UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING SCIENCE AND MATHEMATICS

School of Engineering Sciences

Radial Basis Function Based Meshless Methods for Fluid Flow Problems

by

Phani P. Chinchapatnam

Thesis for the degree of Doctor of Philosophy

September 2006

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING SCIENCE AND MATHEMATICS

SCHOOL OF ENGINEERING SCIENCES

Doctor of Philosophy

RADIAL BASIS FUNCTION BASED MESHLESS METHODS FOR FLUID FLOW PROBLEMS

by Phani P. Chinchapatnam

This thesis is concerned with the development of meshless methods using radial basis functions for solving fluid flow problems. The advantage of meshless methods over traditional mesh-based methods is that they make use of a scattered set of collocation points in the physical domain and no connectivity information is required. An important objective of the present research is to develop novel meshless methods for unsteady flow problems. Symmetric/unsymmetric radial basis function collocation schemes are proposed for solving an unsteady convection-diffusion equation for various Peclet numbers. Both global and compactly supported radial basis functions are used and the convergence behaviours of various radial basis functions are studied. The performance of the presented schemes is shown by using both uniform as well as scattered distribution of points. Numerical results suggest that these schemes are capable of obtaining accurate results for low and medium Peclet numbers. Next, two directions have been explored in this thesis for using radial basis functions to solve large scale problems encountered in fluid flow problems. They are namely, domain decomposition schemes and radial basis functions in finite difference mode. These schemes are shown to be computationally efficient and also aid in circumventing the ill-conditioning problem. The performance of both schemes are evaluated by solving the unsteady convection-diffusion problem. The last part of this thesis is concerned with the solution of the 2D Navier-Stokes equations. Meshless methods based on radial basis collocation and scattered node finite difference schemes are formulated for solving steady and unsteady incompressible Navier-Stokes equations. A novel ghost node strategy is proposed for incorporating the no-slip boundary conditions. Optimisation strategies based on residual error objective and leave-one-out statistical criterion are proposed to evaluate the optimal shape parameter value in case of the multiquadric RBF for collocation and scattered finite difference approaches respectively. Standard benchmark problems like the driven cavity flows in square and rectangular domains and backward facing step flow problem are solved to study the performance of the developed schemes. Finally, a higher order radial basis function based scattered node finite difference method is proposed for solving the incompressible Navier-Stokes equations.

to my parents

Contents

List of Figures

List of Tables

Declaration of authorship

I, **Phani P. Chinchapatnam**, declare that the thesis entitled **Radial Basis Function Based Meshless Methods for Fluid Flow Problems** and the work presented in it are my own. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published and some articles are under review as:

    1. P. P. Chinchapatnam, K. Djidjeli and P. B. Nair, 2006, "Unsymmetric and symmetric meshless schemes for the unsteady convection-diffusion equation." *Computer Methods in Applied Mechanics and Engineering,* **195**, 2432-2453.
    2. Phani P. Chinchapatnam, K. Djidjeli and Prasanth B. Nair, 2006, "Domain decomposition for time-dependent problems using radial based meshless methods." *Numerical Methods for Partial Differential Equations,* Published Online: 25$^{\text{th}}$ May, 2006.
    3. Phani P. Chinchapatnam, K. Djidjeli and Prasanth B. Nair, 2006, "Radial basis function meshless method for the steady incompressible Navier-Stokes equations." *International Journal of Computer Mathematics,* Submitted for review.

4. Phani P. Chinchapatnam, K. Djidjeli and Prasanth B. Nair, 2006, "Meshless RBF collocation for steady incompressible flows." In Proceedings: 36$^{\text{th}}$ *AIAA Fluid Dynamics Conference and Exhibit,* AIAA, San Francisco, California. Ref:AIAA 2006-3525.

5. Phani P. Chinchapatnam, K. Djidjeli, and Prasanth B. Nair, 2005, " Meshless solution of the Burgers equation." In Proceedings: *ECCOMAS Thematic Conference on Meshless Methods,* Instituto Superior Tecnico, Lisboa, Portugal, 1-D43.

6. Phani P. Chinchapatnam, K. Djidjeli, and Prasanth B. Nair, 2005, " Meshless domain decomposition schemes for nonlinear elliptic PDEs." In Proceedings: *Third MIT Conference on Computational Fluid and Solid Mechanics,* Elsevier, Cambridge, USA, 1082-1086.

7. K. Djidjeli, P. P. Chinchapatnam, P. B. Nair and W.G. Price, 2004, "Global and compact meshless schemes for the unsteady convection-diffusion equation." In Proceedings: *International Symposium on Healthcare and Biomedical Research Interaction,* Morocco, 8 pages on CDROM.

Signed:.....

Date:.....

Acknowledgements

I wish to thank my supervisors Dr. Kamal Djidjeli and Dr. Prasanth Nair for being a source of constant support and encouragement without which this research would not have been possible. I am particularly indebted to Prasanth for those innumerable enlightening discussions on all facets of work and life over a cup of coffee. I am thankful to my parents and Prof. G. R. Shevare for supporting and encouraging me to do a PhD. I am also highly obiliged to Mr. Ravi Sankar for introducing me to the joys of mathematics in my college years.

My three years in Southampton would not have been so exciting had it not been for the wonderful friends I had during this time, particularly Apu, Bhaskar, Sree, Sachin, Surya, Arun, Abhi, David, Obul, Praveen and Raka. I will particularly miss those great debates between Apu and Sachin on virtually anything under the sky and Sree's culinary skills. I would also like to thank Nani mama, Prashu, Vijju, Ravi, Sandy and Preets for those lively times which I have had with them.

Finally, I wish to acknowledge the source of financial support that I received over the course of my studies. Support for this research was provided by the Faculty of Engineering Science and Mathematics, University of Southampton.

Nomenclature

All matrices and vectors are denoted in bold format

$\bar{d}$	Minimum distance between any two centres
$\bar{U}$	Average velocity
δt	Time step
δ	Support parameter of CSRBF
$\delta(\boldsymbol{x}, t)$	Residual error function for CD equation
δ_{ij}	Overlap between sub-domains Ω_i and Ω_j
$\frac{\mathrm{D}}{\mathrm{D}t}$	Material derivative
κ	Diffusion coefficient
λ	Radial basis function coefficient
λ_M	Eigenvalue of matrix $\boldsymbol{M}$
$\lceil . \rceil$	Ceiling function
$\lfloor . \rfloor$	Floor function
$\boldsymbol{A}$	Gram matrix
$\mathbb{N}$	Set of natural numbers
$\mathcal{B}$	Boundary operator
$\mathcal{B}^c\phi(\|\boldsymbol{x} - \boldsymbol{c}\|)$	Operator $\mathcal{B}$ applied on the RBF as a function of the second argument $\boldsymbol{c}$
$\mathcal{B}^x\phi(\|\boldsymbol{x} - \boldsymbol{c}\|)$	Operator $\mathcal{B}$ applied on the RBF as a function of the first argument $\boldsymbol{x}$

$\mathcal{C}$ Set of collocation points

$\mathcal{L}$ Arbitrary differential operator

$\mathcal{L}^c\phi(\|\boldsymbol{x}-\boldsymbol{c}\|)$ Operator $\mathcal{L}$ applied on the RBF as a function of the second argument $\boldsymbol{c}$

$\mathcal{L}^x\phi(\|\boldsymbol{x}-\boldsymbol{c}\|)$ Operator $\mathcal{L}$ applied on the RBF as a function of the first argument $\boldsymbol{x}$

$\mathcal{O}(.)$ Computational complexity

$\mathcal{S}$ Artificial boundary operator

Re Reynolds number

∇ Gradient operator

∇^2 Laplacian operator

$\nabla^2\nabla^2$ Biharmonic operator

$\|.\|$ Euclidean norm

$\|.\|_\infty$ Infinity norm

ν Convergence rate

Ω Closed domain in physical space

ω Vorticity

$\phi(.)$ Radial basis function in $\mathbb{R}^d$

Π^d_{m-1} Space of polynomials of degree $\leq (m-1)$ in d variables

ψ Streamfunction

$\mathbb{R}$ Set of real numbers

$\mathbb{R}^+$ Set of positive real numbers

$\mathbb{R}^d$ Set of real numbers of dimension d

ρ Density

σ Shape parameter

σ_{crit} Critical shape parameter

$\boldsymbol{\delta}_{ij}$ Kronecker delta

$\boldsymbol{\lambda}$ RBF coefficient vector

$\boldsymbol{\Phi}$ Gram matrix

$\boldsymbol{\sigma}$ Cauchy stress tensor

$\boldsymbol{x}$ Spatial coordinates

$\vec{n}$ Outward normal vector

$\vec{v}$ Constant velocity vector

Ξ Viscous dissipation function

a,b,c Constants

C_v Specific heat at constant volume

$f(\boldsymbol{x})$,$g(\boldsymbol{x})$ Arbitrary functions in $\boldsymbol{x}$

G Number of ghost centres

h Mesh spacing

i, j, l, k Indices

L Characteristic Length

L Width of the channel in backward-facing step flow

L_x,L_y Scale parameters

N_B Number of boundary centres

n_b Number of boundary centres

n_d Number of interior centres

N_e Number of sub-domains

N_I Number of interior centres

p Unit cost of evaluating an RBF

$p(x)$ Polynomial in x

P_e Peclet number

q Heat flux vector

$Q(\boldsymbol{x}_i, \sigma)$ Cost functional at a node $\boldsymbol{x}_i$for a particular shape parameter σ

r Radial distance

$R(.)$ Residual function

T Temperature

t Time coordinate

t_f Final time

U Characteristic speed of flow

u,v Cartesian velocity components

BE Boundary Element

CD Convection Diffusion

CN Crank Nicholson

CSPH Corrected Smoothed Particle Hydrodynamics

CSRBFs Compactly Supported Radial Basis Functions

DDMs Domain Decomposition Methods

DE Diffuse Element

EFG Element Free Galerkin

FD Finite Difference

FE Finite Element

FP Finite Point

FV Finite Volume

GAU Gaussian

GSRBFs Globally Supported Radial Basis Functions

IMQ Inverse Multiquadric

IRBFN Integrated Radial Basis Function Networks

LOO Leave-One-Out

MLPG Meshless Local Petrov Galerkin

MLS Moving Least Squares

MQ Multiquadric

NS Navier-Stokes

PDEs Partial Differential Equations

PU Partition of Unity

QS Quintic Splines

RBF-DQ Radial Basis Function-Differential Quadrature

RBF-FD Radial Basis Function-Finite Difference

RBFs Radial Basis Functions

RKP Reproducing Kernel Particle

SCPD(m) Strictly Conditionally Positive Definite of order m

SPH Smoothed Particle Hydrodynamics

TPS Thin Plate Splines

Chapter 1

Introduction

1.1 Motivation

The behaviour of physical systems is generally governed by certain partial differential equations (PDEs). In Fluid Mechanics, the Navier-Stokes (NS) equations, developed independently by Navier and Stokes in 1920, form the central system of PDEs governing fluid flows. These equations are derived by satisfying the mass, momentum and energy conservation for an infinitesimal fluid element. The system of NS equations, supplemented by empirical laws for the dependence of viscosity and thermal conductivity on other flow variables and by a constitutive law defining the nature of the fluid, completely describe all flow phenomena.

Analytical solutions of the NS equations exist only for simple cases like the Poiseuille flow, Couette flow and certain other specific flows. In general, numerical methods are needed to predict fluid flows. Traditionally, Finite Difference (FD), Finite Element (FE), Finite Volume (FV) or Boundary Element (BE) methods are used for solving fluid problems (Hirsch, 1991; Brebbia, 1978; Zienkiewicz & Taylor, 2000). These methods are acknowledged to have achieved a high degree of sophistication and success in solving fluid flow problems. These computational methods are all based on a mesh discretisation (a subdivision of the spatial flow domain into numerous finite volume/elements/cells) that has to be generated in advance or dynamically modified as the solution progresses (adaptive meshing). The continuum NS equations are then approximated on these meshed points or volumes, leading to an algebraic linear or nonlinear system of equations. These system of equations are then solved by appropriate numerical methods to obtain the unknown solution. Although the above mentioned methods are highly popular, there exist some unresolved issues with these numerical methods like the awkward treatment of irregular boundary in FDM, the storage of huge data in FEM

and the difficulty of treating singularities and deriving fundamental solutions in BEM (Young *et al.*, 2004). In addition, the accuracy of the presented methods depends on the type of mesh that is used to discretise the physical domain (Tanaka, 1999). Also, typically with these methods only the function is continuous across the meshes, but not its partial derivatives. In practise, one uses a lower order polynomial for function approximation in these methods due to the polynomial snaking problem[1]. While higher order schemes are necessary for more accurate approximations of the spatial derivatives they are not sufficient without monotonicity constraints i.e., the approximation should be either increasing/decreasing or else remain constant. Because of the lower order schemes typically employed the spatial truncation errors can only be controlled by using progressively smaller meshes (Kansa, 1999).

Although significant advances have been made in the area of grid generation over the last few decades, it still remains a complex and time consuming process and, in many cases it can absorb far more time and cost than the numerical solution itself. Not only can mesh based methods be very complex and time consuming but for many flow problems such as free surface, moving boundaries, boundary layer, front tracking/shock (where mesh crossover/distortion is a significant problem), large deformations in materials, crack growth in materials etc., they are acknowledged to be not cost-effective due mainly to the use of the element structure/connectivity of the elements in the mesh (Zerroukat *et al.*, 2000). For example, in FEM, if the element is heavily distorted, shape functions for this element are of poor quality and thus the numerical results may not be acceptable (Wang & Liu, 2002).

In recent years, there has been an upsurge of interest in the development of so-called meshfree methods as an alternative to the mesh based methods. The term "meshfree" indicates the ability of a numerical simulation process being constructed entirely from a set of nodes which generally are randomly scattered through the domain of analysis and do not have any pre-specified connectivity between each other. These meshfree methods are also referred to in the literature as meshless, gridless, element free or cloud methods (Belytschko *et al.*, 1996).

The emergence of these methods in science and engineering (and in particular fluid/structural mechanics) are in their very early stages and have not yet reached the effectiveness and robustness of mesh based methods such as FD, FE, FV and BE methods. However they have major advantages in that a) No mesh structure is needed, b) they are very suitable for problems involving complicated or rapidly changing domain geometry, c) they are highly

[1]Polynomial snaking refers to the highly wiggled approximations obtained using higher order polynomials.

flexible and easily modified (addition and subtraction of nodes) without major implementation difficulties and can be easily extended to higher dimensions, d) these schemes can as well be applied on any kind of meshes or their hybrids. The aim of meshless methods for PDEs is to eliminate at least the structure of the mesh and approximate the solution entirely using the nodes/points (irregular random points) rather than the nodes of an element/grid based discretisation. Since only points are required, meshless methods offer great potential to accurately and efficiently solve fluid flow problems with complex configurations (Batina, 1993; Onate *et al.*, 1996; Shu *et al.*, 2003).

The earliest attempts on meshless methods was probably done by Perrone & Kao (1975); Liszka & Orkisz (1984). They introduced the generalised FD schemes on arbitrary grids. The present day meshfree methods can be grouped under the following divisions based on the kind of interpolation/approximation techniques utilised. The first is based on the Moving Least-Squares (MLS) technique. This type of interpolation technique is adopted by many popular meshfree methods like Element Free Galerkin (EFG) (Belytschko *et al.*, 1994; Lu *et al.*, 1994), Reproducing Kernel Particle (RKP) (Liu *et al.*, 1995), Partition of Unity (PU) (Melenk & Babuska, 1996; Griebel & Schweitzer, 2000), Finite Point (FP) (Onate *et al.*, 1996), Meshless Local Petrov-Galerkin (Atluri & Zhu, 1998) and Diffuse Element (DE) (Nayroles *et al.*, 1992) methods. The least squares technique allows an optimised approximation derived from an over-determined set of equations and generally the resultant coefficient matrix is symmetric and positive definite. However as the LS approximation does not pass through the nodal points, the essential boundary conditions cannot be imposed directly. Most of the above mentioned methods are based on the Galerkin projection and are not truly meshfree as they require an auxiliary mesh to perform integration with respect to space. The next one is motivated from statistical theory and Monte Carlo integrations. Examples of this kind are Smoothed Particle Hydrodynamics (SPH) (Moraghan, 1982, 1988; Gingold & Moraghan, 1997), Corrected Smooth Particle Hydrodynamics (CSPH) (Kulasegaram *et al.*, 2000). SPH is used to model collision and explosion of stars and is well suited for rapidly expanding computational domains. However, the application of these methods is limited to unbounded domains and they perform badly for bounded domains. For an overview and a comparative study of some of these methods see Duarte (1995); Belytschko *et al.* (1996); Fries & Matthies (2003).

The other type of meshfree methods are based on functional approximation using Radial Basis Functions (RBFs). For many years, RBFs have been synonymous with scattered data

interpolation especially in higher dimensions (Franke, 1982; Wendland, 1995). The excellent performance of RBFs for scattered data interpolation motivates their use in developing meshfree schemes for solving PDEs. The Unsymmetric RBF collocation (Kansa, 1990a,b), Symmetric RBF collocation (Fasshauer, 1996), Dual Reciprocity (Chen *et al.*, 1998), and DRM-MFS-RBF method (Golberg *et al.*, 1999), RBF-DQ (Shu *et al.*, 2003) are some of the RBF based meshfree methods available in the literature for solving PDEs.

As all the meshfree schemes are point based schemes, they must possess some essential properties in order for them to be applicable to practical problems. The sensitivity of these schemes to a variable number of points in each interpolation domain must be low enough to preserve the freedom of adding, moving or removing points. This sensitivity is very high in meshfree techniques using the LS approximation (Fries & Matthies, 2003) as compared to RBF based approximations. Also, as RBFs are univariate functions, RBF based schemes can be easily extended to higher dimensions. In addition RBF based schemes have the advantage of being truly meshfree as compared to some of the meshless or element free methods where some kind of auxiliary grid is needed and thus eliminating many of the advantages of the meshfree philosophy. Moreover, any RBF based scheme generally has higher-order accuracy than the standard FD schemes on scattered nodes. More specifically RBF based methods seem to have exponential convergence rates (Cheng *et al.*, 2003; Boztosun & Charafi, 2002; Hon & Mao, 1998). The above mentioned advantages of RBFs leads us to choose RBFs as the basis for developing meshfree schemes for fluid flow problems.

1.2 Scope and objectives

The present research focuses on the development of meshfree methods for fluid flow problems. The primary objective of the present research is to develop RBF based meshless techniques for a wide class of fluid flow problems with a particular emphasis on time-dependent problems. We aim to develop computationally efficient procedures for use in large scale problems so that the developed algorithms are suitable for industrial applications.

The approach used here aims to achieve this objective by systematically solving time-dependent linear and nonlinear problems. The first goal of this research is to develop and validate meshless methods based on RBFs for model flow problems. Next, the aim is to develop novel algorithms to tackle large scale problems. Subsequently, the aim is to integrate all the developed algorithms to develop a robust meshfree solver for fluid flow problems. The

scope of the present research in these directions is summarised below:

1.2.1 Comparison of various RBFs for solving PDEs

It has been shown in the literature that RBFs perform very well for solving a wide class of PDEs (Hon *et al.*, 1997; Zerroukat *et al.*, 1998; Hon & Mao, 1998). Extensive studies on the performance of different RBFs for scattered data interpolation can be found in Franke (1982). However, no study has been done as to how different RBFs perform when applied to the solution of PDEs. The present research aims to bridge this gap by performing extensive numerical studies on a model flow problem. The behaviour of fluid flows, in general, is governed by two components viz., convection and diffusion. Hence the model problem chosen is the unsteady convection-diffusion problem. The convergence behaviour of various globally supported and compactly supported RBFs are examined for both uniform as well as scattered set of points. Also, a symmetric RBF collocation method is proposed for time-dependent problems. We show that the symmetric RBF method is capable of obtaining good results with the additional advantage of the coefficient matrix being (anti)-symmetric and less ill-conditioned as compared to the unsymmetric RBF collocation method.

1.2.2 Large scale problems: Domain decomposition

Although global RBF methods have been shown to have very high convergence rates, the resulting coefficient matrix is dense and becomes highly ill-conditioned. Possible ways of circumventing this ill-conditioning problem have been suggested in Kansa & Hon (2000). Also, as the coefficient matrix is dense this hinders the application of RBFs to solve large scale fluid dynamics problems, as they are computationally intensive when a large number of collocation points are used. In this thesis, we propose two different overlapping domain decomposition techniques to solve time-dependent and nonlinear problems. As the physical domain is divided into small sub-domains, we obtain small coefficient matrices which can be easily solved instead of a large dense matrix, thus making the proposed methods computationally efficient. The condition numbers of the resultant sub-domain coefficient matrices are smaller as compared to the single domain coefficient matrices. Further, these schemes allow for efficient parallelisation. All the above mentioned advantages render the proposed RBF based domain decomposition algorithms suitable for application to large scale problems.

1.2.3 Large scale problems: RBFs in finite difference mode

An alternative approach involving the use of RBFs to construct finite difference approximations (RBF-FD) is explored in this thesis. The weights of the RBF-FD method are obtained by solving local RBF interpolation problems set-up around each node in the computational domain. A key advantage of this method is that the resulting coefficient matrices are sparse and hence it can be applied to solve large scale problems. An optimisation strategy based on the statistical leave-one-out criterion is applied to obtain the optimal value of the shape parameter. A novel ghost node strategy is proposed for satisfying boundary conditions. This strategy preserves the freedom of having completely random distribution of nodes in the domain of interest. Finally, a higher order discretisation of the RBF-FD method is also developed for application to fluid problems using the RBF Hermite interpolation technique.

1.2.4 Applications to incompressible Navier Stokes equations

The developed methodologies are extended for solving the steady and unsteady incompressible Navier-Stokes equations. We consider the driven cavity flows and the backward facing step flow which occur in a variety of industrial applications. These problems have been studied throughout the literature and serve as benchmarks for testing new algorithms. Numerical results are obtained for both uniform and scattered distribution of points.

1.3 Layout of the thesis

The remainder of this thesis is arranged as follows:

Chapter 2 presents a brief overview of RBFs. The properties of RBFs and its application to solving interpolation problems is discussed. Earlier work on solution of PDEs using RBFs is also summarised. The various theoretical and computational issues governing RBF based methods are discussed.

Chapter 3 presents detailed numerical studies on a model linear time-dependent PDE. The symmetric RBF method is extended for solving time-dependent PDEs. The stability analysis of the methods is also presented. The convergence properties of different RBFs are studied when applied to solve the convection-diffusion equation.

Chapter 4 introduces domain decomposition methods for solving large scale problems. Meshfree overlapping Schwarz schemes are proposed for time-dependent problems. These methods are shown to be computationally more efficient than the single domain RBF collo-

cation method. Also, the proposed domain decomposition schemes are shown to reduce the ill-conditioning problem present in RBF schemes.

Chapter 5 introduces the application of radial basis functions in a finite difference mode (RBF-FD). The coefficient matrices obtained are sparse and hence are suitable for application to large scale problems. A leave-one-out statistical criterion is employed as the objective function for optimisation of the shape parameter. Model Poisson and convection-diffusion equations are solved to investigate the performance of this method.

Chapter 6 deals with meshless methods using RBF collocation for the incompressible Navier-Stokes equations. The steady state equations are considered and the test problems of driven cavity flows and backward facing step flow are solved. The merits and demerits of RBF collocation methods are listed.

Chapter 7 demonstrates the suitability of the RBF-FD for solving the incompressible Navier-Stokes equations. A ghost node strategy is employed for satisfying no-slip boundary conditions. Finally, a higher order RBF-FD method using Hermite interpolation techniques is presented for solving the Navier-Stokes equations.

Chapter 8 summarises the contributions and major conclusions of this research. Some directions for future research are also outlined.

Chapter 2

Overview of radial basis functions

In this chapter, we present a brief overview of radial basis functions (RBFs) and their applications in scientific computing. The definitions of RBFs are introduced and interpolation of scattered data using RBFs is discussed. Then we show how RBFs can be used in the numerical solution of partial differential equations by the method of collocation. Some theoretical and computational aspects which arise when using RBFs for developing meshfree methods are also elaborated.

2.1 Radial basis functions

We first present the following definitions for continuous functions.

Definition 2.1. A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called a radial basis function if $\phi(\boldsymbol{x}) = \phi(\boldsymbol{y})$ whenever $\|\boldsymbol{x}\| = \|\boldsymbol{y}\|$ where $\|.\|$ denotes the Euclidean norm and $\mathbb{R}^d$ denotes the d-dimensional space on $\mathbb{R}$ and $\boldsymbol{x}$,$\boldsymbol{y} \in \mathbb{R}^d$.

Definition 2.2. A function $\phi : \mathbb{R}^+ \to \mathbb{R}$ is strictly conditionally positive definite of order m (SCPD(m)) if for every set of distinct data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \subset \mathbb{R}^d$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) > 0$$

for all $\lambda_1, \ldots, \lambda_N$ satisfying,

$$\sum_{i=1}^{N} \lambda_i p(\boldsymbol{x}_i) = 0,$$

for all polynomials p of degree less than m.

Table 2.1 lists some globally supported RBFs (GSRBF) that are commonly used in the literature. The Euclidean norm is denoted by $r = \|\cdot\|$ and σ is a shape parameter. The

influence of GSRBFs extends from $-\infty$ to $+\infty$. The last column of Table 2.1 indicates the strictly conditionally positive definiteness order of each GSRBF.

TABLE 2.1: Globally supported radial basis functions

$\phi(r) = r^{2\beta} \log r, \beta \in \mathbb{N}$	thin plate splines (TPS)	SCPD$(\beta + 1)$
$\phi(r) = (r^2 + \sigma^2)^{\frac{1}{2}}$	multiquadrics (MQ)	SCPD(1)
$\phi(r) = (r^2 + \sigma^2)^{\frac{-1}{2}}$	inverse multiquadrics (IMQ)	SCPD(0)
$\phi(r) = e^{-\frac{r^2}{\sigma}}$	Gaussians (GAU)	SCPD(0)
$\phi(r) = r^{\beta},\ \beta > 0, \beta \notin 2\mathbb{N}$	quintic splines (QS)	SCPD$(\lceil \beta/2 \rceil)$

Figure 2.1 shows a number of globally supported RBFs. We plot the Gaussian RBFs with increasing shape parameter values in the first row of the figure. It can be seen that for the Gaussians, increasing the value of σ leads to flatter and flatter RBFs. Similar trends can be observed for MQ and IMQ, as is apparent from second and third rows of Figure 2.1. In the next two rows, we plot the different TPS and QS RBFs. Throughout this thesis all RBFs which incorporate a shape parameter in their definition like multiquadrics, inverse multiquadrics and Gaussians are collectively referred to as σ-tunable RBFs.

Another class of RBFs, known as compactly supported radial basis functions (CSRBFs), due to Wendland (Wendland, 1995), Wu (Wu, 1995) and Buhmann (Buhmann, 2000) are also used. The central idea of CSRBFs is to use a polynomial as a function of r with support on $[0, 1]$. CSRBFs must be strictly positive definite in $\mathbb{R}^d$ for all d less than or equal to some fixed value d_0. The basic definition of the CSRBF $\phi_{l,k}(r)$ have the form

$$\phi_{l,k}(r) = (1 - r)_+^n p(r), \text{ for } k \geq 1, \tag{2.1}$$

with the following conditions

$$(1 - r)_+^n = \begin{cases} (1 - r)^n & \text{if } 0 \leq r < 1, \\ 0 & \text{if } r \geq 1, \end{cases}$$

where $l = \lfloor \frac{d}{2} \rfloor + k + 1$ is the dimension number, $2k$ is the smoothness[1] of the function and $p(r)$ is a prescribed polynomial. Table 2.2 lists out some of the Wendland CSRBFs generally used in the literature when $d = 3$.

[1]A function is said to have smoothness C^m if all its derivatives up to order m exist and are continuous functions.

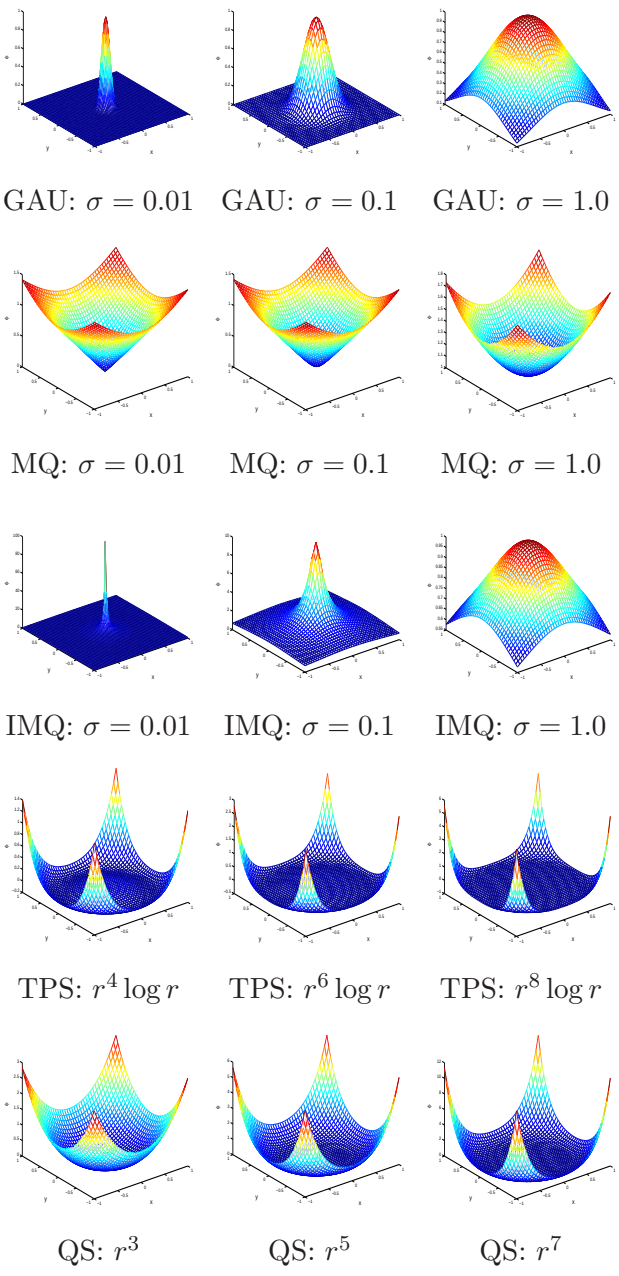


FIGURE 2.1: Globally supported RBFs (The functions are plotted in the $[-1, 1] \times [-1, 1]$ region)

TABLE 2.2: Compactly supported radial basis functions (Wendland, 1995)

RBF	Smoothness
$\phi_{2,0}(r) = (1-r)_+^2$	C^0
$\phi_{3,1}(r) = (1-r)_+^4(4r+1)$	C^2
$\phi_{4,2}(r) = (1-r)_+^6(35r^2+18r+3)$	C^4
$\phi_{5,3}(r) = (1-r)_+^8(32r^3+25r^2+8r+1)$	C^6

Note that unlike GSRBFs, the influence of CSRBFs is local in $[0,1]$ and the influence vanishes on $[1,\infty)$. Also, we can scale a basis function with compact support on $[0,\delta]$ by replacing r with $\frac{r}{\delta}$ where δ is referred to as the support parameter of the CSRBF.

2.2 RBF interpolation

RBFs are widely used for scattered data interpolation (Wendland, 1995). The problem of multivariate interpolation can be stated as follows:

Problem 2.3. Given data $(\boldsymbol{x}_j, f_j),\ j = 1,\ldots,N$ with $\boldsymbol{x}_j \in \mathbb{R}^d,\ f_j \in \mathbb{R}$ find a continuous function $S(\boldsymbol{x})$, such that $S(\boldsymbol{x}_j) = f_j,\ j = 1,\ldots,N$.

The function $S(\boldsymbol{x})$ is assumed to be given by a linear combination of RBFs, i.e.,

$$S(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) + p(\boldsymbol{x}), \tag{2.2}$$

where $\phi(\|\boldsymbol{x}-\boldsymbol{x}_j\|)$ is a RBF centred on the point $\boldsymbol{x}_j$, $p(\boldsymbol{x}) = \sum_{k=1}^{M} \gamma_k p_k(\boldsymbol{x})$ where $p_1(\boldsymbol{x}),\ldots,p_M(\boldsymbol{x})$ form a basis for the $M = \begin{pmatrix} d+m-1 \\ m-1 \end{pmatrix}$-dimensional linear space Π_{m-1}^d of polynomials of total degree less than or equal to $m-1$ in d variables, and $\{\lambda_j\}_{j=1}^N$ are the unknown RBF coefficients. The following conditions are imposed on the approximation $S(\boldsymbol{x})$,

- $S(\boldsymbol{x}_j) = f_j, \quad \forall j = 1,2,\ldots,N,$
- $\sum_{j=1}^{N} \lambda_j p_k(\boldsymbol{x}_j) = 0, \quad \forall k = 1,2,\ldots,M,$

leading to a system of equations which can be written down in matrix form as

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{Bmatrix}, \tag{2.3}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is referred to as the Gram matrix, $\boldsymbol{\lambda} \in \mathbb{R}^N$ is the undetermined coefficient vector and $\boldsymbol{f} \in \mathbb{R}^N$ is the vector of function values. For sake of clarity, the matrix $\boldsymbol{A}$ in expanded form is given below

$$\boldsymbol{A} = \begin{bmatrix} \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_N\|) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\|\boldsymbol{x}_N - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_N - \boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_N - \boldsymbol{x}_N\|) \end{bmatrix}. \tag{2.4}$$

Now from Equation (2.3), it can be seen that Problem 2.3 is well-posed if and only if the coefficient matrix is non-singular, i.e., its inverse exists. Micchelli (1986) proved that the interpolation problem in Equation (2.3) is solvable when the following two conditions are met: (1) the set of points $\{\boldsymbol{x}_j\}_{j=1}^N$ are distinct, and (2) the degree of the appended polynomial is chosen to be the order of strictly conditionally positive definiteness of the RBF used.

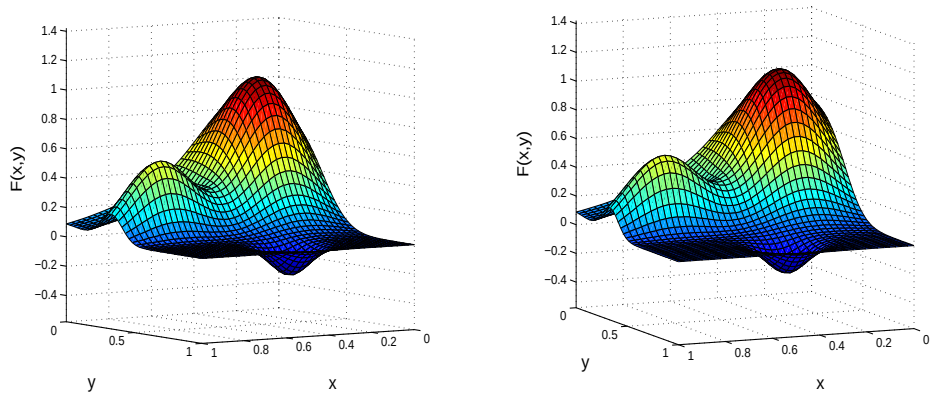
Franke (1982) performed numerical experiments using 29 different interpolation methods on two-dimensional analytic functions and found out that the RBF interpolation technique using multiquadrics outperformed all other interpolation techniques. Madych & Nelson (1989) proved that interpolation with MQ is exponentially convergent based on reproducing kernel Hilbert spaces. Similar results were obtained by Buhmann (1990); Wu & Schaback (1993). The error estimates for other RBFs can be found in Wendland (1997). In fact, it was proved that the RBF interpolant is the best Hilbert-space approximation to the interpolation data in Sun (1994). We now demonstrate the ability of RBF interpolation by approximating Franke's function (Franke, 1982),

$$\begin{aligned} F(x,y) \;=\; & \tfrac{3}{4}\left[\exp\left(-\tfrac{(9x-2)^2}{4} - \tfrac{(9y-2)^2}{4}\right) + \exp\left(-\tfrac{(9x+1)^2}{49} - \tfrac{(9y+1)^2}{10}\right)\right] + \\ & \tfrac{1}{2}\exp\left(-\tfrac{(9x-7)^2}{4} - (9y-3)^2\right) - \tfrac{1}{5}\exp\left(-(9x-4)^2 - (9y-7)^2\right). \end{aligned} \tag{2.5}$$

We use Gaussian RBFs and taking 20 data points, the results obtained are shown in Figure 2.2. The left hand side shows Franke's function and the right hand side shows the RBF interpolant. It can be seen that with just 20 data points, the RBF interpolant approximates Franke's function very closely.

We briefly review another method known as the Hermite-RBF interpolation technique. In this context we are given data $\{\boldsymbol{x}_i,\ \mathcal{L}_i f\}$, $i = 1, \ldots, N$, $\boldsymbol{x}_i \in \mathbb{R}^d$ where $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$ is a linearly independent set of continuous linear functionals. We try to find an interpolant of the form

$$S(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \mathcal{L}_j^c \phi(\|\boldsymbol{x} - \boldsymbol{c}\|), \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{2.6}$$



Franke's function RBF Interpolant

FIGURE 2.2: Comparison of the true function and RBF interpolant using Gaussian RBF

satisfying

$$\mathcal{L}_i S = \mathcal{L}_i f, \quad i = 1, \ldots, N$$

Here, $\mathcal{L}^c$ indicates the functional $\mathcal{L}$ acting on ϕ viewed as a function of the second argument $\boldsymbol{c}$. We then obtain a linear system of the form $\widehat{\boldsymbol{H}}_+ \boldsymbol{\lambda} = \mathcal{L}_i \boldsymbol{f}$ where the entries of the matrix are

$$\widehat{\boldsymbol{H}}_+(i,j) = \mathcal{L}_i \mathcal{L}_j^c \phi, \quad i,j = 1, \ldots, N. \tag{2.7}$$

The matrix $\widehat{\boldsymbol{H}}_+$ is guaranteed to be non-singular when $\phi(\|.\|)$ is positive definite and the data points $\{\boldsymbol{x}\}_{j=1}^N$ are distinct, see Wu (1998) for a detailed theoretical analysis.

Besides direct interpolation, RBFs are widely used in medical imaging (Carr *et al.*, 1997), surface reconstruction (Dinh *et al.*, 2002) and Neural networks (Park & Sandberg, 1991) to name a few.

2.3 RBF collocation for PDEs

In this section, we briefly outline how RBFs can be used for solving partial differential equations (PDEs). The main advantage of RBF based methods is that they are truly meshless schemes since only a scattered set of collocation points is used and no connectivity information is required. Also, since RBFs are dimension independent (ϕ is only a function of r), the extension of these meshless schemes to higher dimensions is straight forward. Generalising the concept of RBF interpolation (Equations (2.2-2.4)), if $\mathcal{L}$ is an interior linear partial

differential operator with some boundary operator $\mathcal{B}$, then an approximation $\widehat{u}(\boldsymbol{x})$ to the solution $u(\boldsymbol{x})$ of $\mathcal{L}u(\boldsymbol{x}) = f(\boldsymbol{x});\ \boldsymbol{x} \in \Omega,\ \ \mathcal{B}u(\boldsymbol{x}) = g(\boldsymbol{x});\ \boldsymbol{x} \in \partial\Omega$ can be obtained by letting $\widehat{u}(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|)$ (known as unsymmetric method (Kansa, 1990a)). The unknown RBF coefficients are obtained by satisfying

$$\begin{aligned} \mathcal{L}\widehat{u}(\boldsymbol{x}_i) &= f(\boldsymbol{x}_i) \quad 1 \leq i \leq n_d, \\ \mathcal{B}\widehat{u}(\boldsymbol{x}_i) &= g(\boldsymbol{x}_i) \quad n_d + 1 \leq i \leq N, \end{aligned} \tag{2.8}$$

where N is the total number of collocation points considered and n_d is the number of interior collocation points.

An alternative procedure, known as the symmetric RBF collocation method, and based on Hermite interpolation was proposed by Fasshauer (1996) and further studied by Franke & Schaback (1998). Here, the approximation of the function $\widehat{u}(\boldsymbol{x})$ is written as

$$\widehat{u}(\boldsymbol{x}) = \sum_{j=1}^{n_d} \lambda_j \mathcal{L}^c \phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|) + \sum_{j=n_d+1}^{N} \lambda_j \mathcal{B}^c \phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|) + p(\boldsymbol{x}), \tag{2.9}$$

where $\mathcal{L}^c$ is the interior differential operator $\mathcal{L}$ operated on the second argument (centres) of the RBF $\phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|)$ and $\mathcal{B}^c$ refers to the boundary operator similarly. The coefficient matrices generated here are (anti)-symmetric as opposed to the unsymmetric coefficient matrices generated by the earlier method.

2.4 Theoretical and computational aspects

In this section, we discuss the theoretical and computational issues encountered in RBF based methods. The RBF interpolation problem is well-posed provided one appends a polynomial of degree m to an RBF of SCPD(m) and the data points $\{\boldsymbol{x}_j\}$ are distinct, see Micchelli (1986). In the case of unsymmetric RBF collocation, the non-singularity of coefficient matrix, cannot be proved from RBF interpolation as the rows of the coefficient matrix are generated by two different operators (Fasshauer, 2005). Even though theoretical results for the unsymmetric method are scarce, this method is quite popular due to the inherent simplicity of implementation and its exponential convergence rate, see Kansa (1990a,b); Sharan *et al.* (1997). In Kansa (1999), it was suggested that if the centres of the RBF are distinct and the PDE problem is well-posed, the coefficient matrix is generally found to be non-singular. Occurrences of singular coefficient matrix are very rare (Hon & Schaback, 2001). Also note that the coefficient matrices generated are dense and hence require $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$

operations for factorisation. A variety of PDEs like initial value problems (Hon & Mao, 1997), tissue engineering problems (Hon *et al.*, 1997), Burger's equation (Hon & Mao, 1998), shallow water equations (Wong *et al.*, 1999) and financial problems (Hon & Mao, 1999) have been solved using unsymmetric RBF method. In comparison the symmetric method is well-posed and the non-singularity of the coefficient matrix has been proved in Wu (1998). Also, as one increases the number of RBF centres in a problem, the coefficient matrices of both methods tend to become highly-ill-conditioned. It is worth mentioning at this point that several proposals have been made to reduce this ill-conditioning (Kansa & Hon, 2000) like preconditioners, domain decomposition methods and block solvers. We address this issue in more detail in the latter part of the thesis. Finally, in the context of RBF interpolation, Beatson *et al.* (1999) showed that by recasting the RBFs into a different set of basis functions based on far field expansions of the RBFs leads to better conditioned matrices. This change in basis leads to approximate cardinal functions and lowers the computational cost of solving the interpolation problem to $\mathcal{O}(N \log N)$ operations. For more details on cardinal functions, the reader is referred to Buhmann & Micchelli (1992); Baxter (1992).

We now turn our attention to the issue of selecting the shape parameter in σ-tunable RBFs. In the case of MQ interpolation, Madych (1992) showed that the convergence rate is $\mathcal{O}(\nu^{\frac{\sigma}{h}})$, where $\nu < 1$ and h is the average distance between pairs of data centres. In case of σ-tunable RBFs, the accuracy of the RBF interpolant increases as the shape parameter σ is increased. However, this increase in σ tends to make the Gram matrix highly ill-conditioned. This condition is referred to as Schaback's *uncertainty principle* (Schaback, 1995). It is found that the RBF approximations are more accurate at the verge of ill-conditioning. Numerical studies by Cheng *et al.* (2003) have shown that when MQ RBF is applied to elliptic PDEs, the convergence rate of the unsymmetric method is $\mathcal{O}(\xi^{\frac{\sigma}{h}})$, where ξ is a constant.

From the above presented facts, it can be clearly seen that the shape parameter influences the accuracy profoundly. Unfortunately, no theoretical results are present in the literature which strongly suggest guidelines on what value of σ should be used. In RBF interpolation methods, the value of σ can be estimated by statistical techniques like the leave-one out procedure . In this procedure, one constructs the RBF interpolant using $N-1$ data points and then predicts the function value at the N^{th} point. The value of σ is then re-adjusted so that the predicted function value is the same as the known function value at $\boldsymbol{x}_N$. More details about how to implement computationally efficient procedures to estimate the value of σ for interpolation and regression problems can be found in Keane & Nair (2005). This

procedure has been later developed to obtain the optimal shape parameter for each stencil in the RBF-FD method. However, such a statistical procedure cannot be directly applied for RBF collocation method for PDEs. Empirical studies have been made by researchers in the past to obtain a suitable value for the shape parameter (see Franke (1982); Hardy (1990); Carlson & Foley (1991); Rippa (1999)). Hardy (1971) suggested the value of the shape parameter σ for MQ RBF to be kept as $0.815d_{\min}$, where $d_{\min}$ is the minimum distance between any two centres in the domain. Kansa et al. (Kansa, 1990b; Kansa & Carlson, 1992) conducted numerical studies on the MQ RBF and came up with the following relation

$$\sigma_i^2 = \sigma_{\min}^2 (\frac{\sigma_{\max}^2}{\sigma_{\min}^2})^{(\frac{i-1}{N-1})} ,$$

where $\sigma_{\min}^2$ and $\sigma_{\max}^2$ are preset parameters, N is the total number of data centres and i denotes the index of the point where the RBF is centred. It has been shown that using variable shape parameters leads to better conditioning of the coefficient matrix. However, these results are specific to MQ and cannot be generalised to other RBFs. In this thesis for RBF collocation methods, we propose an optimisation strategy for obtaining the shape parameter using a suitable norm of the residual as the objective function, for time-dependent PDEs.

Chapter 3

RBF Collocation Schemes

In this chapter, we present a series of numerical experiments conducted on the unsteady convection-diffusion equation (CD). The convection-diffusion equation is widely used to model a variety of physical, chemical, economical and financial forecasting processes to name a few (Roos *et al.*, 1996). The peculiarity of this equation is that it represents the coupling of two different phenomena, convection and diffusion. It also serves as a simplified model problem to the Navier-Stokes equation in fluid dynamics. One major difficulty when solving this problem arises from the fact that when the convective term dominates, the approximation can be contaminated due to spurious oscillation and numerical diffusion (Morton, 1995). The governing equation is parabolic for diffusion dominated cases and turns hyperbolic for convection dominated cases. Traditionally, Finite Difference (FD) and Finite Element (FE) schemes have been utilised to solve the convection-diffusion equation. These schemes work well for diffusion dominated problems. However, when the convective term dominates, special methods with artificial viscosity, upwinding etc., have to be used to stabilise the numerical scheme (Ferziger & Peric, 1999; Zienkiewicz & Taylor, 2000). All the above schemes are grid based schemes which need a discretisation of the domain into elements, which in itself can be a non-trivial task for complicated domains. Previous work focusing on the solution of the steady and unsteady convection-diffusion equations using RBFs can be found in the literature; see, for instance, Kansa (1990b); Boztosun & Charafi (2001); Power & Barraco (2002); Li & Chen (2003); Zerroukat *et al.* (2000); Boztosun *et al.* (2002); Boztosun & Charafi (2002) and the references therein.

In this chapter we present a numerical study to investigate the performance of different RBFs for the unsteady CD equation (Chinchapatnam *et al.*, 2006b). We also develop a new symmetric collocation scheme for time-dependent problems and compare it with the

unsymmetric scheme. The stability analysis of both unsymmetric and symmetric schemes for explicit as well as Crank Nicholson (CN) time-stepping are presented. Finally, numerical results are presented for one-dimensional and two-dimensional problems to compare the performance of the unsymmetric and symmetric collocation techniques. More specifically, we compare the performance of Gaussian ($e^{-r^2/\sigma}$), MQ ($(1+\frac{r^2}{\sigma})^{\frac{1}{2}}$), IMQ ($(r^2+\sigma^2)^{\frac{-1}{2}}$), TPS ($r^4 \log r$, $r^6 \log r$, $r^8 \log r$) and Quintics (r^7). Numerical studies suggest that symmetric collocation is only marginally better than the unsymmetric approach. Further it appears that both collocation techniques require a very dense set of collocation points in order to achieve accurate results for convection dominated cases.

3.1 Formulations

In this section, we present unsymmetric and symmetric collocation schemes using RBFs to spatially discretise the unsteady convection-diffusion equation. We also present a θ-weighted time stepping scheme for temporal discretisation.

Consider an unsteady convection-diffusion equation of the form

$$\frac{\partial u(\boldsymbol{x},t)}{\partial t} + \mathcal{L}u(\boldsymbol{x},t) = f(\boldsymbol{x},t) \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, t > 0, \tag{3.1}$$

where $\mathcal{L}$ is the convection-diffusion operator, of the form

$$\mathcal{L} = \left(\kappa\nabla^2 + V \cdot \nabla\right), \tag{3.2}$$

where ∇^2 and ∇ denote the Laplacian and the gradient operator, respectively. The diffusion coefficient is denoted by κ; V is a constant velocity vector and $u(\boldsymbol{x},t)$ represents a potential function. The CD equation is solved on a bounded physical domain Ω where $\partial\Omega$ denotes its boundary and $f(x,t)$ is a known function.

Equation (3.1) has to be supplemented with an initial condition of the form

$$u(\boldsymbol{x},t) = u_0(\boldsymbol{x}), \quad t = 0, \tag{3.3}$$

and with a boundary condition given by

$$\mathcal{B}u(\boldsymbol{x},t) = g(\boldsymbol{x},t), \quad t > 0, \tag{3.4}$$

where $\mathcal{B}$ can be a Dirichlet, Neumann or a mixed boundary operator; $u_0(\boldsymbol{x})$ and $g(\boldsymbol{x},t)$ are known functions.

A dimensionless number known as the Peclet number defined by $P_e = \frac{LV}{\kappa}$, where L is a characteristic length, relates the convection phenomenon to the diffusion phenomenon for the CD equation. When the Peclet number is high ($\approx P_e > 50.0$), the convection term dominates and when the Peclet number is low ($\approx P_e < 1.0$) the diffusion term dominates.

3.1.1 Unsymmetric RBF-Theta collocation scheme

In the unsymmetric scheme, the solution $u(\boldsymbol{x}, t)$ is approximated by a linear combination of RBFs as

$$u(\boldsymbol{x}, t) = \sum_{j=1}^{N} \lambda_j(t)\, \phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|), \tag{3.5}$$

where $\phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|)\colon \mathbb{R}^d \to \mathbb{R}$ is a RBF with centre $\boldsymbol{c}_j \in \mathbb{R}^d$. $\lambda_j(t)$, $j = 1, 2, \ldots, N$ are undetermined RBF coefficients which evolve with time.

The centres of the RBFs used in Equation (3.5) are chosen from a cloud of points situated within the domain Ω and on the boundary $\partial\Omega$, i.e., $\mathcal{C} = \{(\boldsymbol{c}_i)|_{i=1,n_d} \in \Omega, (\boldsymbol{c}_i)|_{i=n_d+1,n_d+n_b} \in \partial\Omega\}$, where n_d and n_b denote the number of centres inside the domain and on the boundary, respectively. Henceforth, we shall denote the total number of centres as N ($N = n_d + n_b$). For simplicity of presentation, consider the case when the set $\mathcal{C}$ coincides with the set of collocation points. Substituting Equation (3.5) in the governing Equation (3.1) and in the boundary conditions leads to

$$\sum_{j=1}^{N} \frac{\mathrm{d}\lambda_j}{\mathrm{d}t} \phi(\|\boldsymbol{x}_i - \boldsymbol{c}_j\|) = f_i(t) - \sum_{j=1}^{N} \lambda_j \mathcal{L}^x \phi(\|\boldsymbol{x}_i - \boldsymbol{c}_j\|) \qquad i = 1, \ldots, n_d, \tag{3.6}$$

$$\sum_{j=1}^{N} \mathcal{B}^x \phi(\|\boldsymbol{x}_i - \boldsymbol{c}_j\|) = g_i(t) \qquad i = n_d + 1, \ldots, n_d + n_b, \tag{3.7}$$

where $\mathcal{L}^x \phi(\|\boldsymbol{x}_i - \boldsymbol{c}_j\|)$ and $\mathcal{B}^x \phi(\|\boldsymbol{x}_i - \boldsymbol{c}_j\|)$ denote the application of the convection-diffusion and boundary operators on the RBF $\phi(\|\boldsymbol{x} - \boldsymbol{c}\|)$ as a function of the first variable i.e., $\boldsymbol{x}$ and evaluated at $\boldsymbol{x}_i$.

Equations (3.6) and (3.7) can be rewritten in matrix form as

$$\boldsymbol{\Phi}_d \frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} = \boldsymbol{f} - \mathcal{L}^x \boldsymbol{\Phi}_d \boldsymbol{\lambda}, \tag{3.8}$$

$$\mathcal{B}^x \boldsymbol{\Phi}_b \boldsymbol{\lambda} = \boldsymbol{g}, \tag{3.9}$$

where $\boldsymbol{\Phi}_d$, $\mathcal{L}^x \boldsymbol{\Phi}_d \in \mathbb{R}^{n_d \times N}$, $\boldsymbol{\lambda} \in \mathbb{R}^N$, $\boldsymbol{f} \in \mathbb{R}^{n_d}$, $\mathcal{B}^x \boldsymbol{\Phi}_b \in \mathbb{R}^{n_b \times N}$, and $\boldsymbol{g} \in \mathbb{R}^{n_b}$. For sake of

clarity, the matrix $\mathbf{\Phi_d}$ and $\mathbf{\Phi_b}$ can be written in expanded form as

$$\boldsymbol{\Phi}_d = \begin{bmatrix} \phi(\|\boldsymbol{x}_1 - \boldsymbol{c}_1\|) & \cdots & \phi(\|\boldsymbol{x}_1 - \boldsymbol{c}_{n_d}\|) & \cdots & \phi(\|\boldsymbol{x}_1 - \boldsymbol{c}_N\|) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \phi(\|\boldsymbol{x}_{n_d} - \boldsymbol{c}_1\|) & \cdots & \phi(\|\boldsymbol{x}_{n_d} - \boldsymbol{c}_{n_d}\|) & \cdots & \phi(\|\boldsymbol{x}_{n_d} - \boldsymbol{c}_N\|) \end{bmatrix} \in \mathbb{R}^{n_d \times N},$$

$$\boldsymbol{\Phi}_b = \begin{bmatrix} \phi(\|\boldsymbol{x}_{n_d+1} - \boldsymbol{c}_1\|) & \cdots & \phi(\|\boldsymbol{x}_{n_d+1} - \boldsymbol{c}_{n_d}\|) & \cdots & \phi(\|\boldsymbol{x}_{n_d+1} - \boldsymbol{c}_N\|) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \phi(\|\boldsymbol{x}_{n_d+n_b} - \boldsymbol{c}_1\|) & \cdots & \phi(\|\boldsymbol{x}_{n_d+n_b} - \boldsymbol{c}_{n_d}\|) & \cdots & \phi(\|\boldsymbol{x}_{n_d+n_b} - \boldsymbol{c}_N\|) \end{bmatrix} \in \mathbb{R}^{n_b \times N},$$

and the vectors $\boldsymbol{f}$, $\boldsymbol{\lambda}$ and $\boldsymbol{g}$ in their expanded form read as

$$\begin{aligned} \boldsymbol{f} &= [f_1\ f_2 \cdots f_{n_d}]^{\mathrm{T}} && \in \mathbb{R}^{n_d}, \\ \boldsymbol{\lambda} &= [\lambda_1\ \lambda_2 \cdots \lambda_{n_d} \cdots \lambda_N]^{\mathrm{T}} && \in \mathbb{R}^{N}, \\ \boldsymbol{g} &= [g_1\ g_2 \cdots g_{n_b}]^{\mathrm{T}} && \in \mathbb{R}^{n_b}. \end{aligned}$$

Using the notation $\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}(t^{n+1})$, where $t^{n+1} = t^n + \delta t$ and introducing θ-weighting $(0 \leq \theta \leq 1)$, we get

$$\boldsymbol{\Phi}_d\{\frac{\boldsymbol{\lambda}^{n+1} - \boldsymbol{\lambda}^n}{\delta t}\} = \boldsymbol{f}^{n+1} - \{\theta\ \mathcal{L}^x\boldsymbol{\Phi}_d\boldsymbol{\lambda}^{n+1} + (1-\theta)\ \mathcal{L}^x\boldsymbol{\Phi}_d\boldsymbol{\lambda}^{\boldsymbol{n}}\}, \tag{3.10}$$

$$\mathcal{B}^x\boldsymbol{\Phi}_b\boldsymbol{\lambda}^{n+1} = \boldsymbol{g}^{n+1}. \tag{3.11}$$

Equations (3.10) and (3.11) can be combined as follows

$$\begin{bmatrix} \boldsymbol{\Phi}_d + \theta\ \delta t\ \mathcal{L}^x\boldsymbol{\Phi}_d \\ \mathcal{B}^x\boldsymbol{\Phi}_b \end{bmatrix} \boldsymbol{\lambda}^{n+1} = \begin{bmatrix} \boldsymbol{\Phi}_d - (1-\theta)\ \delta t\ \mathcal{L}^x\boldsymbol{\Phi}_d \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda}^n + \begin{bmatrix} \delta t\ \boldsymbol{f}^{n+1} \\ \boldsymbol{g}^{n+1} \end{bmatrix}, \tag{3.12}$$

where $\boldsymbol{0} \in \mathbb{R}^{n_b \times N}$.

Equation (3.12) can be rewritten in compact form as

$$\boldsymbol{\lambda}^{n+1} = \boldsymbol{H}_+^{-1}\boldsymbol{H}_-\boldsymbol{\lambda}^n + \boldsymbol{H}_+^{-1}\boldsymbol{F}^{n+1}, \tag{3.13}$$

where

$$\boldsymbol{H}_+ = \begin{bmatrix} \boldsymbol{\Phi}_d + \theta\ \delta t\ \mathcal{L}^x\boldsymbol{\Phi}_d \\ \mathcal{B}^x\boldsymbol{\Phi}_b \end{bmatrix},\ \boldsymbol{H}_- = \begin{bmatrix} \boldsymbol{\Phi}_d - (1-\theta)\ \delta t\ \mathcal{L}^x\boldsymbol{\Phi}_d \\ \boldsymbol{0} \end{bmatrix}$$

and

$$\boldsymbol{F}^{n+1} = \begin{bmatrix} \delta t\ \boldsymbol{f}^{n+1} \\ \boldsymbol{g}^{n+1} \end{bmatrix}.$$

Equation (3.5) applied for all interior and boundary collocation points can be written in matrix form as

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{\lambda}, \tag{3.14}$$

where $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{\Phi}_d \\ \boldsymbol{\Phi}_b \end{bmatrix} \in \mathbb{R}^{N\times N}$ denotes the Gram matrix.

Using Equation (3.14), Equation (3.13) can be written in terms of the discrete values of the field variable as

$$\boldsymbol{u}^{n+1} = \boldsymbol{A}\boldsymbol{H}_+^{-1}\boldsymbol{H}_-\boldsymbol{A}^{-1}\boldsymbol{u}^n + \boldsymbol{A}\boldsymbol{H}_+^{-1}\boldsymbol{F}^{n+1}. \tag{3.15}$$

Since no theoretical proof exists for the invertibility of the matrix $\boldsymbol{H}_+$ when $\theta > 0$ (Hon & Schaback, 2001), it is not possible to show that unsymmetric collocation scheme is well posed for such cases. For the case of the explicit scheme with $\theta = 0$, only the Gram matrix $\boldsymbol{A}$ needs to be inverted. Provided the set of collocation points are distinct, the invertibility of this matrix can be guaranteed due to the result of Micchelli (1986).

3.1.2 Symmetric RBF-Theta collocation scheme

Next we present a symmetric collocation scheme for the unsteady convection-diffusion equation. In this scheme, as in the case of Fasshauer's method (Fasshauer, 1996), the potential function $u(\boldsymbol{x}, t)$ is approximated as

$$u(\boldsymbol{x},t) = \sum_{j=1}^{n_d} \lambda_j(t)\ \mathcal{L}^c\phi(\|\boldsymbol{x}-\boldsymbol{c}_j\|) + \sum_{j=n_d+1}^{n_d+n_b} \lambda_j(t)\ \mathcal{B}^c\phi(\|\boldsymbol{x}-\boldsymbol{c}_j\|) + \mathcal{P}^m(\boldsymbol{x}), \tag{3.16}$$

where $\mathcal{L}^c$ and $\mathcal{B}^c$ are operators applied on the RBF as a function of the second variable i.e., $\boldsymbol{c}$, and $\mathcal{P}^m(\boldsymbol{x})$ is a polynomial term of degree $m \leq N$.

For positive definite RBFs such as IMQ and Gaussian we do not need to add a polynomial term to guarantee invertibility. Hence, by letting $\mathcal{P}^m(\boldsymbol{x}) \equiv 0$ and substituting Equation (3.16) into the governing Equation (3.1) and boundary conditions and collocating on set $\mathcal{C}$, we obtain

$$\begin{aligned} \sum_{j=1}^{n_d} \frac{\mathrm{d}\lambda_j}{\mathrm{d}t}\mathcal{L}^c\phi(\|\boldsymbol{x}_i-\boldsymbol{c}_j\|) \quad &+ \quad \sum_{j=n_d+1}^{n_d+n_b} \frac{\mathrm{d}\lambda_j}{\mathrm{d}t}\mathcal{B}^c\phi(\|\boldsymbol{x}_i-\boldsymbol{c}_j\|) \\ &- \quad \sum_{j=1}^{n_d} \lambda_j\mathcal{L}^x\mathcal{L}^c\phi(\|\boldsymbol{x}_i-\boldsymbol{c}_j\|) \\ &- \quad \sum_{j=n_d+1}^{n_d+n_b} \lambda_j\mathcal{L}^x\mathcal{B}^c\phi(\|\boldsymbol{x}_i-\boldsymbol{c}_j\|) = f_i(t), \quad i = 1,\ldots,n_d, \end{aligned} \tag{3.17}$$

$$\sum_{j=1}^{n_d} \lambda_j\mathcal{B}^x\mathcal{L}^c\phi(\|\boldsymbol{x}_i-\boldsymbol{c}_j\|) + \sum_{j=n_d+1}^{n_d+n_b} \lambda_j\mathcal{B}^x\mathcal{B}^c\phi(\|\boldsymbol{x}_i-\boldsymbol{c}_j\|) = g_i(t), \quad i = n_d+1,\ldots,n_d+n_b. \tag{3.18}$$

Rewriting Equations (3.16) and (3.18) in matrix form and applying θ-weighting ($0 \leq \theta \leq 1$), we get

$$\begin{aligned}[\mathcal{L}^c\boldsymbol{\Phi}_d \;&+\; \mathcal{B}^c\boldsymbol{\Phi}_d + \delta t\ \theta\ (\mathcal{L}^x\mathcal{L}^c\boldsymbol{\Phi}_d + \mathcal{L}^x\mathcal{B}^c\boldsymbol{\Phi}_d)]\boldsymbol{\lambda}^{n+1} = \delta t\ \boldsymbol{f}^{n+1} \\ &+\; [\mathcal{L}^c\boldsymbol{\Phi}_d + \mathcal{B}^c\boldsymbol{\Phi}_d - \delta t\ (1-\theta)\ (\mathcal{L}^x\mathcal{L}^c\boldsymbol{\Phi}_d + \mathcal{L}^x\mathcal{B}^c\boldsymbol{\Phi}_d)]\boldsymbol{\lambda}^n, \end{aligned} \tag{3.19}$$

$$[\mathcal{B}^x\mathcal{L}^c\boldsymbol{\Phi}_b + \mathcal{B}^x\mathcal{B}^c\boldsymbol{\Phi}_b]\boldsymbol{\lambda}^n = \boldsymbol{g}^{n+1}. \tag{3.20}$$

Let $\widehat{\boldsymbol{\Phi}}_d = \mathcal{L}^c\boldsymbol{\Phi}_d + \mathcal{B}^c\boldsymbol{\Phi}_d$ and $\widehat{\boldsymbol{\Phi}}_b = \mathcal{L}^c\boldsymbol{\Phi}_b + \mathcal{B}^c\boldsymbol{\Phi}_b$. Hence, the preceding system of equations can be written as

$$\begin{bmatrix} \widehat{\boldsymbol{\Phi}}_d + \theta\ \delta t\ \mathcal{L}^x\widehat{\boldsymbol{\Phi}}_d \\ \mathcal{B}^x\widehat{\boldsymbol{\Phi}}_b \end{bmatrix} \boldsymbol{\lambda}^{n+1} = \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_d - (1-\theta)\ \delta t\ \mathcal{L}^x\widehat{\boldsymbol{\Phi}}_d \\ \mathbf{0} \end{bmatrix} \boldsymbol{\lambda}^n + \begin{bmatrix} \delta t\ \boldsymbol{f}^{n+1} \\ \boldsymbol{g}^{n+1} \end{bmatrix}. \tag{3.21}$$

Equation (3.21) can be rewritten in compact form as

$$\boldsymbol{\lambda}^{n+1} = \widehat{\boldsymbol{H}}_+^{-1}\widehat{\boldsymbol{H}}_-\boldsymbol{\lambda}^n + \widehat{\boldsymbol{H}}_+^{-1}\widehat{\boldsymbol{F}}^{n+1}, \tag{3.22}$$

where

$$\widehat{\boldsymbol{H}}_+ = \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_d + \theta\ \delta t\ \mathcal{L}^x\widehat{\boldsymbol{\Phi}}_d \\ \mathcal{B}^x\widehat{\boldsymbol{\Phi}}_b \end{bmatrix},\ \widehat{\boldsymbol{H}}_- = \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_d - (1-\theta\ \delta t\ \mathcal{L}^x\widehat{\boldsymbol{\Phi}}_d \\ \mathbf{0} \end{bmatrix}$$

and

$$\widehat{\boldsymbol{F}}^{n+1} = \begin{bmatrix} \delta t\ \boldsymbol{f}^{n+1} \\ \boldsymbol{g}^{n+1} \end{bmatrix}.$$

Equation (3.16) can be written in matrix form as

$$\boldsymbol{u} = \widehat{\boldsymbol{A}}\boldsymbol{\lambda}, \tag{3.23}$$

where $\widehat{\boldsymbol{A}} = \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_d \\ \widehat{\boldsymbol{\Phi}}_b \end{bmatrix} \in \mathbb{R}^{N\times N}$.

Using Equation (3.23), equation (3.22) can be rewritten as

$$\boldsymbol{u}^{n+1} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{H}}_+^{-1}\widehat{\boldsymbol{H}}_-\widehat{\boldsymbol{A}}^{-1}\boldsymbol{u}^n + \widehat{\boldsymbol{A}}\widehat{\boldsymbol{H}}_+^{-1}\widehat{\boldsymbol{F}}^{n+1}. \tag{3.24}$$

In contrast to the unsymmetric collocation scheme, it can be readily shown, using the results of Wu (1998), that the matrix $\widehat{\boldsymbol{H}}_+$ is invertible for any value of θ provided the set of collocation points are distinct. This implies that the symmetric collocation scheme is well-posed.

3.2 Stability analysis

In this section, we present an analysis of the stability of the unsymmetric and symmetric meshless schemes using a matrix method. Initially, we consider the unsymmetric scheme. A perturbation, $\boldsymbol{e}^n = \boldsymbol{u}^n - \widetilde{\boldsymbol{u}}^n$ is introduced into Equation (3.15), where $\boldsymbol{u}^n$ is the discrete exact solution and $\widetilde{\boldsymbol{u}}^n$ is the numerically computed solution. The equation for the error $\boldsymbol{e}^{n+1}$ can then be written as

$$\boldsymbol{e}^{n+1} = \boldsymbol{K}\boldsymbol{e}^n, \tag{3.25}$$

where the amplification matrix $\boldsymbol{K} = \boldsymbol{A}\boldsymbol{H}_+^{-1}\boldsymbol{H}_-\boldsymbol{A}^{-1}$. The numerical scheme will be stable if as $n \to \infty$, the error $\boldsymbol{e}^n \to 0$. This can be guaranteed provided $\rho(\boldsymbol{K}) \leq 1.0$, where $\rho(\boldsymbol{K})$ denotes the spectral radius of the amplification matrix[1] (necessary condition). Substituting $\boldsymbol{K}$ in Equation (3.25) we get

$$\boldsymbol{H}_+\boldsymbol{A}^{-1}\boldsymbol{e}^{n+1} = \boldsymbol{H}_-\boldsymbol{A}^{-1}\boldsymbol{e}^n. \tag{3.26}$$

Assuming Dirichlet boundary conditions (i.e., $\mathcal{B} = \mathcal{I}$, where $\mathcal{I}$ is the identity operator), Equation (3.26) can be written as

$$[\boldsymbol{I} + \theta\ \delta t\ \boldsymbol{M}]\boldsymbol{e}^{n+1} = [\boldsymbol{I} - (1-\theta)\ \delta t\ \boldsymbol{M}]\boldsymbol{e}^n, \tag{3.27}$$

where $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ is the identity matrix and the matrix $\boldsymbol{M} = \begin{bmatrix} \mathcal{L}\boldsymbol{\Phi}_d \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{A}^{-1}$.

It can be seen from Equation (3.27) that stability is assured if all the eigenvalues of the matrix $[\boldsymbol{I} + \theta\ \delta t\ \boldsymbol{M}]^{-1}[\boldsymbol{I} - (1-\theta)\ \delta t\ \boldsymbol{M}]$ are less than unity, i.e.,

$$\left| \frac{1 - (1-\theta)\ \delta t\ \lambda_M}{1 + \theta\ \delta t\ \lambda_M} \right| \leq 1, \tag{3.28}$$

where λ_M is an eigenvalue of the matrix $\boldsymbol{M}$. The eigenvalues of the matrix $\boldsymbol{M}$ can be calculated by solving the generalised eigenvalue problem

$$\mathcal{L}\boldsymbol{\Phi}_d\boldsymbol{s} = \lambda_M\boldsymbol{A}\boldsymbol{s}. \tag{3.29}$$

For the case of the Crank-Nicholson scheme ($\theta = 0.5$), the inequality (3.28) is always satisfied if $\lambda_M \geq 0$. This implies the scheme is unconditionally stable if $\lambda_M \geq 0$.

When $\theta = 0$, we obtain the purely explicit time-stepping formulation. The condition for stability then becomes

$$|1 - \delta t\ \lambda_M| \leq 1. \tag{3.30}$$

[1]The method is stable provided the spectral norm $\|\boldsymbol{K}\| \leq 1$, i.e., $\rho(\boldsymbol{K}) \leq 1$ (a necessary but not sufficient condition).

Hence the explicit formulation will be stable if

$$\delta t \leq \frac{2}{\lambda_M} \quad \text{and} \quad \lambda_M \geq 0. \tag{3.31}$$

The stability criteria, Equation (3.28) and Equation (3.31), derived for the unsymmetric collocation scheme can be readily extended to the symmetric scheme. Using the same approach as for the unsymmetric scheme, it can be shown that the symmetric collocation scheme (Equation 3.24) is stable, if

$$\left| \frac{1 - (1-\theta)\ \delta t\ \lambda_{M_H}}{1 + \theta\ \delta t\ \lambda_{M_H}} \right| \leq 1, \tag{3.32}$$

where λ_{M_H} is an eigenvalue of the matrix $\boldsymbol{M}_H = \begin{bmatrix} \mathcal{L}^x \widehat{\boldsymbol{\Phi}}_d \\ \mathbf{0} \end{bmatrix} \widehat{\boldsymbol{A}}^{-1}$. The symmetric scheme is unconditionally stable for $\theta = 0.5$, if $\lambda_{M_H} \geq 0$. Similar to the unsymmetric stability analysis, for the explicit time-stepping case ($\theta = 0$), we obtain the same inequality as Equation (3.31). The only difference is that λ_M has to be replaced with λ_{M_H}.

It can be seen from inequalities (3.28) and (3.32) that the stability of the unsymmetric and symmetric collocation schemes depends on three factors, viz., θ, δt and the eigenvalues of the matrix $\boldsymbol{M}$ or $\boldsymbol{M}_H$. In the case of RBFs like TPS and Quintics, which do not have a shape parameter, the eigenvalues of the matrix $\boldsymbol{M}$ or $\boldsymbol{M}_H$ depend only on the mesh spacing parameter h (h is defined to be the minimal distance between any two collocation points in the domain). Hence, a distribution of collocation points is acceptable only if all the eigenvalues (λ_{M_H} or λ_M) are positive and $\theta = 0.5$. However in the case of σ-tunable RBFs, the stability also depends on the value of the shape parameter σ. An ideal solution will be to establish bounds for the eigenvalues of the matrix $\boldsymbol{M}$ or $\boldsymbol{M}_H$ as a function of the mesh spacing parameter h and the shape parameter σ. Since no such result can be derived explicitly, we numerically investigate the influence of the parameters h and σ on stability. We concentrate on the case of the Crank-Nicholson scheme ($\theta = 0.5$).

Figure 3.1 shows how the smallest eigenvalue of $\boldsymbol{M}$ ($\lambda_{\min}$) varies as a function of σ, when the mesh spacing h is kept constant. Recollect that the stability condition is satisfied only when $\lambda_{\min} \geq 0$. It can be seen from Figure 3.1 that stability occurs over a varied region of shape parameters. Also, we can observe that there exist pockets of stability and these pockets tend to become narrower and narrower as the shape parameter σ increases. To further the numerical studies on the issue of stability, we define a critical shape parameter σ_{crit}, where for all $\sigma < \sigma_{\text{crit}}$, the stability conditions of the corresponding numerical scheme are satisfied (For example $\sigma_{\text{crit}} \approx 2$ in Figure 3.1).

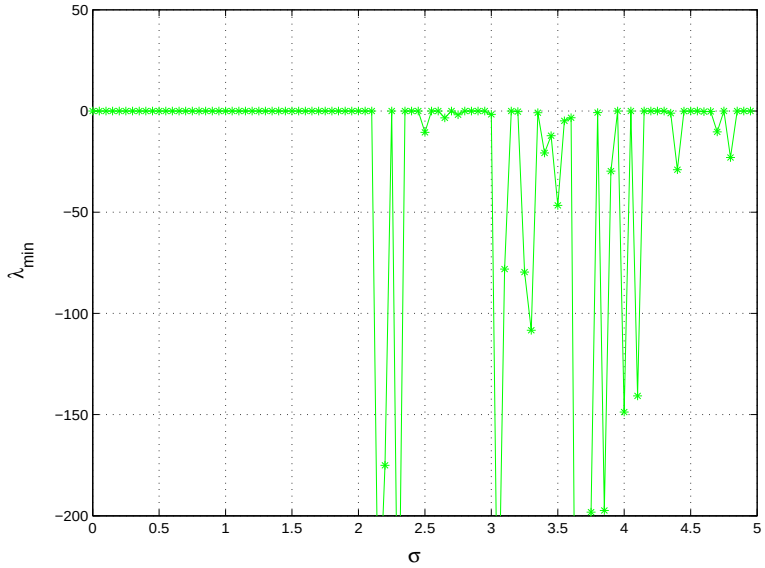
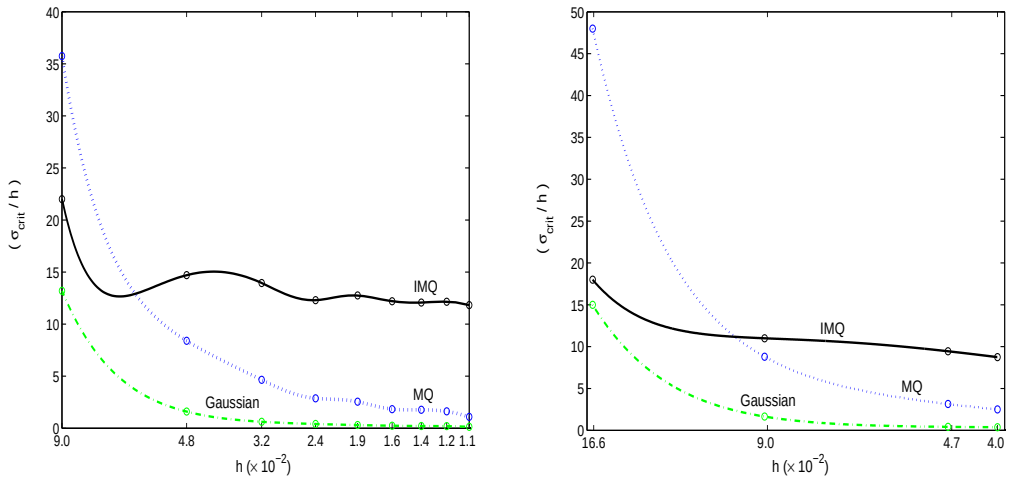


FIGURE 3.1: A typical stability plot using IMQ RBF on the domain $[0,1] \times [0,1]$



1D Problem 2D Problem

FIGURE 3.2: Stability regimes for σ-tunable RBFs (unsymmetric method)

We present the regions of stability for each of the σ-tunable RBF as obtained numerically. Figure 3.2 shows the stability regions of the three σ-tunable RBFs when the unsymmetric formulation is applied. Figure 3.3 shows the stability regions of the Gaussian and IMQ RBFs when the symmetric formulation is applied. On both these graphs, the mesh spacing h is plotted on the x-axis and the values on y-axis represent (σ_{crit}/h). Note that h is decreasing as we go from left to right in Figure 3.2 and Figure 3.3. The critical shape parameter (σ_{crit}) is calculated for 9 different mesh spacings and a second order spline is fitted along

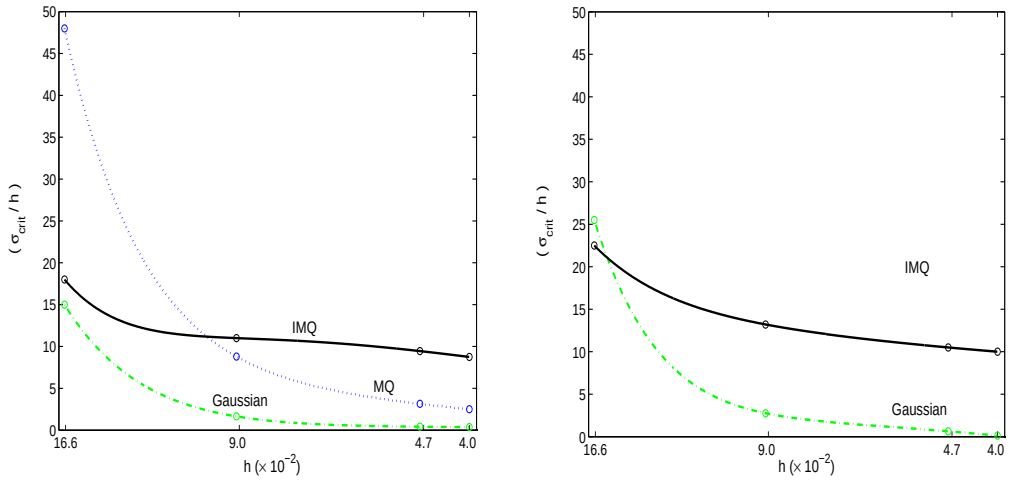


FIGURE 3.3: Stability regimes for σ-tunable RBFs (symmetric method)

the points to obtain the regions of stability. From these figures, it can be seen that as the number of collocation points increases, the range of shape parameter values over which the stability condition is satisfied decreases. This would mean that as more and more collocation points are added in the domain, the freedom of varying the shape parameter is decreased, which would adversely affect the ability of the RBF to capture the solution of the intended problem. Also, it can be seen that both schemes have quite similar regions of stability. For a particular RBF, the stability region of the symmetric scheme is marginally larger than that of the unsymmetric scheme.

3.3 Optimisation of shape parameter via residual minimisation

In the case of σ-tunable RBFs, the optimal value of the shape parameter σ is chosen by observing the behaviour of a suitable residual error calculated on a very fine set of points in the domain. This *a posteriori* error method has been employed earlier in the literature by Cheng *et al.* (2003) in the context of solving elliptic operator problems.

In the case of the unsteady convection-diffusion equation, we calculate the residual error for the unsymmetric scheme from the following relation derived from the governing CD equation and the boundary condition

$$\delta(\boldsymbol{x}, t^{n+1}) = \begin{cases} f(\boldsymbol{x}, t^n) - \sum_{j=1}^{N} \phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|)(\frac{\lambda_j^{n+1} - \lambda_j^n}{\delta t}) - \sum_{j=1}^{N} \lambda_j^{n+1} \left(\mathcal{L}^x \phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|)\right); & \boldsymbol{x} \in \Omega \\ \\ g(\boldsymbol{x}, t^n) - \sum_{j=1}^{N} \lambda_j^{n+1} (\mathcal{B}^x \phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|)); & \boldsymbol{x} \in \partial\Omega \end{cases} \tag{3.33}$$

Ideally, one should calculate the residual after every time step and tune the shape parameter accordingly. This however can be computationally very expensive. Hence, we calculate the residual only for a few time steps and monitor its value at each value of σ.

Figure 3.4 shows how the actual error in the solution as well as the residual error varies as a function of the shape parameter σ for two typical values of $P_e = 1.0$ and $P_e = 10.0$. The actual error ε is defined as the L_∞ norm of the difference between the analytical and the numerically obtained solutions, i.e.,

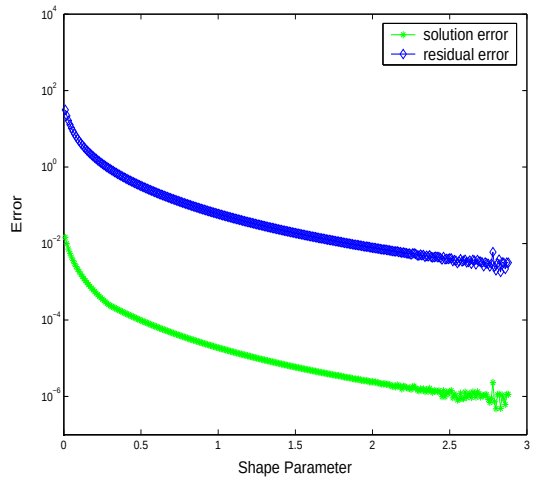
$$\varepsilon = \|\boldsymbol{u}_{\text{analytical}} - \boldsymbol{u}_{\text{numerical}}\|_\infty,$$

and the residual error shown in Figure 3.4 is the L_2 norm of the residual error vector.

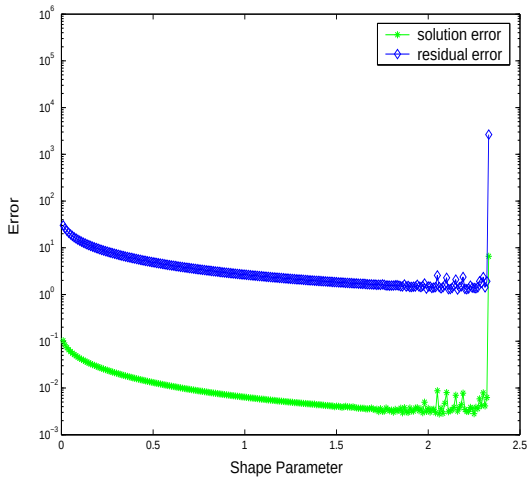
It can be seen from Figure 3.4 that the L_2 norm of the residual error vector behaves similarly as the actual solution error with respect to the shape parameter. Hence the L_2 norm of the residual error vector can be used to estimate the optimal value of the shape parameter when the exact solution of the problem is not known. In the numerical studies presented in the next section, we compute the optimal value of σ using Brent's one-dimensional minimisation procedure (Brent, 1973), with the L_2 norm of the residual error as the objective function to be minimised. It can be noted from Figure 3.4 that there is a significant amount of noise near the right edge of the plots. Hence, we fitted a third-order polynomial to the L_2 norm of the residual error as a function of the shape parameter till the point the first fluctuation occurs. Subsequently, we use Brent's method to search for the minima of the polynomial.

3.4 Numerical study: 1D problem

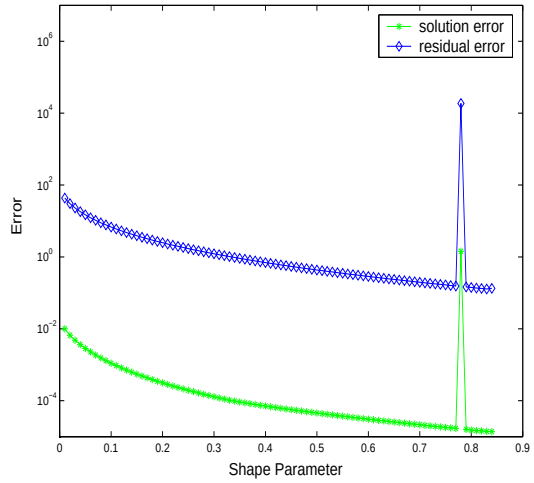
In this section, we present numerical studies for the 1D unsteady convection-diffusion problem employing unsymmetric and symmetric meshless approaches. We also investigate the convergence trends of various RBFs on this problem. For the ease of implementation, in the unsymmetric method, we have removed the polynomial term in the approximation of



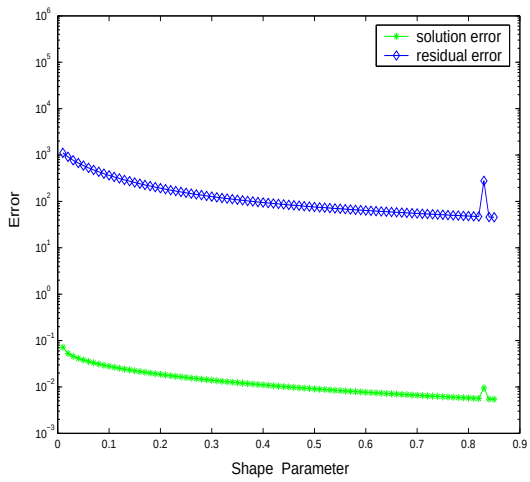
1D:Low Peclet Number



1D:Medium Peclet Number



2D:Low Peclet Number

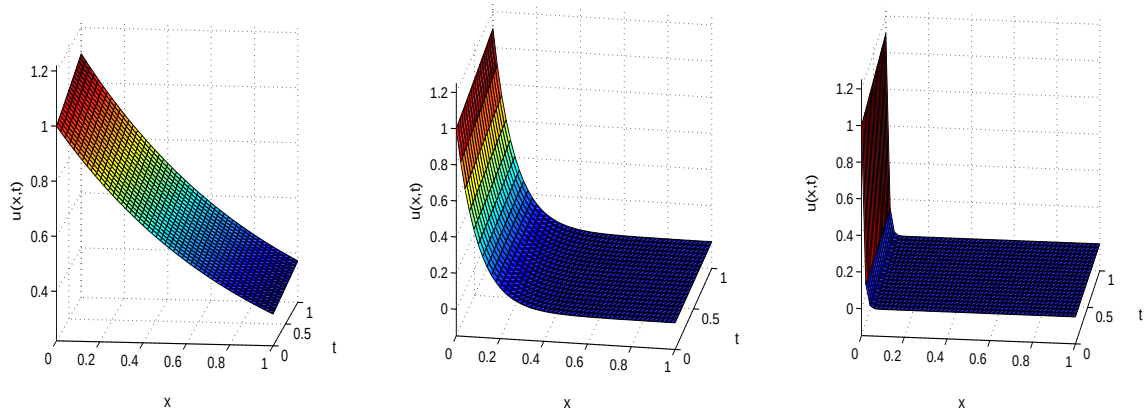


2D:Medium Peclet Number

FIGURE 3.4: Solution and residual error Vs shape parameter (σ) for MQ RBF

the potential function by RBFs. It is worth noting that for conditionally positive definite RBFs, an additional polynomial term needs to be augmented to Equation (3.5) in order to guarantee invertibility of the Gram matrix (Micchelli, 1986). However, it has been shown that with or without the polynomial term the approximation provided by RBFs does not vary much (Power & Barraco, 2002; Wong *et al.*, 1999).

We solve the unsteady 1D CD equation using an increasing number of collocation points. The results are plotted on a logarithmic scale with decreasing mesh spacing h on the x-axis and the error ε on the y-axis. The mesh spacing h is defined as the minimum spacing between any two collocation points in the domain. For σ-tunable RBFs, at every point in subsequent



Peclet number = 1.0 Peclet number = 10.0 Peclet number = 100.0

FIGURE 3.5: Analytical solution behaviour for three different Peclet numbers for the 1D convection-diffusion equation ($a = 1.0$, $b = 0.1$, $V = 1.0$)

graphs, the shape parameter is tuned to its optimal value, using the procedure outlined in Section 3.3.

We consider the following one-dimensional problem,

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + V \frac{\partial u}{\partial x}, \quad 0 \leq x \leq 1 \ , \ t > 0, \tag{3.34}$$

with the following Dirichlet boundary conditions and initial condition

$$u(0,t) = a\mathrm{e}^{bt} \ \ , u(1,t) = a\mathrm{e}^{bt-c} \ \ t > 0,$$

$$u(x,0) = a\mathrm{e}^{-cx}.$$

In Equation (3.34), κ is the diffusion coefficient, V is a constant representing the velocity and a,b,c are some arbitrary constants. The analytical solution for the above problem is given by

$$u(x,t) = a\mathrm{e}^{bt-cx} \quad \text{where} \quad c = \frac{V \pm \sqrt{V^2 + 4\kappa b}}{2\kappa} > 0. \tag{3.35}$$

The Peclet number for the above problem is defined as $P_e = \frac{V}{\kappa}$. The analytical solution is shown in Figure 3.5 for three Peclet numbers. Numerical results obtained for the 1D problem using the unsymmetric meshless collocation scheme are presented for various Peclet numbers in subsequent subsections.

3.4.1 Uniform distribution

We consider a uniform distribution of collocation points initially. Uniformly distributed collocation points ranging from $N_{\min} = 11$ to $N_{\max} = 101$ have been taken in the 1D domain for studying the convergence trends of each RBF.

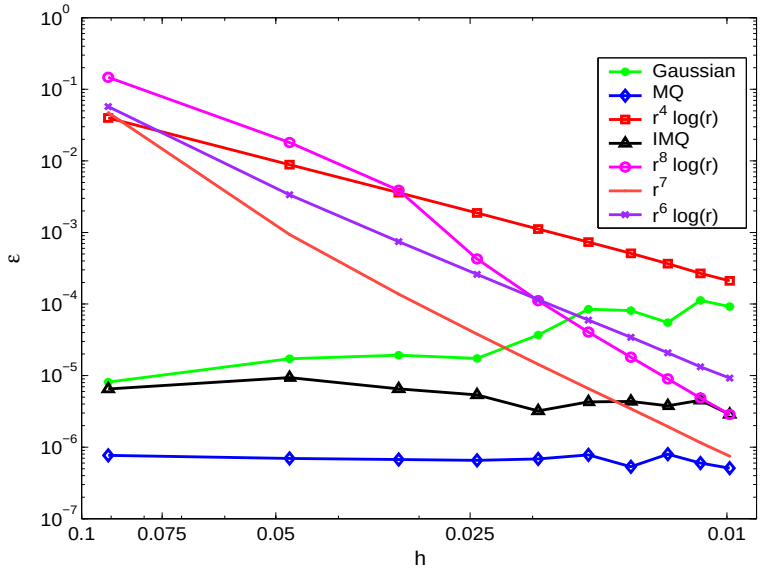


FIGURE 3.6: Accuracy of different GSRBFs for Peclet number 1.0 (1D problem): $a = 1.0$, $b = 0.1$, $V = 1.0$, $\kappa = 1.0$, $\delta t = 0.001$, $t_f = 1.0$, $\theta = 0.5$

Figure 3.6 shows the convergence trends of each of the GSRBF when $P_e = 1.0$. This is the case when the convection term is comparable to the diffusion term. We find the RBFs incorporating a shape parameter (MQ, IMQ and Gaussian), when properly tuned, have very high convergence rates as compared to higher order TPS or quintics. From the figure, it can be seen that the multiquadric (MQ) performs better compared to the other RBFs. We also note that for the Gaussian RBF, the shape parameter optimisation procedure results in a slight increase of the errors obtained as we increase the number of collocation points. This is due to the fact that the Brent's optimisation procedure searches for the optimal value in the region $\sigma < \sigma_{\text{crit}}$ whereas the value of σ leading to better results might lie in the other pockets of stability as can be seen in Figure 3.2. It is of interest to note that higher order TPS (like $r^8 \log r$ and $r^6 \log r$) tend to achieve results comparable to that of the multiquadrics as we move from left to right in the graph (i.e., from a coarse to a dense set of collocation points).

We now investigate the behaviour of all the RBFs for a case when the convection term slightly outgrows the diffusion term, i.e., $P_e = 10.0$. The analytical solution is not completely

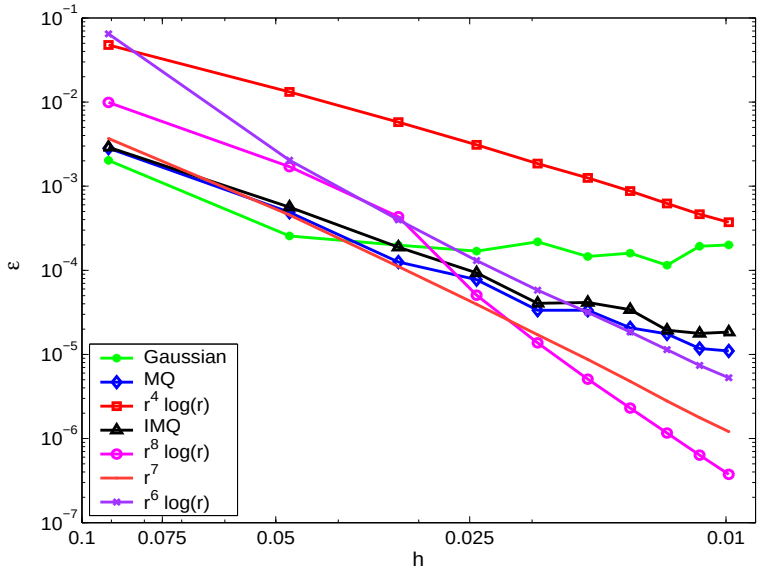


FIGURE 3.7: Accuracy of different GSRBFs for Peclet number 10.0 (1D problem): $a = 1.0$, $b = 0.1$, $V = 1.0$, $\kappa = 0.1$, $\delta t = 0.001$, $t_f = 1.0$, $\theta = 0.5$

smooth and hence for a small number of collocation points, all the RBFs are unable to capture the solution with a high degree of accuracy (see Figure 3.7). However, infinitely differentiable RBFs (MQ, IMQ and Gaussian) produce results with errors, $\varepsilon \approx 10^{-3}$. As we increase the number of collocation points, $r^8 \log r$ and $r^6 \log r$ outperform MQ, IMQ and Gaussians.

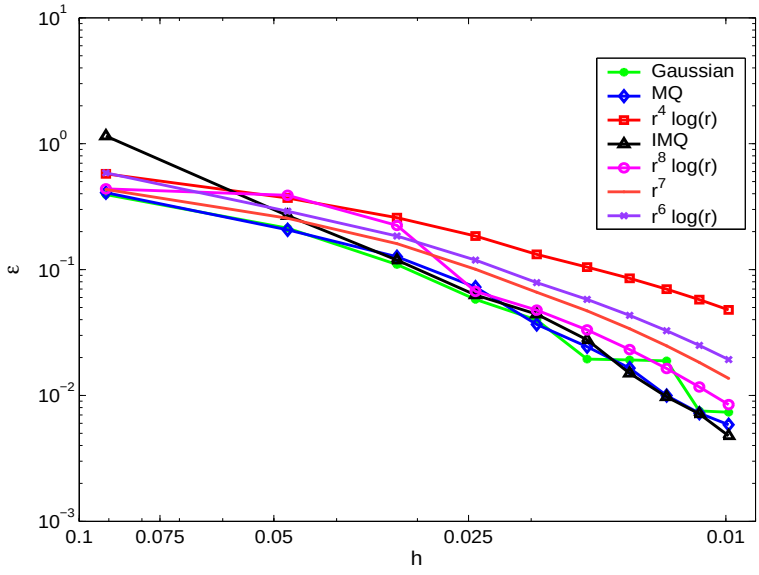


FIGURE 3.8: Accuracy of different GSRBFs for Peclet number 100.0 (1D Problem): $a = 1.0$, $b = 0.1$, $V = 1.0$, $\kappa = 0.01$, $\delta t = 0.001$, $t_f = 1.0$, $\theta = 0.5$

Next consider the case of $P_e = 100.0$, where the convection term completely dominates over the diffusion term. For this case the analytical solution has a sharp discontinuity near the left boundary. The convergence trends of various RBFs for $P_e = 100.0$ are summarised in Figure 3.8. It can be observed from the figure that the errors decrease as the number of collocation points increases. The accuracy suffers as compared to the earlier cases of $P_e = 1.0$ and $P_e = 10.0$. This can be attributed to the numerical oscillations observed in the numerical solution. However, for 271 collocation points spaced regularly in the 1D domain, we obtain errors ε of magnitude $6.600E-03$ and $5.068E-04$ for $r^4 \log r$ and MQ RBFs respectively. This suggests that in principle, the unsymmetric scheme is capable of capturing the solution given sufficient number of collocation points. The main hindrance being that for a large number of collocation points, the matrix $\boldsymbol{H}_+$ turns out to be highly ill-conditioned.

3.4.2 Unsymmetric Vs Symmetric schemes

We now investigate the performance of the symmetric scheme on this 1D problem. Gaussian and IMQ RBFs are used as they do not require a polynomial term in their interpolation to guarantee invertibility of the matrix $\widehat{\boldsymbol{H}}_+$. Our results are summarised in Table 3.1 and Table 3.2 for the Gaussian and IMQ RBFs respectively. The first column shows the number of collocation points used in the domain. The errors obtained from both the schemes for each of the Peclet numbers, are presented in the subsequent columns. It can be seen from the results that the symmetric scheme is marginally better than the unsymmetric scheme. However, the unsymmetric scheme has the advantage of being easier to implement.

TABLE 3.1: Errors ε obtained using unsymmetric and symmetric schemes with Gaussian RBF - 1D problem ($\theta = 0.5$)

	$P_e = 1.0$		$P_e = 10.0$		$P_e = 100.0$	
N	Unsym.	Sym.	Unsym.	Sym.	Unsym.	Sym.
11	1.62E-06	2.50E-08	2.66E-03	1.13E-03	5.67E-01	5.45E-01
21	1.42E-06	7.60E-07	8.54E-05	7.72E-05	4.67E-01	2.29E-01
31	7.06E-05	5.79E-06	3.44E-04	2.81E-05	1.62E-01	9.82E-02
41	2.63E-06	2.68E-05	4.33E-04	4.86E-04	6.11E-02	6.30E-02
51	1.59E-04	9.73E-05	4.85E-04	1.27E-04	3.46E-02	3.43E-02

TABLE 3.2: Errors ε obtained using unsymmetric and symmetric schemes with inverse multiquadric RBF - 1D problem ($\theta = 0.5$)

	$P_e = 1.0$		$P_e = 10.0$		$P_e = 100.0$	
N	Unsym.	Sym.	Unsym.	Sym.	Unsym.	Sym.
11	9.93E-07	9.52E-08	1.66E-03	1.43E-03	4.73E-01	4.31E-01
21	2.44E-06	4.59E-08	1.87E-04	7.92E-05	2.39E-01	2.38E-01
31	2.77E-06	7.82E-07	6.49E-05	3.30E-05	9.55E-02	1.10E-01
41	1.62E-06	1.61E-06	8.58E-05	2.51E-05	6.25E-02	8.63E-02
51	3.20E-06	3.72E-06	5.81E-05	1.97E-05	3.94E-02	3.75E-02

3.5 Numerical study: 2D problem

Here, we investigate the behaviour of RBFs on a two dimensional analog of the one dimensional problem considered earlier. The implementation issues related to extending the problem to the two-dimensional case are trivial since a RBF is a function of the Euclidean distance between any two collocation points in the domain. We first present results obtained using the unsymmetric scheme and subsequently we compare them with those obtained using the symmetric scheme.

The governing equation is written as

$$\frac{\partial u}{\partial t} = \kappa_x \frac{\partial^2 u}{\partial x^2} + \kappa_y \frac{\partial^2 u}{\partial y^2} + V_x \frac{\partial u}{\partial x} + V_y \frac{\partial u}{\partial y}, \;\; 0 \leq x, y \leq 1; \; t > 0 \tag{3.36}$$

with the boundary conditions

$$\begin{aligned} u(0, y, t) &= a\mathrm{e}^{bt}(1 + \mathrm{e}^{-c_y y}), \qquad u(1, y, t) = a\mathrm{e}^{bt}(\mathrm{e}^{-c_x} + \mathrm{e}^{-c_y y}), \\ u(x, 0, t) &= a\mathrm{e}^{bt}(1 + \mathrm{e}^{-c_x x}), \qquad u(x, 1, t) = a\mathrm{e}^{bt}(\mathrm{e}^{-c_x x} + \mathrm{e}^{-c_y y}), \end{aligned} \tag{3.37}$$

and with the initial condition

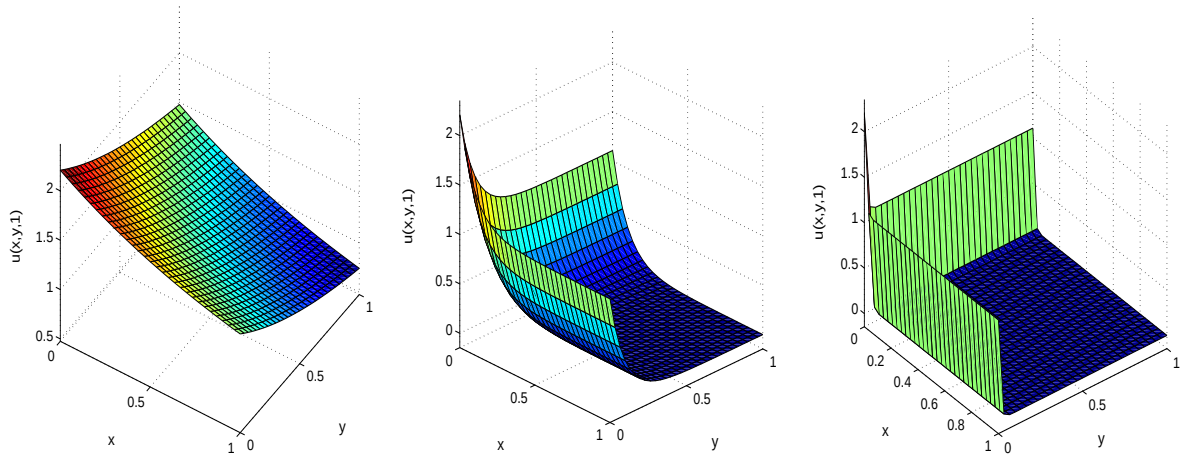
$$u(x, y, 0) = a\left(\mathrm{e}^{-c_x x} + \mathrm{e}^{-c_y y}\right). \tag{3.38}$$

The analytical solution is given by

$$u(x, y, t) = a\mathrm{e}^{bt}\left(\mathrm{e}^{-c_x x} + \mathrm{e}^{-c_y y}\right), \tag{3.39}$$

where

$$c_x = \frac{V_x \pm \sqrt{V_x^2 + 4b\kappa_x}}{2\kappa_x} > 0 \quad \text{and} \quad c_y = \frac{V_y \pm \sqrt{V_y^2 + 4b\kappa_y}}{2\kappa_y} > 0.$$



Peclet number = 1.0 Peclet number = 10.0 Peclet number = 100.0

FIGURE 3.9: Analytical solution behaviour for three different Peclet numbers for the 2D convection-diffusion equation ($a = 1.0$, $b = 0.1$, $\kappa = 1.0$) at time $t = 1.0$

If we put $V_x = V_y = V$ and $\kappa_x = \kappa_y = \kappa$, for the two dimensional case we can define an analogous Peclet number as $P_e = \frac{V}{\kappa}$. As before, we present our results for the 2D problem for three different Peclet numbers (1.0,10.0 and 100.0). The analytical solutions are shown in Figure 3.9. We consider uniform and scattered distribution of collocation points for the 2D problem. The final results were obtained by predicting the solution $u(\boldsymbol{x}, t)$ on a fine mesh points (50×50).

3.5.1 Uniform distribution

We consider uniformly distributed collocation points ranging from $N_{\min} = 6 \times 6$ to $N_{\max} = 25 \times 25$ in the 2D domain to obtain the convergence trends of each RBF.

Figure 3.10 shows the convergence trends of the RBFs for the 2D problem when $P_e = 1.0$. As before, σ-tunable RBFs have high convergence rates and accurate results are obtained with TPS provided there are sufficient number of collocation points.

From Figure 3.11, it can be observed that, for a small number of collocation points, the errors in the approximation provided by various RBFs are quite high. As we move to the right side of the graph we get acceptable results for $r^8 \log r$, $r^6 \log r$, MQ and IMQ RBFs. This is the case of $P_e = 10.0$ when the convection term is one order more than that of the diffusion term. A minimal mesh of 21×21 uniform collocation points is needed for the RBFs to produce acceptable results.

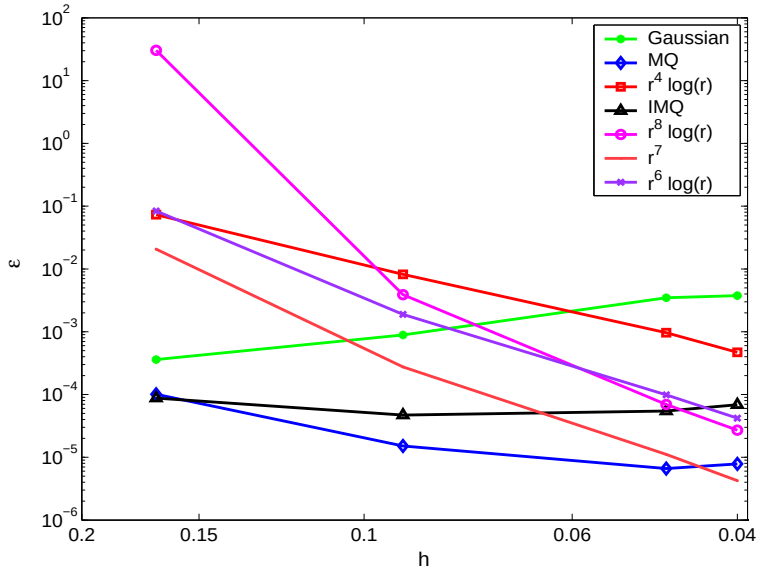


FIGURE 3.10: Accuracy of different GSRBFs for Peclet number 1.0 (2D problem): $a = 1.0$, $b = 0.1$, $V_x = 1.0$, $V_y = 1.0$, $\kappa_x = 1.0$, $\kappa_y = 1.0$, $\delta t = 0.001$, $t_f = 0.1$, $\theta = 0.5$

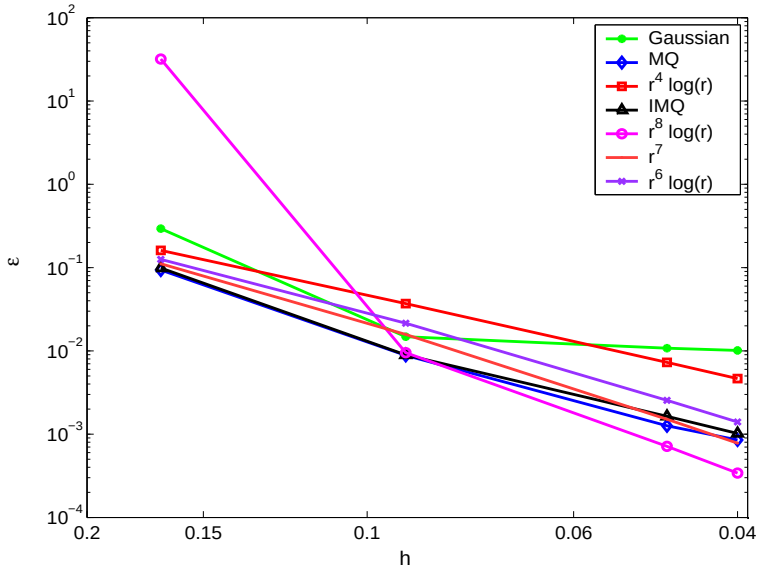


FIGURE 3.11: Accuracy of different GSRBFs for Peclet number 10.0 (2D problem): $a = 1.0$, $b = 0.1$, $V_x = 10.0$, $V_y = 10.0$, $\kappa_x = 1.0$, $\kappa_y = 1.0$, $\delta t = 0.001$, $t_f = 0.1$, $\theta = 0.5$

The accuracy of various RBFs for $P_e = 100.0$ is shown in Figure 3.12. All the RBFs with the given set of collocation points are not able to capture the sharp discontinuity present in the analytical solution. This can be attributed to the fact that more number of collocation points are needed to capture the discontinuity. However, as we increase the number of collocation points it is observed that the coefficient matrix $\boldsymbol{H}_+$ becomes highly

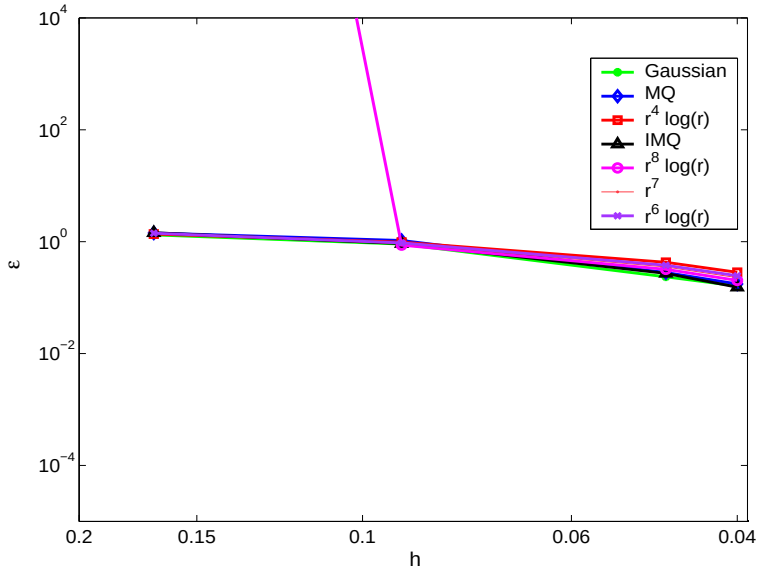


FIGURE 3.12: Accuracy of different GSRBFs for Peclet number 100.0 (2D problem): $a = 1.0$, $b = 0.1$, $V_x = 100.0$, $V_y = 100.0$, $\kappa_x = 1.0$, $\kappa_y = 1.0$, $\delta t = 0.001$, $t_f = 0.1$, $\theta = 0.5$

ill-conditioned.

We now investigate the rate of convergence of the meshless collocation methods. To the best of our knowledge, theoretical results on RBF based meshless collocation methods for time-dependent problems are scarce in the literature. However, theoretical estimates for RBF interpolation are well known (Powell, 1992; Schaback, 1999). Also, for the symmetric RBF method, convergence estimates for linear elliptic PDEs can be found in the work of Franke & Schaback (1998). So, to gain some insight into the rate of convergence of meshless methods for the unsteady convection-diffusion equation, we resort to a numerical study. We consider the case when $\theta = 0.5$, which corresponds to the Crank-Nicholson scheme (CN), which is second-order accurate in time. We let $\delta t = 1.0E-04$ to ensure that the temporal error terms are very small in magnitude. We now estimate the convergence characteristics of the RBF schemes as a function of the spatial distribution of collocation points (h) and Peclet number (P_e). Since the effect of the shape parameter (σ) on the convergence is not well known, we use a TPS ($r^8 \log r$) which does not have a shape parameter. We use a collocation point set of 11×11, 21×21, 25×25 and 30×30 for our numerical study.

Figure 3.13 shows the convergence rates of the unsymmetric method as obtained numerically for different Peclet numbers. Note that we use increasing mesh spacing h on the x-axis. The slope (ν) of each line in Figure 3.13 indicates the convergence rate. From the figure, it can be seen that for low Peclet numbers the convergence rates are very high.

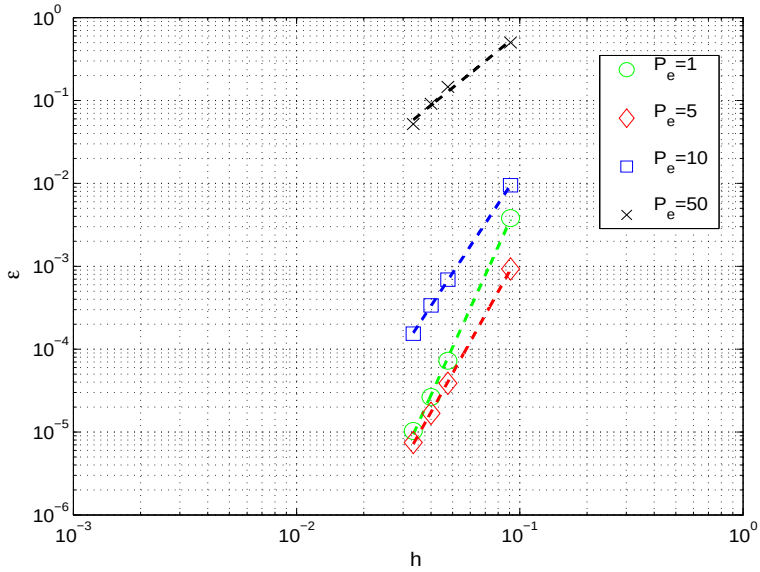


FIGURE 3.13: Convergence rates of unsymmetric method for different Peclet numbers, RBF $= r^8 \log r$

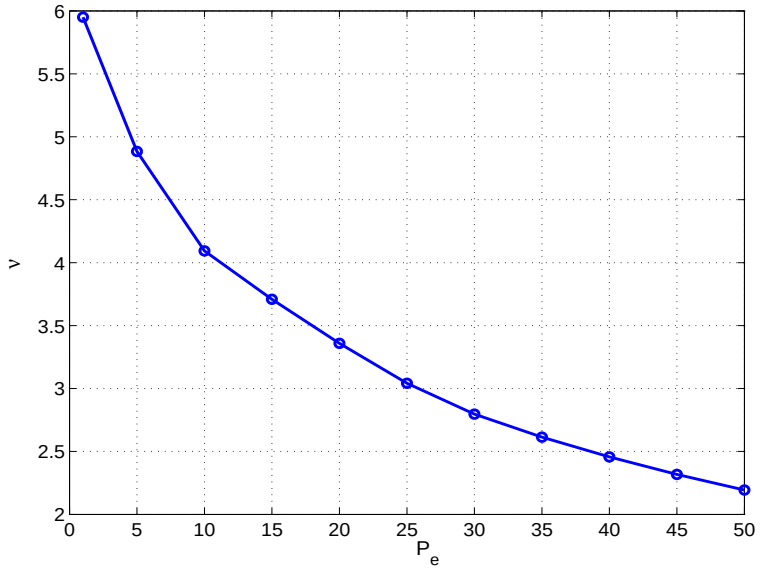


FIGURE 3.14: Convergence rate, ν, Vs Peclet number

Figure 3.14 shows how the convergence rate (ν) varies with Peclet number. From this figure, a qualitative idea of the rate of convergence as a function of P_e can be obtained. For each Peclet number, the unsymmetric method approximately converges at the rate of $\mathcal{O}(h^{\nu})$, when $r^8 \log r$ RBF is used. From Figure 3.14, it can be seen that the convergence rate varies from $\mathcal{O}(h^6)$ for low Peclet numbers to $\mathcal{O}(h^{2.2})$ for $P_e = 50$. Similar behaviour was also observed when the symmetric method is used. This behaviour is expected because for high

Pe, the ill-conditioning problem affects the accuracy of the unsymmetric and symmetric schemes. It is also worth noting that similar trends are obtained when the time step is decreased further.

3.5.2 Scattered distribution

We investigate the convergence trends of the various RBFs when a scattered set of collocation points is taken in the computational domain. The random set of points were generated using Sobol sequences (Sobol, 1979). Figure 3.15 shows the spatial distribution of the collocation points for $N = 121$ and $N = 625$ points respectively.

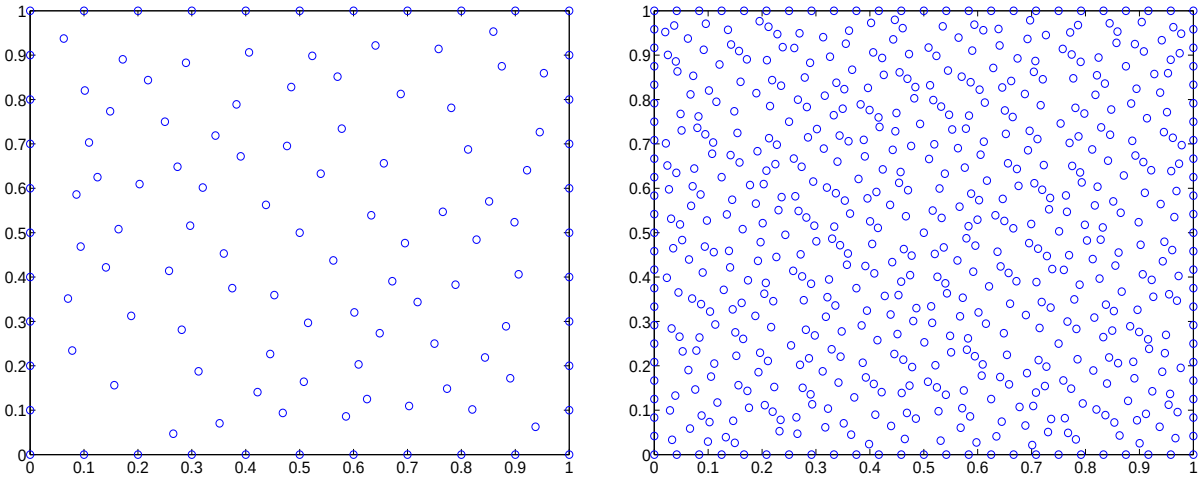


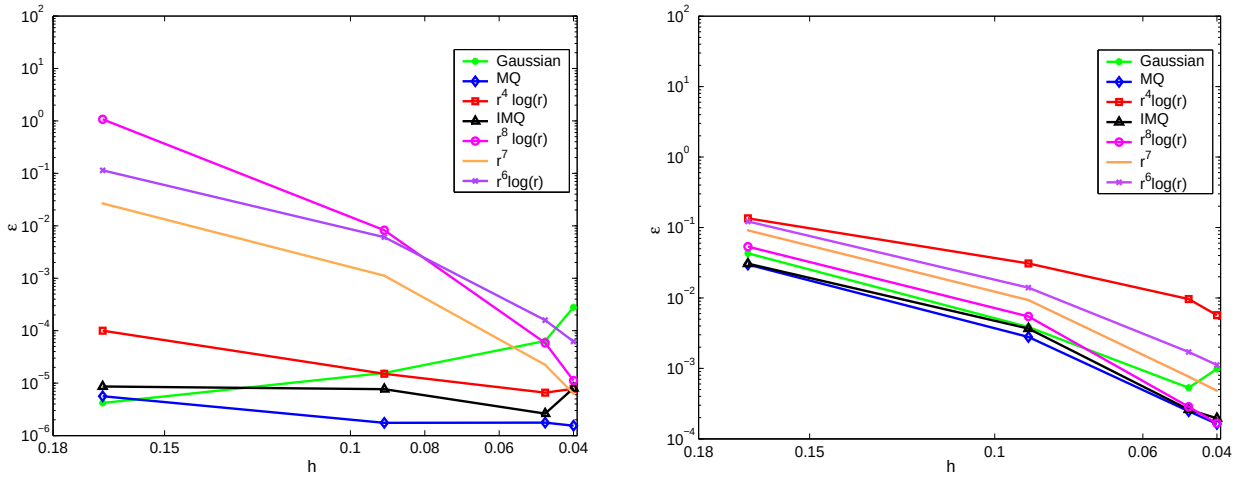
FIGURE 3.15: Scattered distribution of points

The convergence behaviours for Peclet numbers of 1.0 and 10.0 are presented in Figure 3.16. From the figures, it can be observed that the RBFs are capable of approximating the solutions when a set of randomly scattered collocation points are used. The results obtained are comparable with those obtained using a uniformly distributed collocation point set.

3.5.3 Unsymmetric Vs Symmetric schemes

We now present a comparison table between the unsymmetric scheme and the symmetric scheme for the two-dimensional problem. As before, we observe that both the schemes perform equally well. The results for the IMQ RBF are summarised in Table 3.3.

It can be seen from Table 3.3 that for Peclet numbers 10.0 and 100.0, the accuracy of both the methods increases with the number of collocation points. From the last row in Table



Peclet Number = 1.0 Peclet Number = 10.0

FIGURE 3.16: Convergence behaviours of various GSRBFs for the 2D unsteady convection-diffusion problem on a scattered set of points

TABLE 3.3: Errors ε obtained using unsymmetric and symmetric schemes with IMQ RBF - 2D problem ($\theta = 0.5$)

	$P_e = 1.0$		$P_e = 10.0$		$P_e = 100.0$	
N	Unsym.	Sym.	Unsym.	Sym.	Unsym.	Sym.
11×11	4.70E-05	2.09E-05	9.00E-03	5.30E-03	9.22E-01	9.50E-01
21×21	5.48E-05	5.07E-05	1.60E-03	1.60E-03	2.87E-01	3.05E-01
25×25	6.87E-05	6.45E-05	1.00E-03	1.00E-03	1.70E-01	1.90E-01
41×41	5.25E-05	5.21E-05	4.36E-04	4.25E-04	6.36E-02	4.97E-02

3.3, the results obtained for $P_e = 100$ using 41×41 points indicate that to obtain better accuracies using the unsymmetric or symmetric schemes, a large number of collocation points will be needed. However, the collocation matrix $\boldsymbol{H}_+$ for the unsymmetric scheme and $\widehat{\boldsymbol{H}}_+$ for symmetric scheme become highly ill-conditioned with increase in the number of collocation points.

TABLE 3.4: Error and computational cost of CSRBF $\phi(r) = (1-r)_+^6(35r^2 + 18r + 3)$ for different support parameter values and $P_e = 1.0$, $a = 1.0$, $b = 0.1$, $V = 1.0$, $\kappa = 1.0$, $N = 51$

δ	ε	CPU Time (sec)
0.1	0.4413	2.3449
0.2	0.0127	2.5054
0.3	0.0049	2.6349
0.4	0.0024	2.8369
0.5	0.0013	2.9402
0.6	8.30E-04	3.0150
0.7	5.57E-04	3.1793
0.8	3.95E-04	3.2307
0.9	2.92E-04	3.2393
1.0	2.24E-04	3.2417
1.5	8.26E-05	3.3265

3.6 A note on compactly supported RBFs

In this section, we present the results obtained for the CD equation when CSRBFs are used (Djidjeli *et al.*, 2004). The computational cost incurred by the unsymmetric method is recorded for different support parameter (δ) values. We have taken a constant $N = 51$ for the 1D unsteady problem. The results obtained are presented in Table 3.4. We also compare the error vs computational cost in Figure 3.17. From the figure, it can be seen that one obtains better accuracy by increasing the support parameter which brings it nearer and nearer to a dense coefficient matrix. The denser the matrix becomes, the more the ill-conditioning. The same behaviour was observed even for the 2D problem.

3.7 Conclusion

In this chapter, we presented unsymmetric and symmetric meshless schemes for the unsteady convection-diffusion equation. A θ-weighting scheme was used for time stepping. Stability analysis of unsymmetric and symmetric schemes was presented for implicit as well as explicit time stepping. For RBFs with a variable shape parameter, an *a posteriori* residual method was introduced to obtain the optimal value of the shape parameter.

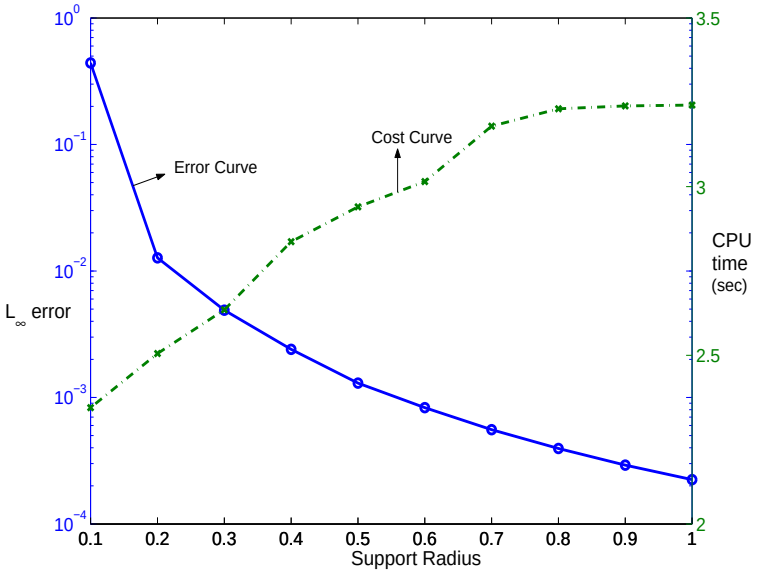


FIGURE 3.17: Error Vs computational cost for a typical CSRBF

Numerical studies for 1D and 2D unsteady convection-diffusion problems have been successfully carried out. The convergence trends of several globally supported RBFs were examined for Peclet numbers 1, 10 and 100. Both uniform and scattered distribution of collocation points were considered. Our numerical results show that RBF based meshless schemes achieve good accuracies even for moderate Peclet numbers. Also, for the particular case of unsteady convection-diffusion problems, the choice of RBF used does make an impact on the accuracy of the numerical solution. The symmetric scheme produces marginally better results as compared to the unsymmetric scheme. We observed that the condition numbers of symmetric collocation matrix ($\widehat{\boldsymbol{H}}_+$) are generally smaller than the condition numbers of the unsymmetric collocation matrix ($\boldsymbol{H}_+$), as reported in Fasshauer (1996). However, the implementation of the symmetric scheme is more difficult. Based on the convergence results obtained for different RBFs, it can be observed that infinitely differentiable RBFs incorporating a shape parameter (MQ and IMQ) produce good results over a variety of mesh spacings. However, RBFs such as TPS or quintics give accurate results when there is a dense set of collocation points. Moreover, there is no need of σ-tuning in these RBFs. Based on these observations, the multiquadric RBF is chosen as the basis function in this thesis. The stability analysis for σ-tunable RBFs suggests that the width of the interval from which the shape parameter can be chosen decreases as we increase the number of collocation points. Also, even though CSRBFs produce banded matrices, their accuracy is very much dependent on the value of the support parameter. For the CD equation, when the support is much less than 1, one

obtains a very sparse matrix and the coefficient matrix is well-conditioned. However, the results obtained for such a sparse matrix are very inaccurate. The accuracy becomes better as we increase the support. Consequently, we end up with a completely dense matrix (support > 1) in order to obtain results equivalent to that of GSRBFs.

For the high Peclet number problem, both the unsymmetric and symmetric schemes are capable of producing acceptable results provided we increase the number of collocation points. This motivates the development of alternate RBF based methods which can solve large scale problems without the condition number becoming worse. In the next two chapters we explore the development of RBF methods capable of solving large scale problems for improving computational efficiency and numerical stability.

Chapter 4

RBF-Domain Decomposition Methods

In this chapter, we discuss different Domain Decomposition Methods (DDMs) using RBFs for solving PDEs. As discussed in the previous chapters, RBF methods suffer from ill-conditioning which hinders their application to large scale problems. The objective of this work is to investigate how DDMs can be leveraged to improve the efficiency of RBF collocation methods (which have such a good convergence rate) for large scale problems. We propose overlapping domain decomposition methods which are illustrated for time-dependent problems and nonlinear problems (Chinchapatnam *et al.*, 2006a, 2005).

In 1870, Schwarz introduced the concept of domain decomposition through the classical Schwarz alternating algorithm. From then till today, DDMs have been well developed and utilised for solving PDEs using FD, FE and FV schemes. For a detailed exposition of the application of DDMs for FD, FE and FV methods, the reader is referred to Smith *et al.* (1996); Quarteroni & Valli (1999). There have been a few works on DDMs using RBFs by Dubal (1994) and Beatson *et al.* (2000). Beatson *et al.* (2000) used the concept of DDMs for efficiently solving the RBF interpolation problem. Recently, some studies on using DDMs to solve PDEs by RBF collocation have appeared in the literature (Wong *et al.*, 1999; Zhou *et al.*, 2003; Li & Hon, 2004). Overlapping DDMs, non-overlapping DDMs with matched and unmatched grids using RBFs have been successfully presented to solve elliptic problems in Li & Hon (2004).

4.1 Time-dependent PDEs

For time-dependent problems, the explicit multizone method due to Wong *et al.* (1999) is the only RBF based domain decomposition method existing in the literature to the best of our knowledge. In this section, we present Schwarz overlapping schemes for the solution of time-dependent problems using RBFs. The proposed schemes are compared with the global RBF-Theta collocation method and the explicit multizone domain decomposition method (Wong *et al.*, 1999) by solving an unsteady convection-diffusion problem for various Peclet numbers. Stability analysis of the presented schemes suggest that for radial basis functions incorporating a free shape parameter, the freedom of varying the shape parameter decreases with increase in the number of collocation points. Also, we find that a major disadvantage of the explicit multizone method arises from the requirement of using a very small time step to ensure numerical stability. In contrast the Schwarz algorithms coupled with a semi-implicit time discretisation RBF scheme permit large values of time step to be used. Numerical studies show that the ill-conditioning problem of the global RBF-Theta method is reduced by the proposed Schwarz schemes. Also, with an increase in the number of sub-domains the efficiency of the Schwarz schemes increases with a slight loss in the accuracy.

We illustrate the domain decomposition scheme for a general linear time-dependent equation of the form

$$\begin{aligned} \frac{\partial u(\boldsymbol{x},t)}{\partial t} + \mathcal{L}u(\boldsymbol{x},t) &= f(\boldsymbol{x},t); \qquad \boldsymbol{x} \in \Omega \in \mathbb{R}^d, \\ \mathcal{B}u(\boldsymbol{x},t) &= g(\boldsymbol{x},t); \qquad \boldsymbol{x} \in \partial\Omega \in \mathbb{R}^d, \end{aligned} \tag{4.1}$$

where Ω denotes a closed physical domain over which the PDE is to be solved and $\partial\Omega$ denotes its boundary. Here, $\mathcal{L}$ is a linear differential operator and $\mathcal{B}$ is an operator which imposes the boundary conditions; $u(\boldsymbol{x},t)$ is the desired field solution and $f(\boldsymbol{x},t)$, $g(\boldsymbol{x},t)$ are prescribed functions.

4.2 Explicit multizone method

In this method, a second order explicit forward difference scheme is used for time stepping. Assuming $f(\boldsymbol{x},t) \equiv 0$ and Dirichlet boundary conditions, temporal discretisation of Equation (4.1) leads to

$$\begin{aligned} u(\boldsymbol{x},t^{n+1}) &= u(\boldsymbol{x},t^n) - \delta t\,[\mathcal{L}u(\boldsymbol{x},t^n)] + (\delta t^2/2)\mathcal{L}^2 u(\boldsymbol{x},t^n); \qquad \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x},t^{n+1}) &= g(\boldsymbol{x},t^{n+1}); \qquad \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{4.2}$$

where $\mathcal{L}^2 u(\boldsymbol{x}, t^n)$ denotes the application of the operator twice on the function $u(\boldsymbol{x}, t^n)$.

The physical domain Ω is divided into m non-overlapping sub-domains such that

$$\Omega = \bigcup_{k=1}^{m} \Omega_k.$$

The set of collocation points $\mathcal{C}$ of cardinality N is divided into m subsets $\mathcal{C}_j$, $j = 1, 2, \ldots, m$ such that

$$\mathcal{C}_k \bigcap \mathcal{C}_j = \emptyset \quad \text{if} \quad k \neq j$$

and

$$\bigcup_{k=1}^{m} \mathcal{C}_k = \mathcal{C}.$$

Note that the points in collocation set $\mathcal{C}_k$ are contained within the sub-domain Ω_k.

Another set $\mathcal{B}_k$ is formed such that for each sub-domain Ω_k, the elements in $\mathcal{B}_k$ contain all the points in the other sub-domains that lie within a certain pre-specified distance (Δ) from the artificial boundary of the sub-domain Ω_k,

$$\begin{aligned}\mathcal{B}_k = \{\boldsymbol{x} \in \Omega_l / l \neq k \ \text{ and } \ \Omega_l \text{ is adjacent to } \Omega_k \text{ and } \boldsymbol{x} \in \mathcal{C}_l, \\ \text{and distance of } \boldsymbol{x} \text{ from artificial boundary } \leq \Delta\}.\end{aligned}$$

For example, $\mathcal{B}_1$ contains points present in the neighbouring sub-domains Ω_j ($j \neq 1$) and adjacent to the artificial boundaries of Ω_1 within a distance Δ.

In addition to the above two sets of points, another set $\mathcal{S}_k$ is chosen such that for a particular sub-domain Ω_k, $\mathcal{S}_k$ contains a set of randomly chosen points from sub-domains Ω_l ($l \neq k$). Care needs to be exercised to ensure that the points in the set $\mathcal{S}$ are sparsely and evenly distributed over the other sub-domains. A schematic figure of the multizone domain decomposition is shown in Figure 4.1. Here, the whole domain is divided into three sub-domains. The solid lines indicate the natural boundaries and the dash-dotted lines indicate the artificial boundaries. In the figure the spatial distributions of the points in the sets $\mathcal{C}_1$, $\mathcal{B}_1$ and $\mathcal{S}_1$ are shown.

For each sub-domain k, we now define $\bar{\Omega}_k$ as

$$\bar{\Omega}_k = \mathcal{C}_k \bigcup \mathcal{B}_k \bigcup \mathcal{S}_k.$$

At a given time step $t = t^n$, an RBF is fitted on each of the computational sub-domain $\bar{\Omega}_k$, i.e.,

$$u_k^n(\boldsymbol{x}) = \sum_{\boldsymbol{x_j} \in \bar{\Omega}_k} \lambda_j \phi(\|\boldsymbol{x} - \boldsymbol{x_j}\|) \ \forall \boldsymbol{x} \in \bar{\Omega}_k. \tag{4.3}$$

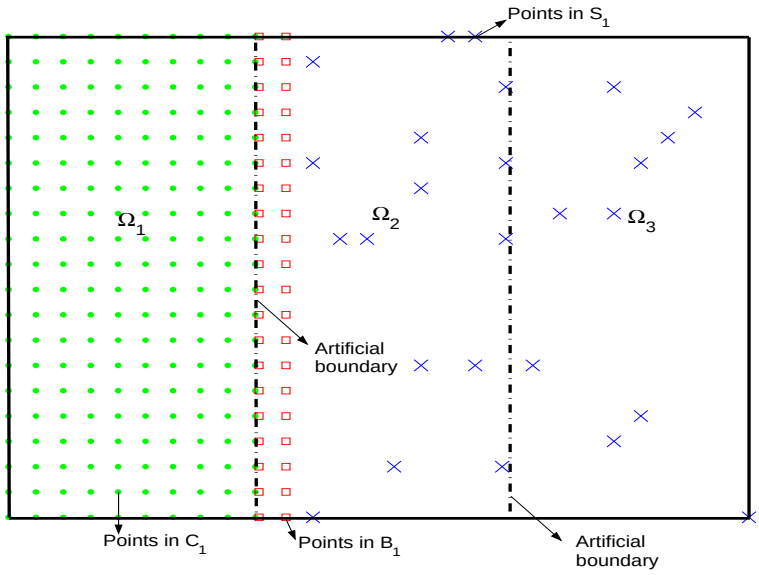


FIGURE 4.1: Point distributions in the multizone sub-domains

Note that the total number of points in the set $\bar{\Omega}_k$ denoted by $\bar{N}_k$ is less than N.

Collocating on each of the point in $\bar{\Omega}_k$, the following RBF interpolation problem for each sub-domain can be solved to obtain the RBF coefficients,

$$\boldsymbol{A}_k \boldsymbol{\lambda}_k^n = \boldsymbol{u}_k^n. \tag{4.4}$$

Note that the sub-domain coefficient matrix $\boldsymbol{A}_k$ does not vary with the time step. Hence $\boldsymbol{A}_k$ needs to be inverted only once and the inverse can be used to calculate $\boldsymbol{\lambda}_k^n$ efficiently at subsequent time steps. Once $\boldsymbol{\lambda}_k^n$ are calculated, the partial derivatives present in the operator $\mathcal{L}$ are determined for each of the collocation points present in $\mathcal{C}_k$. Note that the points present in $\mathcal{B}_k$ and $\mathcal{S}_k$ are $\mathcal{C}$-points of some other sub-domain.

The solution is advanced to the next time step $t = t^{n+1}$ for Ω_k using Equation (4.2). Once $u_k^{n+1}(\boldsymbol{x})$ is obtained for each sub-domain, the solution over the whole domain Ω is

$$u^{n+1}(\boldsymbol{x}) = u_k^{n+1}(\boldsymbol{x})\chi_k(\boldsymbol{x}), \tag{4.5}$$

where

$$\chi_k(\boldsymbol{x}) = \begin{cases} 1, & \text{if} \quad \boldsymbol{x} \in \Omega_k \\ 0, & \text{if} \quad \boldsymbol{x} \notin \Omega_k \end{cases}$$

The continuity of the solution across the artificial boundaries of sub-domain Ω_k is satisfied in an indirect fashion by including the data point sets $\mathcal{B}_k$ and $\mathcal{S}_k$. It is worth noting that the explicit multizone scheme is easy to parallelise since the sub-domain RBF interpolation problems are independent of each other.

4.3 Overlapping Schwarz domain decomposition methods

In this subsection, we present the additive and multiplicative Schwarz algorithms. The computational domain Ω is divided into k non-overlapping sub-domains. Each of the sub-domains Ω_k is extended to a larger sub-domain with overlap δ_{kl} between neighbouring regions Ω_l and Ω_k.

We denote the extended sub-domain, its natural boundary and the artificial boundary overlapped with other neighbouring sub-domains as Ω_k, $\partial\Omega_k$ and Γ_k, respectively. Let $\bar{\Omega}_k = \Omega_k \bigcup \partial\Omega_k \bigcup \Gamma_k$ denote the closed sub-domain, $\mathcal{S}$ denote the artificial boundary operator and γ_k be the artificial boundary value of the sub-domain $\bar{\Omega}_k$ extracted from neighbouring sub-domains. Figure 4.2 shows how the sub-domains are formed when the whole domain Ω is divided into two sub-domains. The spatial distribution of the collocation points is also shown in this figure.

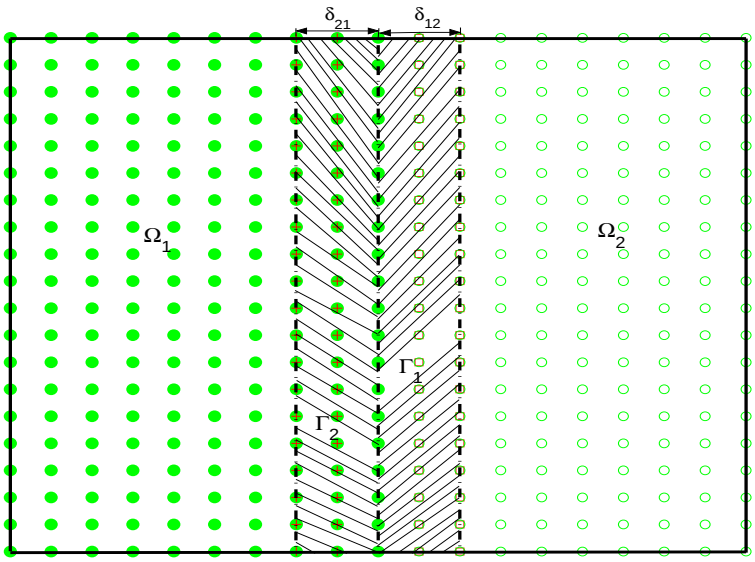


FIGURE 4.2: Point distributions in the Schwarz sub-domains

Applying the RBF-Theta method presented in Chapter 3 to each closed sub-domain $\bar{\Omega}_k$, an operator problem of the following form can be arrived at:

$$\begin{aligned} u_k(\boldsymbol{x}, t^{n+1}) + \delta t\ \theta \mathcal{L} u_k(\boldsymbol{x}, t^{n+1}) &= u_k(\boldsymbol{x}, t^n) - \delta t\ (1-\theta)\mathcal{L} u_k(\boldsymbol{x}, t^n) + \delta t\ f_k(\boldsymbol{x}, t^{n+1}) && \boldsymbol{x} \in \Omega_k \\ \mathcal{B} u_k(\boldsymbol{x}, t^{n+1}) &= g_k(\boldsymbol{x}, t^{n+1}) && \boldsymbol{x} \in \partial\Omega_k \\ \mathcal{S} u_k(\boldsymbol{x}, t^{n+1}) &= \gamma_k(\boldsymbol{x}, t^{n+1}) && \boldsymbol{x} \in \Gamma_k, \end{aligned} \tag{4.6}$$

where $\mathcal{B}$ and $\mathcal{S}$ can be Dirichlet, Neumann or Robin-type boundary conditions. The basic

idea of the overlapping Schwarz schemes is to solve Equation (4.6) for each sub-domain in an iterative fashion.

For each iteration i, Equation (4.6) can be compactly written in matrix form as

$$\boldsymbol{H}_k^+ \boldsymbol{\lambda}_{i,k}^{n+1} = \boldsymbol{H}_k^- \boldsymbol{\lambda}_k^n + \boldsymbol{F}_k^{n+1} + \boldsymbol{S}_{i-1,k}^{n+1}, \tag{4.7}$$

where

$$\boldsymbol{H}_k^+ = \begin{bmatrix} \boldsymbol{\Phi}_{d,k} + \delta t\ \theta \mathcal{L} \boldsymbol{\Phi}_{d,k} \\ \mathcal{B} \boldsymbol{\Phi}_{b,k} \\ \mathcal{S} \boldsymbol{\Phi}_{s,k} \end{bmatrix},\ \boldsymbol{H}_k^- = \begin{bmatrix} \boldsymbol{\Phi}_{d,k} - \delta t (1-\theta) \mathcal{L} \boldsymbol{\Phi}_{d,k} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

and

$$\boldsymbol{F}_k^{n+1} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{g}_k^{n+1} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{S}_{i-1,k}^{n+1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\gamma}_{i-1}^{n+1} \end{bmatrix}.$$

As can be seen from Equation (4.7), only $\boldsymbol{S}_{i-1,k}^{n+1}$ changes across the iterations in the right hand side of the equation, for a particular time $t = t^{n+1}$. Note that the matrix $\boldsymbol{H}_k^+$ for each sub-domain does not change over iterations and hence needs to be inverted only once. For the case when $\mathcal{B}$ and $\mathcal{S}$ are identity operators (Dirichlet conditions), the stability analysis developed for the RBF-Theta method can be applied. Hence, the Schwarz schemes are unconditionally stable for $\theta = 0.5$ and $\lambda_{\boldsymbol{M}} \geq 0$.

In the Schwarz additive algorithm, the values on the artificial boundaries are updated after solving the operator problem for all the sub-domains. On the common boundary, the value of the field variable is taken to be the average of the field variable values obtained from solving the individual sub-domain problems. This enforces continuity of the field variable across the whole domain.

In the Schwarz multiplicative version of the algorithm, the field variable value is updated in a sequential fashion. First, we solve Equation (4.7) on the first sub-domain. Then, the boundary values of the field variable are updated. Then we solve Equation (4.7) for the next sub-domain. We continue in this fashion, until we obtain convergence. Note that parallel implementation of the multiplicative Schwarz scheme is much more difficult than that of the additive Schwarz scheme.

4.4 Computational effort

In this section, the computational effort required of the Schwarz schemes is estimated. It is compared with the global RBF-Theta method and the explicit multizone method.

It can be shown that the operation count of the RBF-Theta method may be shown to be

$$\chi_{\mathrm{RBF-Theta}} \approx 4N^2p + 2N^3p + Nn_dp + T\left(Nn_dp + n_bp + N^2p\right), \tag{4.8}$$

where $N = n_d(\text{interior}) + n_b(\text{boundary})$ is the number of collocation points, $T = \frac{t_f}{\delta t}$ is the number of time steps and p is one unit cost of evaluating the RBF. In Equation (4.8), the first term is the computational cost required to formulate the matrices $\boldsymbol{A}$ and $\boldsymbol{H}_+$. The second term refers to the LU decompositions of the above two matrices. If the given set of collocation points does not change, then we need to decompose the coefficient matrices only once. The third term comes from obtaining the values of the RBF coefficients from the initial condition. Finally, the fourth term represents the computational cost incurred over T time evaluations.

Next, to calculate the operation count for the explicit method, we assume N_e as the number of sub-domains used and $s \ll \frac{N}{N_e}$ refers to the number of extra points needed in each sub-domain (deriving from the subset $\mathcal{S}$). Also $T_{\mathrm{ex}} = \frac{t_f}{\delta t}$ is the number of time evaluations needed to reach the final time t_f.

The operation count for the explicit multizone method is

$$\chi_{\mathrm{explicit}} \approx \overbrace{N_e\left[\left(\frac{N}{N_e}+s\right)^2 p + 2\left(\frac{N}{N_e}+s-n_b\right)^2 p\right]}^{\text{Matrix Formulations}} + \overbrace{N_e\left[\left(\frac{N}{N_e}+s\right)^3 p\right]}^{\text{LU Decompositions}} + \underbrace{T_{\mathrm{ex}}N_e\left(\frac{N}{N_e}+s\right)\left(3\frac{N}{N_e}+3s-2n_b\right)p}_{\text{Time evaluations}} \tag{4.9}$$

In Equation (4.9), the first term comes from the matrix formulations, the second term from the LU factorisations of the N_e sub-domain Gram matrices and the third term is the operation count happening over T_{ex} time evaluations.

We now present the operation count for the Schwarz DDM (additive and multiplicative). We denote the percentage of overlap (assumed constant over all the sub-domains) as δ. Hence, the number of collocation points in each sub-domain are approximately $(\frac{N}{N_e} + \delta N)$. Let n_{iter} be the number of Schwarz iterations needed to converge at each time step (assumed

constant). Then the operation count for the Schwarz schemes is given by

$$\chi_{\text{Schwarz}} \approx \overbrace{N_e\left[2\left(\frac{N}{N_e}+\delta N\right)^2 p+\left(\frac{N}{N_e}-n_b\right)^2 p\right]}^{\text{Matrix Formulations}}+\overbrace{2N_e\left(\frac{N}{N_e}+\delta N\right)^3 p}^{\text{LU Decompositions}}$$
$$+N_e\left(\frac{N}{N_e}+\delta N\right)^2 p+\underbrace{T\left[n_{\text{iter}}\ N_e\left(\frac{N}{N_e}+\delta N\right)\left(\frac{2N}{N_e}+\delta N-n_b\right)\right]p}_{\text{Time evaluations}} \tag{4.10}$$

Note that the difference between Schwarz additive and Schwarz multiplicative schemes approximately is in the value of n_{iter}.

We now compare the computational effort required by the explicit multizone method with respect to the Schwarz schemes. The following assumptions are made:

- $T_{\text{ex}} = \alpha(T \cdot n_{\text{iter}})$ and $\alpha \gg 1$.
- $n_b = \beta\frac{N}{N_e}$ $(\beta < 1)$.
- $s = \delta N = \gamma N$ $(\gamma < 1)$.

Now, subtracting Equation (4.10) from Equation (4.9), we obtain

$$\chi_{\text{explicit}} - \chi_{\text{Schwarz}} \approx p\frac{N^2}{N_e} \times$$
$$\left[\underbrace{(1-\beta)^2}_{\mathcal{O}(1)}+\underbrace{2N_e^2\gamma^2}_{\mathcal{O}(1)}+\underbrace{4\frac{N_e^2}{N}\gamma}_{\mathcal{O}(1)}-\underbrace{2(1+\gamma)^2}_{\mathcal{O}(1)}-\underbrace{\frac{N}{N_e}(1+\gamma)^3}_{\mathcal{O}(\frac{N}{N_e})}+\underbrace{(1+\gamma)}_{\mathcal{O}(1)}\underbrace{(\alpha(3\gamma-2\beta+3)+\beta-\gamma-2)}_{\mathcal{O}(\alpha)}Tn_{\text{iter}}\right]. \tag{4.11}$$

From Equation (4.11), it can be seen that

$$\chi_{\text{explicit}} - \chi_{\text{Schwarz}} \equiv \left(\mathcal{O}(\alpha\ T\ n_{\text{iter}}) - \mathcal{O}(\frac{N}{N_e})\right) >> 0.$$

4.5 RBF-DDM for unsteady convection-diffusion equation

In this section, we apply the domain decomposition algorithms to solve the unsteady convection-diffusion equation for different Peclet numbers. The two dimensional unsteady convection-diffusion problem of the form

$$\frac{\partial u}{\partial t} = \kappa_x\frac{\partial^2 u}{\partial x^2}+\kappa_y\frac{\partial^2 u}{\partial y^2}+V_x\frac{\partial u}{\partial x}+V_y\frac{\partial u}{\partial y},\ 0 \le x, y \le 1,\ t > 0 \tag{4.12}$$

is considered with the initial and boundary conditions given in Equation (3.38) and Equation (3.37) respectively. The analytical solution is given in Equation (3.39).

If we put $V_x = V_y = V$ and $\kappa_x = \kappa_y = \kappa$, for the two dimensional case we can define an analogous Peclet number as $P_e = V/\kappa$. In our numerical studies, the values of the constants a and b are taken to be 1.0 and 0.1 respectively.

For the Schwarz additive and multiplicative schemes, the artificial boundary operator $\mathcal{S}$ is chosen to be a Dirichlet operator and the iterations were terminated when $\frac{\|\boldsymbol{u}_{i,k}-\boldsymbol{u}_{i-1,k}\|_2}{\|\boldsymbol{u}_{i,k}\|_2} < 10^{-6}$ on all the sub-domains $\bar{\Omega}_k$, where i is the iteration number.

4.5.1 Effect of number of collocation points

In this section we present the results of the RBF-Theta collocation method and compare it with the three domain decomposition schemes presented in this chapter. The convection-diffusion equation was solved for three different Peclet numbers (1, 10 and 100). We use the MQ RBF in our numerical studies. The analytical expressions for the derivatives of MQ RBF are given in Appendix A. The shape parameter σ was chosen from a fixed set of values such that the residual error is a minimum for the RBF-Theta method and the obtained value of shape parameter (σ) is also used for the DDMs. The shape parameter σ was chosen such that the matrix $\boldsymbol{M}$ has positive eigenvalues to satisfy the stability conditions as derived earlier. Note that as the multizone method is an explicit method, the time step used in this case is very small ($\delta t = 10^{-5}$). In the case of RBF-Theta and Schwarz domain decomposition methods, the stability criterion does not impose any restriction on the time step for $\theta = 0.5$. We have taken a time step of $\delta t = 10^{-2}$ for these methods. The final times at which the solutions are compared with the exact solution are taken to be $t_f = 0.1$ and $t_f = 1.0$. The error ε is defined as the L-infinity norm of the difference between the analytical and the numerical solutions, i.e.,

$$\varepsilon = \|u_{\text{analytic}} - u_{\text{numerical}}\|_\infty, \tag{4.13}$$

and the computational cost is assumed to be proportional to the CPU time ($\mathcal{T}$) taken by the method. The code was written in C language and the platform used was an AMD Athlon machine with an MP 2600+ processor.

We first study the case of $P_e = 1.0$ for which the convection and diffusion effects are comparable. The analytical solution obtained is smooth. Tables 4.1 and 4.2 summarise the results obtained using DDMs when the physical domain is divided into two sub-domains for $t_f = 0.1$ and $t_f = 1.0$, respectively. 'NIter' is the average number of iterations taken for the

Schwarz schemes to converge.

In Table 4.3 and Table 4.4, we present the results obtained using the three different DDMs with the RBF-Theta method for the case of $P_e = 10.0$ at $t_f = 0.1$ and $t_f = 1.0$, respectively. Here, the convection term is one order of magnitude larger than of the diffusive term and hence the analytical solution obtained is not smooth but has a slight discontinuity in it. Finally, in Table 4.5 and Table 4.6, we present the results for $P_e = 100.0$ at times $t_f = 0.1$ and $t_f = 1.0$, respectively.

We can see from the second columns of Tables 4.1 - 4.6, that the accuracy of RBF-Theta method depends on the value of the shape parameter (σ) used. However as seen from Figure 3.2, the freedom of varying shape parameter decreases with an increase in the number of collocation points. This stability restriction and the ill-conditioning of the coefficient matrix are responsible for loss of accuracy of the RBF-Theta method for a very large number of collocation points. In this section, we concentrate on the behaviour of Schwarz schemes as compared to the explicit method and the RBF-Theta method.

TABLE 4.1: Comparison of DDM Schemes with the global RBF-Theta and explicit multizone methods: Numerical Results for $P_e = 1.0$, $t_f = 0.1$, RBF = MQ, overlap = 30%

	RBF-Theta Method		Explicit Multizone		Additive Schwarz			Multiplicative Schwarz		
N (σ)	ε	$\mathcal{T}$ (sec)	ε	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)
200 (0.60)	1.01E-04	0	7.86E-04	8	1.60E-04	5	0	2.12E-04	5	0
400 (0.40)	9.30E-06	0	5.43E-05	32	1.22E-05	10	0	1.20E-05	7	0
900 (0.20)	2.76E-05	2	1.03E-04	140	2.97E-05	15	2	2.59E-05	10	2
1600 (0.10)	9.73E-05	7	1.03E-04	442	9.75E-05	20	7	9.73E-05	12	7
3600 (0.05)	1.24E-04	48	3.60E-04	2079	1.18E-04	28	43	1.17E-04	17	40
4900 (0.01)	9.52E-04	113			9.37E-04	35	85	9.35E-04	20	78

From these tables, it can be seen that additive and multiplicative Schwarz domain decomposition techniques produce results which are as accurate as the RBF-Theta method but at less computational cost. Although the explicit multizone method produces good results, the CPU time ($\mathcal{T}$) taken by it is much larger compared to Schwarz schemes primarily because of the restriction in the time step. The CPU time taken is in agreement with the theoretical results (computational effort) presented in Section 4.4.

For the case of high Peclet numbers, the Schwarz schemes generate better results as compared to the RBF-Theta method due to better condition numbers ($\mathcal{K}$) of the sub-domain

TABLE 4.2: Comparison of DDM Schemes with the global RBF-Theta and explicit multizone methods: Numerical Results for $P_e = 1.0$, $t_f = 1.0$, RBF = MQ, overlap = 30%

	RBF-Theta Method		Explicit Multizone		Additive Schwarz			Multiplicative Schwarz		
N (σ)	ε	$\mathcal{T}$ (sec)	ε	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)
200 (0.50)	2.33E-04	0	9.07E-04	100	3.93E-04	5	0	5.09E-04	5	0
400 (0.45)	4.61E-06	1	5.54E-05	321	6.75E-06	10	1	6.67E-06	7	1
900 (0.20)	3.08E-05	5	1.17E-04	1448	3.36E-05	15	4	3.34E-05	10	4
1600 (0.10)	1.15E-04	16			1.16E-04	20	13	1.15E-04	12	13
3600 (0.05)	1.61E-04	91			1.47E-04	28	63	1.47E-04	17	63
4900 (0.01)	1.08E-03	182			1.08E-03	35	117	1.06E-03	20	117

TABLE 4.3: Comparison of DDM Schemes with the global RBF-Theta and explicit multizone methods: Numerical Results for $P_e = 10.0$, $t_f = 0.1$, RBF = MQ, overlap = 30%

	RBF-Theta Method		Explicit Multizone		Additive Schwarz			Multiplicative Schwarz		
N (σ)	ε	$\mathcal{T}$ (sec)	ε	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)
200 (0.33)	1.81E-02	0	1.86E-02	8	1.72E-02	5	0	1.74E-02	4	0
400 (0.40)	1.46E-03	1	3.74E-03	32	1.46E-03	9	0	1.46E-03	6	0
900 (0.20)	1.06E-03	2	1.19E-03	143	1.12E-03	12	2	1.11E-03	8	2
1600 (0.10)	1.52E-03	7	6.95E-04	440	1.53E-03	16	7	1.52E-03	10	6
3600 (0.05)	1.23E-03	49	2.39E-03	2041	1.23E-03	23	37	1.23E-03	14	35
4900 (0.01)	7.53E-03	104			7.08E-03	29	80	7.08E-03	17	75

coefficient matrices [for example, at $P_e = 100.0$, $N = 3600$, $\mathcal{K}_{\text{RBF}-\text{Theta}} = \mathcal{O}(10^{+14})$ and $\mathcal{K}_{\text{Schwarz}} = \mathcal{O}(10^{+13})$ for $N_e = 2$ sub-domains]. Further, from Table 4.6 it can be seen that the RBF-Theta method fails to generate results when $N > 5000$. However, the Schwarz schemes can reach to a maximum of $N = 6000$ for $N_e = 2$ sub-domains. It can be noted from Tables 4.1 - 4.6 that the multiplicative scheme converges in about half the number of iterations taken by the additive scheme. We wish to mention that the Schwarz schemes can be implemented in a parallel fashion to further speed up the computations.

4.5.2 Influence of number of domains

Here, we investigate the influence of number of sub-domains on the accuracy and computational cost when Schwarz domain decomposition methods are employed. The unsteady

TABLE 4.4: Comparison of DDM Schemes with the global RBF-Theta and explicit multizone methods: Numerical Results for $P_e = 10.0$, $t_f = 1.0$, RBF = MQ, overlap = 30%

	RBF-Theta Method		Explicit Multizone		Additive Schwarz			Multiplicative Schwarz		
N (σ)	ε	$\mathcal{T}$ (sec)	ε	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)
200 (0.55)	1.24E-02	1	2.20E-02	99	1.22E-02	5	1	1.29E-02	4	1
400 (0.40)	1.63E-03	1	4.10E-03	304	1.61E-03	9	1	1.61E-03	6	1
900 (0.20)	1.25E-03	5	2.76E-03	1431	1.28E-03	12	4	1.28E-03	8	4
1600 (0.10)	1.83E-03	16			1.93E-03	16	11	1.93E-03	10	11
3600 (0.05)	1.59E-03	98			1.68E-03	23	58	1.67E-03	14	57
4900 (0.01)	8.54E-03	183			8.32E-03	29	107	8.31E-03	17	107

TABLE 4.5: Comparison of DDM Schemes with the global RBF-Theta and explicit multizone methods: Numerical Results for $P_e = 100.0$, $t_f = 0.1$, RBF = MQ, overlap = 30%

	RBF-Theta Method		Explicit Multizone		Additive Schwarz			Multiplicative Schwarz		
N (σ)	ε	$\mathcal{T}$ (sec)	ε	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)
200 (0.35)	2.79E-01	0	2.36E-01	9	1.97E-01	13	0	2.45E-01	8	0
400 (0.40)	1.29E-01	1	6.27E-02	30	1.40E-01	28	0	1.39E-01	20	0
900 (0.20)	2.07E-02	2	2.25E-02	141	1.37E-02	12	2	1.38E-02	8	2
1600 (0.10)	1.93E-02	7	4.34E-02	441	1.49E-02	7	7	1.49E-02	5	7
3600 (0.05)	1.61E-02	48	4.28E-02	2055	1.30E-02	7	37	1.38E-02	5	37
4900 (0.01)	1.61E-02	103			1.42E-02	8	74	1.42E-02	6	73

convection-diffusion equation was solved for $P_e = 10.0$. The computational cost is measured by the CPU time taken to solve this problem. The total number of collocation points was fixed at 3600 and the physical domain was subdivided into 2, 4 and 8 sub-domains. A constant overlap of 30% was used and the final time $t_f = 1.0$. The results were compared with the single domain RBF-Theta method. The results for the additive and multiplicative Schwarz schemes are presented in Figures 4.3 and 4.4, respectively.

From these figures, we observe that as the number of sub-domains increases, the computational cost decreases. However, the accuracy of the method also suffers. Similar behaviour was observed for Peclet numbers 1 and 100.

TABLE 4.6: Comparison of DDM Schemes with the global RBF-Theta and explicit multizone methods: Numerical Results for $P_e = 100.0$, $t_f = 1.0$, RBF = MQ, overlap = 30%

	RBF-Theta Method		Explicit Multizone		Additive Schwarz			Multiplicative Schwarz		
N (σ)	ε	$\mathcal{T}$ (sec)	ε	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)	ε	NIter	$\mathcal{T}$ (sec)
200 (0.35)	2.99E-01	1	3.09E-01	99	2.20E-01	13	1	2.48E-01	8	0
400 (0.40)	1.09E-01	1	6.86E-02	319	1.22E-01	99	1	1.21E-01	39	1
900 (0.20)	2.98E-02	5	2.47E-02	1427	1.88E-02	12	5	1.87E-02	8	5
1600 (0.10)	3.93E-02	16			2.96E-02	7	14	2.96E-02	5	13
3600 (0.05)	4.53E-02	91			3.60E-02	7	67	3.60E-02	5	66
4900 (0.01)	7.02E-02	180			6.02E-02	8	124	6.02E-02	6	125
5400 (0.01)					5.97E-02	6	150	5.97E-02	5	150
6000 (0.01)					5.16E-02	6	185	5.16E-02	5	183

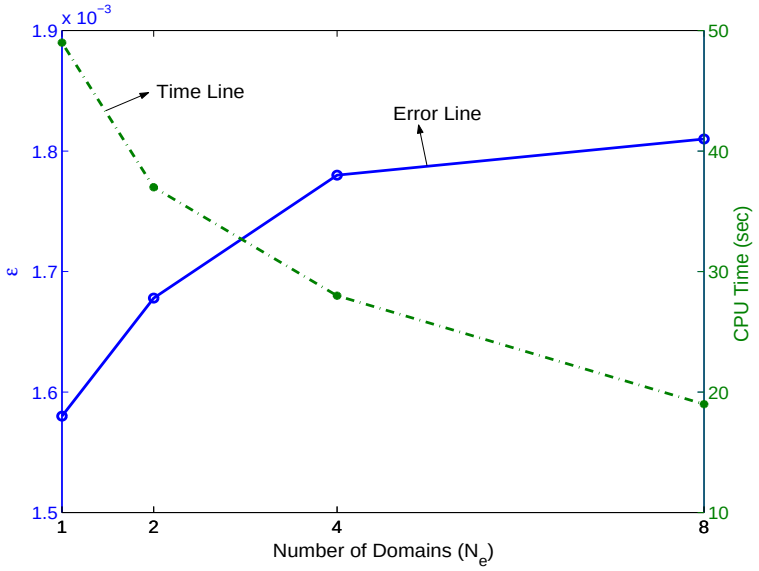


FIGURE 4.3: Comparison of accuracy and computational cost Vs number of sub-domains for the additive Schwarz scheme: $N = 3600$, $P_e = 10.0$, $t_f = 1.0$, RBF = MQ, $\sigma = 0.05$, overlap = 30%

4.5.3 Influence of overlap

In this subsection, we investigate the influence of the degree of overlapping on the Schwarz domain decomposition schemes. The degree of overlap is defined as the ratio of the physical length of the overlapping region to the physical length of each sub-domain. In this study we let $P_e = 10.0$ and the total number of collocation points was taken to be 3600. The results

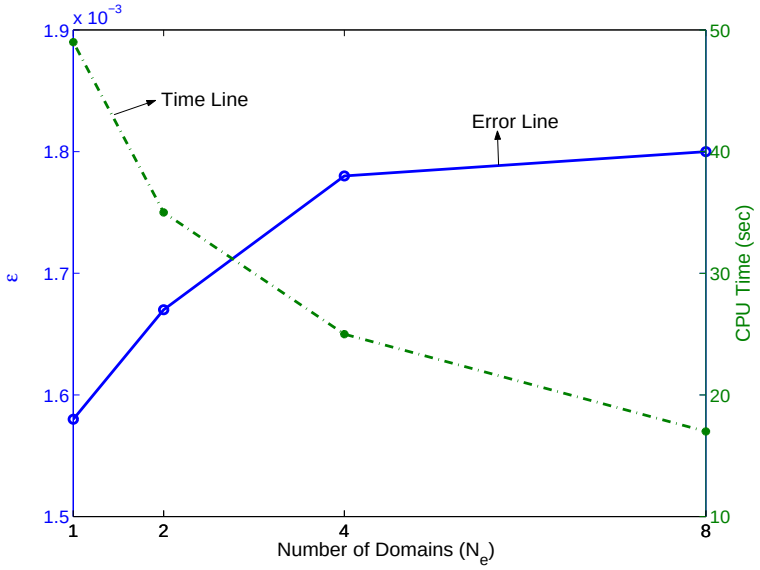


FIGURE 4.4: Comparison of accuracy and computational cost Vs number of sub-domains for the multiplicative Schwarz scheme: $N = 3600$, $P_e = 10.0$, $t_f = 1.0$, RBF = MQ, $\sigma = 0.05$, overlap = 30%

are presented in Table 4.7, when the physical domain Ω is divided into 4 sub-domains.

From Table 4.7, it can be observed that as the overlap increases, the accuracy becomes better. However, the computational cost increases as more amount of work needs to be done on the overlap regions across the iterations. After an overlap of 30%, the computational time taken is almost same as the single domain RBF-Theta method.

4.5.4 Numerical convergence

To the best of our knowledge, theoretical convergence analysis of meshless schemes for time-dependent problems using RBF collocation continues to be an open problem. In this section, we present the convergence characteristics of the domain decomposition schemes as obtained numerically in order to aid the understanding of the developed methods. The convergence history of Schwarz additive and multiplicative schemes for two cases is plotted in Figure 4.5. From the figure, it is apparent that the multiplicative scheme converges much faster than the additive scheme.

We now estimate the convergence orders of RBF-Theta and RBF-Theta with domain decomposition schemes as a function of the shape parameter (σ) and the mesh spacing (h). The error ε is assumed to be $\mathcal{O}(\chi^{(\frac{\sqrt{\sigma}}{h})^{\eta}})$, where $0 < \chi < 1$ (Cheng *et al.*, 2003). Seven different mesh spacings in the range of $h_{\min} = 0.025$ to $h_{\max} = 0.16$ are taken for single

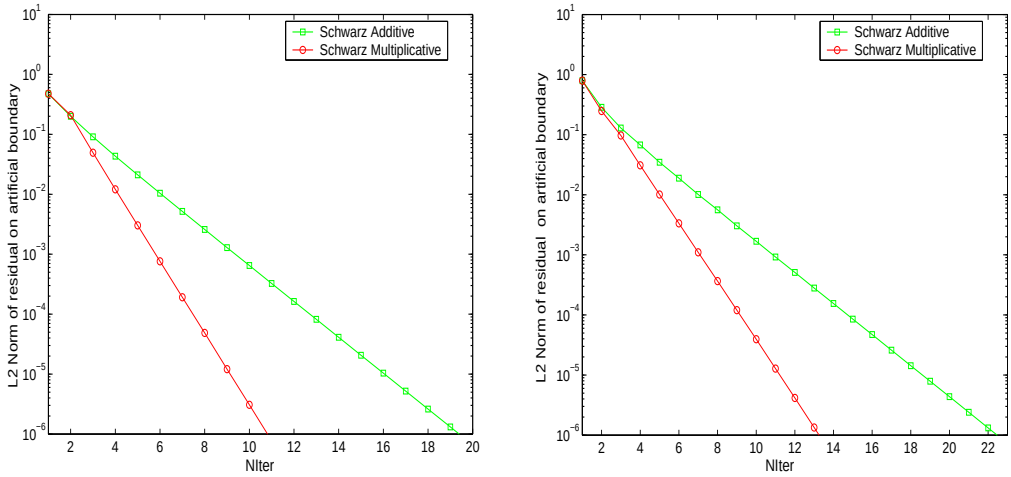
TABLE 4.7: Overlapping influence on the Schwarz DDM schemes for $P_e = 10.0$, $N = 3600$, $N_e = 4$, $t_f = 1.0$

Overlap (%)	Additive Schwarz		Multiplicative Schwarz	
	ε	CPU time (sec)	ε	CPU time (sec)
3	9.89E-03	28	9.896E-03	25
6	9.87E-03	33	9.877E-03	30
10	9.77E-03	36	9.769E-03	32
13	9.67E-03	41	9.678E-03	36
16	9.611E-03	42	9.609E-03	38
26	9.42E-03	52	9.425E-03	52
50	9.155E-03	86	9.156E-03	75
83	8.919E-03	142	8.992E-03	126
96	8.83E-03	181	8.825E-03	183

domain and four different mesh spacings for the two sub-domain case. The shape parameter was also varied in the range of $0.05 \leq \sigma \leq 0.3$ and the resulting convergence plots are shown in Figure 4.6 ($-\log_{10}(\varepsilon)$ Vs $\log_{10}(\frac{\sqrt{\sigma}}{h})$). For a constant shape parameter value, the mesh density decreases as we go to the right on the x-axis and for a constant mesh spacing the value of σ increases as we go to the right. From figure 4.6, it can also be seen that the rates of convergence for the single ($\eta = 3.6366$) and two sub-domain ($\eta = 4.0188$) cases are comparable to each other. The convergence rate, η, is obtained by a linear fit through the data. Similar behaviours were observed for $P_e = 10.0$.

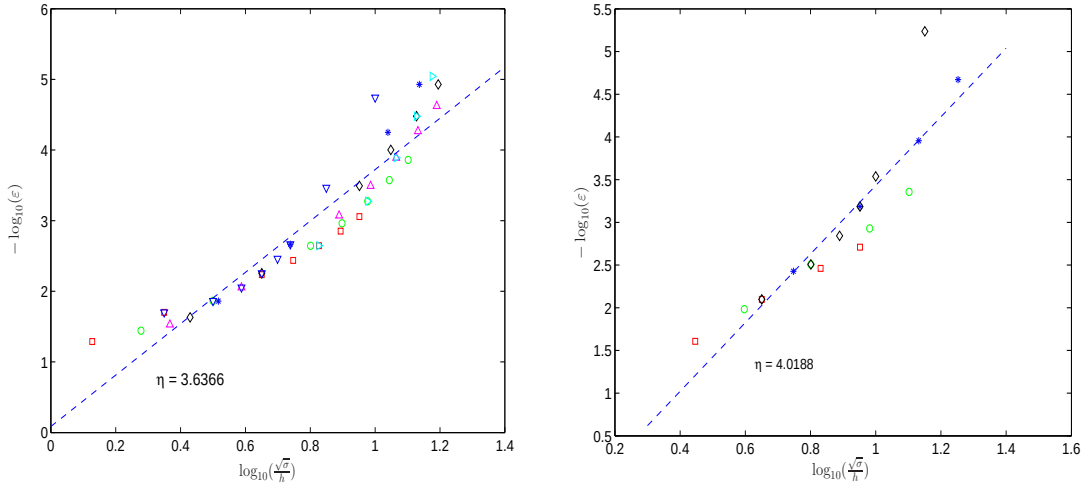
4.6 Concluding remarks

Overlapping Schwarz additive and multiplicative domain decomposition methods were developed for solving time-dependent problems using radial basis function collocation for spatial discretisation and a theta weighting scheme for temporal discretisation. The developed methods are applied to a 2D unsteady convection-diffusion equation and are compared with the global RBF-Theta collocation method presented in the Chapter 3 and an explicit multizone method described in the literature. Numerical results obtained show that the domain decomposition methods give results which are as accurate as the RBF-Theta method but at much



Two sub-domains Eight sub-domains

FIGURE 4.5: Convergence history on artificial boundaries for additive and multiplicative Schwarz schemes ($N = 1600$, $P_e = 1.0$, $\sigma = 0.1$)



Single domain Two sub-domains

FIGURE 4.6: Convergence plot of RBF-Theta and RBF-DDM scheme for $P_e = 1.0$, $t_f = 1.0$

less computational cost, particularly for very large number of collocation points. However as the number of sub-domains are increased, the accuracy suffers. The developed DDM schemes can be made even more computationally effective by parallel implementation. However, the problem of obtaining the optimal value of shape parameter still exists and the optimisation strategy outlined in Section 3.3 is computationally expensive. In the next chapter, we explore an alternative strategy of using RBFs in a finite difference mode as opposed to the

collocation methods proposed till now for solving partial differential equations.

Chapter 5

RBFs in a Finite-Difference Mode (RBF-FD)

In the previous chapter, overlapping domain decomposition schemes were formulated for time dependent PDEs to improve the computational efficiency when a large number of collocation points are needed. This chapter outlines an alternative RBF based formulation which generates a local interpolant using scattered data RBF interpolation method, thus generating sparse coefficient matrices. This idea of using RBFs in a finite-difference mode (RBF-FD) was proposed by Wright & Fornberg (2006), Shu *et al.* (2003) and Tolstykh & Shirobokov (2003) independently in the literature.

We begin by looking at the finite difference methodology. Consider a typical central finite difference scheme for estimating the derivative of function $u(x, y)$ with respect to x. The function derivative at any grid point (i, j) can be written in the form

$$\left.\frac{\partial u}{\partial x}\right|_{(i,j)} \approx \sum_{k\in\{i-1,i,i+1\}} w_{(k,j)} u_{(k,j)}, \tag{5.1}$$

where $u_{(k,j)}$ is the function value at the grid point (k, j).

The unknown coefficients $w_{(k,j)}$ are obtained using polynomial interpolation or Taylor series (Fornberg, 1996, 1998). The set of nodes $\{(i-1, j), (i+1, j)\}$ along with the node (i, j) are collectively referred to as stencil in the finite difference literature. The polynomial interpolation strategy however imposes a restriction that the nodes in the stencil be situated on some kind of a structured grid. This restriction can be circumvented if the approximation of the function derivative can be written as a linear combination of function values on a scattered set of nodes in the stencil. The methodology for obtaining the coefficients or weights of the FD formulas becomes the focus of the issue now. Abgrall (1994); Schonauer &

Adolph (2001) extended the classical polynomial interpolation technique. This however leads to several ambiguities in deriving the scattered node FD formulas and also the problem of well-posedness for polynomial interpolation in more than one dimension (Wright & Fornberg, 2006).

In the RBF-FD concept, the weights of the FD formulas are obtained using the RBF interpolation technique. This approach has the following merits:

- The problem of well-posedness in polynomial interpolation is overcome as RBF interpolation is well-posed in multidimensional problems.
- RBF interpolants are capable of accurately approximating the function derivatives (Tolstykh & Shirobokov, 2003).
- Since the approximations are based on scattered nodes with no connectivity information, this method can be regarded as a truly meshless method.

5.1 Basic formulation

In this section, we present a brief outline of the RBF-FD formulation for solving partial differential equations. We begin with a recap of the scattered RBF interpolation problem using the multiquadric RBF. Given a set of distinct nodes $\boldsymbol{x}_i \in \mathbb{R}^d$, $i = 1, \cdots, n$, and a corresponding scalar function values $u(\boldsymbol{x}_i)$, $i = 1, \cdots, n$, the standard RBF interpolation problem is to find an interpolant of the form

$$u(\boldsymbol{x}) \approx s(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\boldsymbol{x} - \boldsymbol{x}_i\|) + \beta, \tag{5.2}$$

where $\phi(\|.\|)$ is the multiquadric RBF and β is a constant. The expansion coefficients $\{\lambda_i\}_{i=1}^n$ and β are determined by enforcing the conditions $s(\boldsymbol{x}_i) = u(\boldsymbol{x}_i), \quad i = 1, \cdots, n$, and $\sum_{i=1}^n \lambda_i = 0$. Imposing these conditions leads to a symmetric block linear system of equations

$$\begin{bmatrix} \Phi & \mathbf{e} \\ \mathbf{e}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \beta \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ 0 \end{bmatrix}, \tag{5.3}$$

where $\Phi_{i,j} = \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \quad i, j = 1, \cdots, n$, and $\boldsymbol{e}_i = 1, \quad i = 1, \cdots, n$.

The RBF interpolant can alternatively be written in Lagrange form as

$$\bar{s}(\boldsymbol{x}) = \sum_{i=1}^{n} \chi(\|\boldsymbol{x} - \boldsymbol{x}_i\|) u(\boldsymbol{x}_i), \tag{5.4}$$

where $\chi(\|\boldsymbol{x} - \boldsymbol{x}_i\|)$ is of the form Equation (5.2) and satisfies the usual cardinal conditions i.e.,

$$\chi(\|\boldsymbol{x} - \boldsymbol{x}_i\|) = \begin{cases} 1, & \text{if } \boldsymbol{x} = \boldsymbol{x}_i, \\ 0, & \text{if } \boldsymbol{x} \neq \boldsymbol{x}_i, \end{cases} \quad i = 1, \cdots, n. \tag{5.5}$$

The basic idea of the RBF-FD methodology is to approximate function derivatives as a linear combination of the function values like in Equation (5.1). In the derivation that follows, we present the RBF-FD methodology for approximating any arbitrary linear differential operator acting on the function $u(\boldsymbol{x})$, denoted by $\mathcal{L}u(\boldsymbol{x})$. The unknown function $u(\boldsymbol{x})$ at any node, say $\boldsymbol{x}_1$, in the domain is approximated by an RBF interpolant with the centres placed on the node itself and some $n-1$ surrounding nodes. These n nodes constitute the support region/stencil for the node $\boldsymbol{x}_1$. A schematic diagram of the support region for the node $\boldsymbol{x}_1$ is shown in the Figure 5.1.

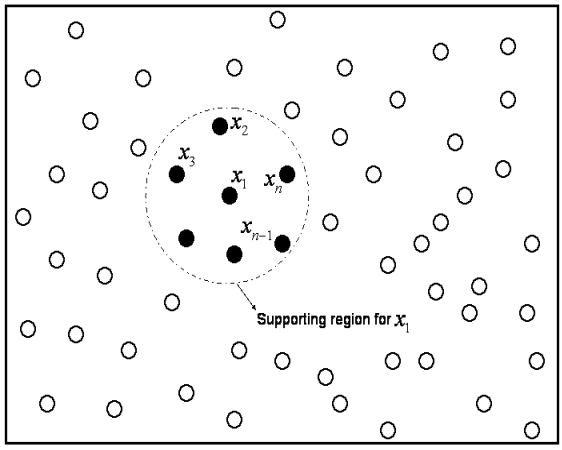


FIGURE 5.1: Schematic diagram of a RBF-FD stencil. The circle indicates the supporting region/stencil for the node $\boldsymbol{x}_1$

To derive RBF-FD formula at the node $\boldsymbol{x}_1$, we approximate the differential operator using the Lagrangian form of the RBF interpolant i.e.,

$$\mathcal{L}u(\boldsymbol{x}_1) \approx \mathcal{L}\bar{s}(\boldsymbol{x}_1) = \sum_{i=1}^{n} \mathcal{L}\chi(\|\boldsymbol{x}_1 - \boldsymbol{x}_i\|)u(\boldsymbol{x}_i). \tag{5.6}$$

Equation (5.6) can be rewritten as a FD formula of the form

$$\mathcal{L}u(\boldsymbol{x}_1) \approx \sum_{i=1}^{n} w_{(1,i)}u(\boldsymbol{x}_i), \tag{5.7}$$

where the RBF-FD weights $\{w_{(1,i)}\}_{i=1}^{n}$ are formally given by the operator $\mathcal{L}$ applied on the

Lagrange form of the basis functions i.e.,

$$w_{(1,i)} = \mathcal{L}\chi(\|\boldsymbol{x}_1 - \boldsymbol{x}_i\|). \tag{5.8}$$

In practise, the RBF-FD weights are computed by solving the linear system

$$\begin{bmatrix} \Phi & \mathbf{e} \\ \mathbf{e}^T & \mathbf{0} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{w} \\ \mu \end{bmatrix} = \begin{bmatrix} \mathcal{L}\Phi_1 \\ 0 \end{bmatrix} \tag{5.9}$$

where $\mathcal{L}\Phi_1$ denotes the evaluation of the column vector $\mathcal{L}\Phi = [\mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{x}_1\|)\ \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{x}_2\|)\ \cdots\ \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{x}_n\|)]^T$ at the node $\boldsymbol{x}_1$. Here, μ is a scalar value related to the constant β in Equation (5.2) and enforces the condition

$$\sum_{i=1}^{n} w_{(1,i)} = 0,$$

which ensures that the stencil is exact for all constants. The complete derivation is given in Appendix C.

As the differential operator $\mathcal{L}$ can be arbitrary, a similar procedure can be used to obtain the weights for all function derivatives. The convention followed for denoting the weights for any point $\boldsymbol{x}_i$ with n supporting points is $\boldsymbol{w}_i^{(x)}$,$\boldsymbol{w}_i^{(y)}$,$\boldsymbol{w}_i^{(xx)}$ or $\boldsymbol{w}_i^{(yy)}$ when the operator ($\mathcal{L}$) is $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, $\frac{\partial^2}{\partial x^2}$ or $\frac{\partial^2}{\partial y^2}$, respectively. Once the coefficients are computed, they are stored and used to discretise the partial differential equation in a similar manner as in the finite difference method.

We now illustrate the RBF-FD approach for a Poisson problem of the form

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (x, y) \in \Omega \tag{5.10}$$

with Dirichlet boundary condition

$$u(x, y) = g(x, y), \quad (x, y) \in \Gamma \tag{5.11}$$

where Γ represents the boundary of the domain Ω.

We consider a five noded stencil for evaluating the function derivatives. At the node $\boldsymbol{x}_1$ with the stencil $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5\}$, the weights for $\frac{\partial^2 u}{\partial x^2}|_{\boldsymbol{x}=\boldsymbol{x}_1}$ and $\frac{\partial^2 u}{\partial y^2}|_{\boldsymbol{x}=\boldsymbol{x}_1}$ are obtained by solving Equation (5.9) written in expanded form as

$$\begin{bmatrix} \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_5\|) & 1 \\ \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_5\|) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi(\|\boldsymbol{x}_5 - \boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_5 - \boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_5 - \boldsymbol{x}_5\|) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1^{(xx)} \\ w_2^{(xx)} \\ \vdots \\ w_5^{(xx)} \\ \mu \end{bmatrix} = \begin{bmatrix} \phi_{,xx}(\boldsymbol{x}_1, \boldsymbol{x}_1) \\ \phi_{,xx}(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ \vdots \\ \phi_{,xx}(\boldsymbol{x}_1, \boldsymbol{x}_5) \\ 0 \end{bmatrix}, \tag{5.12}$$

$$\begin{bmatrix} \phi(\|\boldsymbol{x}_1-\boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_1-\boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_1-\boldsymbol{x}_5\|) & 1 \\ \phi(\|\boldsymbol{x}_2-\boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_2-\boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_2-\boldsymbol{x}_5\|) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi(\|\boldsymbol{x}_5-\boldsymbol{x}_1\|) & \phi(\|\boldsymbol{x}_5-\boldsymbol{x}_2\|) & \cdots & \phi(\|\boldsymbol{x}_5-\boldsymbol{x}_5\|) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1^{(yy)} \\ w_2^{(yy)} \\ \vdots \\ w_5^{(yy)} \\ \mu \end{bmatrix} = \begin{bmatrix} \phi_{,yy}(\boldsymbol{x}_1,\boldsymbol{x}_1) \\ \phi_{,yy}(\boldsymbol{x}_1,\boldsymbol{x}_2) \\ \vdots \\ \phi_{,yy}(\boldsymbol{x}_1,\boldsymbol{x}_5) \\ 0 \end{bmatrix}, \tag{5.13}$$

where $\phi_{,xx}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \frac{\partial^2}{\partial x^2}\left[\phi(\|\boldsymbol{x}-\boldsymbol{x}_j\|)\right]\big|_{\boldsymbol{x}=\boldsymbol{x}_i}$ and $\phi_{,yy}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \frac{\partial^2}{\partial y^2}\left[\phi(\|\boldsymbol{x}-\boldsymbol{x}_j\|)\right]\big|_{\boldsymbol{x}=\boldsymbol{x}_i}$.

It can be seen that the RBF-FD weights solely depend on the relative positioning of the nodes and the basis functions used. Once these two parameters are defined for a particular problem, the estimation of weights can be done in the pre-processing stage. Once the RBF-FD weights are obtained, the discretisation of the governing PDE at the node $\boldsymbol{x}_1$ gives

$$\sum_{j=1}^{5}(w_{(1,j)}^{(xx)} + w_{(1,j)}^{(yy)})u(\boldsymbol{x}_j) = f_1. \tag{5.14}$$

This procedure is repeated for each of the interior nodes to obtain the discretised form of the PDE at each of the interior nodes. Substituting the function values u_j from boundary condition Equation (5.11) whenever the support point $\boldsymbol{x}_j \in \Gamma$, we obtain a system of equations which can be written in matrix form as

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}, \tag{5.15}$$

where $\boldsymbol{u}$ is the vector of the unknown function values at all the interior nodes and $\boldsymbol{f}$ is the source vector including the boundary terms. Note that the matrix $\boldsymbol{A}$ is sparse and well-conditioned and can hence be effectively inverted.

5.2 Implementation and computational aspects

In this section, we outline some of the implementation and computational aspects for the RBF-FD formulation.

5.2.1 Determination of local support for interior nodes

As the derivative approximation for each interior node is performed in a local support region, the problem of determining the local support at each interior node is of importance. A static or dynamic approach can be pursued in order to obtain the support for the node.

In the static approach, the support region for each interior nodal point is fixed before performing the discretisation of the governing PDE. For example, the local support can be

a circle in two-dimension and a sphere in three dimension i.e, for a central node $\boldsymbol{x}_i$, its local support is defined as

$$\text{Support}(\boldsymbol{x}_i) = \{j : 0 < \|\boldsymbol{x}_j - \boldsymbol{x}_i\| < R_i\} \tag{5.16}$$

where R_i is the radius of the circle or sphere, and represents the size of the local support. An alternative approach would be to choose the $n-$nearest neighbours of the point $\boldsymbol{x}_i$.

In the dynamic or adaptive approach, one can select the support points adaptively by deriving a suitable *a posteriori* estimate. This approach can be very useful for solving problems which exhibit a large variation in local characteristics. For example, the support of a node in a region is more when the function gradients are changing rapidly and in regions where the variations are less or smooth, the support may be smaller.

Further, in both the approaches the essential point in computational cost terms is searching for $n-1$ nodes based on a certain criterion (e.g. Euclidean distance) to form the stencil for a particular node. If the complete domain is represented by N discrete points, the computational cost incurred would be $\mathcal{O}(N^2)$. The efficiency of the searching algorithm can be increased by obtaining a triangulation of the whole node distribution which then will enable to locate the supporting points of a particular node quickly. Another idea as implemented in Cecil *et al.* (2004) is to use a binning method. In this method, the entire domain is divided into a coarse structured grid C. Then for each coarse grid cell a list of all nodes that lie inside the cell is created. Hence, to determine the support for a node lying in the coarse grid cell, we need to determine the n nearest nodes by searching through the list of that particular cell. The binning method can be made more efficient by recursively dividing each coarse cell. Note that all these approaches implicitly make use of connectivity information.

5.2.2 Shape parameter

Another factor of importance in determining the accuracy of the numerical approximation by RBFs is the shape parameter value. Wright & Fornberg (2006) indicated that for infinitely smooth radial functions like the multiquadric, the standard RBF interpolant converges to the Lagrange polynomial as the shape value tends to zero, under certain conditions (Driscoll & Fornberg, 2002). This in effect translates to that in the limit, the RBF-FD stencils recover the traditional FD formulas, at least in the case of one dimension. In Wright & Fornberg (2006), the results for RBF-FD method were obtained by varying the shape parameter in a certain fixed range. Shu *et al.* (2003) and Cecil *et al.* (2004) proposed normalisation of scale in the support region, motivated by the finite element method where each element is usually

mapped into a regular shape in the computational space. These strategies for obtaining the shape parameter are however not optimal.

In this chapter, we outline a strategy based on the leave-one-out statistical criterion for determining the optimal value of the shape parameter for a RBF-FD stencil.

5.2.3 Incorporation of boundary conditions

The implementation of the RBF-FD is straightforward when the prescribed boundary conditions are Dirichlet in nature. However, when Neumann or mixed boundary conditions are specified, the implementation issues need to be explained in more detail. On the lines of the RBF-FD discretisation performed for interior nodes, one can define a support region for the boundary nodes and obtain the corresponding weights for the derivatives. This approach has the advantage of consistent discretisation throughout the domain, but suffers from the decrease of accuracy as the information is based from only one side of the boundary (Ding *et al.*, 2005). Shu *et al.* (2003) proposed so-called locally orthogonal grids around boundaries, and then discretised the derivatives by one-sided finite difference schemes. Although this approach has the advantage of obtaining accurate discretisations on the boundary, it may be tedious to generate such grids for complex geometries. Later in Chapter 7, when the RBF-FD method is used to solve the incompressible Navier-Stokes equations, we present a ghost node based strategy for incorporating boundary conditions which alleviates the necessity of any special node placements near the boundary.

5.3 RBF-FD Vs RBF collocation

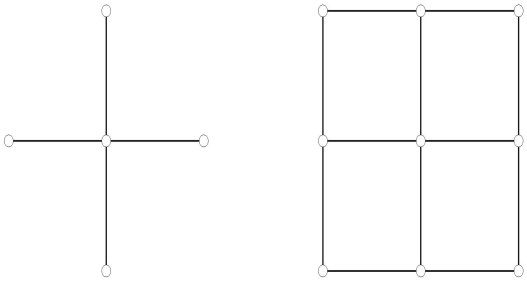
In this section, we study the performance and accuracy of the RBF-FD approach on a model Poisson problem and compare the results obtained with the global RBF collocation method. Consider the Poisson equation of the form as in Equation (5.10) in a $[0,1] \times [0,1]$ square domain with

$$\begin{aligned} f(x,y) \;=\; & -\tfrac{751\pi^2}{144}\sin(\tfrac{\pi x}{6})\sin(\tfrac{7\pi x}{4})\sin(\tfrac{3\pi y}{4})\sin(\tfrac{5\pi y}{4}) + \\ & \tfrac{7\pi^2}{12}\cos(\tfrac{\pi x}{6})\cos(\tfrac{7\pi x}{4})\sin(\tfrac{3\pi y}{4})\sin(\tfrac{5\pi y}{4}) + \\ & \tfrac{15\pi^2}{8}\sin(\tfrac{\pi x}{6})\sin(\tfrac{7\pi x}{4})\cos(\tfrac{3\pi y}{4})\cos(\tfrac{5\pi y}{4}). \end{aligned} \tag{5.17}$$

The exact solution of this problem (Cheng *et al.*, 2003) is given by

$$u_{\text{ex}}(x,y) = \sin(\frac{\pi x}{6})\sin(\frac{7\pi x}{4})\sin(\frac{3\pi y}{4})\sin(\frac{5\pi y}{4}). \tag{5.18}$$

We assume Dirichlet boundary conditions on all four sides with its value calculated from the exact solution Equation (5.18).

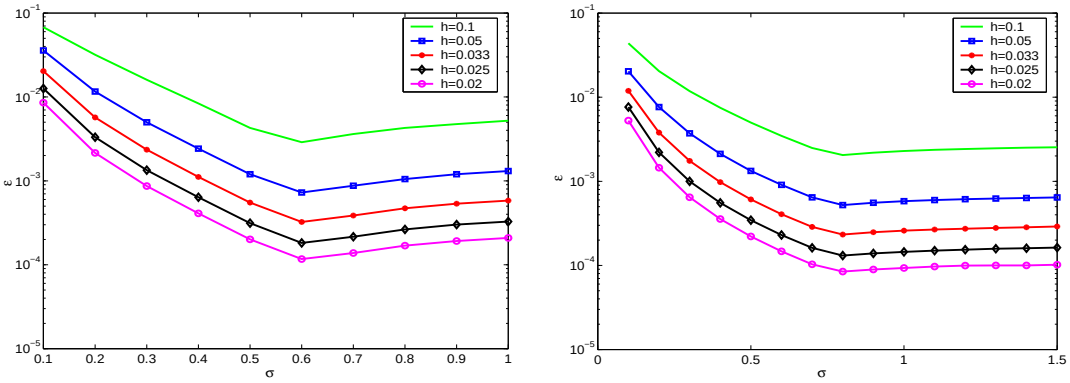


RBF-FD: Stencil A RBF-FD: Stencil B

FIGURE 5.2: RBF-FD stencils

We consider two types of supporting regions/stencils as shown in Figure 5.2. Figure 5.3 displays the L_∞ norm of the error (ε) at different values of the shape parameter (σ) and mesh spacings (h) for each of the considered stencils. Figure 5.4 presents the results for the same Poisson problem as obtained by the RBF collocation method. It can be seen that the global RBF collocation method gives more accurate results than the RBF-FD method provided an optimal value of the shape parameter is used. However, the sensitivity of the solution with respect to the shape parameter is more in RBF collocation method as compared to the RBF-FD method. This high sensitivity of RBF collocation method makes it very difficult for any gradient based optimisation strategy to obtain the shape parameter value, and some sort of smoothing strategies like polynomial fitting (see Chapter 3) need to be used with the objective function to obtain the optimal shape parameter value. This is not the case for RBF-FD method. Another advantage of the RBF-FD method as can be seen in Figure 5.3 is that for a fixed value of shape parameter, the numerical solution is guaranteed to converge as the mesh spacing is decreased. This behaviour is not guaranteed for the collocation method as can be seen in Figure 5.4.

As presented in previous chapters, the global RBF collocation method has limitations in applications to large scale problems because of the dense and ill-conditioned coefficient matrices. The RBF-FD method alleviates this problem. The coefficient matrices generated are sparse and well-conditioned; In the case of uniform discretisations, the coefficient matrices are banded. Figure 5.5 shows the sparsity patterns of the coefficient matrices obtained for a Poisson problem with different radii of support. Note that the sparsity mimics the standard FD discretisations as the nodes are placed in an uniform fashion. The sparsity of



RBF-FD: Stencil A RBF-FD: Stencil B

FIGURE 5.3: Accuracy of RBF-FD method for Poisson problem using Stencil A and Stencil B

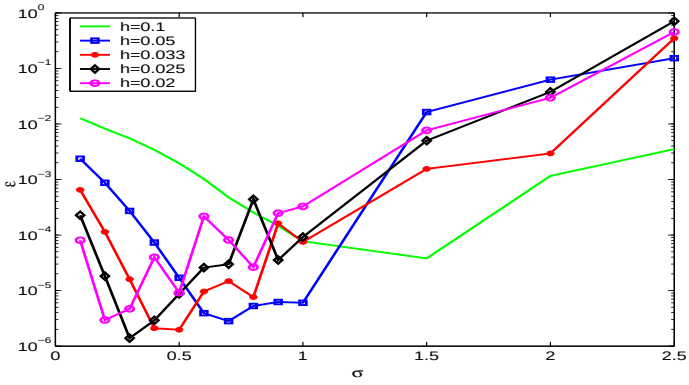
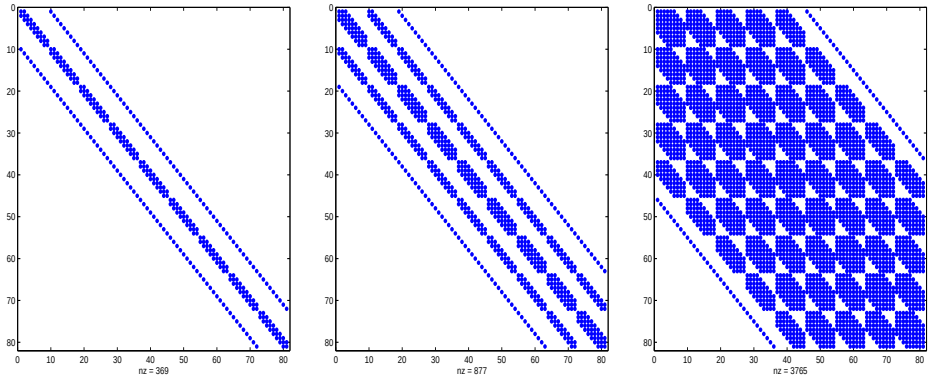


FIGURE 5.4: Accuracy of RBF collocation method for Poisson problem

the coefficient matrices facilitates reduced storage and faster evaluation of numerical solutions at almost comparable accuracy.

5.4 Shape parameter tuning

The RBF-FD method essentially obtains an approximation of the function derivative at a node as a linear combination of function values on its supporting nodes, with its weights obtained using the standard RBF interpolation method. The generalisation performance or the degree of smoothness of the RBF interpolant can depend to a significant extent on the value of shape parameter. It can also be seen from Figure 5.3 that the accuracy of the RBF-FD method also depends significantly on the value of the shape parameter used. Hence, it is of interest to examine techniques for estimating the optimum value of the shape parameter



Radius of support=0.1 Radius of support=0.2 Radius of support=0.5

FIGURE 5.5: Sparsity patterns in RBF-FD coefficient matrices

in order to ensure good generalisation performance of each RBF-FD stencil. In this section, we present a methodology for obtaining the optimal value of the shape parameter for RBF-FD stencils based on the technique of cross-validation commonly employed in statistical data modelling. This methodology has been earlier utilised in context of scattered data interpolation and regression problems; see, for example Rippa (1999); Wang (2004).

5.4.1 Cross-validation and Leave-One-Out (LOO) procedure

The idea of cross-validation is usually employed to determine the effectiveness of a particular model/interpolant. Given the set of data and the observed values at each data point, the methodology involves partitioning the data set into N clusters which may or may not be of equal size. Using the $N-1$ partitions (*learning set*) to construct a model and then predicting the values on the remaining cluster (*validation set*), an error value for the model capability can be obtained. Each of the N error terms can then be averaged to give the prediction error of the complete data set for a particular parameter value. For the case of small data sets as in the RBF-FD method, a particularly useful cross-validation technique for estimating the error of the function approximation is the leave-one-out method. In this method, the function approximation/interpolant is constructed by leaving out one data point and the left-out point is used as the validation point. An N element error vector can be obtained by repeating this N times with each data point as the validation point. The prediction error for a particular shape parameter value can then be calculated by averaging the N error terms.

A brief mathematical derivation for the LOO error predictor function or the cost functional $Q(\boldsymbol{x}, \sigma)$ for the RBF-FD method, based on Rippa (1999) is presented. The cost functional

should be able to imitate the behaviour of the error between the RBF-FD interpolant and the actual function derivative with respect to the shape parameter value. The equation for the cost functional for any node $\boldsymbol{x}_i$ with N supporting points is given by

$$Q(\boldsymbol{x}_i, \sigma) = \|\boldsymbol{E}^i\|_2, \tag{5.19}$$

with each element in the vector $\boldsymbol{E}^i$ defined as

$$E^i_k = f_k - S^{(k)}(\boldsymbol{x}_k, \sigma), \quad k = 1, 2, \cdots, N, \tag{5.20}$$

where $S^{(k)}(\boldsymbol{x}_k)$ is the interpolant of the function derivative obtained without using the supporting node $\boldsymbol{x}_k$ as a RBF centre i.e.,

$$S^{(k)}(\boldsymbol{x}_k, \sigma) = \sum_{j=1, j\neq k}^{N} \lambda^{(k)} \phi(\|\boldsymbol{x}_k - \boldsymbol{x}_j\|,\ \sigma). \tag{5.21}$$

The *learning set* for a particular $\boldsymbol{E}^i_k$ can then be defined as all data values other than f_k, which is the validation point in the leave-one-out form of cross-validation. Note that in RBF-FD, if $\mathcal{L}$ is the operator for which the RBF-FD weights need to be found at the node $\boldsymbol{x}_i$, the data vector is given by

$$f_k = \mathcal{L}\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_k\|,\ \sigma), \qquad k = 1, 2, \cdots, N. \tag{5.22}$$

It can be observed that at any node $\boldsymbol{x}_i$, for a particular value of the shape parameter, a direct evaluation of Equation (5.19) requires solving an $(N-1) \times (N-1)$ system of linear equations N times, and evaluation of $S^{(k)}(\boldsymbol{x}_k, \sigma)$ for $k = 1, 2, \cdots, N$. This method can become computationally expensive even for a moderate number of nodes in the stencil. Fortunately, after some matrix manipulations, the elements of $\boldsymbol{E}^i$ can be efficiently computed as

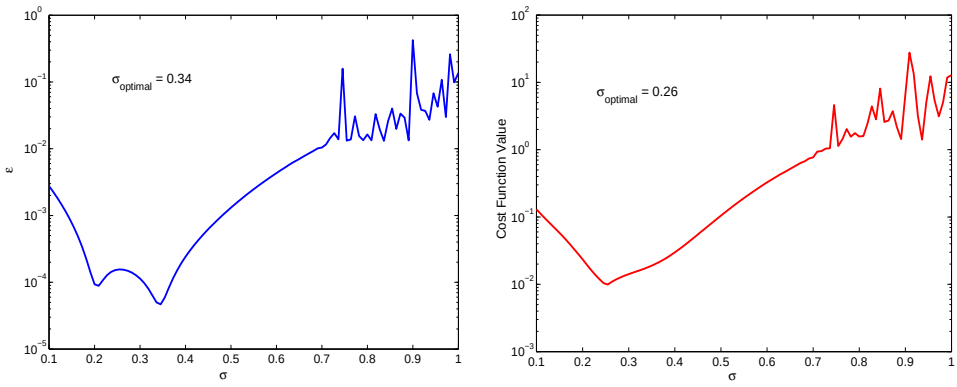
$$E^i_k = \frac{\lambda_k}{m^{(k)}_k}, \tag{5.23}$$

where λ_k is the k^{th} element of the RBF-FD weight vector,

$$\boldsymbol{\lambda} = \boldsymbol{A}^{-1}\boldsymbol{f},$$

and $m^{(k)}_k$ is the (k,k) element of the inverse of the Gram matrix $\boldsymbol{A}$. The complete derivation is shown in Appendix B. The computational cost of estimating $\boldsymbol{E}^i$ is that of performing a LU decomposition of the Gram matrix $\boldsymbol{A}$ and then the cost of N solutions of the linear system,

$$\boldsymbol{A}\boldsymbol{m}^{(k)} = \boldsymbol{e}^{(k)}, \qquad k = 1, 2, \cdots, N \tag{5.24}$$



True Error Behaviour Cost Function Behaviour

FIGURE 5.6: Behaviour comparison of true error and cost function using Leave-One-Out criterion for 1D interpolation problem

where $\boldsymbol{e}^{(k)}$ is the k^{th} column of the $N \times N$ identity matrix. Since the LU factorisation of $\boldsymbol{A}$ is known, the computational cost of solving Equation (5.24) is significantly less than that of the direct evaluation of the cost function. Other computationally efficient algorithms like estimating the cost function when the singular value decomposition of the matrix $\boldsymbol{A}$ is given or using the QR decomposition can also be pursued.

Figure 5.6 shows the behaviour of the true error and the cost functional value for a simple 1D interpolation problem as a function of the shape parameter. The function (Franke, 1982) is given by

$$f(x) = \frac{3}{4}\left(\exp\left(-\frac{(9x-2)^2}{4}\right) + \exp\left(-\frac{(9x+1)^2}{49}\right)\right) + \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4}\right) - \frac{1}{5}\exp\left(-(9x-4)^2\right). \tag{5.25}$$

Ten uniformly spaced data points in $[0, \frac{\pi}{2}]$ were considered and the resultant RBF interpolate was evaluated at 100 uniformly spaced points. The true error is evaluated as

$$\varepsilon = \|f_{\text{exact}} - f_{\text{predicted}}\|_2.$$

The values of the shape parameter for which the minimum of the true error and cost functional are also displayed. From the figure, it can be seen that the cost functional $Q(\boldsymbol{x}, \sigma)$ approximates the behaviour of the true error quite accurately.

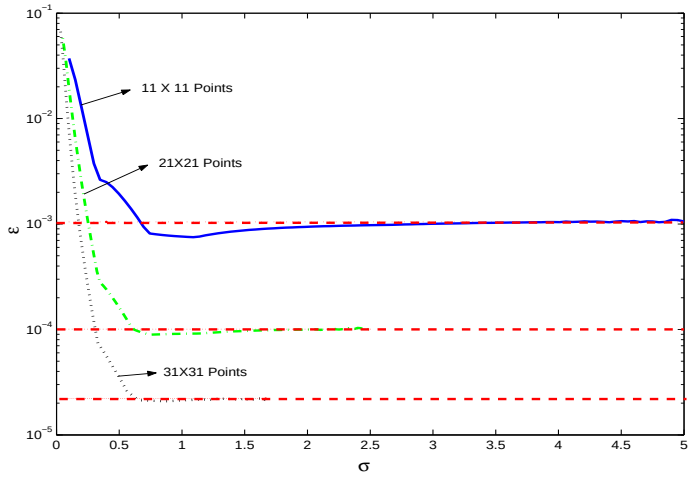


FIGURE 5.7: Accuracy of LOO optimised shape parameter for Poisson equation

5.4.2 Optimisation of LOO

To obtain the optimal value of the shape parameter, the cost functional evaluation procedure must be coupled with an optimisation routine to determine the optimal value of σ through iteration. The simplest way of obtaining the minima of the cost function $Q(\boldsymbol{x}_i, \sigma)$ is to use a grid search method. A shape parameter range is selected and is then divided uniformly and $Q(\boldsymbol{x}_i, \sigma)$ is estimated at each of the divided points. The minima of the cost function is then obtained and the corresponding value of the shape parameter is its optimal value. Another way is to use optimisation routines like the Brent's method or the Nelder search method. In this thesis, we use the Nelder search algorithm provided in MATLAB for optimising the shape parameter value. Note that the computational cost incurred by the optimisation routine for the RBF-FD method is much less than the strategy presented earlier in Chapter 3 for the RBF collocation method.

5.4.3 Numerical studies

Figure 5.7 presents the accuracy of the proposed shape parameter optimisation strategy. The Poisson equation is solved with the source term given in Equation (5.17). We consider a uniform node distribution of 11×11, 21×21 and 31×31. The behaviour of the accuracy of the RBF-FD method (measured in L_2 norm) with respect to shape parameter for each of the node set is shown in the figure, and the corresponding accuracy obtained by optimising the shape parameter is shown by dotted lines. From Figure 5.7, it can be observed that the proposed strategy indeed obtains a very good approximation of the optimal value of

the shape parameter for a little additional cost during the pre-processing stage when the RBF-FD weights are computed.

5.5 RBF-FD for the unsteady convection-diffusion equation

In this section, we present the RBF-FD formulation for the 2D unsteady convection diffusion equation presented in the earlier chapter. The governing PDE is given by

$$\frac{\partial u}{\partial t} = \kappa_x \frac{\partial^2 u}{\partial x^2} + \kappa_y \frac{\partial^2 u}{\partial y^2} + V_x \frac{\partial u}{\partial x} + V_y \frac{\partial u}{\partial y},\ 0 \leq x, y \leq 1,\ t > 0. \tag{5.26}$$

In the RBF-FD method, we begin with representing the complete domain with a set of scattered nodes present in the interior and on the boundary. For each interior node, a supporting region/stencil is identified by choosing N nearest nodes. Then at each node, a local RBF interpolation problem is set up to determine the RBF-FD weights for each derivative (see section 5.1). This completes the pre-processing stage.

Once the RBF-FD method is applied to discretise the spatial derivatives in the governing equation, Equation (5.26), we obtain at any interior node $\boldsymbol{x}_i$,

$$\frac{du_i}{dt} = \kappa_x \sum_{j=1}^{N} w_{(i,j)}^{(xx)} u_j + \kappa_y \sum_{j=1}^{N} w_{(i,j)}^{(yy)} u_j + V_x \sum_{j=1}^{N} w_{(i,j)}^{(x)} u_j + V_y \sum_{j=1}^{N} w_{(i,j)}^{(y)} u_j, \tag{5.27}$$

where N is the total number of interior and boundary nodes which lie in the supporting region/stencil for the node $\boldsymbol{x}_i$, and $w_{(i,j)}^{(x)}$, $w_{(i,j)}^{(y)}$, $w_{(i,j)}^{(xx)}$, $w_{(i,j)}^{(yy)}$ are the RBF-FD weights obtained from the system of Equations (5.9) with the corresponding differential operator $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2})$ applied to the basis functions on the right hand side.

Discretising Equation (5.27) using forward difference in time and θ-weighting scheme; and denoting the value of any physical quantity at $t = t^n$ with the superscript n, we obtain

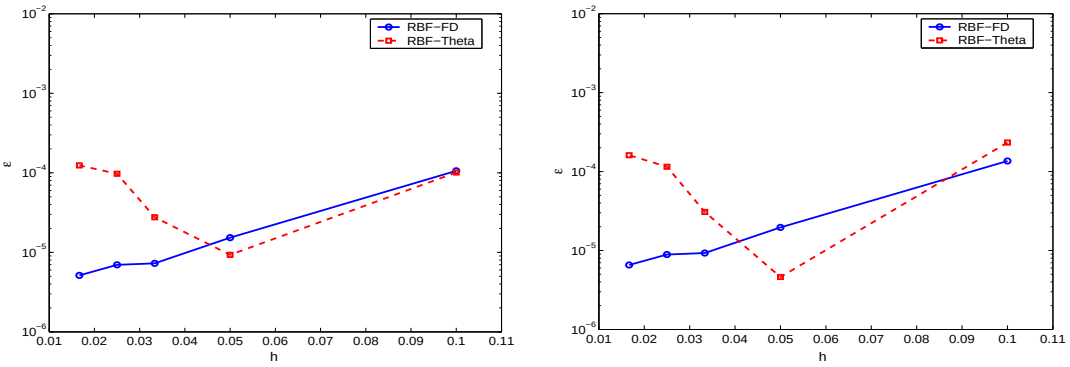
$$\begin{aligned} \frac{u_i^{n+1} - u_i^n}{\delta t} &= \theta \left[\sum_{j=1}^{N} \left(\kappa_x w_{(i,j)}^{(xx)} + \kappa_y w_{(i,j)}^{(yy)} \right) u_j^{n+1} + \sum_{j=1}^{N} \left(V_x w_{(i,j)}^{(x)} + V_y w_{(i,j)}^{(y)} \right) u_j^{n+1} \right] \\ &\quad + (1-\theta) \left[\sum_{j=1}^{N} \left(\kappa_x w_{(i,j)}^{(xx)} + \kappa_y w_{(i,j)}^{(yy)} \right) u_j^{n} + \sum_{j=1}^{N} \left(V_x w_{(i,j)}^{(x)} + V_y w_{(i,j)}^{(y)} \right) u_j^{n} \right]. \end{aligned} \tag{5.28}$$

Equation (5.28) written for each interior node leads to a system of equations which can be solved to obtain the solution at $t = t^{n+1}$.

5.5.1 Numerical results

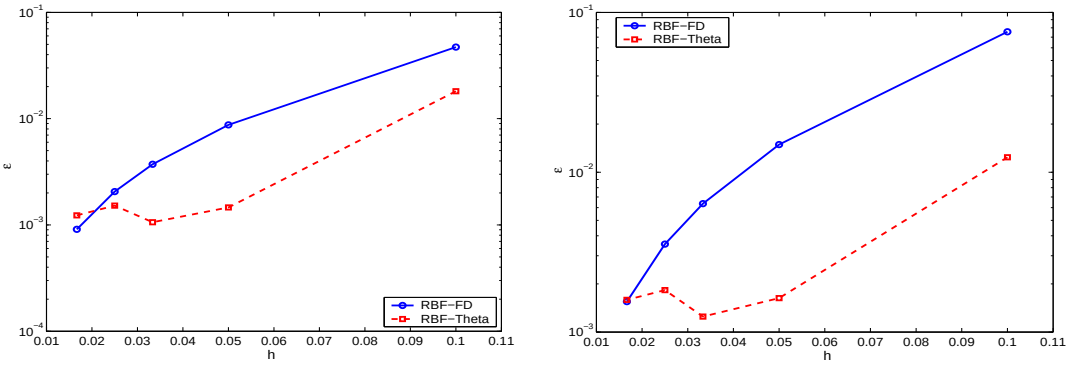
In this subsection, we present a comparison between the RBF-Theta method presented in Chapter 3 and the RBF-FD method for the unsteady convection diffusion equation. The initial and boundary conditions are given in Equations (3.38) and (3.37). The exact solution for this problem is given in Equation (3.39). We consider a uniform distribution of points ranging from $h = 0.1$ to $h = 0.01$, where h is the mesh spacing. The multiquadric RBF is used for numerical studies and for the RBF-FD method, the stencil B shown in Figure 5.2 is used. The shape parameter for the RBF-Theta method is obtained by the optimisation strategy outlined in section 3.3 and for the RBF-FD method, the optimal value of the shape parameter is obtained using the leave-one-out criterion proposed in this chapter (see section 5.4).

Figure 5.8 presents the convergence behaviour for both the methods when the Peclet number is 1.0. The results are compared at final times $t_f = 0.1$ and $t_f = 1.0$ in the left and the right sub-figures. From the plots, it can be seen that the RBF-FD method converges in a linear fashion as compared to the RBF-Theta method, i.e., better accuracy is obtained as we decrease the mesh spacing. However, in the case of RBF-Theta method, for a particular mesh spacing provided we use the optimal value of the shape parameter, the accuracy is better than the RBF-FD method. Similar behaviours are observed for Peclet number 10.0 (see, Figure 5.9) and 100.0 (see, Figure 5.10).



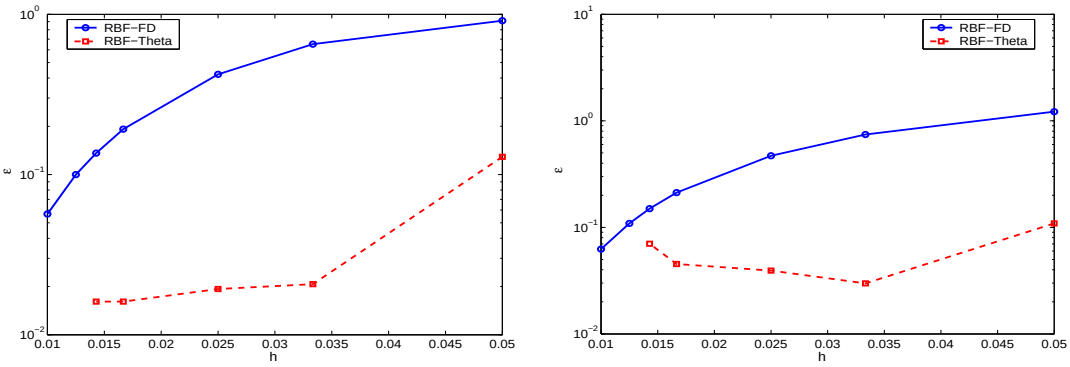
$t_f = 0.1$ $t_f = 1.0$

FIGURE 5.8: Comparison of RBF-FD vs RBF-Theta methods for the unsteady convection diffusion equation, $P_e = 1.0$



$t_f = 0.1$ $t_f = 1.0$

FIGURE 5.9: Comparison of RBF-FD vs RBF-Theta methods for the unsteady convection diffusion equation, $P_e = 10.0$



$t_f = 0.1$ $t_f = 1.0$

FIGURE 5.10: Comparison of RBF-FD vs RBF-Theta methods for the unsteady convection diffusion equation, $P_e = 100.0$

5.6 Conclusions

A methodology using RBFs in a finite difference mode (RBF-FD) is presented for solving time-dependent PDEs. This method approximates the function derivatives at a node in terms of the function values on a scattered set of points present in support region of the node. The RBF-FD method uses local interpolation problems and hence generates sparse and well-conditioned matrices. It also has the property of decreased sensitivity with respect to the shape parameter value in comparison with the RBF collocation method. A shape parameter tuning strategy based on the statistical leave-one-out criterion is proposed for choosing an optimal value of the shape parameter for a RBF-FD stencil. Numerical studies conducted on a Poisson equation and the unsteady convection-diffusion equation show that the RBF-FD

method has the property of monotonic increase in the accuracy of the numerical solution as number of nodes in the domain are increased. This property is not guaranteed for the RBF-Theta collocation method. In addition, due to small RBF coefficient matrices generated at each node, the shape parameter optimisation is computationally very efficient compared to the optimisation strategy proposed for the RBF-Theta collocation method earlier in Chapter 3. The remaining chapters of this thesis are concerned with developing RBF collocation and RBF-FD schemes for the incompressible Navier Stokes equations.

Chapter 6

RBF Collocation Scheme for the Incompressible Navier-Stokes Equations

In this chapter, we give a brief overview of the equations governing incompressible fluid flows and propose an RBF collocation method for solving them. Numerical studies conducted in Chapters 3 and 4 show that the RBF collocation method gives accurate results for the model convection-diffusion equation (at moderate Peclet numbers) which is a precursor to the Navier-Stokes equations. These studies also suggest that the multiquadric RBF produces more accurate solutions provided an appropriate value of the shape parameter is used. We therefore use this RBF in studies on the Navier-Stokes equations.

RBF collocation methods have been applied earlier to solve fluid flow problems in the literature. Young *et al.* (2004) solved the Stokes's equations (Re = 0) using the RBF collocation method. An alternative approach known as the Integrated Radial Basis Function Networks (IRBFN) developed by Mai-Duy & Tran-Cong (2001) was also utilised for solving the incompressible Navier Stokes equations. These are the only occurrences of using RBF collocation for solving the Navier-Stokes equations to the best of our knowledge. In this chapter, we present an improved version of the RBF collocation method for solving incompressible viscous flows. A novel ghost centre strategy is employed to satisfy the boundary conditions. The issue of shape parameter and its influence on the accuracy of the computed solution is discussed. Numerical results are presented for example problems like square and rectangular driven cavity flows and flow over a backward facing step.

6.1 Incompressible Navier-Stokes equations

Incompressible flows can be defined as those flows for which the density is constant on all particle paths. The governing equations represent mathematical statements of the conservation laws of physics, i.e.,

1. Fluid mass is conserved.
2. The rate of change of momentum equals the sum of forces on a fluid particle.
3. The rate of change of energy is equal to the sum of the rate of heat addition and the rate of work done on a particle.

These statements are transformed into mathematical equations as

- Conservation of Mass:
$$\frac{D\rho}{Dt} + \rho\nabla \cdot \boldsymbol{u} = 0, \tag{6.1}$$
where ρ is the density of the fluid, $\boldsymbol{u}$ is the velocity vector and $\frac{D}{Dt}$ denotes the material derivative ($\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{u}\cdot\nabla$). As the density is constant for incompressible flows Equation (6.1) reduces to
$$\nabla \cdot \boldsymbol{u} = 0. \tag{6.2}$$

- Conservation of Momentum:
$$\rho\frac{D\boldsymbol{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho\boldsymbol{f}, \tag{6.3}$$
where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{f}$ is the body force vector measured per unit mass.

- Conservation of Energy:
$$\rho C_v\frac{DT}{Dt} = -\nabla \cdot \boldsymbol{q} + \boldsymbol{Q} + \Xi, \tag{6.4}$$
where T is the temperature, $\boldsymbol{q}$ is the heat flux vector, $\boldsymbol{Q}$ is the internal heat generation (measured per unit volume), Ξ is the viscous dissipation function and C_v is the specific heat at constant volume.

The constitutive relation expressing the Cauchy stress in terms of strain for Newtonian fluids is given by
$$\boldsymbol{\sigma}_{ij} = -p\boldsymbol{\delta}_{ij} + 2\mu\dot{\boldsymbol{u}}_{i,j}, \tag{6.5}$$

where μ is the viscosity coefficient and $\boldsymbol{\delta}_{ij}$ is the Kronecker delta tensor. For a complete derivation of Navier-Stokes equations, the reader is referred to any standard textbook on fluid mechanics (Batchelor, 1967; Ferziger & Peric, 1999).

The incompressible Navier-Stokes equations are a system of nonlinear second order equations. The mathematical classification of these equations can be done and it turns out that for steady flow problems, the equations are elliptic in nature and for unsteady flow problems, they are of mixed type in nature. This classification is important when boundary conditions are specified. In this thesis, we focus on the numerical solution of incompressible viscous flows, which are governed by the steady/unsteady NS equations.

The NS equations and the continuity equation (conservation of mass) for a two-dimensional incompressible Newtonian fluid flow in Cartesian coordinate system are given by

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \\ \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \end{aligned} \tag{6.6}$$

where (u, v) denote the Cartesian components of the velocity vector, ν is the kinematic viscosity and p denotes the pressure field. In these equations, the body force terms are neglected.

The velocity components can be alternatively defined in terms of streamfunction (ψ) as

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}.$$

Equation (6.6) can be simplified by introducing two new variables: the streamfunction (ψ) and the vorticity (ω) defined by

$$\omega = \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right). \tag{6.7}$$

The governing equations then become

$$\begin{aligned} \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \omega &= 0, \\ \frac{\partial \omega}{\partial t} + \left(\frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y}\right) &= \nu\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right). \end{aligned} \tag{6.8}$$

Let L be a characteristic length and U a characteristic speed of the flow, then variables can be non-dimensionalised as follows:

$$\bar{x} = \frac{x}{L},\ \bar{y} = \frac{y}{L},\ \bar{t} = \frac{t}{L/U},\ \bar{u} = \frac{u}{U},\ \bar{v} = \frac{v}{U},\ \bar{\psi} = \frac{\psi}{UL},\ \bar{\omega} = \frac{\omega}{U/L}. \tag{6.9}$$

Equation (6.8) then becomes

$$\begin{aligned} \frac{\partial^2 \bar{\psi}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{\psi}}{\partial \bar{y}^2} + \bar{\omega} &= 0, \\ \frac{\partial \bar{\omega}}{\partial \bar{t}} + \left(\frac{\partial \bar{\psi}}{\partial \bar{y}} \frac{\partial \bar{\omega}}{\partial \bar{x}} - \frac{\partial \bar{\psi}}{\partial \bar{x}} \frac{\partial \bar{\omega}}{\partial \bar{y}} \right) &= \frac{1}{\mathrm{Re}} \left(\frac{\partial^2 \bar{\omega}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{\omega}}{\partial \bar{y}^2} \right) \end{aligned} \tag{6.10}$$

where $\mathrm{Re} = \frac{UL}{\nu}$ is known as the Reynolds number.

For steady state flows, dropping the temporal terms the governing equations (with the bars dropped for the sake of brevity) become

$$\nabla^2 \omega = \mathrm{Re} \left(\frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x} \frac{\partial \omega}{\partial y} \right), \tag{6.11}$$

$$\omega = -\nabla^2 \psi, \tag{6.12}$$

where ∇^2 is the Laplacian operator.

The streamfunction-vorticity formulation has been successfully used by a number of researchers over the past several decades to test new methods for the numerical solution of fluid flow problems. However, there is an uncertainty in the numerical treatment of the vorticity values particularly over a no-slip boundary. The vorticity ω is defined through the Poisson equation, Equation (6.12) , which needs to be solved discretely on the boundaries so that boundary vorticity values can be specified for solving the vorticity transport equation, Equation (6.11). A variety of numerical approximations have been carried out for specifying the vorticity values on the boundary when finite difference schemes are employed to solve the governing equations. For more details on these schemes, see, Thom (1928); Jensen (1959); Woods (1954); Spotz (1995). This difficulty of imposing appropriate vorticity boundary conditions can be circumvented by substituting Equation (6.12) into Equation (6.11) thereby obtaining a nonlinear biharmonic equation in terms of the streamfunction alone, i.e.,

$$\frac{\partial^4 \psi}{\partial x^4} + 2\frac{\partial^4 \psi}{\partial x^2 \partial y^2} + \frac{\partial^4 \psi}{\partial y^4} - \mathrm{Re} \left[u \left(\frac{\partial^3 \psi}{\partial x^3} + \frac{\partial^3 \psi}{\partial x \partial y^2} \right) + v \left(\frac{\partial^3 \psi}{\partial y^3} + \frac{\partial^3 \psi}{\partial x^2 \partial y} \right) \right] = 0. \tag{6.13}$$

- For unsteady flows, in the same way, it can be shown that Equation (6.10) becomes

$$-\mathrm{Re} \frac{\partial}{\partial t} (\nabla^2 \psi) + \frac{\partial^4 \psi}{\partial x^4} + 2\frac{\partial^4 \psi}{\partial x^2 \partial y^2} + \frac{\partial^4 \psi}{\partial y^4} - \mathrm{Re} \left[u \left(\frac{\partial^3 \psi}{\partial x^3} + \frac{\partial^3 \psi}{\partial x \partial y^2} \right) + v \left(\frac{\partial^3 \psi}{\partial y^3} + \frac{\partial^3 \psi}{\partial x^2 \partial y} \right) \right] = 0. \tag{6.14}$$

In the context of traditional finite differences, the numerical solution of Equation (6.13) or Equation (6.14) is cumbersome since a higher order stencil is typically required to solve the

above equations satisfactorily. However, as we show next, in RBF collocation methods, the complexity of the problem does not change as one tries to solve Equation (6.13)/Equation (6.14).

6.2 Formulation

In this section, we present an improved version of the RBF collocation method and its application to steady incompressible viscous flows.

The nonlinear PDE in Equation (6.13) is solved using the RBF collocation method and the Newton iteration technique. In Newton iteration, one starts with an initial solution $\psi^{(0)}$ and then obtains a sequence $\{\psi^{(i)}\}$ of approximate solutions using the Newton iteration formula, $\psi^{(i)} = \psi^{(i-1)} + \xi$. At each Newton iteration i, the following linear PDE needs to be solved for the correction ξ:

$$\nabla^2\nabla^2\xi - \mathrm{Re}\left[\left(\frac{\partial\psi^{(i-1)}}{\partial y}\frac{\partial}{\partial x}(\nabla^2\xi) + \frac{\partial}{\partial x}(\nabla^2\psi^{(i-1)})\frac{\partial\xi}{\partial y}\right)\right.$$
$$\left. - \left(\frac{\partial\psi^{(i-1)}}{\partial x}\frac{\partial}{\partial y}(\nabla^2\xi) + \frac{\partial}{\partial y}(\nabla^2\psi^{(i-1)})\frac{\partial\xi}{\partial x}\right)\right] = -R(\psi^{(i-1)}) \tag{6.15}$$

where $R(\psi^{(i-1)})$ is the residual of the governing equation at iteration i which is given by

$$R(\psi^{(i-1)}) = \frac{\partial^4\psi^{(i-1)}}{\partial x^4} + 2\frac{\partial^4\psi^{(i-1)}}{\partial x^2\partial y^2} + \frac{\partial^4\psi^{(i-1)}}{\partial y^4} - \mathrm{Re}\left[\frac{\partial\psi^{(i-1)}}{\partial y}\right.$$
$$\left. \times \left(\frac{\partial^3\psi^{(i-1)}}{\partial x^3} + \frac{\partial^3\psi^{(i-1)}}{\partial x\partial y^2}\right) - \frac{\partial\psi^{(i-1)}}{\partial x}\left(\frac{\partial^3\psi^{(i-1)}}{\partial y^3} + \frac{\partial^3\psi^{(i-1)}}{\partial x^2\partial y}\right)\right]. \tag{6.16}$$

The boundary condition that needs to be satisfied at iteration i is given by

$$\mathcal{B}\xi = g(x,y) - \mathcal{B}\psi^{(i-1)}. \tag{6.17}$$

where $\mathcal{B}$ can be Dirichlet, Neumann or a mixed differential operator.

In the present formulation, the unknown streamfunction is expanded in terms of a linear combination of RBFs centred at randomly spaced points in the domain i.e.,

$$\psi(\boldsymbol{x}) = \sum_{j=1}^{N_I+N_B} \alpha_j\phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|). \tag{6.18}$$

where N_I and N_B are the number of RBF centres placed in the interior and on the boundary, respectively. The undetermined weights, $\{\alpha_j\}_{j=1}^{(N_I+N_B)}$, are which are calculated by collocation. For simplicity, the set of RBF centres is assumed to coincide with the set of collocation

points. The governing PDE is satisfied on each of the interior collocation points and the boundary conditions on the boundary collocation points. By virtue of representation of the unknown function as in Equation (6.18), the derivatives of the unknown functions are nothing but a linear combination of the corresponding derivatives of the basis functions. Thus, for example, if $\mathcal{L}$ is any linear partial differential operator, one obtains

$$\mathcal{L}\psi(\boldsymbol{x}) = \sum_{j=1}^{N_I+N_B} \alpha_j \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) \tag{6.19}$$

As an initial approximation to the unknown function is provided in case of the Newton iteration, an interpolation problem is solved to obtain the RBF coefficients ($\alpha_j^{(0)}$). Thus, the initial approximation to streamfunction satisfies

$$\psi^{(0)}(\boldsymbol{x}) = \sum_{j=1}^{N_I+N_B} \alpha_j^{(0)} \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|). \tag{6.20}$$

The function approximation at any Newton iteration i is obtained by solving the linear PDE in Equation (6.15). Hence at iteration i, the streamfunction is given by

$$\psi^{(i)}(\boldsymbol{x}) = \sum_{j=1}^{N_I+N_B} (\alpha_j^{(i-1)} + \delta_j)\phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|), \tag{6.21}$$

where $\boldsymbol{\delta}$ is obtained by solving the following linear system of equations:

$$\begin{aligned}\sum_{j=1}^{N_I+N_B} \delta_j \Bigg[\nabla^2\nabla^2\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) - \left(a(\boldsymbol{x}_i)\frac{\partial}{\partial x}(\nabla^2\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)) + b(\boldsymbol{x}_i)\frac{\partial}{\partial y}(\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|))\right) \\ + \left(c(\boldsymbol{x}_i)\frac{\partial}{\partial y}(\nabla^2\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) + d(\boldsymbol{x}_i)\frac{\partial}{\partial x}(\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|))\right)\Bigg] = R\left(\psi^{(i-1)}(\boldsymbol{x}_i)\right) \\ i = 1, 2, \cdots, N_I,\end{aligned} \tag{6.22}$$

and

$$\begin{aligned}\sum_{j=1}^{N_I+N_B} \delta_j \left[\mathcal{B}\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)\right] = g(\boldsymbol{x}_i) - \mathcal{B}\psi^{(i-1)}(\boldsymbol{x}_i) \\ i = N_I + 1, N_I + 2, \cdots, N_I + N_B.\end{aligned} \tag{6.23}$$

In Equation (6.22), $a(\boldsymbol{x})$, $b(\boldsymbol{x})$, $c(\boldsymbol{x})$ and $d(\boldsymbol{x})$ are all functions of the previous iteration estimate $\psi^{(i-1)}$ and are given by

$$\begin{aligned} a(\boldsymbol{x}) &= \mathrm{Re}\frac{\partial\psi^{(i-1)}}{\partial y}, & b(\boldsymbol{x}) &= \mathrm{Re}\frac{\partial}{\partial x}(\nabla^2\psi^{(i-1)}), \\ c(\boldsymbol{x}) &= \mathrm{Re}\frac{\partial\psi^{(i-1)}}{\partial x} \quad \text{and} & d(\boldsymbol{x}) &= \mathrm{Re}\frac{\partial}{\partial y}(\nabla^2\psi^{(i-1)}). \end{aligned} \tag{6.24}$$

The Newton iterations are continued until

$$\| R\left(\psi^{(i)}(\boldsymbol{x})\right) \|_\infty \leq \varepsilon, \tag{6.25}$$

where ε is a pre-determined convergence limit.

In the case of unsteady flows, a nonlinear system of equations is obtained at each time step which can be solved using the Newton iteration method as shown above.

6.3 Incorporation of boundary conditions

In the case of incompressible viscous flows, as two boundary conditions are prescribed at each boundary collocation point, we obtain an overdetermined system of equations for the RBF weights which can then be solved in a least squares sense.

In this section, we present an alternative strategy based on ghost centres for incorporating the boundary conditions. For example, the no-slip boundary conditions over any boundary Γ are given by

$$\begin{aligned} \psi &= C_1 \qquad \boldsymbol{x} \in \Gamma \\ \tfrac{\partial \psi}{\partial n} &= C_2 \qquad \boldsymbol{x} \in \Gamma \end{aligned} \tag{6.26}$$

where C_1 and C_2 are constants and $\vec{n}$ is the outward normal direction from the boundary.

It can be seen from Equation (6.26) that we have a pair of boundary conditions for the streamfunction. Hence, it is proposed to express the streamfunction in terms of RBFs chosen over the domain and also a set of ghost centres chosen outside the domain equal to the number of boundary centres. The RBF approximation for the streamfunction can be rewritten as

$$\psi(\boldsymbol{x}) = \sum_{j=1}^{N+G} \alpha_j \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|), \tag{6.27}$$

where N is the total number of collocation points (including interior and boundary) and G is the number of ghost centres, respectively. A schematic diagram of the RBF centre distribution along with the ghost centres is shown in Figure 6.1.

By using the ghost centres, a square system of equations is obtained which can be solved for the RBF weights. In order to compare the accuracy of the proposed ghost centres strategy with the least squares approach, we solve a model biharmonic equation of the form

$$\frac{\partial^4 \psi}{\partial x^4} + 2\frac{\partial^4 \psi}{\partial^2 x \partial^2 y} + \frac{\partial^4 \psi}{\partial y^4} = F(x, y) \tag{6.28}$$

with boundary conditions in terms of ψ and $\frac{\partial \psi}{\partial n}$.

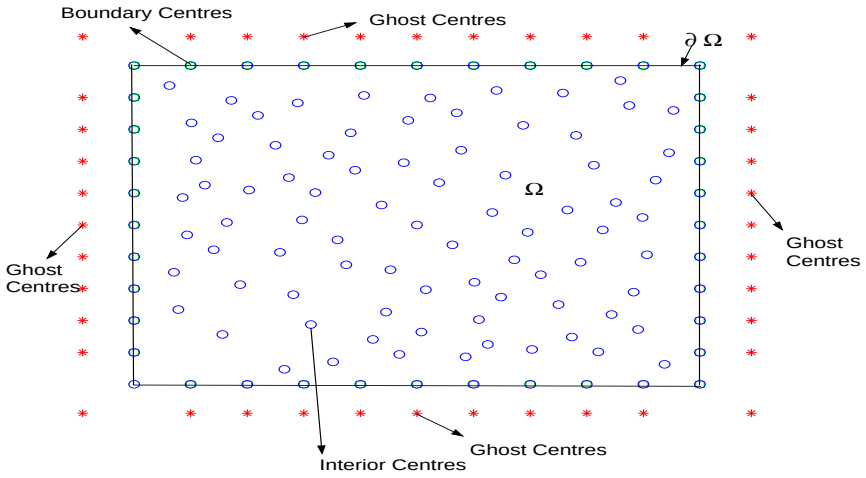


FIGURE 6.1: A Schematic representation of RBF centres in the domain

Consider the test problem (Mai-Duy & Tanner, 2005), where

$$F(x,y) = 16\pi^4[4\cos(2\pi x)\cos(2\pi y) - \cos(2\pi x) - \cos(2\pi y)], \tag{6.29}$$

and the exact solution is given by

$$\psi_{\text{exact}} = 4\sin^2(\pi x)\sin^2(\pi y). \tag{6.30}$$

Figure 6.2 presents the accuracy and convergence behaviour of RBF collocation method with least squares and ghost centres strategy. The error ε is defined as the L_∞ norm of the difference between the exact and computed solutions. From the figure, it can be seen that the ghost centres strategy gives significantly better accuracy.

6.4 Implementation aspects

In this section, we discuss some of the implementation aspects of the presented meshless method.

Many of the RBFs incorporate a user-defined shape parameter. This scalar parameter determines the region of influence of the RBF. Numerical studies on RBF collocation methods have shown that the multiquadric RBF gives better performance as compared to other RBFs (Larsson & Fornberg (2003); see also Chapter 3). In this study, we employ the multiquadric RBF to test the accuracy of the present formulation. It has been observed that in the case of RBF collocation, the accuracy of the numerical solution depends heavily on the value of the shape parameter. However, obtaining the optimal value of the shape parameter remains

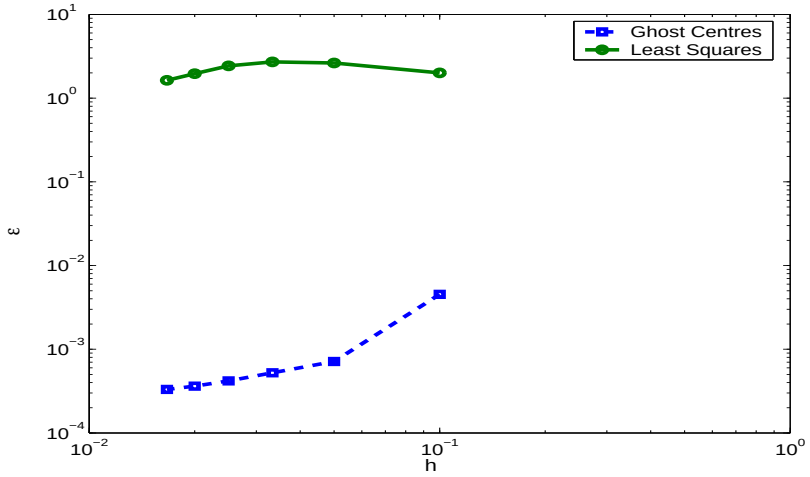


FIGURE 6.2: Convergence plot obtained for least squares and ghost centres strategy for RBF collocation method ($\sigma = 3d_{\min}$)

elusive and is still an open problem in the literature. The influence of the shape parameter on the accuracy of the obtained numerical solution generally follows an U-shaped curve with the accuracy becoming worse after a certain value of the shape parameter is exceeded due to ill-conditioning effects. In fact good accuracy is achieved only at the on-set of ill-conditioning (Schaback, 1995).

Various empirical estimates for the shape parameter have been proposed in the literature (Kansa, 1990b; Hardy, 1990). In this chapter, we estimate the shape parameter as

$$\sigma = \beta \bar{d}, \tag{6.31}$$

where β is a positive scalar and $\bar{d}$ is the minimum distance between any two centres in the domain. Numerical studies in Chapter 3 have shown that the residual error on a suitable fine grid is a good indicator of the accuracy of the solution. Hence one way of obtaining the optimal value of the shape parameter is as follows. We start with an initial value of 1 for β and its value is progressively increased in discrete steps while monitoring the value of L_2 norm of the residual error.

Note that since a global shape parameter is used, the radial basis function $\phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|)$ must incorporate scale parameters L_x and L_y in each direction, particularly for problems where the magnitudes of the distances between points in x and y directions are significantly different. For example when scale parameters are used, the expression for the multiquadric RBF becomes

$$\phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) = \sqrt{\left(\frac{x - x_j}{L_x}\right)^2 + \left(\frac{y - y_j}{L_y}\right)^2 + \sigma^2} \tag{6.32}$$

Also, it is well known that the Newton iteration technique does not guarantee convergence when the starting point is far from the actual solution and when the Jacobian matrix is ill-conditioned. Hence, one resorts to secant techniques or trust-region techniques when the initial guess is far from the solution (Coleman & Li, 1994, 1996). Details of trust region techniques can be obtained elsewhere (Coleman & Li, 1994, 1996). In the numerical studies presented here, we employ the trust region algorithm provided in MATLAB 6(release 13) package.

6.5 Numerical results

In this section, we present numerical results obtained by using the presented RBF collocation method for square driven cavity flow, a rectangular driven cavity with aspect ratio 2.0 and flow over a backward facing step.

6.5.1 Square driven cavity flow

In this section a two-dimensional lid-driven cavity problem is solved and the results are compared with those obtained by Ghia *et al.* (1982) who used a multigrid finite difference method with a mesh size of 129×129.

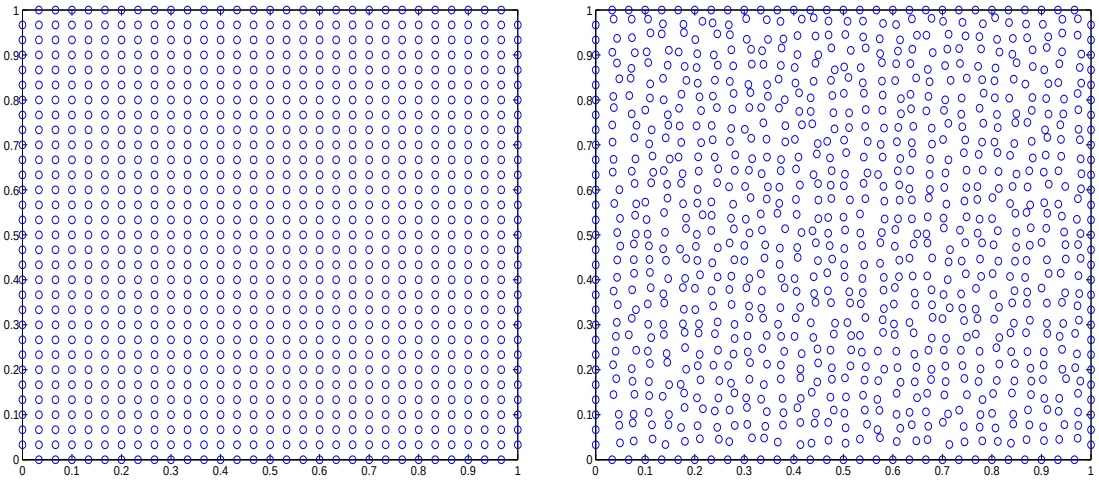
The boundary conditions for the driven cavity problem on a $[0 \times 1]^2$ domain are given by

$$\begin{aligned} &\psi = 0, \frac{\partial \psi}{\partial x} = 0 \quad \text{on} \quad x = 0 \text{ and } x = 1, \\ &\psi = 0, \frac{\partial \psi}{\partial y} = 0 \quad \text{on} \quad y = 0, \\ &\psi = 0, \frac{\partial \psi}{\partial y} = 1 \quad \text{on} \quad y = 1. \end{aligned} \tag{6.33}$$

We consider uniform collocation point sets ranging from 11×11 to 61×61. A random set of collocation points obtained by perturbing the uniform distribution set is also considered to show the accuracy of the presented method for randomly spaced points. The Reynolds numbers used in the present study are $\{0, 100, 400, 1000, 3200\}$. For each Reynolds number, the solution obtained by solving the previous Reynolds number in the set was taken as the initial guess for the trust-region algorithm. During the numerical experiments, we observed that the Newton algorithm converges in about 10-15 iterations. As compared to the large mesh used by Ghia *et al.* (1982), it is found that the RBF approximations give results of comparable accuracy even with a grid as small as 41×41 for moderate Reynolds numbers (see Figs 6.4-6.12). In fact for $Re < 400$, a coarse grid of 11×11 predicts the solution with good accuracy. The figures presented were those obtained using the multiquadric RBF. From

the figures, it can be seen that the primary and secondary vortices were captured accurately using the present method. For high Reynolds numbers, the viscous boundary layer near each wall is captured satisfactorily.

The presented formulation is checked first for the Stokes problem (Re=0). Uniform and randomly spaced distribution of points are considered. The point distributions for 31×31 are shown in Figure 6.3. Figure 6.4 and Figure 6.5 show the streamfunction and vorticity contours for 31×31 set of points. The value of $\bar{d}$ is taken to be 0.033.



a) Uniform distribution b) Random distribution

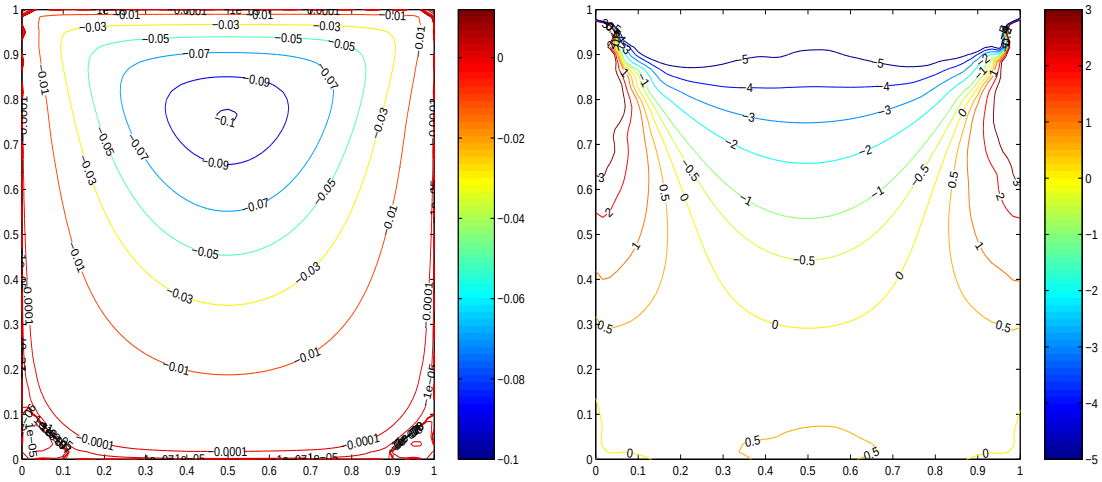
FIGURE 6.3: Point distributions for 31×31 points

We now present comparison studies for Reynolds number 100.0. Figure 6.6 shows the streamfunction and vorticity contours obtained for 31×31 uniformly distributed set of points. Figure 6.7 presents the comparison of the velocities obtained on the horizontal and vertical centre-lines of the cavity with those obtained by Ghia *et al.* (1982). From the figures, it can be seen that the RBF collocation method can accurately capture the solution.

The results obtained for Reynolds number 400 are presented in Figure 6.8, and comparison with Ghia's results is presented in Figure 6.9. Similar comparison studies are presented for higher Reynolds numbers in Figures 6.10-6.12. It can be observed that the present meshless formulation results agree very well with those of Ghia *et al.* (1982).

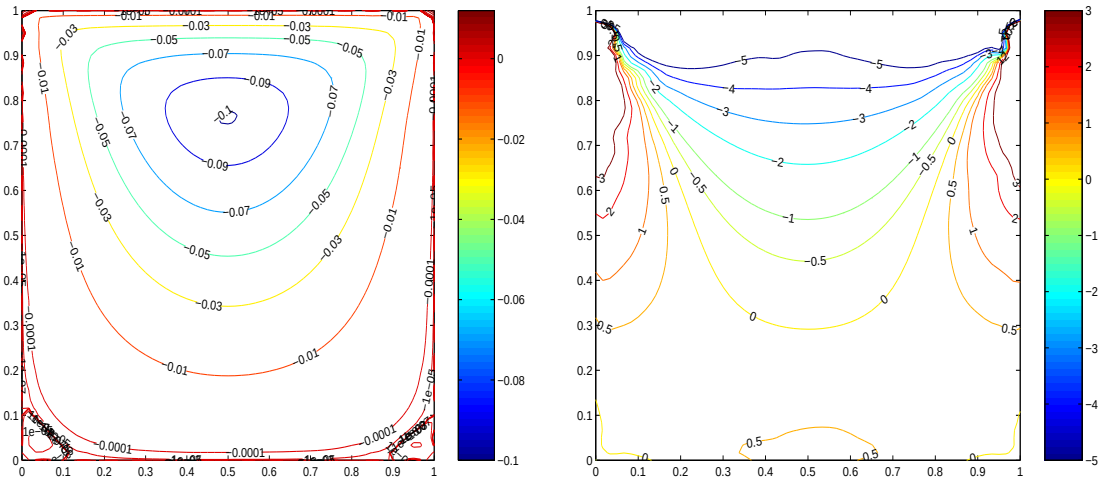
6.5.2 Rectangular driven cavity flow

We now consider the problem of a lid driven flow in a rectangular cavity with aspect ratio of 2. The problem is defined and solved in the rectangle $0 \leq x \leq 1$, $0 \leq y \leq 2$. This problem



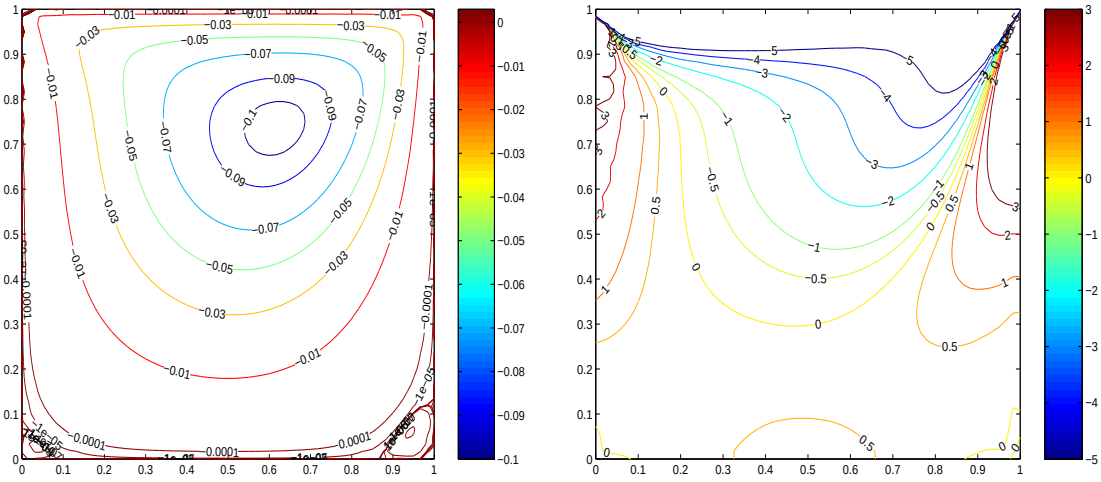
a) Streamfunction b) Vorticity

FIGURE 6.4: Square driven cavity: Re = 0, contours of streamfunction and vorticity obtained using 31×31 uniform point distribution



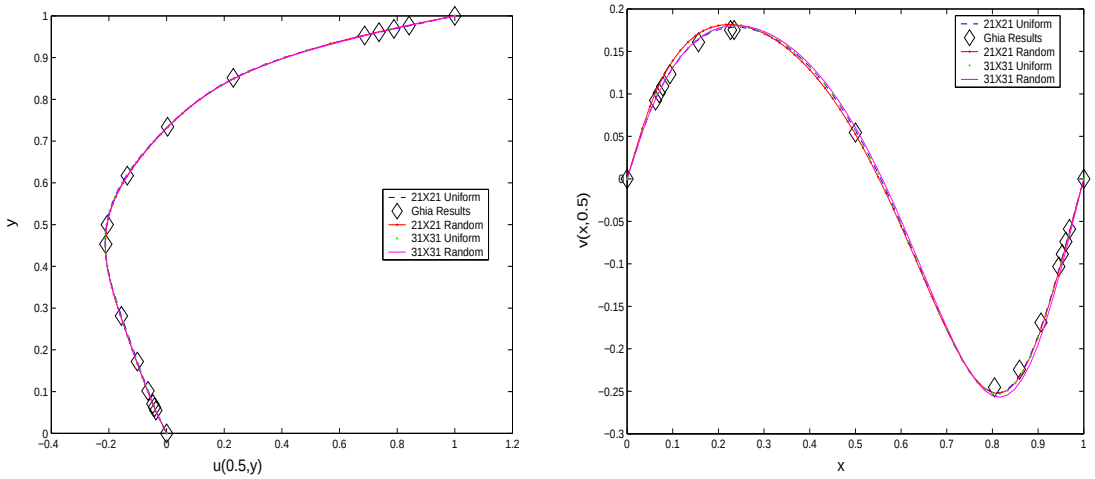
a) Streamfunction b) Vorticity

FIGURE 6.5: Square driven cavity: Re = 0, contours of streamfunction and vorticity obtained using 31×31 random point distribution



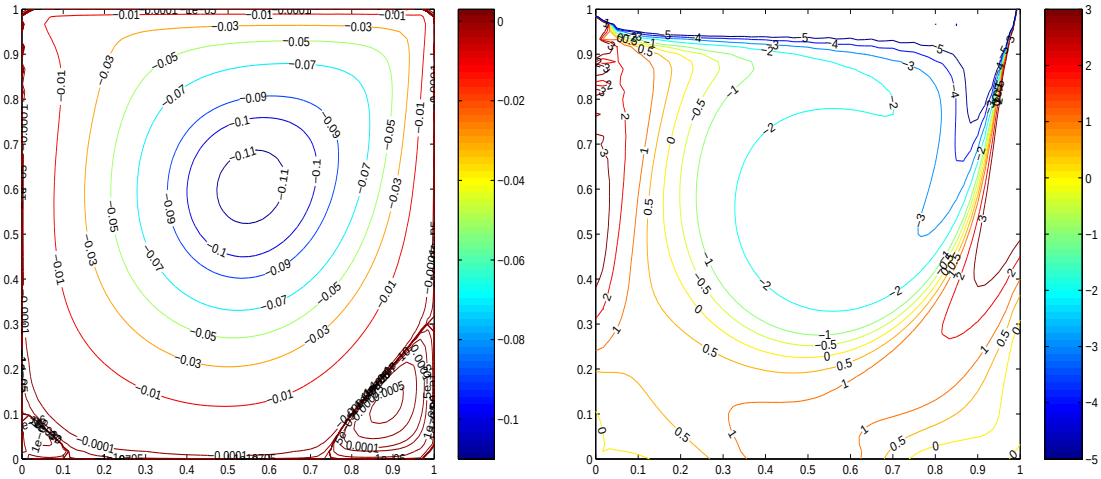
a) Streamfunction b) Vorticity

FIGURE 6.6: Square driven cavity: Re = 100, contours of streamfunction and vorticity obtained using 31×31 uniform point distribution



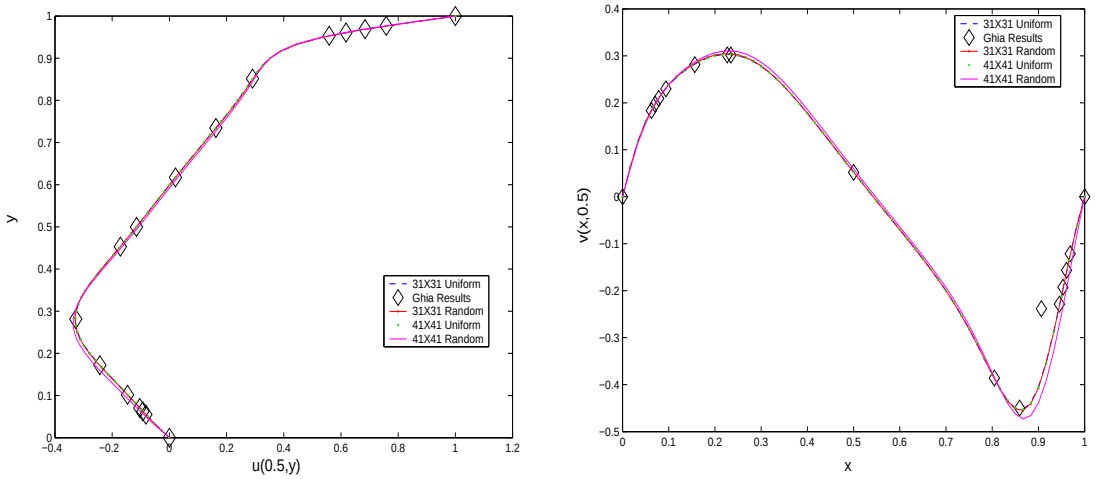
a) Vertical centre-line b) Horizontal centre-line

FIGURE 6.7: Square driven cavity: Comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with Ghia *et al.* (1982) for Re = 100



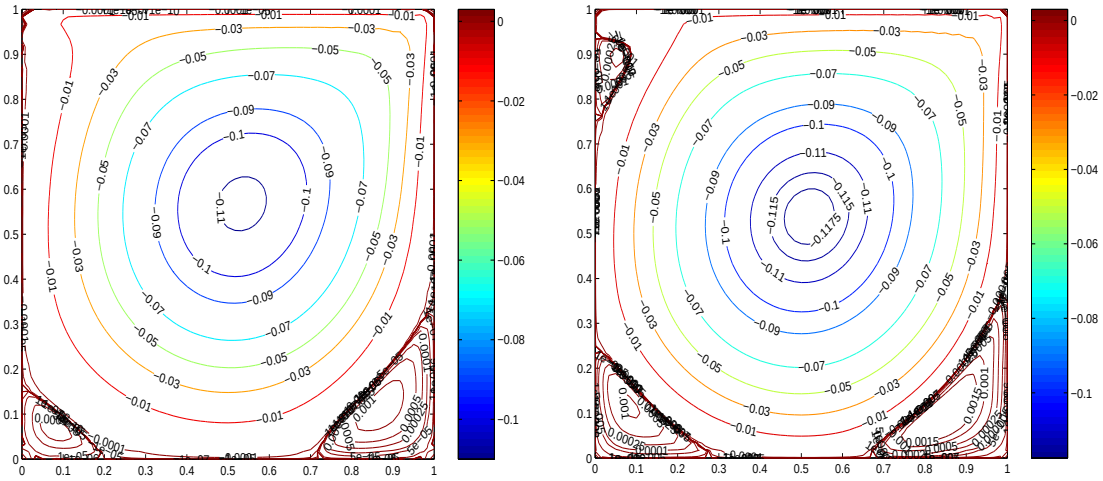
c) Streamfunction b) Vorticity

FIGURE 6.8: Square driven cavity: Re = 400, contours of streamfunction and vorticity obtained using 31×31 uniform point distribution



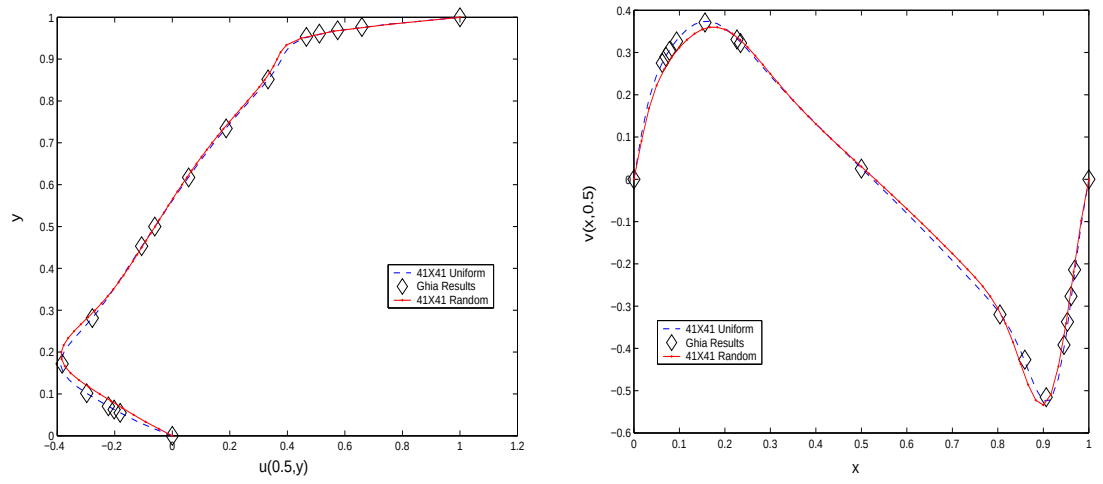
a) Vertical centre-line b) Horizontal centre-line

FIGURE 6.9: Square driven cavity: Comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with Ghia *et al.* (1982) for Re = 400



a) Re = 1000 b) Re = 3200

FIGURE 6.10: Square driven cavity: Streamline patterns obtained for Re = 1000 using 41×41 uniform points and Re = 3200 using 61×61 uniform points



b) Vertical centre-line c) Horizontal centre-line

FIGURE 6.11: Square driven cavity: Comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with Ghia *et al.* (1982) for Re = 1000 using 41×41 uniform and random distributions

is solved for three different Reynolds numbers of 100, 400 and 1000. A uniform distribution of points is considered for solving this problem.

The streamfunction contours obtained for the three Reynolds numbers are shown in Figures 6.13 - 6.14. From the figures, it can be observed that there are two rotating primary vortices as well as secondary vortices in the bottom corners of the rectangular cavity. The top primary vortex properties are reported in Table 6.1 and are compared with those obtained by Bruneau & Jouron (1990). It can be seen that the primary vortex strength and location values obtained are in in close agreement with the benchmark results.

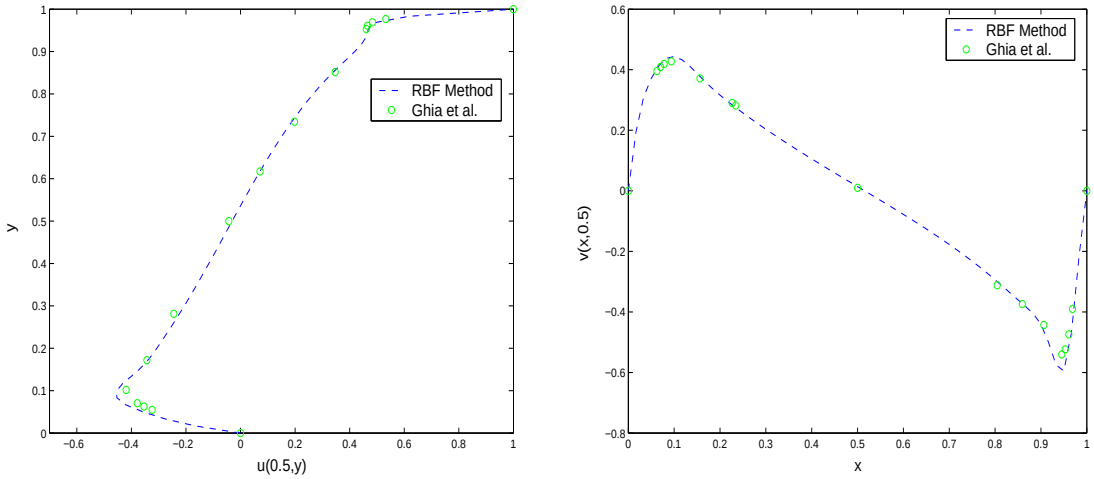
TABLE 6.1: Rectangular driven cavity: Top primary vortex strength and location and comparison with Bruneau & Jouron (1990)

Reynolds Number		$\psi_{\min}$	$\psi_{\min}$ location
100	(Bruneau & Jouron, 1990)	-0.1033	$(0.6172, 1.7344)$
	Present Method	-0.1032	$(0.617, 1.734)$
400	(Bruneau & Jouron, 1990)	-0.1124	$(0.5547, 1.5938)$
	Present Method	-0.1125	$(0.555, 1.610)$
1000	(Bruneau & Jouron, 1990)	-0.1169	$(0.5273, 1.5625)$
	Present Method	-0.1178	$(0.525, 1.57)$

6.5.3 Backward-facing step flow

The final model problem presented is flow over a backward-facing step. We use Gartling's problem definition (Gartling, 1990). Consider a channel of width L downstream of origin and width $\frac{L}{2}$ upstream of origin, separated by a backward facing step as shown in Figure 6.15. Flow is assumed to be fully developed as it passes the inlet at $x = 0$ and has an average velocity $\bar{U}$. The problem domain is the channel starting at the inlet and extends downstream a distance D long enough for the flow to again become fully developed. Reynolds number is defined as $\frac{\bar{U}L}{\nu}$. The boundary conditions for the problem are given in Table 6.2. The downstream distance D is taken to be $30L$ in order for the flow to be fully developed. This completes the specification of the problem.

Figure 6.16 presents contours of streamfunction and vorticity for Re = 200. It can be seen from the figure that a recirculation zone is formed downstream of the step face. The recirculation zone details obtained by the present method are compared against those obtained



b) Vertical centre-line c) Horizontal centre-line

FIGURE 6.12: Square driven cavity: Comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with Ghia *et al.* (1982) for Re = 3200 using 61×61 uniform distribution

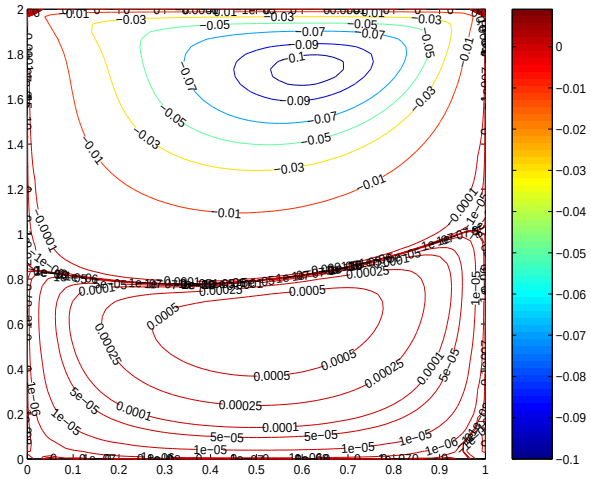
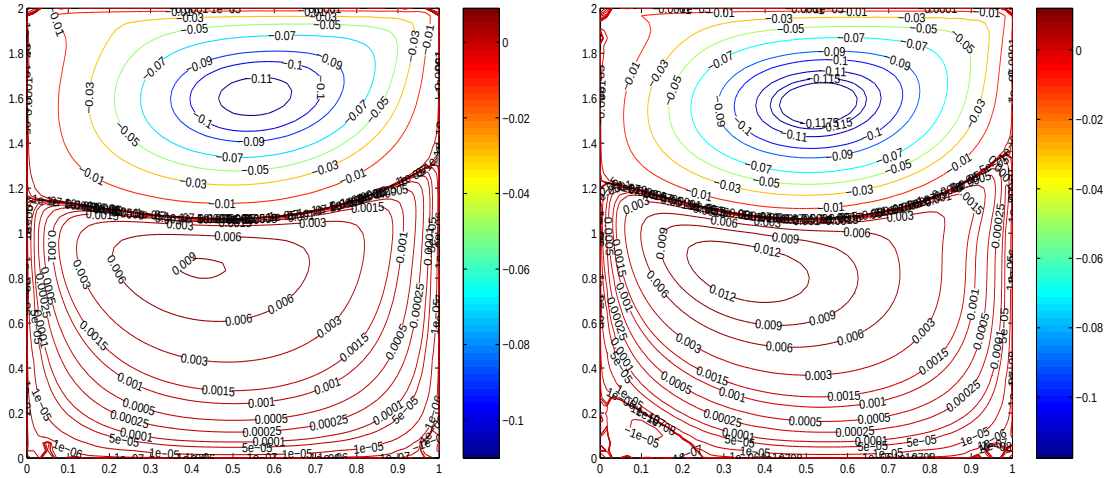


FIGURE 6.13: Rectangular driven cavity: Streamline patterns obtained for Re = 100 using 31×61 uniform point distribution



b) Re = 400 c) Re = 1000

FIGURE 6.14: Rectangular driven cavity: Streamline patterns obtained for Re = 400 using 31×61 uniform points and Re = 1000 using 41×81 uniform points

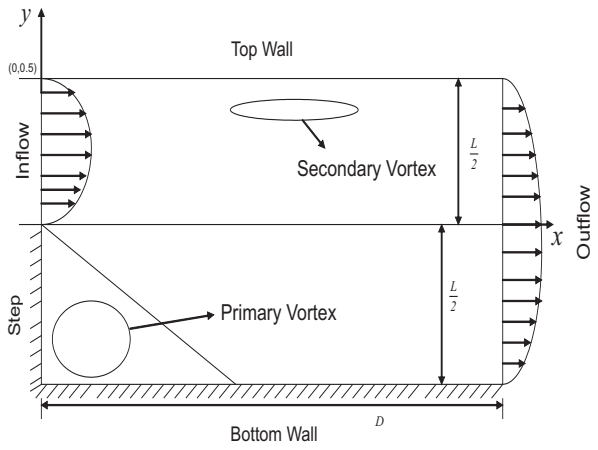


FIGURE 6.15: Backward-facing step flow

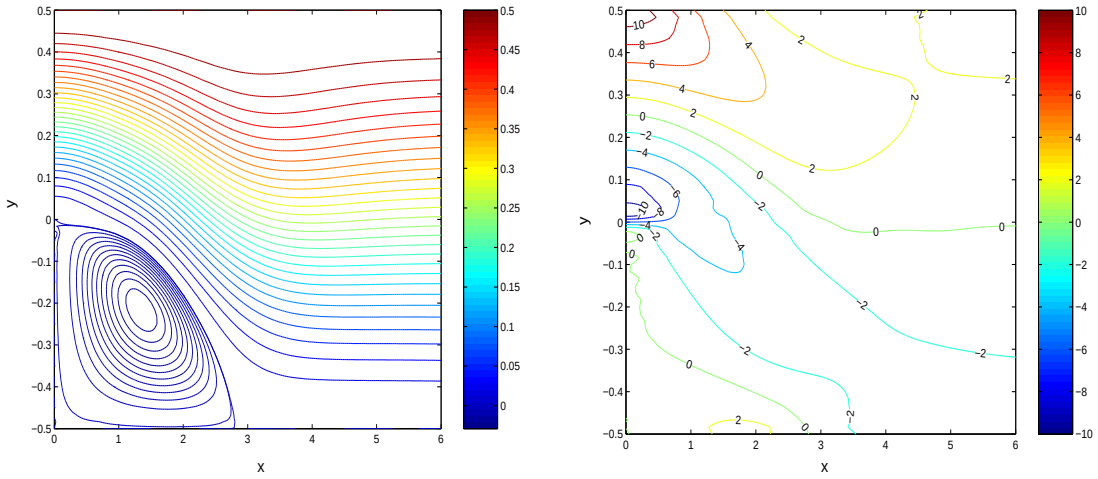
by using finite elements (Barragy, 1993) in Table 6.3. It can be seen that the finite element method under estimates the recirculation region from the table.

6.6 Conclusions

A radial basis function based meshless method is presented for the numerical solution of incompressible Navier-Stokes equations in streamfunction formulation. This method is capable of obtaining numerical solutions on a uniformly spaced or random set of points. A

TABLE 6.2: Boundary conditions for backward-facing step flow

$$\begin{aligned}
\text{Inlet:} \quad & \psi = 2y^2\,(3-4y),\ \frac{\partial \psi}{\partial x} = 0. \\
\text{Outlet:} \quad & \psi = \tfrac{1}{4}\left(1+3y-4y^3\right),\ \frac{\partial \psi}{\partial x} = 0. \\
\text{Step:} \quad & \psi = 0,\ \frac{\partial \psi}{\partial x} = 0. \\
\text{Bottom Wall:} \quad & \psi = 0,\ \frac{\partial \psi}{\partial y} = 0. \\
\text{Top Wall:} \quad & \psi = 0.5,\ \frac{\partial \psi}{\partial y} = 0.
\end{aligned}$$



b) Streamfunction c) Vorticity

FIGURE 6.16: Backward-facing step: Re = 200, contours of streamfunction and vorticity obtained using 41×41 uniform point distribution. Note that the scale parameters used for this problem are $L_x = 30$ and $L_y = 1.0$

TABLE 6.3: Backward-facing step: Re = 200, Primary vortex strength and location, length of recirculation region and its comparison with higher order finite elements (Barragy, 1993)

	Present Method	Finite Elements (Barragy, 1993)
Length of Recirculation	2.72	2.67
$\psi_{\min}$	-0.0315	-0.0331
$\psi_{\min}$ location	$(1.333, -0.2167)$	$(1.0021, -0.2030)$

novel ghost centre strategy was employed for incorporating the boundary conditions which circumvents the difficulty of specifying vorticity boundary conditions. A square driven cavity, rectangular cavity with aspect ratio 2 and backward-facing step flow were solved and

the results obtained by the present method are compared with benchmark solutions. From the comparisons made, it can be seen that the presented method can solve incompressible viscous flow problems accurately. However, the present method generates dense coefficient matrices which can be expensive to invert for large scale problems. The domain decomposition methods developed earlier in this thesis can be used when large scale problems are solved. However, the issue of optimal shape parameter value still remains as the shape parameter optimisation using L_2 norm of the residual error is computationally expensive. The next chapter presents a formulation for the incompressible NS equations using the RBF-FD method presented earlier in Chapter 5 that allows the shape parameter to be tuned efficiently.

Chapter 7

RBF-FD Schemes for the Incompressible Navier-Stokes Equations

In this chapter, we present the RBF-FD formulation for the incompressible Navier Stokes equations in streamfunction vorticity form. As shown in Chapter 5, the RBF-FD formulation generates sparse coefficient matrices and is hence suitable for large scale problems. The spatial discretisation of the incompressible NS equations is done using the RBF-FD method and the temporal discretisation is achieved by explicit Euler time-stepping and the Crank-Nicholson scheme. A novel ghost node strategy is employed for incorporating the no-slip boundary conditions. The performance of the RBF-FD method with the ghost node strategy is evaluated by solving driven cavity flow problems. Finally, a higher-order RBF-FD scheme which uses ideas from Hermite interpolation is proposed for solving the steady NS equations.

7.1 Incompressible Navier-Stokes equations

The non-dimensional governing equations for unsteady incompressible Navier Stokes equations expressed in terms of vorticity (ω) and streamfunction (ψ) are given by

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right), \tag{7.1}$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega, \tag{7.2}$$

where Re is the Reynolds number and u, v denote the components of velocity in the x and y directions which can be expressed in terms of the streamfunction as

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}. \tag{7.3}$$

In the RBF-FD method, we begin with representing the complete domain by a set of scattered nodes present in the interior and on the boundary. For each interior node, a supporting region/stencil is identified by choosing N nearest nodes. Then at each node, a local RBF interpolation problem is set up to determine the RBF-FD weights for each derivative (see section 5.1). This completes the pre-processing stage.

Once the RBF-FD method is applied to discretise the spatial derivatives in the governing equations, Equation (7.1) and Equation (7.2), we obtain at any interior node $\boldsymbol{x}_i$,

$$\begin{aligned} \frac{d\omega_i}{dt} + u_i \sum_{j=1}^{N} w_{(i,j)}^{(x)} \omega_j \quad &+ \quad v_i \sum_{j=1}^{N} w_{(i,j)}^{(y)} \omega_j \\ &= \quad \frac{1}{\mathrm{Re}} \left[\sum_{j=1}^{N} (w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)}) \omega_j \right], \end{aligned} \tag{7.4}$$

and

$$\sum_{j=1}^{N} (w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)}) \psi_j = -\omega_i, \tag{7.5}$$

where N is the total number of interior and boundary nodes which lie in the supporting region/stencil for the node $\boldsymbol{x}_i$, and $w_{(i,j)}^{(x)}$, $w_{(i,j)}^{(y)}$, $w_{(i,j)}^{(xx)}$, $w_{(i,j)}^{(yy)}$ are the RBF-FD weights obtained from the system of Equations (5.9) with the corresponding differential operator $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2})$ applied to the basis functions on the right hand side.

The system of ordinary differential equations obtained for vorticity after spatial discretisation, Equation (7.4), is advanced in time using the basic Euler time-stepping scheme. Denoting the value of any physical quantity at $t = t^n$ with the superscript n, we obtain

$$\begin{aligned} \frac{\omega_i^{n+1} - \omega_i^n}{\delta t} + u_i^n \sum_{j=1}^{N} w_{(i,j)}^{(x)} \omega_j^n \quad &+ \quad v_i^n \sum_{j=1}^{N} w_{(i,j)}^{(y)} \omega_j^n \\ &= \quad \frac{1}{\mathrm{Re}} \left[\sum_{j=1}^{N} (w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)}) \omega_j^n \right], \end{aligned} \tag{7.6}$$

where δt is the time-step. Similarly Equation (7.4) is temporally discretised using a θ-

weighting scheme ($0 \leq \theta \leq 1$), the discretised equation at the node $\boldsymbol{x}_i$ reads as

$$\begin{aligned}\frac{\omega_i^{n+1}-\omega_i^n}{\delta t} &= \theta\left[\frac{1}{\mathrm{Re}}\sum_{j=1}^{N}\left(w_{(i,j)}^{(xx)}+w_{(i,j)}^{(yy)}\right)\omega_j^{n+1}-\right.\\ &\qquad \left. u_i^n\sum_{j=1}^{N}w_{(i,j)}^{(x)}\omega_j^{n+1}-v_i^n\sum_{j=1}^{N}w_{(i,j)}^{(y)}\omega_j^{n+1}\right]\\ &\quad +(1-\theta)\left[\frac{1}{\mathrm{Re}}\sum_{j=1}^{N}\left(w_{(i,j)}^{(xx)}+w_{(i,j)}^{(yy)}\right)\omega_j^{n}-\right.\\ &\qquad \left. u_i^n\sum_{j=1}^{N}w_{(i,j)}^{(x)}\omega_j^{n}-v_i^n\sum_{j=1}^{N}w_{(i,j)}^{(y)}\omega_j^{n}\right]. \end{aligned} \tag{7.7}$$

Equations (7.6) and (7.7) need to be supplemented by the boundary condition for vorticity. The value of vorticity at the boundary is obtained by higher-order finite difference expressions (Spotz, 1995); see Table 7.1. Here, the subscript b refers to the value of the quantity on the boundary and subscript 1 refers to the interior node which is locally orthogonal to the boundary and at a distance h from the boundary.

TABLE 7.1: $\mathcal{O}(h^3)$ wall boundary conditions (Spotz, 1995)

Left Wall:	$\omega_b = -\frac{3}{h}\left[v_b + \left(\frac{\psi_1-\psi_b}{h}\right) + \frac{\omega_1 h}{6}\right]$
Right Wall:	$\omega_b = \frac{3}{h}\left[v_b + \left(\frac{\psi_b-\psi_1}{h}\right) - \frac{\omega_1 h}{6}\right]$
Bottom Wall:	$\omega_b = \frac{3}{h}\left[u_b - \left(\frac{\psi_1-\psi_b}{h}\right) - \frac{\omega_1 h}{6}\right]$
Top Wall:	$\omega_b = -\frac{3}{h}\left[u_b - \left(\frac{\psi_b-\psi_1}{h}\right) + \frac{\omega_1 h}{6}\right]$

Once the value of vorticity in the whole domain is obtained, the governing equation for the streamfunction, Equation (7.2), is solved with Dirichlet boundary conditions to update the streamfunction. This process is repeated until convergence,

$$\frac{\|\omega_{\mathrm{new}}-\omega_{\mathrm{old}}\|_2}{\|\omega_{\mathrm{new}}\|_2} \leq \varepsilon, \tag{7.8}$$

where ϵ is a pre-determined convergence limit. The complete procedure is outlined in Table 7.2.

TABLE 7.2: RBF-FD Algorithm for Incompressible Navier Stokes equations

Given an initial guess ψ^0 and ω^0 and a particular node configuration:

1. For each interior node, determine the support/stencil size.
2. Obtain the RBF-FD weights by solving the RBF interpolation problem.
3. Advance the vorticity solution to the next step using a suitable time-stepping algorithm.
4. Calculate the vorticity on the boundary using Table 7.1.
5. Solve Equation (7.2) with Dirichlet boundary conditions for streamfunction to obtain the new streamfunction values.
6. Check for convergence. If converged, stop Else go to step 3.

7.2 Ghost node strategy for incorporating boundary conditions

In the previous section, a locally orthogonal grid at the boundary is used to enforce the no-slip boundary conditions. This restriction on the nodes near the boundary makes the implementation of boundary conditions very straight forward. Considerable amount of work would be needed however to ensure a locally orthogonal grid near curved surfaces, and hence this approach would be cumbersome for complex geometries. In this section, we propose a method for implementing the no-slip boundary conditions based on ghost nodes. This ghost node strategy enables randomly placed points near the boundary and is still able to satisfy the boundary conditions accurately. Sample point distributions used in the locally orthogonal grid and the ghost node strategies are shown in Figure 7.1.

The no-slip boundary conditions at a boundary Γ are given by

$$\begin{aligned} \psi &= C_1 \qquad \boldsymbol{x} \in \Gamma, \\ \frac{\partial \psi}{\partial \vec{n}} &= C_2 \qquad \boldsymbol{x} \in \Gamma \end{aligned} \tag{7.9}$$

where C_1 and C_2 are constants and $\vec{n}$ is the outward normal direction from the boundary. In the proposed strategy, each boundary node is associated with a support region/stencil which also includes a ghost node placed outside the computational domain. The RBF-FD discretisation is carried out to approximate the normal derivative at the boundary node $\boldsymbol{x}_i$, i.e.,

$$\left.\frac{\partial \psi}{\partial \vec{n}}\right|_{\boldsymbol{x}_i} = \sum_{j=1}^{N} w_{(i,j)}^{(\vec{n})} \psi_j + w_{(i,\text{ghost})}^{(\vec{n})} \psi_{\text{ghost}}, \tag{7.10}$$

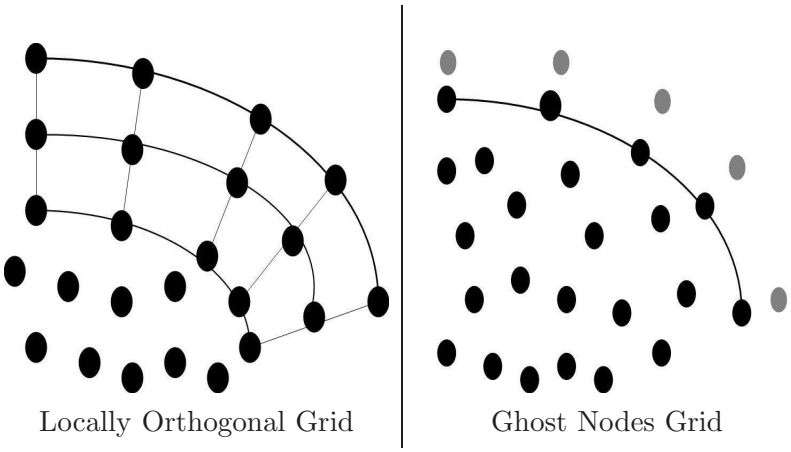


FIGURE 7.1: Schematic figure depicting the locally orthogonal boundary and the ghost nodes. Note that the ghost nodes are represented as grey shaded circles.

where N is the number of supporting points inside and on the boundary. The value of the streamfunction at the ghost node is evaluated by substituting the no-slip boundary condition Equation (7.9) and the streamfunction values of the interior nodes evaluated at the previous time step in Equation (7.10). The value of vorticity on the boundaries can then be evaluated by RBF-FD discretisation of Equation (7.2) at the boundary node $\boldsymbol{x}_i$.

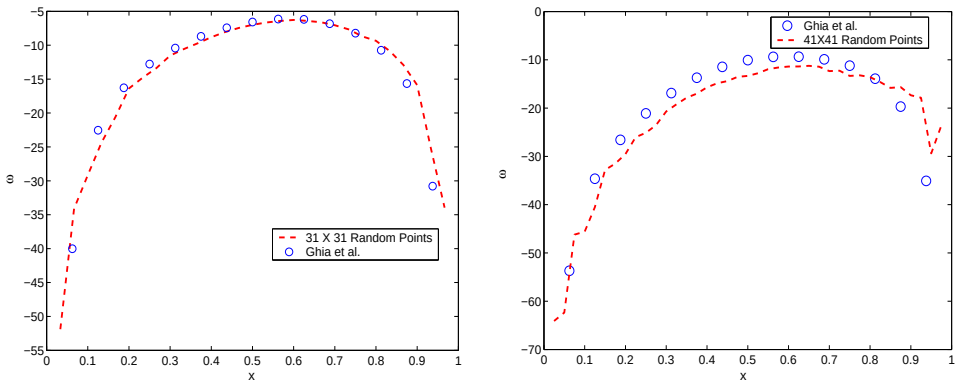
7.3 Numerical studies

In this section, we present numerical studies conducted on two test problems using the modified RBF-FD scheme with the ghost node strategy.

7.3.1 Square driven cavity flow

We first present numerical studies conducted on the lid-driven cavity flow problem in a square $[0,1] \times [0,1]$ domain. The boundary conditions for this problem are given in Equation (6.33). The results obtained using the presented RBF-FD formulation are validated against the benchmark multigrid finite difference results obtained in Ghia *et al.* (1982).

We use the time-dependent form of the governing equations in streamfunction-vorticity form. The spatial discretisation is done using the RBF-FD scheme while the temporal discretisation is carried out using the Crank-Nicholson method (Equation (7.7) with $\theta = 0.5$), with a time step $\delta t = 0.01$. Both uniform and random point distributions are considered and the flow problem is solved for three different Reynolds numbers ($\mathrm{Re} = \{100, 400, 1000\}$). We



Re = 100 Re = 400

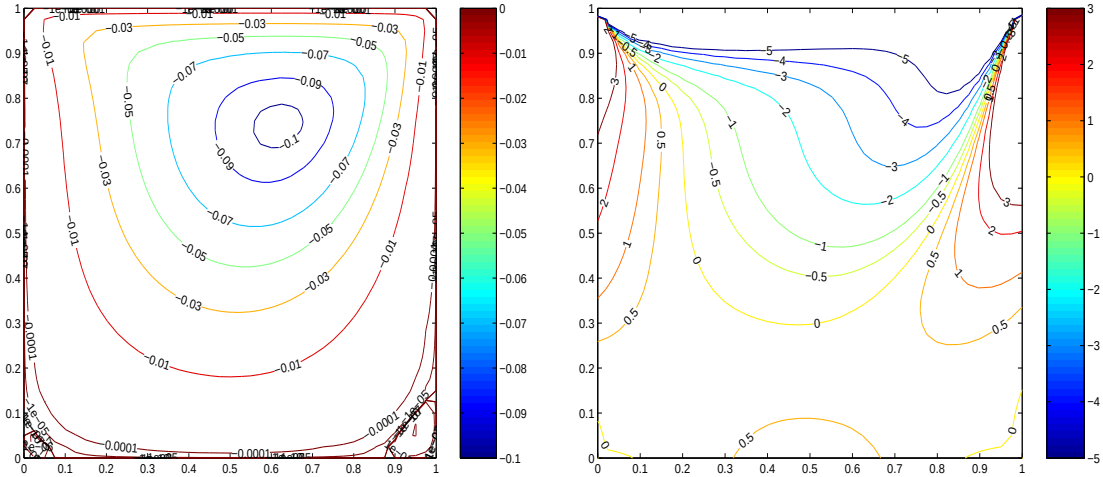
FIGURE 7.2: Comparison of wall vorticity obtained using ghost nodes on the moving boundary of the square driven cavity flow with Ghia *et al.* (1982)

use 9 supporting points in each RBF-FD stencil for discretisations of the function derivatives and the value of shape parameter is obtained using the leave-one-out optimisation strategy, outlined in Chapter 5, for each RBF-FD stencil. To apply the no-slip boundary conditions, the ghost node strategy proposed in Section 7.2 is employed on the boundary RBF-FD stencils. This facilitates a complete random point distribution in the interior of the domain.

We begin by examining the accuracy of the proposed ghost node strategy. Figure 7.2 shows the wall vorticity distribution obtained on the moving lid for the square lid-driven cavity flow problem at two different Reynolds numbers. A complete random distribution of points without any restriction at the nodes near the boundary was considered for obtaining the results. The results are compared with the wall vorticity values obtained by Ghia *et al.* (1982) for the purpose of validation. From the figures, it can be seen that the obtained vorticity distribution agrees well with the benchmark results.

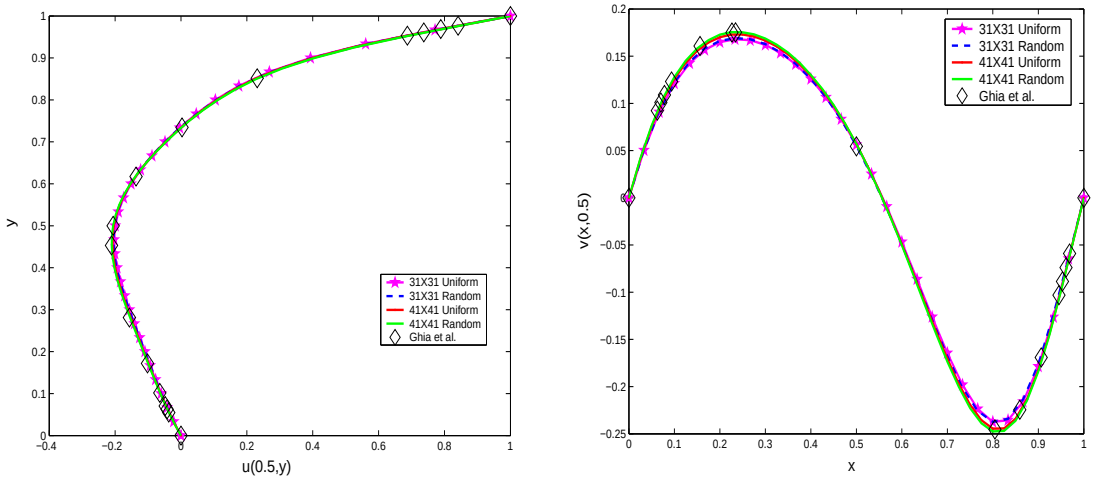
In Figure 7.3, the streamfunction and vorticity contours obtained using the RBF-FD method for Re = 100 are shown. The results displayed are generated using 41×41 randomly spaced points. From the plot of the streamfunction contours it can be seen that the secondary and tertiary vortices near the bottom wall are also captured. It is worth noting that the global features of the flow were captured with relatively small 21×21 distribution of points.

The comparison of velocity components at the horizontal and vertical centres of the cavity with those obtained by Ghia *et al.* (1982) are displayed in Figure 7.4. The velocity profiles obtained using 31×31 and 41×41 uniform and random distribution of points are presented in Figure 7.4. From Figure 7.4, it can clearly be seen that the velocity profiles are captured



a) Streamfunction b) Vorticity

FIGURE 7.3: Square driven cavity: Re = 100, contours of streamfunction and vorticity obtained using 41×41 random point distribution



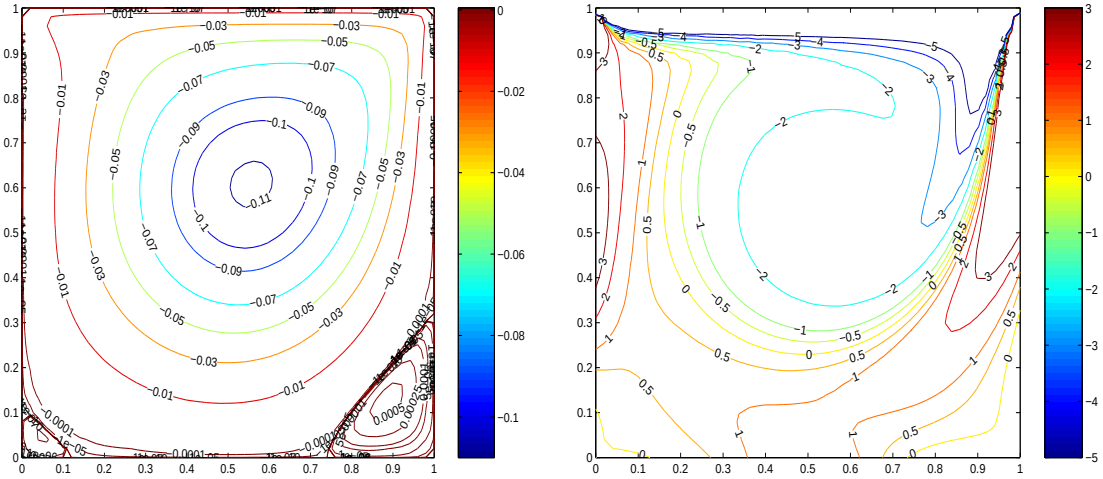
a) Vertical centre-line b) Horizontal centre-line

FIGURE 7.4: Square driven cavity: Comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with Ghia *et al.* (1982) for Re = 100

accurately as the number of points in the domain is increased.

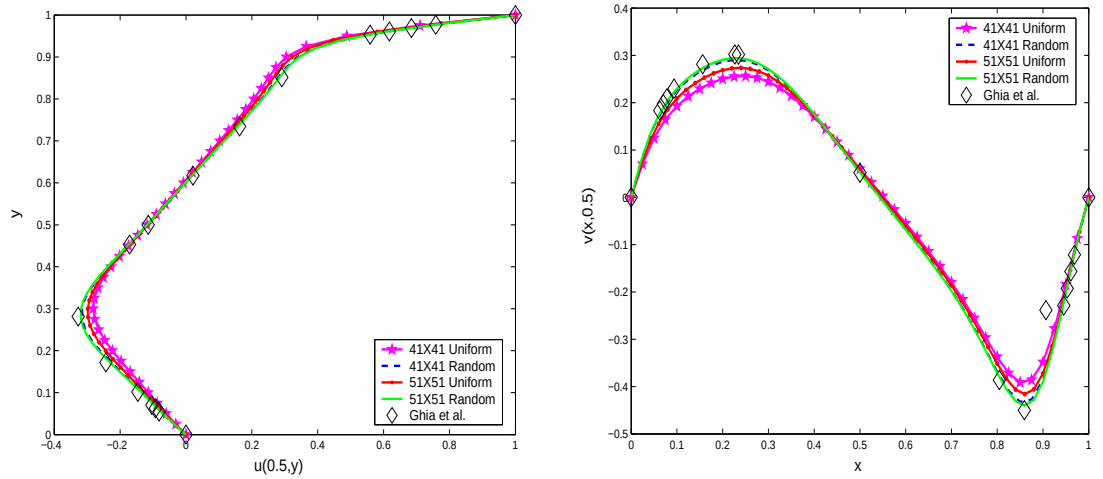
Figure 7.5 shows the streamfunction and vorticity contours obtained for Re = 400 and 51×51 random point distribution.The comparison of velocity profiles is presented in Figure 7.6. To accurately capture the velocity profiles for Re = 400, a larger number of points (51×51) were needed as compared to those required for Re = 100 (41×41). However,

the points required were much less than that of second order finite difference method which required about 129×129 points in order to capture the velocity profiles (Ghia *et al.*, 1982).



a) Streamfunction b) Vorticity

FIGURE 7.5: Square driven cavity: Re = 400, contours of streamfunction and vorticity obtained using 51×51 random point distribution

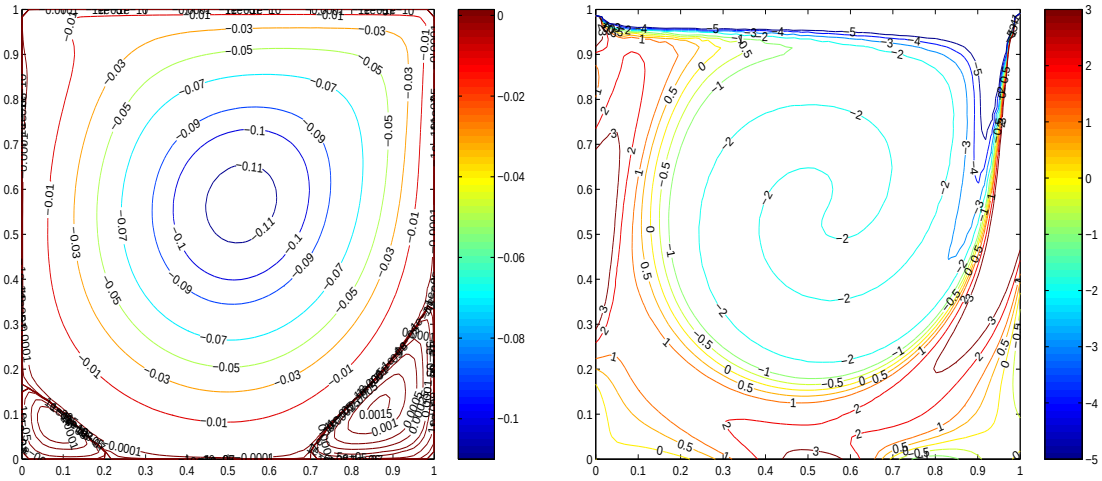


a) Vertical centre-line b) Horizontal centre-line

FIGURE 7.6: Square driven cavity: Comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with Ghia *et al.* (1982) for Re = 400

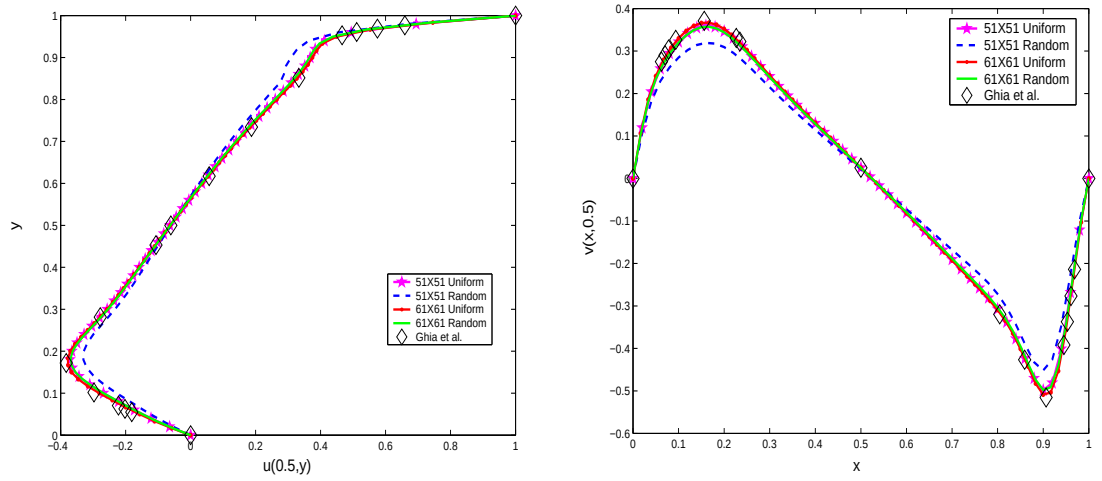
Similar results for Re = 1000 are shown in Figure 7.7 and Figure 7.8. The primary, secondary and tertiary vortices are captured satisfactorily. The velocity profiles obtained using 51×51 and 61×61 uniform and random point distributions are displayed in Figure

7.8.



a) Streamfunction b) Vorticity

FIGURE 7.7: Square driven cavity: Re = 1000, contours of streamfunction and vorticity obtained using 61×61 random point distribution



a) Vertical centre-line b) Horizontal centre-line

FIGURE 7.8: Square driven cavity: Comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with Ghia *et al.* (1982) for Re = 1000

In comparison with the RBF collocation method developed earlier in Chapter 6, the RBF-FD method is able to provide similar accuracy but at much lower computational cost. This reduction in computational cost is mainly due to the sparse structure of the coefficient matrices. It is also observed that although the sensitivity of the shape parameter is reduced

for the RBF-FD method, the shape parameter still influences the accuracy of the obtained solution, particularly when random node stencils are used for spatial discretisation.

7.3.2 Rectangular driven cavity flow

In this subsection, we apply the RBF-FD approach with the ghost node strategy for solving the driven cavity flow in a rectangular cavity with aspect ratio 2. The problem definition is provided in Section 6.5.2. The results are validated against those obtained by Gupta & Kalita (2005).

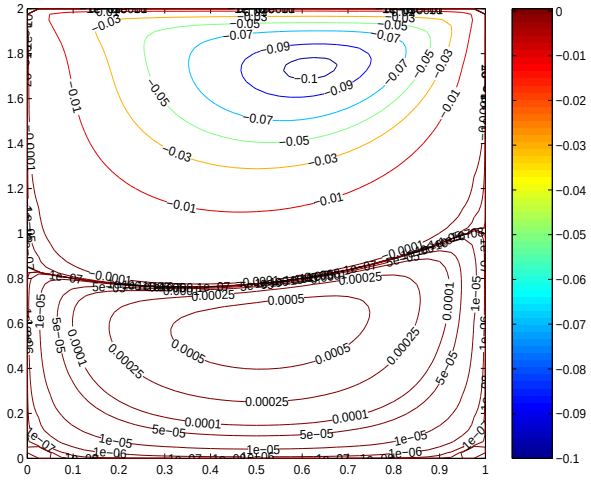


FIGURE 7.9: Rectangular driven cavity: Streamline patterns obtained for Re = 100 using 41×81 uniform point distribution

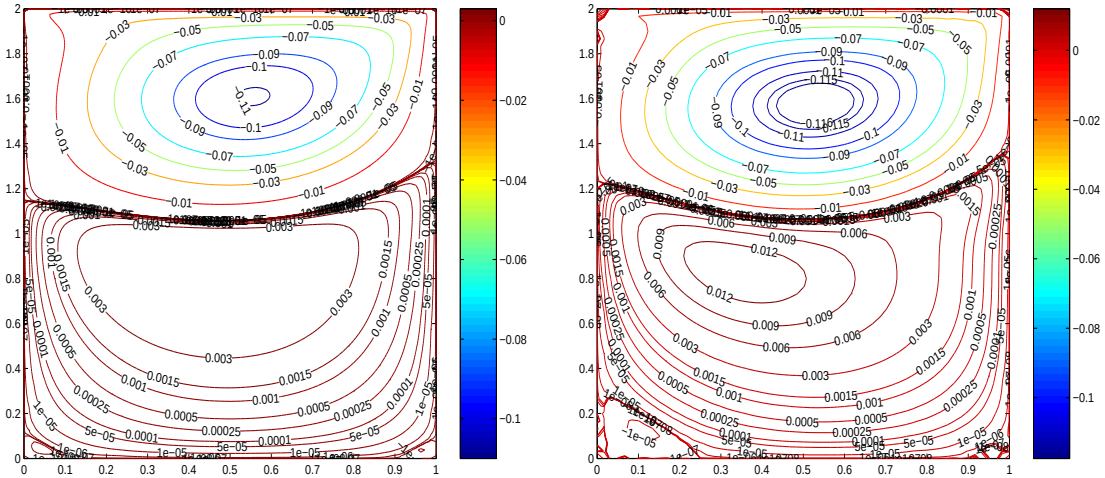
The streamfunction contours obtained for the three Reynolds numbers are shown in Figures 7.9 - 7.10. From the figures, it can be observed that there are two rotating primary vortices as well as secondary vortices in the bottom corners of the rectangular cavity. The top primary vortex properties are reported in Table 7.3, and are compared with those obtained by Bruneau & Jouron (1990). It can be seen that the RBF-FD method results are in close agreement with the benchmark results.

7.4 Higher-order RBF-FD schemes

In the first part of this chapter, we demonstrated the applicability of RBF-FD schemes for the incompressible Navier-Stokes equations. We now explore a higher-order version of the RBF-FD scheme using ideas from Hermite interpolation. This higher-order discretisation method using RBFs can be regarded as a generalisation of the *Mehrstellenvarfahren* introduced by

TABLE 7.3: Rectangular driven cavity: Top primary vortex strength and location and comparison with Bruneau & Jouron (1990)

Reynolds Number		$\psi_{\min}$	$\psi_{\min}$ location
100	(Bruneau & Jouron, 1990)	-0.1033	$(0.6172, 1.7344)$
	Present Method	-0.1030	$(0.625, 1.721)$
400	(Bruneau & Jouron, 1990)	-0.1124	$(0.5547, 1.5938)$
	Present Method	-0.1120	$(0.555, 1.6125)$
1000	(Bruneau & Jouron, 1990)	-0.1169	$(0.5273, 1.5625)$
	Present Method	-0.1165	$(0.525, 1.57)$



b) Re = 400 c) Re = 1000

FIGURE 7.10: Rectangular driven cavity: Streamline patterns obtained for Re = 400 using 81×161 uniform points and Re = 1000 using 101×201 uniform points

Collatz (1960) and later developed into compact FD formulas by Lele (1992). In the compact FD methodology, for example, the partial derivative of an unknown function with respect to the x-coordinate at any grid point (i, j) is given by

$$\left.\frac{\partial u}{\partial x}\right|_{(i,j)} \approx \sum_{k\in\{i-1,i,i+1\}} w_{(k,j)} u_{(k,j)} + \sum_{k\in\{i-1,i+1\}} \widetilde{w}_{(k,j)} \left.\frac{\partial u}{\partial x}\right|_{(k,j)} . \tag{7.11}$$

The accuracy of the finite difference approximation is increased by adding the second term (derivative information), as shown in Equation (7.11). Note that this additional term does not change the stencil size at the grid point (i, j) in the compact finite difference methodology.

Higher order RBF-FD methods were earlier used for the solution of linear and nonlinear Poisson problems (Wright & Fornberg, 2006). In this thesis, this work is extended to develop higher-order schemes for the incompressible Navier-Stokes equations. We demonstrate the higher-order accuracy of the presented formulation by solving the steady state incompressible Navier Stokes equations.

7.4.1 Basic formulation

The RBF-FD method generates a local RBF interpolant for expressing the function derivatives at a node as a linear combination of the function values on the nodes present in the support region of the considered node. In the spirit of compact finite difference schemes, the accuracy of the RBF-FD discretisation can be increased by considering not only the function values but also the derivative values on the nodes present in the supporting region. The weights of the higher-order stencil are computed using the Hermite interpolation technique.

We begin with a brief introduction to the Hermite interpolation method. Let $\mathcal{L}$ be an arbitrary linear differential operator and let $\boldsymbol{\eta}$ be a vector containing some combination of $m \leq n$ distinct numbers from the set $\{1, 2, \cdots, n\}$. The function values $u(\boldsymbol{x}_i)$ are specified at each of the n distinct data points $\{\boldsymbol{x}_i\}_{i=1}^n$. In addition, data corresponding to the differential operator operating on the the function, $\mathcal{L}u(\boldsymbol{x}_{\eta_l})$, is specified at m points $\{\boldsymbol{x}_{\eta_l}\}_{l=1}^m$. Note that the point set $\{\boldsymbol{x}_{\eta_l}\}_{l=1}^m$ is a subset of the set $\{\boldsymbol{x}_i\}_{i=1}^n$. Then, the interpolant passing through all the data can be written as

$$u(\boldsymbol{x}) \approx s(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\boldsymbol{x} - \boldsymbol{x}_i\|) + \sum_{l=1}^{m} \widetilde{\lambda}_l \mathcal{L}_2 \phi(\|\boldsymbol{x} - \boldsymbol{x}_{\eta_l}\|) + \beta, \tag{7.12}$$

where $\mathcal{L}_2\phi(\|.\|)$ is a basis function derived by the functional $\mathcal{L}$ acting on the multiquadric basis $\phi(\|.\|)$ as a function of the second variable (centre) and β is a constant. The unknown coefficients are obtained by enforcing the conditions $s(\boldsymbol{x}_i) = u(\boldsymbol{x}_i),\ i = 1, \cdots, n$; $\mathcal{L}s(\boldsymbol{x}_{\eta_l}) = \mathcal{L}u(\boldsymbol{x}_{\eta_l}),\ l = 1, \cdots, m$; and $\sum_{i=1}^n \lambda_i = 0$. Imposing these conditions leads to the following block linear system of equations

$$\begin{bmatrix} \Phi & \mathcal{L}_2\Phi & \mathbf{e} \\ \mathcal{L}\Phi & \mathcal{L}\mathcal{L}_2\Phi & \mathbf{0} \\ \mathbf{e}^T & \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \widetilde{\boldsymbol{\lambda}} \\ \beta \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ \mathcal{L}\boldsymbol{u} \\ 0 \end{bmatrix}, \tag{7.13}$$

where

$$\begin{aligned}
\Phi_{i,j} &= \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \quad i,j = 1,\cdots,n,\\
\mathcal{L}_2\Phi_{i,j} &= \mathcal{L}_2\phi(\|\boldsymbol{x}_i - \boldsymbol{x}_{\eta_j}\|), \quad i = 1,\cdots,n, \quad j = 1,\cdots,m,\\
\mathcal{L}\Phi_{i,j} &= \mathcal{L}\phi(\|\boldsymbol{x}_{\eta_i} - \boldsymbol{x}_j\|), \quad i = 1,\cdots,m, \quad j = 1,\cdots,n,\\
\mathcal{L}\mathcal{L}_2\Phi_{i,j} &= \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\eta_i} - \boldsymbol{x}_{\eta_j}\|), \quad i = 1,\cdots,m, \quad j = 1,\cdots,m,
\end{aligned}$$

and $\boldsymbol{e}_i = 1, \quad i = 1,\cdots,n$. Equation (7.13) is solved using a backward substitution routine.

The Hermite interpolant can also be written in Lagrange form as

$$\bar{s}(\boldsymbol{x}) = \sum_{i=1}^{n} \chi(\|\boldsymbol{x} - \boldsymbol{x}_i\|)u(\boldsymbol{x}_i) + \sum_{l=1}^{m} \widetilde{\chi}(\|\boldsymbol{x} - \boldsymbol{x}_{\eta_l}\|)\mathcal{L}u(\boldsymbol{x}_{\eta_l}), \tag{7.14}$$

where $\chi(\|\boldsymbol{x} - \boldsymbol{x}_i\|)$ and $\widetilde{\chi}(\|\boldsymbol{x} - \boldsymbol{x}_{\eta_l}\|)$ are of the form Equation (7.12) and satisfy the cardinal conditions, i.e.,

$$\chi(\|\boldsymbol{x}_k - \boldsymbol{x}_i\|) = \begin{cases} 1, & \text{if} \quad k = i,\\ 0, & \text{if} \quad k \neq i, \end{cases} \qquad k = 1,\cdots,n, \tag{7.15}$$

$$\mathcal{L}\chi(\|\boldsymbol{x}_{\eta_k} - \boldsymbol{x}_i\|) = 0, \quad k = 1,\cdots,m, \tag{7.16}$$

and

$$\widetilde{\chi}(\|\boldsymbol{x}_k - \boldsymbol{x}_{\eta_l}\|) = 0, \quad k = 1,\cdots,n, \tag{7.17}$$

$$\mathcal{L}\widetilde{\chi}(\|\boldsymbol{x}_{\eta_k} - \boldsymbol{x}_{\eta_l}\|) = \begin{cases} 1, & \text{if} \quad k = l,\\ 0, & \text{if} \quad k \neq l, \end{cases} \qquad k = 1,\cdots,m. \tag{7.18}$$

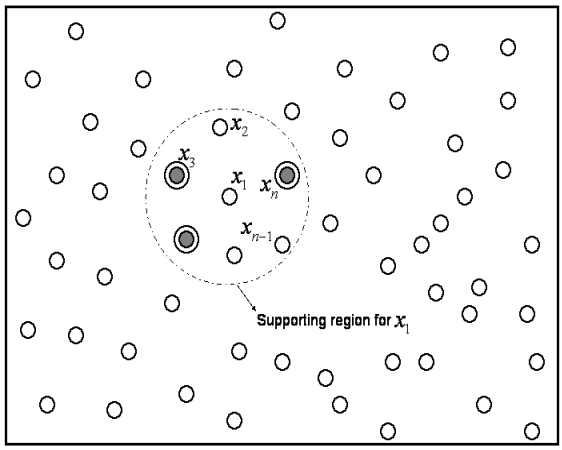


FIGURE 7.11: Schematic diagram of a higher-order RBF-FD stencil. The circle indicates the supporting region/stencil for the node $\boldsymbol{x}_1$.

Equation (7.14) is the basis for deriving higher-order RBF-FD stencils. Consider the node $\boldsymbol{x}_1$ with its support region containing n points, denoted by a dashed circle around $\boldsymbol{x}_1$

in Figure 7.11. The goal is to obtain a higher-order RBF-FD discretisation of $\mathcal{L}u(\boldsymbol{x}_1)$. The nodes in the support region which are shaded grey are those nodes where both the function values ($u(\boldsymbol{x})$) and the functional values ($\mathcal{L}u(\boldsymbol{x})$) are used i.e., the set η of cardinality, say, $m \leq n$.

The higher-order RBF-FD discretisation for $\mathcal{L}u(\boldsymbol{x}_1)$ is given by the Lagrange form of the interpolant, i.e.,

$$\mathcal{L}u(\boldsymbol{x}_1) \approx \mathcal{L}\bar{s}(\boldsymbol{x}_1) = \sum_{i=1}^{n} \mathcal{L}\chi(\|\boldsymbol{x}_1 - \boldsymbol{x}_i\|)u(\boldsymbol{x}_i) + \sum_{l=1}^{m} \mathcal{L}\widetilde{\chi}(\|\boldsymbol{x}_1 - \boldsymbol{x}_{\eta_l}\|)\mathcal{L}u(\boldsymbol{x}_{\eta_l}). \tag{7.19}$$

Equation (7.19) can be rewritten as a compact FD formula of the form

$$\mathcal{L}u(\boldsymbol{x}_1) \approx \sum_{i=1}^{n} w_{(1,i)}^{\mathcal{L}} u(\boldsymbol{x}_i) + \sum_{l=1}^{m} \widetilde{w}_{(1,l)}^{\mathcal{L}} \mathcal{L}u(\boldsymbol{x}_{\eta_l}), \tag{7.20}$$

where the weights for higher-order RBF-FD $\{w_{(1,i)}^{\mathcal{L}}\}_{i=1}^{n}$ and $\{\widetilde{w}_{(1,l)}^{\mathcal{L}}\}_{l=1}^{m}$ are now given by

$$w_{(1,i)}^{\mathcal{L}} = \mathcal{L}\chi(\|\boldsymbol{x}_1 - \boldsymbol{x}_i\|), \quad \widetilde{w}_{(1,l)}^{\mathcal{L}} = \mathcal{L}\widetilde{\chi}(\|\boldsymbol{x}_1 - \boldsymbol{x}_{\eta_l}\|), \tag{7.21}$$

where the superscript $\mathcal{L}$ on the weights denote that the higher-order RBF-FD weights are computed for that particular operator.

In practise, the weights are computed by solving the linear system

$$\begin{bmatrix} \Phi & \mathcal{L}_2\Phi & \mathbf{e} \\ \mathcal{L}\Phi & \mathcal{L}\mathcal{L}_2\Phi & \mathbf{0} \\ \mathbf{e}^T & \mathbf{0}^T & 0 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{w} \\ \widetilde{\boldsymbol{w}} \\ \mu \end{bmatrix} = \begin{bmatrix} \mathcal{L}^\star\Phi_1 \\ \mathcal{L}^\star\widetilde{\Phi}_1 \\ 0 \end{bmatrix}, \tag{7.22}$$

where $\mathcal{L}^\star\Phi_1$ and $\mathcal{L}^\star\widetilde{\Phi}_1$ denote the evaluation of the column vectors $\mathcal{L}^\star\Phi = [\mathcal{L}\phi(\|\boldsymbol{x}-\boldsymbol{x}_1\|)\ \mathcal{L}\phi(\|\boldsymbol{x}-\boldsymbol{x}_2\|)\ \cdots\ \mathcal{L}\phi(\|\boldsymbol{x}-\boldsymbol{x}_n\|)]^T$ and $\mathcal{L}^\star\widetilde{\Phi} = [\mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}-\boldsymbol{x}_{\eta_1}\|)\ \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}-\boldsymbol{x}_{\eta_2}\|)\ \cdots\ \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}-\boldsymbol{x}_{\eta_m}\|)]^T$ at the node $\boldsymbol{x}_1$. Here, μ is a scalar value related to the constant β in Equation (7.12) and enforces the condition

$$\sum_{i=1}^{n} w_{(1,i)} = 0,$$

which ensures that the stencil is exact for all constants. The derivation of Equation (7.22) is very similar to that of the derivation presented in Appendix C.

Once the weights are computed by solving Equation (7.22) for each node, they can be stored and used to discretise the partial differential equation in a similar manner as in the compact finite difference schemes.

We now illustrate the higher-order RBF-FD approach for a Poisson problem of the form

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \quad (x,y) \in \Omega \tag{7.23}$$

with Dirichlet boundary condition

$$u(x,y) = g(x,y), \quad (x,y) \in \Gamma \tag{7.24}$$

where Γ represents the boundary of the domain Ω.

We consider a stencil for a node $\boldsymbol{x}_1$ as shown in Figure 7.12, where the function values $u(\boldsymbol{x})$ are taken for nodes with single circle and for the double circled nodes both $u(\boldsymbol{x})$ are $\mathcal{L}u(\boldsymbol{x})$ are used. Following the previous notation, $\boldsymbol{\eta} = \{2,4,6,8\}$ and the value of any physical quantity at the node $\boldsymbol{x}_i$ is denoted by the subscript i.

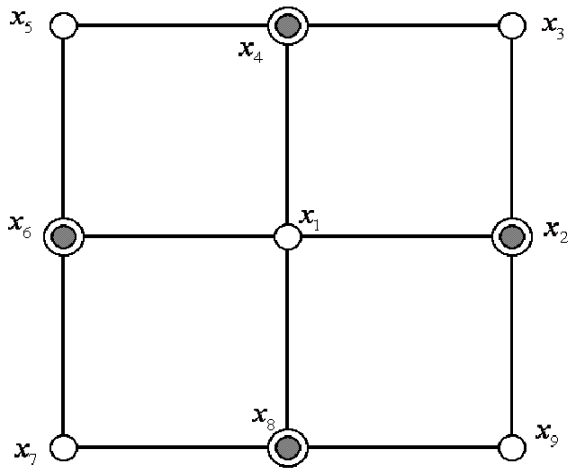


FIGURE 7.12: Higher order RBF-FD stencil for Poisson Problem

The weights for this stencil are obtained by solving

$$\boldsymbol{A}_H \boldsymbol{W} = \boldsymbol{b}, \tag{7.25}$$

where

$$\boldsymbol{A}_H = \begin{bmatrix} \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) & \cdots & \phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_9\|) & \mathcal{L}_2\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_{\eta_1}\|) & \cdots & \mathcal{L}_2\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_{\eta_4}\|) & 1 \\ \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|) & \cdots & \phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_9\|) & \mathcal{L}_2\phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_{\eta_1}\|) & \cdots & \mathcal{L}_2\phi(\|\boldsymbol{x}_2 - \boldsymbol{x}_{\eta_4}\|) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi(\|\boldsymbol{x}_9 - \boldsymbol{x}_1\|) & \cdots & \phi(\|\boldsymbol{x}_9 - \boldsymbol{x}_9\|) & \mathcal{L}_2\phi(\|\boldsymbol{x}_9 - \boldsymbol{x}_{\eta_1}\|) & \cdots & \mathcal{L}_2\phi(\|\boldsymbol{x}_9 - \boldsymbol{x}_{\eta_4}\|) & 1 \\ \mathcal{L}\phi(\|\boldsymbol{x}_{\eta_1} - \boldsymbol{x}_1\|) & \cdots & \mathcal{L}\phi(\|\boldsymbol{x}_{\eta_1} - \boldsymbol{x}_9\|) & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\eta_1} - \boldsymbol{x}_{\eta_1}\|) & \cdots & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\eta_1} - \boldsymbol{x}_{\eta_4}\|) & 0 \\ \mathcal{L}\phi(\|\boldsymbol{x}_{\eta_2} - \boldsymbol{x}_1\|) & \cdots & \mathcal{L}\phi(\|\boldsymbol{x}_{\eta_2} - \boldsymbol{x}_9\|) & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\eta_2} - \boldsymbol{x}_{\eta_1}\|) & \cdots & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\eta_2} - \boldsymbol{x}_{\eta_4}\|) & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{L}\phi(\|\boldsymbol{x}_{\eta_4} - \boldsymbol{x}_1\|) & \cdots & \mathcal{L}\phi(\|\boldsymbol{x}_{\eta_4} - \boldsymbol{x}_9\|) & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\eta_4} - \boldsymbol{x}_{\eta_1}\|) & \cdots & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\eta_4} - \boldsymbol{x}_{\eta_4}\|) & 0 \\ 1 & \cdots & 1 & 0 & \cdots & 0 & 0 \end{bmatrix}_{14\times 14},$$

$$\boldsymbol{W} = \begin{bmatrix} w^{\mathcal{L}}_{(1,1)} & \cdots & w^{\mathcal{L}}_{(1,9)} & \widetilde{w}^{\mathcal{L}}_{(1,1)} & \cdots & \widetilde{w}^{\mathcal{L}}_{(1,4)} & \mu \end{bmatrix}^T \text{ and}$$

$$\boldsymbol{b} = [\mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) \ \cdots \ \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_9\|) \ \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_{\eta_1}\|) \ \cdots \ \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_{\eta_4}\|) \ 0]^T .$$

Once the higher-order RBF-FD weights are determined we proceed to the next step, i.e., discretisation. The discretisation of the governing equation, Equation (7.23), at node $\boldsymbol{x}_1$ is given by

$$\sum_{j=1}^{9} w^{\mathcal{L}}_{(1,j)} u_j + \sum_{l=1}^{4} \widetilde{w}^{\mathcal{L}}_{(1,l)} \mathcal{L}u_{\eta_l} = f_1. \tag{7.26}$$

Note that from the governing equation, (Equation (7.23)), $\mathcal{L}u_{\eta_l}$ at the node $\boldsymbol{x}_{\eta_l}$ is given by f_{η_l} which is known. Hence, Equation (7.26) becomes

$$\sum_{j=1}^{9} w^{\mathcal{L}}_{(1,j)} u_j = f_1 - \sum_{l=1}^{4} \widetilde{w}^{\mathcal{L}}_{(1,l)} f_{\eta_l}. \tag{7.27}$$

This procedure is repeated for each of the interior nodes to obtain the discretised form of the PDE at each of the interior nodes. Substituting the function values u_j from boundary condition, Equation (7.24), whenever the support point $\boldsymbol{x}_j \in \Gamma$, we obtain a system of equations which can be written in matrix form as

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}, \tag{7.28}$$

where $\boldsymbol{u}$ is the vector of the unknown function values at all the interior nodes and $\boldsymbol{f}$ is the source vector including the boundary terms. This equation can be solved to obtain the unknown vector $\boldsymbol{u}$.

7.4.2 Numerical studies

In this section, we present numerical studies conducted on a model Poisson and steady state convection-diffusion equation using the higher-order RBF-FD method presented in the earlier subsections.

The source term and exact solution for Poisson equation are given in Equation (5.17) and Equation (5.18). We consider the 9 noded stencil for the RBF-FD method and the stencil shown in Figure 7.12 for the higher-order RBF-FD method. The shape parameter obtained for the RBF-FD stencil using the leave-one-out criterion is used for the higher-order RBF-FD method.

Figure 7.13 presents the convergence behaviours obtained for the Poisson problem using both the RBF-FD and higher-order RBF-FD method. It can be clearly seen that the higher-order RBF-FD method is more accurate than the RBF-FD method for the same number of nodes or mesh spacing (h).

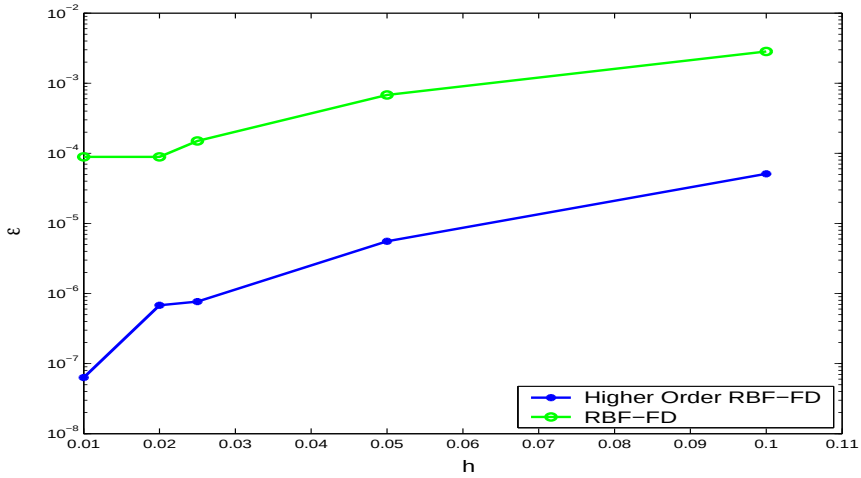


FIGURE 7.13: Comparison of convergence behaviours of RBF-FD and higher-order RBF-FD for a model Poisson equation

The steady convection-diffusion problem that is considered next is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - P_e \frac{\partial u}{\partial x} = 0, \tag{7.29}$$

in the domain $[0,1] \times [0,0.6]$ with the boundary conditions

$$u = 1 \quad \text{on } x = 0, \quad u = 2 \quad \text{on } x = 1, \tag{7.30}$$

$$\frac{\partial u}{\partial y} = 0 \quad \text{on } y = 0, \quad \tfrac{\partial u}{\partial y} = 0 \quad \text{on } y = 1. \tag{7.31}$$

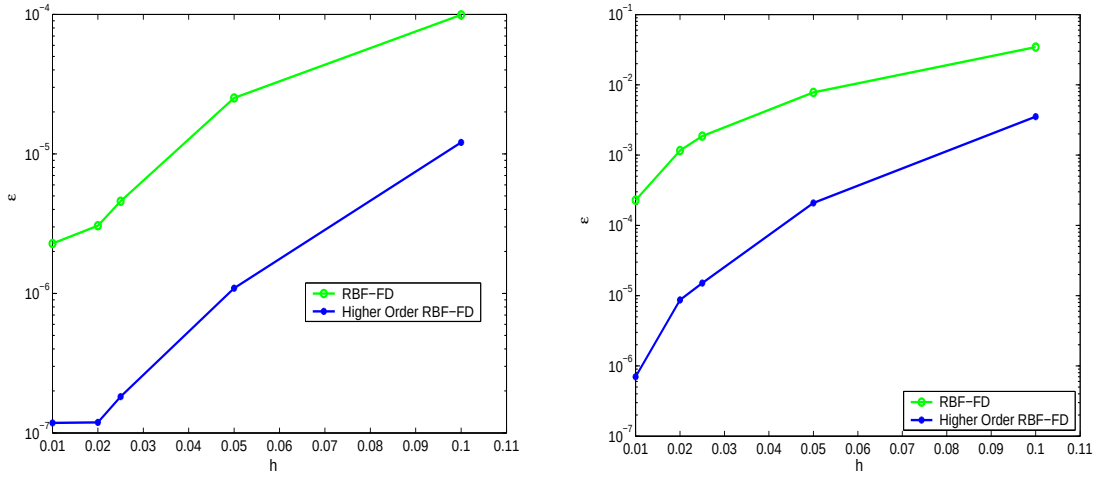
The exact solution for this problem is given by

$$u_{\text{exact}} = 2 - \frac{1 - \exp\left[P_e(x-1)\right]}{1 - \exp\left(-P_e\right)}. \tag{7.32}$$

Figure 7.14 presents the convergence plots of the RBF-FD and higher-order RBF-FD for the convection diffusion problem, Equation (7.29), for two Peclet numbers 1.0 and 10.0. The operator $\mathcal{L}$ is taken as

$$\mathcal{L} \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - P_e \frac{\partial}{\partial x},$$

and the higher-order RBF-FD weights are obtained using Equation (7.22). From Figure 7.14, it can be seen that the results obtained using the higher-order method is at least two orders more accurate than that of RBF-FD method. It is also worth mentioning that for $P_e = 100.0$, with a uniform discretisation of 201×201 nodes, the error norm observed for the computed solution using higher-order RBF-FD was $\varepsilon = \mathcal{O}(10^{-4})$.



a) $P_e = 1.0$ b) $P_e = 10.0$

FIGURE 7.14: Comparison of convergence behaviours of RBF-FD and higher-order RBF-FD for a model steady state convection-diffusion equation

7.5 Higher order RBF-FD for the incompressible NS equations

The previous section outlined the idea of using the Hermite interpolation technique to obtain a higher-order RBF-FD discretisation for each RBF-FD stencil. It can be observed that a family of higher-order schemes can be derived by defining which operator $\mathcal{L}$ one is using for the higher-order RBF-FD stencil. In this section, we present one such formulation for the steady state incompressible NS equations. This formulation has the advantage of easier implementation of the no-slip boundary conditions. We begin with recalling the governing equations of steady incompressible flows in streamfunction (ψ) - vorticity (ω) formulation:

$$\nabla^2 \psi = -\omega, \tag{7.33}$$

$$\nabla^2 \omega = \mathrm{Re}\left(u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y}\right), \tag{7.34}$$

where Re is the Reynolds number and (u, v) are the Cartesian velocity components of the flow.

As usual, we begin by discretising the entire domain into a set of interior and boundary nodes and determine the stencil at each node. Each higher-order RBF-FD stencil contains n nodes and the vector $\boldsymbol{\eta}$ of cardinality $m \leq n$. The stencil information consists of the

function values ($\psi(\boldsymbol{x})$ or $\omega(\boldsymbol{x})$) at each of the n nodes and the functional information ($\mathcal{L}\psi(\boldsymbol{x})$ or $\mathcal{L}\omega(\boldsymbol{x})$) on each of the m nodes. Note that the operator $\mathcal{L}$ is arbitrary.

The higher-order RBF-FD discretisations for the Laplacian of streamfunction at each interior node $\boldsymbol{x}_i$ is given by

$$\nabla^2\psi(\boldsymbol{x}_i) \approx \sum_{j=1}^{n} w_{(i,j)}^{\nabla^2}\psi(\boldsymbol{x}_j) + \sum_{l=1}^{m} \widetilde{w}_{(i,l)}^{\nabla^2}\left\{\nabla^2\psi(\boldsymbol{x}_{\eta_l})\right\}, \tag{7.35}$$

where the higher-order RBF-FD weights $\{w_{(i,j)}^{\nabla^2}\}_{j=1}^{n}$ and $\{\widetilde{w}_{(i,l)}^{\nabla^2}\}_{l=1}^{m}$ are obtained using Equation (7.25) with the operator $\mathcal{L} \equiv \nabla^2$. Similarly the discretisation of Laplacian of vorticity can be obtained as

$$\nabla^2\omega(\boldsymbol{x}_i) \approx \sum_{j=1}^{n} w_{(i,j)}^{\nabla^2}\omega(\boldsymbol{x}_j) + \sum_{l=1}^{m} \widetilde{w}_{(i,l)}^{\nabla^2}\left\{\nabla^2\omega(\boldsymbol{x}_{\eta_l})\right\}. \tag{7.36}$$

For the sake of brevity, we denote the value of any physical quantity at node $\boldsymbol{x}_j$ by the subscript j. In Equation (7.35), the second term (quantity in curly brackets) can be replaced by the right hand side of Equation (7.33). Similarly, in Equation (7.36), the second term can be replaced by the right hand side of Equation (7.34). The modified discretisations now become

$$\nabla^2\psi_i \approx \sum_{j=1}^{n} w_{(i,j)}^{\nabla^2}\psi_j + \sum_{l=1}^{m} \widetilde{w}_{(i,l)}^{\nabla^2}(-\omega_{\eta_l}), \tag{7.37}$$

and

$$\nabla^2\omega_i \approx \sum_{j=1}^{n} w_{(i,j)}^{\nabla^2}\omega_j + \mathrm{Re}\sum_{l=1}^{m} \widetilde{w}_{(i,l)}^{\nabla^2}\left(u\frac{\partial\omega}{\partial x} + v\frac{\partial\omega}{\partial y}\right)_{\eta_l}. \tag{7.38}$$

We now return to the solution of the governing NS equations via a fixed point iteration scheme. Denoting the iteration number k with a superscript k on the physical variable, the governing equations at iteration $k+1$ for the node $\boldsymbol{x}_i$ are given by

$$\begin{aligned} \nabla^2\psi_i^{k+1} &= -\omega_i^k, \\ \nabla^2\omega_i^{k+1} &= \mathrm{Re}\left[\bar{u}_i\left(\frac{\partial\omega}{\partial x}\right)_i^{k+1} + \bar{v}_i\left(\frac{\partial\omega}{\partial y}\right)_i^{k+1}\right], \end{aligned} \tag{7.39}$$

where $\bar{u}_i$ and $\bar{v}_i$ are the current estimates of components of the velocity vector. Substituting the derived higher-order RBF-FD discretisations for Laplacian of streamfunction (Equation (7.37)) and vorticity (Equation (7.38)) in to Equation (7.39), we obtain

$$\sum_{j=1}^{n} w_{(i,j)}^{\nabla^2}\psi_j^{k+1} + \sum_{l=1}^{m} \widetilde{w}_{(i,l)}^{\nabla^2}\left(-\omega_{\eta_l}^k\right) = -\omega_i^k, \tag{7.40}$$

and

$$\sum_{j=1}^{n} w^{\nabla^2}_{(i,j)}\omega_j^{k+1} \quad +\mathrm{Re}\sum_{l=1}^{m}\widetilde{w}^{\nabla^2}_{(i,l)}\left[\bar{u}_{\eta_l}\left(\frac{\partial\omega}{\partial x}\right)^k_{\eta_l} + \bar{v}_{\eta_l}\left(\frac{\partial\omega}{\partial y}\right)^k_{\eta_l}\right]$$
$$-\mathrm{Re}\left[\bar{u}_i\sum_{j=1}^{n} w^{(x)}_{(i,j)}\omega_j^{k+1} + \bar{v}_i\sum_{j=1}^{n} w^{(y)}_{(i,j)}\omega_j^{k+1}\right] \quad = 0. \tag{7.41}$$

Note that in Equation (7.41), the vorticity gradients $\left(\frac{\partial\omega}{\partial x}, \frac{\partial\omega}{\partial y}\right)$ were discretised using the RBF-FD method.

The velocity components $(\bar{u}, \bar{v})$ in Equation (7.41) are obtained using the higher-order RBF-FD discretisations given by

$$\bar{u}_i \equiv \left.\frac{\partial\bar{\psi}}{\partial y}\right|_i \approx \sum_{j=1}^{n} w^{y}_{(i,j)}\bar{\psi}_j + \sum_{l=1}^{m}\widetilde{w}^{y}_{(i,l)}\left.\frac{\partial\bar{\psi}}{\partial y}\right|_{\eta_l}, \tag{7.42}$$

$$\bar{v}_i \equiv -\left.\frac{\partial\bar{\psi}}{\partial x}\right|_i \approx -\left[\sum_{j=1}^{n} w^{x}_{(i,j)}\bar{\psi}_j + \sum_{l=1}^{m}\widetilde{w}^{x}_{(i,l)}\left.\frac{\partial\bar{\psi}}{\partial x}\right|_{\eta_l}\right], \tag{7.43}$$

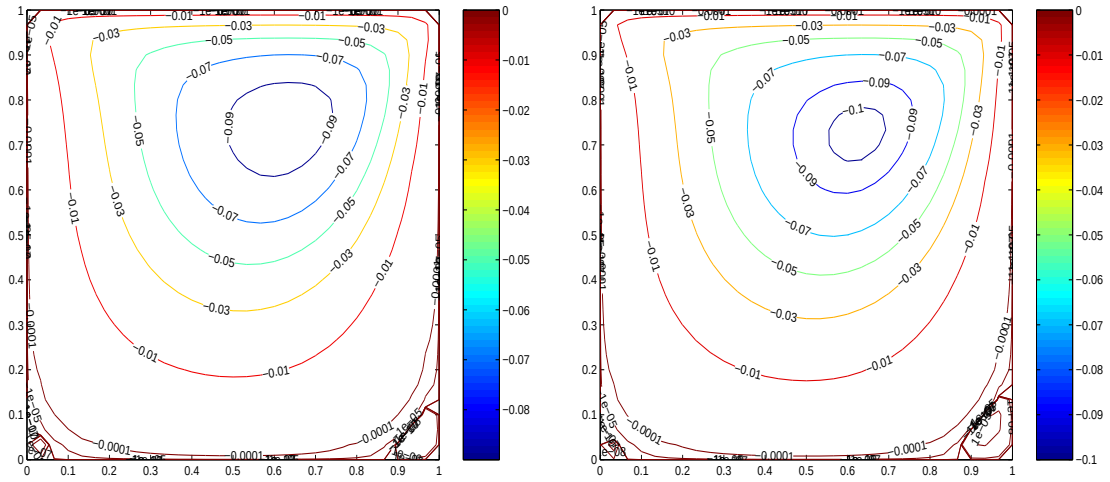
where $\bar{\psi}$ is the current estimate of the streamfunction.

The iteration procedure is explained for a problem with no-slip boundary conditions. Recall that the no-slip boundary condition consists of a Dirichlet and a Neumann condition for streamfunction at each boundary point, see Equation (7.9). Now, given an initial guess for streamfunction and vorticity, we solve the system of equations arising from satisfying Equation (7.40) at all interior nodes along with the Dirichlet boundary conditions for the streamfunction to obtain the new estimate for streamfunction $(\bar{\psi})$. To obtain the new velocity vector estimate, the system of equations arising from satisfying Equation (7.42)/ Equation (7.43) at all interior nodes is solved. Note that whenever the support point $\boldsymbol{x}_{\eta_l}$ for a node is on the boundary, the Neumann condition for streamfunction is used thus facilitating an easier implementation of no-slip boundary conditions. Next, the new estimate of vorticity on the boundary is obtained using the ghost node strategy proposed in Section 7.2. Once the velocity vector estimate is known, the linear system of equations arising from satisfying Equation (7.41) is solved with the Dirichlet vorticity conditions obtained using the ghost node strategy. Once the physical quantities are obtained, we advance to the next iteration. This procedure is repeated until convergence.

7.5.1 Numerical results

We now present numerical results obtained for the higher-order RBF-FD method for the steady incompressible NS equations. The formulation outlined in Section 7.5 is used to solve

the square lid-driven cavity flow. Figure 7.15 presents the streamfunction contours obtained for the cavity flow at Re = 100 with a uniform distribution of 31×31 nodes. The left subfigure solution is obtained using the RBF-FD method and the right subfigure is obtained using the higher-order RBF-FD method. From Figure 7.15, it can be clearly seen that the higher-order method captures the solution more accurately with a small number of 31×31 points.



a) RBF-FD b) Higher Order RBF-FD

FIGURE 7.15: Comparison of convergence behaviours of RBF-FD and higher-order RBF-FD for a model steady state convection-diffusion equation

Figure 7.16 estimates the performance of higher-order RBF-FD and RBF-FD in terms of accuracy. On the x-axis, the mesh spacing h is plotted on a log scale in the reverse direction. On the y-axis, the minimum value of streamfunction $\psi_{\min}$ in the whole domain (strength of the primary vortex) is plotted. The benchmark value obtained by Ghia *et al.* (1982) is shown as a horizontal dotted line in the figure. From Figure 7.16, it can clearly be observed that the higher-order method captures the true solution at considerably less points ($h \approx 0.033$) as compared to the original RBF-FD method.

7.6 Concluding remarks

The RBF-FD method is presented for solving incompressible Navier-Stokes equations. This method approximates the function derivatives at a node in terms of the function values on a scattered set of points present in support region of the node. The RBF-FD method uses local

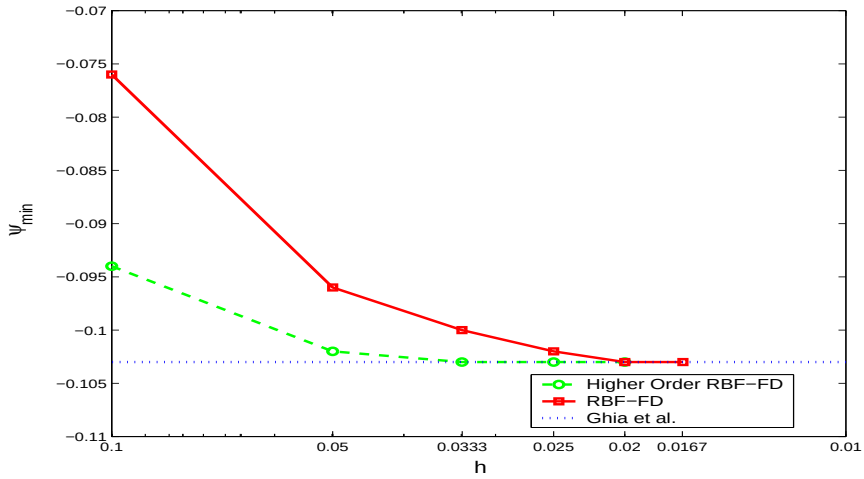


FIGURE 7.16: Convergence of $\psi_{\min}$ for higher-order RBF-FD and RBF-FD at Reynolds number 100

interpolation problems and hence generates sparse and well-conditioned matrices. It also has the property of decreased sensitivity with respect to shape parameter value in comparison with the RBF collocation method. A ghost node strategy employed for incorporating no-slip boundary conditions removes the limitations of having a locally orthogonal grid near the boundary and thus makes the method more suitable for complete random node discretisations. Numerical studies conducted on the driven cavity flow problems using the RBF-FD method show that this method achieves accurate results which are in good agreement with the benchmark results.

A higher-order RBF-FD method is explored for solving partial differential equations. The higher-order method is obtained by using Hermite RBF interpolation method to construct the function approximation at each node in the domain. A higher-order formulation for steady incompressible Navier-Stokes equations is presented. The accuracy of the higher-order method is investigated by solving for a model Poisson equation, convection diffusion equation and square lid-driven cavity flow. Numerical results obtained indicate that this method indeed is a higher-order method with a higher capability of spatial resolution with respect to the RBF-FD method.

Chapter 8

Conclusions and Future Areas of Research

This chapter concludes the thesis with a brief synopsis of the primary conclusions and contributions of the present research work. Some directions for future research are also outlined.

8.1 Research summary and contributions

The main focus of this thesis is to develop meshless methods for fluid dynamics problems using radial basis function collocation methods. These methods require only a scattered set of nodes or points in the domain instead of a mesh which is the case for traditional methods such as FD, FE or FV methods. Two methodologies for solving PDEs have been presented in this thesis. The first is the collocation and second one is using RBFs in a finite difference mode. Benchmark flow problems for the incompressible Navier Stokes equations have been solved using meshless methods developed on the basis of both the methodologies. The main conclusions and contributions made in this doctoral research are summarised below.

8.1.1 Comparisons of various RBFs for unsteady flow problems

We presented a detailed comparison on the performance of various RBFs when applied to solve the unsteady convection-diffusion equation. A symmetric RBF collocation method for time-dependent problems was proposed and comparisons were made with the existing unsymmetric RBF collocation method. Both global and compactly supported RBFs were used and the convergence behaviours of each RBF were investigated for three different Peclet

numbers. A residual based optimisation strategy was employed to determine the shape parameter value in the case of σ-tunable RBFs. Numerical results suggest that the presented meshless methods are capable of generating accurate results for low and medium Peclet numbers. At very high Peclet numbers, both the unsymmetric as well as the symmetric schemes are not able to capture the sharp discontinuity in the solution due to worsening of the condition numbers of the coefficient matrices. The discontinuity may be captured by adding artificial dissipation or by combining the RBF collocation methodology with the flux limiter schemes as in computational fluid dynamics literature.

8.1.2 RBFs & Domain decomposition methods

Overlapping Schwarz domain decomposition algorithms using RBFs were proposed for solution of unsteady linear PDEs. The proposed algorithms were compared with an existing multizone algorithm using RBFs (Wong *et al.*, 1999). It was shown that the proposed algorithms are much more computationally efficient than the multizone method. We also showed that the Schwarz schemes are much faster than the global RBF collocation method due to smaller matrices. Further, with increasing number of subdomains the proposed schemes are much faster with an acceptable loss of accuracy. Finally, these schemes reduce the ill-conditioning problem associated with the RBF collocation methods. All these features make the Schwarz overlapping schemes attractive for solving large scale problems. The proposed Schwarz schemes were also extended to nonlinear elliptic PDEs. The behaviour of the schemes for nonlinear problems was shown to be similar to that of the unsteady linear problem.

8.1.3 RBFs in finite difference mode

An alternative strategy to domain decomposition methodology was also pursued in order to develop meshless methods using RBFs for large scale problems. The basic idea is to generate function approximations on a cloud of nodes in the local support region/stencil of a node. This method results in sparse coefficient matrices and hence is suitable for solving large scale fluid flow problems. A novel shape parameter optimisation strategy was developed using statistical estimators like the leave-one-out criterion as the objective function to determine the optimal shape parameter value for each RBF-FD stencil. Numerical studies were conducted on model unsteady convection-diffusion equation to ascertain the efficiency of the RBF-FD method.

8.1.4 Meshless schemes for incompressible NS equations

Meshless methods based on global RBF collocation and RBF-FD were developed for solving steady and unsteady incompressible Navier-Stokes equations. A novel ghost node strategy was employed for satisfying the no-slip boundary conditions for both collocation and RBF-FD methods. For the RBF-FD method, this strategy enables the method to be suitable for complete random point discretisation in the domain. Benchmark test problems like the driven cavity flow in a square region, rectangular driven cavity with aspect ratio 2 and a backward facing step flow were solved using the developed meshless methods. Finally, a higher-order RBF-FD method was explored for solving the incompressible Navier Stokes equations. Numerical results suggest a high spatial resolution for the higher-order scheme for model convection-diffusion and NS equations.

8.2 Future areas of research

An outline for some directions for future research is presented below

- *Shape parameter tuning for collocation methods:*
  In this thesis, research conducted on the application of global RBF collocation methods for fluid flow problems show that these methods have the potential of obtaining very good accuracies with considerably less number of points as compared to traditional mesh based methods. However, the high convergence rates are subject to obtaining the optimal value of shape parameter (σ). An optimisation strategy based on residual error minimisation is proposed in this thesis. However, this strategy may be computationally expensive especially for unsteady flow problems, if one wants to obtain the optimal value of σ at each time step. Further research is required in this direction to develop more computationally efficient algorithms for tuning the shape parameter.

- *Higher order RBF-FD:*
  Better spatial resolution techniques using Hermite interpolation methods were explored for solving the NS equations in this thesis. Further work is required to extend the technique for unsteady NS equations and for compressible flows. Also, a variety of higher-order schemes may be developed for NS equations depending on the operator which is considered during Hermite interpolation. In this thesis, two examples of such discretisations were investigated. It remains to investigate other type of discretisations

and establish the computational efficiency of each of the discretisations.

- *Domain decomposition for NS*

  Domain decomposition methods for RBF collocation were developed for linear time-dependent problems and nonlinear elliptic PDEs. These domain decomposition methods can be further extended to solve nonlinear time dependent PDEs and the Navier-Stokes equations. These methods have the potential of high convergence rates provided the optimal value of shape parameter is used. Further, it would be interesting to apply these schemes for large-scale problems encountered in 3D fluid flow problems.

Appendix A

Derivatives of multiquadric RBFs

The analytical expressions for the derivatives of the multiquadric radial basis function are presented. The definition of the basis function used is given by

$$\phi(\|\boldsymbol{x}-\boldsymbol{c}\|)=\sqrt{(x-h)^2+(y-k)^2+\sigma^2}, \tag{A.1}$$

where $\boldsymbol{x}=(x,y)$ is the collocation point and $\boldsymbol{c}=(h,k)$ is the centre of the RBF and σ is the shape parameter. The partial derivatives for the multiquadric RBF are presented below.

$$\frac{\partial\phi}{\partial x} = \frac{x-h}{\sqrt{(x-h)^2+(y-k)^2+\sigma^2}}, \tag{A.2}$$

$$\frac{\partial\phi}{\partial y} = \frac{y-k}{\sqrt{(x-h)^2+(y-k)^2+\sigma^2}}, \tag{A.3}$$

$$\nabla^2\phi = \frac{(x-h)^2+(y-k)^2+2\sigma^2}{\left[(x-h)^2+(y-k)^2+\sigma^2\right]^{\frac{3}{2}}}, \tag{A.4}$$

$$\frac{\partial^3\phi}{\partial x^2\partial y} = \frac{(y-k)\left[2(x-h)^2-(y-k)^2-\sigma^2\right]}{\left[(x-h)^2+(y-k)^2+\sigma^2\right]^{\frac{5}{2}}}, \tag{A.5}$$

$$\frac{\partial^3\phi}{\partial x\partial y^2} = -\frac{(x-h)\left[(x-h)^2-2(y-k)^2+\sigma^2\right]}{\left[(x-h)^2+(y-k)^2+\sigma^2\right]^{\frac{5}{2}}}, \tag{A.6}$$

$$\begin{aligned}\nabla^2\nabla^2\phi = {} & \frac{15\left[(x-h)^2+(y-k)^2+2\sigma^2\right]\left[(x-h)^2+(y-k)^2\right]}{\left[(x-h)^2+(y-k)^2+\sigma^2\right]^{\frac{7}{2}}} \\ & -\frac{18\left((x-h)^2+(y-k)^2\right)+12\sigma^2}{\left[(x-h)^2+(y-k)^2+\sigma^2\right]^{\frac{5}{2}}} \\ & +\frac{4}{\left[(x-h)^2+(y-k)^2+\sigma^2\right]^{\frac{3}{2}}},\end{aligned} \tag{A.7}$$

where ∇^2 and $\nabla^2\nabla^2$ denote the Laplacian and biharmonic operators respectively.

Appendix B

Derivation of Leave-One-Out objective function for RBF-FD stencil

The leave-one-out cross validation estimator is given by

$$E(\sigma) = \sum_{i=1}^{N} \left(\bar{f}_i - f_i\right)^2, \tag{B.1}$$

where $\{\boldsymbol{x}_i, f_i\}_{i=1}^{N}$ is the observed data and $\bar{f}_i$ is the function value predicted at the i^{th} data point using multiquadric basis approximation based on the database that excludes the i^{th} data.

Let

$$e_i = \bar{f}_i - f_i. \tag{B.2}$$

Partition matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = \begin{bmatrix} \bar{\boldsymbol{A}} & \widetilde{\boldsymbol{A}}(:,i) \\ \widetilde{\boldsymbol{A}}^T(:,i) & A_{ii} \end{bmatrix}, \tag{B.3}$$

where $\widetilde{\boldsymbol{A}}(:,i)$ is the i^{th} column of the MQ coefficient matrix without the i^{th} row and the superscript T denotes the transpose. $\bar{\boldsymbol{A}}$ denotes the rest of matrix $\boldsymbol{A}$.

By definition of $\bar{f}_i$, it can be computed from

$$\bar{f}_i = \widetilde{\boldsymbol{A}}^T(:,i)\bar{\boldsymbol{\lambda}}, \tag{B.4}$$

where $\bar{\boldsymbol{\lambda}}$ is the solution of the following system of equations:

$$\bar{\boldsymbol{A}}\bar{\boldsymbol{\lambda}} = \widetilde{\boldsymbol{f}}. \tag{B.5}$$

Rewriting Equation (B.4) and Equation (B.5) as

$$\begin{aligned} \bar{\boldsymbol{A}}\boldsymbol{\lambda} + \widetilde{\boldsymbol{A}}(:,i) &= \bar{f} \\ \widetilde{\boldsymbol{A}}^T\bar{\boldsymbol{\lambda}} + A_{ii} - \mathbf{0} &= \widetilde{f}_i, \end{aligned} \tag{B.6}$$

Equation (B.6) can be expressed in matrix form as

$$\begin{bmatrix} \bar{\boldsymbol{A}} & \widetilde{\boldsymbol{A}}(:,i) \\ \widetilde{\boldsymbol{A}}^T(:,i) & A_{ii} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\lambda}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}} \\ \bar{f}_i \end{bmatrix}, \tag{B.7}$$

or

$$\boldsymbol{A} \begin{bmatrix} \bar{\boldsymbol{\lambda}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}} \\ \bar{f}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \bar{f}_i - f_i \end{bmatrix}. \tag{B.8}$$

Premultiplying both sides of Equation (B.8) by $\boldsymbol{B} = \boldsymbol{A}^{-1}$, we get

$$\begin{bmatrix} \bar{\boldsymbol{\lambda}} \\ \mathbf{0} \end{bmatrix} = \boldsymbol{B} \begin{bmatrix} \bar{\boldsymbol{f}} \\ \bar{f}_i \end{bmatrix} + \boldsymbol{B} \begin{bmatrix} \mathbf{0} \\ \bar{f}_i - f_i \end{bmatrix}. \tag{B.9}$$

Now the first term on the right hand side of Equation (B.9) is the partitioned MQ approximation coefficient of the whole data set. Hence,

$$\begin{bmatrix} \bar{\boldsymbol{\lambda}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{\lambda}} \\ \lambda_i \end{bmatrix} + \boldsymbol{B} \begin{bmatrix} \mathbf{0} \\ \bar{f}_i - f_i \end{bmatrix}. \tag{B.10}$$

From the last system of equations we have

$$E(\sigma) = \sum_{i=1}^{N} \left(\frac{\lambda_i}{B_{ii}} \right)^2. \tag{B.11}$$

Appendix C

Derivation for obtaining RBF-FD weights for a typical RBF-FD stencil

The derivation for obtaining the RBF-FD weights for a typical RBF-FD stencil is presented. We denote the unknown function by $u(\boldsymbol{x})$. Consider a three noded RBF-FD stencil given in Figure C.1. The RBF interpolant for the function $u(\boldsymbol{x})$ on the considered stencil is given by

$$\widehat{u}(\boldsymbol{x}) = \lambda_1\phi(\|\boldsymbol{x}-\boldsymbol{x}_1\|) + \lambda_2\phi(\|\boldsymbol{x}-\boldsymbol{x}_2\|) + \lambda_3\phi(\|\boldsymbol{x}-\boldsymbol{x}_3\|) + \beta, \tag{C.1}$$

where $\phi(\|\boldsymbol{x}-\boldsymbol{x}_1\|)$, $\phi(\|\boldsymbol{x}-\boldsymbol{x}_2\|)$ and $\lambda_3\phi(\|\boldsymbol{x}-\boldsymbol{x}_3\|)$ are the multiquadric RBFs centred on each of the nodes $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ nodes respectively and β is an unknown constant. λ_1, λ_2 and λ_3 are the uknown RBF coefficients.

By satisfying the conditions that the interpolant should pass through the function values

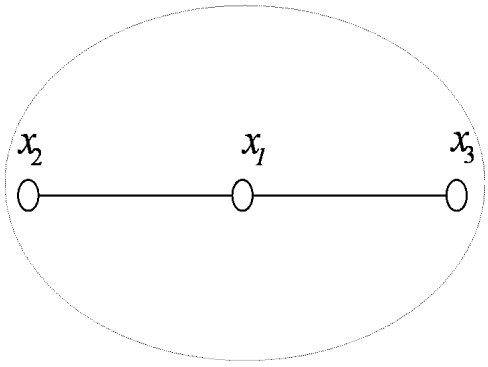


FIGURE C.1: Schematic diagram of a three noded RBF-FD stencil

at the nodes $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ and $\lambda_1 + \lambda_2 + \lambda_3 = 0$, we obtain the system of equations

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ 0 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & 1 \\ \phi_{21} & \phi_{22} & \phi_{23} & 1 \\ \phi_{31} & \phi_{32} & \phi_{33} & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \beta \end{bmatrix}, \tag{C.2}$$

where $\phi_{ij} = \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|),\ i, j = 1, 2, 3$. Let the matrix in Equation (C.2) be denoted by $\boldsymbol{A}$.

The unknown coefficients are given by

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \beta \end{bmatrix} = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} & \varphi_{14} \\ \varphi_{21} & \varphi_{22} & \varphi_{23} & \varphi_{24} \\ \varphi_{31} & \varphi_{32} & \varphi_{33} & \varphi_{34} \\ \varphi_{41} & \varphi_{42} & \varphi_{43} & \varphi_{44} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ 0 \end{bmatrix}, \tag{C.3}$$

where φ_{ij} denotes the ij^{th} element of $\boldsymbol{A}^{-1}$.

The goal of RBF-FD method is to express

$$\mathcal{L}u(\boldsymbol{x}_1) \approx w_1 u_1 + w_2 u_2 + w_3 u_3, \tag{C.4}$$

where w_1, w_2 and w_3 are the RBF-FD weights. Applying the operator $\mathcal{L}$ on the interpolant (Equation (C.1)), we obtain

$$\mathcal{L}u(\boldsymbol{x}_1) \approx \mathcal{L}\widehat{u}(\boldsymbol{x}_1) = \lambda_1 \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) + \lambda_2 \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|) + \lambda_3 \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_3\|). \tag{C.5}$$

Substituting the values of λ_1, λ_2 and λ_3 from Equation (C.3) in Equation (C.5) and rearranging in the form of Equation (C.4), we obtain

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \mu \end{bmatrix} = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} & \varphi_{14} \\ \varphi_{21} & \varphi_{22} & \varphi_{23} & \varphi_{24} \\ \varphi_{31} & \varphi_{32} & \varphi_{33} & \varphi_{34} \\ \varphi_{41} & \varphi_{42} & \varphi_{43} & \varphi_{44} \end{bmatrix}^T \begin{bmatrix} \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) \\ \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|) \\ \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_3\|) \\ 0 \end{bmatrix}. \tag{C.6}$$

Note that μ is a dummy constant which enforces that the stencil is exact for all constants. Equation (C.6) can be rewritten as

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & 1 \\ \phi_{21} & \phi_{22} & \phi_{23} & 1 \\ \phi_{31} & \phi_{32} & \phi_{33} & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \mu \end{bmatrix} = \begin{bmatrix} \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_1\|) \\ \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|) \\ \mathcal{L}\phi(\|\boldsymbol{x}_1 - \boldsymbol{x}_3\|) \\ 0 \end{bmatrix}. \tag{C.7}$$

References

Abgrall, R. (1994). On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*, **114**, 45–58.

Atluri, T. & Zhu, T. (1998). A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, **22**, 117–127.

Barragy, E. (1993). *Parallel Finite Element Methods and Iterative Solution Techniques for Viscous Incompressible Flows*. Ph.D. thesis, University of Texas at Austin.

Batchelor, G.K. (1967). *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, UK.

Batina, J.T. (1993). A gridless Euler/Navier-Stokes solution algorithm for complex-aircraft applications. Tech. Rep. AIAA-1993-333, AIAA.

Baxter, B. (1992). The asymptotic cardinal function of the multiquadric $\phi(r) = (r^2 + c^2)^{\frac{1}{2}}$ as $c \to \infty$. *Computers and Mathematics with Applications*, **24**, 1–6.

Beatson, R.K., Cherrie, J.B. & Mouat, C.T. (1999). Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration. *Advances in Computers and Mathematics*, **11**, 253–270.

Beatson, R.K., Light, W.A. & Billings, S. (2000). Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM Journal on Scientific Computing*, **22**, 1717–1740.

Belytschko, T., Lu, Y. & Gu, L. (1994). Element free Galerkin methods. *International Journal for Numerical Methods in Engineering*, **37**, 229–256.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. & P.Krysl (1996). Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, **139**, 49–74.

Boztosun, I. & Charafi, A. (2001). RBF-based meshless schemes for advection-diffusion problems. *Advances in Boundary Elements Technical Series*, **II**, 573–.

Boztosun, I. & Charafi, A. (2002). An analysis of the linear advection-diffusion equation using mesh-free and mesh-dependent methods. *Engineering Analysis with Boundary Elements*, **26**, 889–895.

Boztosun, I., Charafi, A., Zerroukat, M. & Djidjeli, K. (2002). Thin-plate spline radial basis function scheme for advection-diffusion problems. *Electronic J. Bound. Elem.*, **BETEQ**, 267–282.

Brebbia, C.A. (1978). *The Boundary Element Method for Engineers*. Pentech Press, London.

Brent, R.P. (1973). *Algorithms for Minimization without Derivatives*. Prentice Hall, Englewood Cliffs, NJ.

Bruneau, C.H. & Jouron, C. (1990). An efficient scheme for solving steady incompressible Navier-Stokes equations. *Journal of Computational Physics*, **89**, 389–413.

Buhmann, M.D. (1990). Mutivariate interpolation in odd dimensional Euclidean spaces using multiquadrics. *Constructive Approximation*, **6**, 21–34.

Buhmann, M.D. (2000). A new class of radial basis functions with compact support. *Mathematics of Computation*, **70**, 307–318.

Buhmann, M.D. & Micchelli, C.A. (1992). Multiquadric interpolation improved. *Computers and Mathematics with Applications*, **24**, 21–25.

Carlson, R.E. & Foley, T.A. (1991). The Parameter R^2 in multiquadric interpolation. *Computers and Mathematics with Applications*, **21**, 29–42.

Carr, J.C., Fright, W.R. & Beatson, R.K. (1997). Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions on Medical Imaging*, **16**, 96–107.

Cecil, T., Qian, J. & Osher, S. (2004). Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions. *Journal of Computational Physics*, **196**, 327–347.

Chen, C.S., Brebbia, C.A. & Power, H. (1998). Dual Reciprocity method for Helmholtz-type operators. *Boundary Elements*, **20**, 495–504.

Cheng, A.H.D., Golberg, M.A., Kansa, E.J. & Zammito, G. (2003). Exponential convergence and H-c multiquadric collocation method for partial differential equations. *Numerical Methods for Partial Differential Equations*, **19**, 571–594.

Chinchapatnam, P.P., Djidjeli, K. & Nair, P.B. (2005). Meshless domain decomposition schemes for nonlinear elliptic PDEs. In K.J. Bathe, ed., *Third MIT Conference on Computational Fluid and Solid Mechanics*, 1082–1086.

Chinchapatnam, P.P., Djidjeli, K. & Nair, P.B. (2006a). Domain decomposition for time-dependent problems using radial based meshless methods. *Numerical Methods for Partial Differential Equations*, Published Online: 25$^{\text{th}}$ May, 2006.

Chinchapatnam, P.P., Djidjeli, K. & Nair, P.B. (2006b). Unsymmetric and symmetric meshless schemes for the unsteady convection-diffusion equation. *Computer Methods in Applied Mechanics and Engineering*, **195**, 2432–2453.

Coleman, T.F. & Li, Y. (1994). On the convergence of reflective newton methods for large scale nonlinear minimization subject to bounds. *Mathematical Programming*, **67**, 189–224.

Coleman, T.F. & Li, Y. (1996). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, **6**, 418–445.

Collatz, L. (1960). *The Numerical Treatment of Differential Equations*. Springer, Berlin.

Ding, H., Shu, C. & Tang, D.B. (2005). Error estimates of local multiquadric-based differential quadrature (LMQDQ) method through numerical experiments. *International Journal for Numerical Methods in Engineering*, **63**, 1513–1529.

Dinh, H.Q., Turk, G. & Slabaugh, G. (2002). Reconstruction surfaces by volumetric regularization using radial basis functions. *IEEE Transactions on Pattern analysis and Machine Intelligence*, **24**, 1358–1371.

Djidjeli, K., Chinchapatnam, P.P., Nair, P.B. & Price, W.G. (2004). Global and compact meshless schemes for the unsteady convection-diffusion equation. In *Proceedings of the International Symposium on Health Care and Biomedical Research Interaction*, 8 pages on CDROM.

Driscoll, T.A. & Fornberg, B. (2002). Interpolation in the limit of increasingly flat radial basis functions. *Computers and Mathematics with Applications*, **43**, 413–422.

Duarte, C.A. (1995). A review of some meshless methods to solve partial differential equations. Tech. rep., Texas institute for Computational and Applied Mathematics, University of Texas at Austin.

Dubal, M.R. (1994). Domain decomposition and local refinement for multiquadric approximations. *Applied Scientific Computing*, **1**, 146–171.

Fasshauer, G.E. (1996). Solving partial differential equations by collocation with radial basis functions. In A.L. Méchauté, ed., *Proceedings of Chamonix*, 1–8, Vanderbilt university press, Nashville TN.

Fasshauer, G.E. (2005). Meshfree methods. In *Handbook of Theoretical and Computational Nanotechnology*, American Scientific publishers.

Ferziger, J.H. & Peric, M. (1999). *Computational Methods for Fluid Dynamics*. Springer Verlag.

Fornberg, B. (1996). *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, Cambridge.

Fornberg, B. (1998). Calculation of weights in finite difference formulas. *SIAM Reviews*, **40**, 685–691.

Franke, C. & Schaback, R. (1998). Convergence order estimates of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics*, **8**, 381–399.

Franke, R. (1982). Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, **38**, 181–200.

Fries, T.P. & Matthies, H.G. (2003). Classification and overview of meshfree methods. Tech. Rep. 03, Institute of Scientific Computing, Technical University Braunschweig, Brunswick, Germany.

Gartling, D.K. (1990). A test problem for outflow boundary conditions-flow over a backward-facing step. *International Journal for Numerical Methods in Fluids*, **11**, 953–967.

Ghia, U., Ghia, K.N. & Shin, C.T. (1982). High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, **48**, 387–411.

Gingold, R.A. & Moraghan, J.J. (1997). Smooth particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, **181**, 375–389.

Golberg, M.A., Chen, C.S. & Bowman, H. (1999). Some recent results and proposals for the use of radial basis functions in the BEM. *Engineering Analysis with Boundary Elements*, **23**, 285–296.

Griebel, M. & Schweitzer, M.A. (2000). A particle-partition of unity method for the solution of elliptic, parabolic and hyperbolic PDEs. *SIAM Journal on Scientific Computing*, **22**, 853–890.

Gupta, M.M. & Kalita, J.C. (2005). A new paradigm for solving Navier-Stokes equations: streamfunction-velocity formulation. *Journal of Computational Physics*, **207**, 52–68.

Hardy, R.L. (1971). Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, **176**, 1905–1915.

Hardy, R.L. (1990). Theory and application of the multiquadric-biharmonic method. *Computers and Mathematics with Applications*, **19**, 163–208.

Hirsch, C. (1991). *Numerical Computation of Internal and External Flows*. John Wiley, New York.

Hon, Y.C. & Mao, X.Z. (1997). A multiquadric interpolation method for solving initial value problems. *Scientific Computing*, **12**, 51–55.

Hon, Y.C. & Mao, X.Z. (1998). An efficient numerical scheme for Burger's equation. *Applied Mathematics and Computation*, **95**, 37–50.

Hon, Y.C. & Mao, X.Z. (1999). A radial basis function method for solving options pricing model. *Financial Engineering*, **8**, 31–50.

Hon, Y.C. & Schaback, R. (2001). On unsymmetric collocation by radial basis functions. *Applied Mathematics and Computation*, **119**, 177–186.

Hon, Y.C., Lu, M.W., Xue, W. & Zhu, Y. (1997). Multiquadric method for the numerical solution of a biphasic mixture model. *Applied Mathematics and Computation*, **88**, 153–175.

Jensen, V.G. (1959). Viscous flow round a sphere at low reynolds number (≤ 40). *Proceedings of Royal Society London: Series A*, **249**, 346–366.

Kansa, E.J. (1990a). Multiquadrics–a scattered data approximation scheme with applications to computation fluid dynamics-I. Surface approximations and partial derivatives estimates. *Computers and Mathematics with Applications*, **19(8/9)**, 127–145.

Kansa, E.J. (1990b). Multiquadrics–a scattered data approximation scheme with applications to computation fluid dynamics-II. Solution to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, **19(8/9)**, 147–161.

Kansa, E.J. (1999). Motivation for using radial basis functions to solve PDEs. Tech. rep., Lawrence Livermore Laboratory.

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

Kansa, E.J. & Hon, Y.C. (2000). Circumventing the ill-conditioning problem with multiquadric radial basis functions. *Computers and Mathematics with Applications*, **39**, 123–137.

Keane, A.J. & Nair, P.B. (2005). *Computational Approaches for Aerospace Design*. John Wiley.

Kulasegaram, S., Bonet, J., Lok, T.S. & Rodriguez-Paz, M. (2000). Corrected smooth particle hydrodynamics - a meshless method for computational mechanics. Tech. rep., ECCOMAS.

Larsson, E. & Fornberg, B. (2003). A numerical study of some radial basis functions based solution methods for elliptic PDEs. *Computers and Mathematics with Applications*, **46**, 891–902.

Lele, S.K. (1992). Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, **103**, 16–42.

Li, J. & Chen, C.S. (2003). Some observations on unsymmetric radial basis function collocation methods for convection-diffusion problems. *International Journal for Numerical Methods in Engineering*, **57**, 1085–1094.

Li, J. & Hon, Y.C. (2004). Domain decomposition for radial basis meshless methods. *Numerical Methods for Partial Differential Equations*, **20**, 450–462.

Liszka, T. & Orkisz, J. (1984). The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, **20**, 1594–1612.

Liu, W.K., Jun, S. & Belytschko, T. (1995). Reproducing Kernel Particle methods. *International Journal for Numerical Methods in Fluids*, **20**, 1081–1106.

Lu, Y., Belytschko, T. & Gu, L. (1994). A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, **113**, 397–414.

Madych, W.A. & Nelson, S.A. (1989). Error bounds for multiquadric interpolation. In C.K. Chui, L.L. Schumaker & J.D. Ward, eds., *Approximation Theory VI*, vol. 2, 413–416.

Madych, W.R. (1992). Miscellaneous error bounds for multiquadric and related interpolators. *Computers and Mathematics with Applications*, **24**, 121–138.

Mai-Duy, N. & Tanner, R.I. (2005). Solving high-order partial differential equations with indirect radial basis function networks. *International Journal for Numerical Methods in Engineering*, **63**, 1636–1654.

Mai-Duy, N. & Tran-Cong, T. (2001). Numerical solution of Navier-Stokes equations using multiquadric radial basis function networks. *International Journal for Numerical Methods in Fluids*, **37**, 65–86.

Melenk, J.M. & Babuska, I. (1996). The partition of unity finite element method:Basic theory and applications. Tech. Rep. 96-01, Texas institute for Computational and Applied Mathematics, University of Texas at Austin.

Micchelli, C.A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, **2**, 11–22.

Moraghan, J.J. (1982). Why particle methods work. *SIAM Journal on Scientific and Statistical Computing*, **3**, 422–433.

Moraghan, J.J. (1988). An introduction to SPH. *Computer Physics Communications*, **48**, 89–96.

Morton, K.W. (1995). *Numerical Solution of Convection-Diffusion Equation*. Chapman and Hall.

Nayroles, B., Touzat, G. & Villon, P. (1992). Generalizing the FEM: diffuse approximation and diffuse elements. *Computational Mechanics*, **10**, 307–318.

Onate, E., Idelsohn, S., Zienkiewicz, O.C., R.L.Taylor & Sacco, C. (1996). A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, **139**, 315–346.

Park, J. & Sandberg, I.W. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, **3**, 246–257.

Perrone, N. & Kao, R. (1975). A general finite difference method for arbitrary meshes. *Computers and Structures*, **5**, 45–47.

Powell, M.J.D. (1992). The theory of radial basis function approximation in 1990. In W. Light, ed., *Advances in numerical analysis, II: wavelets, subdivision algorithms and radial basis functions*, 105–210, Clarendon Press.

Power, H. & Barraco, V. (2002). A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations. *Computers and Mathematics with Applications*, **43**, 551–583.

Quarteroni, A. & Valli, A. (1999). *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press.

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

Roos, H.G., Stynes, M. & Tobiska, L. (1996). *Numerical Methods for Singularly Perturbed Differential Equations (Convection-Diffusion and Flow Problems)*. Springer Verlag, Berlin.

Schaback, R. (1995). Error estimates and condition numbers for radial basis interpolation. *Advances in Computational Mathematics*, **3**, 251–264.

Schaback, R. (1999). Improved error bounds for scattered data interpolation by radial basis functions. *Mathematics of Computation*, **68**, 201–206.

Schonauer, W. & Adolph, T. (2001). How we solve PDEs. *Journal of Computational and Applied Mathematics*, **131**, 473–492.

Sharan, M., Kansa, E.J. & Gupta, S. (1997). Applications of the multiquadric method for the solution of elliptic partial differential equations. *Applied Mathematics and Computation*, **84**, 275–302.

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

Smith, B.F., Bjorstad, P.E. & Gropp, W.D. (1996). *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press.

Sobol, I.M. (1979). On the systematic search in a hypercube. *SIAM Journal on Numerical Analysis*, **16**, 790–793.

Spotz, W.F. (1995). *High-order compact finite difference schemes for computational mechanics*. Ph.D. thesis, University of Texas at Austin.

Sun, X. (1994). Cardinal Hermite interpolation using positive definite functions. *Numerical Algorithms*, **7**, 253–268.

Tanaka, N. (1999). Development of a highly accurate interpolation method for mesh-free flow simulations I. integration of a gridless particle and CIP methods. *International Journal for Numerical Methods in Fluids*, **30**, 957–976.

Thom, A. (1928). An investigation of fluid flow in two dimensions. Tech. rep., Aerospace Research Center, United Kingdom.

Tolstykh, A.I. & Shirobokov, D.A. (2003). On using radial basis functions in a finite-difference mode with applications to elasticity problems. *Computational Mechanics*, **33**, 68–79.

Wang, B.P. (2004). Parameter optimization in multiquadric response surface approxiamtions. *Structures in Multidisciplinary Optimisation*, **26**, 219–223.

Wang, J.G. & Liu, G.R. (2002). A point interpolation method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, **54**, 1623–1648.

Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, **4**, 389–396.

Wendland, H. (1997). Sobolev-type error estimates for interpolation by radial basis functions. In A. Lemehaute, C. Rabut & L.L. Schumaker, eds., *Surface Fitting and Multiresolution Methods*, 337–344, Vanderbilt University Press, Nashville, TN.

Wong, A.S.M., Hon, Y.C., Li, T.S., Cheng, S.L. & Kansa, E.J. (1999). Multizone decomposition for simulation of time-dependent problems using the multiquadric scheme. *Computers and Mathematics with Applications*, **37**, 23–43.

Woods, L.C. (1954). A note on the numerical solution of fourth order differential equations. *Aeronautical Quarterly*, **5**, 176.

Wright, G.B. & Fornberg, B. (2006). Scattered node compact finite difference-type formulas generated from radial basis functions. *Journal of Computational Physics*, **212**, 99–123.

Wu, Z. (1995). Compactly supported positive definite radial functions. *Advances in Computational Mathematics*, **4**, 283–292.

Wu, Z. (1998). Solving PDE with radial basis function and the error estimation. In Z. Chen, Y. Li, C.A. Micchelli & Y. Xu, eds., *Advances in Computational Mathematics, Lecture Notes on Pure and Applied Mathematics*, vol. 202, Guang Zhou.

Wu, Z. & Schaback, R. (1993). Local error estimates for radial basis function interpolation of scattered data. *IMA Journal of Numerical Analysis*, **13**, 13–27.

Young, D.L., Jane, S.C., Lin, C.Y., Chiu, C.L. & Chen, K.C. (2004). Solutions of 2D and 3D Stokes laws using multiquadrics method. *Engineering Analysis with Boundary Elements*, **28**, 1233–1243.

Zerroukat, M., Power, H. & Chen, C.S. (1998). A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal for Numerical Methods in Engineering*, **42**, 1263.

Zerroukat, M., Djidjeli, K. & Charafi, A. (2000). Explicit and implicit meshless methods for linear advection-diffusion type partial differential equations. *International Journal for Numerical Methods in Engineering*, **48**, 19–35.

Zhou, X., Hon, Y.C. & Li, J. (2003). Overlapping domain decomposition method by radial basis functions. *Applied Numerical Mathematics*, **44**, 241–255.

Zienkiewicz, O.C. & Taylor, R.L. (2000). *Finite Element Method (5th edition) volume 3 - Fluid Dynamics*. Elsevier.