2} & \cdots & \hat{d}_{1,M}^{N,2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$\begin{pmatrix} d_2^{1,1} & d_2^{1,2} & \cdots & d_2^{1,N} \\ d_2^{2,1} & d_2^{2,2} & \cdots & d_2^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_2^{N,1} & d_2^{N,2} & \cdots & d_2^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{2,1}^{1,2} & \hat{d}_{2,2}^{1,2} & \cdots & \hat{d}_{2,M}^{1,2} \\ \hat{d}_{2,1}^{2,2} & \hat{d}_{2,2}^{2,2} & \cdots & \hat{d}_{2,M}^{2,2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{2,1}^{N,2} & \hat{d}_{2,2}^{N,2} & \cdots & \hat{d}_{2,M}^{N,2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$

up to

$$\begin{pmatrix} d_M^{1,1} & d_M^{1,2} & \cdots & d_M^{1,N} \\ d_M^{2,1} & d_M^{2,2} & \cdots & d_M^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_M^{N,1} & d_M^{N,2} & \cdots & d_M^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{M,1}^{1,2} & \hat{d}_{M,2}^{1,2} & \cdots & \hat{d}_{M,M}^{1,2} \\ \hat{d}_{M,1}^{2,2} & \hat{d}_{M,2}^{2,2} & \cdots & \hat{d}_{M,M}^{2,2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{M,1}^{N,2} & \hat{d}_{M,2}^{N,2} & \cdots & \hat{d}_{M,M}^{N,2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Continuing this process for every block column in system (3.27) until, finally, the system

$$\begin{pmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,N} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ D_{N,1} & D_{N,2} & \cdots & D_{N,N} \end{pmatrix} \begin{pmatrix} \hat{D}_{1,N} \\ \hat{D}_{2,N} \\ \vdots \\ \hat{D}_{N,N} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I_M \end{pmatrix}, \quad (3.30)$$

is equivalent to

$$\begin{pmatrix} d_1^{1,1} & d_1^{1,2} & \cdots & d_1^{1,N} \\ d_1^{2,1} & d_1^{2,2} & \cdots & d_1^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_1^{N,1} & d_1^{N,2} & \cdots & d_1^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{1,1}^{1,N} & \hat{d}_{1,2}^{1,N} & \cdots & \hat{d}_{1,M}^{1,N} \\ \hat{d}_{1,1}^{2,N} & \hat{d}_{1,2}^{2,N} & \cdots & \hat{d}_{1,M}^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{1,1}^{N,N} & \hat{d}_{1,2}^{N,N} & \cdots & \hat{d}_{1,M}^{N,N} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix},$$

$$\begin{pmatrix} d_2^{1,1} & d_2^{1,2} & \cdots & d_2^{1,N} \\ d_2^{2,1} & d_2^{2,2} & \cdots & d_2^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_2^{N,1} & d_2^{N,2} & \cdots & d_2^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{2,1}^{1,N} & \hat{d}_{2,2}^{1,N} & \cdots & \hat{d}_{2,M}^{1,N} \\ \hat{d}_{2,1}^{2,N} & \hat{d}_{2,2}^{2,N} & \cdots & \hat{d}_{2,M}^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{2,1}^{N,N} & \hat{d}_{2,2}^{N,N} & \cdots & \hat{d}_{2,M}^{N,N} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \end{pmatrix},$$

up to

$$\begin{pmatrix} d_M^{1,1} & d_M^{1,2} & \cdots & d_M^{1,N} \\ d_M^{2,1} & d_M^{2,2} & \cdots & d_M^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_M^{N,1} & d_M^{N,2} & \cdots & d_M^{N,N} \end{pmatrix} \begin{pmatrix} \hat{d}_{M,1}^{1,N} & \hat{d}_{M,2}^{1,N} & \cdots & \hat{d}_{M,M}^{1,N} \\ \hat{d}_{M,1}^{2,N} & \hat{d}_{M,2}^{2,N} & \cdots & \hat{d}_{M,M}^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{M,1}^{N,N} & \hat{d}_{M,2}^{N,N} & \cdots & \hat{d}_{M,M}^{N,N} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$



The solution of system (3.27) is equivalent to solving the systems

$$E_m F_{mn} = \tilde{I}_{mn}, \quad m = 1, \dots, M, n = 1, \dots, N,$$

where

$$E_m = \begin{pmatrix} d_m^{1,1} & d_m^{1,2} & \cdots & d_m^{1,N} \\ d_m^{2,1} & d_m^{2,2} & \cdots & d_m^{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_m^{N,1} & d_m^{N,2} & \cdots & d_m^{N,N} \end{pmatrix}, m = 1, \dots, M,$$

$$(F_{mn})_{i,j} = \left( \tilde{d}_{m,j}^{i,n} \right), \quad i, n = 1, \dots, N, j, m = 1, \dots, M,$$

and  $\tilde{I}_{mn}, m = 1, \dots, M, n = 1, \dots, N$ , are zero matrices with 1 at the position  $(n, m)$ . This can be written more compactly as  $M$  systems with  $MN$  right hand sides,

$$E_m (F_{m1} | F_{m2} | \dots | F_{mN}) = (\tilde{I}_{m1} | \tilde{I}_{m2} | \dots | \tilde{I}_{mN}), \quad m = 1, \dots, M. \quad (3.31)$$

Since the matrix  $D^{-1}$  is nonsingular, each matrix  $E_m$  is nonsingular, and from (3.31), for each  $n = 1, \dots, N$ , the only nonzero column in matrix  $F_{mn}$  is column  $m$ . This means that in each submatrix  $\hat{D}_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , the only nonzero elements are the elements  $(\hat{d}_{m,m}^{n_1, n_2})_{m=1}^M$ , which implies that each submatrix  $\hat{D}_{n_1, n_2}$  is diagonal, and hence  $D^{-1}$  is block diagonal. Therefore,

$$B^{-1} = (I_N \otimes U_M^*) D^{-1} (I_N \otimes U_M) = \begin{pmatrix} U_M^* \hat{D}_{1,1} U_M & U_M^* \hat{D}_{1,2} U_M & \cdots & U_M^* \hat{D}_{1,N} U_M \\ U_M^* \hat{D}_{2,1} U_M & U_M^* \hat{D}_{2,2} U_M & \cdots & U_M^* \hat{D}_{2,N} U_M \\ \vdots & \vdots & \ddots & \vdots \\ U_M^* \hat{D}_{N,1} U_M & U_M^* \hat{D}_{N,2} U_M & \cdots & U_M^* \hat{D}_{N,N} U_M \end{pmatrix} \quad (3.32)$$

and from [13, Theorem 3.2.3] each of the matrices  $U_M^* \hat{D}_{n_1, n_2} U_M, n_1, n_2 = 1, \dots, N$ , is circulant. Therefore  $B^{-1}$  is block circulant.

From (3.24),

$$A = B^{-1} C, \quad (3.33)$$

where  $B^{-1}$  and  $C$  are block circulant, each consisting of  $N^2$  circulant matrices of order  $M$ . Since from [13, Theorem 3.2.4] the sum and the product of circulant matrices is circulant, it easily follows that the matrix  $A$  is block circulant, consisting of  $N^2$  circulant matrices of order  $M$ .  $\square$

**Corollary 1.** Consider the system

$$BA = C, \quad (3.34)$$

where the  $MN \times MN$  matrix  $B$  is block circulant, consisting of  $N^2$  circulant submatrices  $B_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ , and the  $MN \times M$  matrix  $C$  is block circulant, consisting of  $N$  circulant submatrices  $C_n, n = 1, \dots, N$ , each of order  $M$ . Assume that the matrix  $B$  is nonsingular. Then the  $MN \times N$  matrix  $A$  will also be block circulant consisting of  $N$  circulant submatrices  $A_n, n = 1, \dots, N$ , each of order  $M$ .

**Proof.** From Theorem 2 it follows that the inverse of matrix  $B$  in (3.34) is block circulant, consisting of  $N^2$  circulant matrices of order  $M$ . Since  $C$  is also block circulant, consisting of  $N$  circulant submatrices  $C_n, n = 1, \dots, N$ , each of order  $M$ , since the sum and the product of circulant matrices is circulant, it follows that the matrix  $A$  is block circulant, consisting of  $N$  circulant matrices of order  $M$ .

The following theorem and corollary will be necessary for solving the Cauchy Navier elasticity problem.  $\square$

**Theorem 3.** Consider the system

$$\left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right) \left( \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) = \left( \begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array} \right), \quad (3.35)$$

where the  $MN \times MN$  matrices  $B_{ij}, C_{ij}, i, j = 1, 2$ , are block circulant, each consisting of  $N^2$  circulant submatrices  $B_{n_1, n_2}^{ij}, C_{n_1, n_2}^{ij}, n_1, n_2 = 1, \dots, N, i, j = 1, 2$ , respectively, each of order  $M$ . Assume that the matrices  $B_{ij}, i, j = 1, 2$ , are nonsingular. Then the  $MN \times MN$  matrices  $A_{ij}, i, j = 1, 2$ , will also be block circulant consisting of  $N^2$  circulant submatrices  $A_{n_1, n_2}^{ij}, n_1, n_2 = 1, \dots, N, i, j = 1, 2$ , each of order  $M$ .

**Proof.** It is first necessary to prove that if the inverse of the coefficient matrix in system (3.35) is

$$\left( \begin{array}{c|c} \hat{B}_{11} & \hat{B}_{12} \\ \hline \hat{B}_{21} & \hat{B}_{22} \end{array} \right) = \left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right)^{-1},$$

then each of the matrices  $\hat{B}_{ij}, i, j = 1, 2$ , is block circulant consisting of  $N^2$  circulant submatrices of order  $M$ .

From the properties of circulant matrices it follows that

$$\left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right) = (I_2 \otimes I_N \otimes U_M^*) \left( \begin{array}{c|c} D_{11} & D_{12} \\ \hline D_{21} & D_{22} \end{array} \right) (I_2 \otimes I_N \otimes U_M), \quad (3.36)$$

where each of the matrices  $D_{ij}, i, j = 1, 2$ , is block diagonal and nonsingular, consisting of  $N^2$  diagonal submatrices each of order  $M$ . The inverse of

$$\left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right)$$

can be written as

$$\left( \begin{array}{c|c} \hat{B}_{11} & \hat{B}_{12} \\ \hat{B}_{21} & \hat{B}_{22} \end{array} \right) = (I_2 \otimes I_N \otimes U_M^*) \left( \begin{array}{c|c} D_{11} & D_{12} \\ D_{21} & D_{22} \end{array} \right)^{-1} (I_2 \otimes I_N \otimes U_M). \quad (3.37)$$

Next, it must be proven that if

$$\left( \begin{array}{c|c} \hat{D}_{11} & \hat{D}_{12} \\ \hat{D}_{21} & \hat{D}_{22} \end{array} \right) = \left( \begin{array}{c|c} D_{11} & D_{12} \\ D_{21} & D_{22} \end{array} \right)^{-1},$$

then each of the matrices  $\hat{D}_{i,j}, i, j = 1, 2$ , will consist of  $N^2$  diagonal submatrices each of order  $M$ . It is true that

$$\left( \begin{array}{c|c} D_{11} & D_{12} \\ D_{21} & D_{22} \end{array} \right) \left( \begin{array}{c|c} \hat{D}_{11} & \hat{D}_{12} \\ \hat{D}_{21} & \hat{D}_{22} \end{array} \right) = \left( \begin{array}{c|c} I_{MN} & 0 \\ 0 & I_{MN} \end{array} \right), \quad (3.38)$$

where  $I_{MN}$  is the  $MN \times MN$  identity matrix. System (3.38) is equivalent to

$$\begin{aligned} D_{11}\hat{D}_{11} + D_{12}\hat{D}_{21} &= I_{MN} \\ D_{11}\hat{D}_{12} + D_{12}\hat{D}_{22} &= 0 \\ D_{21}\hat{D}_{11} + D_{22}\hat{D}_{21} &= 0 \\ D_{21}\hat{D}_{12} + D_{22}\hat{D}_{22} &= I_{MN}. \end{aligned} \quad (3.39)$$

Combining the first and third equations in (3.39) yields

$$[-D_{11}D_{21}^{-1}D_{22} + D_{12}]\hat{D}_{21} = I_{MN}. \quad (3.40)$$

Since the matrix  $D_{21}$  is block diagonal consisting of  $N^2$  diagonal submatrices each of order  $M$ , from Theorem 2, so will  $D_{21}^{-1}$ . It follows that the matrix  $[-D_{11}D_{21}^{-1}D_{22} + D_{12}]$  will be block diagonal consisting of  $N^2$  diagonal submatrices each of order  $M$ . Therefore, from the results of Theorem 2, matrix  $\hat{D}_{21}$  will also be block diagonal consisting of  $N^2$  diagonal submatrices each of order  $M$ . Since  $\hat{D}_{11} = -D_{21}^{-1}D_{22}\hat{D}_{21}$ , it will also be block diagonal consisting of  $N^2$  diagonal submatrices each of order  $M$ . A similar argument combining the second and fourth equations (3.39) yields that both  $\hat{D}_{12}$  and  $\hat{D}_{22}$  will also be block diagonal consisting of  $N^2$  diagonal submatrices each of order  $M$ .

From (3.37) and [13, Theorem 3.2.3] it follows that each of the matrices  $\hat{B}_{ij}, i, j = 1, 2$ , is block circulant consisting of  $N^2$  circulant submatrices of order  $M$ . Now from (3.35) it follows that

$$\left( \begin{array}{c|c} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array} \right) = \left( \begin{array}{c|c} \hat{B}_{11} & \hat{B}_{12} \\ \hat{B}_{21} & \hat{B}_{22} \end{array} \right) \left( \begin{array}{c|c} C_{11} & C_{12} \\ C_{21} & C_{22} \end{array} \right), \quad (3.41)$$

and since each of the matrices  $\hat{B}_{ij}, C_{ij}, i, j = 1, 2$ , is block circulant, so will the matrices  $A_{ij}, i, j = 1, 2$ .  $\square$

**Corollary 2.** *Consider the system*

$$\left( \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) \left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right) = \left( \begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array} \right), \quad (3.42)$$

where the  $MN \times MN$  matrices  $B_{ij}, i, j = 1, 2$ , are block circulant, each consisting of  $N^2$  circulant submatrices  $B_{n_1, n_2}^{ij}, n_1, n_2 = 1, \dots, N, i, j = 1, 2$ , each of order  $M$ , and now the  $MN \times M$  matrices  $C_{ij}, i, j = 1, 2$ , are block circulant, each consisting of  $N$  circulant submatrices  $C_n^{ij}, n = 1, \dots, N, i, j = 1, 2$ , each of order  $M$ . Assume that the  $MN \times MN$  matrices  $B_{ij}, i, j = 1, 2$  are nonsingular. Then the  $MN \times M$  matrices  $A_{ij}, i, j = 1, 2$ , are also block circulant, each consisting of  $N$  circulant submatrices  $A_n^{ij}, n = 1, \dots, N, i, j = 1, 2$ , each of order  $M$ .

**Proof.** From (3.42) it follows that

$$\left( \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) = \left( \begin{array}{c|c} \hat{B}_{11} & \hat{B}_{12} \\ \hline \hat{B}_{21} & \hat{B}_{22} \end{array} \right) \left( \begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array} \right)$$

where, from Theorem 2, each of the matrices  $\hat{B}_{ij}, i, j = 1, 2$ , is block circulant consisting of  $N^2$  circulant submatrices of order  $M$ . Since each of the matrices  $C_{ij}, i, j = 1, 2$ , is also block circulant consisting of  $N$  circulant submatrices of order  $M$ , and since the sum and the product of circulant matrices is circulant, it follows that the matrices  $A_{ij}, i, j = 1, 2$ , are block circulant, each consisting of  $N$  circulant matrices of order  $M$ .  $\square$

## Chapter 4

### Global and Local RBF-DQ MDA

#### 4.1 RBF-DQ MDA

##### 4.1.1 Poisson

Consider the following PDE

$$\Delta u = f(x,y) \quad \text{in } \Omega, \quad (4.1a)$$

subject to either the Dirichlet boundary conditions

$$u = g_1(x,y) \quad \text{on } \partial\Omega_1, \quad (4.1b)$$

$$u = g_2(x,y) \quad \text{on } \partial\Omega_2, \quad (4.1c)$$

or the mixed Neumann/Dirichlet boundary conditions

$$\frac{\partial u}{\partial \mathbf{n}} = g_1(x,y) \quad \text{on } \partial\Omega_1, \quad (4.1d)$$

$$u = g_2(x,y) \quad \text{on } \partial\Omega_2. \quad (4.1e)$$

In (4.1d),  $\partial/\partial n$  denotes the derivative along the outward unit normal vector to the boundary. Problem (4.1a), (4.1b-4.1c) is a Dirichlet boundary value problem, and problem (4.1a), (4.1d-4.1e) is a mixed Neumann/Dirichlet boundary value problem, where  $u = u(x,y)$  is the dependent variable to be solved, and  $f, g_1, g_2$  are given functions.

In the RBF-DQ method, for each point  $(x_{ij}, y_{ij})$ ,  $i = 1, \dots, M$ ,  $j = 1, \dots, N$ , and for each RBF  $\phi_{mn}$ ,  $m = 1, \dots, M$ ,  $n = 1, \dots, N$ ,

$$\Delta \phi_{mn}(x_{ij}, y_{ij}) = \sum_{k=1}^M \sum_{\ell=1}^N a_{k\ell}^{ij} \phi_{mn}(x_{k\ell}, y_{k\ell}),$$

which in vector form gives, for each point  $(x_{ij}, y_{ij})$ ,

$$\phi_{\Delta}^{ij} = G a^{ij}, \quad (4.2)$$

where

$$G = \begin{pmatrix} \varphi_{11}(r_{11}) & \varphi_{11}(r_{12}) & \varphi_{11}(r_{13}) & \cdots & \varphi_{11}(r_{MN}) \\ \varphi_{12}(r_{11}) & \varphi_{12}(r_{12}) & \varphi_{12}(r_{13}) & \cdots & \varphi_{12}(r_{MN}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi_{MN}(r_{11}) & \varphi_{MN}(r_{12}) & \varphi_{MN}(r_{13}) & \cdots & \varphi_{MN}(r_{MN}) \end{pmatrix}, \quad (4.3)$$

$$\varphi_{\Delta}^{ij} = [\Delta\varphi(r_{11}^{ij}) \Delta\varphi(r_{12}^{ij}) \cdots \Delta\varphi(r_{MN}^{ij})]^T, \text{ and } a^{ij} = [a_{11}^{ij} a_{12}^{ij} \cdots a_{MN}^{ij}]^T.$$

Due to the circular distribution of the collocation points, the  $(MN \times MN)$  matrix  $G$  consists of  $N^2$  circulant submatrices  $A_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ , and is therefore block circulant. Moreover, the  $MN \times MN$  matrix  $\Phi = [\varphi_{\Delta}^{11} | \varphi_{\Delta}^{12} | \varphi_{\Delta}^{13} | \dots | \varphi_{\Delta}^{MN}]$  also consists of  $N^2$  circulant [13] submatrices  $\Phi_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ , and is therefore also block circulant. It follows that, by Theorem 2 in Chapter 3, the  $MN \times MN$  matrix  $A = [a^{11} | a^{12} | \dots | a^{MN}]$  will also be block circulant, consisting of  $N^2$  circulant submatrices  $A_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ .

Equation (4.2) can be written more compactly as

$$GA = \Phi. \quad (4.4)$$

Once  $A = [a^{11} | a^{12} | \dots | a^{MN}]$  is computed, for each point  $(x_{ij}, y_{ij})$ ,

$$\Delta u(x_{ij}, y_{ij}) \approx \sum_{k=1}^M \sum_{l=1}^N a_{kl}^{ij} u_{kl}, \quad (4.5)$$

where the  $u_{kl}$  are approximations of  $u(x_{kl}, y_{kl})$ . Substitution in the differential equation yields

$$\sum_{k=1}^M \sum_{l=1}^N a_{kl}^{ij} u_{kl} = f(x_{ij}, y_{ij}), i = 1, \dots, M, j = 2, \dots, N-1. \quad (4.6)$$

In the case of the Dirichlet boundary condition,

$$u_{i1} = g_1(x_{i1}, y_{i1}) \text{ and } u_{iN} = g_2(x_{iN}, y_{iN}), i = 1, \dots, M. \quad (4.7)$$

In the case of the Neumann/Dirichlet boundary condition, for each point  $(x_{i1}, y_{i1}), i = 1, \dots, M$ , on the boundary  $\partial\Omega_1$ , and for each RBF  $\varphi_{mn}, m = 1, \dots, M, n = 1, \dots, N$ ,

$$\frac{\partial \varphi_{mn}}{\partial n}(x_{i1}, y_{i1}) = \sum_{k=1}^M \sum_{l=1}^N b_{kl}^{i1} \varphi_{mn}(x_{kl}, y_{kl}), \quad (4.8)$$

which in vector form gives, for each boundary point  $(x_{i1}, y_{i1})$  on  $\partial\Omega_1$ ,

$$\varphi_n^{i1} = Gb^{i1}. \quad (4.9)$$

The  $MN \times M$  matrix  $\Psi = [\varphi_n^{11} | \varphi_n^{21} | \dots | \varphi_n^{M1}]$  now consists of  $N$  circulant submatrices  $\Phi_n, n = 1, \dots, N$ , each of order  $M$ , and is therefore block circulant. It follows from Corollary 1 that the matrix  $\mathbf{B} = [b_n^{11} | b_n^{21} | \dots | b_n^{M1}]$  will also have the same block circulant structure. Equation (4.9) can be written more compactly as

$$GB = \Psi. \quad (4.10)$$

Once  $B = [b^{11} | b^{12} | \dots | b^{MN}]$  is computed, for each boundary point  $(x_{i1}, y_{i1})$  on  $\partial\Omega_1$ ,

$$\frac{\partial u}{\partial n}(x_{i1}, y_{i1}) \approx \sum_{k=1}^M \sum_{l=1}^N b_{kl}^{i1} u_{kl}. \quad (4.11)$$

So now in addition to (4.6), instead of (4.7), the equations corresponding to the boundary conditions is written as

$$\sum_{k=1}^M \sum_{l=1}^N b_{kl}^{i1} u_{kl} = g_1(x_{i1}, y_{i1}) \quad \text{and} \quad u_{iN} = g_2(x_{iN}, y_{iN}), \quad i = 1, \dots, M. \quad (4.12)$$

The system can now be effectively solved using the Matrix Decomposition Algorithm as detailed below.

#### 4.1.2 Matrix Decomposition Algorithm

As shown in section 3.1, the system (4.2) can be premultiplied by the matrix  $I_N \otimes U_M$ . To take advantage of the fact that the right hand side is block circulant, the system can also be post multiplied by the matrix  $I_N \otimes U_M^*$  in order to diagonalize the submatrices on the right hand side for additional computational savings. The system then becomes

$$(I_N \otimes U_M) G (I_N \otimes U_M^*) (I_N \otimes U_M) \mathbf{A} (I_N \otimes U_M^*) = (I_N \otimes U_M) \Phi (I_N \otimes U_M^*), \quad (4.13)$$

or

$$\hat{G} \hat{\mathbf{A}} = \hat{\Phi}, \quad (4.14)$$

where

$$\hat{G} = (I_N \otimes U_M) G (I_N \otimes U_M^*), \quad \hat{\mathbf{A}} = (I_N \otimes U_M) \mathbf{A} (I_N \otimes U_M^*), \quad \hat{\Phi} = (I_N \otimes U_M) \Phi (I_N \otimes U_M^*).$$

The matrices  $\hat{G}$  and  $\hat{\Phi}$  are sparse block diagonal and can be calculated using the fast Fourier transform (FFT). More specifically,

$$\begin{aligned} \hat{G} = (I_N \otimes U_M) G (I_N \otimes U_M^*) &= \begin{pmatrix} U_M G_{1,1} U_M^* & U_M G_{1,2} U_M^* & \cdots & U_M G_{1,N} U_M^* \\ U_M G_{2,1} U_M^* & U_M G_{2,2} U_M^* & \cdots & U_M G_{2,N} U_M^* \\ \vdots & \vdots & & \vdots \\ U_M G_{N,1} U_M^* & U_M G_{N,2} U_M^* & \cdots & U_M G_{N,N} U_M^* \end{pmatrix} \\ &= \begin{pmatrix} \hat{G}_{1,1} & \hat{G}_{1,2} & \cdots & \hat{G}_{1,N} \\ \hat{G}_{2,1} & \hat{G}_{2,2} & \cdots & \hat{G}_{2,N} \\ \vdots & \vdots & & \vdots \\ \hat{G}_{N,1} & \hat{G}_{N,2} & \cdots & \hat{G}_{N,N} \end{pmatrix}, \end{aligned} \quad (4.15)$$

$$\begin{aligned} \hat{A} = (I_N \otimes U_M) A (I_N \otimes U_M^*) &= \begin{pmatrix} U_M A_{1,1} U_M^* & U_M A_{1,2} U_M^* & \cdots & U_M A_{1,N} U_M^* \\ U_M A_{2,1} U_M^* & U_M A_{2,2} U_M^* & \cdots & U_M A_{2,N} U_M^* \\ \vdots & \vdots & & \vdots \\ U_M A_{N,1} U_M^* & U_M A_{N,2} U_M^* & \cdots & U_M A_{N,N} U_M^* \end{pmatrix} \\ &= \begin{pmatrix} \hat{A}_{1,1} & \hat{A}_{1,2} & \cdots & \hat{A}_{1,N} \\ \hat{A}_{2,1} & \hat{A}_{2,2} & \cdots & \hat{A}_{2,N} \\ \vdots & \vdots & & \vdots \\ \hat{A}_{N,1} & \hat{A}_{N,2} & \cdots & \hat{A}_{N,N} \end{pmatrix}, \end{aligned} \quad (4.16)$$

and

$$\begin{aligned} \hat{\Phi} = (I_N \otimes U_M) \Phi (I_N \otimes U_M^*) &= \begin{pmatrix} U_M \Phi_{1,1} U_M^* & U_M \Phi_{1,2} U_M^* & \cdots & U_M \Phi_{1,N} U_M^* \\ U_M \Phi_{2,1} U_M^* & U_M \Phi_{2,2} U_M^* & \cdots & U_M \Phi_{2,N} U_M^* \\ \vdots & \vdots & & \vdots \\ U_M \Phi_{N,1} U_M^* & U_M \Phi_{N,2} U_M^* & \cdots & U_M \Phi_{N,N} U_M^* \end{pmatrix} \\ &= \begin{pmatrix} \hat{\Phi}_{1,1} & \hat{\Phi}_{1,2} & \cdots & \hat{\Phi}_{1,N} \\ \hat{\Phi}_{2,1} & \hat{\Phi}_{2,2} & \cdots & \hat{\Phi}_{2,N} \\ \vdots & \vdots & & \vdots \\ \hat{\Phi}_{N,1} & \hat{\Phi}_{N,2} & \cdots & \hat{\Phi}_{N,N} \end{pmatrix}. \end{aligned} \quad (4.17)$$

From the properties of circulant matrices, the matrices  $\hat{G}$  and  $\hat{\Phi}$  both consist of  $N^2$  blocks of order  $M$ , each of which is diagonal. It follows that the matrix  $\hat{A}$  will also be diagonal.

The coefficient matrix  $\hat{A}$  can therefore be obtained by decomposing system (4.2) into the  $M$  independent systems of order  $N$

$$(\hat{G}_{n_1, n_2})_m (\hat{A}_{n_1, n_2})_m = (\hat{\Phi}_{n_1, n_2})_m, \quad n_1, n_2 = 1, \dots, N, \quad m = 1, \dots, M. \quad (4.18)$$

For the mixed Neumann/Dirichlet problem, the corresponding coefficients,  $\hat{B} = (I_n \otimes U_M) B (I_n \otimes U_M^*)$  can be obtained in a similar fashion. By pre-multiplying system



(4.9) by the matrix  $I_N \otimes U_M$  and post-multiplying it by the matrix  $U_M^*$ , the system becomes

$$(I_N \otimes U_M) G (I_N \otimes U_M^*) (I_N \otimes U_M) \mathbf{B} U_M^* = (I_N \otimes U_M) \Psi U_M^*, \quad (4.19)$$

or

$$\hat{G} \hat{\mathbf{B}} = \hat{\Psi}, \quad (4.20)$$

where the matrix  $\hat{\Psi}$  is sparse block diagonal and can be calculated using FFT. More specifically,

$$\hat{\mathbf{B}} = (I_N \otimes U_M) \mathbf{B} U_M^* = \begin{pmatrix} U_M \mathbf{B}_1 U_M^* \\ U_M \mathbf{B}_2 U_M^* \\ \vdots \\ U_M \mathbf{B}_N U_M^* \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{B}}_1 \\ \hat{\mathbf{B}}_2 \\ \vdots \\ \hat{\mathbf{B}}_N \end{pmatrix}, \quad (4.21)$$

and

$$\hat{\Psi} = (I_N \otimes U_M) \Psi U_M^* = \begin{pmatrix} U_M \Psi_1 U_M^* \\ U_M \Psi_2 U_M^* \\ \vdots \\ U_M \Psi_N U_M^* \end{pmatrix} = \begin{pmatrix} \hat{\Psi}_1 \\ \hat{\Psi}_2 \\ \vdots \\ \hat{\Psi}_N \end{pmatrix}. \quad (4.22)$$

The matrix  $\hat{G}$  consists of  $N^2$  diagonal blocks of order  $M$  and from the properties of circulant matrices, the matrix  $\hat{\Psi}$  will consist of  $N$  blocks of order  $M$ , each of which is diagonal. In addition, since the coefficient matrix  $\mathbf{B}$  is block circulant, consisting of  $N$  circulant submatrices of order  $M$ ,  $\hat{\mathbf{B}}$  will consist of  $N$  diagonal blocks of order  $M$ .  $\hat{\mathbf{B}}$  can therefore be obtained by decomposing system (4.20) into solving the  $M$  independent systems of order  $N$

$$(\hat{G}_{n_1, n_2})_m (\hat{\mathbf{B}}_n)_m = (\hat{\Psi}_n)_m, \quad n = 1, \dots, N, \quad m = 1, \dots, M. \quad (4.23)$$

By appropriately arranging the matrices  $\hat{\mathbf{B}}$  (for the Neumann boundary condition) and  $\hat{\mathbf{A}}$  (for the differential equation), the matrix  $\hat{A}_1$  can be obtained to be used in the following system

$$\hat{A}_1 \hat{u} = \hat{f}, \quad (4.24)$$

where  $\hat{u} = (I_N \otimes U_M)u$  and  $\hat{f} = (I_N \otimes U_M)f$ .

System (4.24) is sparse block diagonal and may be solved efficiently to yield  $\hat{u}$ . Finally,  $u$  is obtained from  $u = (I_N \otimes U_M^*) \hat{u}$ . Note that all operations involving multiplication by the matrices  $I_N \otimes U_M$  and  $I_N \otimes U_M^*$  are carried out using the MATLAB<sup>®</sup> [41] FFT commands `ifft` and `fft`, respectively.

### 4.1.3 Biharmonic

Consider the following biharmonic equation

$$\Delta^2 u = f(x, y) \quad \text{in } \Omega, \quad (4.25a)$$

subject to the boundary conditions

$$u = g_1(x, y) \quad \text{and} \quad \frac{\partial u}{\partial \mathbf{n}} = h_1(x, y) \quad \text{on } \partial\Omega_1, \quad (4.25b)$$

and

$$u = g_2(x, y) \quad \text{and} \quad \frac{\partial u}{\partial \mathbf{n}} = h_2(x, y) \quad \text{on } \partial\Omega_2, \quad (4.25c)$$

or the boundary conditions

$$u = g_1(x, y) \quad \text{and} \quad \Delta u = h_1(x, y) \quad \text{on } \partial\Omega_1, \quad (4.25d)$$

and

$$u = g_2(x, y) \quad \text{and} \quad \Delta u = h_2(x, y) \quad \text{on } \partial\Omega_2. \quad (4.25e)$$

Problem (4.25a),(4.25b)-(4.25c) is known as the *first biharmonic problem* whereas the problem (4.25a),(4.25d)-(4.25e) is known as the *second biharmonic problem*.

The collocation points are defined in the same way as the previous case for the Poisson problem. However, the interior points are defined as  $(x_{ij}, y_{ij})_{i=1}^M, j = 3, \dots, N-2$ , while the boundary points are defined as  $\{(x_{i1}, y_{i1})\}_{i=1}^M$  and  $\{(x_{iN}, y_{iN})\}_{i=1}^M$ . Fewer interior points are taken than in the previous case due to the fact that it is necessary to impose two boundary conditions on the boundary instead of one in the biharmonic case. Hence, there are  $(N-4)M$  interior points while there are  $4M$  boundary points for a total of  $MN$  collocation points.

In the RBF-DQ method, for each point  $(x_{ij}, y_{ij}), i = 1, \dots, M, j = 1, \dots, N$ , and for each RBF  $\phi_{mn}, m = 1, \dots, M, n = 1, \dots, N$ ,

$$\Delta^2 \phi_{mn}(x_{ij}, y_{ij}) = \sum_{k=1}^M \sum_{\ell=1}^N a_{kl}^{ij} \phi_{mn}(x_{k\ell}, y_{k\ell}),$$

which in vector form, gives for each point  $(x_{ij}, y_{ij})$ ,

$$\phi_{\Delta^2}^{ij} = G a^{ij}. \quad (4.26)$$

As was the case in the Poisson equation, due to the circular distribution of the collocation points, the matrix  $G$  is block circulant of size  $(MN \times MN)$ . The matrix  $\Phi = \left[ \phi_{\Delta^2}^{11} | \phi_{\Delta^2}^{12} | \phi_{\Delta^2}^{13} | \dots | \phi_{\Delta^2}^{MN} \right]$  consists of  $N^2$  circulant submatrices of order  $M$  and is therefore

also block circulant. This means that the matrix  $A = [a^{11}|a^{12}|a^{13}|\dots|a^{MN}]$  will also be block circulant. Equation (4.26) can be written more compactly as

$$GA = \Phi. \quad (4.27)$$

Once  $A = [a^{11}|a^{12}|\dots|a^{MN}]$  is computed, for each point  $(x_{ij}, y_{ij})$ ,

$$\Delta^2 u(x_{ij}, y_{ij}) \approx \sum_{k=1}^M \sum_{l=1}^N a_{kl}^{ij} u_{kl}, \quad (4.28)$$

where the  $u_{kl}$  are approximations of  $u(x_{kl}, y_{kl})$ . Substitution in the differential equation yields

$$\sum_{k=1}^M \sum_{l=1}^N a_{kl}^{ij} u_{kl} = f(x_{ij}, y_{ij}), i = 1, \dots, M, j = 3, \dots, N-2. \quad (4.29)$$

In the case of the first biharmonic problem, for each boundary point  $(x_{is}, y_{is})$ ,  $i = 1, \dots, M$ ,  $s = 1, N$ , on  $\partial\Omega$  and for each RBF  $\varphi_{mn}$ ,  $m = 1, \dots, M$ ,  $n = 1, \dots, N$ ,

$$\frac{\partial \varphi_{mn}}{\partial n}(x_{is}, y_{is}) = \sum_{k=1}^M \sum_{\ell=1}^N b_{k\ell}^{is} \varphi_{mn}(x_{k\ell}, y_{k\ell}), \quad (4.30)$$

which in vector form gives for each boundary point  $(x_{is}, y_{is})$ ,

$$\varphi_n^{is} = G b^{is}. \quad (4.31)$$

By observing that the  $MN \times M$  matrix  $\Psi = [\varphi_n^{1s}|\varphi_n^{2s}|\dots|\varphi_n^{Ms}]$ ,  $s = 1, N$ , consists of  $N$  circulant submatrices of order  $M$ ,  $B = [b^{1s}|b^{2s}|\dots|b^{Ms}]$ ,  $s = 1, N$ , is also block circulant. Equation (4.31) can be written more compactly as

$$GB = \Psi. \quad (4.32)$$

Once  $B = [b^{1s}|b^{2s}|\dots|b^{Ms}]$ ,  $s = 1, \dots, N$ , is computed, for each boundary point  $(x_{is}, y_{is})$ ,  $i = 1, \dots, M$ ,  $s = 1, N$ ,

$$\frac{\partial u}{\partial n}(x_{is}, y_{is}) \approx \sum_{k=1}^M \sum_{l=1}^N b_{kl}^{is} u_{kl}, \quad (4.33)$$

In the case of the second biharmonic problem, instead of (4.30), for each boundary point  $(x_{is}, y_{is})$ ,  $i = 1, \dots, M$ ,  $s = 1, N$ , and for each RBF  $\varphi_{mn}$ ,  $m = 1, \dots, M$ ,  $n = 1, \dots, N$ ,

$$\Delta \varphi_{mn}(x_{is}, y_{is}) = \sum_{k=1}^M \sum_{\ell=1}^N b_{k\ell}^{is} \varphi_{mn}(x_{k\ell}, y_{k\ell}), \quad (4.34)$$

which in vector form gives for each boundary point  $(x_{is}, y_{is})$ ,

$$\varphi_{\Delta}^{is} = G b^{is}. \quad (4.35)$$

As in the first biharmonic problem, by observing that the matrix  $\Psi = [\varphi_{\Delta}^{1s} | \varphi_{\Delta}^{2s} | \dots | \varphi_{\Delta}^{Ms}]$ ,  $s = 1, N$ , consists of  $N$  circulant submatrices of order  $M$ ,  $B = [b^{1s} | b^{2s} | \dots | b^{Ms}]$ ,  $s = 1, N$ , will also be block circulant. Equation (4.35) can be written more compactly as Equation (4.32). Once  $B = [b^{1s} | b^{2s} | \dots | b^{Ms}]$  is computed, for each boundary point  $(x_{is}, y_{is})$ ,  $i = 1, \dots, M$ ,  $s = 1, N$ ,

$$\Delta u(x_{is}, y_{is}) \approx \sum_{k=1}^M \sum_{l=1}^N b_{kl}^{is} u_{kl}. \quad (4.36)$$

So now, in addition to Equation (4.29), the boundary condition (4.25b) or (4.25d) is written as

$$u_{i1} = g_1(x_{i1}, y_{i1}), \sum_{k=1}^M \sum_{l=1}^N b_{kl}^{i1} u_{kl} = h_1(x_{i1}, y_{i1}), i = 1, \dots, M, \quad (4.37)$$

and boundary conditions (4.25c) or (4.25e) is written as

$$u_{iN} = g_2(x_{in}, y_{in}), \sum_{k=1}^M \sum_{l=1}^N b_{kl}^{iN} u_{kl} = h_1(x_{iN}, y_{iN}), i = 1, \dots, M. \quad (4.38)$$

The system can then effectively be solved using the same MDA technique as in the Poisson problem described in the previous section.

In this dissertation, the selection of the collocation points must be determined strategically to achieve a circulant structure. The collocation points,  $(x_{ij}, y_{ij})$ , are determined by

$$x_{mn} = r_n \cos\left(\theta_m + \frac{2\pi\alpha_n}{M}\right), \quad y_{mn} = r_n \sin\left(\theta_m + \frac{2\pi\alpha_n}{M}\right), \quad m = 1, \dots, M, \quad n = 1, \dots, N, \quad (4.39)$$

where  $r_n = \gamma_1 + (\gamma_2 - \gamma_1) \frac{n-1}{N-1}$  is the radius of each of the  $N$  concentric circles,  $\theta_m$  is the angle of each of the  $m$  collocation points on a circle, and  $\alpha_n$  is the rotation of the collocation points, which is used to produce more uniform distributions. For the Poisson equation,  $\alpha_n = (-1)^n/4$ ,  $n = 1, \dots, N$ , is used for selecting the collocation points. However, when using this same value for the biharmonic equation, possible singularities may occur as shown in the following theorem.

**Theorem 4.** *For the biharmonic problems (4.25) considered, in the case  $\alpha_n = (-1)^n/4$ ,  $n = 1, \dots, N$ , and  $M = 2m$  is even, the global matrix  $G$  in (4.26) resulting from the proposed discretization is singular.*

**Proof.** For simplicity, consider the RBF-DQ discretization for the case of the domain  $\Omega$  being a disk. The extension for the case of the annulus is trivial. The global matrix  $G$  has the block structure

$$G = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,N} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N,1} & A_{N,2} & \cdots & A_{N,N} \end{pmatrix}, \quad (4.40)$$

where each of the  $M \times M$  submatrices  $A_{i,j}$ ,  $i, j = 1, \dots, N$ , is circulant. Because of the imposition of the Dirichlet boundary condition,

$$A_{N,N} = I_M \quad \text{and} \quad A_{i,N} = 0_M, \quad i = 1, \dots, N-1,$$

where  $I_M$  and  $0_M$  are the  $M \times M$  identity and zero matrices, respectively. Therefore, showing that  $G$  is singular is equivalent to showing that the matrix

$$\mathcal{G} = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,N-1} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N-1,1} & A_{N-1,2} & \cdots & A_{N-1,N-1} \end{pmatrix}, \quad (4.41)$$

is singular.

Without loss of generality, assume that  $N = 2n$  is even.

For the submatrices corresponding to

$$i = 1, 3, \dots, 2n-3, \quad j = 1, 3, \dots, 2n-1,$$

and

$$i = 2, 4, \dots, 2n-2, \quad j = 2, 4, \dots, 2n-2,$$

the circulant structure is

$$A_{i,j} = \text{circ} \left( a_1^{i,j}, a_2^{i,j}, \dots, a_m^{i,j}, a_{m+1}^{i,j}, a_m^{i,j}, \dots, a_3^{i,j}, a_2^{i,j} \right). \quad (4.42)$$

Because of the imposition of the second boundary condition and the omittance of the collocation points on the circle corresponding to  $i = N-1$ , the same structure also applies to the submatrices corresponding to

$$i = 2n-1, \quad j = 2, 4, \dots, 2n-2.$$

For the remaining submatrices corresponding to

$$i = 1, 3, \dots, 2n-3, \quad j = 2, 4, \dots, 2n-2,$$

and

$$i = 2, 4, \dots, 2n - 2, j = 1, 3, \dots, 2n - 1,$$

and also

$$i = 2n - 1, j = 1, 3, \dots, 2n - 1,$$

the circulant structure is

$$A_{i,j} = \text{circ} \left( a_1^{i,j}, a_2^{i,j}, \dots, a_m^{i,j}, a_m^{i,j}, \dots, a_2^{i,j}, a_1^{i,j} \right). \quad (4.43)$$

First, consider the submatrices possessing the structure (4.42). The sum of the odd rows of these matrices yields a row of the form

$$(\alpha^{i,j}, \beta^{i,j}, \alpha^{i,j}, \beta^{i,j}, \dots, \alpha^{i,j}, \beta^{i,j}),$$

while the sum of the even rows yields a row of the form

$$(\beta^{i,j}, \alpha^{i,j}, \beta^{i,j}, \alpha^{i,j}, \dots, \beta^{i,j}, \alpha^{i,j}).$$

Subtracting the sum of columns from the sum of odd rows from the sum of even rows will give one row of the form

$$\gamma^{i,j} = (\gamma^{i,j}, -\gamma^{i,j}, \gamma^{i,j}, -\gamma^{i,j}, \dots, \gamma^{i,j}, -\gamma^{i,j}), \quad (4.44)$$

where  $\gamma^{i,j} = \beta^{i,j} - \alpha^{i,j}$ . Next, consider the submatrices possessing structure (4.43). In this case, the sum of the odd rows of these matrices is equal to the sum of the even rows. As a result, subtracting the sum of columns from the sum of odd rows from the sum of even rows will give one row of zeros.

The matrix (4.41) is thus equivalent to a matrix in which each submatrix with structure (4.44) has a row (say the last row) of type (4.44) while each submatrix with structure (4.43) has a corresponding zero row denoted by 0. Considering the last rows of the matrices in blocks  $i = 2, 4, \dots, 2n - 2$  and  $2n - 1$ , the  $n \times MN$  matrix is defined as

$$T = \begin{bmatrix} 0 & \gamma^{2,2} & 0 & \gamma^{2,4} & \dots & 0 & \gamma^{2,2n-2} & 0 \\ 0 & \gamma^{4,2} & 0 & \gamma^{4,4} & \dots & 0 & \gamma^{4,2n-2} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \gamma^{2n-2,2} & 0 & \gamma^{2n-2,4} & \dots & 0 & \gamma^{2n-2,2n-2} & 0 \\ 0 & \gamma^{2n-1,2} & 0 & \gamma^{2n-1,4} & \dots & 0 & \gamma^{2n-1,2n-2} & 0 \end{bmatrix}.$$

By subtracting appropriate multiples of the first row of matrix  $T$  from the remaining rows, a matrix is obtained where the row vectors  $\gamma^{4,2}, \dots, \gamma^{2n-1,2}$  are replaced by zero row vectors. In a similar subsequent step, subtracting appropriate multiples of the second row from the

third row through the  $n^{\text{th}}$  row yields zero row vectors in place of row vectors  $\gamma^{6,4}, \dots, \gamma^{2n-1,4}$ . Repeating this will eventually yield a full zero row in place of row  $n$  of matrix  $T$ . Thus matrix  $\mathcal{G}$  and hence  $G$  is singular.

Structure (4.43) in some of the submatrices is due to the relationship between any two consecutive circles in the RBF discretization, say circles  $j-1$  and  $j$  with radii  $r_{j-1}$  and  $r_j$ , respectively. Without loss of generality, assume that  $j$  is odd. From (4.39), the first point of circle  $j$  will be the point  $A = (r_j \cos(-\frac{\pi}{2M}), r_j \sin(-\frac{\pi}{2M}))$ . Similarly, the first point of circle  $j-1$  will be the point  $B = (r_{j-1} \cos(\frac{\pi}{2M}), r_{j-1} \sin(\frac{\pi}{2M}))$  and the last ( $M^{\text{th}}$ ) point of circle  $j-1$  will be the point  $C = (r_{j-1} \cos(-\frac{3\pi}{2M}), r_{j-1} \sin(-\frac{3\pi}{2M}))$ . The distances  $|AB|$  and  $|AC|$  are equal and are given by

$$|AB| = |AC| = \sqrt{r_j^2 + r_{j-1}^2 - 2r_j r_{j-1} \cos(\frac{\pi}{M})}.$$

For similar reasons, the distance of  $A$  from the second point of circle  $j-1$  is equal to the distance of  $A$  from the point  $M-1$  of circle  $j-1$  and so on. This yields the first circulant structure in (4.43). In a similar way, the distance of point  $B$  on circle  $j-1$  from the first and second points on circle  $j$  are also equal, leading to the second structure in (4.43).  $\square$

To avoid possible singularities,  $\alpha_n = \frac{(-1)^n}{5}$  will be used for the biharmonic equations.

#### 4.1.4 Cauchy Navier

Finally, consider the Cauchy–Navier system in  $\mathbb{R}^2$  for the displacements  $(u_1, u_2)$  in the form (see, e.g. [21])

$$\begin{cases} \mathcal{L}_1(u_1, u_2) \equiv \mathcal{L}_{11}u_1 + \mathcal{L}_{12}u_2 \equiv \mu\Delta u_1 + \frac{\mu}{1-2\nu} \left( \frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_2}{\partial x \partial y} \right) = f_1, \\ \mathcal{L}_2(u_1, u_2) \equiv \mathcal{L}_{21}u_1 + \mathcal{L}_{22}u_2 \equiv \frac{\mu}{1-2\nu} \left( \frac{\partial^2 u_1}{\partial x \partial y} + \frac{\partial^2 u_2}{\partial y^2} \right) + \mu\Delta u_2 = f_2, \end{cases} \quad \text{in } \Omega, \quad (4.45a)$$

subject to the Dirichlet boundary conditions

$$u_1 = g_1 \quad \text{and} \quad u_2 = h_1 \quad \text{on} \quad \partial\Omega_1, \quad (4.45b)$$

and

$$u_1 = g_2 \quad \text{and} \quad u_2 = h_2 \quad \text{on} \quad \partial\Omega_2, \quad (4.45c)$$

or the mixed Neumann/Dirichlet boundary conditions

$$t_1 = g_1 \quad \text{and} \quad t_2 = h_1 \quad \text{on} \quad \partial\Omega_1, \quad (4.45d)$$

and

$$u_1 = g_2 \quad \text{and} \quad u_2 = h_2 \quad \text{on} \quad \partial\Omega_2. \quad (4.45e)$$

In (4.45a) the constant  $\nu \in [0, 1/2)$  is Poisson's ratio, and  $\mu > 0$  is the shear modulus. Also, in (4.45a) the operators  $\mathcal{L}_{11}, \mathcal{L}_{12}, \mathcal{L}_{21}$ , and  $\mathcal{L}_{22}$  are defined by

$$\mathcal{L}_{11} \equiv \mu\Delta + \frac{\mu}{1-2\nu} \frac{\partial^2}{\partial x^2}, \quad \mathcal{L}_{12} \equiv \frac{\mu}{1-2\nu} \frac{\partial^2}{\partial x \partial y}$$

$$\mathcal{L}_{21} \equiv \mathcal{L}_{12}, \quad \mathcal{L}_{22} \equiv \mu\Delta + \frac{\mu}{1-2\nu} \frac{\partial^2}{\partial y^2}.$$

In (4.45d),  $t_1$  and  $t_2$  are the tractions [21] defined by

$$t_1 = 2\mu \left[ \left( \frac{1-\nu}{1-2\nu} \right) \frac{\partial u_1}{\partial x} + \left( \frac{\nu}{1-2\nu} \right) \frac{\partial u_2}{\partial y} \right] n_x + \mu \left[ \frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} \right] n_y,$$

$$t_2 = \mu \left[ \frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} \right] n_x + 2\mu \left[ \left( \frac{\nu}{1-2\nu} \right) \frac{\partial u_1}{\partial x} + \left( \frac{1-\nu}{1-2\nu} \right) \frac{\partial u_2}{\partial y} \right] n_y.$$

Problem (4.45a), (4.45b)-(4.45c) is a *Dirichlet boundary value problem* whereas problem (4.45a), (4.45d)-(4.45e) is a *mixed Neumann/Dirichlet boundary value problem*.

In the RBF-DQ method, for each point  $(x_{ij}, y_{ij})$ ,  $i = 1, \dots, M$ ,  $j = 1, \dots, N$ , and for each RBF  $\phi_{mn}$ ,  $m = 1, \dots, M$ ,  $n = 1, \dots, N$ ,

$$\mathcal{L}_{11} \phi_{mn}(x_{ij}, y_{ij}) = \sum_{k=1}^M \sum_{\ell=1}^N a_{kl}^{ij} \phi_{mn}(x_{k\ell}, y_{k\ell}), \quad \mathcal{L}_{12} \phi_{mn}(x_{ij}, y_{ij}) = \sum_{k=1}^M \sum_{\ell=1}^N b_{kl}^{ij} \phi_{mn}(x_{k\ell}, y_{k\ell}),$$

$$\mathcal{L}_{21} \phi_{mn}(x_{ij}, y_{ij}) = \sum_{k=1}^M \sum_{\ell=1}^N c_{kl}^{ij} \phi_{mn}(x_{k\ell}, y_{k\ell}), \quad \mathcal{L}_{22} \phi_{mn}(x_{ij}, y_{ij}) = \sum_{k=1}^M \sum_{\ell=1}^N d_{kl}^{ij} \phi_{mn}(x_{k\ell}, y_{k\ell}),$$

which in vector form gives for each point  $(x_{ij}, y_{ij})$ ,

$$\varphi_{\mathcal{L}_{11}}^{ij} = G a^{ij}, \quad \varphi_{\mathcal{L}_{12}}^{ij} = G b^{ij}, \quad \varphi_{\mathcal{L}_{21}}^{ij} = G c^{ij}, \quad \varphi_{\mathcal{L}_{22}}^{ij} = G d^{ij}. \quad (4.46)$$

The matrix  $G$  is block circulant due to the distribution of the collocation points, but unlike the Poisson and Biharmonic problems, this is the only matrix having a block circulant structure. However, it is possible to transform the matrices to a block circulant matrix by using an appropriate transformation which will be introduced in the next section. Equations (4.46) can be written more compactly as

$$\left( \begin{array}{c|c} G & 0 \\ \hline 0 & G \end{array} \right) \left( \begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right) = \left( \begin{array}{c|c} \Phi_{11} & \Phi_{12} \\ \hline \Phi_{21} & \Phi_{22} \end{array} \right), \quad (4.47)$$

where the matrix  $G$  is defined in (4.3) and

$$\mathbf{A} = [a^{11}|a^{12}|a^{13}|\dots|a^{MN}], \quad \mathbf{B} = [b^{11}|b^{12}|b^{13}|\dots|b^{MN}],$$

$$\mathbf{C} = [c^{11}|c^{12}|c^{13}|\dots|c^{MN}], \quad \mathbf{D} = [d^{11}|d^{12}|d^{13}|\dots|d^{MN}],$$



and

$$\begin{aligned}\Phi_{11} &= [\varphi_{\mathcal{L}_{11}}^{11} | \varphi_{\mathcal{L}_{11}}^{12} | \varphi_{\mathcal{L}_{11}}^{13} | \dots | \varphi_{\mathcal{L}_{11}}^{MN}], & \Phi_{12} &= [\varphi_{\mathcal{L}_{12}}^{11} | \varphi_{\mathcal{L}_{12}}^{12} | \varphi_{\mathcal{L}_{12}}^{13} | \dots | \varphi_{\mathcal{L}_{12}}^{MN}], \\ \Phi_{21} &= [\varphi_{\mathcal{L}_{21}}^{11} | \varphi_{\mathcal{L}_{21}}^{12} | \varphi_{\mathcal{L}_{21}}^{13} | \dots | \varphi_{\mathcal{L}_{21}}^{MN}], & \Phi_{22} &= [\varphi_{\mathcal{L}_{22}}^{11} | \varphi_{\mathcal{L}_{22}}^{12} | \varphi_{\mathcal{L}_{22}}^{13} | \dots | \varphi_{\mathcal{L}_{22}}^{MN}],\end{aligned}$$

are  $MN \times MN$  matrices. Each of the  $MN \times MN$  matrices  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$  consists of  $N^2$  submatrices  $A_{n_1, n_2}, B_{n_1, n_2}, C_{n_1, n_2}, D_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , each of order  $M$ , respectively. Similarly, each of the  $MN \times MN$  matrices  $\Phi_{ij}$  consists of  $N^2$  submatrices  $\Phi_{ij_{n_1, n_2}}, n_1, n_2 = 1, \dots, N$ , of order  $M$ , respectively.

Once  $A, B, C, D$  are computed, for each point  $(x_{ij}, y_{ij})$ ,

$$\begin{aligned}\mathcal{L}_{11}u_1(x_{ij}, y_{ij}) &\approx \sum_{k=1}^M \sum_{\ell=1}^N a_{kl}^{ij} u_{1k\ell}, & \mathcal{L}_{12}u_2(x_{ij}, y_{ij}) &\approx \sum_{k=1}^M \sum_{\ell=1}^N b_{kl}^{ij} u_{2k\ell}, \\ \mathcal{L}_{21}u_1(x_{ij}, y_{ij}) &\approx \sum_{k=1}^M \sum_{\ell=1}^N c_{kl}^{ij} u_{1k\ell}, & \mathcal{L}_{22}u_2(x_{ij}, y_{ij}) &\approx \sum_{k=1}^M \sum_{\ell=1}^N d_{kl}^{ij} u_{2k\ell},\end{aligned}$$

where the  $u_{1k\ell}$  and  $u_{2k\ell}$  are approximations of  $u_1(x_{k\ell}, y_{k\ell})$  and  $u_2(x_{k\ell}, y_{k\ell})$ , respectively.

In the Dirichlet problem, substitution in the differential equations yields

$$\begin{aligned}\sum_{k=1}^M \sum_{\ell=1}^N a_{kl}^{ij} u_{1k\ell} + \sum_{k=1}^M \sum_{\ell=1}^N b_{kl}^{ij} u_{2k\ell} &= f_1(x_{ij}, y_{ij}), \\ \sum_{k=1}^M \sum_{\ell=1}^N c_{kl}^{ij} u_{1k\ell} + \sum_{k=1}^M \sum_{\ell=1}^N d_{kl}^{ij} u_{2k\ell} &= f_2(x_{ij}, y_{ij}), \quad i = 1, \dots, M, j = 2, \dots, N-1\end{aligned}\quad (4.48)$$

and the Dirichlet boundary conditions yield

$$u_{1i1} = g_1(x_{i1}, y_{i1}), \quad u_{2i1} = h_1(x_{i1}, y_{i1}), \quad i = 1, \dots, M. \quad (4.49)$$

$$u_{1iN} = g_2(x_{iN}, y_{iN}), \quad u_{2iN} = h_2(x_{iN}, y_{iN}), \quad i = 1, \dots, M. \quad (4.50)$$

In the case of the mixed Neumann/Dirichlet problem, for each boundary point  $(x_{i1}, y_{i1})$ ,  $i = 1, \dots, M$  on  $\partial\Omega_1$ , and for each RBF  $\varphi_{mn}$ ,  $m = 1, \dots, M$ ,  $n = 1, \dots, N$ ,

$$\mathcal{T}_{11}\varphi_{mn}(x_{i1}, y_{i1}) = \left[ 2\mu \left( \frac{1-v}{1-2v} \right) \frac{\partial \varphi_{mn}}{\partial x} n_x + \mu \frac{\partial \varphi_{mn}}{\partial y} n_y \right] (x_{i1}, y_{i1}) = \sum_{k=1}^M \sum_{\ell=1}^N e_{kl}^{i1} \varphi_{mn}(x_{k\ell}, y_{k\ell}),$$

$$\mathcal{T}_{12}\varphi_{mn}(x_{i1}, y_{i1}) = \left[ 2\mu \left( \frac{v}{1-2v} \right) \frac{\partial \varphi_{mn}}{\partial y} n_x + \mu \frac{\partial \varphi_{mn}}{\partial x} n_y \right] (x_{i1}, y_{i1}) = \sum_{k=1}^M \sum_{\ell=1}^N f_{kl}^{i1} \varphi_{mn}(x_{k\ell}, y_{k\ell}),$$

$$\mathcal{T}_{21}\varphi_{mn}(x_{i1}, y_{i1}) = \left[ \mu \frac{\partial \varphi_{mn}}{\partial y} n_x + 2\mu \left( \frac{v}{1-2v} \right) \frac{\partial \varphi_{mn}}{\partial x} n_y \right] (x_{i1}, y_{i1}) = \sum_{k=1}^M \sum_{\ell=1}^N g_{kl}^{i1} \varphi_{mn}(x_{k\ell}, y_{k\ell}),$$

$$\mathcal{T}_{22}\varphi_{mn}(x_{i1}, y_{i1}) = \left[ \mu \frac{\partial \varphi_{mn}}{\partial x} \mathbf{n}_x + 2\mu \left( \frac{1-\nu}{1-2\nu} \right) \frac{\partial \varphi_{mn}}{\partial y} \mathbf{n}_y \right] (x_{i1}, y_{i1}) = \sum_{k=1}^M \sum_{\ell=1}^N h_{kl}^{i1} \varphi_{mn}(x_{k\ell}, y_{k\ell}),$$

which in vector form gives for each boundary point  $(x_{i1}, y_{i1})$  on  $\partial\Omega_1$ ,

$$\varphi_{\mathcal{T}_{11}}^{i1} = G e^{i1}, \quad \varphi_{\mathcal{T}_{12}}^{i1} = G f^{i1}, \quad \varphi_{\mathcal{T}_{21}}^{i1} = G g^{i1}, \quad \varphi_{\mathcal{T}_{22}}^{i1} = G h^{i1}. \quad (4.51)$$

Similarly, we can transform the matrices to block circulant using an appropriate transformation. Equation (4.51) can be written more compactly as

$$\left( \begin{array}{c|c} G & 0 \\ \hline 0 & G \end{array} \right) \left( \begin{array}{c|c} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hline \hat{\mathbf{C}} & \hat{\mathbf{D}} \end{array} \right) = \left( \begin{array}{c|c} \hat{\Phi}_{11} & \hat{\Phi}_{12} \\ \hline \hat{\Phi}_{21} & \hat{\Phi}_{22} \end{array} \right), \quad (4.52)$$

where

$$\begin{aligned} \hat{\mathbf{A}} &= [e^{11}|e^{21}|\dots|e^{M1}], & \hat{\mathbf{B}} &= [f^{11}|f^{21}|\dots|f^{M1}], \\ \hat{\mathbf{C}} &= [g^{11}|g^{21}|\dots|g^{M1}], & \hat{\mathbf{D}} &= [h^{11}|h^{21}|\dots|h^{M1}], \end{aligned}$$

and

$$\begin{aligned} \hat{\Phi}_{11} &= [\varphi_{\mathcal{T}_{11}}^{11}|\varphi_{\mathcal{T}_{11}}^{21}|\dots|\varphi_{\mathcal{T}_{11}}^{M1}], & \hat{\Phi}_{12} &= [\varphi_{\mathcal{T}_{12}}^{11}|\varphi_{\mathcal{T}_{12}}^{21}|\dots|\varphi_{\mathcal{T}_{12}}^{M1}], \\ \hat{\Phi}_{21} &= [\varphi_{\mathcal{T}_{21}}^{11}|\varphi_{\mathcal{T}_{21}}^{21}|\dots|\varphi_{\mathcal{T}_{21}}^{M1}], & \hat{\Phi}_{22} &= [\varphi_{\mathcal{T}_{22}}^{11}|\varphi_{\mathcal{T}_{22}}^{21}|\dots|\varphi_{\mathcal{T}_{22}}^{M1}], \end{aligned}$$

are  $MN \times M$  matrices. Each of the  $MN \times M$  matrices  $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}$  consists of  $N$  submatrices  $A_n, B_n, C_n, D_n, n = 1, \dots, N$ , of order  $M$ , respectively. Similarly, each of the  $MN \times M$  matrices  $\hat{\Phi}_{ij}$  consists of  $N$  submatrices  $\Phi_{ij,n}, n = 1, \dots, N$ , of order  $M$ , respectively.

Once  $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}$  is computed, for each boundary point  $(x_{i1}, y_{i1})$  on  $\partial\Omega_1$ ,

$$\mathcal{T}_{11}u_1(x_{ij}, y_{ij}) \approx \sum_{k=1}^M \sum_{\ell=1}^N e_{kl}^{i1} u_{1k\ell}, \quad \mathcal{T}_{12}u_2(x_{ij}, y_{ij}) \approx \sum_{k=1}^M \sum_{\ell=1}^N f_{kl}^{i1} u_{2k\ell},$$

$$\mathcal{T}_{21}u_1(x_{ij}, y_{ij}) \approx \sum_{k=1}^M \sum_{\ell=1}^N g_{kl}^{i1} u_{1k\ell}, \quad \mathcal{T}_{22}u_2(x_{ij}, y_{ij}) \approx \sum_{k=1}^M \sum_{\ell=1}^N h_{kl}^{i1} u_{2k\ell}.$$

In this case, equations (4.49) are replaced by

$$\begin{aligned} \sum_{k=1}^M \sum_{\ell=1}^N e_{kl}^{i1} u_{1k\ell} + \sum_{k=1}^M \sum_{\ell=1}^N f_{kl}^{i1} u_{2k\ell} &= g_1(x_{i1}, y_{i1}), \\ \sum_{k=1}^M \sum_{\ell=1}^N g_{kl}^{i1} u_{1k\ell} + \sum_{k=1}^M \sum_{\ell=1}^N h_{kl}^{i1} u_{2k\ell} &= h_1(x_{i1}, y_{i1}), \quad i = 1, \dots, M. \end{aligned} \quad (4.53)$$

## 4.2 Circulant Matrix Transform

This section presents a transformation which creates a block circulant structure for any matrix. This is necessary for problems such as the Cauchy-Navier system since the right hand side does not possess the block circulant structure that is required for the MDA to be applied [39]. Begin by rewriting system (4.47) as

$$\mathfrak{G}\mathcal{A} = \Theta, \quad (4.54)$$

where the  $2MN \times 2MN$  matrix  $\mathfrak{G}$  consists of  $N^2$ ,  $2M \times 2M$  submatrices  $\mathfrak{G}_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , where each  $2 \times 2$  array

$$(\mathfrak{G}_{n_1, n_2})_{m_1, m_2} = \left( \begin{array}{c|c} (G_{n_1, n_2})_{m_1, m_2} & 0 \\ \hline 0 & (G_{n_1, n_2})_{m_1, m_2} \end{array} \right),$$

and  $(G_{n_1, n_2})_{m_1, m_2} = \varphi_{m_2 n_2}(x_{m_1 n_1}, y_{m_1 n_1})$ . The  $2MN \times 2MN$  matrix  $\mathcal{A}$  consists of  $N^2$ ,  $2M \times 2M$  submatrices  $\mathcal{A}_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , where each  $2 \times 2$  array

$$(\mathcal{A}_{n_1, n_2})_{m_1, m_2} = \left( \begin{array}{c|c} (A_{n_1, n_2})_{m_1, m_2} & (B_{n_1, n_2})_{m_1, m_2} \\ \hline (C_{n_1, n_2})_{m_1, m_2} & (D_{n_1, n_2})_{m_1, m_2} \end{array} \right), \quad m_1, m_2 = 1, \dots, M,$$

and the  $2MN \times 2MN$  matrix  $\Theta$  consists of  $N^2$ ,  $2M \times 2M$  submatrices  $\Theta_{n_1, n_2}, n_1, n_2 = 1, \dots, N$ , where each  $2 \times 2$  array

$$(\Theta_{n_1, n_2})_{m_1, m_2} = \left( \begin{array}{c|c} (\Phi_{11n_1, n_2})_{m_1, m_2} & (\Phi_{12n_1, n_2})_{m_1, m_2} \\ \hline (\Phi_{21n_1, n_2})_{m_1, m_2} & (\Phi_{22n_1, n_2})_{m_1, m_2} \end{array} \right).$$

Next, the  $(2M \times 2M)$  matrix

$$R = \left( \begin{array}{c|c|c|c|c|c} R_{\theta_1} & 0 & 0 & \cdots & 0 & 0 \\ \hline 0 & R_{\theta_2} & 0 & \cdots & 0 & 0 \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \hline 0 & 0 & 0 & \cdots & R_{\theta_{M-1}} & 0 \\ \hline 0 & 0 & 0 & \cdots & 0 & R_{\theta_M} \end{array} \right) \quad (4.55)$$

where  $R_{\theta_k} = \begin{pmatrix} \cos \theta_k & \sin \theta_k \\ \sin \theta_k & -\cos \theta_k \end{pmatrix}$ ,  $\theta_k = \frac{2\pi(k-1)}{M}$ , can be used to complete the transformation. It can be shown that  $R_{\theta_k}^2 = I_2$  which implies that  $R^2 = I_{2M}$ . By pre-multiplying and post-multiplying the  $2MN \times 2MN$  system (4.54) by the  $2MN \times 2MN$  matrix  $I_N \otimes R$ , the system becomes

$$(I_N \otimes R) \mathfrak{G}\mathcal{A} (I_N \otimes R) = (I_N \otimes R) \Theta (I_N \otimes R), \quad (4.56)$$

or

$$\tilde{\mathcal{G}}\tilde{\mathcal{A}} = \tilde{\Theta}, \quad (4.57)$$

where

$$\tilde{\mathcal{G}} = (I_N \otimes R) \mathcal{G} (I_N \otimes R), \quad \tilde{\mathcal{A}} = (I_N \otimes R) \mathcal{A} (I_N \otimes R), \quad \tilde{\Theta} = (I_N \otimes R) \Theta (I_N \otimes R).$$

The  $2MN \times 2MN$  matrices  $\tilde{\mathcal{G}}$ ,  $\tilde{\mathcal{A}}$  and  $\tilde{\Theta}$  can be written as

$$\tilde{\mathcal{G}} = \begin{pmatrix} \tilde{\mathcal{G}}_{1,1} & \tilde{\mathcal{G}}_{1,2} & \cdots & \tilde{\mathcal{G}}_{1,N} \\ \tilde{\mathcal{G}}_{2,1} & \tilde{\mathcal{G}}_{2,2} & \cdots & \tilde{\mathcal{G}}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathcal{G}}_{N,1} & \tilde{\mathcal{G}}_{N,2} & \cdots & \tilde{\mathcal{G}}_{N,N} \end{pmatrix}, \quad \tilde{\mathcal{A}} = \begin{pmatrix} \tilde{\mathcal{A}}_{1,1} & \tilde{\mathcal{A}}_{1,2} & \cdots & \tilde{\mathcal{A}}_{1,N} \\ \tilde{\mathcal{A}}_{2,1} & \tilde{\mathcal{A}}_{2,2} & \cdots & \tilde{\mathcal{A}}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathcal{A}}_{N,1} & \tilde{\mathcal{A}}_{N,2} & \cdots & \tilde{\mathcal{A}}_{N,N} \end{pmatrix}, \quad (4.58)$$

and

$$\tilde{\Theta} = \begin{pmatrix} \tilde{\Theta}_{1,1} & \tilde{\Theta}_{1,2} & \cdots & \tilde{\Theta}_{1,N} \\ \tilde{\Theta}_{2,1} & \tilde{\Theta}_{2,2} & \cdots & \tilde{\Theta}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\Theta}_{N,1} & \tilde{\Theta}_{N,2} & \cdots & \tilde{\Theta}_{N,N} \end{pmatrix}, \quad (4.59)$$

where each of the  $2M \times 2M$  submatrices  $\tilde{\mathcal{G}}_{m,\ell} = R\mathcal{G}_{m,\ell}R$ ,  $\tilde{\mathcal{A}}_{m,\ell} = R\mathcal{A}_{m,\ell}R$ , and  $\tilde{\Theta}_{m,\ell} = R\Theta_{m,\ell}R$ .

The elements  $(\tilde{\mathcal{G}}_{n_1,n_2})_{m_1,m_2} = \left( (\mathcal{G}_{n_1,n_2})_{m_1,m_2} \right)_{i,j=1}^2$ ,  $(\tilde{\mathcal{A}}_{n_1,n_2})_{m_1,m_2} = \left( (\mathcal{A}_{n_1,n_2})_{m_1,m_2} \right)_{i,j=1}^2$

and

$(\tilde{\Theta}_{n_1,n_2})_{m_1,m_2} = \left( (\Theta_{n_1,n_2})_{m_1,m_2} \right)_{i,j=1}^2$  are  $2 \times 2$  arrays, where

$$(\tilde{\mathcal{G}}_{n_1,n_2})_{m_1,m_2} = R_{m_1} (\mathcal{G}_{n_1,n_2})_{m_1,m_2} R_{m_2}, \quad (\tilde{\mathcal{A}}_{n_1,n_2})_{m_1,m_2} = R_{m_1} (\mathcal{A}_{n_1,n_2})_{m_1,m_2} R_{m_2},$$

and

$$(\tilde{\Theta}_{n_1,n_2})_{m_1,m_2} = R_{m_1} (\Theta_{n_1,n_2})_{m_1,m_2} R_{m_2}, \quad m_1, m_2 = 1, \dots, M, \quad n_1, n_2 = 1, \dots, N. \quad (4.60)$$

Now, each of the submatrices  $\tilde{\mathcal{G}}_{n_1,n_2}$ ,  $\tilde{\mathcal{A}}_{n_1,n_2}$ ,  $n_1, n_2 = 1, \dots, N$ , has a block  $2 \times 2$  block circulant structure. Finally, the system (4.57) can be rewritten in the form

$$\left( \begin{array}{c|c} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{array} \right) \left( \begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{array} \right) = \left( \begin{array}{c|c} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \end{array} \right), \quad (4.61)$$

where the  $MN \times MN$  matrices  $\mathbf{G}_{ij}$ ,  $\mathbf{A}_{ij}$ ,  $\mathbf{Y}_{ij}$ ,  $i, j = 1, 2$ , are expressed in the form

$$\mathbf{G}_{ij} = \begin{pmatrix} G_{1,1}^{ij} & G_{1,2}^{ij} & \cdots & G_{1,N}^{ij} \\ G_{2,1}^{ij} & G_{2,2}^{ij} & \cdots & G_{2,N}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ G_{N,1}^{ij} & G_{N,2}^{ij} & \cdots & G_{N,N}^{ij} \end{pmatrix}, \quad \mathbf{A}_{ij} = \begin{pmatrix} A_{1,1}^{ij} & A_{1,2}^{ij} & \cdots & A_{1,N}^{ij} \\ A_{2,1}^{ij} & A_{2,2}^{ij} & \cdots & A_{2,N}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N,1}^{ij} & A_{N,2}^{ij} & \cdots & A_{N,N}^{ij} \end{pmatrix},$$

and

$$\Upsilon_{ij} = \begin{pmatrix} \Upsilon_{1,1}^{ij} & \Upsilon_{1,2}^{ij} & \cdots & \Upsilon_{1,N}^{ij} \\ \Upsilon_{2,1}^{ij} & \Upsilon_{2,2}^{ij} & \cdots & \Upsilon_{2,N}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \Upsilon_{N,1}^{ij} & \Upsilon_{N,2}^{ij} & \cdots & \Upsilon_{N,N}^{ij} \end{pmatrix}.$$

The matrices  $G_{ij}, \Upsilon_{ij}, i, j = 1, 2$ , are block circulant and therefore, so will the matrices  $A_{ij}, i, j = 1, 2$ , as stated in Theorem 3 in Chapter 3.

Hence, each of the  $M \times M$  submatrices  $G_{n_1, n_2}^{ij}, A_{n_1, n_2}^{ij}, \Upsilon_{n_1, n_2}^{ij}, i, j = 1, 2, n_1, n_2 = 1, \dots, N$ , is circulant and defined from

$$(G_{n_1, n_2}^{ij})_{m_1, m_2} = \left( (\tilde{G}_{n_1, n_2})_{m_1, m_2} \right)_{i, j}, \quad (A_{n_1, n_2}^{ij})_{m_1, m_2} = \left( (\tilde{A}_{n_1, n_2})_{m_1, m_2} \right)_{i, j},$$

and

$$(\Upsilon_{n_1, n_2}^{ij})_{m_1, m_2} = \left( (\tilde{\Theta}_{n_1, n_2})_{m_1, m_2} \right)_{i, j}, \quad m_1, m_2 = 1, \dots, M, \text{ respectively.}$$

In the case of the mixed Neumann/Dirichlet boundary problem, Other than the matrix  $G$  which is block circulant, none of the other matrices involved in system (4.52) is block circulant. The system may, however, be transformed into one with a block circulant structure by means of a simple transformation. The system (4.52) can first be rewritten as

$$\mathcal{G}\hat{\mathcal{B}} = \hat{\Theta}, \quad (4.62)$$

where the  $2MN \times 2MN$  matrix  $\mathcal{G}$  is defined in (4.54). The  $2MN \times 2M$  matrix  $\hat{\mathcal{B}}$  consists of  $N, 2M \times 2M$  submatrices  $\hat{\mathcal{B}}_n, n = 1, \dots, N$ , where each  $2 \times 2$  array

$$\left( \hat{\mathcal{B}}_n \right)_{m_1, m_2} = \left( \begin{array}{c|c} (A_n)_{m_1, m_2} & (B_n)_{m_1, m_2} \\ \hline (C_n)_{m_1, m_2} & (D_n)_{m_1, m_2} \end{array} \right), \quad m_1, m_2 = 1, \dots, M,$$

and the  $2MN \times 2M$  matrix  $\hat{\Theta}$  consists of  $N, 2M \times 2M$  submatrices  $\hat{\Theta}_n, n = 1, \dots, N$ , where each  $2 \times 2$  array

$$\left( \hat{\Theta}_n \right)_{m_1, m_2} = \left( \begin{array}{c|c} (\Phi_{11n})_{m_1, m_2} & (\Phi_{12n})_{m_1, m_2} \\ \hline (\Phi_{21n})_{m_1, m_2} & (\Phi_{22n})_{m_1, m_2} \end{array} \right).$$

Pre-multiplying and post-multiplying the system (4.62) by the  $2MN \times 2M$  matrices  $I_N \otimes R$  and  $R$ , respectively, yields

$$(I_N \otimes R)\mathcal{G}\hat{\mathcal{B}}R = (I_N \otimes R)\hat{\Theta}R, \quad (4.63)$$

or

$$\tilde{\mathcal{G}}\tilde{\mathcal{B}} = \tilde{\Theta}, \quad (4.64)$$

where  $\tilde{\mathcal{G}}$  is defined in (4.57) and

$$\tilde{\mathcal{B}} = (I_N \otimes R) \hat{\mathcal{B}}R, \quad \tilde{\mathcal{C}} = (I_N \otimes R) \hat{\mathcal{C}}R.$$

The  $2MN \times 2M$  matrices  $\tilde{\mathcal{B}}$  and  $\tilde{\mathcal{C}}$  can be written as

$$\tilde{\mathcal{B}} = \begin{pmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \\ \vdots \\ \mathcal{B}_N \end{pmatrix} \quad \text{and} \quad \tilde{\mathcal{C}} = \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_N \end{pmatrix} \quad (4.65)$$

where each of the  $2M \times 2M$  submatrices  $\mathcal{B}_n = R\hat{\mathcal{B}}_nR$  and  $\Theta_n = R\hat{\mathcal{C}}_nR$ . The elements  $(\mathcal{B}_n)_{m_1, m_2} = \left( (\mathcal{B}_n)_{m_1, m_2} \right)_{i, j=1}^2$  and  $(\Theta_n)_{m_1, m_2} = \left( (\Theta_n)_{m_1, m_2} \right)_{i, j=1}^2$  are  $2 \times 2$  arrays where

$$(\mathcal{B}_n)_{m_1, m_2} = R_{m_1} \left( \hat{\mathcal{B}}_{n_1, n_2} \right)_{m_1, m_2} R_{m_2},$$

$$(\Theta_n)_{m_1, m_2} = R_{m_1} \left( \hat{\mathcal{C}}_{n_1, n_2} \right)_{m_1, m_2} R_{m_2}, \quad m_1, m_2 = 1, \dots, M, \quad n = 1, \dots, N. \quad (4.66)$$

Each of the submatrices  $\tilde{\mathcal{B}}_{n_1, n_2}$ ,  $n_1, n_2 = 1, \dots, N$ , and  $\Theta_n$ ,  $n = 1, \dots, N$ , has a  $2 \times 2$  block circulant structure. Next, the system (4.64) is rewritten in the form

$$\left( \begin{array}{c|c} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \hline \mathbf{G}_{21} & \mathbf{G}_{22} \end{array} \right) \left( \begin{array}{c|c} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \hline \mathbf{B}_{21} & \mathbf{B}_{22} \end{array} \right) = \left( \begin{array}{c|c} \Lambda_{11} & \Lambda_{12} \\ \hline \Lambda_{21} & \Lambda_{22} \end{array} \right), \quad (4.67)$$

where the  $MN \times MN$  block circulant matrices  $\mathbf{G}_{ij}$  are defined in (4.61) while the  $MN \times M$  matrices  $\mathbf{B}_{ij}$  and  $\Lambda_{ij}$ ,  $i, j = 1, 2$ , are expressed in the form

$$\mathbf{B}_{ij} = \begin{pmatrix} \mathbf{B}_1^{ij} \\ \mathbf{B}_2^{ij} \\ \vdots \\ \mathbf{B}_N^{ij} \end{pmatrix} \quad \text{and} \quad \Lambda_{ij} = \begin{pmatrix} \Lambda_1^{ij} \\ \Lambda_2^{ij} \\ \vdots \\ \Lambda_N^{ij} \end{pmatrix}.$$

The  $\Lambda_{ij}$ ,  $i, j = 1, 2$ , are block circulant and therefore so will the matrices  $\mathbf{B}_{ij}$ ,  $i, j = 1, 2$ , as shown in the Corollary 2 in Chapter 3.

Hence, each of the  $M \times M$  submatrices  $\mathbf{B}_n^{ij}, \Lambda_n^{ij}$ ,  $i, j = 1, 2$ ,  $n = 1, \dots, N$ , is circulant and defined from

$$(\mathbf{B}_n^{ij})_{m_1, m_2} = \left( (\mathcal{B}_n)_{m_1, m_2} \right)_{i, j}, \quad \text{and} \quad (\Lambda_n^{ij})_{m_1, m_2} = \left( (\Theta_n)_{m_1, m_2} \right)_{i, j}, \text{ respectively.}$$

Finally, system (4.57) can be solved after an appropriate transformation, using a slight alteration of the MDA used in the Poisson and Biharmonic problems.

### 4.3 Matrix Decomposition Algorithm for Cauchy-Navier

By pre-multiplying system (4.61) by the matrix  $I_2 \otimes I_N \otimes U_M$  and post-multiplying it by the matrix  $I_2 \otimes I_N \otimes U_M^*$ , the result is

$$\begin{aligned} (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array} \right) \left( \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) (I_2 \otimes I_N \otimes U_M^*) \\ = (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} \Upsilon_{11} & \Upsilon_{12} \\ \hline \Upsilon_{21} & \Upsilon_{22} \end{array} \right) (I_2 \otimes I_N \otimes U_M^*), \end{aligned} \quad (4.68)$$

or

$$\left( \begin{array}{c|c} \hat{G}_{11} & \hat{G}_{12} \\ \hline \hat{G}_{21} & \hat{G}_{22} \end{array} \right) \left( \begin{array}{c|c} \hat{A}_{11} & \hat{A}_{12} \\ \hline \hat{A}_{21} & \hat{A}_{22} \end{array} \right) = \left( \begin{array}{c|c} \hat{\Upsilon}_{11} & \hat{\Upsilon}_{12} \\ \hline \hat{\Upsilon}_{21} & \hat{\Upsilon}_{22} \end{array} \right), \quad (4.69)$$

where

$$\begin{aligned} \left( \begin{array}{c|c} \hat{G}_{11} & \hat{G}_{12} \\ \hline \hat{G}_{21} & \hat{G}_{22} \end{array} \right) &= (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array} \right) (I_2 \otimes I_N \otimes U_M^*), \\ \left( \begin{array}{c|c} \hat{A}_{11} & \hat{A}_{12} \\ \hline \hat{A}_{21} & \hat{A}_{22} \end{array} \right) &= (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) (I_2 \otimes I_N \otimes U_M^*), \end{aligned}$$

and

$$\left( \begin{array}{c|c} \hat{\Upsilon}_{11} & \hat{\Upsilon}_{12} \\ \hline \hat{\Upsilon}_{21} & \hat{\Upsilon}_{22} \end{array} \right) = (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} \Upsilon_{11} & \Upsilon_{12} \\ \hline \Upsilon_{21} & \Upsilon_{22} \end{array} \right) (I_2 \otimes I_N \otimes U_M^*).$$

Because each of the submatrices  $G_{ij}$ ,  $A_{ij}$ , and  $\Upsilon_{ij}$ ,  $i, j = 1, 2$ , is block circulant, the matrices  $\hat{G}_{ij}$ ,  $\hat{A}_{ij}$ , and  $\hat{\Upsilon}_{ij}$  will be block diagonal. More precisely,

$$\hat{G}_{ij} = \begin{pmatrix} \hat{G}_{1,1}^{ij} & \hat{G}_{1,2}^{ij} & \cdots & \hat{G}_{1,N}^{ij} \\ \hat{G}_{2,1}^{ij} & \hat{G}_{2,2}^{ij} & \cdots & \hat{G}_{2,N}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{G}_{N,1}^{ij} & \hat{G}_{N,2}^{ij} & \cdots & \hat{G}_{N,N}^{ij} \end{pmatrix}, \quad \hat{A}_{ij} = \begin{pmatrix} \hat{A}_{1,1}^{ij} & \hat{A}_{1,2}^{ij} & \cdots & \hat{A}_{1,N}^{ij} \\ \hat{A}_{2,1}^{ij} & \hat{A}_{2,2}^{ij} & \cdots & \hat{A}_{2,N}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{A}_{N,1}^{ij} & \hat{A}_{N,2}^{ij} & \cdots & \hat{A}_{N,N}^{ij} \end{pmatrix},$$

and

$$\hat{\Upsilon}_{ij} = \begin{pmatrix} \hat{\Upsilon}_{1,1}^{ij} & \hat{\Upsilon}_{1,2}^{ij} & \cdots & \hat{\Upsilon}_{1,N}^{ij} \\ \hat{\Upsilon}_{2,1}^{ij} & \hat{\Upsilon}_{2,2}^{ij} & \cdots & \hat{\Upsilon}_{2,N}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Upsilon}_{N,1}^{ij} & \hat{\Upsilon}_{N,2}^{ij} & \cdots & \hat{\Upsilon}_{N,N}^{ij} \end{pmatrix},$$

where each of the  $M \times M$  matrices  $\hat{G}_{n_1, n_2}^{ij}$ ,  $\hat{A}_{n_1, n_2}^{ij}$ , and  $\hat{\Upsilon}_{n_1, n_2}^{ij}$ ,  $n_1, n_2 = 1, \dots, N$ , is diagonal. The solution of system (4.69) can thus be decomposed into solving the  $M$  systems of order  $2N$

$$\left( \begin{array}{c|c} G_{11}^m & G_{12}^m \\ \hline G_{21}^m & G_{22}^m \end{array} \right) \left( \begin{array}{c|c} A_{11}^m & A_{12}^m \\ \hline A_{21}^m & A_{22}^m \end{array} \right) = \left( \begin{array}{c|c} \Phi_{11}^m & \Phi_{12}^m \\ \hline \Phi_{21}^m & \Phi_{22}^m \end{array} \right), \quad m = 1, \dots, M, \quad (4.70)$$

where

$$(G_{ij}^m)_{n_1, n_2} = \hat{G}_{n_1, n_2}^{ij}, (A_{ij}^m)_{n_1, n_2} = \hat{A}_{n_1, n_2}^{ij}, (\Phi_{ij}^m)_{n_1, n_2} = \hat{Y}_{n_1, n_2}^{ij}, n_1, n_2 = 1, \dots, N,$$

and  $\hat{G}_{n_1, n_2}^{ij}, \hat{A}_{n_1, n_2}^{ij}, \hat{Y}_{n_1, n_2}^{ij}, m = 1, \dots, M$ , are the diagonal elements of the matrices  $\hat{G}_{n_1, n_2}^{ij}, \hat{A}_{n_1, n_2}^{ij}, \hat{Y}_{n_1, n_2}^{ij}$ , respectively.

In the case of the Neumann/Dirichlet boundary problem, pre-multiplying system (4.67) by the matrix  $I_2 \otimes I_N \otimes U_M$  and post-multiplying it by the matrix  $I_2 \otimes U_M^*$  yields

$$\begin{aligned} (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array} \right) \left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right) (I_2 \otimes U_M^*) \\ = (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} \Lambda_{11} & \Lambda_{12} \\ \hline \Lambda_{21} & \Lambda_{22} \end{array} \right) (I_2 \otimes U_M^*), \end{aligned} \quad (4.71)$$

or

$$\left( \begin{array}{c|c} \hat{G}_{11} & \hat{G}_{12} \\ \hline \hat{G}_{21} & \hat{G}_{22} \end{array} \right) \left( \begin{array}{c|c} \hat{B}_{11} & \hat{B}_{12} \\ \hline \hat{B}_{21} & \hat{B}_{22} \end{array} \right) = \left( \begin{array}{c|c} \hat{\Lambda}_{11} & \hat{\Lambda}_{12} \\ \hline \hat{\Lambda}_{21} & \hat{\Lambda}_{22} \end{array} \right), \quad (4.72)$$

where

$$\left( \begin{array}{c|c} \hat{B}_{11} & \hat{B}_{12} \\ \hline \hat{B}_{21} & \hat{B}_{22} \end{array} \right) = (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right) (I_2 \otimes U_M^*),$$

and

$$\left( \begin{array}{c|c} \hat{\Lambda}_{11} & \hat{\Lambda}_{12} \\ \hline \hat{\Lambda}_{21} & \hat{\Lambda}_{22} \end{array} \right) = (I_2 \otimes I_N \otimes U_M) \left( \begin{array}{c|c} \Lambda_{11} & \Lambda_{12} \\ \hline \Lambda_{21} & \Lambda_{22} \end{array} \right) (I_2 \otimes U_M^*).$$

Because each of the submatrices  $B_{ij}$  and  $\Lambda_{ij}, i, j = 1, 2$ , is block circulant, the matrices  $\hat{B}_{ij}$  and  $\hat{\Lambda}_{ij}$  will be block diagonal. More precisely,

$$\hat{B}_{ij} = \begin{pmatrix} \hat{B}_1^{ij} \\ \hat{B}_2^{ij} \\ \vdots \\ \hat{B}_N^{ij} \end{pmatrix} \quad \text{and} \quad \hat{\Lambda}_{ij} = \begin{pmatrix} \hat{\Lambda}_1^{ij} \\ \hat{\Lambda}_2^{ij} \\ \vdots \\ \hat{\Lambda}_N^{ij} \end{pmatrix},$$

where each of the  $M \times M$  matrices  $\hat{B}_n^{ij}$  and  $\hat{\Lambda}_n^{ij}, n = 1, \dots, N$ , is diagonal. The solution of system (4.72) can thus be decomposed into solving the  $M$  systems of order  $2N$

$$\left( \begin{array}{c|c} G_{11}^m & G_{12}^m \\ \hline G_{21}^m & G_{22}^m \end{array} \right) \left( \begin{array}{c|c} \hat{B}_{11}^m & \hat{B}_{12}^m \\ \hline \hat{B}_{21}^m & \hat{B}_{22}^m \end{array} \right) = \left( \begin{array}{c|c} \tilde{\Lambda}_{11}^m & \tilde{\Lambda}_{12}^m \\ \hline \tilde{\Lambda}_{21}^m & \tilde{\Lambda}_{22}^m \end{array} \right), \quad m = 1, \dots, M, \quad (4.73)$$

where the coefficient matrix is the same as in (4.70) and

$$(\hat{B}_{ij}^m)_n = \hat{B}_{n_m}^{ij}, (\tilde{\Lambda}_{ij}^m)_n = \hat{\Lambda}_{n_m}^{ij}, n_1, n_2 = 1, \dots, N,$$



and  $\hat{B}_{n_m}^{ij}, \hat{\Lambda}_{n_m}^{ij}$ ,  $m = 1, \dots, M$ , are the diagonal elements of the matrices  $\hat{B}_n^{ij}, \hat{\Lambda}_n^{ij}$ , respectively. By appropriately arranging the matrices  $\left( \begin{array}{c|c} \hat{B}_{11} & \hat{B}_{12} \\ \hat{B}_{21} & \hat{B}_{22} \end{array} \right)$  (for the Neumann boundary condition) and  $\left( \begin{array}{c|c} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{array} \right)$  (for the differential equation), the matrix  $\hat{C}_2$  can be obtained to be used in the following system

$$\left( \begin{array}{c|c} \hat{C}_{11} & \hat{C}_{12} \\ \hat{C}_{21} & \hat{C}_{22} \end{array} \right) \begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \end{pmatrix} = \begin{pmatrix} \hat{f}_1 \\ \hat{f}_2 \end{pmatrix}, \quad (4.74)$$

where  $\hat{u}_i = (I_2 \otimes I_N \otimes U_M)(I_N \otimes R)u_i$  and  $\hat{f}_i = (I_2 \otimes I_N \otimes U_M)(I_N \otimes R)f_i$ ,  $i = 1, 2$ . System (4.74) is sparse block diagonal and may be solved efficiently to yield  $\hat{u}_i$ . Finally,  $u_i$  is obtained from  $u_i = (I_N \otimes R)(I_2 \otimes I_N \otimes U_M)\hat{u}_i$ ,  $i = 1, 2$ .

## 4.4 LRBF-DQ MDA

### 4.4.1 Poisson

In the LRBF-DQ method for the solution of the boundary value problem (4.1), for each point  $(x_i, y_i)$ ,  $i = 1, \dots, MN$ , in  $\Omega$  and for each RBF  $\varphi_k$ ,  $k = 1, \dots, n$ ,

$$\Delta \varphi_k(x_i, y_i) = \sum_{j=1}^n a_{ij} \varphi_k(x_j, y_j), \quad (4.75)$$

where  $n$  is the number of points in the local domain  $\Omega_i$ , or in vector form,

$$\Delta \varphi = G_i a_i, \quad (4.76)$$

where  $G_i = \begin{pmatrix} \varphi_1(x_1) & \varphi_1(x_2) & \cdots & \varphi_1(x_n) \\ \varphi_2(x_1) & \varphi_2(x_2) & \cdots & \varphi_2(x_n) \\ \vdots & \vdots & & \vdots \\ \varphi_n(x_1) & \varphi_n(x_2) & \cdots & \varphi_n(x_n) \end{pmatrix}$  is unique to each local domain  $\Omega_i$ .

After solving for the coefficients  $a_i = G_i^{-1} \Delta \varphi$ , this can be used to construct the global sparse coefficient matrix  $A$ .

In the case of the Dirichlet boundary condition,

$$\begin{aligned} u_i &= g_1(x_i, y_i), i = 1, \dots, M \\ u_i &= g_2(x_i, y_i), i = (M-1)N \dots MN, \end{aligned} \quad (4.77)$$

which means that the lines in the global matrix corresponding to these equations will consist of only one non-zero element (1) in the global position of the corresponding boundary point.

In the case of the Neumann/Dirichlet boundary condition, consider for each point  $(x_i, y_i), i = 1, \dots, M$  on the boundary  $\partial\Omega_1$ , and for each RBF  $\phi_k, k = 1, \dots, n$ ,

$$\frac{\partial \phi_k}{\partial n}(x_i, y_i) = \sum_{j=1}^n b_{ij} \phi_k(x_j, y_j), \quad (4.78)$$

or in vector form, for each boundary point  $(x_i, y_i), i = 1, \dots, M$  on  $\partial\Omega_1$ ,

$$\frac{\partial \phi^i}{\partial n} = G_i b_i \quad (4.79)$$

Once the coefficients  $b_i$  have been computed, they can be placed appropriately in the global coefficient matrix  $A$ .

In general, the assembly of all equations for the  $MN$  centers  $x_i$ , yields an  $MN \times MN$  system of the form

$$Au = b, \quad (4.80)$$

where the matrix  $A$  is sparse. The right hand side vector  $b = [b_1, b_2, \dots, b_n]^T$  is defined as:

$$\begin{aligned} b_i &= g_D \text{ or } g_N, \quad i = 1, \dots, M \\ b_i &= f(x_i), \quad i = M + 1 \dots, (M - 1)N + 1, \\ b_i &= g_D, \quad i = (M - 1)N + 1, \dots, MN. \end{aligned} \quad (4.81)$$

With the distribution of collocation points being placed on concentric circles as described in section 3.1.2,  $A$  has the special structure

$$Au = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,N} \\ \vdots & \vdots & & \vdots \\ A_{N,1} & A_{N,2} & \cdots & A_{N,N} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} = b. \quad (4.82)$$

On each concentric circle  $k$ , where  $k = 1, \dots, N$ , there are  $M$  centers. The local system (4.75) is the same for each of the  $M$  centers. This implies that for each center  $(x_i, y_i), i = 1, \dots, M$ , on a concentric circle  $k$ , the coefficients  $\{a^i\}_{i=1}^M$  will be the same. Hence, the coefficients relating each center  $(x_i, y_i)$  with its  $n$  neighboring points in the local domain  $\Omega_i$  are the same for each center on the circle  $k$ . When transferring this relation to the global matrix  $A$ , the submatrix  $A_{kj}$  resulting from each of the  $M$  points on the circle  $k$  to the  $M$  points on any circle  $j = 1, \dots, N$  will be circulant due to the fact the sets of neighboring points for each point  $(x_i, y_i)$  on the circle  $k$  are globally circulant. Therefore, matrix  $A$  will be block circulant consisting of sparse blocks  $A_{n_1, n_2}, n_1, n_2 = 1, \dots, N$  [12]. Figure 4.4.1 presents a typical distribution of collocation points. Two consecutive centers are represented

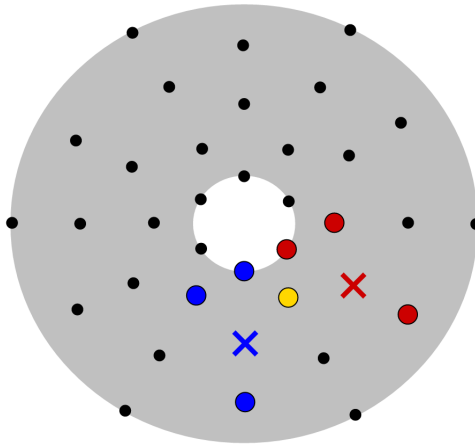


Figure 4.1: Distribution of collocation points in local domains. The four neighboring points for the two centers (X) are highlighted.

by an X with the local domain of the four nearest neighbors highlighted in blue and red, while the point belonging to both sets are highlighted in yellow. A similar argument can be made for the boundary points where the Neumann conditions are applied, while the Dirichlet case is trivial. Due to this block circulant structure, (4.82) can be solved using an appropriate matrix decomposition algorithm that takes advantage of the sparsity of the matrices  $A_{n_1, n_2=1, \dots, N}$ .

Let  $U$  continue to denote the unitary  $M \times M$  Fourier matrix and  $I_N$  the  $N \times N$  identity matrix.  $A$  is the sparse coefficient matrix that was calculated from each local domain  $\Omega_i$ . Multiplying system (4.82) by  $I_N \otimes U_M$  results in

$$(I_N \otimes U_M)Au = (I_N \otimes U_M)f \quad (4.83)$$

or written more condensely,

$$\hat{A}\hat{u} = \hat{f} \quad (4.84)$$

where

$$\hat{A} = (I_N \otimes U_M)A(I_N \otimes U_M^*) = \begin{pmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,N} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,N} \\ \vdots & \vdots & & \vdots \\ D_{N,1} & D_{N,2} & \cdots & D_{N,N} \end{pmatrix}, \quad (4.85)$$

$$\hat{u} = (I_N \otimes U_M)u, \quad (4.86)$$

and

$$\hat{f} = (I_N \otimes U_M)f. \quad (4.87)$$

System (4.84) is sparse block diagonal and may be solved efficiently to yield  $\hat{u}$ . Once  $\hat{u}$  has been calculated,  $u$  can be obtained from  $u = (I_N \otimes U_M^*)\hat{u}$ .

#### 4.4.2 Biharmonic

The same approach for the local approximation of the biharmonic problem can be used as in the Poisson case. In the LRBF-DQ method for the solution of the boundary value problem (4.25), for each point  $(x_i, y_i), i = 1, \dots, MN$ , in  $\Omega$  and for each RBF  $\varphi_k, k = 1, \dots, n$ ,

$$\Delta^2 \varphi_k(x_i, y_i) = \sum_{j=1}^n a_{ij} \varphi_k(x_j, y_j), \quad (4.88)$$

where  $n$  is the number of points in the local domain  $\Omega_i$ , or in vector form,

$$\Delta^2 \varphi = G_i a_i. \quad (4.89)$$

After solving for the coefficients  $a_i = G_i^{-1} \Delta \varphi$ , this can be used to construct the global sparse coefficient matrix  $A$ .

In the case of the first biharmonic problem, each boundary point  $(x_i, y_i), i = 1, \dots, M$ , on the boundary  $\partial\Omega_1$ , and for each RBF  $\varphi_k, k = 1, \dots, n$ ,

$$\frac{\partial \varphi_k}{\partial n} = \sum_{j=1}^n b_{ij} \varphi_k(x_j, y_j), \quad (4.90)$$

or in vector form,

$$\frac{\partial \varphi^i}{\partial n} = G_i b_i. \quad (4.91)$$

Once the coefficients  $b_i$  have been computed, they can be placed appropriately in the global sparse coefficient matrix  $A$ .

In the case of the second biharmonic problem we take, instead of (4.90), for each boundary point  $(x_i, y_i), i = 1, \dots, M$ , on the boundary  $\partial\Omega_1$ , and for each RBF  $\varphi_k, k = 1, \dots, n$ ,

$$\Delta \varphi_k(x_i, y_i) = \sum_{j=1}^n b_{ij} \varphi_k(x_j, y_j), \quad (4.92)$$

which in vector form gives

$$\varphi_{\Delta}^i = G_i b_i. \quad (4.93)$$

As in the first biharmonic problem, the coefficients  $b_i$  can be put in the global coefficient matrix  $A$ . In both biharmonic problems, the assembly of all equations for the  $MN$  centers  $(x_i, y_i)$ , yields an  $MN \times MN$  system of the form (4.80). Having the same circulant structure as in the Poisson case, the system can be solved using the MDA as explained in the previous section.

#### 4.4.3 Cauchy Navier

In the LRBF-DQ method for the solution of the boundary value problem (4.45), for each point  $(x_i, y_i)$ ,  $i = 1, \dots, MN$ , in  $\Omega$  and for each RBF  $\varphi_k$ ,  $k = 1, \dots, n$ ,

$$\begin{aligned}\mathcal{L}_{11}\varphi_k(x_i, y_i) &= \sum_{j=1}^n a_{kl}^i \varphi_k(x_j, y_j), & \mathcal{L}_{12}\varphi_k(x_i, y_i) &= \sum_{j=1}^n b_{kl}^i \varphi_k(x_j, y_j), \\ \mathcal{L}_{21}\varphi_k(x_i, y_i) &= \sum_{j=1}^n c_{kl}^{ij} \varphi_k(x_j, y_j), & \mathcal{L}_{22}\varphi_k(x_i, y_i) &= \sum_{j=1}^n d_{kl}^{ij} \varphi_k(x_j, y_j),\end{aligned}$$

which in vector form gives for each point  $(x_i, y_i)$ ,

$$\varphi_{\mathcal{L}_{11}}^i = G_i a_i, \quad \varphi_{\mathcal{L}_{12}}^i = G_i b_i, \quad \varphi_{\mathcal{L}_{21}}^i = G_i c_i, \quad \varphi_{\mathcal{L}_{22}}^i = G_i d_i. \quad (4.94)$$

After solving for the coefficients,  $a_i, b_i, c_i, d_i$ , this can be used to construct the global sparse coefficient matrix  $A = \left( \begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right)$

The Dirichlet boundary condition is applied separately for  $u_1$  and  $u_2$  as in (4.77). The Neumann boundary condition is applied as follows

$$\begin{aligned}\mathcal{T}_{11}\varphi(x_i, y_i) &= \left[ 2\mu \left( \frac{1-\nu}{1-2\nu} \right) \frac{\partial}{\partial x} n_x + \mu \frac{\partial}{\partial y} n_y \right] \varphi(x_i, y_i) = \sum_{j=1}^n e_{ij} \varphi(x_j, y_j), \\ \mathcal{T}_{12}\varphi(x_i, y_i) &= \left[ 2\mu \left( \frac{\nu}{1-2\nu} \right) \frac{\partial}{\partial y} n_x + \mu \frac{\partial}{\partial x} n_y \right] \varphi(x_i, y_i) = \sum_{j=1}^n f_{ij} \varphi(x_j, y_j), \\ \mathcal{T}_{21}\varphi(x_i, y_i) &= \left[ \mu \frac{\partial}{\partial y} n_x + 2\mu \left( \frac{\nu}{1-2\nu} \right) \frac{\partial}{\partial x} n_y \right] \varphi(x_i, y_i) = \sum_{j=1}^n g_{ij} \varphi(x_j, y_j), \\ \mathcal{T}_{22}\varphi(x_i, y_i) &= \left[ \mu \frac{\partial}{\partial x} n_x + 2\mu \left( \frac{1-\nu}{1-2\nu} \right) \frac{\partial}{\partial y} n_y \right] \varphi(x_i, y_i) = \sum_{j=1}^n h_{ij} \varphi(x_j, y_j),\end{aligned}$$

where  $n$  is the number of points in the local domain  $\Omega_i$ , or in vector form,

$$\varphi_{\mathcal{T}_{11}}^i = G_i e_i, \quad \varphi_{\mathcal{T}_{12}}^i = G_i f_i, \quad \varphi_{\mathcal{T}_{21}}^i = G_i g_i, \quad \varphi_{\mathcal{T}_{22}}^i = G_i h_i. \quad (4.95)$$

Once the coefficients  $e_i, f_i, g_i, h_i$  have been computed, they can be placed appropriately in the global coefficient matrix  $A$ . Assembling all equations for the  $MN$  centers  $(x_i, y_i)$ , yields

a  $2MN \times 2MN$  system of the form (4.80). While the matrix  $A$  is sparse, it is not block circulant. However, a transformation is possible for the local case as explained in section 4.2. Once  $A$  is transformed to a block circulant matrix, the system can be solved using the MDA as explained in section 4.3.

## Chapter 5

### NUMERICAL RESULTS

In this chapter, numerical results obtained from the Radial Basis Function Differential Quadrature method (RBF-DQ) using the Matrix Decomposition Algorithm (MDA) are presented. This method approximates the particular solution for large scale elliptic partial differential equations (PDEs). The normal multiquadric (NMQ) has achieved excellent results in the RBF literature. Other notable RBFs include the Inverse Multiquadric (IMQ) and Gaussian. The accuracy of the method is dependent upon the selection of the shape parameter for each of the RBFs. For the global RBF-DQ method, the adjusted Fausshauer estimate will be applied for the selection of the shape parameter. Experiments will also be considered using the well known leave-one-out cross validation (LOOCV) method, a method described in Chapter 2. As the number of collocation points increases, LOOCV becomes computationally expensive, thus a modification is required. Instead of using the entire set of  $MN$  collocation points, LOOCV will be applied to the set of  $M$  collocation points on a randomly selected concentric circle. This will provide satisfactory results given the nature of the point distribution. For the local RBF-DQ method, LOOCV is also used to determine a suitable shape parameter. Instead of applying LOOCV to each local domain, it can be applied to only one local domain to calculate the shape parameter, and this value will be used for all local domains.

Experiments are also conducted using the dimensionless shape parameter, where the proposed value of  $c = MN/1000$  is used, and is compared with the value found using LOOCV. Several problems with known exact solutions are used to verify the effectiveness of the proposed method including the Poisson, Biharmonic, and Cauchy-Navier equations. While implementing the numerical schemes, an annulus is used for the domain. The computations were carried out using MATLAB on a Macbook Pro laptop with 2.8 GHz Intel Core i7, 16 GB memory, in OS X 10.11.6 for up to  $M = N = 700$  in the global method. For larger sets of collocation points, it was necessary to use a workstation with a larger memory storage of 256GB. Using this workstation, a problem using up to one million points ( $M = N = 1000$ ) was successfully solved. The local method, however, is able to handle up to one million points without the need of using the larger workstation.

To validate the numerical accuracy, the maximum relative error  $E$  is calculated and is

defined as

$$E = \frac{\|u - u_N\|_{\infty, \bar{\Omega}}}{\|u\|_{\infty, \bar{\Omega}}}, \quad (5.1)$$

where  $u$  and  $u_N$  denote the exact solution and the approximate solution, respectively.

### 5.1 Poisson Equation RBF-DQ

Consider the Poisson problem

$$\Delta u = 5e^{2x+y} \quad \text{in } \Omega, \quad (5.2a)$$

$$u = e^{2x+y} \quad \text{on } \partial\Omega_1, \partial\Omega_2. \quad (5.2b)$$

with the domain  $\Omega$  defined by  $\gamma_1 = 0.3$  and  $\gamma_2 = 1$ . The boundary conditions correspond to the exact solution which is given by  $u = e^{2x+y}$ . The profile of the exact solution is shown in Figure 5.1. The profile of the relative error is shown in Figure 5.2. In Figures 5.3-5.5 the

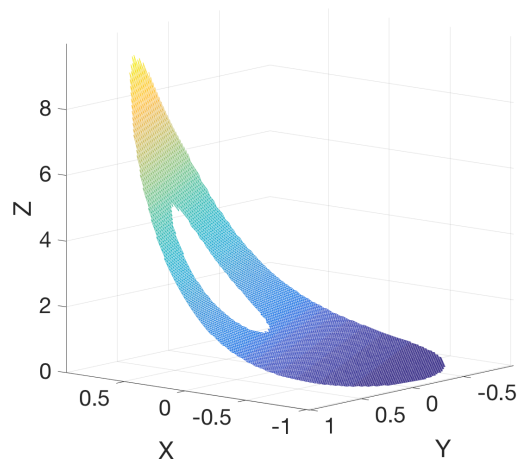


Figure 5.1: Example 1: Profile of the Exact Solution.

maximum relative error  $E$  is plotted versus the shape parameter  $c$  for  $M = N = 100$  for each of the MQ, IMQ, and Gaussian RBFs. In these figures, it can be observed that all three RBFs behave similarly for values up to  $c = 4$ . Beyond this value, the accuracy for IMQ begins to deteriorate while the accuracy for MQ continues to improve up to  $c = 5$  before deteriorating. While the Gaussian RBF displays less stability, the accuracy appears to be fairly steady. In fact, this trend continues through  $c = 100$  as shown in Figure 5.6.

Table 5.1 displays the optimal shape parameter with its corresponding maximum relative error for each of the three RBFs. Excellent results are obtained from each of the three tested



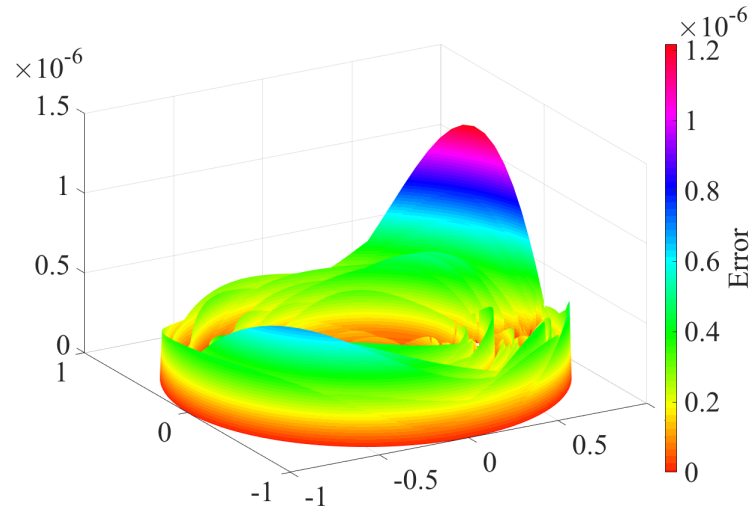


Figure 5.2: Example 1: Profile of Relative Error.

RBFs. Perhaps an advantage to the Gaussian RBF is that excellent results can be obtained over a wide range of values for the shape parameter. This comes with the disadvantage that it is less stable.

RBF	$c$	E
MQ	4.987	1.045(-7)
IMQ	3.963	1.732(-7)
Gaussian	90.188	9.200(-8)

Table 5.1: Example 1, Optimal Shape Parameter  $c$  and Max Rel Error for MQ, IMQ, and Gaussian RBFs

It is not practical to use the trial and error method for approximating the solution unless the exact solution is already known. This, of course, would not provide any significant applications. Therefore, the methods for finding a suitable shape parameter presented in Chapter 2 must be considered. The adjusted Fasshauer estimate for the shape parameter  $c = \sqrt[4]{MN}/m_x$  will first be explored. Recall that  $m_x$  is a parameter that is dependent on the density of the collocation points. As the number of collocation points increases, the appropriate value of  $m_x$  will decrease. Using  $m_x = 2.0$ ,  $c = 5.000$  is obtained with a corresponding maximum relative error of  $1.300(-7)$ , which is close to the optimal shape parameter achieving similar accuracy. In Figure 5.7, the relationship between  $m_x$  and the maximum relative error  $E$  is shown for  $M = N = 100, 200$ , and  $300$ . It can be observed that  $E$  begins to deteriorate beyond the values of  $m_x = 2.0, 1.4$ , and  $0.9$  for  $M = N = 100, 200$ , and  $300$ , respectively.

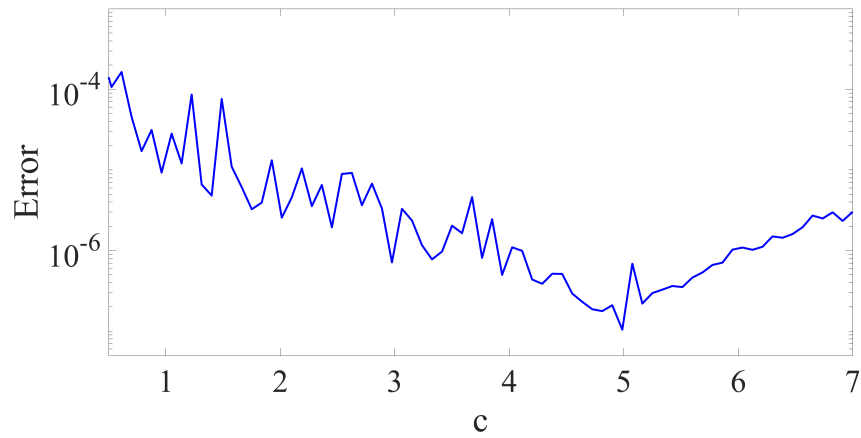


Figure 5.3: Example 1, Multiquadric (MQ): Maximum relative error vs. shape parameter,  $M = 100, N = 100$ .

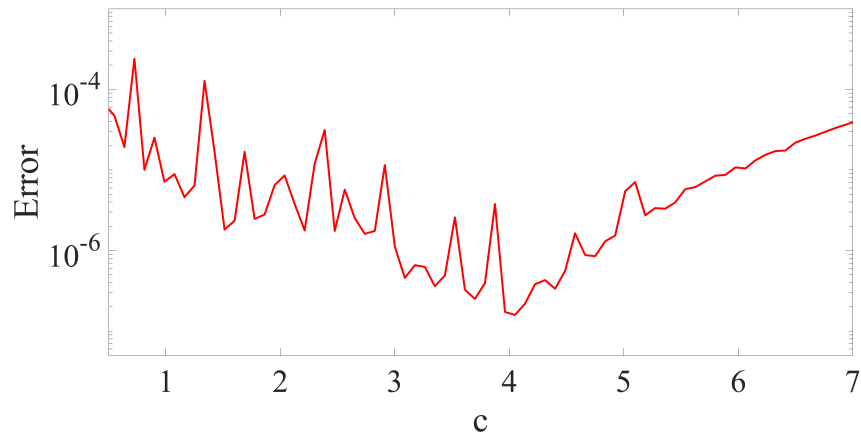


Figure 5.4: Example 1, Inverse Multiquadric (IMQ): Maximum relative error vs. shape parameter,  $M = 100, N = 100$ .

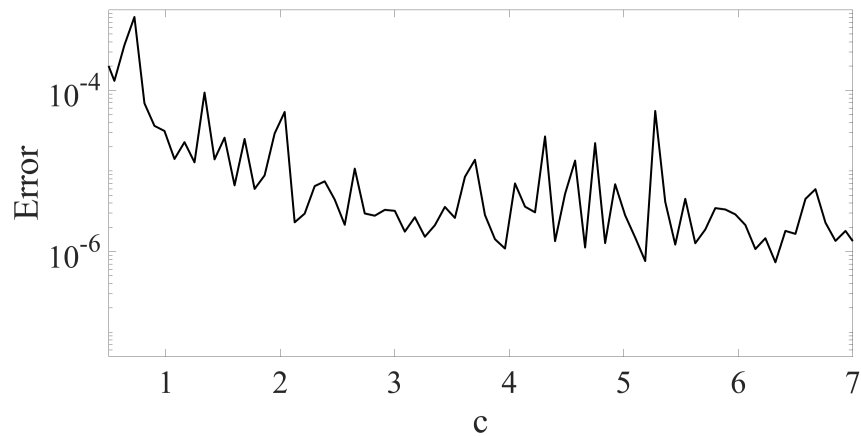


Figure 5.5: Example 1, Gaussian (GA): Maximum relative error vs. shape parameter,  $M = 100, N = 100$ .

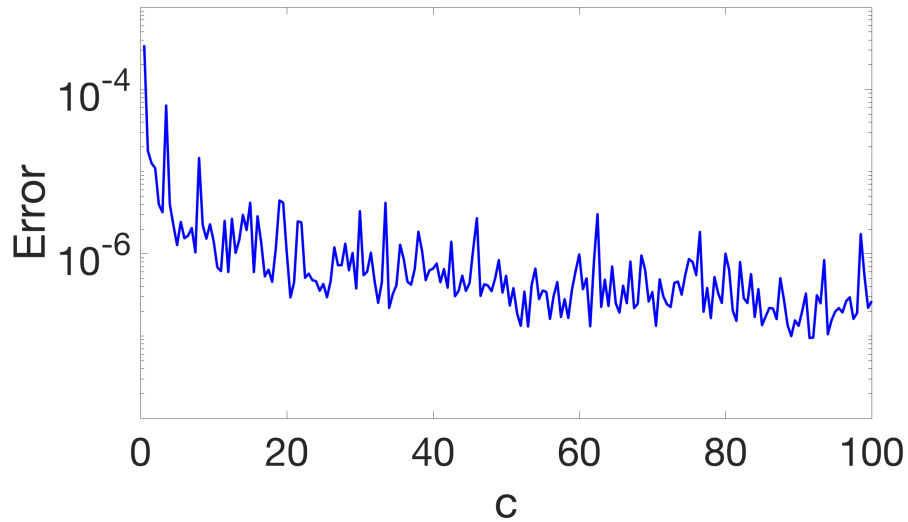


Figure 5.6: Example 1, Gaussian (GA): Maximum relative error versus shape parameter with  $M = 100, N = 100$ .

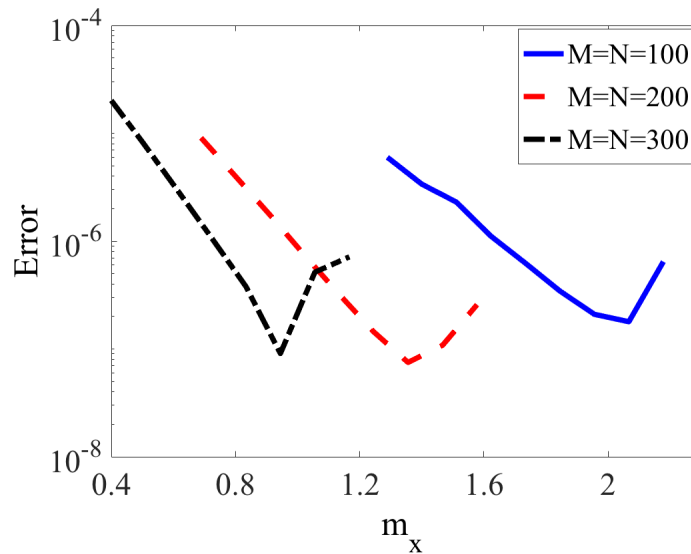


Figure 5.7: Example 1, Poisson Dirichlet problem: Maximum relative error versus  $m_x$  using MQ.

Using the above selection procedure, the values of  $m_x$  have been chosen as shown in Table 5.2 for  $M = N$  values up to 700 for the Dirichlet boundary problem (4.1a-4.1c) with the MQ RBF.

To demonstrate the accuracy up to one million points ( $M = 1000, N = 1000$ ), Table 5.3 shows the results using the workstation with 256 GB memory.

$M = N$	$m_x$	E	CPU (s)	sub-optimal $c$
100	2.0	1.300(-7)	0.261	5.000
200	1.4	2.903(-7)	1.728	10.102
300	0.9	2.075(-7)	6.098	19.245
400	0.8	1.503(-7)	14.877	25.000
500	0.7	2.931(-7)	33.969	31.944
600	0.6	1.153(-7)	53.190	40.825
700	0.5	1.956(-7)	107.654	52.915

Table 5.2: Example 1, Poisson Dirichlet Problem: Max Rel Error and shape parameter using MQ.

$M = N$	$m_x$	E	CPU (s)	sub-optimal $c$
600	0.6	7.754(-7)	82.050	40.825
700	0.5	2.204(-7)	127.867	52.915
800	0.5	4.781(-7)	174.710	56.569
900	0.4	3.322(-7)	251.210	75.000
1000	0.4	4.780(-7)	338.773	79.057

Table 5.3: Example 1, Poisson Dirichlet problem: Max Rel Error and shape parameter using MQ up to 1 million points

It is also of interest to compare these results with the Gaussian RBF. As stated previously, the accuracy has a wide range of acceptable shape parameters. It is less critical to choose a proper shape parameter in this case. In Table 5.4, a value of  $c = 90$  is a conveniently chosen value that is near the optimal value as shown in Figure 5.6. The Gaussian RBF is competitive with the MQ's adjust Fausshauer estimate. While it does not quite reach the same accuracy, it is able to achieve this using the same shape parameter for the different number of collocation points.

$M = N$	E
100	1.327(-7)
200	6.512(-7)
300	8.579(-7)
400	8.755(-7)
500	6.093(-7)
600	8.563(-7)
700	5.810(-7)

Table 5.4: Example 1, Poisson Dirichlet problem: Max Rel Error with  $c = 90$  using Gaussian RBF.

Table 5.5 shows results for the Dirichlet/Neumann boundary problem (5.2) replacing (5.2b) with  $\frac{\partial u}{\partial \mathbf{n}} = \nabla e^{2x+y} \cdot \mathbf{n}$  on the boundary  $\partial\Omega_1$  for  $M = N$  up to 700 using MQ.

$M = N$	$m_x$	E	CPU (s)	sub-optimal $c$
100	2.0	1.462(-7)	0.263	5.000
200	1.4	1.447(-7)	1.730	10.102
300	0.9	2.040(-7)	6.075	19.245
400	0.8	3.096(-7)	14.970	25.000
500	0.7	1.236(-7)	33.989	31.944
600	0.6	4.919(-7)	53.988	40.825
700	0.5	4.966(-7)	109.273	52.915

Table 5.5: Example 1, Poisson Neumann problem: Max Rel Error and shape parameter using MQ RBF.

It is also noteworthy to compare the MDA with the full RBF-DQ method in which the full system (4.1) is solved to fully realize the computational savings. In Table 5.6, the errors,  $E$ , and CPU times for the full RBF-DQ method and the corresponding values using the proposed RBF-DQ MDA for a range of collocation points using LOOCV to estimate the optimal shape parameter is displayed. While LOOCV is a valid and well known method, the most time consuming part in both approaches is the search of a suitable shape parameter using LOOCV, in which systems are repeatedly solved. Taking advantage of the circulant structure and the distribution of the collocation points, instead of using all of the points in the domain, we can take a subset of points. Choose for this subset the points on a randomly selected concentric circle, and use LOOCV with an initial search interval  $[0, 8]$  to calculate the suboptimal shape parameter. This approach is used for both the full RBF-DQ method and the RBF-DQ MDA method with the MQ RBF. Although the LOOCV is using a minimum amount of required computational time, it can be observed that the full RBF-DQ method quickly becomes burdened by the obligation of solving the full matrix. Time and memory become so large, calculations were not able to be recorded beyond  $M = N = 120$  for the full RBF-DQ method. This clearly demonstrates the advantage and usefulness that the MDA algorithm is able to offer large scale problems.

## 5.2 Biharmonic Equation RBF-DQ

Consider the First Biharmonic problem

$$\Delta^2 u = 25e^{2x+y} \quad \text{in } \Omega, \quad (5.3a)$$

$$u = e^{2x+y} \quad \text{and} \quad \frac{\partial u}{\partial \mathbf{n}} = \nabla e^{2x+y} \cdot \mathbf{n} \quad \text{on } \partial\Omega_1, \partial\Omega_2, \quad (5.3b)$$

$M = N$	Full RBF-DQ		RBF-DQ MDA	
	E	CPU (s)	E	CPU (s)
40	4.927(-4)	0.430	4.925(-4)	0.1484
60	1.124(-4)	2.456	1.124(-4)	0.180
80	4.852(-6)	10.808	3.845(-6)	0.293
100	7.659(-6)	36.489	2.688(-6)	0.400
120	1.071(-5)	108.471	5.553(-7)	0.620
150	--	--	8.497(-7)	0.981

Table 5.6: Example 1, Poisson Dirichlet problem: Comparison of Errors, E, and CPU times for full RBF-DQ and RBF-DQ MDA solutions for various  $M = N$  with initial search interval  $[0, 8]$ .

with the annulus domain  $\Omega$  defined by  $\gamma_1 = 0.3$  and  $\gamma_2 = 1$ . In this example  $\alpha_n = (-1)^n/5, n = 1, \dots, N$ , is used to define the collocation points in order to avoid possible singularities as detailed in Theorem 4 of Chapter 3. The profile of the relative error is shown in Figure 5.8. In Figure 5.9 the maximum relative error versus the shape parameter for

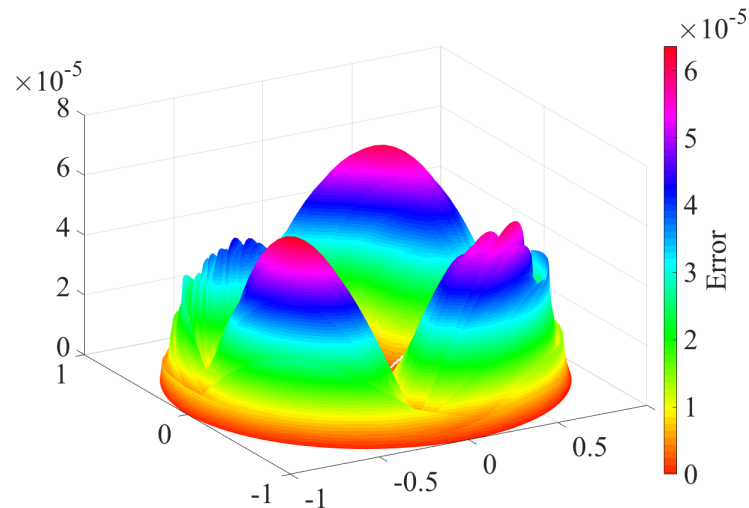


Figure 5.8: Example 2: Profile of Relative Error.

$M = 100, N = 100$  is plotted for the first biharmonic problem using MQ.

In Table 5.7, the maximum relative error for values of  $M = N$  up to 700 is shown for the first biharmonic problem. The numerical results is consistent with that in Table 5.2.

Since the biharmonic problem is a fourth order differential equation, it is more challenging to solve. Thus, the results do not reach the same accuracy as that of the Poisson problem. However, the accuracy for the biharmonic problem is still able to maintain excellent results. While the Biharmonic problem follows a similar pattern for the shape parameter, in that as

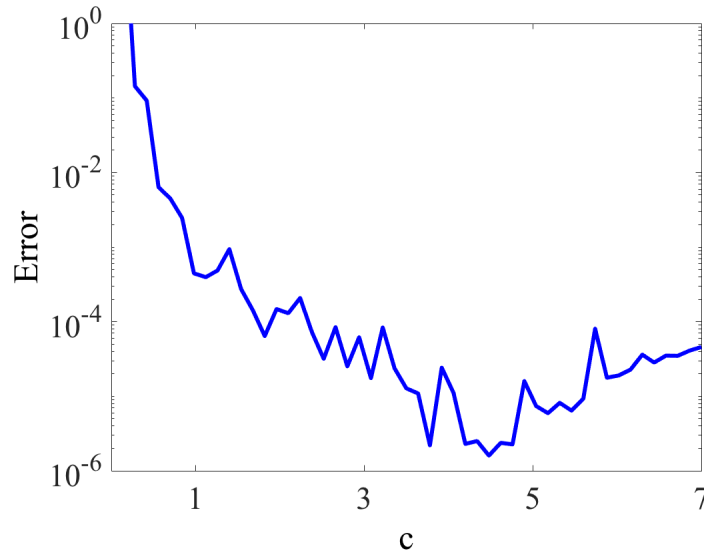


Figure 5.9: Example 2, First Biharmonic Problem: Maximum relative error versus shape parameter with  $M = 100, N = 100$  using MQ.

$M = N$	$m_x$	E	CPU (s)	sub-optimal $c$
100	2.3	$2.027(-6)$	0.306	4.348
200	2.0	$8.915(-6)$	2.068	7.071
300	1.9	$1.188(-5)$	7.126	9.116
400	1.9	$1.338(-5)$	17.639	10.526
500	1.8	$2.334(-5)$	38.524	12.423
600	1.8	$1.945(-5)$	62.811	13.608
700	1.7	$2.494(-5)$	128.802	15.563

Table 5.7: Example 2, First Biharmonic Problem: Max Rel Error and shape parameter using MQ.

the number of collocation points increases, the value  $m_x$  decreases. However, the values have a narrower range beginning at  $m_x = 2.3$  for  $M = N = 100$  and ending at  $m_x = 1.7$  for  $M = N = 700$ . This is due to the fourth order problem requiring a lower shape parameter value to achieve optimal accuracy.

Similar results are found using the Gaussian RBF. Again, the advantage that the Gaussian offers is that the same shape parameter can be used for any size set of collocation points as shown in Table 5.8. For the fourth order problem, a smaller shape parameter is chosen,  $c = 60$ .

The same results can be seen when changing the boundary conditions for the biharmonic problem. Tables (5.9) and (5.10) show results for  $M = N$  up to 700 for the second biharmonic

$M = N$	E
100	2.268(-6)
200	4.123(-6)
300	1.694(-5)
400	4.768(-6)
500	7.868(-6)
600	4.447(-5)
700	1.182(-5)

Table 5.8: Example 2, First Biharmonic problem: Max Rel Error with  $c = 60$  using Gaussian RBF.

problem replacing the equation (5.3b) with  $u = e^{2x+y}$  and  $\Delta u = 5e^{2x+y}$  on  $\partial\Omega_1, \partial\Omega_2$  for the MQ and Gaussian RBF, respectively.

$M = N$	$m_x$	E	CPU (s)	sub-optimal $c$
100	2.3	5.042(-5)	0.302	4.348
200	2.0	1.010(-5)	2.175	7.071
300	1.9	1.835(-5)	7.085	9.116
400	1.9	2.239(-5)	17.495	10.526
500	1.8	2.113(-5)	38.544	12.423
600	1.8	6.577(-5)	62.423	13.608
700	1.7	9.731(-5)	125.409	15.563

Table 5.9: Example 2, Second Biharmonic problem: Max Rel Error and shape parameter using MQ RBF.

$M = N$	E
100	6.797(-6)
200	1.152(-5)
300	1.768(-5)
400	1.764(-5)
500	2.852(-5)
600	9.021(-6)
700	1.778(-5)

Table 5.10: Example 2, Second Biharmonic problem: Max Rel Error for  $c = 60$  using Gaussian RBF.



### 5.3 Cauchy-Navier RBF-DQ

Consider the Cauchy–Navier problem

$$\begin{cases} \mathcal{L}_1(u_1, u_2) \equiv \mathcal{L}_{11}u_1 + \mathcal{L}_{12}u_2 \equiv \mu\Delta u_1 + \frac{\mu}{1-2\nu} \left( \frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_2}{\partial x \partial y} \right) = f_1, \\ \mathcal{L}_2(u_1, u_2) \equiv \mathcal{L}_{21}u_1 + \mathcal{L}_{22}u_2 \equiv \frac{\mu}{1-2\nu} \left( \frac{\partial^2 u_1}{\partial x \partial y} + \frac{\partial^2 u_2}{\partial y^2} \right) + \mu\Delta u_2 = f_2, \end{cases} \quad \text{in } \Omega, \quad (5.4a)$$

subject to the Dirichlet boundary conditions

$$u_1 = g_1 \quad \text{and} \quad u_2 = h_1 \quad \text{on} \quad \partial\Omega_1, \quad (5.4b)$$

and

$$u_1 = g_2 \quad \text{and} \quad u_2 = h_2 \quad \text{on} \quad \partial\Omega_2, \quad (5.4c)$$

or the mixed Neumann/Dirichlet boundary conditions

$$t_1 = g_1 \quad \text{and} \quad t_2 = h_1 \quad \text{on} \quad \partial\Omega_1, \quad (5.4d)$$

and

$$u_1 = g_2 \quad \text{and} \quad u_2 = h_2 \quad \text{on} \quad \partial\Omega_2, \quad (5.4e)$$

with the domain  $\Omega$  defined by  $\gamma_1 = 0.3$  and  $\gamma_2 = 1$ . The boundary conditions correspond to the exact solutions which are given by  $u_1 = e^{x+2y}$  and  $u_2 = \sin(3x + y)$ . The profile of the exact solutions are shown in Figure 5.10.

In Figure 5.11 the plot of the maximum relative errors in  $u_1$  and  $u_2$  versus the shape parameter for  $M = 100, N = 100$  is presented for the Dirichlet problem. In Tables 5.11 and 5.12, the maximum relative error is shown for  $M = N$  up to 400 for the Dirichlet problem (4.45a-5.4c) and the mixed Neumann/Dirichlet problem (5.4a), (5.4d)-(5.4e), respectively, for the MQ RBF. Results are also produced using the Gaussian RBF in Tables 5.13 and 5.14.

$M = N$	$m_x$	E1	E2	CPU (s)	sub-optimal $c$
100	2.0	1.068(-6)	1.323(-5)	1.071	5.000
200	1.4	8.712(-6)	8.278(-5)	7.787	10.102
300	0.9	9.017(-7)	6.564(-6)	30.469	19.245
400	0.8	9.038(-7)	9.729(-6)	78.360	25.000

*Table 5.11:* Example 3, Cauchy Navier Dirichlet problem: Max Rel Error and shape parameter using MQ RBF.

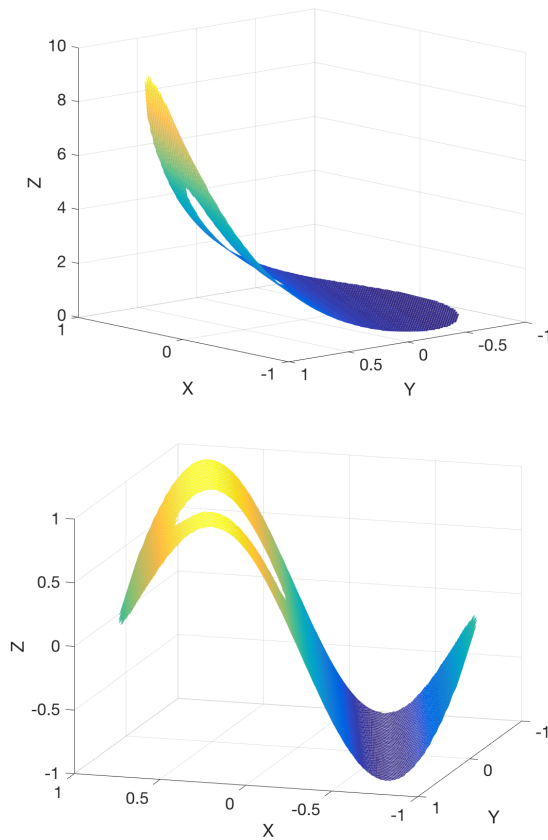


Figure 5.10: Example 3: Profile of the Exact Solutions,  $u_1$  and  $u_2$ , respectively.

$M = N$	$m_x$	E1	E2	CPU (s)	sub-optimal $c$
100	2.0	1.929(-7)	1.923(-6)	1.293	5.000
200	1.4	1.489(-7)	1.833(-6)	9.435	10.102
300	0.9	2.234(-7)	3.217(-6)	36.538	19.245
400	0.8	1.246(-7)	1.831(-6)	99.261	25.000

Table 5.12: Example 3, Cauchy Navier Neumann/Dirichlet problem: Max Rel Error and shape parameter using MQ RBF.

$M = N$	E1	E2
100	2.351(-7)	2.530(-6)
200	2.772(-6)	2.490(-5)
300	3.083(-6)	2.149(-5)
400	8.088(-7)	1.068(-5)

Table 5.13: Example 3, Cauchy Navier Dirichlet problem: Max Rel Error with  $c = 90$  using Gaussian RBF.

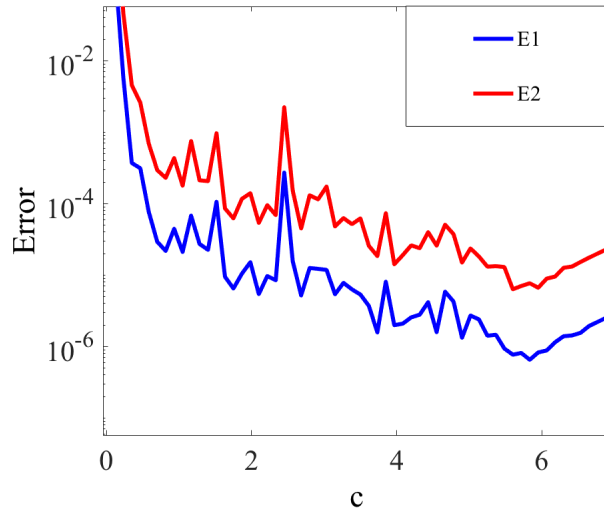


Figure 5.11: Example 3, Cauchy Navier Dirichlet problem: Maximum relative error versus shape parameter with  $M = 100, N = 100$  using MQ RBF.

$M = N$	E1	E2
100	3.902(-7)	4.358(-6)
200	1.890(-6)	1.980(-5)
300	3.361(-6)	2.282(-5)
400	3.865(-6)	6.451(-5)

Table 5.14: Example 3, Cauchy Navier Neumann/Dirichlet problem: Max Rel Error with  $c = 90$  using Gaussian RBF.

Since the Cauchy-Navier problem is harmonic, the values of  $m_x$  from the Poisson problem can be applied to find an appropriate value for the shape parameter for the MQ RBF. Having a system of two equations implies that there will be double the collocation points being used, which requires more computational time and memory. Therefore, it is only possible to go up to  $M = N = 400$  using the computer with 16 GB of memory. Yet, this is still a remarkable accomplishment while achieving a high level of accuracy.

#### 5.4 Poisson Equation LRBf-DQ

Consider the Poisson problem (5.2) with the domain  $\Omega$  defined by  $\gamma_1 = 0.3$  and  $\gamma_2 = 1$ . The boundary conditions correspond to the exact solution which is given by  $u = e^{2x+y}$ . In the local approach, the leave-one-out cross validation method is a valid tool for finding a suboptimal shape parameter. Because of the circulant nature of the distribution of collocation

points, all local domains are similar in nature and will have shape parameters close in value. Therefore, the shape parameter is calculated only once and is used throughout all of the local domains. For LOOCV, an initial search interval must be selected. For the local domain, the shape parameter is fairly consistent for various lengths of initial search intervals  $[min, max]$  for the small local domain size of  $n = 9$  as shown in Table 5.15. The stability of the shape parameter begins to deteriorate as the local domain increases in size, as shown for  $n = 30$  and  $n = 50$ . A slight improvement can be observed as  $n$  increases although an increase in computational time can be expected as well. In Tables 5.16 and 5.20, the results are presented using a large number of collocation points using LOOCV for both the Dirichlet boundary condition as well as the Neumann/Dirichlet boundary condition. The fact that it takes only 43 seconds of computational time to solve the PDE using one million collocation points displays the efficiency of the proposed LRBF-DQ MDA.

$[min, max]$	$n = 9$		$n = 30$		$n = 50$	
	$c$	$E$	$c$	$E$	$c$	$E$
[0, 3]	0.5036	1.0129(-5)	2.0213	5.3332(-6)	1.1955	2.4492(-6)
[0, 4]	0.6256	6.0946(-5)	2.3870	7.9265(-5)	1.9989	9.5581(-6)
[0, 5]	0.6053	1.9654(-5)	3.0902	1.5571(-5)	3.8121	5.7068(-6)
[0, 6]	0.5744	9.4325(-6)	4.3144	1.7955(-5)	4.5905	1.5259(-5)
[0, 7]	0.6312	1.9471(-5)	2.5961	3.2400(-5)	3.9546	7.6296(-6)
[0, 8]	0.5666	4.2063(-5)	2.3344	7.2283(-7)	3.1615	3.3218(-6)

Table 5.15: Example 4, Poisson Dirichlet problem: Max Rel Error and optimal shape parameter with various search intervals of LOOCV for  $M = N = 200$  using MQ RBF.

$M = N$	$c$	E	CPU (s)
200	0.5036	1.0129(-5)	0.365
300	0.8065	3.9362(-6)	0.951
400	0.8098	3.1808(-5)	2.079
500	1.1291	5.3057(-6)	3.591
600	1.4208	1.8988(-5)	6.970
700	1.8885	2.9185(-6)	10.846
800	1.9523	6.5150(-6)	16.061
900	1.9684	2.0352(-5)	28.352
1000	2.2015	1.0283(-5)	42.943

Table 5.16: Example 4, Poisson Dirichlet problem: Max Rel Error and optimal shape parameter via LOOCV for  $n = 9$  using MQ RBF.

$M = N$	$c$	E	CPU (s)
200	0.5036	7.5960(-4)	0.577
300	0.3967	2.4211(-4)	1.083
400	0.7201	4.0176(-4)	2.054
500	1.1152	2.8359(-4)	3.4601
600	1.6787	3.8463(-4)	6.176
700	2.0111	3.7611(-4)	10.553
800	1.9383	3.1873(-4)	16.575
900	1.9541	3.4984(-4)	28.367
1000	2.4492	3.5562(-4)	42.789

Table 5.17: Example 4, Poisson Neumann problem: Max Rel Error and optimal shape parameter via LOOCV for  $n = 9$  using MQ RBF.

A second approach to choosing an appropriate shape parameter is the dimensionless shape parameter, which was introduced in Chapter 2. The advantage to the dimensionless shape parameter is the stability in choosing the shape parameter value. In Figure 5.12, the error is stable over the range  $[30, 150]$  for  $M = N = 100$ . This makes it easier to choose a shape parameter without sacrificing accuracy. Table 5.18 displays the error corresponding to different values of the dimensionless shape parameter  $c$  for  $M = N = 200$ .

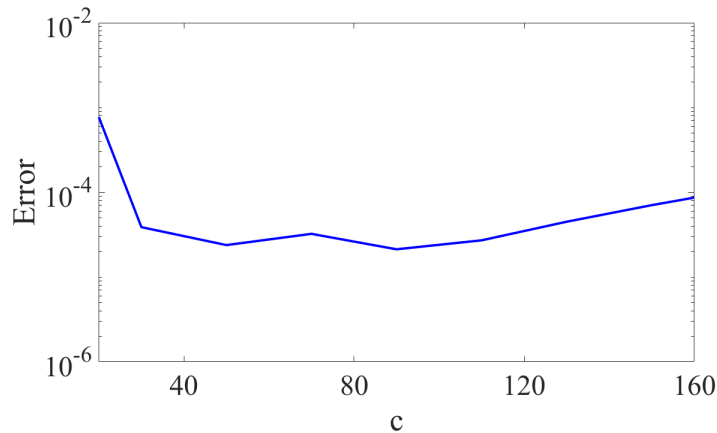


Figure 5.12: Example 4, Poisson Dirichlet problem: Maximum relative error versus dimensionless shape parameter with  $M = N = 100$  using MQ RBF.

As the number of points increases, the required shape parameter will also increase, thus the acceptable range of values will increase as well. After experimenting with the dimensionless shape parameter, it is found that the best value for  $c$  is  $MN/1000$  since it falls within the range of stability and produces good results. Using the above selection procedure,

$c$	$cr_0$	E
10	0.1881	1.5167(-2)
30	0.5643	3.8797(-5)
50	0.9404	2.3853(-5)
70	1.3166	3.2355(-5)
110	2.0690	2.7129(-5)
150	2.8213	7.0490(-5)
190	3.5737	1.5479(-4)

*Table 5.18:* Example 4, Poisson Dirichlet problem  $M = N = 200$ : Max Rel Error for dimensionless shape parameter using MQ RBF.

the value of  $c$  has been chosen as shown in Table 5.19 for  $M = N$  values up to 1000 for the Dirichlet boundary problem. The results are consistent with LOOCV without the need of choosing an appropriate initial search interval.

$M = N$	$c$	E	CPU (s)
200	40	1.6939(-5)	0.371
300	90	4.9842(-6)	1.003
400	160	1.545(-5)	2.144
500	250	1.4673(-5)	3.630
600	360	3.2208(-6)	6.959
700	490	3.3203(-6)	10.949
800	640	1.5099(-6)	16.979
900	810	2.7664(-6)	28.101
1000	1000	6.2346(-5)	41.485

*Table 5.19:* Example 4, Poisson Dirichlet problem: Max Rel Error with  $c = MN/1000$  using MQ RBF.

In order to obtain a more accurate RBF approximation, it is necessary to balance the size of the supporting domain and the sensitivity of the shape parameter. In Table 5.21, the maximum error is calculated with various local domain sizes using the dimensionless shape parameter. As the number of nodes in the local domain increases, the computational time increases and does not have a notable difference in accuracy with the exception of  $n = 5$ . The LRBF-DQ provides accurate approximate solutions even with a small local domain size. For the given distribution of the collocation points, it is only necessary to use nine points in the local domain to reach satisfactory results.

$M = N$	$c$	E	CPU (s)
200	40	8.3935(-4)	0.611
300	90	5.8175(-4)	1.240
400	160	5.1726(-4)	2.391
500	250	5.3007(-4)	3.967
600	360	5.0164(-4)	7.365
700	490	4.5828(-4)	11.958
800	640	5.0732(-4)	16.976
900	810	5.5363(-4)	30.167
1000	1000	5.4880(-4)	44.911

Table 5.20: Example 4, Poisson Neumann problem: Max Rel Error with  $c = MN/1000$  using MQ RBF.

$n$	E	CPU (s)	$c$
5	1.2341(-2)	5.5479	1.6062
9	1.4673(-5)	5.6120	1.8860
15	6.6101(-5)	5.8642	2.9091
30	4.4999(-5)	6.2422	4.1956
40	1.1590(-5)	5.9229	4.8168
80	6.6619(-6)	6.425	7.0315
121	3.1574(-5)	6.554	8.8442

Table 5.21: Example 4, Poisson Dirichlet problem: Influence Domain for  $M=N=500$  using MQ RBF.

## 5.5 Biharmonic Equation LRBF-DQ

Consider the Biharmonic problem (5.3) with the domain  $\Omega$  defined by  $\gamma_1 = 0.3$  and  $\gamma_2 = 1$ . In this example, collocation points are defined with  $\alpha_n = (-1)^n/5, n = 1, \dots, N$ , to avoid possible singularities. The boundary conditions correspond to the exact solution, which is given by  $u = e^{2x+y}$ .

The profile of the error versus the shape parameter  $c$  for  $M = N = 80, 200, n = 30$  is given in Figure 5.13.

Although there is fluctuation in the curve for  $M = N = 80$ , the estimation of a good shape parameter using LOOCV is fairly stable for various initial search intervals as shown in Tables 5.22 and 5.23 for the first and second biharmonic equation, respectively.

In the case  $n = 9$ , the solution is stable in terms of the shape parameter, however the accuracy does not achieve the same results as that of  $n = 30$ . The accuracy does not appear to have a significant improvement with an increase in the size of the local domain for  $n = 50$ .

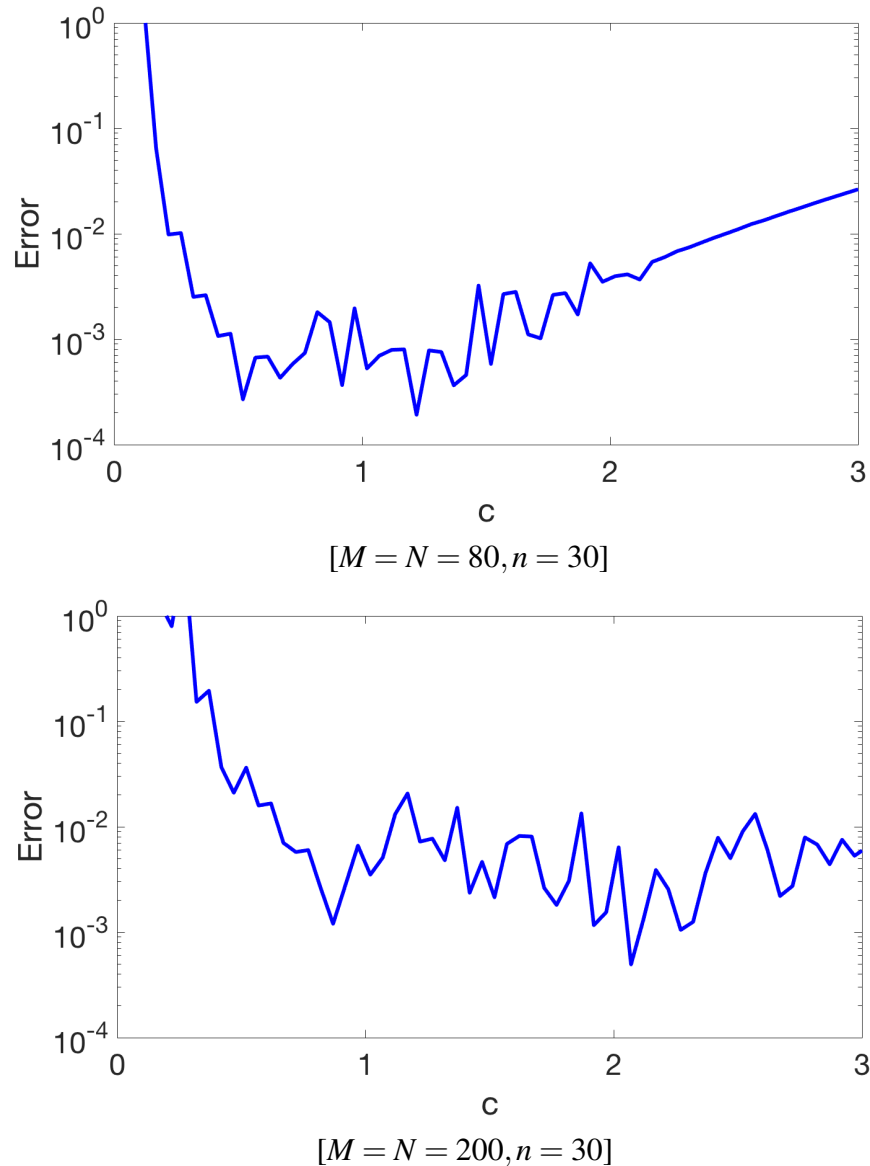


Figure 5.13: Example 5: First Biharmonic problem: Error versus shape parameter for the cases  $M = N = 80, 200, n = 30$  using MQ RBF.

The stability of the shape parameter also appears to decrease as  $n$  increases. When the number of collocation points is increased ( $M = N = 200$ ), the stability continues to deteriorate more intensely, and the shape parameter becomes unpredictable for various initial search intervals, as shown in Tables 5.24 and 5.25 for the first and second biharmonic equations, respectively.

The fluctuation in the accuracy curve in Figure 5.13 also increases with an increase in collocation points. Additionally, the accuracy does not have a noticeable improvement when increasing the number of collocation points. The low accuracy of the results obtained in



	$n = 9$		$n = 30$		$n = 50$	
$[min, max]$	$c$	$E$	$c$	$E$	$c$	$E$
[0, 3]	0.3819	1.0850(-1)	2.1554	3.7893(-2)	1.6428	2.9329(-2)
[0, 4]	0.3916	3.6849(-1)	2.4357	3.8542(-2)	2.6184	2.7642(-2)
[0, 5]	0.3915	1.9127(-1)	2.8283	4.0640(-2)	3.4451	5.1483(-2)
[0, 6]	0.3754	4.6133(-1)	2.4333	3.9040(-2)	3.5419	4.4531(-2)
[0, 7]	0.3971	1.8752(-1)	2.4336	3.9006(-2)	2.6203	2.7123(-2)
[0, 8]	0.3832	6.0438(-1)	2.1512	3.8240(-2)	3.5417	4.3549(-2)

Table 5.22: Example 5, First Biharmonic problem: Max Rel Error and optimal shape parameter via LOOCV for  $M = N = 80$  using MQ RBF.

	$n = 9$		$n = 30$		$n = 50$	
$[min, max]$	$c$	$E$	$c$	$E$	$c$	$E$
[0, 3]	0.3819	4.3539(-1)	2.3718	5.4498(-4)	2.1554	5.3486(-3)
[0, 4]	0.3916	3.7193(-1)	1.0239	3.5424(-3)	2.4357	9.7082(-3)
[0, 5]	0.3915	3.4028(-1)	1.0240	3.8391(-3)	2.8283	1.9726(-2)
[0, 6]	0.3754	4.8182(-1)	0.8318	2.0337(-3)	2.4333	9.3419(-3)
[0, 7]	0.3971	4.1428(-1)	0.8354	3.5208(-2)	2.4336	1.0045(-2)
[0, 8]	0.3832	3.8092(-1)	1.0231	2.7986(-3)	2.1512	5.1182(-3)

Table 5.23: Example 5, Second Biharmonic problem: Max Rel Error and optimal shape parameter via LOOCV for  $M = N = 80$  using MQ RBF.

	$n = 30$		$n = 40$	
$[min, max]$	$c$	$E$	$c$	$E$
[0, 3]	1.9873	6.9393(-3)	1.9705	1.8006(-3)
[0, 4]	1.9839	7.4400(-3)	3.5619	2.4767(-2)
[0, 5]	4.1629	1.7991(-2)	2.6742	8.2833(-3)
[0, 6]	3.6560	1.7690(-2)	4.1751	1.2148(-2)
[0, 7]	3.2882	2.8798(-2)	2.6738	8.9452(-2)
[0, 8]	4.1630	1.9839(-2)	2.6711	3.5229(-2)

Table 5.24: Example 5, First Biharmonic problem: Max Rel Error versus shape parameter with various LOOCV initial search intervals for  $M = N = 200$  using MQ RBF.

this example is not considered unusual. It is noted in [54] that the accuracy of the RBF-DQ decreases as the order of the derivative increases. The difficulties of specifically solving biharmonic problems using local methods are also well documented in the literature [2], [37].

Similar results occur when using the dimensionless shape parameter, perhaps with a slight

$[min, max]$	$n = 30$		$n = 50$	
	$c$	$E$	$c$	$E$
[0, 3]	1.9873	3.0261(-3)	1.9705	4.6244(-4)
[0, 4]	1.9839	1.6197(-2)	3.5619	6.6223(-3)
[0, 5]	4.1629	2.6217(-2)	2.6742	1.8432(-2)
[0, 6]	3.6560	1.3596(-2)	4.1751	4.5175(-2)
[0, 7]	3.2882	6.1351(-2)	2.6738	8.2696(-2)
[0, 8]	4.1630	2.6051(-2)	2.6711	4.3496(-3)

Table 5.25: Example 5, Second Biharmonic problem: Max Rel Error versus shape parameter with various LOOCV initial search intervals for  $M = N = 200$  using MQ RBF.

advantage in that it is not necessary to choose the initial search interval as when using LOOCV. Results are shown in Table 5.26.

$M = N$	$cc$	$E$	CPU (s)
200	40	3.7882(-3)	0.781
300	90	7.7997(-3)	1.711
400	160	5.2316(-3)	3.781
500	250	7.7577(-3)	6.136
600	360	6.6737(-3)	11.432

Table 5.26: Example 5, Second Biharmonic Problem: Shape parameters and corresponding errors obtained using dimensionless shape parameter.

## 5.6 Cauchy-Navier LRBF-DQ

Consider the Cauchy-Navier problem (5.4) with the domain  $\Omega$  defined by  $\gamma_1 = 0.3$  and  $\gamma_2 = 1$ . The boundary conditions correspond to the exact solution which is given by  $u_1 = e^{x+2y}$ ,  $u_2 = \sin(3x+y)$ . In the numerical experiments,  $M = N = 100$ ,  $n = 9$  are chosen, and in Figure 5.14 it is shown how the errors behave with respect to the shape parameter.

As was the case in Example 3, the two error curves  $E_1$  and  $E_2$  are relatively smooth and predictable, which is ideal for the application of LOOCV. The values of the shape parameters obtained using LOOCV and the corresponding errors, which are presented in Table 5.27, are in agreement with the optimal solutions in Figure 5.14 for the same input data. In Table 5.28, the results obtained for the Dirichlet Cauchy Navier problem using a large number of collocation points up to the value of 600 for  $M = N$  are presented.

As shown in Tables 5.29 and 5.30, similar results are obtained for the Cauchy Navier problem with mixed Neumann/Dirichlet boundary condition, that is, the boundary value

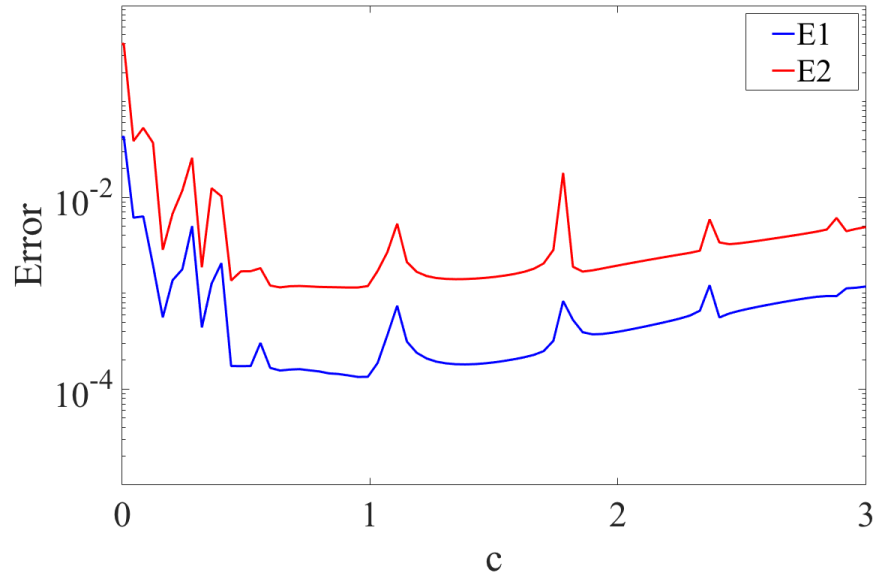


Figure 5.14: Example 6: Cauchy Navier problem. Maximum relative error versus shape parameter for  $M = N = 100, n = 9$  using MQ RBF.

$[min, max]$	$c$	$E_1$	$E_2$
[0, 3]	0.556	2.005(-4)	1.343(-3)
[0, 4]	0.563	2.422(-4)	1.394(-3)
[0, 5]	0.611	1.917(-4)	1.283(-3)
[0, 6]	0.612	2.982(-4)	2.332(-3)
[0, 7]	0.616	2.020(-4)	1.325(-3)
[0, 8]	0.559	2.255(-4)	1.402(-3)

Table 5.27: Example 6, Cauchy Navier Dirichlet problem: Shape parameters and corresponding errors obtained with various search intervals of LOOCV;  $M = N = 100, n = 9$  using MQ RBF.

$M = N$	$c$	$E_1$	$E_2$	CPU (s)
200	1.130	4.636(-4)	3.298(-3)	1.729
300	1.706	9.842(-5)	1.049(-3)	4.836
400	2.275	1.416(-4)	5.752(-4)	12.322
500	2.842	1.888(-4)	1.080(-3)	25.185
600	2.796	4.275(-4)	1.804(-3)	50.760

Table 5.28: Example 6, Cauchy Navier Dirichlet problem: Shape parameters and corresponding errors obtained using LOOCV with local domain size  $n = 9$  using MQ RBF.

problem consisting of (5.4d)-(5.4e).

$[min, max]$	$c$	$E_1$	$E_2$
[0, 3]	0.556	1.002(-3)	4.115(-3)
[0, 4]	0.563	1.049(-3)	4.092(-3)
[0, 5]	0.611	9.928(-4)	4.066(-3)
[0, 6]	0.612	9.567(-4)	6.526(-3)
[0, 7]	0.616	1.015(-3)	4.067(-3)
[0, 8]	0.559	1.030(-3)	4.155(-3)

*Table 5.29:* Example 6, Cauchy Navier Neumann problem: Shape parameters and corresponding errors obtained using various search intervals of LOOCV;  $M = N = 100, n = 9$  using MQ RBF.

$M = N$	$c$	$E_1$	$E_2$	CPU (s)
200	1.130	3.827(-3)	4.433(-2)	1.885
300	1.706	2.472(-4)	2.604(-3)	4.890
400	2.275	4.008(-4)	1.933(-3)	9.839
500	2.842	5.131(-4)	3.650(-3)	23.529
600	2.796	6.442(-4)	3.126(-3)	52.138

*Table 5.30:* Example 6, Cauchy Navier Neumann problem: Shape parameters and corresponding errors obtained using dimensionless shape parameter using MQ RBF.

It is worth noting that a problem consisting of 360,000 collocation points can be solved in only 50 seconds using a computer with 16 GB of memory space, which demonstrates the efficiency of the proposed algorithm.

## Chapter 6

# CONCLUSIONS AND FUTURE WORKS

### 6.1 Conclusions

This dissertation developed a new numerical algorithm to solve various types of large scale partial differential equations. Both a global and local scheme was used for the radial basis function differential quadrature method combined with a matrix decomposition algorithm.

The global method is simple, efficient, and accurate in solving these large scale problems quickly and efficiently. In particular, the global method has been applied to PDEs with up to one million collocation points. The method, due to the nature of RBFs, can easily extend to problems in high-dimensional spaces. The matrix decomposition algorithm developed in this paper requires that the right hand side of the equation be circulant. For the Poisson and Biharmonic problems, this requirement is certainly satisfied. For the Cauchy Navier problem, however, this is not the case. To overcome this obstacle, the right hand side of the differential equation is able to be transformed so that it becomes circulant. Once the transformation has been completed, the proposed procedure can be continued. The global method has the capability of solving problems up to half a million points before memory becomes an issue on a laptop using 16 GB of memory. Even so, the global method was used to solve up to a million points using a workstation with 256 GB of memory.

To overcome the problem of memory as the number of collocation points become large, a local method has been developed for the RBF Differential Quadrature Matrix Decomposition Algorithm. Instead of a large dense matrix, as in the global RBF-DQ, the local method creates a sparse matrix, which makes it more efficient to solve. Moreover, this sparse matrix is also block circulant allowing the implementation of the matrix decomposition algorithm. In the sense of the Cauchy Navier problem, although this block circulant structure is not present, a transformation can again be applied to create the block circulant structure. Specific examples in this paper have demonstrated that the local method has the ability of solving problems with up to a million points in about one minute using a computer with memory space of only 16GB.

For the local method, the leave-one-out cross validation method (LOOCV) is effective in calculating a good shape parameter. It was only necessary to calculate the shape parameter for one local domain and to apply this value for all other local domains due to the circulant

structure, which creates similar patterns in each local domain. While LOOCV is not practical for the global method due to the large size of the matrix, the adjusted Fasshauer estimate was employed to calculate an appropriate suboptimal shape parameter. It was observed that this value is dependent on the density of the collocation points. As the number of collocation points increased, the shape parameter should increase as well, which is controlled by the decreasing value  $m_x$ . This Fasshauer estimate works well for solving large scale problems and does not require significant computational time.

This paper primarily focused on the normalized multiquadric method for the radial basis function. While there are many RBFs that can be used to achieve accurate results, the MQ is the most commonly used RBF and has consistently produced excellent results. Furthermore, the normalized MQ is stable with regard to the shape parameter meaning that accurate results can be achieved over a large range of values making it less critical in choosing the shape parameter. Similar results were also achieved using the inverse multiquadric and the Gaussian RBF.

While the accuracy of the local method does not achieve the same level of accuracy of the global scheme, excellent results are still achieved and provides a significant reduction in the required time and memory. It is worthy to note that the biharmonic problem has limitations on the accuracy, especially for a local domain size of nine. Better results are achieved when increasing the number of collocation points in the local domain, but this comes with the expense of the shape parameter becoming less stable in terms of the search interval. This difficulty is not unusual and has been documented in the RBF literature.

It is apparent that meshless methods using local schemes can compete with traditional numerical methods for solving large scale PDEs, such as the finite element, finite difference, or finite volume methods. Due to the collocation approach, no numerical integration is required. In this dissertation, the numerical results were calculated using a circular domain, more specifically an annulus. This method can extend to other domain types as well as irregular domains using a conformal mapping procedure. Furthermore, the proposed methods can be easily extended to solving three-dimensional problems. This is indeed where its strengths may lie. As such, the approach offers the prospects of an efficient algorithm for solving more challenging problems in science and engineering. While this work has not completely validated the method, the results obtained along with the given analysis does provide strong evidence to the utility of the approach and supports the belief that these methods developed have a substantial future in solving a wide range of difficult problems that occur in the natural sciences and engineering.

## 6.2 Future Works

In this dissertation, the radial basis function differential quadrature method was combined with a modified matrix decomposition algorithm that takes advantage of the right hand side of the equation being block circulant. Both the global and local methods can be used for solving large scale problems efficiently. As with all new methods, thorough theoretical analysis remains to be done, as does the possible extension of the methods to other types of partial differential equations. The following topics are of particular interest:

- An MDA method has already been applied to three dimensional problems in [30]. The matrix decomposition algorithm employed in the RBF-DQ method suggests that this method can be effective for the solution of three dimensional problems as well.
- This study limited the domain to that of an annulus. It is of interest to test the proposed method for a variety of domains using a conformal mapping so that the method becomes appropriate for irregular domains [22].
- Many engineering problems and other physical phenomena are mostly modeled by nonlinear PDEs. The RBF-DQ method can be applied to nonlinear PDEs. The results in this paper suggests that the MDA can also be appropriate for these nonlinear PDEs for large scale problems.

## BIBLIOGRAPHY

- [1] R. Bellman, B. G. Kashef, and J. Casti. Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. *J. Comput. Phys.*, 10, 40-52, 1972.
- [2] M. Ben-Artzi, I. Chorev, J. P. Croisille, and D. Fishelov. A compact difference scheme for the biharmonic equation in planar irregular domains. *SIAM J. Numer. Anal.*, 47:3087-3108, 2009.
- [3] B. Bialecki, G. Fairweather, and A. Karageorghis. Matrix decomposition algorithms for elliptic boundary value problems: A survey. *Numer. Algorithms*, 56,253-295, 2011.
- [4] M.D. Buhmann. *Radial Basis Functions, Theory and Implementations*. Cambridge University Press, 2003.
- [5] Y. L. Chan, L. H. Shen, C. T. Wu, and D. L. Young. Interpolation techniques for scattered data by local radial basis function differential quadrature method. *Int. J. Comput. Methods*, 10:02, 2013.
- [6] Y. L. Chan, L. H. Shen, C. T. Wu, and D. L. Young. A novel upwind-based local radial basis function differential quadrature method for convection-dominated flows. *Computers and Fluids*, 89:157-166, 2014.
- [7] 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.
- [8] 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. doi: 10.1002/num.20631.
- [9] C. S. Chen, M. A. Golberg, M. Ganesh, and A. H. D. Cheng. Multilevel compact radial functions based computational schemes for some elliptic problems. *Computers and Mathematics with Application*, 43:359–378, 2002.
- [10] C. S. Chen, Y. C. Hon, and R. A. Schaback. *Scientific computing with radial basis functions*. Tech. Rep., Department of Mathematics, University of Southern Mississippi, Hattiesburg, MS 39406, USA, preprint, 2005.
- [11] W. Chen, Z. Fu, and C.S. Chen. *Recent advances in Radial basis Function Collocation Methods*. Springer, 2013.
- [12] C.S. Chen, A. Karageorghis, O. Macchion. A Kansa-Radial Basis Function Method for Elliptic Boundary Value Problems in Annular Domains. *SIAM J. Sci. Comput.*, 65:1240-1269, 2015.
- [13] P. J. Davis. *Circulant Matrices, Second edition*. AMS Chelsea Publishing, Providence, Rhode Island, 1994.



- [14] H. Ding, C. Shu, K. S. Yeo, and Z. L. Lu. Simulation of natural convection in eccentric annuli between a square outer cylinder and a circular inner cylinder using local MQ-DQ method. *Numerical Heat Transfer, Part A*, 47:291-313, 2005.
- [15] G. E. Fasshauer and J. G. Zhang. On choosing optimal shape parameters for RBF approximation. *Numer. Algorithms*, 45, 345-368, 2007.
- [16] G. Fasshauer. Solving partial differential equations by collocation with radial basis functions. *Proceedings of Chamonix pp 1-6*, 1996.
- [17] G. Fasshauer. *Meshfree Application Method with Matlab*. Interdisciplinary Mathematical Sciences, 2007.
- [18] G. Fornberg, B. and Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *CoComput. Math. Appl.*:853-867, 2004.
- [19] R. Franke. Scattered data interpolation: Tests of some method. *Mathematics of Computation*, 38:181-200, 1982.
- [20] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, 76:1905-1915, 1971.
- [21] F. Hartmann. *Elastostatics, Progress in Boundary Element Methods Vol. 1* (C. A. Brebbia, ed.). Pentech Press, London, 1981, pp. 84-167.
- [22] A. R. H. Heryudono and T. A. Driscoll. Radial basis function interpolation on irregular domain through conformal transplantation. *J. Sci. Comput.*, 44, 286-300, 2010.
- [23] L. Homayoon, M. J. Abadini, and S. M. R. Hashemi. RBF-DQ solution for shallow water equations. *J. Waterway, Port, Coastal, Ocean Eng.*, 139:45-60, 2013.
- [24] Y. C. Hon, R. Schaback, X. Zhou. An adaptive greedy algorithm for solving large rbf collocation problems. *Numerical Algorithms*, 32(1):13-25, 2003.
- [25] C. S. Huang, Cheng-Feng Lee, and A. H. D. Cheng. Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method. *Eng. Analy. Boundary Elements*, 31(7):614-623, 2007.
- [26] E.J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - I. *Comput. Math. Applic.*, 19(8/9):127-145, 1990.
- [27] 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.
- [28] A. Karageorghis, The method of fundamental solutions for elliptic problems in circular domains with mixed boundary conditions. *Numer. Algorithms*, 68:185-211, 2015.
- [29] A. Karageorghis, C. S. Chen, and X-Y. Liu, Kansa-RBF algorithms for elliptic problems in axisymmetric domains. *SIAM J. Sci. Comput.*, 38, A435-A470, 2016.
- [30] A. Karageorghis, C. S. Chen, and Y.-S. Smyrlis, A matrix decomposition RBF algorithm: approximation of functions and their derivatives. *Appl. Numer. Math.*, 57:304-319, 2007.

- [31] A. Karageorghis, C. S. Chen, and Y.-S. Smyrlis, Matrix decomposition RBF algorithm for solving 3D elliptic problems. *Eng. Anal. Bound. Elem.*, 33:1368-1373, 2009.
- [32] A. Korkmaz and I. Dağ, Solitary wave simulations of complex modified Korteweg-de Vries equation using differential quadrature method. *Comput. Phys. Commun.*, 180:1516-1523, 2009.
- [33] A. Krowiak, Hermite type radial basis function-based differential quadrature method for higher order equations. *Appl. Math. Model.*, 40, 2421-2430, 2016.
- [34] E. Larsson and B. Fornberg. A numerical study of some radial basis function based solution methods for elliptic PDEs. *Comput. math. Appl.*, 46:891-902, 2003.
- [35] C. K. Lee, X. Liu, and S. C. Fan, Local multiquadric approximation for solving boundary value problems. *Comput. Mech.*, 30:396-409, 2003.
- [36] X. Li and Chen C. S. A mesh-free method using hyperinterpolation and fast fourier transform for solving differential equation. *Eng. Anal. Boundary Elements*, 28, 1253-1260, 2004.
- [37] M Li, G. Amazzar, A. Naji, and C. S. Chen, Solving biharmonic equations using the localized method of approximate particular solutions. *Int. J. Comput. Math.*, 91:1790-1801, 2014.
- [38] N. A. Libre, A. Emdadi, E. J. Kansa, M. Rahimian, and M. Shekarchi. A stabilized rbf collocation scheme for neumann type boundary value problems. *Computer Modeling in Engineering and Science*, 24:61-80, 2008.
- [39] X. Y. Liu, A. Karageorghis, and C. S. Chen, A Kansa-radial basis function method for elliptic boundary value problems in annular domains. *J. Sci. Comput.*, 65:1240-1269, 2015.
- [40] N. Mai-Duy, T. Tran-Cong. Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of poisson's equation. *Eng. Anal. Boundary Elements*, 2002, 26:133-156.
- [41] The MathWorks, Inc., 3 Apple Hill Dr., Natick, MA, *Matlab*.
- [42] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite function. *Constr. Approx.*, 11-22, 1986.
- [43] K W Morton and D F Mayers. *Numerical Solution of PDEs, an Introduction*. Cambridge University Press, 2005.
- [44] J N Reddy. *An Introduction to the Finite Element Method*. McGraw-Hill, Inc., 2006.
- [45] 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.
- [46] R. Schaback, L. Ling. Stable and convergent unsymmetric meshless collocation methods. *SIAM Journal on Numerical Analysis*, 46:1097-1115, 2008.
- [47] I. J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, 39:811-841, 1938.
- [48] L. H. Shen, K. H. Tseng, and D. L. Young. Evaluation of multi-order derivatives by local radial basis function differential quadrature method. *J. Mech.*, 29:67-78, 2013.

- [49] C. Shu, Y. T. Chew. Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems. *Commun. in Numerical Methods*, 13:643-653, 1997.
- [50] C. Shu, H. Ding, and K. S. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 192:941-954, 2003).
- [51] C. Shu, B. E. Richards. Application of generalised differential quadrature to solve two-dimension incompressible Navier-Stokes equations. *Int. J. Numer. Methods Fluids*, 15:791-798, 1992.
- [52] C. Shu and Y. L. Wu, Integrated radial basis functions-based differential quadrature method and its performance. *Int. J. Numer. Meth. Fluids*, 53:969-984, 2007.
- [53] R. Vertnik and B. Sarler. Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *International Journal of Numerical Methods for Eather and Fluid Flow*, 16:617-640, 2006.
- [54] Y. L. Wu and C. Shu. Development of RBF-DQ method for derivative approximation and its application to simulate natural convection in concentric annuli. *Comput. Mech.* 29:477-485, 2002.
- [55] J. Yang, X. Liu, and P.H. Wen. The local kansa's method for solving berger equation. *EngEng. Anal. Boundary Elements*, 16-22, 2015.
- [56] G. Yao, J. Kolibal, and C. S. Chen. A localized approach for the method of approximate particular solutions. *Comput. Math. Appl.*, 61(9):2376-2387, May 2011.
- [57] G. Yao. *Local radial basis function methods for solving partial differential equations*. PhD thesis, University of Southern Mississippi, 2010.
- [58] X. Zhang, K. Z. Song, M. W. Lu, and X. Liu. Meshless methods based on collocation with radial basis functions. *Computational Mechanics*, 26:333-343, 2000.
- [59] X. Zhang, H. Zhu, and L. Kuo. A comparison study of the LMAPS method and the LDQ method for time-dependent problems. *Eng. Anal Boundary Elements*, 11:1408-1415, 2013.