UNIVERSITY OF SOUTHERN QUEENSLAND
AUSTRALIA

COMPACT NON-SYMMETRIC AND SYMMETRIC STENCILS BASED ON INTEGRATED RADIAL BASIS FUNCTIONS FOR DIFFERENTIAL PROBLEMS

A thesis submitted by

Thi Thuy Van Le, **M.Math**

For the award of

Doctor of Philosophy

October 2019

Abstract

This PhD project is concerned with the development of compact local stencils based on integrated radial basis functions (IRBFs) for both spatial and temporal discretisations of partial differential equations (PDEs), and their applications in heat transfer and fluid flows. The proposed approximation stencils are effective and efficient since (i) Cartesian grids are employed to represent both rectangular and non-rectangular domains; (ii) high levels of accuracy of the solution and sparseness of the resultant algebraic system are achieved together; and (iii) time derivatives are discretised with high order approximation.

For spatial discretisation, a compact non-symmetric flat-IRBF stencil is developed. Significant improvements in the matrix condition number, solution accuracy and convergence rate with grid refinement over the usual approaches are obtained. Furthermore, IRBFs are used for Hermite interpolation in the solution of PDEs, resulting in symmetric stencils defined on structured/random nodes. For temporal discretisation, a compact IRBF stencil is proposed, where the time derivative is approximated in terms of, not only nodal function values at the current and previous time levels, but also nodal derivative values at the previous time level. When dealing with moving boundary problems (e.g. particulate suspensions and fluid structure interacting problems), to avoid the grid regeneration issue, an IRBF-based domain embedding method is also developed, where a geometrically-complex domain is extended to a larger, but simpler shaped domain, and a body force is introduced into the momentum equations to represent the moving boundaries.

The proposed methods are verified in the solution of differential problems defined on simply- and multiply-connected domains. Accurate results are achieved using relatively coarse Cartesian grids and relatively large time steps. The rate of convergence with grid refinement can be up to the order of about 5. Converged solutions are obtained in the simulation of highly nonlinear fluid flows and they are in good agreement with benchmark/well-known existing solutions.

Certification of Thesis

This Thesis is the work of Thi Thuy Van Le except where otherwise acknowledged, with the majority of the authorship of the papers presented as a Thesis by Publication undertaken by the Student. The work is original and has not previously been submitted for any other award, except where acknowledged.

Principal supervisor: Prof. Nam Mai-Duy

Associate Supervisor: Prof. Thanh Tran-Cong

Student and supervisors signatures of endorsement are held at the University.

Statement of Contribution

The following detail is the agreed share of contribution for candidate and co-authors in the presented publications in this thesis:

Journal Papers

- **Article I:** Le, T.T.V., Mai-Duy, N., Le-Cao, K. and Tran-Cong, T., (2019). A non-boundary-fitted-grid method, based on compact integrated-RBF stencils, for viscous flows. Submitted to International Journal for Numerical Methods in Fluids.
  All the authors contributed to the development of this paper: devise a numerical scheme, apply the proposed scheme for simulating engineering problems and discuss the obtained results. T.T.V Le made a significant contribution to all these parts. T.T.V Le mainly carried out the writing and implementation of computer codes and the writing of the manuscript (85%).

- **Article II:** Le, T.T.V., Mai-Duy, N., Le-Cao, K. and Tran-Cong, T., (2018). A time discretisation scheme based on integrated radial basis functions for heat transfer and fluid flow problems. Numerical Heat Transfer, Part B: Fundamentals **74**(2): 498-518.
  All the authors contributed to the development of this paper: devise a numerical scheme, apply the proposed scheme for simulating engineering problems and discuss the obtained results. T.T.V Le made a significant

contribution to all these parts. T.T.V Le mainly carried out the writing and implementation of computer codes and the writing of the manuscript (90%).

- **Article III:** Mai-Duy, N., Dalal, D., Le, T.T.V., Ngo-Cong, D. and Tran-Cong, T., (2018). A symmetric integrated radial basis function method for solving differential equations. Numerical Methods for Partial Differential Equations **34**(3): 959-981.

  All the authors contributed to the development of this paper: devise a numerical scheme, apply the proposed scheme for simulating engineering problems and discuss the obtained results. T.T.V Le made a significant contribution to all these parts. T.T.V Le mainly carried out the writing and implementation of computer codes and the writing of the manuscript (80%).

- **Article IV:** Mai-Duy, N., Le, T.T.V., Tien, C.M.T., Ngo-Cong, D. and Tran-Cong, T., (2017). Compact approximation stencils based on integrated flat radial basis functions. Engineering Analysis with Boundary Elements **74**: 79-87.

  All the authors contributed to the development of this paper: devise a numerical scheme, apply the proposed scheme for simulating engineering problems and discuss the obtained results. T.T.V Le made a significant contribution to all these parts. T.T.V Le mainly carried out the writing and implementation of computer codes and the writing of the manuscript (80%).

Conference Papers

- **Article I:** Le, T.T.V., Mai-Duy, N., and Tran-Cong, T. (2017). An effective two-point time discretisation scheme using integrated radial basis functions. International Conference on Advances in Computational Mechanics (ACOME), 2nd - 4th August 2017, Phu Quoc, Vietnam.

The paper was presented by T.T.V Le.

- **Article II:** Le, T.T.V., Mai-Duy, N., and Tran-Cong, T. (2018). An immersed-grid method for simulation of viscous flows, 13th International Conference on CFD in the Minerals and Process Industries, 4th- 8th December 2018, Melbourne, Australia.
  The paper was presented by T.T.V Le.

Acknowledgments

First of all, I would like to express my sincere appreciation to my supervisors, Prof. Nam Mai-Duy and Prof. Thanh Tran-Cong, for their invaluable supervision during my research and for their philosophical attitude which has encouraged my research interest. Unquestionably, without their continuing support and encourage, this thesis would not have been completed. Additionally, I wish to express my sincere thanks to Prof. David Buttsworth for his recommendation on the thesis proposal.

I would like to thank all co-authors of the articles for their feedback and cooperation, especially Dr. Khoa Le-Cao and Prof. Stephane Bordas for fruitful discussions. In addition, I would like to express my gratitude to Ms. Sandra Cochrane for proofreading this thesis.

Furthermore, I appreciatively acknowledge the financial support provided by the University of Southern Queensland (USQ), a USQ postgraduate award, and the Vietnam Academy of Science and Technology, which provided a scholarship supplement.

I would also like to dedicate this work to my parents. I owe my family a great deal for their assistance, understanding and love over the years and for their infinite reassurance during my academic pursuits.

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

1D-IRBF	One-Dimensional Indirect/Integrated Radial Basis Function Network
ADI	Alternating direction implicit
BFG/NBFG	Boundary-fitted-grid/Non-boundary-fitted-grid
CFD	Computational Fluid Dynamics
CRBF	Compact Indirect/Integrated Radial Basis Function Network
DRBF	Direct/Differentiated Radial Basis Function Network
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
GEs	Governing Equations
IRBF	Indirect/Integrated Radial Basis Function Network
MQ	MultiQuadric
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation

Chapter 1

Introduction

This chapter starts with an overview of numerical methods, including those based on radial basis functions (RBFs), for fluid dynamics. The motivation and objectives of this PhD project are then presented. Finally, the structure of the dissertation is outlined.

1.1 A brief review of numerical methods for fluid dynamics

Fluids exist all around us and we can see many different types of fluid flows. Most fluid flow problems cannot be solved in an analytic/exact manner, and one should rely on numerical methods to obtain their approximate solutions. Examples of numerical methods include the finite difference methods (FDMs), finite element methods (FEMs), finite volume methods (FVMs) and spectral methods. These methods, although apparently different, have some common features: (i) reducing the infinite degrees of freedom of a continuous system to a finite set and (ii) converting the governing equations (GEs) into sets of algebraic equations from which a computer solution to GEs can be obtained.

The process of discretising GEs is based on a set of elements in FEMs and FVMs, and a Cartesian grid in the FDMs and spectral methods. For the RBF methods and spectral methods, the problem domain can simply be represented by a set of unstructured and structured nodes, respectively.

1.1.1 Finite difference

FDM is a commonly used method for solving ordinary differential equations (ODEs) and partial differential equations (PDEs). In FDM, derivative terms of the field variables in differential equations (DEs) are approximated using Taylor series expansions. The set of DEs are discretised by means of point collocation (strong form) on grid nodes of orthogonal coordinate systems (e.g. Cartesian systems). This method was used in commercial software such as HEC-RAS which allows the user to perform one and two-dimensional steady/unsteady flow calculations, sediment transport/mobile bed computations and water temperature/water quality modelling (Brunner, 2004) and Mike 11 developed by DHI (2003) which is a software package for simulation flows, water quality and sediment transport in estuaries, rivers, irrigation channels and other water bodies. A FDM for solving initial boundary value problems in the form of a non-linear system of one dimensional (1D) differential equations describing shallow water flows was suggested by Rasulov et al. (2005). A FDM for a two-dimensional (2D) hybrid numerical model for sediment transport based on the lattice Boltzmann method was presented by Liu et al. (2015), where the GEs for water model and sediment transport are the shallow water equations and the advection-diffusion equations, respectively. In FDMs, the computational domain needs to be a rectangular one that is usually represented by a uniform grid. In the case of irregular domains, there might exist suitable coordinate transformations to achieve a rectangular computational domain and the GEs are then transformed into new forms that are usually more complicated.

1.1.2 Finite element

FEM discretises the domain into a set of small elements that are non-overlapping. In elements, shape functions (linear, polynomial, bilinear, piecewise, Hermite, etc.) are used to approximate the field variables. The PDEs are then transformed into integral equations. Different from FDM, here the PDEs are discretised by means of weak form. FEM is well suited to the simulation of PDEs defined on complex geometries. The method has increasingly been applied to surface water and soil transport problems (Carey, 1995). The major potentials and perspectives of applications of finite element analysis in solving shallow water wave equations were discussed in (Young, 1991). In this paper, FEM was also utilised in the hydraulic simulation during heavy rain in the Dan-Shui River system and the Te-Chi reservoir of Taiwan. A FEM based CFD software is TELEMAC (Galland et al., 1991). TELEMAC-2D for a 2D hydrodynamics module was used to analyse a shallow water model using the finite-element or finite-volume method with a mesh of triangular elements, and TELEMAC-3D for a 3D hydrodynamics module uses the same horizontally unstructured mesh as TELEMAC-2D but solves the Navier-Stokes equations, whether in hydrostatic or non-hydrostatic mode so allowing shorter waves than those in a shallow water context (where wavelengths are required to be at least 20 times the water depth). FEM for the analysis of transport and deposition of cohesive sediments in a two-dimensional flow field was developed by Ariathurai and Krone (1976).

1.1.3 Finite volume

FVM is widely used in CFD. Similar to FEM, the domain in the FVM is discretised into non-overlapping elements called finite volumes. The PDEs are then transformed into algebraic equations by integrating them over each volume. One then employs the Gauss's theorem to transform the volume integrals to surface integrals across the boundaries. This is to guarantee that the fluxes of the field variables across the boundary of the volumes are conserved. In practice, for fluid simulation, one might prefer FVM over FD because FVM has better physical conservation due to its law (to conserve the fluxes). Some of the popular CFD commercial codes using FVM are ANSYS Fluent (www.ansys.com) and Flow-3D (www.flow3d.com), to name a few. FVM is also employed in research of free surface and multiphase flows. For example, Toro (2001) developed a shock-capturing scheme based on FVM for the solution of free-surface flows which satisfy the shallow-water assumption; Zhang et al. (2014) set up a FVM based dynamically linked 1D and 2D hydrodynamic and sediment transport models for dam break flow and Wu et al. (2000) presented a three dimensional numerical model for calculating flow and sediment transport in open channels in which the convection-diffusion equation and flow equations were solved numerically with a finite-volume method on an adaptive, non-staggered grid.

1.1.4 Spectral methods

Like FDM, spectral methods are based on strong form. However, spectral methods make use of global approximation functions (eg. Fourier series or high order polynomials) to represent the variable fields rather than local approximation (e.g. Taylor series) in FDM. The spectral methods, possessing exponential convergence rate, are known as the most accurate approximation for smooth solutions. Spectral methods were, thus, applied widely in simulations in the early days of CFD when computer memory was expensive. The use of spectral methods for meteorological problems was first reported in (Silberman, 1954). In the field of CFD, spectral methods are continually being developed. For example, water wave simulation (Dalrymple et al., 1994), solving the shallow water equation on the sphere (Swarztrauber, 1996), simulation of mudslides (non-Newtonian fluid) due to periodic surface pressure (Huang et al., 2006), global weather modelling (Canuto et al., 2007), and investigating multiphase flows such as wave propagation in water and mud layers (Hejazi et al., 2014). For large scale simulations, a new domain decomposition Chebyshev method for parallel computation was also proposed in (Tsai et al., 2012).

1.1.5 Radial basis function methods

Radial basis function (RBF) network based methods are high-order discretisation methods (e.g. Kansa, 1990; Power and Barraco, 2002; Power et al., 2007; Šarler, 2005, 2009; Šarler et al., 2010; Divo and Kassab, 2007, 2008; Kosec and Šarler, 2008a,b; Mai-Duy and Tran-Cong, 2001). They have the characteristic of universal approximation, i.e. an arbitrary continuous function can be approximated to a guided degree of accuracy by raising the number of nodes (Poggio and Girosi, 1990; Park and Sandberg, 1991, 1993). They also require just a set of unstructured discrete points to support the approximation, which generally offers the benefit of being meshfree (Fasshauer, 2007). The methods are able to provide reliable simulations for highly nonlinear problems such as buoyancy flows with very thin boundary layers using relatively coarse grids/meshes. A detailed review of these methods can be found in (Liu and Gu, 2005; Liu, 2009). These books provide the fundamental knowledge of meshfree methods in much detail and classify them in different function approximation schemes such as meshfree methods based on the moving least squares approximation, meshfree methods based on the point interpolation method, meshfree methods based on the other interpolation schemes.

Unlike spectral methods, RBF methods can work with a set of unstructured discrete points, so these methods have emerged as a powerful approximation tool and become one of the main fields of research in numerical analysis (Haykin, 1999). Approximation schemes based on some RBFs, such as the multiquadric and Gaussian functions, can offer an exponential rate of convergence (Madych and Nelson, 1990). Wang and Liu (2002) presented a point interpolation meshless method based on RBFs in which the radial and polynomial basis functions are incorporated. Numerical results showed that the accuracy and convergence rate are high through patch tests and some problems in solid mechanics. RBF-based methods have been developed and applied to solve different types of differential problems encountered in applied mathematics, science and engineering (e.g. Zerroukat et al., 1998; Šarler et al., 2004; Šarler, 2005; Šarler et al., 2006; Šarler, 2009; Šarler et al., 2010; Divo and Kassab, 2005, 2006, 2007, 2008; Vertnik and Šarler, 2006; Vertnik et al., 2006; Yun-Xin and Yong-Ji, 2006; Kosec and Šarler, 2008a,b, 2009; Bernal and Kindelan, 2007; Khattak and Tirmizi, 2008; Zahab et al., 2009; Chen, Ye and Sun, 2010; Roque et al., 2010).

Global RBF/IRBF methods

For global RBF methods, all RBFs are employed for the function approximation at a node. Calculation of a field variable and its derivatives in terms of RBFs can be built through the differential process (DRBF) (Kansa, 1990) or the integral process (Mai-Duy and Tran-Cong, 2003). In the DRBF approach, the function to be approximated is first decomposed into RBFs, and its derivatives are then calculated by differentiation. In order to avoid the problem of reduced convergence rates caused by differentiation, the integral collocation formulation was proposed in (Mai-Duy and Tran-Cong, 2001). For the integral formulation, highest-order derivatives of the field variable in the partial differential equation (PDE) are discretised into RBFs and these RBFs are then integrated to obtain formulations for its lower-order derivatives and the variable itself (integrated RBFs (IRBFs)). RBF based methods are capable of producing a numerical solution that can converge at a high rate with respect to grid/mesh refinement. Nevertheless, their matrix is not as sparse as those produced by low-order methods. The system RBF matrix is thoroughly populated, and its condition number rises quickly with the growing number of nodes. Li and Hon (2004) showed that the system matrix becomes unsolvable when the entire number of collocation points are beyond 1000. Straightforward applications of RBFs for large-scale problems can thus be seen to be restricted. Several efforts to bypass these challenges/limitations have been suggested in the literature. They involve the use of domain decomposition and local approximation.

Domain decomposition RBF/IRBF

RBFs were additionally consolidated with domain decomposition (e.g. Li and Chen, 2003; Li and Hon, 2004; Divo and Kassab, 2006; Chinchapatnam et al., 2007a; Power et al., 2007). A region of interest is segmented into a set of subdomains reaching to a series of linked smaller subproblems. These subproblems can be calculated independently, and are appropriate for parallel computing. Li and Chen (2003) manipulated RBF collocation methods in conjunction with domain decomposition for working convection-diffusion problems at high Péclet numbers. Li and Hon (2004) showed both overlapping and non-overlapping domain decomposition schemes united with the meshless RBF method. Divo and Kassab (2006) developed a domain decomposition RBF based method for viscous incompressible fluid flow problems. Chinchapatnam et al. (2007a) introduced a numerical procedure, based on RBFs and Schwarz domain decomposition technique, to solve time-dependent problems. Power et al. (2007) examined the impact of the non-overlapping domain decomposition method on the symmetric RBF collocation method.

Local RBF/IRBF

It is recognised that the global RBF system matrix is thoroughly populated and its condition number increases quickly with an increasing number of nodes. To avoid this problem, several researchers developed local RBF methods, where only a small subregion, specifically the influence domain, is examined for the formation of the RBF approximations at a nodal point. Local methods lead to a sparse and better-conditioned system matrix. Wu and Liu (2003) suggested the local radial point interpolation method for incompressible flows. Liu et al. (2002) proposed a local radial point interpolation method based on local residual formulation using RBF and applied it for solving solid mechanic problems. Shu et al. (2003) combined local RBFs into the differential quadrature method to calculate incompressible flows. Divo and Kassab (2007) introduced a localised RBF meshless method for coupling viscous fluid flow and convective heat transfer problem. Šarler and Vertnik (2006) developed localised RBF approximations using a set of overlapping subregions. Vertnik and Šarler (2006) produced a meshless local RBF collocation method for convective-diffusive solid-liquid phase change problems. Skouras et al. (2011) coupled local multiquadrics RBFs with moving least square. Chinchapatnam et al. (2009) suggested a mesh-free computational method based on RBFs in a finite difference mode (RBF-FD). Li et al. (2011) employed Hardy multiquadrics for localised RBF expansions. On the other hand, local IRBF methods were developed by Mai-Duy and Tran-Cong (2009), Mai-Duy and Tran-Cong (2011), Thai-Quang et al. (2013). There is a notable improvement in the matrix condition number, but the accuracy of the solution is reduced. This can be overwhelmed by using compact approximations which include, not only the field variable but also its derivatives, (Wright and Fornberg, 2006; Mai-Duy and Tran-Cong, 2011; Tien et al., 2015). The sparseness of the system matrix and high level of the solution accuracy can be accomplished simultaneously with compact RBF approximation. The numerical practice has shown that the accuracy of an RBF solution is completely controlled by the shape parameter a_i. When a_i is grown (or ε_i decreased), the RBF solution becomes more accurate (Fornberg and Wright, 2004).

1.2 Research gaps

It can be seen that FVMs and FEMs need elements to support their approximations of the field variables and their discretisations of PDEs. Generating a mesh is the most costly and time-consuming part of the solution process:"A very time-consuming portion of overall computation is the mesh generation from CAD input data. Typically, more than 70 per cent of overall computing time is spent by mesh generators" (Griebel and Schweitzer, 2000). FDMs, which are based on Cartesian grids, are shown to produce accurate results for problems defined on rectangular domains. For non-rectangular domains, there is a need for conversion into rectangular domains through coordinate transformations, which may not be feasible in many cases. Furthermore, the approximations in the standard FEMs, FVMs and FDMs are based on low-order polynomials. As a result, their solutions are only first/second order accurate, and a fine mesh/grid is typically required to achieve a high level of accuracy. To avoid mesh generating process, one can employ meshless methods such as RBF methods. The use of RBFs for solving PDEs has received a great deal of attention in the last 30 years. RBF methods are meshless and have the ability to offer a high order accuracy. These attractive features are accompanied by densely populated matrices. Such systems are costly to solve and, more importantly, their condition number grows rapidly with the number of RBFs used, which limits the application of global RBF methods. Local RBF methods are then developed. Since only local regions are considered for the construction of the RBF approximations, the resultant algebraic systems are sparse. Although the matrix condition number is improved, the accuracy of the RBF solution is observed to deteriorate.

1.3 Motivation

In this PhD project, a new RBF approach is proposed to overcome the disadvantages of global and local RBF methods. In comparison with FVMs and FEMs, there is no finite-element mesh required. In comparison with FDMs and spectral methods, Cartesian grids are also employed, but the proposed approach can work for problems defined on non-rectangular domains.

Salient features of the proposed RBF approach include:

1. The RBF approximations are constructed through integration, which avoids the reduction in the convergence rate caused by differentiation and enhances the stability of a numerical solution

2. Not only function values, but also derivative values, are incorporated into the local RBF approximations (compact local approximations), which can achieve both high levels of accuracy and sparseness of the system matrix

3. Compact local IRBF stencils are employed in both spatial and temporal domains, allowing relatively low numbers of nodes and relatively large time steps to be employed for a given accuracy

4. Compact local IRBF stencils are incorporated into the domain embedding method, allowing multiply-connected domains to be converted into rectangular domains

5. An IRBF scheme for time derivative discretisation can achieve consistency in discretisation in both space and time. Thus, the numerical solutions are not limited by the accuracy of the time discretisation scheme (e.g. FD)

6. An advanced domain embedding scheme allows one to solve moving boundaries problems without regenerating grids.

1.4 Objectives of the thesis

The main objectives of this PhD research project are:

1. To develop a compact flat IRBF stencil for spatial discretisation, where large RBF widths are employed to improve the accuracy of the RBF solution for smooth problems without needing to use denser grids

2. To develop an IRBF-based Hermite interpolation for obtaining the resultant symmetric system, where one can save computer storage space and use a more efficient algebraic solver

3. To develop a compact IRBF stencil for time discretisation, where larger time steps can be used

4. To incorporate compact local IRBF stencils into the domain embedding method for an improved simulation of fluid flows defined on complex domains (e.g. multiply-connected domain)

5. To apply the proposed stencils to some practical heat transfer and fluid flow problems.

1.5 Outline of the thesis

The thesis has a total of seven chapters including this chapter (Introduction), and each chapter is self-explanatory. Brief descriptions of the remaining chapters are as follows.

Chapter 2 presents the fundamental background of the research comprising basic GEs, the formulation of IRBF approximation and a brief review of RBF numerical methods for solving fluid equations.

Chapter 3 reports several new techniques for constructing compact integrated RBF stencils based on extended precision, definite integrals, higher-order IRBFs and a minimum number of derivative equations to enhance their performance over a large value of RBF width. The proposed techniques are numerically verified in analytic tests.

Chapter 4 presents IRBFs employed for Hermite interpolation in the solution of differential equations which leads to a new meshless symmetric RBF method. Both global and local schemes are derived. The proposed method is applied to solve some elliptic boundary-value problems governed by Poisson and convection-diffusion equations. High levels of solution accuracy are obtained using relatively coarse discretisations.

Chapter 5 reports a new compact two-point approximation based on IRBFs for time discretisation, and a numerical scheme based on compact IRBF stencils only for solving time-dependent problems. The use of the RBF width as an additional parameter to enhance the approximation quality with respect to time is also explored. Various kinds of test problems of heat transfer and fluid flow are conducted to demonstrate the attractiveness of the present compact approximations.

Chapter 6 reports the incorporation of compact IRBF stencils into a domain embedding technique for numerical simulation of viscous flows in multiply-connected domains. The basic feature of the domain embedding technique is to extend the problem defined on a geometrically-complex domain to that on a larger, but more simply shaped, domain. Several kinds of linear and nonlinear test problems are conducted to demonstrate the effectiveness of the proposed technique.

Chapter 7 concludes the present research and suggests some possible future research developments.

Chapter 2

RBF methods for fluid flows

In this chapter, conservation laws for mass, momentum and energy over a control volume V are first summarised. The basis equations governing the motion of incompressible fluids are then reviewed. Finally, a brief discussion of RBF numerical methods for solving the fluid equations is given.

2.1 Fluid governing equations

2.1.1 Mass conservation

Mass conversation requires that the rate of change of mass within the control volume V is equal to the mass flux crossing its surface S

$$\frac{\partial}{\partial t}\int_V \rho dV = -\int_S \rho\mathbf{u}\cdot\mathbf{n}dS, \tag{2.1}$$

where ρ is fluid density, $\mathbf{u}$ the velocity vector, and $\mathbf{n}$ the unit normal vector. By applying Gauss's divergence theorem, this is stated as

$$\int_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u})\right] dV = 0, \tag{2.2}$$

Since equation (2.2) is valid for any size of volume V, the implication is that

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{2.3}$$

2.1.2 Momentum conservation

The law of momentum conservation is

$$\frac{\partial}{\partial t}\int_V \rho \mathbf{u} dV + \frac{\partial}{\partial t}\int_V \rho \mathbf{u}\mathbf{u} \cdot \mathbf{n} dS = \int_V \rho \mathbf{g} dV + \int_V \boldsymbol{\sigma} \cdot \mathbf{n} dS, \tag{2.4}$$

where $\mathbf{g}$ is a body acceleration field, $\boldsymbol{\sigma}$ the total stress; and the first term is the rate of accumulation of momentum in V and the second term is the flux of momentum across S. Their sum is equal to the rate of change in momentum due to body forces and surface stresses.

2.1.3 Energy conservation

The law of energy conservation is

$$\frac{\partial}{\partial t}\int_V \rho Q dV + \int_S \rho Q(\mathbf{n} \cdot \mathbf{u}) dS = -\int_S \mathbf{n} \cdot \mathbf{q} dS + \int_S \mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{u}) dS, \tag{2.5}$$

where the first term is the rate of accumulation of energy Q in V and the second term is the flux of energy across S. Their sum is equal to the flux of heat coming in through S and the rate of change in energy due to surface stresses; $\mathbf{q}$ the heat flux vector which is related to temperature gradients $\mathbf{q} = -k\nabla T$, where k

the thermal conductivity. One can employ the Gauss's divergence theorem to (2.5) in an arbitrary volume V to obtain

$$\frac{\partial}{\partial t}(\rho Q) + \nabla \cdot (\rho Q \mathbf{u}) = -\nabla \cdot \mathbf{q} + \nabla \cdot (\mathbf{q} \cdot \mathbf{u}). \tag{2.6}$$

2.1.4 Navier-Stokes equations

In this study, incompressible fluids are considered. Since their mass densities are constant, the term $\partial \rho / \partial t$ is always 0 regardless of whether the flow is steady or unsteady. The conservation law of mass (2.3) thus reduces to

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0. \tag{2.7}$$

Applying Gauss's divergence theorem to (2.4) obtains

$$\int_V \left[\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) \right] dV = \int_V (\rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma})\, dV. \tag{2.8}$$

Due to the arbitrariness of V, we have

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}. \tag{2.9}$$

Applying the continuity equation we have

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}, \tag{2.10}$$

or

$$\rho \frac{D\mathbf{u}}{Dt} = \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}, \tag{2.11}$$

where $D[.]/Dt$ the material derivative is defined as

$$\frac{D[\cdot]}{Dt} = \frac{\partial[\cdot]}{\partial t} + (\mathbf{u} \cdot \nabla)[\cdot].$$

For a Newtonian fluid, the stress tensor can be represented by

$$\boldsymbol{\sigma} = -p\mathbf{1} + 2\eta\mathbf{D}, \tag{2.12}$$

where p is the hydrodynamic pressure, $\mathbf{1}$ the unit tensor, η the viscosity and $\mathbf{D}$ the strain rate tensor

$$\mathbf{D} = \frac{1}{2}[\nabla\mathbf{u} + (\nabla\mathbf{u})^T]. \tag{2.13}$$

One can write the governing equations (2.7)-(2.11) in the following dimensionless form in 2D

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{2.14}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f_x, \tag{2.15}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + f_y, \tag{2.16}$$

where u, v are the components of $\mathbf{u}$, $\boldsymbol{f} = (f_x, f_y)$ the body force vector, and Re is the Reynolds number defined as $Re = UL/\nu$ in which ν is the kinematic viscosity, L a characteristic length, and U a characteristic velocity. Since there is no transport equation for the pressure in (2.14)-(2.16), velocity equations (2.15)-(2.16) need to be solved iteratively towards the satisfaction of the continuity condition (2.14). Several implementations have been reported, including the projection method (Chorin, 1968), the semi-implicit method for pressure-linked equations (SIMPLE) (Patankar and Spalding, 1972), and the pressure-implicit with splitting of operators (PISO) (Issa, 1986).

The pressure solver problem can be eliminated by introducing two new variables, namely the stream function (ψ) and the vorticity (ω),

$$\begin{aligned} u &= \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}, \\ \omega &= \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}, \end{aligned} \tag{2.17}$$

the primitive variable form, (2.14)-(2.16), reduces to

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega, \tag{2.18}$$

$$\frac{\partial \omega}{\partial t} + \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right). \tag{2.19}$$

In comparison with the $\mathbf{u} - p$ formulation, the continuity equation is satisfied automatically and the number of the field equations is reduced to two. The given velocity boundary conditions can be transformed into two boundary conditions on the stream function and its normal derivative

$$\psi = \gamma, \quad \frac{\partial \psi}{\partial n} = \xi,$$

where n is the direction normal to the boundary, and γ and ξ are prescribed functions. It can be seen that boundary conditions are over-prescribed for (2.18) and under-prescribed for (2.19). In practice, the boundary condition on ψ is used to solve (2.18), while the boundary condition on $\partial\psi/\partial n$ is employed to derive a computational vorticity boundary condition to solve (2.19). It is noted that the employment of the $\psi-\omega$ formulation are restricted to two-dimensional (2D) problems only.

2.2 Basic RBF formulations

With RBF approximation, calculation of a field variable and its derivatives can be built through the differential process (DRBF) or the integral process (IRBF). The integral formulation significantly improves the accuracy of the RBF scheme, especially for evaluating derivative functions (Mai-Duy and Tran-Cong, 2003). In the direct approach (DRBF), a function can be represented by RBFs as follows

$$f\left(\mathbf{x}\right)=\sum_{i=1}^{m}w_iG_i\left(\mathbf{x}\right), \tag{2.20}$$

where m is number of RBFs, $\{w_i\}_{i=1}^{m}$ are RBF weights to be determined, and $\{G_i\left(\mathbf{x}\right)\}_{i=1}^{m}$ are known RBFs. The RBFs can be written in a general form as $G_i\left(\mathbf{x}\right)=G_i\left(\|\mathbf{x}-\mathbf{c_i}\|\right)$, where $\|\cdot\|$ denotes the Euclidean norm and $\{\mathbf{c_i}\}_{i=1}^{m}$ is a set of RBF centres.

Several types of RBFs contain a free parameter. The following are some common types of RBFs that are of particular interest in the study of RBF methods:

1. Multiquadric function (MQ)

$$G_i\left(\mathbf{x}\right)=G_i\left(\|\mathbf{x}-\mathbf{c_i}\|\right)=\sqrt{r^2+{a_i}^2} \tag{2.21}$$

2. Inverse multiquadric function

$$G_i\left(\mathbf{x}\right)=G_i\left(\|\mathbf{x}-\mathbf{c_i}\|\right)=\frac{1}{\sqrt{r^2+{a_i}^2}} \tag{2.22}$$

3. Gaussian function

$$G_i\left(\mathbf{x}\right)=G_i\left(\|\mathbf{x}-\mathbf{c_i}\|\right)=\exp\left(-\frac{r^2}{{a_i}^2}\right) \tag{2.23}$$

where a_i is usually referred to as the width of the ith basis function and $r=\|\mathbf{x}-\mathbf{c_i}\|=\sqrt{\left(\mathbf{x}-\mathbf{c}_i\right)^T\left(\mathbf{x}-\mathbf{c}_i\right)}$.

One of the most widely used RBFs is the multiquadric function (2.21) which can be also written as

$$G_i(\mathbf{x})=\sqrt{\varepsilon_i(\mathbf{x}-\mathbf{c}_i)^T\left(\mathbf{x}-\mathbf{c}_i\right)+1}, \tag{2.24}$$

where ε_i is the shape parameter.

The MQ function becomes increasingly flat when $a_i \to \infty$ or $\varepsilon_i \to 0$.

In the DRBF approach, the function to be approximated is first decomposed into RBFs as in (2.20), and its derivatives are then calculated by differentiation

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i G_i(\mathbf{x}) \tag{2.25}$$

$$\frac{\partial^k f(\mathbf{x})}{\partial \alpha^k} = \sum_{i=1}^{m} w_{[\alpha]i} y_{[\alpha]i}^{(k)}(\mathbf{x}), \tag{2.26}$$

where α is a component of the independent spatial variable $\mathbf{x}$, the subscript $[\alpha]$ is used to differentiate the approximations with respect to each coordinate, k is the order of the derivatives of f, and $y_{[\alpha]i}^{(k)}(\mathbf{x}) = \partial^k G_i(\mathbf{x})/\partial \alpha^k$.

On the other hand, the integrated RBFs (IRBFs) method was proposed in (Mai-Duy and Tran-Cong, 2003), where the highest-order derivatives are first decomposed into RBFs and expressions for lower-order derivatives are then obtained through integration. Highest-order derivatives of the field variable f in the ordinary/partial differential equations (ODEs/PDEs) are decomposed into RBFs, from which expressions for lower-order derivatives and the variable itself are derived through integration

$$\frac{\partial^q f(\mathbf{x})}{\partial \alpha^q} = \sum_{i=1}^{m} w_{[\alpha]i} G_i(\mathbf{x}) = \sum_{i=1}^{m} w_{[\alpha]i} \mathcal{I}_{[\alpha]i}^{(q)}(\mathbf{x}), \tag{2.27}$$

$$\frac{\partial^{q-1} f(\mathbf{x})}{\partial \alpha^{q-1}} = \sum_{i=1}^{m} w_{[\alpha]i} \mathcal{I}_{[\alpha]i}^{(q-1)}(\mathbf{x}) + c_{[\alpha]1}, \tag{2.28}$$

$$\cdots \qquad \cdots \qquad \cdots$$

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_{[\alpha]i} \mathcal{I}_{[\alpha]i}^{(0)}(\mathbf{x}) + \frac{\alpha^{q-1}}{(q-1)!} c_{[\alpha]1} + \frac{\alpha^{q-2}}{(q-2)!} c_{[\alpha]2} + ... + c_{[\alpha]q}, \tag{2.29}$$

where m is the number of RBFs, $(w_{[\alpha]1}, w_{[\alpha]2}, ..., w_{[\alpha]m})$ the coefficients, $G_i(\mathbf{x})$ the RBF, $\mathcal{I}_{[\alpha]i}^{(q-1)}(\mathbf{x}) = \int \mathcal{I}_{[\alpha]i}^{(q)}(\mathbf{x}) d\alpha, ..., \mathcal{I}_{[\alpha]i}^{(0)}(\mathbf{x}) = \int \mathcal{I}_{[\alpha]i}^{(1)}(\mathbf{x}) d\alpha$, and $(c_{[\alpha]1}, ..., c_{[\alpha]q})$ the integration constants that are functions of variables other than α. Making use of (2.27)-(2.29) and point collocation, one can transform the ODE/PDE into a set of algebraic equations, from which the coefficients and integration constants can be acquired (Mai-Duy and Tran-Cong, 2003).

This approach has some strengths over the conventional approach (DRBF). It was developed to: (i) avoid the reduction in the convergence rate caused by differentiation, and (ii) make the numerical solution more stable.

Meshfree methods use both RBF and Moving Least Square (MLS) for interpolation/approximation purposes. In RBF based methods, since the shape functions have the Kronecker delta function property, the boundary conditions can be enforced in a direct manner (Liu and Gu, 2005). Here we made a comparison between the IRBF interpolation method and the MLS method using Gauss shape functions. Considering a simple function $f(x) = \sin(x)$, its first derivative $f'(x) = \cos(x)$ and second derivative $f''(x) = -\sin(x)$. The domain of interest is of $[0, 10]$. Number of nodes for both IRBF and MLS are $\{11, 21, 31, \cdots, 411\}$. In Figure 2.1, it can be seen that IRBF technique provides more accurate results for function approximation as well as its first and second derivatives. The convergence rate of IRBF method is $O(h^{3.54})$ and that of MSL is $O(h^{1.42})$ for the case of function approximation (Figure 2.1a). Similarly, Figure 2.1b and 2.1c show that the convergence rate of IRBF method is $O(h^{3.16})$ and $O(h^{1.68})$ and that of MSL is $O(h^{1.09})$ and $O(h^{0.81})$ for the case of first and second derivative approximations, respectively. Thus, the integral RBF methods are much more accurate than the methods using MLS shape functions in term of function and derivative approximations.

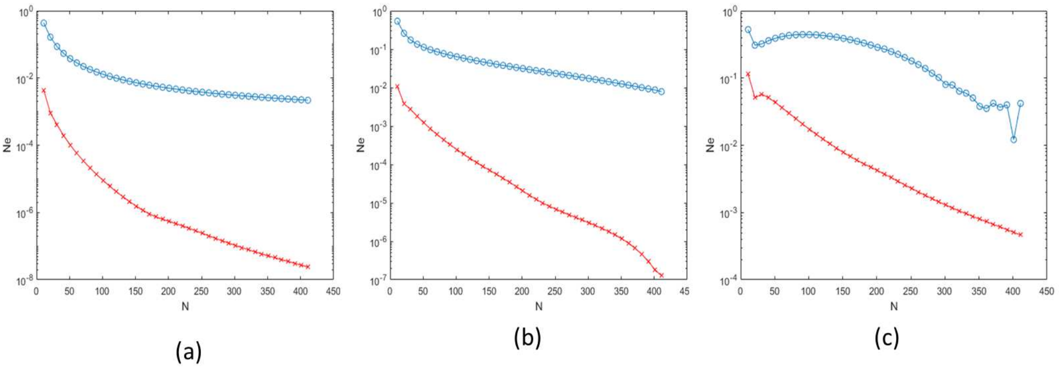


Figure 2.1: Function (a), first (b) and second (c) derivative approximation, double precision: Comparison of accuracy between the IRBF scheme and the MLS method

2.3 Review of RBF discretisations of the Navier-Stokes equations

As stated above, the basic equations governing the motion of a fluid can be written in different dependent variables, including the velocity - pressure ($u - v - p$) formulation, the stream function - vorticity ($\psi - \omega$) formulation and the stream function (ψ) formulation. The last two formulations are limited in two dimensional (2D) problems. Special attention should be given to the handling of the vorticity boundary condition for the $\psi - \omega$ formulation and the double boundary conditions as well as high-order derivatives including the cross derivatives for the ψ formulation.

Furthermore, in some cases, the pressure field needs be computed after solving the governing equations, which is generally regarded as a complicated process. In the case of multiply-connected domains, an added difficulty is that the stream-function variable generally has different and unknown values, on separate boundaries. On the other hand, the $u - v - p$ formulation is able to work for two- and three-dimension flows in a similar manner. One main concern is that there are no explicit transport equations and boundary conditions for the pressure variable. The resultant algebraic system could be solved iteratively where the pressure value is obtained using the continuity equation.

2.3.1 RBF discretisations of the stream function and stream function- vorticity formulation

With the introduction of the stream function variable, the pressure variable does not have to be considered, resulting in an easy implementation. However, several issues arise, to which special attention should be paid. For example, in the stream function approach, one has to cope with fourth-order derivatives and double boundary conditions. Fourth-order systems are known to have higher matrix condition numbers than first- and second-order systems. Errors for approximating higher-order derivatives are generally larger. In the implementation of double boundary conditions, special treatments are required because of two values given at a boundary point. To avoid the noise when approximating the high order derivative, IRBF methods have been employed in (Mai-Duy and Tran-Cong, 2001; Mai-Duy et al., 2008; Mai-Duy and Tran-Cong, 2008; Le-Cao et al., 2010). For the stream function - vorticity formulation, one has to derive computational boundary conditions for the transport vorticity equation. The boundary vorticity values are defined through the Poisson's equation, which needs to be solved discretely on the boundaries. The stream function - vorticity approach requires approximations for derivatives of an order up to 2 (instead of 4), leading to a significant improvement in the matrix condition number over the stream function approach. Le-Cao et al. (2009) proposed an IRBF method to solve the boundary problems for vorticity in a Cartesian grid for the stream function-vorticity formulation. This feature is very attractive when dealing with flows with a fine structure as a large number of nodes is usually required for an accurate simulation. Yun-Xin and Yong-Ji (2006) applied a RBF meshless scheme for unsteady vorticity-stream function formulation. In (Wu and Liu, 2003), a local radial point interpolation method (LRPIM) was adopted to simulate the two-dimensional natural convection problems with different geometries of domains. The numerical results showed that the accuracy achieved by the LRPIM method is considerably higher than that of the FD method. Chinchapatnam et al. (2007b) discretised the stream function formulation with RBF, and employed a trust-region method for solving the nonlinear problem. The no-slip boundary conditions are imposed using ghost nodes. Kim et al. (2007) introduced a meshfree point collocation method for the stream- vorticity formulation. The vorticity boundary condition was approximated linearly with the boundary velocity and the stream function. In (Fan et al., 2013), the local RBF collocation method was applied for solving the double-diffusive natural convection in porous media. The local RBF-based differential quadrature method was proposed by Shu et al. (2003), Shu et al. (2005) was employed for incompressible Navier-Stokes equations. In this method, the weighting coefficients were calculated by the RBFs rather than high order polynomials as the test functions. Wang et al. (2015a) employed a local RBF method for both velocity-pressure and stream function-vorticity formulations. The distributed nodes used to store the variables were obtained by the philosophy of an unstructured mesh.

2.3.2 RBF discretisations of the velocity-pressure formulation

Since there is no transport equation for the pressure in (2.14)-(2.16), velocity equations (2.15)-(2.16) need be solved iteratively towards the satisfaction of the continuity condition (2.14). Several implementations were reported, including the projection method (Chorin, 1968), the semi-implicit method for pressure-linked equations (SIMPLE) (Patankar and Spalding, 1972), and the pressure-implicit with splitting of operators (PISO) (Issa, 1986). Vertnik and Šarler (2006) developed a meshless local RBF collocation method for convective-diffusive solid-liquid phase change problems. Divo and Kassab (2007) presented a localised RBF meshless method for coupled viscous fluid flow and convective heat transfer problems. Chinchapatnam et al. (2009) proposed a mesh-free computational method based on RBF in a FD mode (RBF-FD). Skouras et al. (2011) coupled local multiquadric RBFs with moving least square (MLS). Local IRBF methods for velocity-pressure formulation are also developed (Thai-Quang et al., 2012). Flyer and Wright (2009) reported a numerical shallow water model on the sphere using RBF spatial discretization. Divo and Kassab (2007) applied the method to viscous fluid flow and conjugate heat transfer (CHT) modelling. The incompressible Navier-Stokes is time marched using a Helmholtz potential decomposition for the velocity field. In (Divo and Kassab, 2008), a localised RBF collocation meshless method is produced for natural convection heat transfer problems in completely viscous fluid flow. The extension method is based on the localized collocation of polynomial-augmented Hardy multiquadric RBF. In (Šarler et al., 2004), the authors represented the solution of a steady state natural convection problem in porous media with the RBF collocation method (RBFCM). The solution is expressed in primitive variables and required iterative treatment of pressure, and pressure correction. Thai-Quang et al. (2013) presented a high-order approximation scheme based on compact integrated RBF (CIRBF) stencils and second-order Adams Bashforth/Crank Nicolson algorithms for solving time-dependent problems such as torsionally oscillating lid-driven cavity flows. Demirkaya et al. (2008) reported an RBF approach with a direct technique for solving pressure fields instead of using an iterative algorithm for the primitive variables by using the Levenberga Marquardt method. Mramor et al. (2013) solved a more complex problem in which natural convection was influenced by a magnetic field. The fractional step method was also employed to link the pressure and velocity fields. The mass, momentum, energy and induction equations were couple into one system and then solved by a local RBF collocation method. Low Reynolds number, creeping flow problems were solved by a global meshless collocation particular solution method in (Bustamante et al., 2013). In this approach, the continuity equation is not explicitly imposed for pressure computing. The velocity components and the pressure are approximated by a linear superposition of particular solutions of the non-homogeneous Stokes. Waters and Pepper (2015) made a comparison between global and localized RBF meshless methods to solve viscous fluid flows with heat transfer. Numerical results show that the local approach is much more efficient but has the same order of accuracy as the global approaches.

2.4 Concluding remarks

This chapter has provided the elemental background of the research topic comprising basis governing equations of fluid dynamics, the IRBF formulation and a brief review of RBF methods. In general, each numerical method has some advantages and disadvantages. Therefore, this research is a further development of IRBF approach combining a high level of accuracy of IRBF and sparseness system of local approximation.

Chapter 3

Compact non-symmetric IRBF stencils for spatial approximations

For smooth problems, a high level of accuracy occurs at large value RBF width, however, corresponding RBF matrices become ill-conditioned. In this chapter, very large values of the RBF width are utilised with some special treatments to handle the interpolation matrix condition number. These effective treatments will be further discussed in the section on methodology. This chapter presents improved ways of constructing compact integrated RBF (CIRBF) stencils, based on extended precision, definite integrals, higher-order IRBFs and a minimum number of derivative equations to enhance performance over large values of the RBF width. The proposed approaches are numerically verified through second-order linear differential equations in one and two variables. Significant improvements in the matrix condition number, solution accuracy and convergence rate with grid refinement over the usual approaches are achieved.

3.1 Introduction

Several types of RBFs contain a free parameter. This type can exhibit an exponential rate of convergence with the number of RBFs and the RBF's width (Madych, 1992). One of the most widely used RBFs is the MQ function (2.21) or (2.24). The MQ function becomes increasingly flat when $a_i \to \infty$ or $\epsilon_i \to 0$. When all RBFs are employed for the approximation at a point, the RBF method is regarded as a global method. It is easy to implement global RBF methods since no mesh (i.e. no connection between nodes) is involved. A highly accurate solution is typically obtained. Furthermore, the system matrix is fully populated and as a result, only a relatively low number of nodes can be employed in practice. Global approximations can work with small values of a_i only, typically the minimum distance between the ith RBF and its neighbours.

When only a few RBFs are activated for the approximation at a point (local approximation), there is a significant improvement in the matrix condition number but the solution accuracy is significantly reduced. The latter can be overcome by using compact approximations, where the approximation involves nodal values of not only the field variable, but also its derivatives (Tolstykh and Shirobokov, 2003, 2005; Wright and Fornberg, 2006; Mai-Duy and Tran-Cong, 2011; Tien et al., 2015). With compact RBF approximations, high levels of the solution accuracy and sparseness of the system matrix can be achieved together. They are capable of providing a very efficient solution to a differential problem. In contrast to global RBF methods, larger values of a_i can be employed here. It was shown in (Fornberg and Wright, 2004; Fornberg and Flyer, 2011) that the RBF approximation is more accurate when a_i is increased (or ϵ_i is reduced) and the most accurate approximation occurs before a_i approaches infinity (or $\epsilon_i \to 0$). Furthermore, in the limit of $\epsilon_i \to 0$, the RBF approximation for a set of centres in one dimension reduces to the Lagrange interpolating polynomial on that set of nodes (Driscoll and Fornberg, 2002). Numerical experiments indicated that the interpolation matrices for local compact RBF stencils at large values of the RBF width are ill-conditioned and special treatments are needed. Effective treatments for compact RBF Hermite interpolation schemes (differentiated) were reported in (Wright and Fornberg, 2006), where the Contour-Pade algorithm is employed. This work presents several simple but effective approaches to extend the working range of a_i for compact integrated RBF approximations.

The chapter is organised as follows. A brief overview of CIRBF stencils is given in Section 3.2. In Section 3.3, some numerical investigations are conducted to identify numerical issues due to the use of large values of a_i. In Section 3.4, improved constructions for CIRBF stencils to extend the working range of a_i are presented and then numerically verified in analytic tests. Section 3.5 gives some concluding remarks.

3.2 Compact local IRBF stencils

Consider a 3-point stencil $[x_1, x_2, x_3]$. On the stencil, the second derivative of the dependent variable f is decomposed into

$$\frac{d^2 f(x)}{dx^2} = \sum_{i=1}^{3} w_i G_i(x) \tag{3.1}$$

where $\{G_i(x)\}_{i=1}^3$ is the set of RBFs and $\{w_i\}_{i=1}^3$ the set of weights to be found. In one dimension, the MQ function takes the form $G_i(x) = \sqrt{(x-c_i)^2 + a_i^2}$. We choose the width according to $a_i = \beta d_i$, where β is a scalar and d_i is the smallest distance between c_i and its neighbours.

Its first derivative and function are then obtained through integration

$$\frac{df(x)}{dx} = \sum_{i=1}^{3} w_i \mathcal{I}_i^{(1)}(x) + C_1 \tag{3.2}$$

$$f(x) = \sum_{i=1}^{3} w_i \mathcal{I}_i^{(0)}(x) + C_1 x + C_2 \tag{3.3}$$

where $\mathcal{I}_i^{(1)}(x) = \int G_i(x)dx$ and $\mathcal{I}_i^{(0)}(x) = \int \mathcal{I}_i^{(1)}(x)dx$ are integrated basis functions and C_1 and C_2 the constants of integration.

For compact approximations, nodal values of the derivative (or the differential equation) at the side nodes of the stencil are also incorporated into the process of converting the RBF space into the physical space. Assuming that the differential equation takes the form $d^2f(x)/dx^2 = f(x)$ ($f(x)$ is a prescribed function), the mapping can be constructed as

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \frac{d^2f_1}{dx^2} \\ \frac{d^2f_3}{dx^2} \end{pmatrix} = \underbrace{\begin{bmatrix} \mathcal{I}_1^{(0)}(x_1), & \mathcal{I}_2^{(0)}(x_1), & \mathcal{I}_3^{(0)}(x_1), & x_1, & 1 \\ \mathcal{I}_1^{(0)}(x_2), & \mathcal{I}_2^{(0)}(x_2), & \mathcal{I}_3^{(0)}(x_2), & x_2, & 1 \\ \mathcal{I}_1^{(0)}(x_3), & \mathcal{I}_2^{(0)}(x_3), & \mathcal{I}_3^{(0)}(x_3), & x_3, & 1 \\ G_1(x_1), & G_2(x_1), & G_3(x_1), & 0, & 0 \\ G_1(x_3), & G_2(x_3), & G_3(x_3), & 0, & 0 \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} \qquad (3.4)$$

where $\mathcal{C}$ is a 5×5 matrix that will hereafter be called the conversion matrix. Solving (3.4) leads to

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \frac{d^2f_1}{dx^2} \\ \frac{d^2f_3}{dx^2} \end{pmatrix} \qquad (3.5)$$

The second derivative of function f at the middle node is thus computed as

$$\frac{d^2f_2}{dx^2} = [G_1(x_2), G_2(x_2), G_3(x_2), 0, 0]\, \mathcal{C}^{-1} \left(f_1, f_2, f_3, \frac{d^2f_1}{dx^2}, \frac{d^2f_3}{dx^2} \right)^T \qquad (3.6)$$

or

$$\frac{d^2f_2}{dx^2} = \eta_1 f_1 + \eta_2 f_2 + \eta_3 f_3 + \eta_4 \frac{d^2f_1}{dx^2} + \eta_5 \frac{d^2f_3}{dx^2} \qquad (3.7)$$

where $d^2f_1/dx^2 = f(x_1)$, $d^2f_3/dx^2 = f(x_3)$ and $\{\eta_i\}_{i=1}^5$ are known values. In the case of Dirichlet boundary conditions and the domain represented by a set of N nodes, the collocation of the differential equation at the interior nodes results in the following system

$$\mathcal{A}\overrightarrow{f} = \overrightarrow{b} \tag{3.8}$$

where $\mathcal{A}$ is the system matrix of dimensions $(N-2)\times(N-2)$, $\overrightarrow{f}$ the vector consisting of values of f at interior nodes and $\overrightarrow{b}$ the vector formed by the RHS of the differential equation and the boundary conditions. Like the central finite-difference method, the structure of $\mathcal{A}$ is tri-diagonal and the system can be efficiently solved for the nodal variable values.

3.3 Numerical investigation

We apply the CIRBF solution procedure to the following second-order ODE

$$\frac{d^2f}{dx^2} = -\exp(-5x)\left(9975\sin(100x) + 1000\cos(100x)\right), \quad 0 \leq x \leq 1 \tag{3.9}$$

subject to Dirichlet boundary conditions. The exact solution can be verified to be

$$f_e(x) = \sin(100x)\exp(-5x) \tag{3.10}$$

and is displayed in Figure 3.1.

Two 3-point stencils, IRBF and compact IRBF (CIRBF), are implemented. The two system matrices have the same structure (tri-diagonal) but, as shown in Figure 3.2, the latter is much more accurate than the former. The RBF solution converges as $O(h^{4.79})$ for CIRBF and $O(h^{1.95})$ for IRBF, indicating that the inclusion of nodal second derivative values significantly enhances the performance of local IRBF stencils.

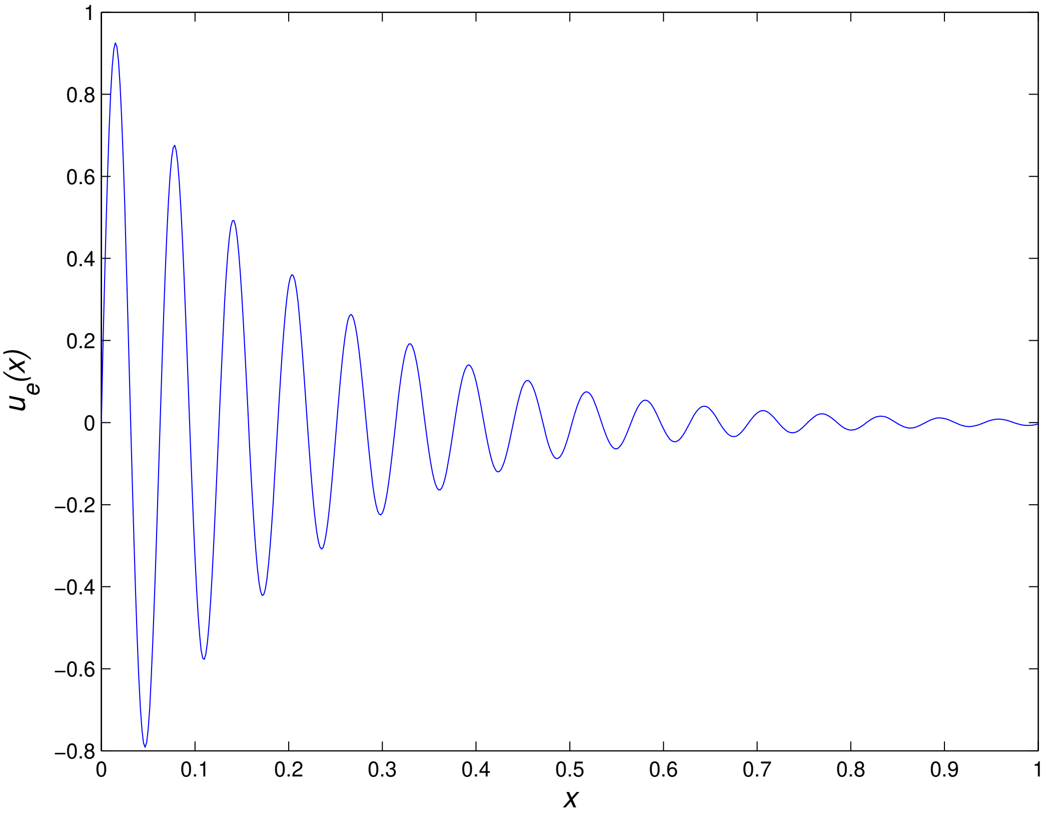


Figure 3.1: Second-order ODE, $f_e(x) = \sin(100x)\exp(-5x)$: Exact solution. The function is smooth and varies significantly over the domain. Such a variation requires a relatively large number of nodes for an accurate interpolation.

Figure 3.3 shows variations in the condition number of the conversion and system matrices against the MQ width represented by β for a fixed grid size ($N_x = 1001$). For the system matrix $\mathcal{A}$, the condition number is rather low ($O(10^5)$) and it has similar values over a wide range of β. In contrast, the condition number of the conversion matrix $\mathcal{C}$ grows fast at a rate of 4.5 and the matrix becomes ill-conditioned at large values of β. Therefore, in using CIRBF stencils, attention should be paid to the handling of matrix C resulting from flat MQ functions.

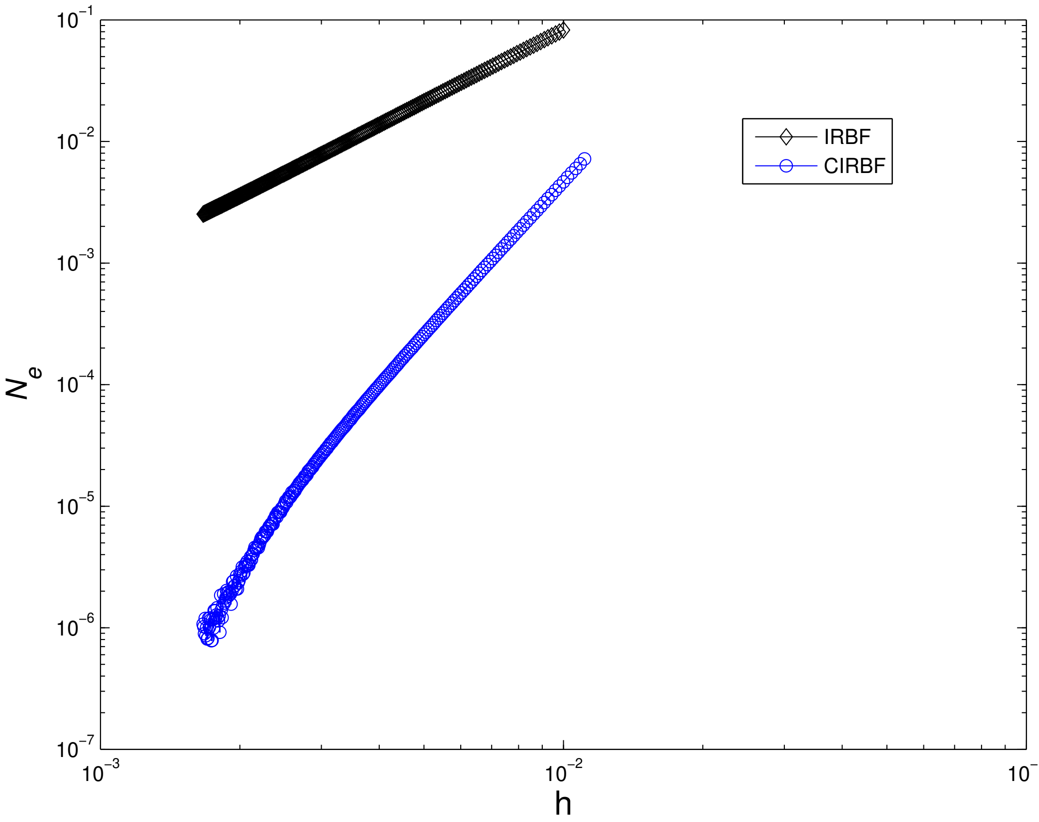


Figure 3.2: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $91 \leq N_x \leq 601$, $\beta = 20$: Solution accuracy by IRBF and CIRBF. The solution converges as $O(h^{1.95})$ for IRBF and $O(h^{4.79})$ for CIRBF. Note that the 3-point stencil is constructed on a unit length.

3.4 Improved constructions for compact IRBF stencils

Below are several treatments proposed to stably compute $\mathcal{C}$ at large values of β.

3.4.1 Approach 1: Extended precision

As shown in (Fornberg and Wright, 2004), by constructing the RBF interpolation with the Contour-Pade algorithm, the numerical solution still behaves

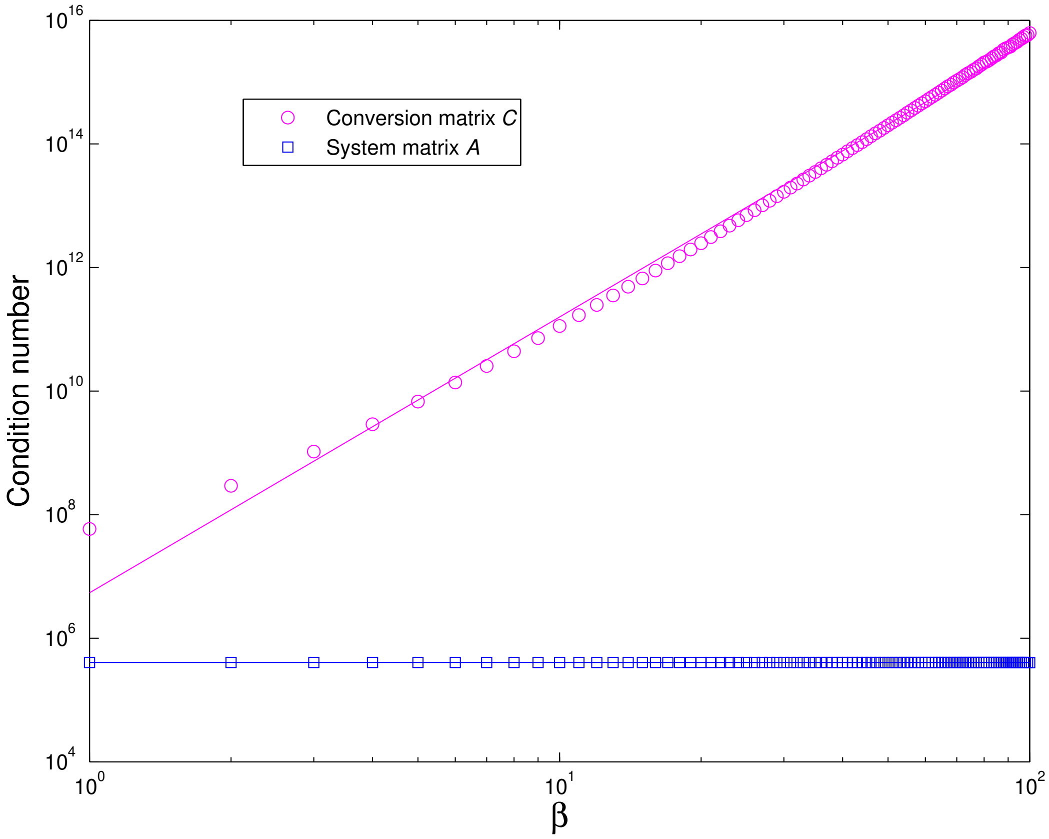


Figure 3.3: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $N_x = 1001$: Condition numbers of the system matrix A and conversion matrix C as functions of β representing the RBF width. When β increases, the growth rate is about 4.46 for cond(C) and 0.00 for cond(A). Note that the 3-point stencil is constructed on a unit length.

stably when the basis functions become increasingly flat. The trade-off between accuracy and stability, which has been widely reported in the RBF literature, is due to the use of finite (double) precision in computation. In this regard, the employment of higher precision is expected to improve the stability of the RBF solution, which was verified in (Huang et al., 2007, 2010). Our program is written in Matlab and we employ function vpa (variable-precision arithmetic) to increase the number of significant decimal digits from 16 to 50 in constructing the conversion matrix $\mathcal{C}$ and computing its inverse. Higher computational cost is required. On the other hand, as shown in Figure 3.4, the IRBF solution is stable at large values of β and the optimal value of β is also clearly detected. It is noted that: (i) by defining a stencil on the unit length, one may need to compute the inversion of the conversion matrix once and the result can be applied for any grid size to be employed, and (ii) in the present code, parts other than the computation of $\mathcal{C}$ are carried out using double precision, and numerical results indicate that the same level of accuracy is obtained as in the case of using extended precision for the entire computation.

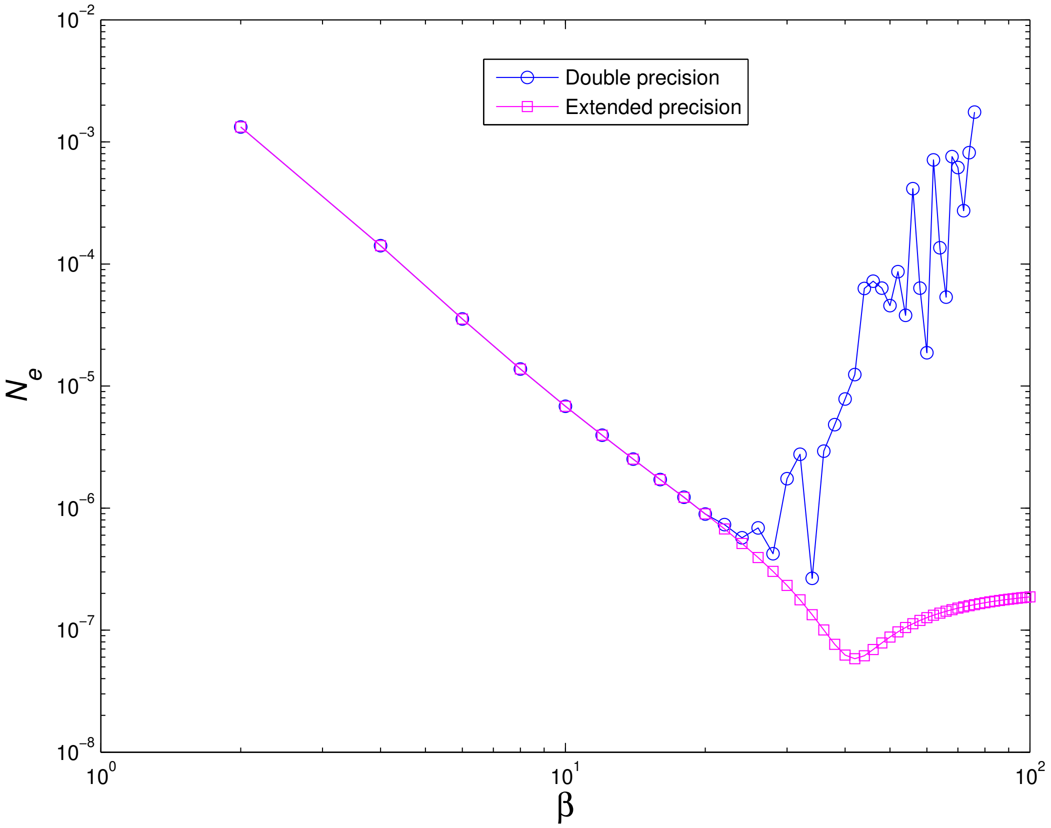


Figure 3.4: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $N_x = 1201$: Solution accuracy by using double precision and extended precision (50 digits) in constructing and computing the conversion matrix.

3.4.2 Approach 2: Definite integral

We propose to compute the integrals in their definite form rather than indefinite in constructing the conversion matrix $\mathcal{C}$. The advantage of this approach is that the size of $\mathcal{C}$ is reduced from 5×5 to 3×3, and the numerical stability is thus expected to be improved.

The integrals in (3.2)-(3.3) can be rewritten as

$$\begin{aligned}\frac{df(x)}{dx}-\frac{df_1}{dx} &= \sum_{i=1}^{3}\int_{x_1}^{x} w_i G_i(x)dx \\ &= \sum_{i=1}^{3} w_i\left[\mathcal{I}_i^{(1)}(x)-\mathcal{I}_i^{(1)}(x_1)\right] \qquad (3.11)\\ f(x)-f_1-(x-x_1)\frac{df_1}{dx} &= \sum_{i=1}^{3}\int_{x_1}^{x} w_i\left[\mathcal{I}_i^{(1)}(x)-\mathcal{I}_i^{(1)}(x_1)\right]dx \\ &= \sum_{i=1}^{3} w_i\left[\mathcal{I}_i^{(0)}(x)-\mathcal{I}_i^{(0)}(x_1)-(x-x_1)\mathcal{I}_i^{(1)}(x_1)\right] \\ & \qquad (3.12)\end{aligned}$$

Letting $\mathcal{I}_i^{(1)d(x)}=\mathcal{I}_i^{(1)}(x)-\mathcal{I}_i^{(1)}(x_1)$, $\mathcal{I}_i^{(0)d}=\mathcal{I}_i^{(0)}(x)-\mathcal{I}_i^{(0)}(x_1)-(x-x_1)\mathcal{I}_i^{(1)}(x_1)$ and $f'(x)=df(x)/dx$, expressions (3.11) and (3.12) reduce to

$$f'(x)-f_1'=\sum_{i=1}^{3} w_i\mathcal{I}_i^{(1)d(x)} \qquad (3.13)$$

$$f(x)-f_1-(x-x_1)f_1'=\sum_{i=1}^{3} w_i\mathcal{I}_i^{(0)d}(x) \qquad (3.14)$$

Our objective now is to express the weights w_1, w_2 and w_3 in terms of f_1, f_2, f_3, f_1'' and f_3''. The conversion system is generated by collocating the function expression (3.14) at $x=x_2$ and $x=x_3$, and the second-derivative expression (3.1) at $x=x_1$

$$\begin{pmatrix} f_2-f_1-(x_2-x_1)f_1' \\ f_3-f_1-(x_3-x_1)f_1' \\ f_1'' \end{pmatrix} = \underbrace{\begin{bmatrix} \mathcal{I}_1^{(0)d(x_1)}, & \mathcal{I}_2^{(0)d(x_1)}, & \mathcal{I}_3^{(0)d(x_1)} \\ \mathcal{I}_1^{(0)d(x_2)}, & \mathcal{I}_2^{(0)d(x_2)}, & \mathcal{I}_3^{(0)d(x_2)} \\ G_1(x_1), & G_2(x_1), & G_3(x_1) \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \qquad (3.15)$$

Solving this system for the weights yields

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} f_2 - f_1 - (x_2 - x_1)f_1' \\ f_3 - f_1 - (x_3 - x_1)f_1' \\ f_1'' \end{pmatrix} \tag{3.16}$$

A next step is to incorporate f_3'' into the vector on the RHS of (3.16). We first collocate the second-derivative expression (3.1) at $x = x_3$

$$f_3'' = [G_1(x_3), G_2(x_3), G_3(x_3)]\, \mathcal{C}^{-1} \begin{pmatrix} f_2 - f_1 - (x_2 - x_1)f_1' \\ f_3 - f_1 - (x_3 - x_1)f_1' \\ f_1'' \end{pmatrix} \tag{3.17}$$

and then solve this equation for f_1'. Making substitution into the RHS of (3.16), the mapping of the RBF space into the physical space takes the form

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \mathcal{C}^{-1}\mathcal{T} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \frac{d^2 f_1}{dx^2} \\ \frac{d^2 f_3}{dx^2} \end{pmatrix} \tag{3.18}$$

where $\mathcal{C}$ is of dimension 3×3 and $\mathcal{T}$ is of 3×5, which is constructed using results from solving equation (3.17).

Figure 3.5 concerns the conversion matrix $\mathcal{C}$ and shows a significant improvement in the condition number of the present definite-integral approach over the usual indefinite-integral approach. The former grows as $O(\beta^{3.88})$ only while the rate of the latter is much higher; up to 6.32. Figure 3.6 indicates that the present approach makes the solution accuracy significantly less fluctuating over large values of β.

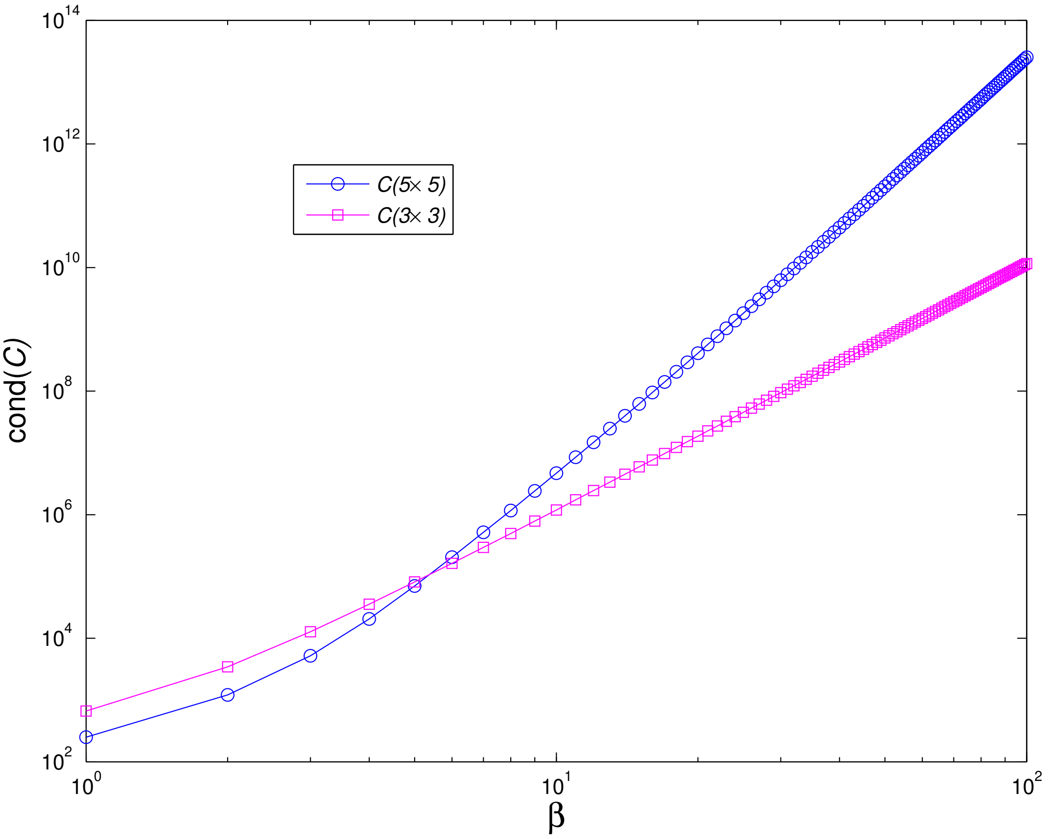


Figure 3.5: 3-point stencil, $N_x = 3$: Condition number of conversion matrix C computed through indefinite integrals, resulting in a matrix of 5×5 and through definite integral, resulting in a matrix of 3×3. The matrix condition number grows as $O(\beta^{6.32})$ for the former and $O(\beta^{3.88})$ for the latter.

3.4.3 Approach 3: Higher-order IRBF approximations

The MQ function $G_i(x)$ is now integrated four times (IRBF4) instead of twice (IRBF2). We employ the integrated basis function $\mathcal{I}_i^{(0)}(x)$ instead of $G_i(x)$ to approximate the second-order derivative

$$\frac{d^2 f}{dx^2} = \sum_{i=1}^{3} w_i \mathcal{I}_i^{(0)}(x) \tag{3.19}$$

$$\frac{df}{dx} = \sum_{i=1}^{3} w_i \int \mathcal{I}_i^{(0)}(x) + C_1 \tag{3.20}$$

$$f = \sum_{i=1}^{3} w_i \int \int \mathcal{I}_i^{(0)}(x) + C_1 x + C_2 \tag{3.21}$$

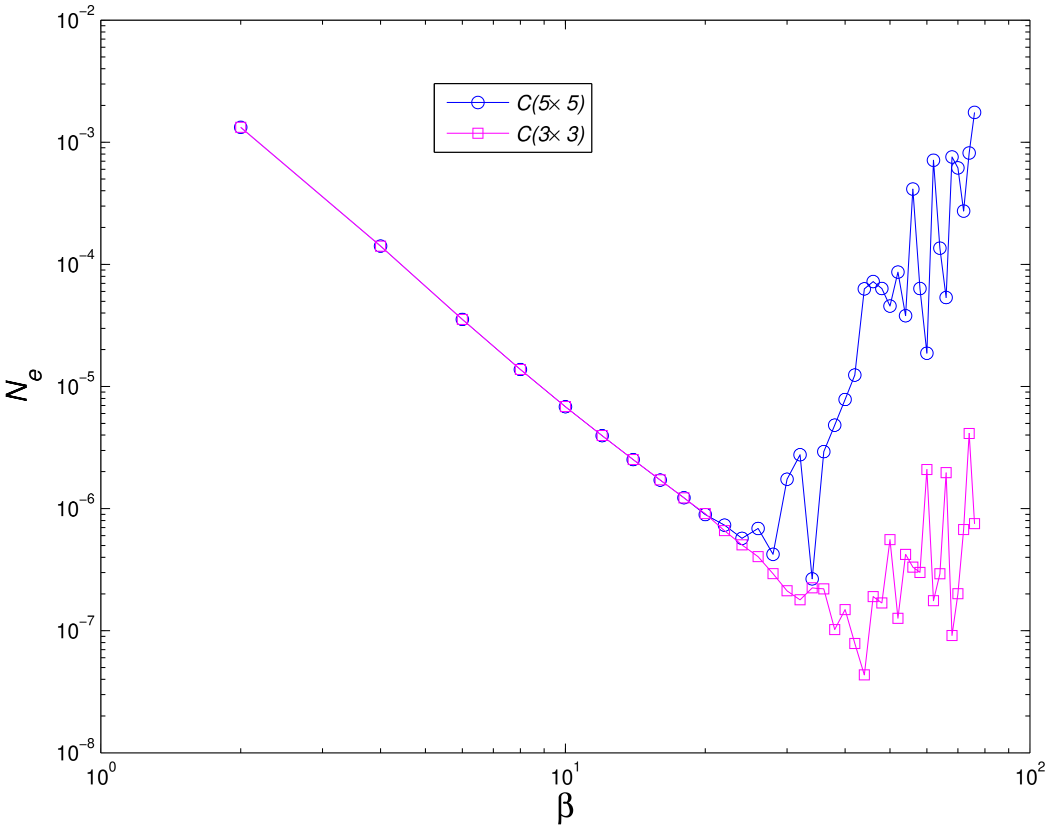


Figure 3.6: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $N_x = 1201$: Solution accuracy by the two methods (indefinite and definite integrals) of constructing matrix C.

It was reported in (Sarra, 2006) that the matrix condition number of IRBF4 is higher than that of IRBF2. However, with only three RBFs involved, the trend is reversed. As RBFs are integrated, the corresponding interpolation matrix has a lower condition number, particularly over a large range of β (Figure 3.7). When second-derivative values are added, as shown in Figure 3.8, the observation is similar. CIRBF4 is more stable than CIRBF2. This interesting property of higher-order IRBFs with 3 centres will be utilised here to construct compact IRBF stencils. The conversion system in this approach is formed as

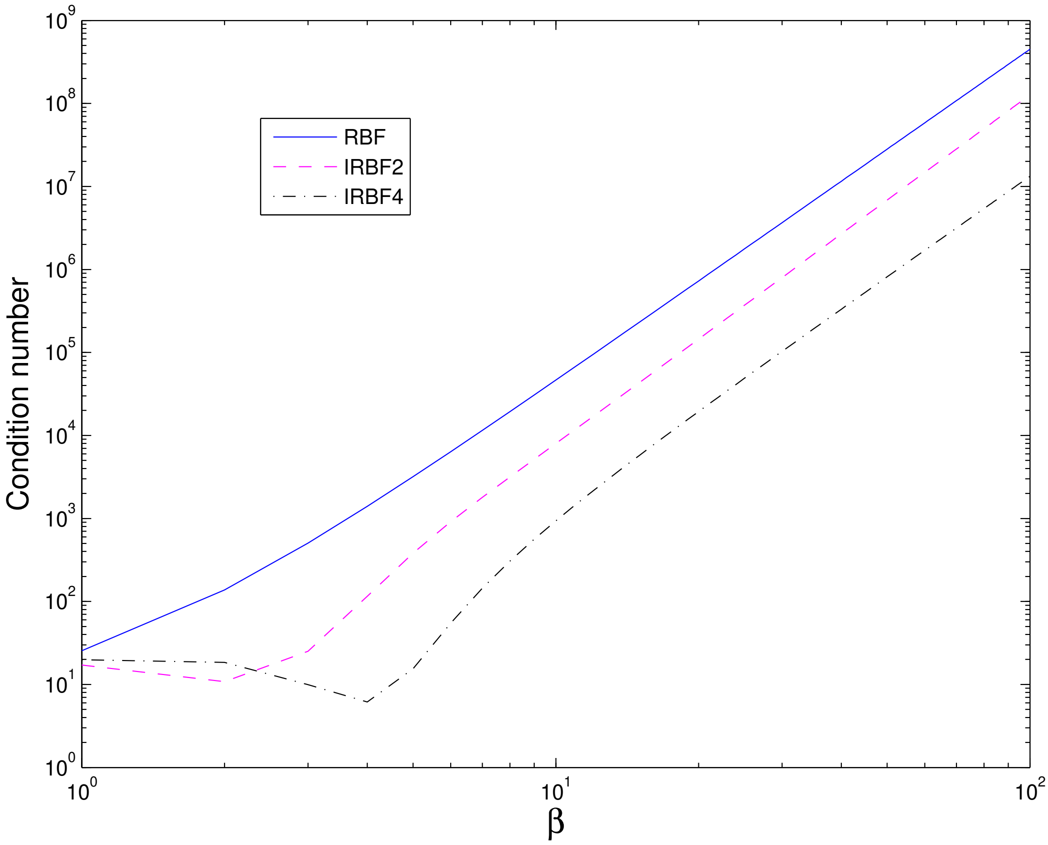


Figure 3.7: 3-point stencil: Condition numbers of the interpolation matrix generated by RBF, IRBF2 and IRBF4.

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \frac{d^2 f_1}{dx^2} \\ \frac{d^2 f_3}{dx^2} \end{pmatrix} = \underbrace{\begin{bmatrix} \int\int \mathcal{I}_1^{(0)}(x_1), & \int\int \mathcal{I}_2^{(0)}(x_1), & \int\int \mathcal{I}_3^{(0)}(x_1), & x_1, & 1 \\ \int\int \mathcal{I}_1^{(0)}(x_2), & \int\int \mathcal{I}_2^{(0)}(x_2), & \int\int \mathcal{I}_3^{(0)}(x_2), & x_2, & 1 \\ \int\int \mathcal{I}_1^{(0)}(x_3), & \int\int \mathcal{I}_2^{(0)}(x_3), & \int\int \mathcal{I}_3^{(0)}(x_3), & x_3, & 1 \\ \mathcal{I}_1^{(0)}(x_1), & \mathcal{I}_2^{(0)}(x_1), & \mathcal{I}_3^{(0)}(x_1), & 0, & 0 \\ \mathcal{I}_1^{(0)}(x_3), & \mathcal{I}_2^{(0)}(x_3), & \mathcal{I}_3^{(0)}(x_3), & 0, & 0 \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} \tag{3.22}$$

It can be seen from Figure 3.9 that, for a given β, a much more stable solution is obtained with the present approach as the grid size is reduced. At a very small grid size, the present approach is much more accurate and more stable over a large value range of β than the usual approach (Figure 3.10).

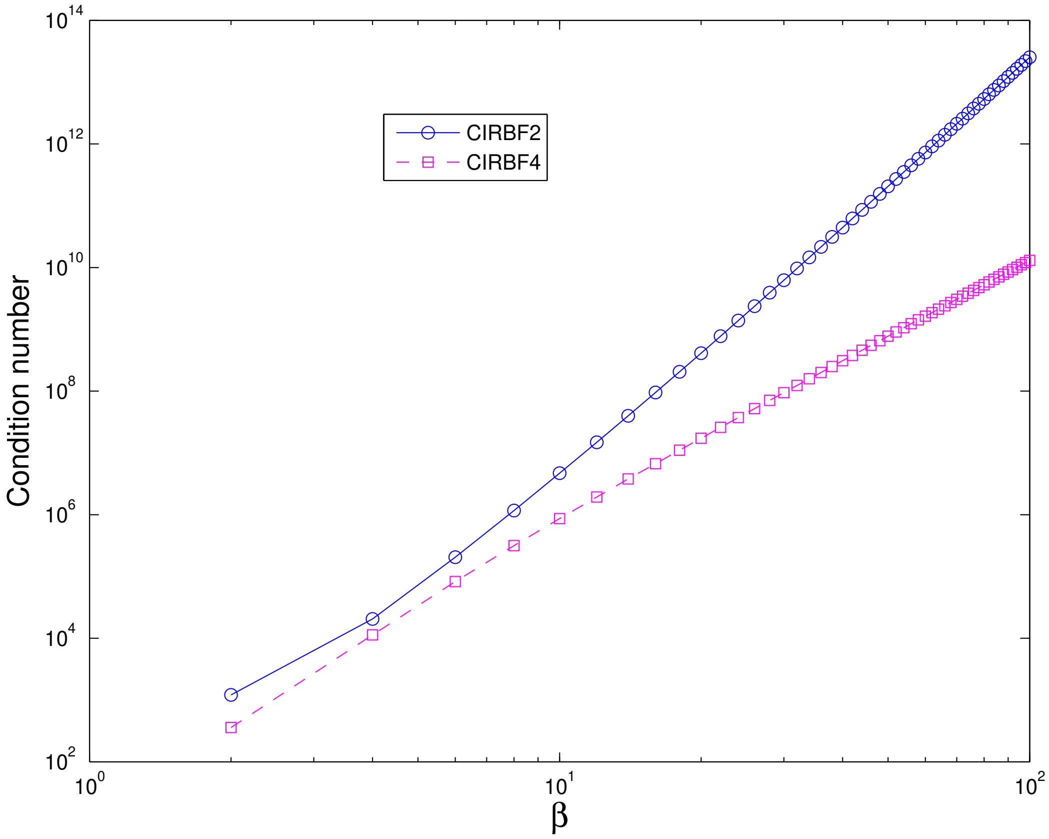


Figure 3.8: 3-point stencil, indefinite integral, $N_x = 3$: Condition numbers of the interpolation matrix generated by compact IRBF2 (indefinite integral) and compact IRBF4.

These improved 3-point CIRBF stencils can be extended to construct 5-point stencils for solving problems in two dimensions. The implementation process is exactly the same as that presented in (Mai-Duy and Tran-Cong, 2013). For elliptic PDEs, the algebraic system, where each row has 5 non-zero entries, can be solved iteratively using a Picard scheme. For parabolic PDEs, systems of tridiagonal equations can be formed and solved efficiently with the Thomas algorithm. It requires that the problem domain is represented by a Cartesian grid (not by a set of scattered points). Thus, for non-rectangular domains, the discretisation is still based on a Cartesian grid but with non-uniformly-spaced stencils. Consider Poisson's equation (3.30) defined on a non-rectangular domain (Figure 3.11) and subjected to Dirichlet boundary conditions. The exact solution is chosen to be $f^{(e)}(x,y) = \exp(-(x-0.25)^2 - (y-0.5)^2)\sin(\pi x)\cos(2\pi y)$.

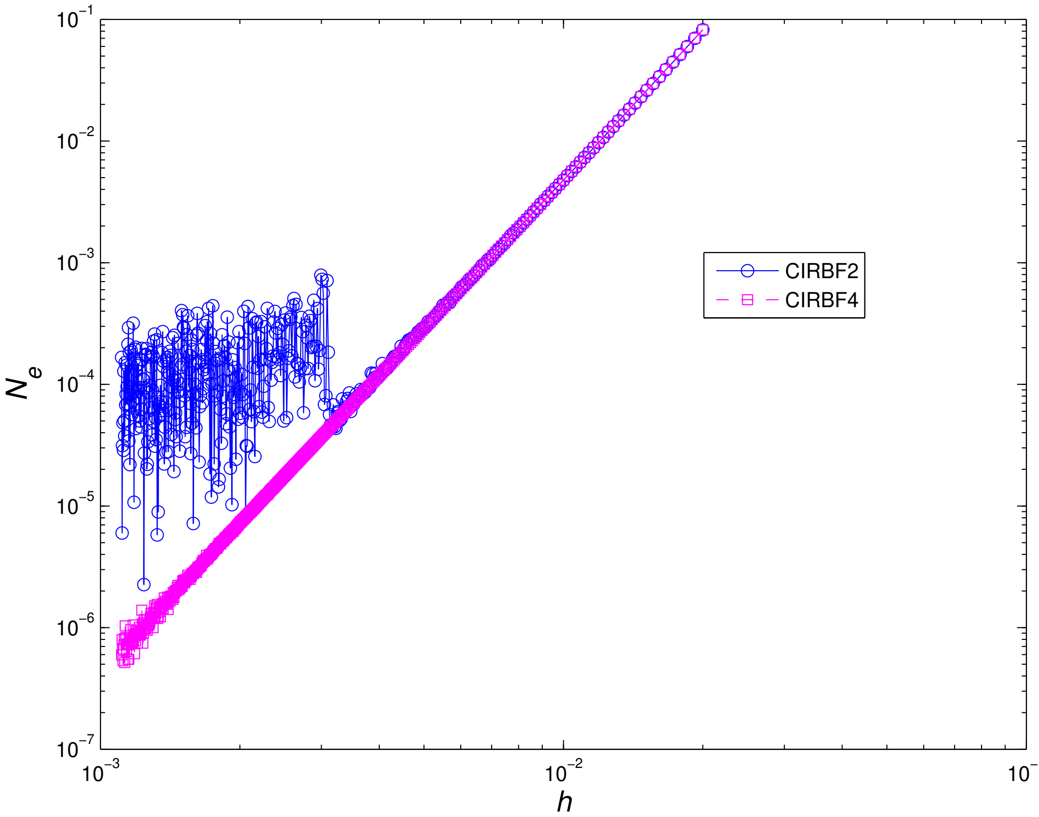


Figure 3.9: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $\beta = 50$, $N_x = (51, 53, \cdots, 901)$: Solution accuracy against the grid size by CIRBF2 (indefinite integral) and CIRBF4. For the latter, the solution converges as $O(h^{4.05})$.

The problem domain is embedded in a Cartesian grid, where the interior nodes are grid nodes inside the domain and the boundary nodes are the intersections of the grid lines and the boundary. Figure 3.11 also shows that as the RBF width increases, the present construction of CIRBF approximations results in a much more accurate and stable solution than the usual approach.

3.4.4 Approach 4: Separate construction in each direction and minimum number of derivative equations

This approach is developed for CIRBF stencils based on two-dimensional approximations. In this section, new compact 9-point IRBF stencils are con-

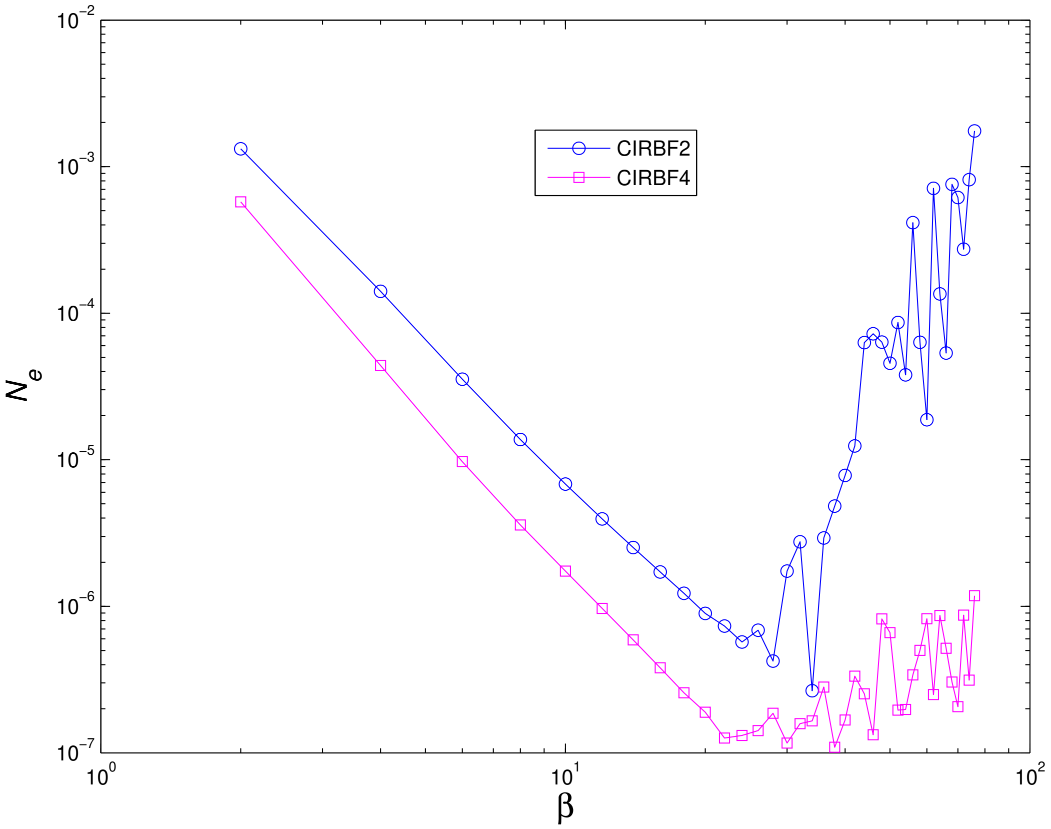


Figure 3.10: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $N_x = 1201$: Solution accuracy by CIRBF2 (indefinite integral) and CIRBF4.

structed. Unlike our previous work (Mai-Duy and Tran-Cong, 2011), the conversion process of the RBF space into the physical space is now conducted independently in each direction, where the size of the conversion matrix is reduced by about half. Below is a schematic diagram 9-point stencil associated with node (i, j)

$$\begin{bmatrix} \mathbf{x}_3 & \mathbf{x}_6 & \mathbf{x}_9 \\ \mathbf{x}_2 & \mathbf{x}_5 & \mathbf{x}_8 \\ \mathbf{x}_1 & \mathbf{x}_4 & \mathbf{x}_7 \end{bmatrix}$$

The nodes are locally numbered from left to right and from bottom to top, where node (i, j) is located at the centre (i.e. $(i, j) \equiv$ node 5). In the x direction, the

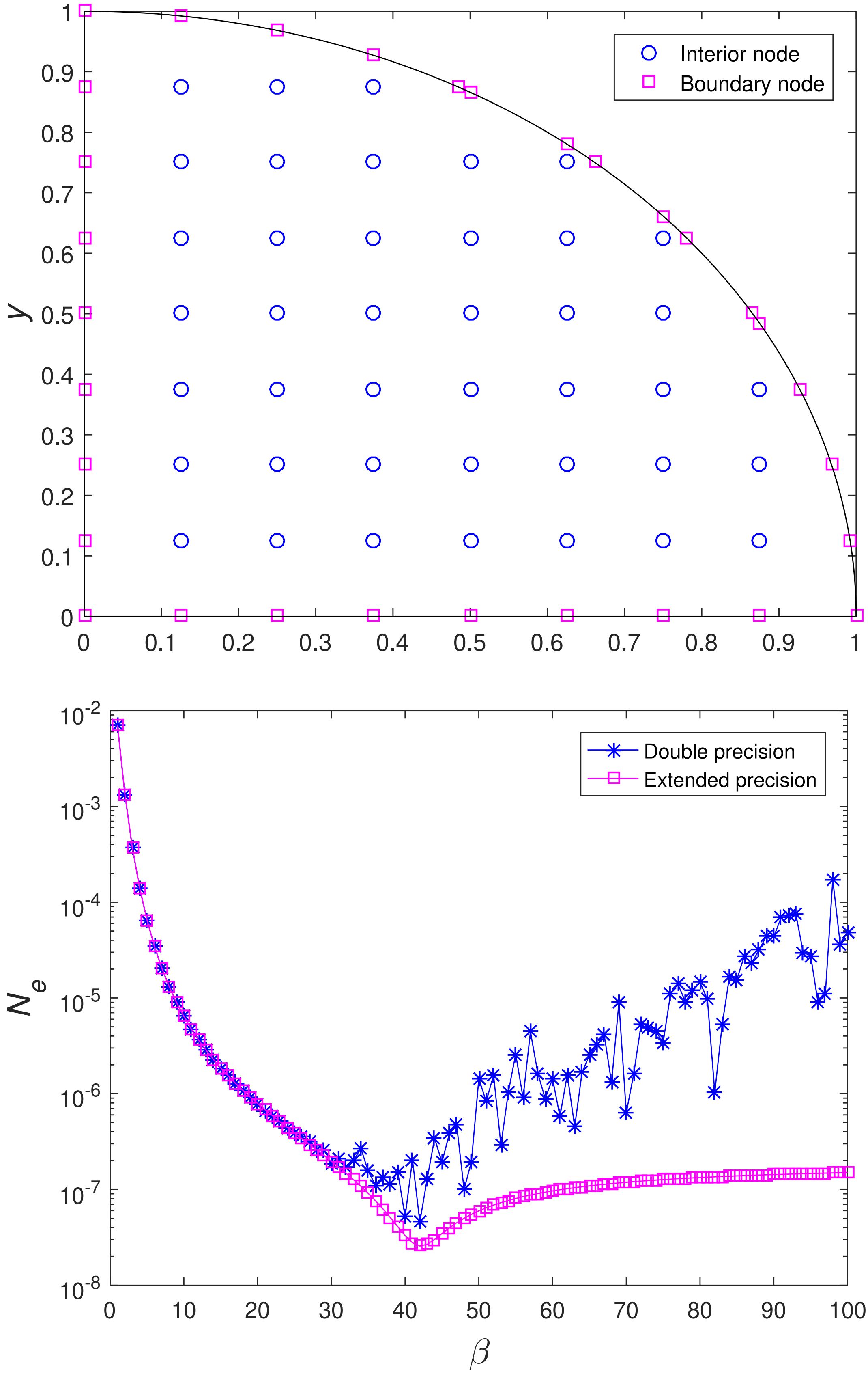


Figure 3.11: PDE, non-rectangular domain: Cartesian grid for non-rectangular domain, where the boundary nodes are the intersections of the grid lines and the boundary; and solution accuracy by using double precision and extended precision (50 digits) in constructing and computing the conversion matrix, where 81×81 grid lines are employed.

process of approximating the field variable and its derivatives starts with

$$\frac{\partial^2 f(x,y)}{\partial x^2} = \sum_{i=1}^{9} w_i G_i(x,y), \tag{3.23}$$

where $G_i(x,y) = \sqrt{[x-(c_i)_x]^2 + [y-(c_i)_y]^2 + a_i^2}$. Integrating (3.23) once and twice yields

$$\frac{\partial f}{\partial x}(x,y) = \sum_{i=1}^{9} w_i \mathcal{I}_i^{(1)}(x,y) + C_1(y) \tag{3.24}$$

$$f(x,y) = \sum_{i=1}^{9} w_i \mathcal{I}_i^{(0)}(x,y) + xC_1(y) + C_2(y) \tag{3.25}$$

where C_1 and C_2 are functions of y. It was shown in (Mai-Duy and Tran-Cong, 2011) that the most accurate approximation is achieved when the derivative values incorporated into the conversion system are taken at nodes 2, 4, 6 and 8. We follow this strategy in the present construction.

The conversion system is formed as

$$\begin{pmatrix} \overrightarrow{f} \\ \overrightarrow{\frac{\partial^2 f}{\partial x^2}} \end{pmatrix} = \underbrace{\begin{bmatrix} \overline{\mathcal{H}} \\ \mathcal{K} \end{bmatrix}}_{\mathcal{C}^{[x]}} \begin{pmatrix} \overrightarrow{w} \\ \overrightarrow{C} \end{pmatrix} \tag{3.26}$$

where $\mathcal{C}^{[x]}$ is the conversion matrix;

$$\overrightarrow{f} = (f_1, f_2, \cdots, f_9)^T$$

$$\overrightarrow{w} = (w_1, w_2, \cdots, w_9)^T$$

$$\overrightarrow{C} = (C_1(y_1), C_1(y_2), C_1(y_3), C_2(y_1), C_2(y_2), C_2(y_3))^T$$

$$\overline{\mathcal{H}} = \begin{bmatrix} \mathcal{I}_1^{(0)}(\mathbf{x}_1), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_1), & x_1, & 0, & 0, & 1, & 0, & 0 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_2), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_2), & 0, & x_2, & 0, & 0, & 1, & 0 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_3), & \cdots, & \mathcal{I}_9^{(0)[x]}(\mathbf{x}_3), & 0, & 0, & x_3, & 0, & 0, & 1 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_4), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_4), & x_4, & 0, & 0, & 1, & 0, & 0 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_5), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_5), & 0, & x_5, & 0, & 0, & 1, & 0 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_6), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_6), & 0, & 0, & x_6, & 0, & 0, & 1 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_7), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_7), & x_7, & 0, & 0, & 1, & 0, & 0 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_8), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_8), & 0, & x_8, & 0, & 0, & 1, & 0 \\ \mathcal{I}_1^{(0)}(\mathbf{x}_9), & \cdots, & \mathcal{I}_9^{(0)}(\mathbf{x}_9), & 0, & 0, & x_9, & 0, & 0, & 1 \end{bmatrix}$$

and

$$\frac{\overrightarrow{\partial^2 f}}{\partial x^2} = \mathcal{K} \begin{pmatrix} \overrightarrow{w} \\ \overrightarrow{C} \end{pmatrix}$$

are derivative equations. We observe that using a larger number of derivative equations can lead to a more accurate approximation but also increase the condition number of $\mathcal{C}$. We investigate the following two typical cases:

1. Case 1: two derivative equations

$$\frac{\overrightarrow{\partial^2 f}}{\partial x^2} = \left(\frac{\partial^2 f_2}{\partial x^2}, \frac{\partial^2 f_8}{\partial x^2} \right)^T$$

$$\mathcal{K} = \begin{bmatrix} G_1(\mathbf{x}_2), & \cdots, & G_9(\mathbf{x}_2), & 0, & \cdots, & 0 \\ G_1(\mathbf{x}_8), & \cdots, & G_9(\mathbf{x}_8), & 0, & \cdots, & 0 \end{bmatrix}$$

2. Case 2: four derivative equations

$$\overrightarrow{\frac{\partial^2 f}{\partial x^2}} = \left(\frac{\partial^2 f_2}{\partial x^2}, \frac{\partial^2 f_4}{\partial x^2}, \frac{\partial^2 f_6}{\partial x^2}, \frac{\partial^2 f_8}{\partial x^2}\right)^T$$

$$\mathcal{K} = \begin{bmatrix} G_1(\mathbf{x}_2), & \cdots, & G_9(\mathbf{x}_2), & 0, & \cdots, & 0 \\ G_1(\mathbf{x}_4), & \cdots, & G_9(\mathbf{x}_4), & 0, & \cdots, & 0 \\ G_1(\mathbf{x}_6), & \cdots, & G_9(\mathbf{x}_6), & 0, & \cdots, & 0 \\ G_1(\mathbf{x}_8), & \cdots, & G_9(\mathbf{x}_8), & 0, & \cdots, & 0 \end{bmatrix}$$

One can compute $\partial^2 f/\partial x^2$ at node 5 as

$$\frac{\partial^2 f_5}{\partial x^2} = [G_1(\mathbf{x}_5), \cdots, G_9(\mathbf{x}_5), 0, \cdots, 0] \left(\mathcal{C}^{[x]}\right)^{-1} \left(\overrightarrow{f}, \overrightarrow{\frac{\partial^2 f}{\partial x^2}}\right)^T \tag{3.27}$$

The approximation in the y direction can be derived in a similar fashion

$$\frac{\partial^2 f_5}{\partial y^2} = [G_1(\mathbf{x}_5), \cdots, G_9(\mathbf{x}_5), 0, \cdots, 0] \left(\mathcal{C}^{[y]}\right)^{-1} \left(\overrightarrow{f}, \overrightarrow{\frac{\partial^2 f}{\partial y^2}}\right)^T \tag{3.28}$$

where

$$\overrightarrow{\frac{\partial^2 f}{\partial y^2}} = \left(\frac{\partial^2 f_4}{\partial y^2}, \frac{\partial^2 f_6}{\partial y^2}\right)^T$$

for the case of two derivative equations, and

$$\overrightarrow{\frac{\partial^2 f}{\partial y^2}} = \left(\frac{\partial^2 f_2}{\partial y^2}, \frac{\partial^2 f_4}{\partial y^2}, \frac{\partial^2 f_6}{\partial y^2}, \frac{\partial^2 f_8}{\partial y^2}\right)^T$$

for the case of four derivative equations.

At each interior node, there are 3 unknowns, namely f, $\partial^2 f/\partial x^2$ and $\partial^2 f/\partial y^2$, and one can also establish three independent algebraic equations derived from collocating the differential equation

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2} = f(x, y) \tag{3.29}$$

and applying the CIRBF equations of second derivative in the x (i.e. (3.27)) and y (i.e. (3.28)) direction at the interior node.

We employ an iterative procedure to reduce the number of unknowns from 3 to 1. Substituting (3.27) and (3.28) into (3.29) and then collocating the obtained equation at node 5 leads to the following algebraic equation, e.g. for the case of two derivative equations,

$$\sum_{i=1}^{9} f_i^k = f_5 + \sum_{i=(2,8)} \gamma_i \frac{\partial^2 f_i^{k-1}}{\partial x^2} + \sum_{i=(4,6)} \lambda_i \frac{\partial^2 f_i^{k-1}}{\partial y^2} \tag{3.30}$$

where the superscript k is used to denote the present iteration. The solution procedure is as follows:

1. Guess a distribution of the field variable $f_{i,j}$

2. Compute second derivatives at grid nodes using equations (3.27) and (3.28).

3. Collocate (3.30) at the interior grid nodes, impose the prescribed boundary conditions and solve the obtained system of equations. Note that the system matrix is sparse as each row contains only 9 non-zero entries.

4. Check the convergence of the iterative procedure

$$CM = \frac{\sqrt{\sum \left(f_{i,j}^k - f_{i,j}^{k-1}\right)^2}}{\sqrt{\sum \left(f_{i,j}^k\right)^2}} < 10^{-12}$$

5. If not, relax the solution and then go back to step 2

$$f_{i,j}^k = \alpha f_{i,j}^k + (1-\alpha) f_{i,j}^{k-1}$$

   where α is the relaxation factor $(0 < \alpha \leq 1)$

6. If yes, stop and output the solution.

Consider Poisson's equation (3.30) defined on $0 \leq x, y \leq 1$ and subjected to Dirichlet boundary conditions. The exact solution is chosen to be $f^{(e)}(x,y) = \exp(-(x-0.25)^2 - (y-0.5)^2)\sin(\pi x)\cos(2\pi y)$. Using a grid of 37×37 and $\beta = 35$, the iterative scheme reaches $CM = 10^{-12}$ with 314 iterations for $\alpha = 0.1$, 95 for $\alpha = 0.3$, 51 for $\alpha = 0.5$, 31 for $\alpha = 0.7$ and 14 for $\alpha = 1$. The larger the value of α the faster the convergence will be. It is noted that the present iterative scheme can work with the largest value of α. In (3.30), the values of the second derivative at the side nodes of the stencil are imposed. Alternatively, one can impose the differential equation by making the following replacements

$$\left(\frac{\partial^2 f}{\partial x^2}\right)^{k-1}_{i-1,j} \to f_{i-1,j} - \left(\frac{\partial^2 f}{\partial y^2}\right)^{k-1}_{i-1,j}, \quad \left(\frac{\partial^2 f}{\partial x^2}\right)^{k-1}_{i+1,j} \to f_{i+1,j} - \left(\frac{\partial^2 f}{\partial y^2}\right)^{k-1}_{i+1,j}$$

$$\left(\frac{\partial^2 f}{\partial y^2}\right)^{k-1}_{i,j-1} \to f_{i,j-1} - \left(\frac{\partial^2 f}{\partial x^2}\right)^{k-1}_{i,j-1}, \quad \left(\frac{\partial^2 f}{\partial y^2}\right)^{k-1}_{i,j+1} \to f_{i,j+1} - \left(\frac{\partial^2 f}{\partial x^2}\right)^{k-1}_{i,j+1}$$

Numerical results indicate that the imposition of PDE rather than second derivatives results in a much faster convergence of the iterative scheme. For example, for $\alpha = 0.5$, the number of iteration is reduced from 51 to 34, as shown in Figure 3.12. Figure 3.13 shows the effect of the MQ width represented by β on the condition number of matrix $\mathcal{C}$ and the solution accuracy for a given grid size. Reducing the number of derivative equations leads to a much more stable calculation over large values of β. At $\beta = 38$, the condition number of matrix $\mathcal{C}$ using two derivative equations is about six orders of magnitude lower than the case of four derivative equations. The former produces highly accurate solutions at large β. The optimal value of β is clearly detected; the corresponding error Ne is 1.02×10^{-08}. When β is small (i.e. $\beta < 10$), it can be seen that matrix $\mathcal{C}$ is well conditioned, and using more derivative equations results in improved accuracy. Note that at large values of β, better accuracy is also obtained with the case of more derivative equations if extended precision is employed. Figure 3.14 shows the effect of the grid size on the matrix condition number and the solution accuracy at a large value of β. By constructing CIRBF

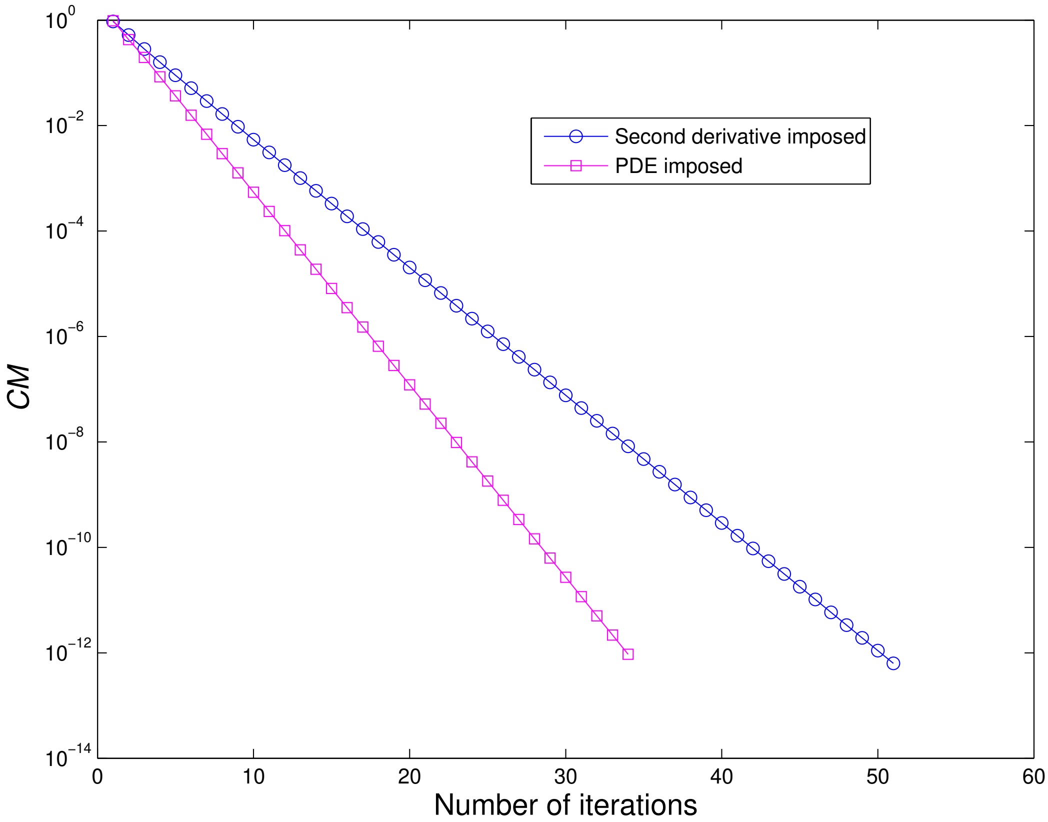
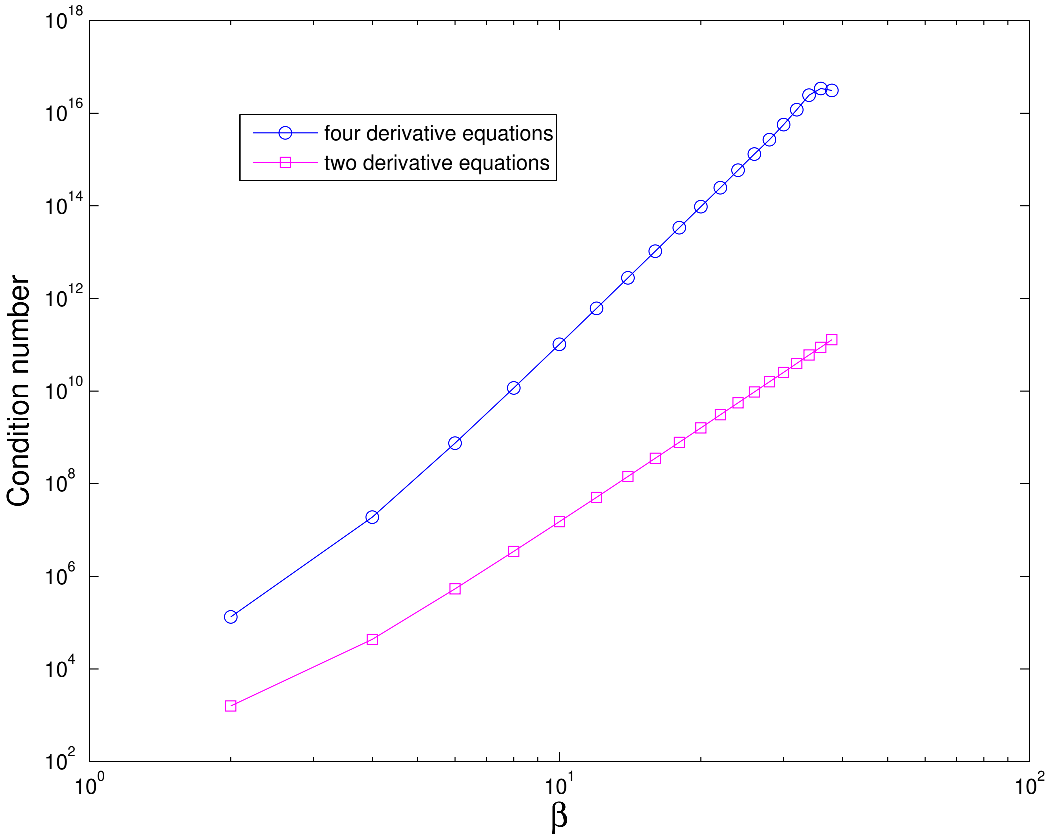


Figure 3.12: PDE, 37×37, $\beta = 35$, $\alpha = 0.5$, 2 derivative equations: Imposition of PDE converges faster than imposition of second derivatives.

approximations on a stencil defined on $[0,1] \times [0,1]$, the conversion matrix $\mathcal{C}$ is independent of the grid size. It can be seen that the condition numbers of $\mathcal{C}$ by the use of two and four derivative equations differ by six orders of magnitude for all grid sizes. However, the matrix A is well-conditioned for the two cases, where their condition numbers all grow slowly at the rate $O(h^{-2.00})$. The solution converges as $O(h^{5.12})$ for the case of two derivative equations and only $O(h^{2.31})$ for the case of four equations. At small values of h, the solution by the former is highly accurate with its error Ne being reduced to $O(10^{-9})$. The solution accuracy for the case of four derivative equations is significantly improved when extended precision is used; it produces greater accuracy than the case of two derivative equations.



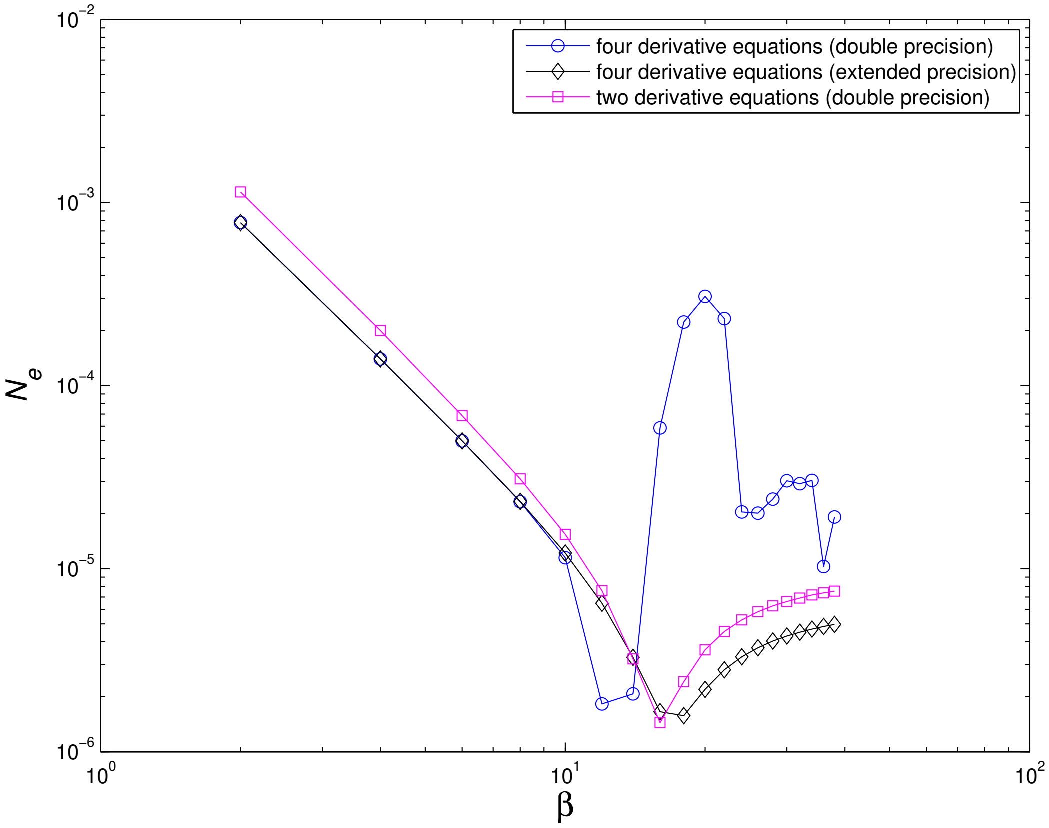


Figure 3.13: PDE, $\beta = (2, 4, 6, \cdots, 38)$, 31×31, $\alpha = 0.7$: Condition number of $\mathcal{C}$ and solution accuracy against the MQ width represented by β for two cases: four and two derivative equations. The 4 derivative equation case becomes unstable as β is increased. The fluctuation at large values of β is overcome by using extended precision or reducing the number of derivative equations.

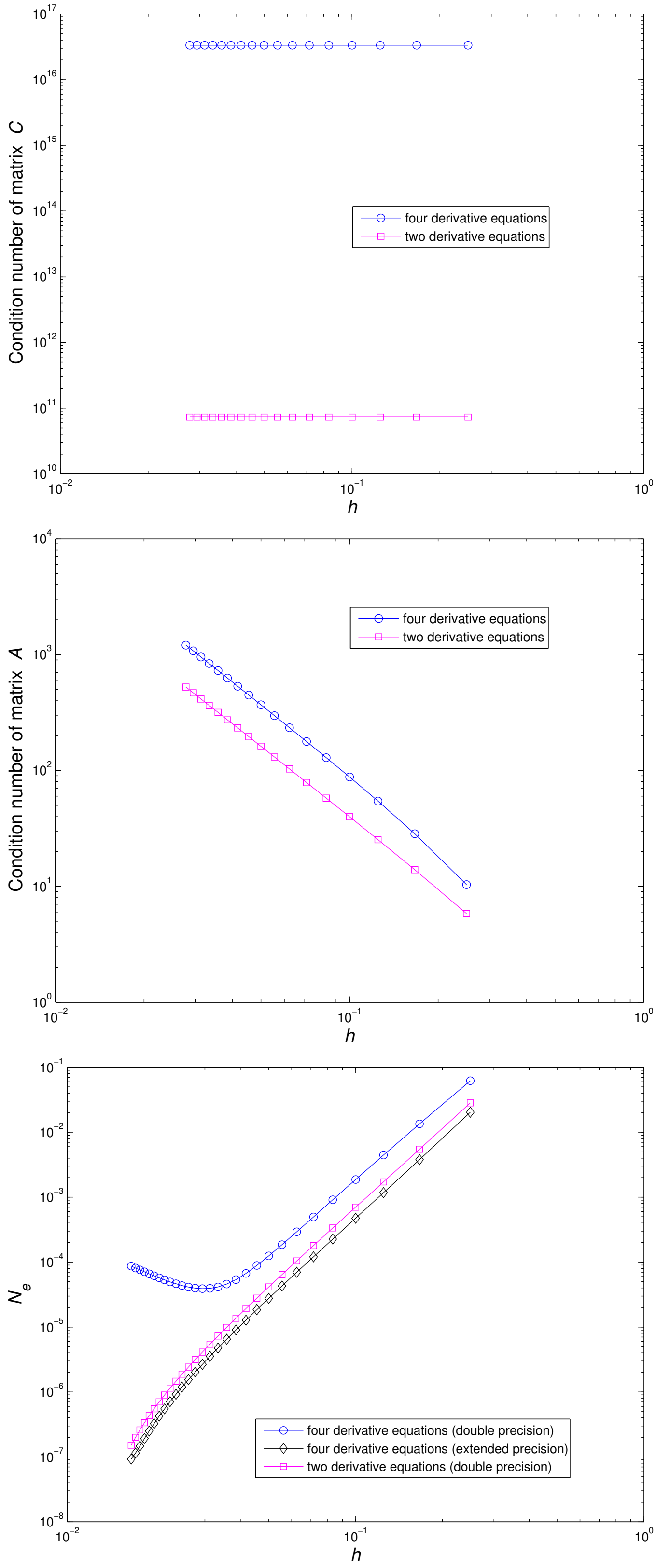


Figure 3.14: PDE, $5\times 5, 7\times 7, \cdots, 61\times 61$, $\beta = 35$, $\alpha = 0.7$: Condition numbers of $\mathcal{C}$ and $\mathcal{A}$, and solution accuracy against grid size for two cases: four and two derivative equations. The solution converges as $O(h^{2.37})$ for the former and $O(h^{4.41})$ for the latter. The four derivative equation case is much less accurate due to the fact that its associated matrix $\mathcal{C}$ is ill-conditioned; using extended precision, its performance becomes superior to the case of using two derivative equations.

3.4.5 ADI-Compact IRBF

The efficiency of the method can be improved by combining CIRBF with the alternating direction implicit (ADI): one of the most efficient schemes for solving time dependent problems. The ADI scheme was proposed by Douglas and Peaceman (1955). The method is obtained from the Crank-Nicolson scheme and has second-order accuracy in time. The ADI is efficient and is, therefore, suitable for large scale problems because it can decompose a 2-D or 3-D problem into a system of two or three 1-D problems which can be solved in parallel. This method is usually employed with FDM. To improve accuracy, high order ADI (You, 2006) or compact ADI (Dai and Nassar, 2002) have been developed. Applications of ADI schemes in various fluid flow problems can be found in (An et al., 2011; Navarro et al., 2007; Hejranfar and Khajeh-Saeed, 2011; Singh and You, 2011).

ADI scheme

We consider a parabolic differential equation subjected to initial solutions and boundary values

$$\frac{\partial f}{\partial t} - \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) = b(x, y, t), \tag{3.31}$$

with

$$\begin{aligned} f(x, y, 0) &= \zeta(x, y) \quad (x, y) \in \Omega \\ f(x, y, t) &= \eta(x, y) \quad (x, y) \in \partial\Omega \end{aligned}$$

where $\zeta(x,y), \eta(x,y)$ are given functions, Ω and $\partial\Omega$ the problem domain and its boundary.

The main idea of ADI scheme is to discretise one dimension implicitly while using it explicitly in other dimensions. For 2D-problems, the time derivative can be split as follows

$$\frac{f^{n+1/2}-f^n}{\Delta t/2} = \left(\frac{\partial^2 f}{\partial x^2}\right)^{n+1/2} + \left(\frac{\partial^2 f}{\partial y^2}\right)^{n} + b^{n+1/2}, \tag{3.32}$$

$$\frac{f^{n+1}-f^{n+1/2}}{\Delta t/2} = \left(\frac{\partial^2 f}{\partial x^2}\right)^{n+1/2} + \left(\frac{\partial^2 f}{\partial y^2}\right)^{n+1} + b^{n+1/2}. \tag{3.33}$$

By using local IRBF, the second derivative of f can be represented by the form

$$\widehat{\frac{\partial^2 f}{\partial x^2}} = \widehat{\mathcal{D}}_{2x}\widehat{f} + \widehat{k}_{2x}, \tag{3.34}$$

and

$$\widehat{\frac{\partial^2 f}{\partial y^2}} = \widehat{\mathcal{D}}_{2y}\widehat{f} + \widehat{k}_{2y}. \tag{3.35}$$

Substitution of (3.34-3.35) into (3.32) yields

$$\widehat{f}^{n+1/2} - \widehat{f}^{n} = \frac{\Delta t}{2}\left(\widehat{\mathcal{D}}_{2x}\widehat{f}^{n+1/2} + \widehat{\mathcal{D}}_{2y}\widehat{f}^{n} + b^{n+1/2} + \widehat{k}_{2x} + \widehat{k}_{2y}\right)$$

or

$$\widehat{f}^{n+1/2} = \left(\mathbf{1} - \frac{\Delta t}{2}\widehat{\mathcal{D}}_{2x}\right)^{-1}\left(\left(\mathbf{1} + \frac{\Delta t}{2}\widehat{\mathcal{D}}_{2y}\right)\widehat{f}^{n} + \frac{\Delta t}{2}\left(b^{n+1/2} + \widehat{k}_{2x} + \widehat{k}_{2y}\right)\right) \tag{3.36}$$

For the second step, the value of $\widehat{f}^{n+1/2}$ in (3.36) is used to compute $\widehat{f}^{n+1}$

$$\widehat{f}^{n+1} - \widehat{f}^{n+1/2} = \frac{\Delta t}{2}\left(\widehat{\mathcal{D}}_{2x}\widehat{f}^{n+1/2} + \widehat{\mathcal{D}}_{2y}\widehat{f}^{n+1} + b^{n+1/2}\right) \tag{3.37}$$

or

$$\widehat{f}^{n+1} = \left(\mathbf{1} - \frac{\Delta t}{2}\widehat{\mathcal{D}}_{2y}\right)^{-1}\left(\left(\mathbf{1} + \frac{\Delta t}{2}\widehat{\mathcal{D}}_{2x}\right)\widehat{f}^{n+1/2} + \frac{\Delta t}{2}\left(b^{n+1/2} + \widehat{k}_{2x} + \widehat{k}_{2y}\right)\right). \tag{3.38}$$

It is noted that the solution is obtained on each grid line.
Considering second-order PDE as follows.

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = -18\pi^2 \sin(3\pi x)\sin(3\pi y) \tag{3.39}$$

As can be seen from Figure 3.15, when β increases, the growth rate is about 3.08 for cond (C) and 0.00 for cond (A).

The present technique is first verified through the solution of a test problem governed by equation (3.39) and subject to Dirichlet boundary conditions. The domain of interest is the region inside a square of 1×1. The exact solution for this problem is $\sin(3\pi x) \times \sin(3\pi y)$ from which Dirichlet boundary conditions can be derived analytically. A wide range of β, namely $\{2, 4, \cdots, 100\}$, is employed to study the convergence behaviour of the solution. In order to apply the ADI scheme, a pseudo temporal derivative $\partial f/\partial t$ is added to the left hand side of (3.39). The solution is achieved with a tolerance of 1×10^{-9}. Results concerning the error Ne and β are given in Figure 3.16. It can be seen that the local-Flat-IRBF solution is stable even at a high value of β, and the optimal value of β is also clearly found. Condition numbers of the system matrix, a very important property for the direct solver (i.e. inverse of system matrices), is relatively low (e.g. 10^4 for a grid of 111×111). With a local approxima-

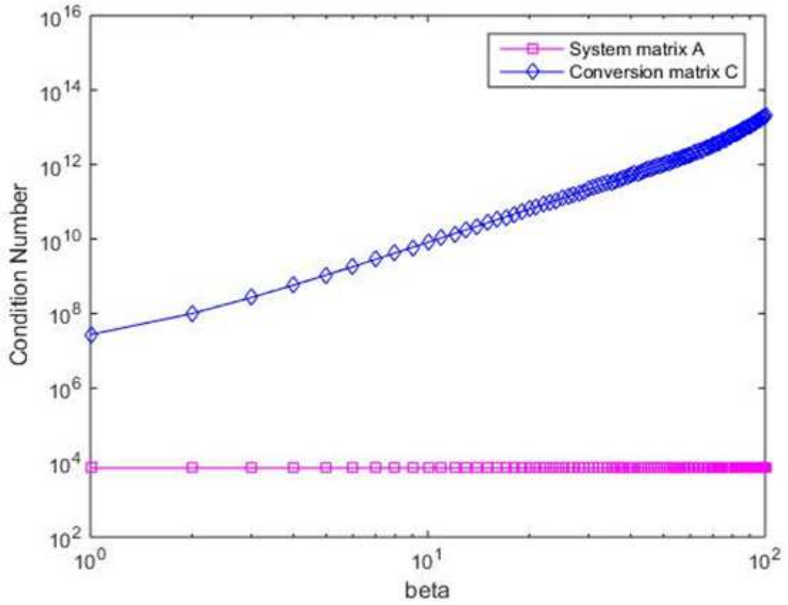


Figure 3.15: Condition number of system matrix A and conversion matrix C as a function of β.

tion property, the present method results in a sparse system matrix (Figure 3.17). Comparison of the computational accuracy between the 1D-IRBF, higher-order compact FD methods (HOC) (Tian et al., 2011), compact IRBF (CIRBF) (ThaiQuang et al., 2012), couple compact IRBF (CIRBF) (Tien et al., 2015) and the present flat (IRBF) is shown in Table 3.1, where the grid increases as $\{21 \times 21, 31 \times 31, ...\}$. It can be seen that the Flat kernel IRBF provides the most accurate results. For example, it can reach a low error 1.8×10^{-5} at grid (31×31). In order to get the same accuracy, 1D-IRBF needs a grid of 81×81, and the HOC method, CIRBF and CCIRBF need a grid of 41×41. The mesh convergence of 1D-IRBF, HOC, CIRBF, CCIRBF and the present flat IRBF is illustrated in Figure 3.18. As shown in Figure 3.18 the flat kernel IRBF is the most accurate.

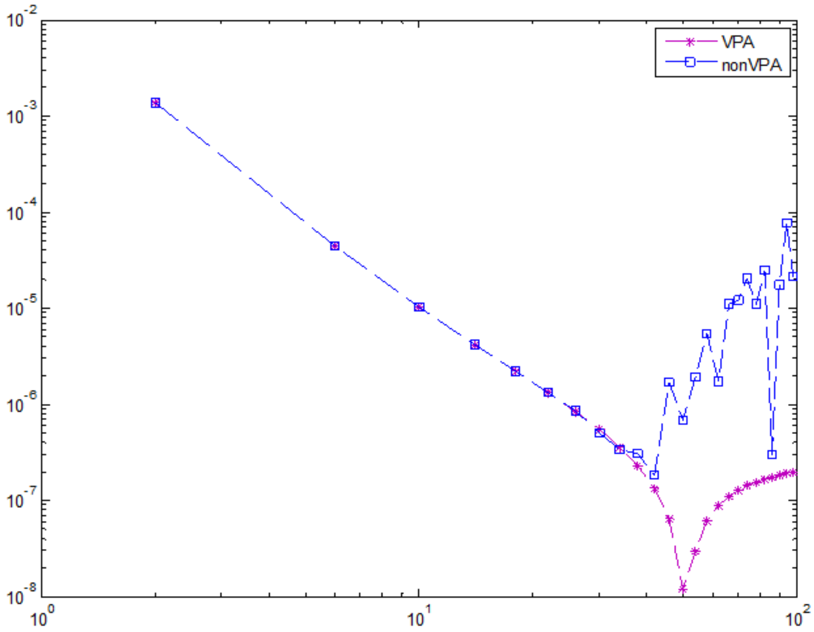


Figure 3.16: Investigation of β and Ne 81 × 81.

3.5 Concluding remarks

This chapter shows that, by taking appropriate ways of constructing approximations in the process of converting the RBF space into the physical space, compact local integrated RBF stencils based on one- and two-dimensional approximations are capable of producing a stable solution over large values of the RBF width. Four approaches, based on extended precision, definite integrals, higher IRBF approximations and minimum number of derivative equations, have been presented and numerically verified. For differential problems with smooth solutions, much more stable calculations and highly accurate results over the usual approaches have been obtained. Each approach has its own strengths and weaknesses. A more accurate and stable solution is achieved with extended precision at the expense of higher computational costs and the need for specialized computational tools such as function vpa in Matlab. However, in the case of uniform grids, by defining stencils on the unit length (1D) and the unit square (2D), one may need to compute the inversion once and then store/apply for any grid to be employed. The other approaches, which are sim-

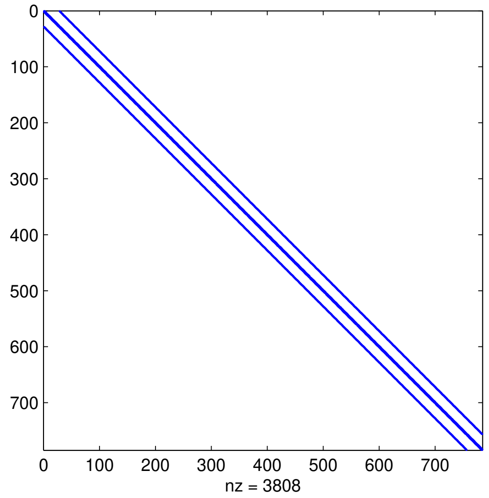


Figure 3.17: Local-flat-IRBF yields symmetric and sparse system matrices.

ple and easy to implement, are capable of making the working range of the RBF witdth much larger. Their computational costs are relatively low. This work further shows a great potential of compact IRBF stencils in solving differential problems.

Table 3.1: Poisson's equation: Accuracy obtained by the other RBFs (1D-IRBF, HOC (Tian et al., 2011), CIRBF (ThaiQuang et al., 2012) , CCIRBF (Tien et al., 2015)) and the proposed flat-IRBF methods. Notice that $a(-b)$ means $a \times 10^{-b}$.

Grid	*RMS*				
	1D-IRBF	HOC	CIRBF	CCIRBF	Present flat IRBF
21 × 21	1.2311(-3)	3.3579(-4)	3.3492(-4)	2.5405(-4)	9.6500(-5)
31 × 31	3.6879(-4)	5.6856(-5)	5.6674(-5)	4.2362(-5)	1.8700(-5)
41 × 41	1.5624(-4)	1.4589(-5)	1.4594(-5)	1.0997(-5)	5.7300(-6)
51 × 51	7.9915(-5)	4.9330(-6)	4.7148(-6)	3.7709(-6)	2.2300(-6)
61 × 61	4.6060(-5)	2.0151(-6)	1.9227(-6)	1.5371(-6)	1.0000(-6)
71 × 71	2.8837(-5)	9.4467(-7)	9.2935(-7)	7.1799(-7)	4.9300(-7)
81 × 81	1.9185(-5)	4.9199(-7)	4.6935(-7)	3.8210(-7)	2.5600(-7)
91 × 91	1.3375(-5)	2.7850(-7)	3.0597(-7)	2.0317(-7)	1.3600(-7)
101 × 101	9.6748(-6)	1.6869(-7)	1.5204(-7)	1.3230(-7)	7.1600(-8)
111 × 111	7.2123(-6)	1.0805(-7)	1.4662(-7)	7.8442(-8)	3.5300(-8)

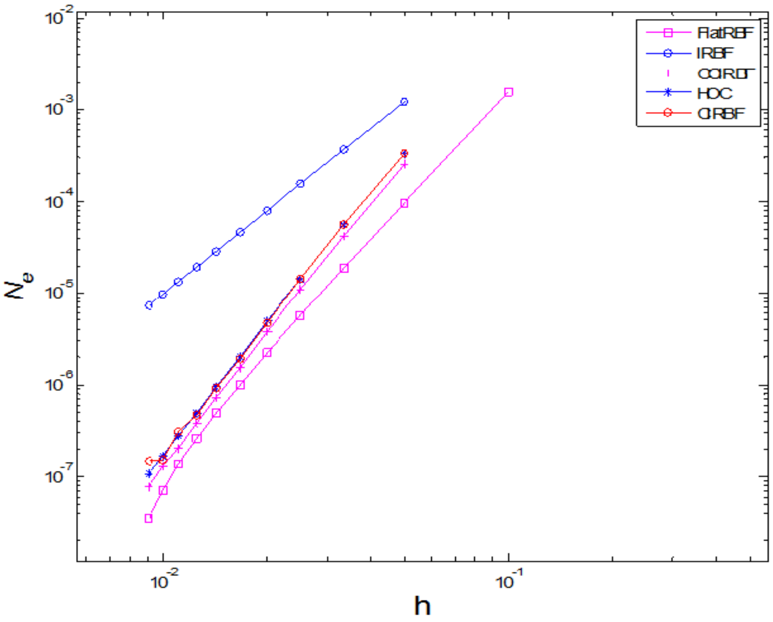


Figure 3.18: Mesh convergence of 1D-IRBF, HOC, CIRBF, CCIRBF and present flat IRBF.

Chapter 4

Compact symmetric IRBF stencils for spatial approximations

In this chapter, CIRBFs are employed for Hermite interpolation to solve differential equations, resulting in a new meshless symmetric RBF method. The symmetric property allows for the saving of computer storage space and the use of a more efficient algebraic solver. The focus is on the construction of compact approximation stencils, where a sparse system matrix and a high-order accuracy can both be achieved. Cartesian-grid-based stencils are possible for problems defined on non-rectangular domains. Furthermore, the effects of the RBF width on the solution accuracy for a given grid size are fully explored with reasonable computational cost. The proposed schemes are numerically verified in some elliptic boundary-value problems governed by the Poisson and convection-diffusion equations. High levels of solution accuracy are obtained using relatively coarse discretisations.

4.1 Introduction

For a given node distribution, solution accuracy can still be improved by changing the value of the RBF width. Finding the optimal RBF width for a general case presents a great challenge. In practice, one may rely on numerical algorithms such as those based on statistics (cross validation and maximum likelihood estimation) for determining this value. For a smooth function, the best accuracy can often be achieved at a large RBF width (i.e. near-flat RBF). As the RBF width increases, its matrix condition number grows rapidly and one needs stable-calculation algorithms to obtain a reliable numerical solution (see, e.g., Fornberg and Wright, 2004; Wright and Fornberg, 2006; Huang et al., 2007; Huang et al., 2010; Fornberg and Flyer, 2011; Rashidinia et al., 2016). The issue of stagnation errors (i.e. failure of convergence under continuing node refinement) was recently discussed in (Flyer and Barnett, 2016) along with several treatments proposed to overcome it.

For data containing both function and derivative values, one can employ the RBF Hermite interpolation approach (Hardy, 1975; Wu, 1992; Sun, 1994). Its applications in the solution of ODEs/PDEs have been reported in studies such as (Fasshauer, 1997; Power and Barraco, 2002; Larsson and Fornberg, 2003). The main advantage of this approach is that it can yield an interpolation matrix that is symmetric and invertible for both function representation and solution of ODEs/PDEs. The symmetric property also allows for the saving of computer storage space and the use of a more efficient algebraic solver. In addition, the RBF Hermite interpolation approach was utilised to construct compact local approximations (Tolstykh and Shirobokov, 2005; Wright and Fornberg, 2006; Mai-Duy and Tran-Cong, 2011). This kind of application has attracted more attention in recent years as both a sparse system matrix and a fast convergence rate of the solution can be achieved simultaneously.

In the previous chapters, IRBFs have been employed to solve ODEs/PDEs (see, e.g., Mai-Duy and Tran-Cong, 2001; Ling and Trummer, 2004; Sarra, 2006; Shu and Wu, 2007; Chen, Fan. and Wen., 2010; Kansa et al., 2004). In IRBF methods, basis functions used for the approximation of a field variable are obtained by integrating RBFs. Numerical experiments have shown that IRBF methods can yield an improved rate of convergence. In previous reports (Mai-Duy and Tran-Cong, 2011, 2013; Mai-Duy et al., 2017), we integrated RBFs with respect to the Cartesian coordinates (i.e. x, y and z). Through integration constants, nodal derivative values can be incorporated into the IRBF expressions. Their associated basis functions are generally not radial and the resultant IRBF matrices are nonsymmetric. In this work, RBFs are integrated with respect to the radius without the addition of integration constants. All derived basis functions are radial and they are employed for Hermite interpolation. Both global and local approximations are considered, producing new strong forms of the IRBF approach. For the former, the interpolation at a point involves function values at all nodes and, therefore, its system matrix is fully populated. For the latter, compact IRBF approximations are constructed on small stencils, resulting in a sparse system matrix. For both versions, the interpolation matrix is symmetric. The obtained IRBF results are compared with those by the classical FDMs, compact FDMs, and Hermite methods based on differentiated RBFs. The remainder of the chapter is organised as follows. In Section 4.2, relevant basis functions for DRBFs and IRBFs are provided. Global and local schemes of the proposed IRBF Hermite-based method are presented and verified in Sections 4.3 and 4.4, respectively. Section 4.5 gives some concluding remarks. In the Appendix, the process of acquiring the limit of the fourth-order cross derivative as the radius approaches zero is described.

4.2 Basis functions for DRBFs and IRBFs

For DRBFs, a function can be represented by a linear combination of RBFs

$$f(\mathbf{x}) = \sum_{j=1}^{N} w_j G(\|\mathbf{x} - \mathbf{x}_j\|), \tag{4.1}$$

where N is the number of given data points, $\{G(r = \|\mathbf{x} - \mathbf{x}_j\|)\}_{j=1}^{N}$ a set of RBFs, $\{\mathbf{x}_j\}_{j=1}^{N}$ a set of centres which is normally chosen to be the same as a set of data points, and $\{w_j\}_{j=1}^{N}$ a set of weights to be found. It has been theoretically shown that the interpolation matrix derived from (4.1) on a set of distinct points is nonsingular if G is a positive definite function such as the inverse multiquadric and Gaussian functions, or a conditionally positive definite function of order 1 such as the multiquadric function (Micchelli, 1986). In this work, RBF is taken as the MQ function

$$G(r) = \sqrt{r^2 + a^2}, \tag{4.2}$$

where a is the MQ width.

Derivatives of function f can then be determined as, e.g., in two dimensions

$$\frac{\partial^q f(\mathbf{x})}{\partial x^q} = \sum_{j=1}^{N} w_j \frac{\partial^q G(\|\mathbf{x} - \mathbf{x}_j\|)}{\partial x^q}, \quad q = 1, 2, 3, \cdots, \tag{4.3}$$

$$\frac{\partial^q f(\mathbf{x})}{\partial y^q} = \sum_{j=1}^{N} w_j \frac{\partial^q G(\|\mathbf{x} - \mathbf{x}_j\|)}{\partial y^q}, \quad q = 1, 2, 3, \cdots, \tag{4.4}$$

$$\frac{\partial^q f(\mathbf{x})}{\partial x^m \partial y^n} = \sum_{j=1}^{N} w_j \frac{\partial^q G(\|\mathbf{x} - \mathbf{x}_j\|)}{\partial x^m \partial y^n}, \quad m = 1, 2, \cdots, \ n = 1, 2, \cdots, \ q = m + n. \tag{4.5}$$

Expressions for computing derivatives of (4.2) with respect to r up to the fourth order are

$$\frac{dG}{dr} = \frac{r}{\sqrt{r^2+a^2}}, \tag{4.6}$$

$$\frac{d^2G}{dr^2} = \frac{a^2}{(r^2+a^2)^{3/2}}, \tag{4.7}$$

$$\frac{d^3G}{dr^3} = -\frac{3a^2r}{(r^2+a^2)^{5/2}}, \tag{4.8}$$

$$\frac{d^4G}{dr^4} = -\frac{3a^2\left(a^2-4r^2\right)}{(r^2+a^2)^{7/2}}. \tag{4.9}$$

For IRBFs, a function is decomposed into a set of basis functions that are obtained from integrating (4.2) with respect to r. Below is the case, where the MQ is integrated 4 times

$$\bar{G}(r) = \mathcal{I}^{(0)} = \left(\frac{a^4}{45} - \frac{83a^2r^2}{720} + \frac{r^4}{120}\right)\sqrt{r^2+a^2} + \left(-\frac{a^4r}{16} + \frac{a^2r^3}{12}\right)\ln\left(\frac{r+\sqrt{r^2+a^2}}{a}\right), \tag{4.10}$$

$$\frac{d\bar{G}}{dr} = \mathcal{I}^{(1)} = \left(-\frac{13a^2r}{48} + \frac{r^3}{24}\right)\sqrt{r^2+a^2} + \left(-\frac{a^4}{16} + \frac{a^2r^2}{4}\right)\ln\left(\frac{r+\sqrt{r^2+a^2}}{a}\right), \tag{4.11}$$

$$\frac{d^2\bar{G}}{dr^2} = \mathcal{I}^{(2)} = \left(-\frac{a^2}{3} + \frac{r^2}{6}\right)\sqrt{r^2+a^2} + \frac{a^2r}{2}\ln\left(\frac{r+\sqrt{r^2+a^2}}{a}\right), \tag{4.12}$$

$$\frac{d^3\bar{G}}{dr^3} = \mathcal{I}^{(3)} = \frac{r}{2}\sqrt{r^2+a^2} + \frac{a^2}{2}\ln\left(\frac{r+\sqrt{r^2+a^2}}{a}\right), \tag{4.13}$$

$$\frac{d^4\bar{G}}{dr^4} = \mathcal{I}^{(4)} = \sqrt{r^2+a^2}. \tag{4.14}$$

Figure 4.1 illustrates the shape of the MQ (4.2) and the integrated MQ (4.10) for several values of the MQ width. It was shown in (Sarra, 2006) that the integrated MQ approaches a large constant as $1/a$ approaches zero. Both DRBFs and IRBFs are implemented in this work. For simplicity of notation, expression (4.1) is now used for the two approaches, where function $G(r)$ is taken in the form of (4.2) for DRBFs and in the form of (4.10) for IRBFs. We introduce the concept of order for IRBF. An IRBF is said to be of order α if its (original) RBF is integrated α times. For function G defined in (4.10), one has $\alpha = 4$. As shown in (Sarra, 2006), this IRBF is a conditionally positive definite function of order $(\alpha + 2)/2 = 3$ and from a theoretical point of view, one needs to add to the interpolant a polynomial whose order is less by 1 (i.e. 2) to acquire an invertible interpolation matrix. However, to our best knowledge, from the numerical experiments reported, a singular interpolation matrix was never observed when the IRBF approximations were not augmented with polynomial terms. Furthermore, the addition of a polynomial did not lead to any significant improvement in the solution accuracy at relatively coarse discretisations.

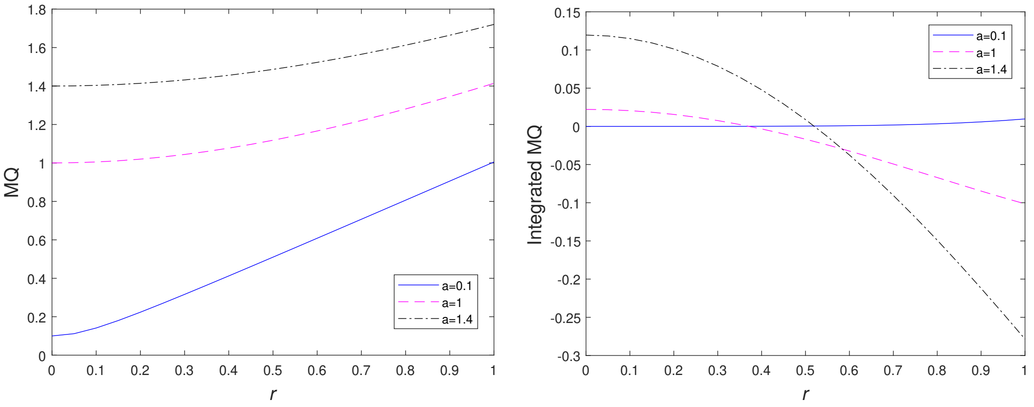


Figure 4.1: Variations of the MQ (left) and integrated MQ (right) for several values of the MQ width a

An effective way to compute derivatives of function G with respect to x and y on RHS of (4.3)-(4.5) ($k = \{1, 2, 3, 4\}$, $m = n = 2$) is to express them in terms of derivatives of G with respect to r. Since their expressions in the x and y coordinates are of similar forms, only pure derivatives with respect to x together with cross derivatives are given below

$$\frac{\partial G}{\partial x} = \frac{dG}{dr}\frac{\partial r}{\partial x}, \tag{4.15}$$

$$\frac{\partial^2 G}{\partial x^2} = \frac{dG}{dr}\frac{\partial^2 r}{\partial x^2} + \frac{d^2 G}{dr^2}\left(\frac{\partial r}{\partial x}\right)^2, \tag{4.16}$$

$$\frac{\partial^3 G}{\partial x^3} = \frac{dG}{dr}\frac{\partial^3 r}{\partial x^3} + 3\frac{d^2 G}{dr^2}\frac{\partial r}{\partial x}\frac{\partial^2 r}{\partial x^2} + \frac{d^3 G}{dr^3}\left(\frac{\partial r}{\partial x}\right)^3, \tag{4.17}$$

$$\frac{\partial^4 G}{\partial x^4} = \frac{dG}{dr}\frac{\partial^4 r}{\partial x^4} + \frac{d^2 G}{dr^2}\left[4\frac{\partial r}{\partial x}\frac{\partial^3 r}{\partial x^3} + 3\left(\frac{\partial^2 r}{\partial x^2}\right)^2\right] + 6\frac{d^3 G}{dr^3}\left(\frac{\partial r}{\partial x}\right)^2\frac{\partial^2 r}{\partial x^2} + \\ \frac{d^4 G}{dr^4}\left(\frac{\partial r}{\partial x}\right)^4, \tag{4.18}$$

$$\frac{\partial^4 G}{\partial x^2 \partial y^2} = \frac{d^3 G}{dr^3}\left[\frac{\partial^2 r}{\partial y^2}\left(\frac{\partial r}{\partial x}\right)^2 + 4\frac{\partial r}{\partial x}\frac{\partial r}{\partial y}\frac{\partial^2 r}{\partial x \partial y} + \frac{\partial^2 r}{\partial x^2}\left(\frac{\partial r}{\partial y}\right)^2\right] + \\ \frac{dG}{dr}\frac{\partial^4 r}{\partial x^2 \partial y^2} + \frac{d^2 G}{dr^2}\left[\frac{\partial^2 r}{\partial x^2}\frac{\partial^2 r}{\partial y^2} + 2\left(\frac{\partial^2 r}{\partial x \partial y}\right)^2 + 2\frac{\partial r}{\partial x}\frac{\partial^3 r}{\partial x \partial y^2} + 2\frac{\partial r}{\partial y}\frac{\partial^3 r}{\partial x^2 \partial y}\right] + \\ \frac{d^4 G}{dr^4}\left(\frac{\partial r}{\partial x}\right)^2\left(\frac{\partial r}{\partial y}\right)^2. \tag{4.19}$$

Since

$$r = \|\mathbf{x} - \mathbf{x}_j\| = \sqrt{(x - x_j)^2 + (y - y_j)^2}, \tag{4.20}$$

expressions for computing pure and cross derivatives of r on RHS of (4.15)-(4.19) are given by

$$\frac{\partial r}{\partial x} = \frac{x - x_j}{r}, \tag{4.21}$$

$$\frac{\partial^2 r}{\partial x^2} = \frac{r^2 - (x - x_j)^2}{r^3}, \tag{4.22}$$

$$\frac{\partial^3 r}{\partial x^3} = -\frac{3(x - x_j)\left[r^2 - (x - x_j)^2\right]}{r^5}, \tag{4.23}$$

$$\frac{\partial^4 r}{\partial x^4} = -\frac{3\left[r^2 - (x - x_j)^2\right]\left[r^2 - 5(x - x_j)^2\right]}{r^7}, \tag{4.24}$$

$$\frac{\partial^2 r}{\partial x \partial y} = -\frac{(x - x_j)(y - y_j)}{r^3}, \tag{4.25}$$

$$\frac{\partial^3 r}{\partial x \partial y^2} = \frac{(x - x_j)\left[-(x - x_j)^2 + 2(y - y_j)^2\right]}{r^5}, \tag{4.26}$$

$$\frac{\partial^3 r}{\partial x^2 \partial y} = \frac{(y - y_j)\left[-(y - y_j)^2 + 2(x - x_j)^2\right]}{r^5}, \tag{4.27}$$

$$\frac{\partial^4 r}{\partial x^2 \partial y^2} = \frac{2}{r^3} - \frac{15(x - x_j)^2(y - y_j)^2}{r^7}. \tag{4.28}$$

The limits of derivatives of function G when $r \to 0$ are

$$\frac{\partial G}{\partial x} \to 0, \quad \frac{\partial^2 G}{\partial x^2} \to \frac{1}{a}, \quad \frac{\partial^3 G}{\partial x^3} \to 0, \quad \frac{\partial^4 G}{\partial x^4} \to -\frac{3}{a^3}, \tag{4.29}$$

$$\frac{\partial G}{\partial y} \to 0, \quad \frac{\partial^2 G}{\partial y^2} \to \frac{1}{a}, \quad \frac{\partial^3 G}{\partial y^3} \to 0, \quad \frac{\partial^4 G}{\partial y^4} \to -\frac{3}{a^3}, \tag{4.30}$$

$$\frac{\partial^4 G}{\partial x^2 \partial y^2} \to -\frac{1}{a^3}, \tag{4.31}$$

for DRBF and

$$\frac{\partial G}{\partial x} \to 0, \quad \frac{\partial^2 G}{\partial x^2} \to -\frac{a^3}{3}, \quad \frac{\partial^3 G}{\partial x^3} \to 0, \quad \frac{\partial^4 G}{\partial x^4} \to a, \tag{4.32}$$

$$\frac{\partial G}{\partial y} \to 0, \quad \frac{\partial^2 G}{\partial y^2} \to -\frac{a^3}{3}, \quad \frac{\partial^3 G}{\partial y^3} \to 0, \quad \frac{\partial^4 G}{\partial y^4} \to a, \tag{4.33}$$

$$\frac{\partial^4 G}{\partial x^2 \partial y^2} \to \frac{a}{3}, \tag{4.34}$$

for IRBF. Obtaining results (4.29), (4.30), (4.32) and (4.33) is straightforward. For (4.31) and (4.34), one may need to replace the biharmonic operator with Laplace ones, and the detailed process is described in Appendix.

As discussed earlier, the quality of approximations by the MQs is dependent on both their spacing and width. For an easy interpretation, the MQ width a is expressed in terms of a typical distance from the MQ centre to its neighbours, denoted by h, as

$$a = \beta h, \tag{4.35}$$

where β is a constant that can run from a small to large positive value. For a given node distribution, the value of h can be determined. The advantage of (4.35) lies in its simplicity with β being a dimensionless quantity.

For the node refinement (scheme resolution), the value of h is reduced. In practice, the RBF width is then chosen to be smaller by, for example, keeping β fixed. It was shown in (Flyer and Barnett, 2016), this common practice may lead to the issue of stagnation errors (i.e. failure of convergence in the $h \to 0$ limit), which can be overcome by adding polynomial terms to the interpolant or keeping the RBF width fixed (i.e. fixed a). On the other hand, for a given grid size (fixed h), one can change β to improve the RBF approximations. In this case, it is expected that the addition of polynomial terms will not affect the solution accuracy very much. In this study, we focus on investigating the effects of β on the solution accuracy for a given grid size. A wide range of

β is explored by using the extended precision approach. Grid refinements are also studied; however, only relatively coarse grids are considered, for which the formula (4.35) can be applied without causing stagnation errors. Later it will be shown that the simple formula (4.35) can produce results that are very close to the ones corresponding to the best values of β over a range of grid sizes. Numerical experiments indicate that IRBFs lead to matrices of higher condition numbers than DRBFs. For a smooth function, accurate approximations by the former may thus occur earlier as β increases. In this regard, the comparison of accuracy between DRBFs and IRBFs should be made over a wide range of β rather than at its some particular values. For the presented numerical examples, in comparing the two RBF methods, a range of β as wide as possible is considered.

4.3 IRBF Hermite-based method: global scheme

Consider a differential problem

$$\mathcal{L}f(\mathbf{x}) = b(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{4.36}$$

$$\mathcal{B}f(\mathbf{x}) = s(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{4.37}$$

where Ω and Γ are a bounded domain and its boundary, $\mathcal{L}$ and $\mathcal{B}$ some linear differential operators, and b and s given functions. Let N be the total number of nodes and N_b the number of boundary nodes ($N_b < N$). The field variable is approximated as

$$f(\mathbf{x}) = \sum_{j=1}^{N_b} w_j \mathcal{B}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{j=N_b+1}^{N} w_j \mathcal{L}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{k=1}^{M} v_k p_k(\mathbf{x}), \tag{4.38}$$

where G is given by (4.10) for IRBFs and (4.2) for DRBFs, the notations $\mathcal{L}^{\mathbf{x}_j}$ and $\mathcal{B}^{\mathbf{x}_j}$ mean that $\mathcal{L}$ and $\mathcal{B}$ act on G considered as a function of the variable $\mathbf{x}_j$, and $\{p_k(\mathbf{x})\}_{k=1}^{M}$ is a basis for the M-dimensional space $(\prod_m^d)$ of all

d-variate polynomials that have degree less than or equal to m. The degree of the additional polynomial in (4.38) is dependent on the form of G employed, for example, $m = 2$ for (4.10) and $m = 0$ for (4.2) as shown in Section 4.2. To account for the addition of polynomial terms, the following extra constraints are imposed

$$\sum_{j=1}^{N_b} w_j \mathcal{B}^{\mathbf{x}} p_k(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_j} + \sum_{j=N_b+1}^{N} w_j \mathcal{L}^{\mathbf{x}} p_k(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_j} = 0, \quad k = 1, 2, \ldots, M. \tag{4.39}$$

Substitution of (4.38) into (4.37) and (4.36) yield

$$\sum_{j=1}^{N_b} w_j \mathcal{B}^{\mathbf{x}} \mathcal{B}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{j=N_b+1}^{N} w_j \mathcal{B}^{\mathbf{x}} \mathcal{L}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{k=1}^{M} v_k \mathcal{B}^{\mathbf{x}} p_k(\mathbf{x}) = s(\mathbf{x}), \tag{4.40}$$

$$\sum_{j=1}^{N_b} w_j \mathcal{L}^{\mathbf{x}} \mathcal{B}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{j=N_b+1}^{N} w_j \mathcal{L}^{\mathbf{x}} \mathcal{L}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{k=1}^{M} v_k \mathcal{L}^{\mathbf{x}} p_k(\mathbf{x}) = b(\mathbf{x}). \tag{4.41}$$

Collocation of (4.40) at the boundary points and of (4.41) at the interior grid nodes, together with (4.39), result in a set of $(N + M)$ algebraic equations for $(N + M)$ unknowns, namely $\{w_j\}_{j=1}^{N}$ and $\{v_k\}_{k=1}^{M}$, in which the system matrix is symmetric.

When the augmented polynomial is excluded from the RBF approximations, equations (4.40), (4.41) and (4.39) reduce to

$$\sum_{j=1}^{N_b} w_j \mathcal{B}^{\mathbf{x}} \mathcal{B}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{j=N_b+1}^{N} w_j \mathcal{B}^{\mathbf{x}} \mathcal{L}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) = s(\mathbf{x}), \qquad (4.42)$$

$$\sum_{j=1}^{N_b} w_j \mathcal{L}^{\mathbf{x}} \mathcal{B}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{j=N_b+1}^{N} w_j \mathcal{L}^{\mathbf{x}} \mathcal{L}^{\mathbf{x}_j} G(\|\mathbf{x}-\mathbf{x}_j\|) = b(\mathbf{x}), \qquad (4.43)$$

which lead to a set of only N algebraic equations for N unknowns.

4.3.1 ODEs

We apply the methods to the following second-order ODE $d^2f/dx^2 = -4\pi^2\sin(2\pi x)$, $0 \le x \le 1$, subject to Dirichlet boundary conditions. The exact solution can be verified to be $u_e(x) = \sin(2\pi x)$.

Equations (4.40), (4.41) and (4.39) take the form

$$\sum_{j=1}^{2} w_j G(\|x-x_j\|) + \sum_{j=3}^{N} w_j \frac{d^2G(\|x-x_j\|)}{dx_j^2} + \sum_{k=1}^{3} v_k p_k(x) = \sin(2\pi x), \qquad (4.44)$$

$$\sum_{j=1}^{2} w_j \frac{d^2G(\|x-x_j\|)}{dx^2} + \sum_{j=3}^{N} w_j \frac{d^4G(\|x-x_j\|)}{dx^2dx_j^2} + \sum_{k=1}^{3} v_k \frac{d^2p_k(x)}{dx^2} = -4\pi^2\sin(2\pi x), \qquad (4.45)$$

$$\sum_{j=1}^{2} w_j p_k(x_j) + \sum_{j=3}^{N} w_j \frac{d^2p_k(x_j)}{dx^2} = 0, \quad k = \{1,2,3\}. \qquad (4.46)$$

When the RBF approximations are not augmented with the polynomial terms, the above equations become

$$\sum_{j=1}^{2} w_j G(\|x-x_j\|) + \sum_{j=3}^{N} w_j \frac{d^2G(\|x-x_j\|)}{dx_j^2} = \sin(2\pi x), \qquad (4.47)$$

$$\sum_{j=1}^{2} w_j \frac{d^2G(\|x-x_j\|)}{dx^2} + \sum_{j=3}^{N} w_j \frac{d^4G(\|x-x_j\|)}{dx^2dx_j^2} = -4\pi^2\sin(2\pi x). \qquad (4.48)$$

We first compare the numerical performance of (4.44)-(4.46) and (4.47)-(4.48). The problem domain is discretised using a set of uniformly distributed points. We take h in (4.35) as the grid size. In the global scheme, the RBF approximations involve all nodes and therefore their matrix condition number is expected to grow rapidly. Values of β here should be chosen to be relatively small. The IRBF and DRBF results concerning the relative L_2 error, denoted by N_e, against the RBF width, displayed through β, are depicted in Figure 4.2, showing that the IRBF/DRBF solutions of the two systems (i.e. (4.44)-(4.46) and (4.47)-(4.48)) have similar behaviour. However, for IRBFs, the one without the augmented polynomial is slightly more accurate. It appears that adding polynomial terms to the interpolants for the case of a fixed h does not lead to an improvement in accuracy. For both cases (i.e. with and without the polynomial terms), we did not experience any singular interpolation matrix over a full range of the RBF width. These observations are consistent with remarks of other computational works in the RBF literature. In Figure 4.3, results by the IRBF and DRBF Hermite-based methods are compared for a given grid size. It can be seen that the former is generally more accurate than the latter over a wide range of β. As β increases, the computed errors of the two methods fluctuate due to their higher matrix condition numbers. Several algorithms to extend the working range of the RBF width have been proposed in the liter-

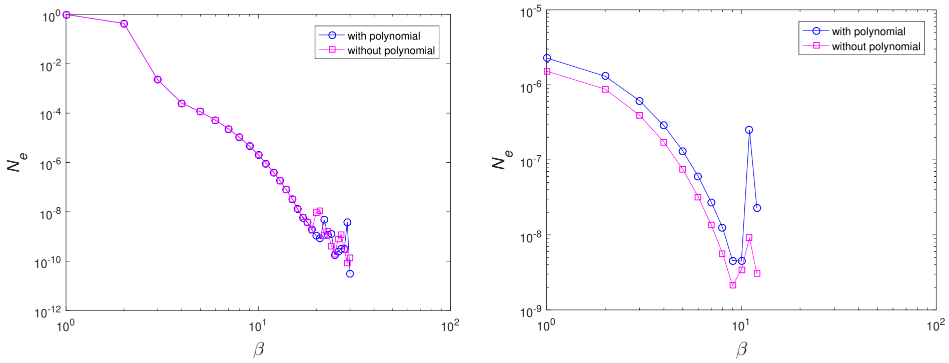


Figure 4.2: Second-order ODE, 151 grid nodes: Effects of the augmented polynomial in the RBF approximations on their solution accuracy over a wide range of the RBF width by the global DRBF (left) and IRBF (right) Hermite-based methods

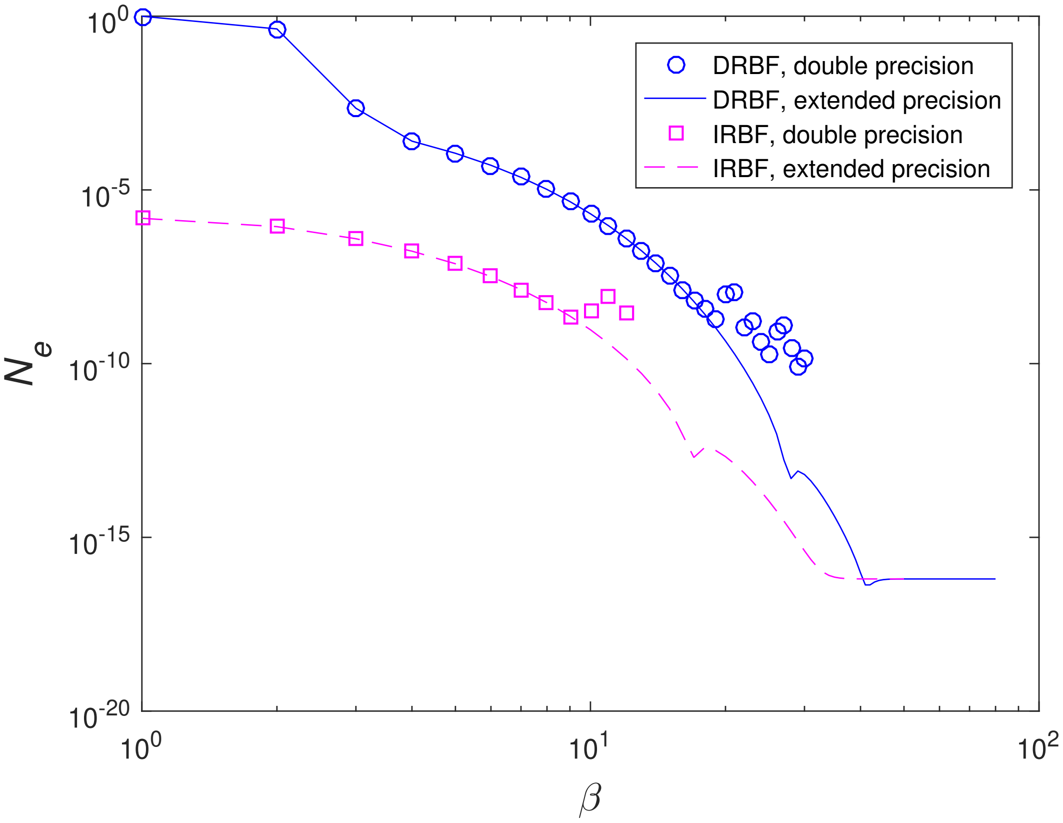


Figure 4.3: Second-order ODE, 151 grid nodes, double/extended precision: Effects of the RBF width on the solution accuracy by the global Hermite-based methods.

ature (Fornberg and Wright, 2004; Wright and Fornberg, 2006; Huang et al., 2007; Huang et al., 2010; Rashidinia et al., 2016; Mai-Duy et al., 2017). In this work, we employ the extended precision approach. Our programs are written in Matlab with function vpa being utilised to increase the number of significant figures from 16 to 50. As shown in Figure 4.3, the calculation is now stable over the full range of the RBF width. Results obtained indicate that the use of IRBFs leads to a significantly improved accuracy from a small to large RBF width. The best accuracy by the two methods corresponds to similar values of β. Figure 4.4 displays the effects of the grid size on the solution accuracy by the proposed global Hermite-based method for several values of β. The domain is represented by uniform grids, $\{101, 111, \cdots, 1001\}$. A relation between N_e and h in the log-log scale is fitted by a linear function with its slope being regarded as an average rate of convergence. The IRBF solution converges as $O(h^{2.83})$ for

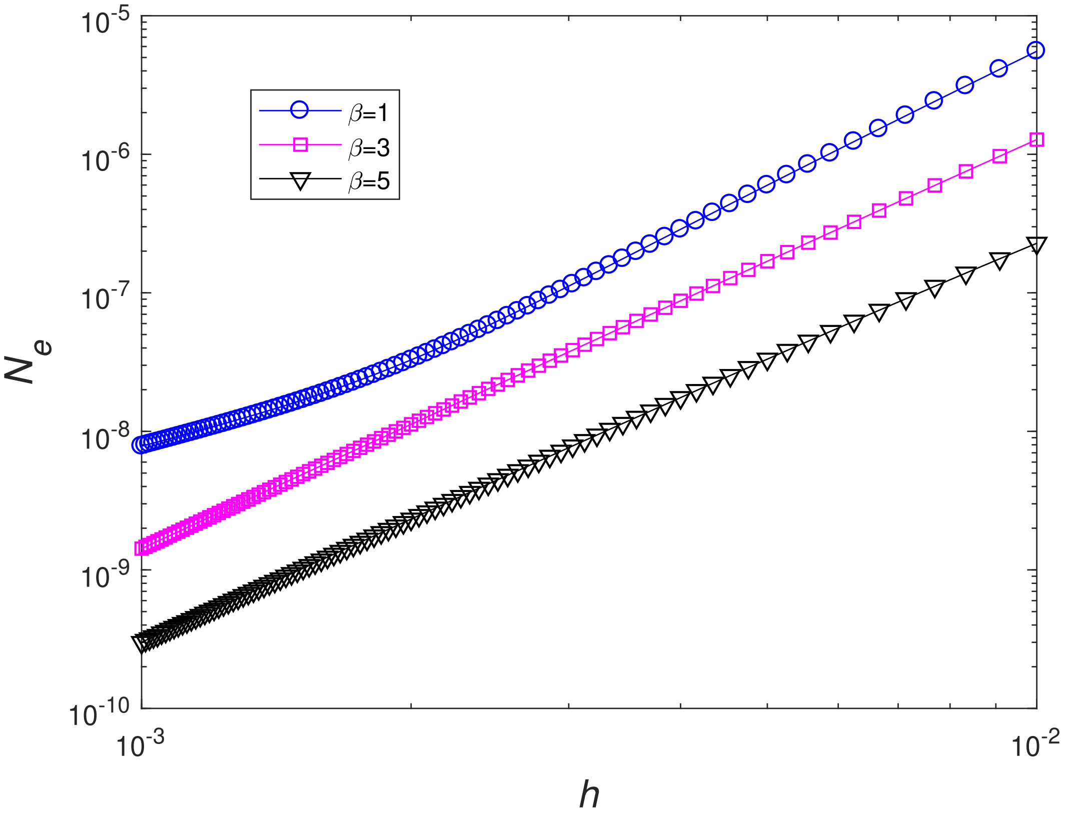


Figure 4.4: Second-order ODE, double precision: Grid convergence by the proposed global Hermite-based method for several values of β.

$\beta = 1$, $O(h^{2.96})$ for $\beta = 3$ and $O(h^{2.90})$ for $\beta = 5$. Highly accurate results are obtained; the relative L_2 error is reduced to $O(10^{-10})$ for $\beta = 5$. Also, constant values of β produce similar rates of convergence; larger β corresponds to a higher level of accuracy. These simple behaviours provide some useful guidance on how to choose the RBF width in practical applications.

4.3.2 PDEs

We apply the methods to Poisson's equation with its driving function $b = -2\pi^2 \sin(\pi x)\sin(\pi y)$ and $0 \leq x, y \leq 1$. The exact solution can be verified to be $f_e(x, y) = \sin(\pi x)\sin(\pi y)$ from which one can obtain Dirichlet boundary conditions.

For Poisson's equation, $\mathcal{L}^{\mathbf{x}}$ and $\mathcal{L}^{\mathbf{x}_j}$ take the form

$$\mathcal{L}^{\mathbf{x}} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \tag{4.49}$$

$$\mathcal{L}^{\mathbf{x}_j} = \frac{\partial^2}{\partial x_j^2} + \frac{\partial^2}{\partial y_j^2}. \tag{4.50}$$

Since $\mathbf{x}$ and $\mathbf{x}_j$ can be interchanged in defining the input r of MQ, one has

$$\begin{aligned} \mathcal{L}^{\mathbf{x}}\mathcal{L}^{\mathbf{x}_j} &= \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\left(\frac{\partial^2}{\partial x_j^2} + \frac{\partial^2}{\partial y_j^2}\right), \\ &= \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4} = \frac{\partial^4}{\partial x_j^4} + 2\frac{\partial^4}{\partial x_j^2 \partial y_j^2} + \frac{\partial^4}{\partial y_j^4}. \end{aligned} \tag{4.51}$$

We employ four set of unstructured nodes to represent the problem domain (Figure 4.5). The MQ width is defined here by assuming that the nodes are of uniform distribution; the distance h in (4.35) is chosen as the equivalent grid size (i.e. $h = 1/(\sqrt{N} - 1)$. Figure 4.6 shows the solution accuracy N_e versus the grid size h for some constant values of β. Similar remarks to ODEs can be made here. It can be seen that highly accurate results are obtained. The relative L_2 error is reduced to $O(10^{-7})$ for $\beta = 4$. As β increases, the level of accuracy is clearly improved; however, their average rates are only about 3, e.g. 3.51 for $\beta = 1$, 3.45 for $\beta = 2$, 3.22 for $\beta = 3$, and 3.10 for $\beta = 4$, probably due to the use of unstructured nodes. In Figure 4.6, for large β, local rates/slopes are observed to vary with the grid size. Since global approximations result in full matrices, their computations can be very expensive. The global schemes are thus not suitable for large-scale applications. There is a need for having local schemes, which is discussed next.

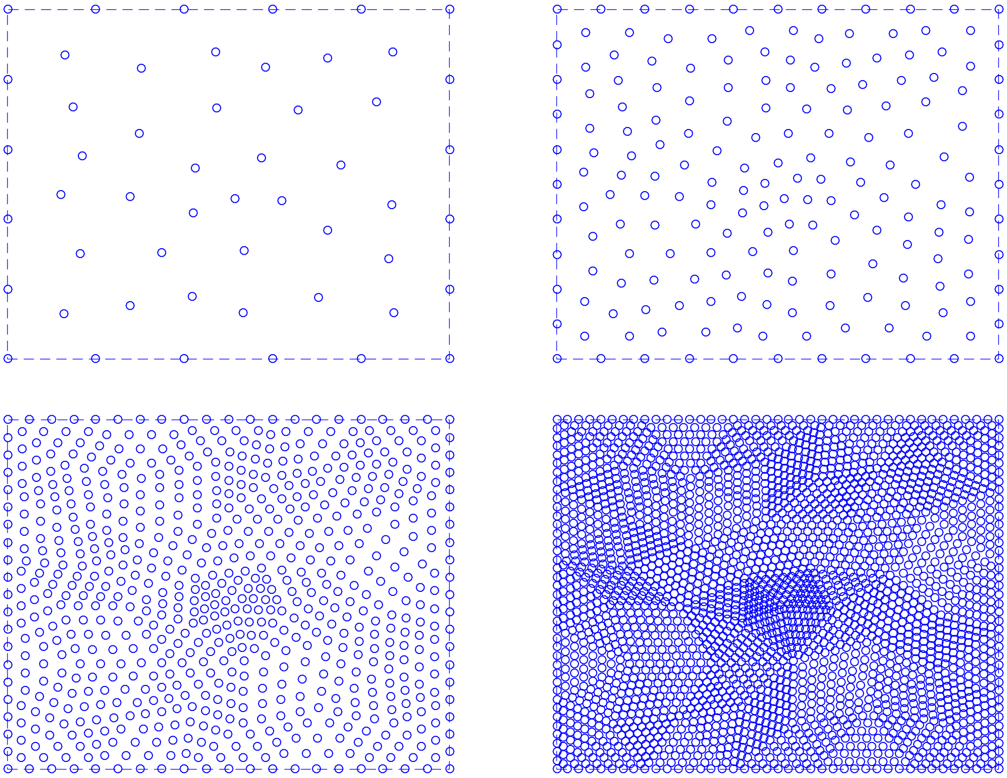


Figure 4.5: Domain discretisations using 52, 185, 697 and 2705 unstructured nodes.

4.4 IRBF Hermite-based method: local scheme

In a local version, only neighbouring nodes are activated for the approximation at a point. Consider a stencil associated with node i. For the Hermite type, some nodes on the stencil are selected to include information about ODE/PDE ($\mathcal{L}f = b$, $\mathcal{L}$ a linear differential operator, b a given function). Let n be the total number of nodes of the stencil and q the number of special nodes just mentioned ($q < n$). A function is approximated as

$$f(\mathbf{x}) = \sum_{j=1}^{n} w_j G(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{j=1}^{q} \bar{w}_j \mathcal{L}^{\bar{\mathbf{x}}_j} G(\|\mathbf{x} - \bar{\mathbf{x}}_j\|) + \sum_{k=1}^{M} v_k p_k(\mathbf{x}), \quad (4.52)$$

where the notation $\mathcal{L}^{\bar{\mathbf{x}}_j}$ means that $\mathcal{L}$ acts on G considered as a function of $\bar{\mathbf{x}}_j$, $\{\bar{\mathbf{x}}_j\}_{j=1}^{q}$ is a subset of $\{\mathbf{x}_j\}_{j=1}^{n}$, and $\{p_k(\mathbf{x})\}_{k=1}^{M}$ is a basis for the M-dimensional space ($\prod_m^d$) of all d-variate polynomials that have degree less than or equal

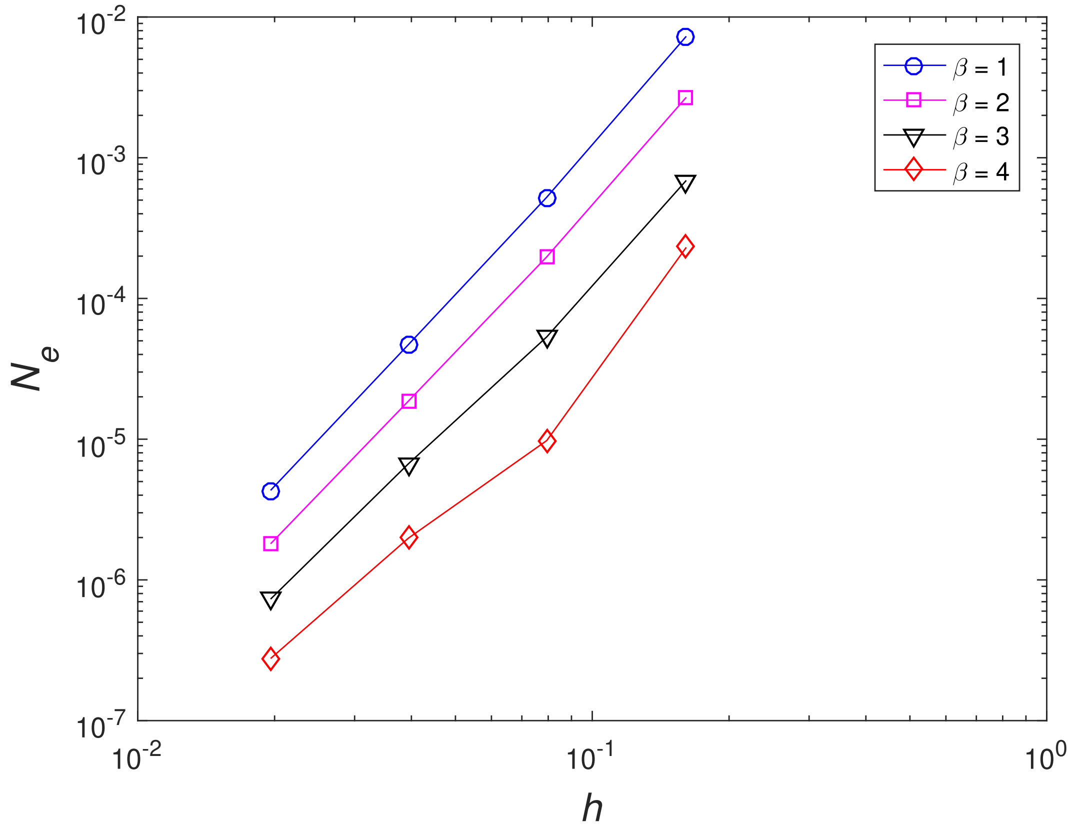


Figure 4.6: PDE, double precision: Grid convergence by the proposed global Hermite-based method for several values of β

to m. To account for the addition of polynomial terms, the following extra constraints are imposed

$$\sum_{j=1}^{n} w_j p_k(\mathbf{x}_j) + \sum_{j=1}^{q} \bar{w}_j \mathcal{L}^{\mathbf{x}} p_k(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}_j} = 0, \quad k = 1, 2, \ldots, M. \tag{4.53}$$

Unlike Lagrange interpolation (function values only), expression (4.52) contains q extra coefficients (i.e. $\{\bar{w}_j\}_{j=1}^{q}$) that allow for the process of converting the

RBF coefficient space into the physical space to take the form

$$\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \\ \mathcal{L}^{\mathbf{x}} f(\bar{\mathbf{x}}_1) \\ \vdots \\ \mathcal{L}^{\mathbf{x}} f(\bar{\mathbf{x}}_q) \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{bmatrix} \mathcal{C}^{11} & \mathcal{C}^{12} & \mathcal{C}^{13} \\ \mathcal{C}^{21} & \mathcal{C}^{22} & \mathcal{C}^{23} \\ \mathcal{C}^{31} & \mathcal{C}^{32} & \mathcal{C}^{33} \end{bmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \\ \bar{w}_1 \\ \vdots \\ \bar{w}_q \\ v_1 \\ \vdots \\ v_M \end{pmatrix}, \tag{4.54}$$

where the first n equations are for function values, the next q equations for derivative values (i.e. ODE/PDE), the last M equations for the extra constraints to account for the addition of polynomial terms, the square matrix on RHS is referred to as a conversion matrix, denoted by $\mathcal{C}$, and

$$\left(\mathcal{C}^{11}\right)_{ij} = G(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad 1 \leq i \leq n, \ 1 \leq j \leq n, \tag{4.55}$$

$$\left(\mathcal{C}^{12}\right)_{ij} = \mathcal{L}^{\bar{\mathbf{x}}_j} G(\|\mathbf{x}_i - \bar{\mathbf{x}}_j\|), \quad 1 \leq i \leq n, \ 1 \leq j \leq q, \tag{4.56}$$

$$\left(\mathcal{C}^{13}\right)_{ij} = p_j(\mathbf{x}_i), \quad 1 \leq i \leq n, \ 1 \leq j \leq M, \tag{4.57}$$

$$\left(\mathcal{C}^{21}\right)_{ij} = \mathcal{L}^{\mathbf{x}} G(\|\mathbf{x} - \mathbf{x}_j\|)_{\mathbf{x}=\bar{\mathbf{x}}_i}, \quad 1 \leq i \leq q, \ 1 \leq j \leq n, \tag{4.58}$$

$$\left(\mathcal{C}^{22}\right)_{ij} = \mathcal{L}^{\mathbf{x}} \mathcal{L}^{\bar{\mathbf{x}}_j} G(\|\mathbf{x} - \bar{\mathbf{x}}_j\|)_{\mathbf{x}=\bar{\mathbf{x}}_i}, \quad 1 \leq i \leq q, \ 1 \leq j \leq q, \tag{4.59}$$

$$\left(\mathcal{C}^{23}\right)_{ij} = \mathcal{L}^{\mathbf{x}} p_j(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}_i}, \quad 1 \leq i \leq q, \ 1 \leq j \leq M, \tag{4.60}$$

$$\left(\mathcal{C}^{31}\right)_{ij} = p_i(\mathbf{x}_j), \quad 1 \leq i \leq M, \ 1 \leq j \leq n, \tag{4.61}$$

$$\left(\mathcal{C}^{32}\right)_{ij} = \mathcal{L}^{\mathbf{x}} p_i(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}_j}, \quad 1 \leq i \leq M, \ 1 \leq j \leq q, \tag{4.62}$$

$$\left(\mathcal{C}^{33}\right)_{ij} = 0, \quad 1 \leq i \leq M, \ 1 \leq j \leq M. \tag{4.63}$$

When the RBF approximations are not augmented with the polynomial terms, the conversion system reduces to

$$\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \\ \mathcal{L}^{\mathbf{x}} f(\bar{\mathbf{x}}_1) \\ \vdots \\ \mathcal{L}^{\mathbf{x}} f(\bar{\mathbf{x}}_q) \end{pmatrix} = \begin{bmatrix} \mathcal{C}^{11} & \mathcal{C}^{12} \\ \mathcal{C}^{21} & \mathcal{C}^{22} \end{bmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \\ \bar{w}_1 \\ \vdots \\ \bar{w}_q \end{pmatrix}, \tag{4.64}$$

It can be seen that $\mathcal{C}$ is a symmetric matrix of dimensions $[n+q+M, n+q+M]$ if the polynomial terms are included and of dimensions $[n+q, n+q]$ if the polynomial terms are excluded. Making use of (4.54) and (4.64), a function and its derivatives at a point on the stencil can be expressed in terms of function values at $\{\mathbf{x}_j\}_{j=1}^{n}$, which are nodal unknowns to be found, and derivative values at $\{\bar{\mathbf{x}}_j\}_{j=1}^{q}$, which can be derived from the ODE/PDE. By collocating the ODE/PDE at the interior grid nodes, and then replacing the obtained nodal derivative values with nodal variable values on their associated stencils, the ODE/PDE is transformed into a set of algebraic equations, which can be solved for the values of f at the grid nodes.

4.4.1 ODEs

We apply the methods to the following second-order ODE

$$d^2f/dx^2 = \exp(-40x)\left(1500\sin(10x) - 800\cos(10x)\right), \quad 0 \leq x \leq 1,$$

subject to Dirichlet boundary conditions. The exact solution can be verified to be $f_e(x) = \sin(10x)\exp(-40x)$. The domain is represented by sets of equi-spaced nodes. A stencil associated with node $\mathbf{x}_i$ is proposed to have 3 nodes. We take the distance h as the grid size. In contrast to the global approximation case, the value of β here can be chosen to be much larger.

Figure 4.7 shows the effects of β on the solution accuracy N_e. Results by the DRBF Hermite-based method are also included. For both methods, systems are tri-diagonal and of the same dimensions. It can be seen that IRBF outperforms DRBF over a wide range of β. As β increases, the RBF approximations can be more accurate but its interpolation matrix condition number also grows quickly, making the computed error N_e fluctuating at large values of β. One can bypass this issue by using extended precision in computation. In (Mai-Duy et al., 2017), numerical investigations indicated that the condition number of the conversion matrix grows much faster than that of the final system matrix. Here, we only employ extended precision for constructing and inverting small conversion matrices (other computational parts including the solving of the final system of equations are conducted using double precision). The obtained results are also depicted in Figure 4.7. At low values of β, where the matrix is not ill-conditioned, double and extended precision basically yield the same errors. At large values of β, by extending the calculation precision, fluctuations in the computed error are eliminated.

Table 4.1 displays the computed solutions by several numerical methods. When compared to the classical central difference scheme, the compact approximations produce much more accurate results. Both compact FD (Collatz, 1966) and IRBF methods are able to yield high rates of convergence (about fourth-order accuracy). To this problem, since its exact solution is available, the best values of β can be determined and their corresponding solutions are also included in the table, showing that: (i) RBF accuracy can be further improved by varying β; and (ii) the simple formula (4.35) can work well for relatively-coarse grids.

4.4.2 PDEs

Like the global version, the present local methods are also meshless. However, our attention will be focused on the case of using a Cartesian grid to represent

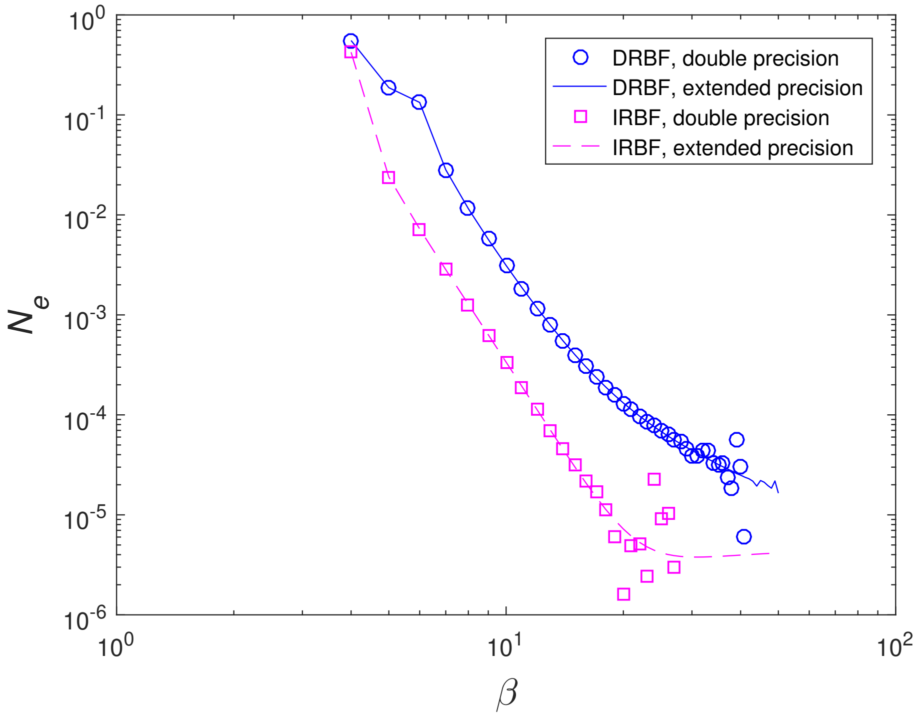


Figure 4.7: Second-order ODE, $N = 501$, double/extended precision: Effects of the RBF width (represented through β) on the solution accuracy by the local Hermite-based methods. Both DRBF and IRBF schemes are examined over a wide range of β

.

the problem domain. The main reason for us to pursue this kind of discretisation lies in its economic pre-processing, easy implementation and its ability to work with non-rectangular domains.

For 2D problems, a stencil associated with node $\mathbf{x}_i$ is proposed to have 9 nodes

$$\begin{bmatrix} \mathbf{x}_3 & \mathbf{x}_6 & \mathbf{x}_9 \\ \mathbf{x}_2 & \mathbf{x}_5 & \mathbf{x}_8 \\ \mathbf{x}_1 & \mathbf{x}_4 & \mathbf{x}_7 \end{bmatrix} \tag{4.65}$$

where the fifth node (i.e. $\mathbf{x}_5$) is a node i in a global numbering. Four nodes, 2, 4, 6 and 8 (i.e. nodes nearer to the stencil centre), are selected to include the PDE information.

Table 4.1: ODE, double precision: Relative L_2 errors of the computed solutions. Compact approximations outperform those based on the classical central differences. Both compact FD and IRBF schemes are able to yield high rates of convergence with respect to grid refinement. Given an analytic form of the solution f, the best values of β can be determined numerically and their corresponding solutions are also included for comparison purposes.

$N_x \times N_y$	FDM	Compact FDM	Compact IRBF	Compact IRBF
			$\beta = 20$	optimal β
31×31	1.9807e+00	3.4981e-01	3.4935e-01	3.2995e-01
51×51	7.4623e-01	4.5186e-02	4.5034e-02	4.0292e-02
71×71	3.8853e-01	1.1845e-02	1.1760e-02	1.0211e-02
91×91	2.3735e-01	4.3536e-03	4.3270e-03	3.6870e-03
111×111	1.5973e-01	1.9559e-03	1.9408e-03	1.6401e-03
131×131	1.1473e-01	1.0043e-03	9.9094e-04	8.3703e-04
151×151	8.6348e-02	5.6717e-04	5.4784e-04	4.7088e-04
171×171	6.7319e-02	3.4403e-04	3.2732e-04	2.8486e-04
191×191	5.3945e-02	2.2060e-04	2.1582e-04	1.8228e-04
211×211	4.4191e-02	1.4788e-04	1.3573e-04	1.2193e-04
	$O(h^{1.96})$	$O(h^{3.99})$	$O(h^{4.02})$	$O(h^{4.07})$

For a stencil associated with an interior node that is near an irregular boundary (e.g. curved boundary), it is proposed that the stencil consists of regular and irregular nodes (Figure 4.10). Regular nodes are simply the intersection points of the stencil grid lines, while irregular nodes are generated from the intersection of the boundary and the stencil grid lines. As a result, for boundary stencils, the number of nodes are typically greater than 9. The imposition of information about PDE is also implemented at side nodes on the horizontal and vertical grid lines (i.e. four nodes).

Poisson's equation

A PDE to be employed is Poisson's equation, where its driving function and Dirichlet boundary conditions are derived from the following solution $f_e(x, y) = e^{-6(x+y)} \cos(2\pi x) \sin(2\pi y)$. Both rectangular and non-rectangular domains are

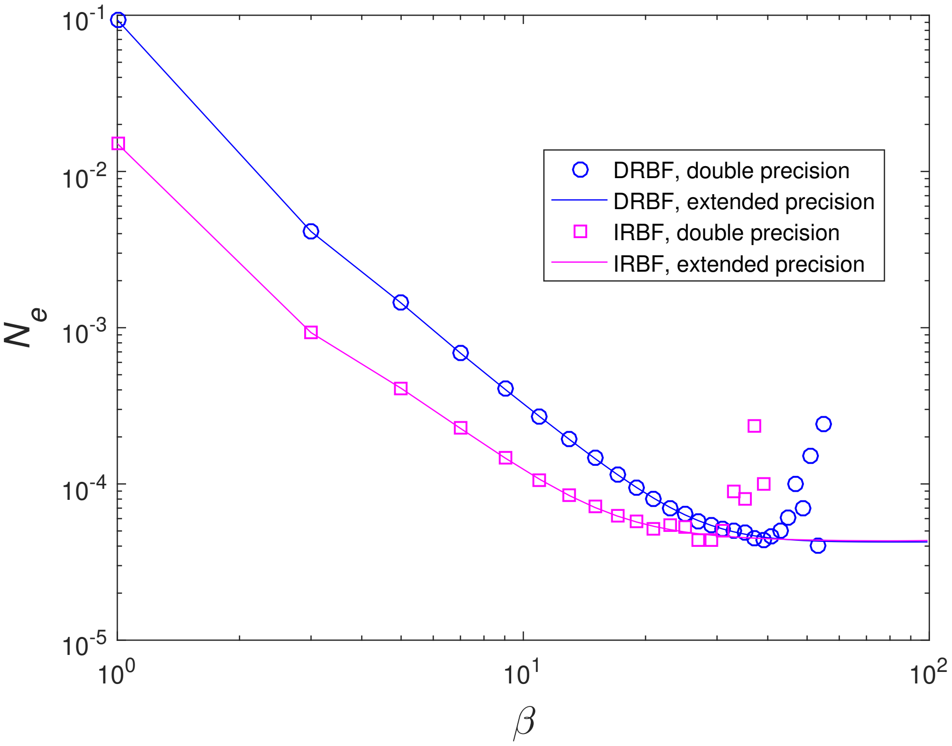


Figure 4.8: Poisson's equation, rectangular domain, 21 × 21, double/extended precision: Effects of the RBF width (represented through β)
on the solution accuracy by the local Hermite-based methods. Both DRBF and IRBF schemes are examined over a wide range of β.

considered. For the latter, a quarter of a circle is chosen (Figure 3.11).

The effects of the MQ width on the solution accuracy by the IRBF and DRBF Hermite-based methods are displayed in Figure 4.8 for the rectangular domain, and in Figure 4.12 for the non-rectangular domain. It can be seen that the IRBF solutions are generally more accurate than DRBF solutions over a wide range of β. As expected, at large values of β, there are some fluctuations in the computed error N_e. To make the calculation stable, we employ extended precision in forming the conversion matrix and computing its inverse. Since other computational parts are carried out with double precision, a full range of β is explored with a reasonable computational cost. With extended precision, as shown in the two figures, fluctuations no longer occur in the computed error at large values of β.

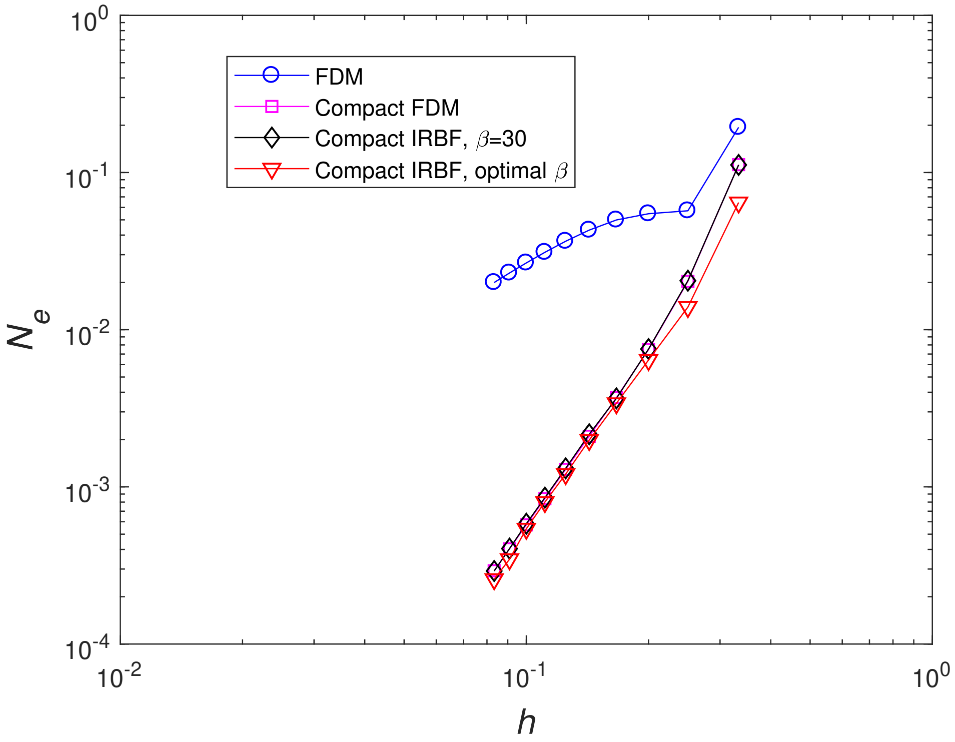


Figure 4.9: Poisson's equation, rectangular domain, $\beta = 30$, double precision: Comparison of accuracy between the compact IRBF scheme and FDMs (central difference (CD) and compact). The computed solution converges apparently as $O(h^{1.34})$, $O(h^{4.11})$ and $O(h^{4.11})$ for the CD, compact FD and compact IRBF methods, respectively. The IRBF results at the best values of β are also included for comparison purposes.

In Figure 4.9, results by the compact IRBF, central difference and compact FD (Collatz, 1966) methods are displayed. Similar to second-order ODEs, the compact approximations for PDEs outperform those based on the central differences. The compact IRBF and FD methods yield high rates of convergence (about fourth-order accuracy). Exploiting the exact solution, the best values of β can be found. It can be seen that the use of a fixed β for relatively coarse grids can lead to results that are very close to those corresponding to the best values of β. Since the present (integrated) interpolation process is based on analytic formulas as shown in Section 2.2, it is expected that this type of interpolation does not affect much the computational cost. As can be seen in Figure 4.9, compact FDM and compact IRBF achieve a similar level of accuracy. In relation to computational time, the present compact IRBF consumes 0.936(s),

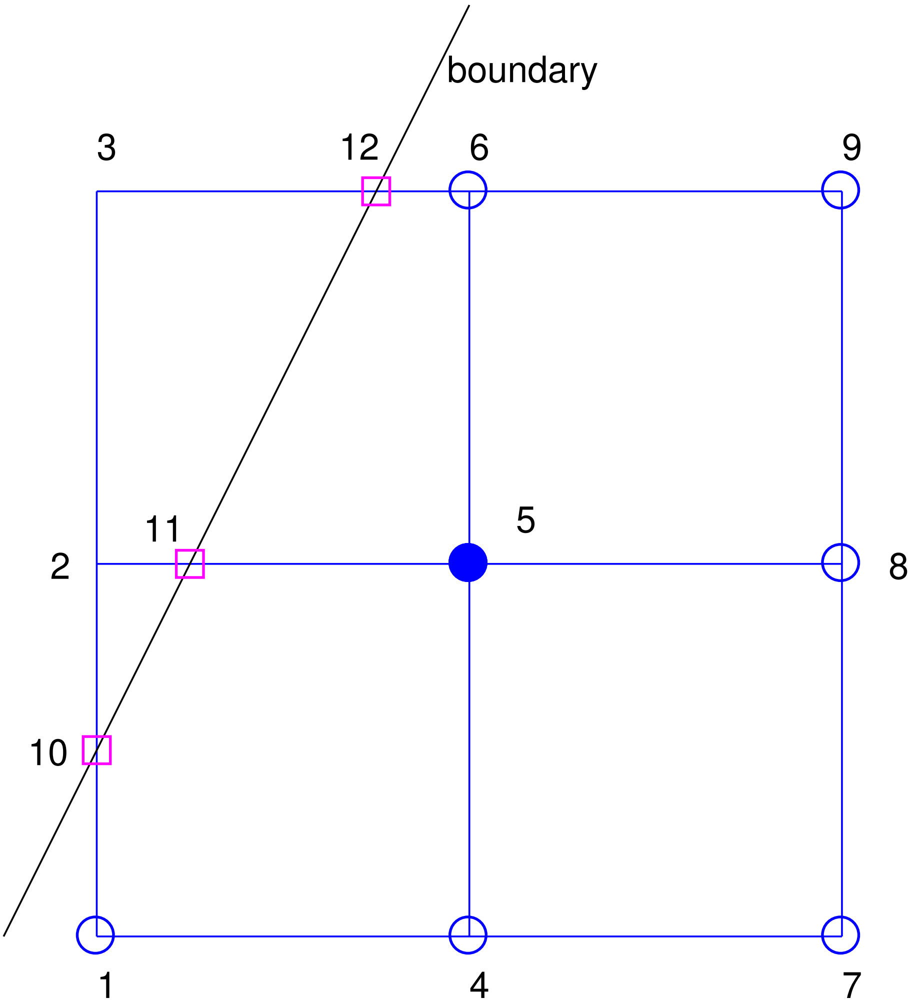


Figure 4.10: Non-rectangular domain: a schematic diagram for boundary stencils. The stencil, which is associated with node 5, consists of regular nodes: (1,4,5,6,7,8,9) and irregular nodes: (10,11,12). The PDE is imposed at side nodes: (11,4,6,8).

and compact FDM takes 0.933(s) (CPU Intel Xeon E31245 3.3 Mhz) for the numerical solutions on grid sizes of $4 \times 4, 5 \times 5, \cdots, 30 \times 30$. As expected, the computational costs by the two methods are shown to be similar.

For non-rectangular domains, the problem domain is simply embedded in a rectangle that is discretised using Cartesian grids of $7 \times 7, 9 \times 9, \cdots, 31 \times 31$. Table 4.2 displays the results obtained by the compact IRBF scheme. The solution converges as $O(h^{3.88})$ for $\beta = 16$, $O(h^{3.99})$ for $\beta = 18$ and $O(h^{4.09})$ for $\beta = 20$.

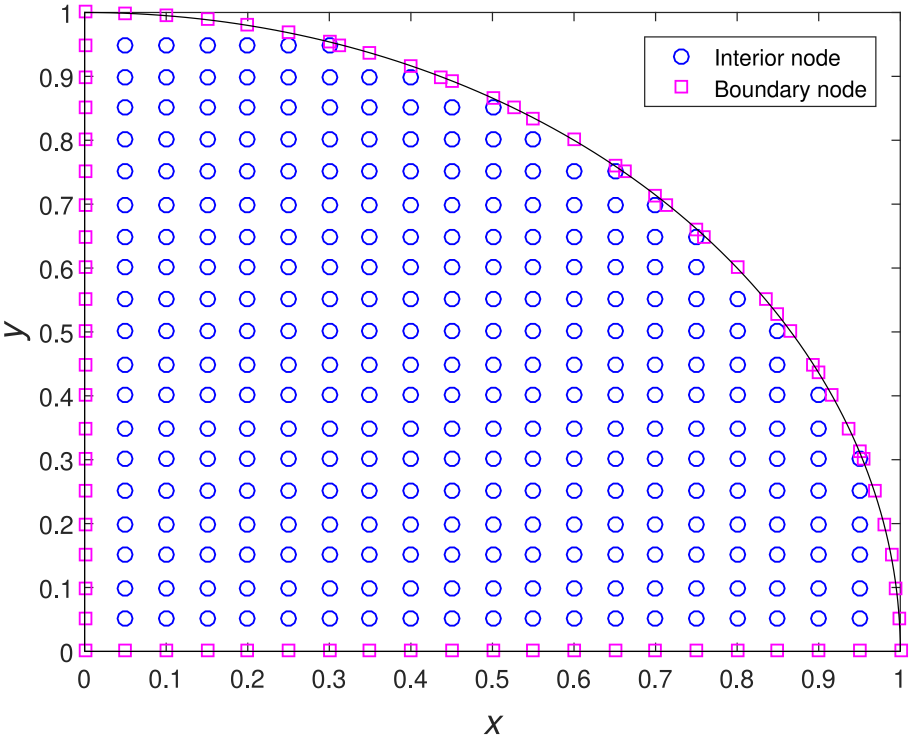


Figure 4.11: Non-rectangular domain: The problem domain is embedded in a rectangle that is then discretised by a Cartesian grid. Interior nodes are grid nodes within the problem domain. Boundary nodes are generated by the intersection of the grid lines and the boundary of the domain.

Convection-diffusion equation

We test the proposed local method with the following steady-state convection-diffusion equation and boundary conditions

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} - P_e \frac{\partial f}{\partial x} = 0, \quad 0 \le x, y \le 1, \tag{4.66}$$

$$f(x,0) = f(x,1) = 0, \quad 0 \le x \le 1, \tag{4.67}$$

$$f(0,y) = \sin(\pi y), \; f(1,y) = 2\sin(\pi y), \quad 0 \le y \le 1, \tag{4.68}$$

where P_e is the Péclet number. The exact solution to this problem is given by

$$f_e(x,y) = \exp(P_e x/2)\sin(\pi y)\left[2\exp(-P_e/2)\sinh(\sigma x) + \sinh(\sigma(1-x))\right]/\sinh(\sigma),$$

Table 4.2: PDE, non-rectangular domain, double precision: Grid convergence by the local IRBF Hermite-based scheme for several RBF widths .

	Relative L_2 error		
$N_x \times Ny$	$\beta = 16$	$\beta = 18$	$\beta = 20$
7×7	1.8576e-02	1.8407e-02	1.8324e-02
9×9	4.0162e-03	3.9526e-03	3.9217e-03
11×11	1.4255e-03	1.3987e-03	1.3714e-03
13×13	6.5265e-04	6.2664e-04	6.1840e-04
15×15	3.5023e-04	3.3471e-04	3.2295e-04
17×17	2.1138e-04	1.9490e-04	1.8655e-04
19×19	1.3937e-04	1.2728e-04	1.2112e-04
21×21	9.8276e-05	8.8445e-05	8.0223e-05
23×23	7.3275e-05	6.2924e-05	5.8075e-05
25×25	5.6198e-05	4.8013e-05	4.4473e-05
27×27	4.4900e-05	3.8746e-05	3.7525e-05
29×29	3.6775e-05	3.1344e-05	2.4388e-05
31×31	3.0377e-05	2.6381e-05	2.2324e-05
	$O(h^{3.88})$	$O(h^{3.99})$	$O(h^{4.09})$

(4.69)

where $\sigma = \sqrt{\pi^2 + P_e^2/4}$. As P_e increases, the boundary layer will be formed. Its gradient becomes very steep at large P_e values, presenting a great challenge for any numerical simulation. We simply employ uniform grids to represent the problem domain. The optimal RBF width is observed to occur earlier with respect to the RBF width when P_e increases from 10 to 100 (Figure 4.13). In contrast to problems whose solutions are smooth, the most accurate approximation for convection-dominated problems takes place at relatively-low values of the RBF width, where the RBF system is known to be stable. Note that all smooth curves depicted here are obtained with double-precision computations. By simply taking $\beta = \{10, 8, 6, 4\}$ for $P_e = \{10, 20, 40, 80\}$, respectively, a fast rate of convergence (i.e. about 4) is achieved (Figure 4.14). Figure 4.15 displays the present RBF solutions for several P_e values. It can be seen that they are all captured very well. At high P_e values, there are no oscillations in the solution

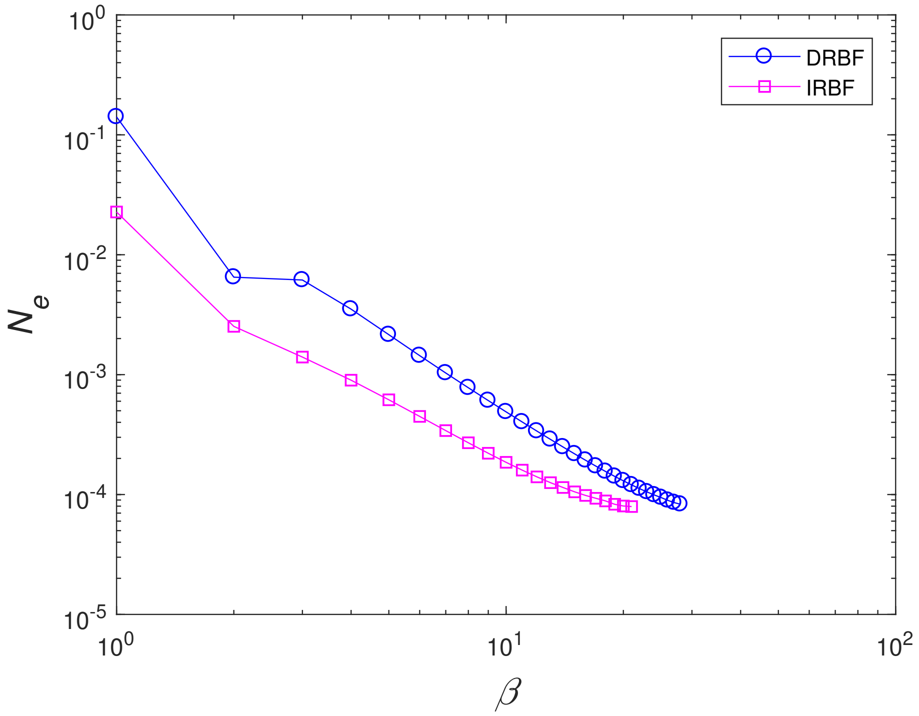


Figure 4.12: Poisson's equation, nonrectangular domain, double precision: Effects of the RBF width on the solution accuracy by the local Hermite-based methods.

near the boundary layer.

4.5 Concluding remarks

In this chapter, we have introduced IRBFs into the Hermite interpolation method for the numerical solution of ODEs/PDEs. Its main purpose is to yield a new strong (collocation) form of IRBF whose interpolation matrices are symmetric and non-singular. Several schemes based on global and local approximations for rectangular and non-rectangular domains are presented. The extended precision approach is utilised to extend the working range of the IRBF width for a given grid size. Numerical examples show an improvement in accuracy achieved over conventional compact IRBF Hermite-based methods. The local version is a preferred option for the handling of large-scale problems as

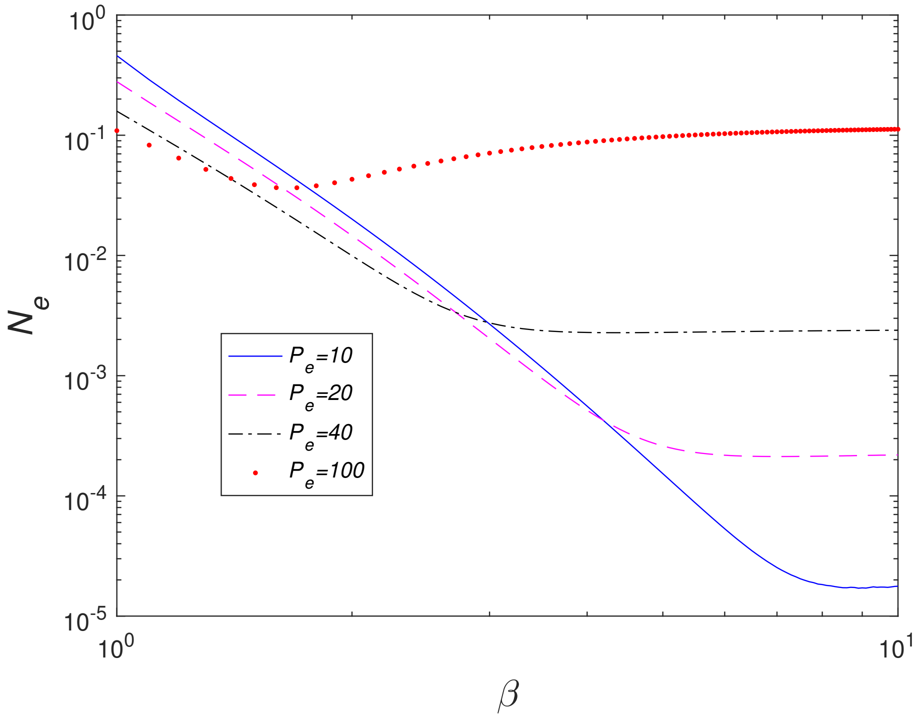


Figure 4.13: Convection-diffusion equation, 33 × 33, double precision: Effects of the RBF width on the solution accuracy for several Péclet numbers by the proposed local Hermite-based method.

it possesses several attractive features including: (i) sparse system matrix; (ii) fast convergence rate; and (iii) its ability to also work with large values of the RBF width with a relatively low computational cost. Highly accurate results are obtained using relatively coarse grids.

Appendix

The following equation is utilised to derive the limit of the fourth-order cross derivative of function G as $r \to 0$

$$\nabla^4 G = \nabla^2 \nabla^2 G, \tag{4.70}$$

or

$$\frac{\partial^4 G}{\partial x^4} + 2\frac{\partial^4 G}{\partial x^2 \partial y^2} + \frac{\partial^4 G}{\partial y^4} = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\left(\frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}\right). \tag{4.71}$$

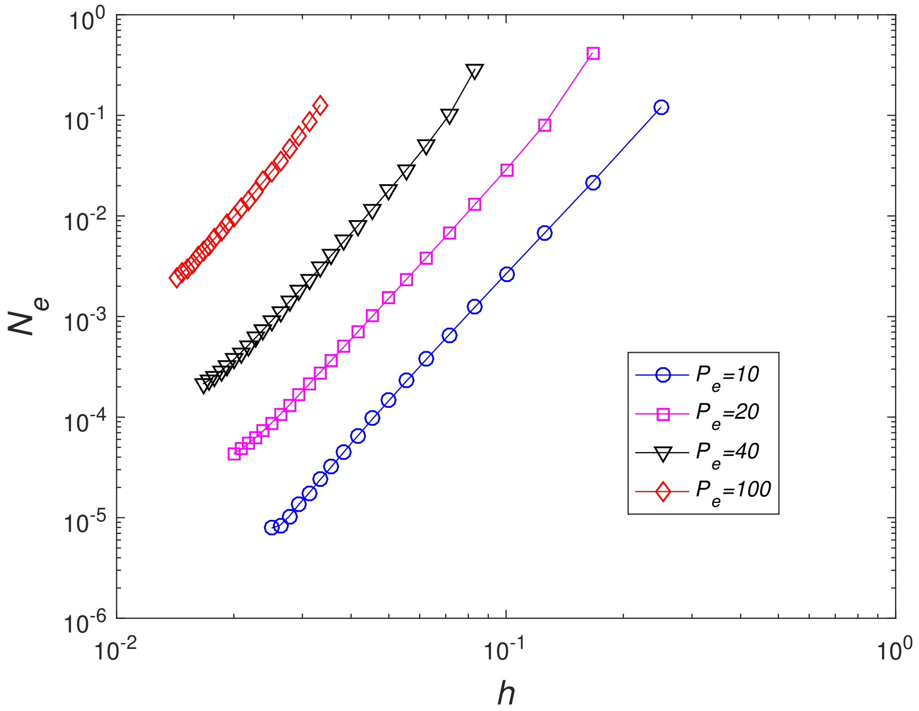


Figure 4.14: Convection-diffusion equation, double precision: Effects of the grid size on the solution accuracy for several Péclet numbers by the proposed local Hermite-based method. Values of β used are $\{10, 8, 6, 4\}$ for $P_e = \{10, 20, 40, 100\}$, respectively. The solution converges as $O(h^{4.24})$ for $P_e = 10$, $O(h^{4.23})$ for $P_e = 20$, $O(h^{4.34})$ for $P_e = 40$ and $O(h^{4.61})$ for $P_e = 100$.

Taking into account

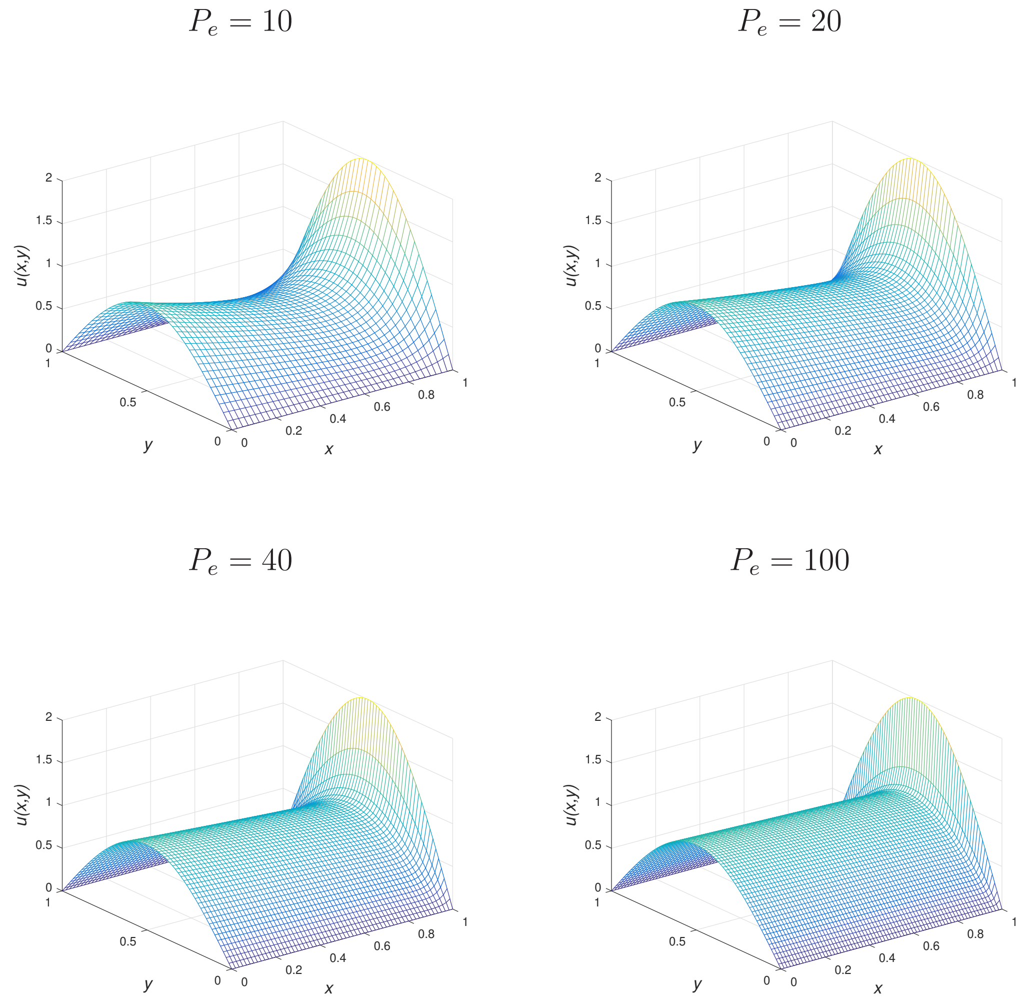
$$\left(\frac{\partial r}{\partial x}\right)^2 + \left(\frac{\partial r}{\partial y}\right)^2 = 1, \quad \frac{\partial^2 r}{\partial x^2} + \frac{\partial^2 r}{\partial y^2} = \frac{1}{r}, \tag{4.72}$$

the RHS of (4.71) can be rewritten in terms of derivatives of G with respect to r only, and the equation becomes

$$\frac{\partial^4 G}{\partial x^4} + 2\frac{\partial^4 G}{\partial x^2 \partial y^2} + \frac{\partial^4 G}{\partial y^4} = \frac{d^4 G}{dr^4} + \frac{2}{r}\frac{d^3 G}{dr^3} - \frac{1}{r^2}\frac{d^2 G}{dr^2} + \frac{1}{r^3}\frac{dG}{dr}, \tag{4.73}$$

from there, as $r \to 0$, one can acquire

$$\frac{\partial^4 G}{\partial x^2 \partial y^2} \to -\frac{1}{a^3} \quad \text{for DRBF,}$$
$$\frac{\partial^4 G}{\partial x^2 \partial y^2} \to \frac{a}{3} \quad \text{for IRBF.}$$


$P_e = 10$

$P_e = 20$

$P_e = 40$

$P_e = 100$

Figure 4.15: Convection-diffusion equation, double precision: Numerical solutions by the proposed local Hermite-based method using grid of 41 × 41 for $P_e = 10$, 51 × 51 for $P_e = 20$, 61 × 61 for $P_e = 40$ and 71 × 71 for $P_e = 100$

Chapter 5

Compact IRBF stencils for time approximations

The CIRBF approach used for space discretisation has been reported in the previous chapters as having the ability to produce a high level of accuracy. However, for unsteady problems, if low order approximations (e.g. FD) are employed for time derivative terms, the overall accuracy of numerical solutions is still of a low order whatever the accuracy of the spatial discretisation scheme. Therefore, in this chapter, a new numerical procedure, based on high order approximation (CIRBFs) to discretise both space and time is presented.

For space discretisations, compact five-point IRBF stencils are utilised. For time discretisations, a two-point IRBF scheme is proposed, where the time derivative is approximated in terms of not only nodal function values at the current and previous time levels, but also nodal derivative values at the previous time level. This allows functions other than a linear one to also be captured well on a time step. The use of the RBF width as an additional parameter to enhance the approximation quality with respect to time is also explored. Various kinds of test problems of heat transfer and fluid flows are conducted to demonstrate the attractiveness of the present compact approximations.

5.1 Introduction

Temporal discretisation is a mathematical technique applied to transient problems that occur in the fields of applied physics and engineering, which require discretising the governing equations in both space and time. Such problems are unsteady (e.g. flow problems) and, therefore, require solutions in which position varies as a function of time. In using RBFs to solve differential problems, the time derivative terms are usually discretised by means of low-order FDs, for which small time steps are typically required.

In this study, we propose a discretisation procedure based on compact IRBF stencils only for time-dependent heat and fluid flow problems in two dimensions. An IRBF stencil is of two nodes and five nodes (Mai-Duy and Tran-Cong, 2013) for time and space discretisations, respectively. The remainder of the chapter is organised as follows. Section 5.2 gives a brief review of IRBFs and their compact forms for space discretisations. Section 5.3 describes a new compact two-point approximation based on IRBFs for time discretisations, and a numerical procedure based on compact IRBF stencils only for solving time-dependent differential problems. Numerical results are presented in Section 5.4. Section 5.5 concludes the chapter.

5.2 Compact approximation scheme

IRBFs have been used to construct the approximations on Cartesian grids representing a domain of rectangular/non-rectangular shape (Mai-Duy and Tran-Cong, 2013, 2007). Advantages of this approach lie in its economic preprocessing. Consider a domain that is embedded in a Cartesian grid as shown in Figure 5.1. Grid points outside the domain (external points) and the internal points that fall very close - within a small distance - to the boundary, are removed. The remaining grid points are taken to be the interior nodes. The boundary nodes are points that are generated by the intersection of the grid lines with the boundaries. In this work, second order differential problems are considered and for a space discretisation a 5-point stencil associated with node (i,j) is employed with nodes being locally numbered from left to right and from bottom to top $((i,j) \equiv 3)$ (Figure 5.1). Derivatives of the dependent variable f in the x and y directions are approximated by IRBFs along the lines defined by $1-3-5$ and $2-3-4$, respectively. One can utilise the integration constants in the IRBF formulation to incorporate some nodal derivative values in the approximations. In the x direction, evaluation of (2.29) at (x_1, x_3, x_5) and of (2.27) at (x_1, x_5) using $q=2$ result in

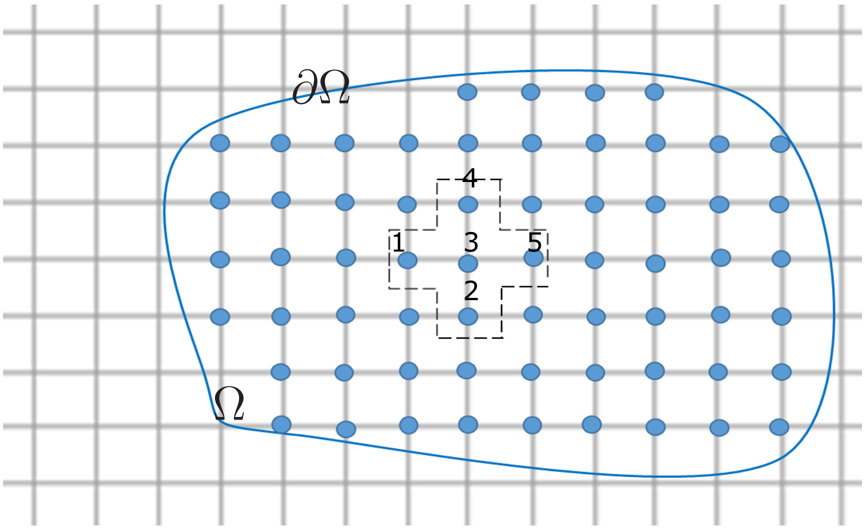


Figure 5.1: Domain of interest and its Cartesian-grid representation.

$$\widetilde{f} = \underbrace{\begin{bmatrix} \mathbf{I} \\ \mathbf{B} \end{bmatrix}}_{\mathcal{C}} \widetilde{w}, \tag{5.1}$$

where

$$\widetilde{f} = \begin{pmatrix} f_1 & f_3 & f_5 & \frac{\partial^2 f_1}{\partial x^2} & \frac{\partial^2 f_5}{\partial x^2} \end{pmatrix}^T, \tag{5.2}$$

$$\widetilde{w} = \begin{pmatrix} w_{[x]1} & w_{[x]3} & w_{[x]5} & c_{[x]1} & c_{[x]2} \end{pmatrix}^T, \tag{5.3}$$

$$\mathbf{I} = \begin{bmatrix} \mathcal{I}^{(0)}_{[x]1}(x_1) & \mathcal{I}^{(0)}_{[x]3}(x_1) & \mathcal{I}^{(0)}_{[x]5}(x_1) & x_1 & 1 \\ \mathcal{I}^{(0)}_{[x]1}(x_3) & \mathcal{I}^{(0)}_{[x]3}(x_3) & \mathcal{I}^{(0)}_{[x]5}(x_3) & x_3 & 1 \\ \mathcal{I}^{(0)}_{[x]1}(x_5) & \mathcal{I}^{(0)}_{[x]3}(x_5) & \mathcal{I}^{(0)}_{[x]5}(x_5) & x_5 & 1 \end{bmatrix},$$

and

$$\mathbf{B} = \begin{bmatrix} \mathcal{I}^{(2)}_{[x]1}(x_1) & \mathcal{I}^{(2)}_{[x]3}(x_1) & \mathcal{I}^{(2)}_{[x]5}(x_1) & 0 & 0 \\ \mathcal{I}^{(2)}_{[x]1}(x_5) & \mathcal{I}^{(2)}_{[x]3}(x_5) & \mathcal{I}^{(2)}_{[x]5}(x_5) & 0 & 0 \end{bmatrix}.$$

The system (5.1) can be solved for the unknown coefficient vector $\widetilde{w}$, resulting in

$$\widetilde{w} = \mathcal{C}^{-1}\widetilde{f}, \tag{5.4}$$

where $\mathcal{C}^{-1}$ is the inverse of $\mathcal{C}$.

Expressions for computing f and its derivatives at point x, where $x_1 \leq x \leq x_5$, can then be obtained by substituting (5.4) into (2.29), (2.28) and (2.27) with $q = 2$

$$f(x) = \left[\begin{array}{ccccc} \mathcal{I}^{(0)}_{[x]1}(x) & \mathcal{I}^{(0)}_{[x]3}(x) & \mathcal{I}^{(0)}_{[x]5}(x) & x & 1 \end{array} \right] \mathcal{C}^{-1}\widetilde{f}, \tag{5.5}$$

$$\frac{\partial f(x)}{\partial x} = \left[\begin{array}{ccccc} \mathcal{I}^{(1)}_{[x]1}(x) & \mathcal{I}^{(1)}_{[x]3}(x) & \mathcal{I}^{(1)}_{[x]5}(x) & 1 & 0 \end{array} \right] \mathcal{C}^{-1}\widetilde{f}, \tag{5.6}$$

$$\frac{\partial^2 f(x)}{\partial x^2} = \left[\begin{array}{ccccc} \mathcal{I}^{(2)}_{[x]1}(x) & \mathcal{I}^{(2)}_{[x]3}(x) & \mathcal{I}^{(2)}_{[x]5}(x) & 0 & 0 \end{array} \right] \mathcal{C}^{-1}\widetilde{f}, \tag{5.7}$$

which can be rewritten as

$$f\left(x\right) = \phi_1\left(x\right) f_1 + \phi_3\left(x\right) f_3 + \phi_5\left(x\right) f_5 + \bar{\phi}_1\left(x\right) \frac{\partial^2 f_1}{\partial x^2} + \bar{\phi}_5\left(x\right) \frac{\partial^2 f_5}{\partial x^2}, \tag{5.8}$$

$$\frac{\partial f\left(x\right)}{\partial x} = \frac{d\phi_1\left(x\right)}{dx} f_1 + \frac{d\phi_3\left(x\right)}{dx} f_3 + \frac{d\phi_5\left(x\right)}{dx} f_5 + \frac{d\bar{\phi}_1\left(x\right)}{dx}\frac{\partial^2 f_1}{\partial x^2} + \frac{d\bar{\phi}_5\left(x\right)}{dx}\frac{\partial^2 f_5}{\partial x^2}, \tag{5.9}$$

$$\frac{\partial^2 f(x)}{\partial x^2} = \frac{d^2\phi_1(x)}{dx^2} f_1 + \frac{d^2\phi_3(x)}{dx^2} f_3 + \frac{d^2\phi_5(x)}{dx^2} f_5 + \frac{d^2\bar{\phi}_1(x)}{dx^2}\frac{\partial^2 f_1}{\partial x^2} + \frac{d^2\bar{\phi}_5(x)}{dx^2}\frac{\partial^2 f_5}{\partial x^2}. \tag{5.10}$$

At $x = x_3$, they reduce to

$$\frac{\partial f_3}{\partial x} = \mu_1 f_1 + \mu_3 f_3 + \mu_5 f_5 + \bar{\mu}_1 \frac{\partial^2 f_1}{\partial x^2} + \bar{\mu}_5 \frac{\partial^2 f_5}{\partial x^2}, \tag{5.11}$$

$$\frac{\partial^2 f_3}{\partial x^2} = \eta_1 f_1 + \eta_3 f_3 + \eta_5 f_5 + \bar{\eta}_1 \frac{\partial^2 f_1}{\partial x^2} + \bar{\eta}_5 \frac{\partial^2 f_5}{\partial x^2}, \tag{5.12}$$

where $\mu_1 = d\phi_1(x_3)/dx$, $\mu_3 = d\phi_3(x_3)/dx$, $\mu_5 = d\phi_5(x_3)/dx$, $\bar{\mu}_1 = d\bar{\phi}_1(x_3)/dx$, $\bar{\mu}_5 = d\bar{\phi}_5(x_3)/dx$, $\eta_1 = d^2\phi_1(x_3)/dx^2$, $\eta_3 = d^2\phi_3(x_3)/dx^2$, $\eta_5 = d^2\phi_5(x_3)/dx^2$, $\bar{\eta}_1 = d^2\bar{\phi}_1(x_3)/dx^2$, and $\bar{\eta}_5 = d^2\bar{\phi}_5(x_3)/dx^2$.

Similarly, on the line $2-3-4$, one obtains

$$\frac{\partial f_3}{\partial y} = \nu_2 f_2 + \nu_3 f_3 + \nu_4 f_4 + \bar{\nu}_2 \frac{\partial^2 f_2}{\partial y^2} + \bar{\nu}_4 \frac{\partial^2 f_4}{\partial y^2}, \tag{5.13}$$

$$\frac{\partial^2 f_3}{\partial y^2} = \theta_2 f_2 + \theta_3 f_3 + \theta_4 f_4 + \bar{\theta}_2 \frac{\partial^2 f_2}{\partial y^2} + \bar{\theta}_4 \frac{\partial^2 f_4}{\partial y^2}. \tag{5.14}$$

With nodal derivative values being approximated in the form of (5.11), (5.12), (5.13) and (5.14), collocating the ODE/PDE at grid nodes will lead to a sparse system matrix, of which each row has only five entries. Note that the nodal derivative values on the right hand side of (5.11)-(5.14) can be treated as known quantities.

5.3 Proposed IRBF-based method

5.3.1 An IRBF-based two-point time discretisation scheme

In the proposed scheme, the variation of the dependent variable f on each interval (time step) defined by two points, t_{k-1} and t_k, is represented by IRBFs.

Without loss of generality, we consider the following parabolic PDE

$$\frac{\partial f}{\partial t}(x,y,t) - \left(\frac{\partial^2 f}{\partial x^2}(x,y,t) + \frac{\partial^2 f}{\partial y^2}(x,y,t)\right) = b(x,y,t), \tag{5.15}$$

defined on the domain Ω and subjected to initial values and boundary conditions. In (5.15), b is a given function (the source). Using the conventional finite difference method, one can reduce the PDE to

$$\frac{f_{ij}^{k} - f_{ij}^{k-1}}{\Delta t} - \lambda\left(\frac{\partial^2 f_{ij}^{k}}{\partial x^2} + \frac{\partial^2 f_{ij}^{k}}{\partial y^2}\right) - (1-\lambda)\left(\frac{\partial^2 f_{ij}^{k-1}}{\partial x^2} + \frac{\partial^2 f_{ij}^{k-1}}{\partial y^2}\right) = b_{ij}^{k-1+\lambda}, \tag{5.16}$$

where the subscript ij is used to denote the function at grid node (i,j), the superscript k the function evaluated at the time level t_k, $\Delta t = t_k - t_{k-1}$, and $\lambda = 0$ and $\lambda = 1$ correspond to the explicit and implicit schemes, respectively. Our goal here is to construct an approximating function from RBFs, which can capture a curved line rather than a straight line over two nodes t_{k-1} and t_k. It is proposed that the first-order derivative of f with respect to t is decomposed into RBFs

$$\frac{\partial f(t)}{\partial t} = w_{k-1}G_{k-1}(t) + w_k G_k(t), \tag{5.17}$$

where, for the MQ case, $G_{k-1}(t) = \sqrt{(t - t_{k-1})^2 + a_{k-1}^2}$ and $G_k(t) = \sqrt{(t - t_k)^2 + a_k^2}$ in which a_{k-1} and a_k are the MQ widths. Expression for computing f is then derived as

$$f(t) = w_{k-1}Q_{k-1}(t) + w_k Q_k(t) + c_1, \tag{5.18}$$

where $Q_{k-1}(t) = \int G_{k-1}(t)\,dt$, $Q_k(t) = \int G_k(t)\,dt$, and c_1 is the constant of integration. It should be emphasised that function f in (5.18) is defined with three coefficients (i.e. w_{k-1}, w_k and c_1) over two nodal points (i.e. t_{k-1} and t_k). This allows one to add an extra equation in the system of converting the RBF space into the physical space. Here we use this extra equation to include the derivative value of f evaluated at the previous time level. Its details are as follows

$$\begin{pmatrix} f^k \\ f^{k-1} \\ \frac{\partial f}{\partial t}^{k-1} \end{pmatrix} = \mathcal{C}_t \begin{pmatrix} w_{k-1} \\ w_k \\ c_1 \end{pmatrix}, \tag{5.19}$$

where $\mathcal{C}_t$ is the conversion matrix defined as

$$\mathcal{C}_t = \begin{pmatrix} Q_{k-1}(t_k) & Q_k(t_k) & 1 \\ Q_{k-1}(t_{k-1}) & Q_k(t_{k-1}) & 1 \\ G_{k-1}(t_{k-1}) & G_k(t_{k-1}) & 0 \end{pmatrix}.$$

Making use of (5.19), the three coefficients can be expressed in terms of the nodal variable values and the derivative value at the previous time level

$$\begin{pmatrix} w_{k-1} \\ w_k \\ c_1 \end{pmatrix} = \mathcal{C}_t^{-1} \begin{pmatrix} f^k \\ f^{k-1} \\ \frac{\partial f}{\partial t}^{k-1} \end{pmatrix}. \tag{5.20}$$

Expression for computing the first-order derivative at the current time level thus becomes

$$\frac{\partial f}{\partial t}^k = \begin{bmatrix} G_{k-1}(t_k) & G_k(t_k) & 0 \end{bmatrix} \begin{pmatrix} w_{k-1} \\ w_k \\ c_1 \end{pmatrix}, \tag{5.21}$$

$$= \begin{bmatrix} G_{k-1}(t_k) & G_k(t_k) & 0 \end{bmatrix} \mathcal{C}_t^{-1} \begin{pmatrix} f^k \\ f^{k-1} \\ \frac{\partial f}{\partial t}^{k-1} \end{pmatrix}, \tag{5.22}$$

which can be rewritten as

$$\frac{\partial f}{\partial t}^k = D_1 f^k + D_2 f^{k-1} + D_3 \dot{f}^{k-1}, \tag{5.23}$$

with D_1, D_2, D_3 being computed from the RBFs and the inverse of $\mathcal{C}_t$ - they are known values. The time derivative term is now expressed in term of values of f at t_{k-1} and t_k (i.e. f^{k-1} and f^k) and its time derivative at t_{k-1} (i.e. $\frac{\partial f}{\partial t}^{k-1}$ or simply $\dot{f}^{k-1}$). An alternative to the discretisation scheme (5.16) is

$$D_1 {f_{ij}}^k + D_2 {f_{ij}}^{k-1} + D_3 \dot{f}_{ij}^{k-1} - \lambda \left(\frac{\partial^2 f_{ij}}{\partial x^2}^k + \frac{\partial^2 f_{ij}}{\partial y^2}^k \right) - (1-\lambda) \left(\frac{\partial^2 f_{ij}}{\partial x^2}^{k-1} + \frac{\partial^2 f_{ij}}{\partial y^2}^{k-1} \right) = b_{ij}^{k-1+\lambda}. \tag{5.24}$$

As shown in Figure 5.2, a function approximated by IRBFs on a time step can be of nonlinear form. It is expected that larger time steps can be used in simulating time-dependent differential problems, where the slope of the solution varies between time levels.

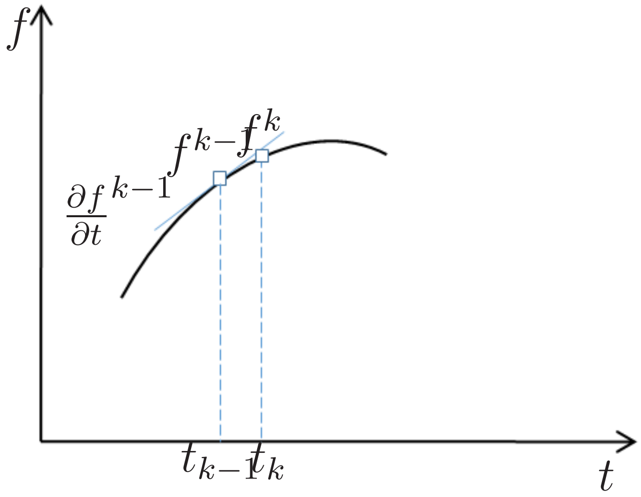


Figure 5.2: Information used to approximate the time derivative term on a time step includes the variable values at t_{k-1} and t_k, and the derivative value at t_{k-1}.

5.3.2 An IRBF-based space-time discretisation scheme

The combination of the proposed compact 2-point stencil for time and the presented compact 5-point stencil for space results in a numerical procedure, which is based on IRBFs only, for solving time-dependent differential problems.

With the explicit scheme (i.e. $\lambda = 0$), the calculation is based on the solution of (5.12) and (5.14) evaluated at the previous time level

$$-\bar{\eta}_1 {\frac{\partial^2 f_1}{\partial x^2}}^{k-1} + {\frac{\partial^2 f_3}{\partial x^2}}^{k-1} - \bar{\eta}_5 {\frac{\partial^2 f_5}{\partial x^2}}^{k-1} = \eta_1 f_1^{k-1} + \eta_3 f_3^{k-1} + \eta_5 f_5^{k-1}, \tag{5.25}$$

$$-\bar{\theta}_2 {\frac{\partial^2 f_2}{\partial y^2}}^{k-1} + {\frac{\partial^2 f_3}{\partial y^2}}^{k-1} - \bar{\theta}_4 {\frac{\partial^2 f_4}{\partial y^2}}^{k-1} = \theta_2 {f_2}^{k-1} + \theta_3 {f_3}^{k-1} + \theta_4 {f_4}^{k-1}. \tag{5.26}$$

These two equations for nodal derivative values lead to systems of tridiagonal algebraic equations on the x and y grid lines that can be solved efficiently by the Thomas algorithm. Note that nodal values of second derivatives on the boundary can be calculated using any 1D approximation scheme on their associated grid lines. In some cases such as rectangular domains, instead of using 1D approximations, one can directly derive these values from the governing equation and the given boundary conditions.

With the implicit schemes (i.e. $0 < \lambda \leq 1$), there are three unknowns at an interior grid node (i.e. values of f and its second derivatives in the x and y directions). A set of three algebraic equations needed for each node consists of the two equations (5.12) and (5.14) evaluated at the current time level, i.e.

$$-\bar{\eta}_1 {\frac{\partial^2 f_1}{\partial x^2}}^{k} + {\frac{\partial^2 f_3}{\partial x^2}}^{k} - \bar{\eta}_5 {\frac{\partial^2 f_5}{\partial x^2}}^{k} = \eta_1 f_1^{k} + \eta_3 f_3^{k} + \eta_5 f_5^{k}, \tag{5.27}$$

$$-\bar{\theta}_2 {\frac{\partial^2 f_2}{\partial y^2}}^{k} + {\frac{\partial^2 f_3}{\partial y^2}}^{k} - \bar{\theta}_4 {\frac{\partial^2 f_4}{\partial y^2}}^{k} = \theta_2 {f_2}^{k} + \theta_3 {f_3}^{k} + \theta_4 {f_4}^{k}. \tag{5.28}$$

and the equation directly derived from the PDE (i.e. equation (5.24)). It is possible to combine these three equations to form two tridiagonal algebraic equations through the implicit elimination approach as discussed in (Mai-Duy and Tran-Cong, 2013).

5.4 Numerical examples

In this study, IRBFs are implemented with the MQ function in the form of

$$G_i(\alpha) = \sqrt{(\alpha - c_i)^2 + a_i^2}, \tag{5.29}$$

where c_i and a_i are the centre and the width of the ith MQ, respectively and α can be x or y in the spatial approximation and t in the temporal approximation. The MQ width is simply chosen according to the relation

$$a_i = \beta_s d_i \quad \text{for space}, \tag{5.30}$$

$$a_i = \beta_t \Delta t \quad \text{for time}, \tag{5.31}$$

where β_s and β_t are positive values, d_i the smallest distance between the centre c_i and its neighbours, and Δt the time step. Different types of time-dependent problems are chosen to study the performance of the proposed numerical procedure. The first three examples are concerned with the heat transfer, convection-diffusion and shallow water equations, for which analytic solutions are available. In the fourth (final) example, the proposed method is applied for the simulation of natural convection flows in the region between a square outer cylinder and a circular inner cylinder. Some standard FD schemes are also employed where appropriate to provide the base for the evaluation of accuracy of the proposed time stencil. Note that a distinguishing feature of the RBF solution is that its accuracy can be controlled not only by the grid size/time step but also by the RBF width. For all numerical examples, the problem domain is simply discretised using a uniform Cartesian grid. The value of d_i in (5.30) thus becomes a grid size. In the case of curved boundaries, a distance to the boundary used for the removing of interior nodes is chosen as $d_i/8$. When the analytic solution is available, the numerical error is measured in the form of:

1. Discrete relative L_2 norm

$$Ne = \frac{\sqrt{\sum_{i=1}^{m}(f_i^e - f_i)^2}}{\sqrt{\sum_{i=1}^{m}(f_i^e)^2}}, \tag{5.32}$$

2. Root-mean-square error (RMSE)

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}(f_i^e - f_i)^2}, \tag{5.33}$$

3. Maximum of absolute error (MAE)

$$MAE = \|f_i^e - f_i\|_{\max}, \tag{5.34}$$

where m is the number of nodal points, n_t the number of time steps, and f^e and f respectively denote the exact and approximate solutions. In the last example, the flow is considered to reach the steady state when the following condition is satisfied

$$CM = \frac{\sqrt{\sum_{i=1}^{n_{ip}}\left(f_i^k - f_i^{k-1}\right)^2}}{\sqrt{\sum_{i=1}^{n_{ip}}\left(f_i^k\right)^2}} < \epsilon, \tag{5.35}$$

where n_{ip} is the number of interior points, k the time level, f the stream function and ϵ the tolerance. In this study, ϵ is taken to be 10^{-12}.

5.4.1 Example 1: Parabolic PDEs

One dimensional space

The proposed method is first verified in the following PDE

$$\frac{\partial f}{\partial t}(x,t) = \frac{\partial^2 f}{\partial^2 x}(x,t) + b(x,t), \quad 0 \le x \le 1, \tag{5.36}$$

with $b\,(x,t) = 50xe^{50t}$. Its exact solution is given by

$$f^e(x,y,t) = xe^{50t}, \tag{5.37}$$

from which one can derive the initial values and Dirichlet boundary conditions. As shown in Figure 5.3, function f grows very quickly with time. To assess accuracy of the time discretisation only, we approximate the time derivative term in (5.36) explicitly using the forward differences and the proposed compact time stencils, and employ the same spatial approximation for the two schemes. The second derivative $\partial^2 f/\partial^2 x$ is approximated by compact IRBF stencils on a set of 10 nodes with $\beta_s = 3.5$. Figure 5.4 displays the solution error by the two schemes at $\Delta t = 10^{-3}$. It can be seen that the IRBF solution is much more accurate than the FD one. To achieve the same accuracy level of the IRBF time scheme, as shown in Figure 5.5, the FD time scheme needs a much smaller time step (i.e. $\Delta t = 10^{-6}$). The obtained results of this example demonstrate that

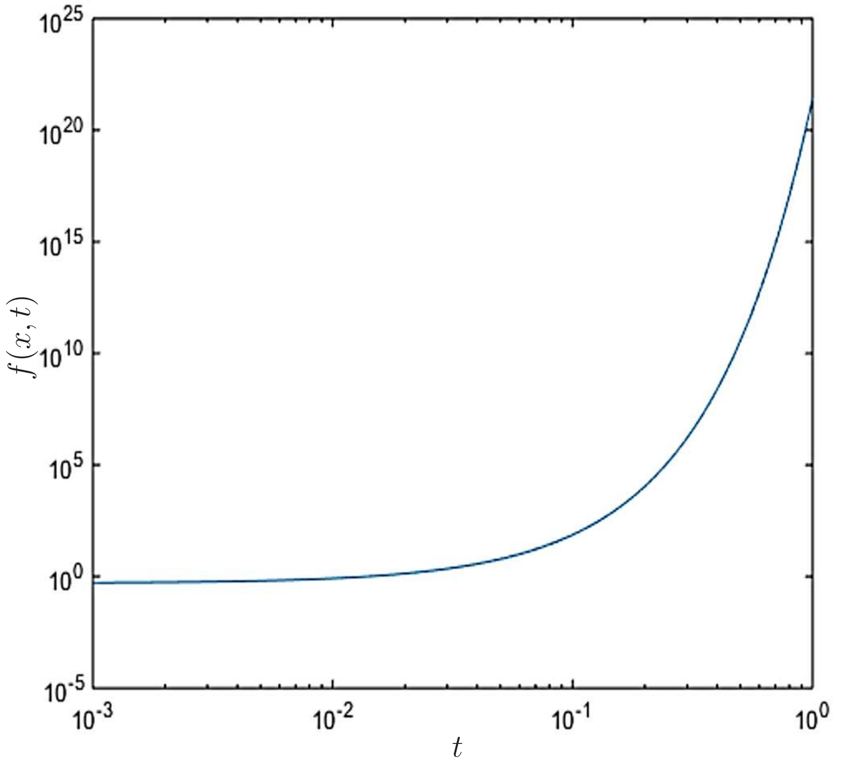


Figure 5.3: Example 1.1, parabolic PDE: Variation of $f(x,t)$ with time at $x = 0.5$.

the proposed compact time stencil has the ability to work with relatively large time steps for a given accuracy.

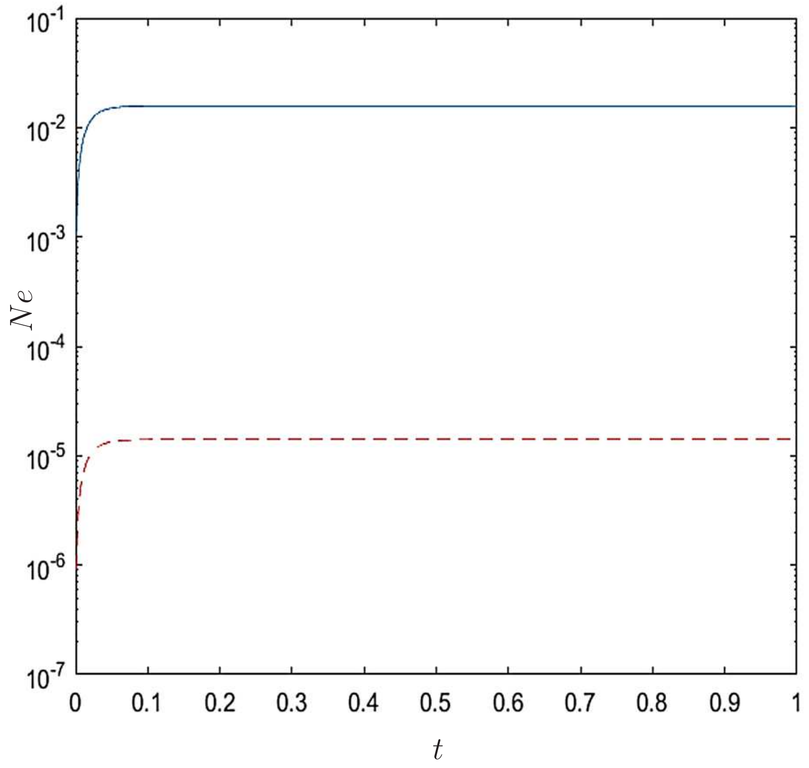


Figure 5.4: Example 1.1, parabolic PDE, spatial compact IRBF stencils, $\Delta t = 10^{-3}$: Comparison of the solution accuracy between the FD ('$-$') and IRBF ('--', $\beta_t = 18$) time discretisations.

Two dimensional space

The PDE to be considered here is in the form of

$$\frac{\partial f}{\partial t}(x,y,t) - \left(\frac{\partial^2 f}{\partial x^2}(x,y,t) + \frac{\partial^2 f}{\partial y^2}(x,y,t)\right) = 3e^t \sin(x) \sin(y), \tag{5.38}$$

The exact solution is given by

$$f^e(x,y,t) = \sin(x)\sin(y)e^t. \tag{5.39}$$

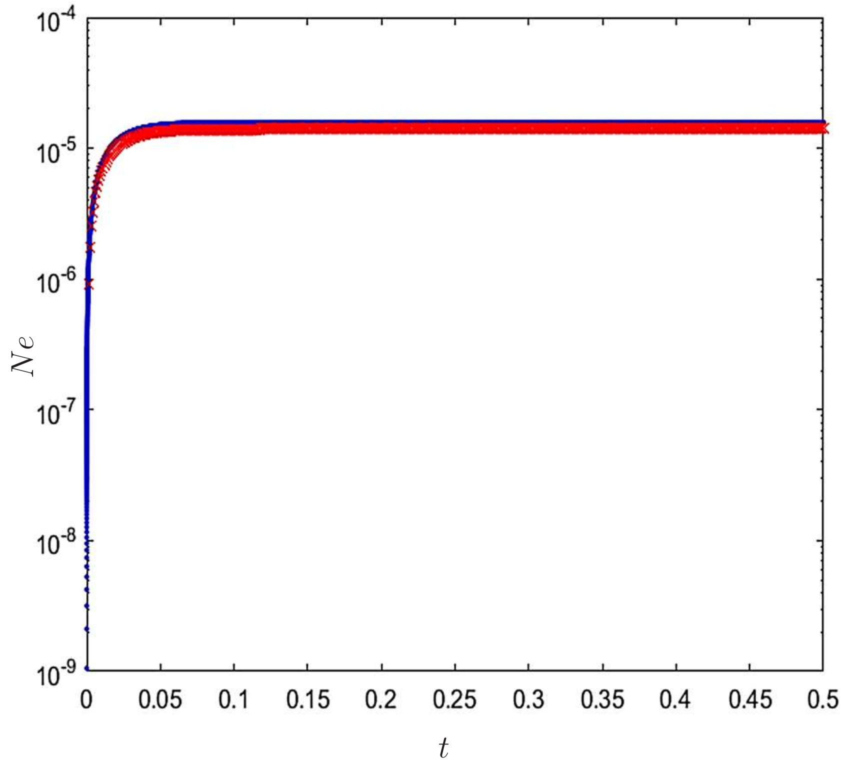


Figure 5.5: Example 1.1, parabolic PDE, spatial compact IRBF stencils: Comparison of the solution accuracy between the FD ('$\cdot$', $\Delta t = 10^{-6}$) and the IRBF ('$\times$', $\Delta t = 10^{-3}$, $\beta_t = 18$) time discretisations.

This function grows exponentially with time and thus provides a good test for the proposed compact time stencil. The initial values and Dirichlet boundary conditions can be derived from (5.39).

We consider two types of domains, a unit square and a multiply-connected domain that is a region lying between a unit square and a circle of radius 0.2. The explicit approach is employed to obtain the numerical solutions of (5.38).

For the unit square, to examine the accuracy of the proposed compact time stencils, we also implement the forward differences. These two time approximation schemes are employed with the same time step of 10^{-3} and the same spatial approximation that is based on central differences on a grid density of 10×10. Figure 5.6 shows that a much improved accuracy is obtained with the proposed scheme ($\beta_t = 12$). It is noted that the accuracy is computed over the whole spatial domain. As also shown in the figure, a further improvement can be achieved by replacing the spatial central differences with the compact 5-node IRBF stencils using $\beta_s = 3.5$. For the multi-connected domain, because of its non-rectangular shape, we only employ the compact 5-node IRBF stencils for the spatial approximation. The obtained results on a grid density of 22×22 and with $\Delta t = 10^{-4}$ are displayed in Figure 5.7. Again, it can be seen that results by the proposed compact time stencil are more accurate than those by the forward differences.

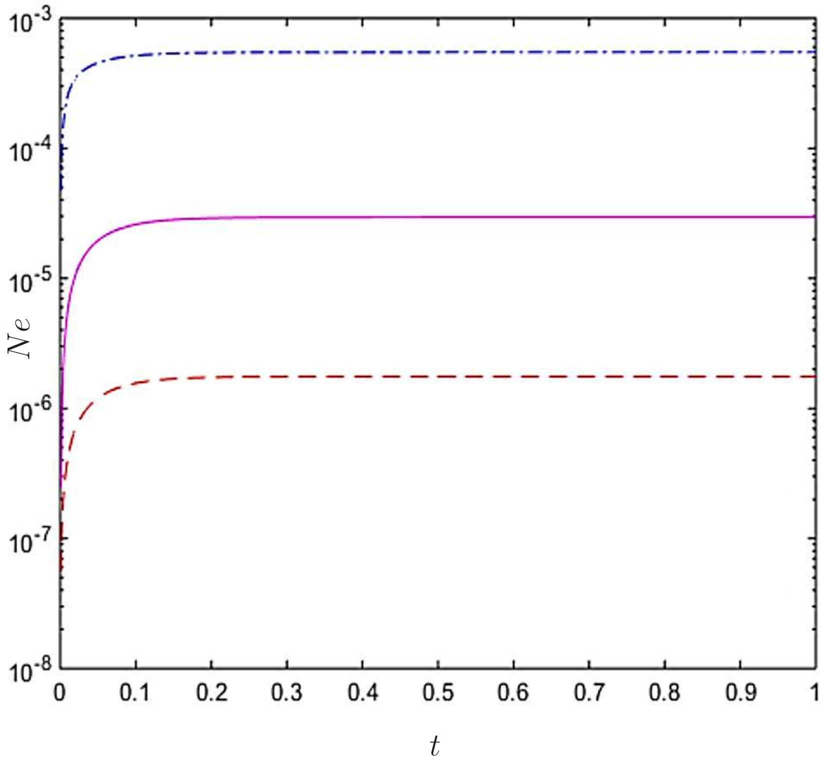


Figure 5.6: Example 1.2, parabolic PDE, rectangular domain, $\Delta t = 10^{-3}$: Numerical errors obtained by the FD time-FD space ('·—'), IRBF time-FD space ('—') and IRBF time-IRBF space ('- -') discretisations.

5.4.2 Example 2: Convection-Diffusion equations

The proposed method is further verified with the convection-diffusion equations in one and two dimensional space.

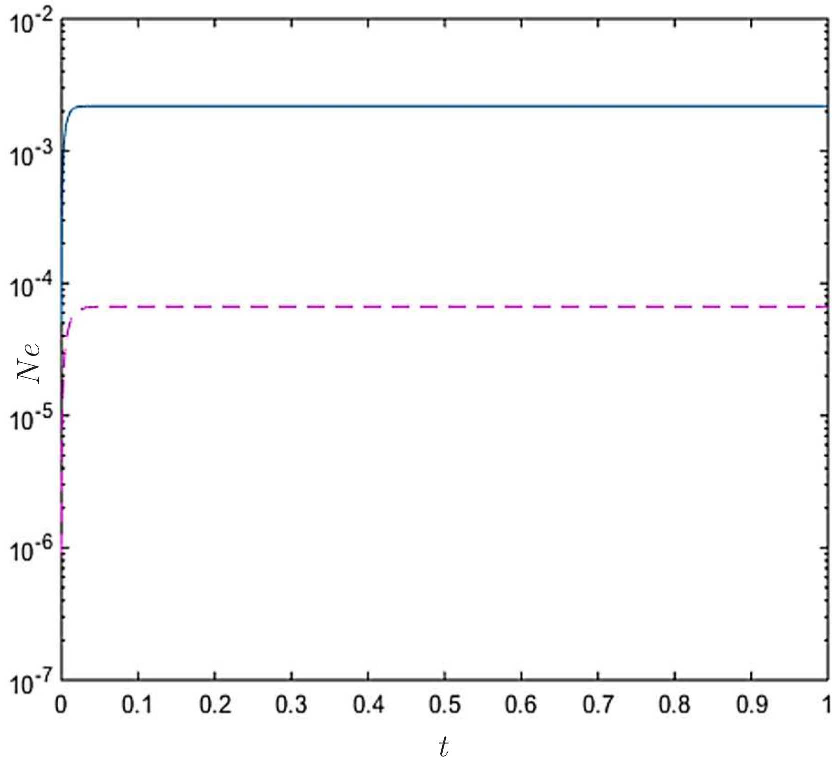


Figure 5.7: Example 1.2, parabolic PDE, non-rectangular domain, spatial compact IRBF stencils, $\Delta t = 10^{-4}$ and $\beta_t = 10$: Numerical errors obtained by the FD time ('—') and IRBF time ('- -') discretisations.

One dimensional space

Consider the following equation

$$\frac{\partial f}{\partial t}(x,t)+f(x,t)\frac{\partial f}{\partial x}(x,t) = \frac{\partial^2 f}{\partial x^2}(x,t)+2\sin(x)e^t+\sin(x)\cos(x)e^{2t}, \quad (5.40)$$

on an interval $[0,1]$ with the initial and boundary conditions

$$f(x,0) = \sin(x), \quad (5.41)$$

$$f(0,t) = 0, \quad (5.42)$$

$$f(1,t) = \sin(1)e^t. \quad (5.43)$$

The exact solution to this problem can be verified to be $f^e(x,t) = \sin(x)e^t$.

We employ compact 3-point IRBF stencils on a grid of 10 nodes with $\beta_s = 3.5$ for the spatial approximation, and compact 2-point IRBF stencils for the temporal approximation. Attention here is given to the effects of the RBF width in the time domain on the solution accuracy. The obtained results at a time step of 10^{-3} are shown in Figure 5.8. Results by the forward differences are also included for comparison purposes. It can be seen that better accuracy can be achieved by changing the RBF width. The effect of increasing β_t here is similar to the effect of reducing Δt however, changing β_t does not lead to any increase in computational cost.

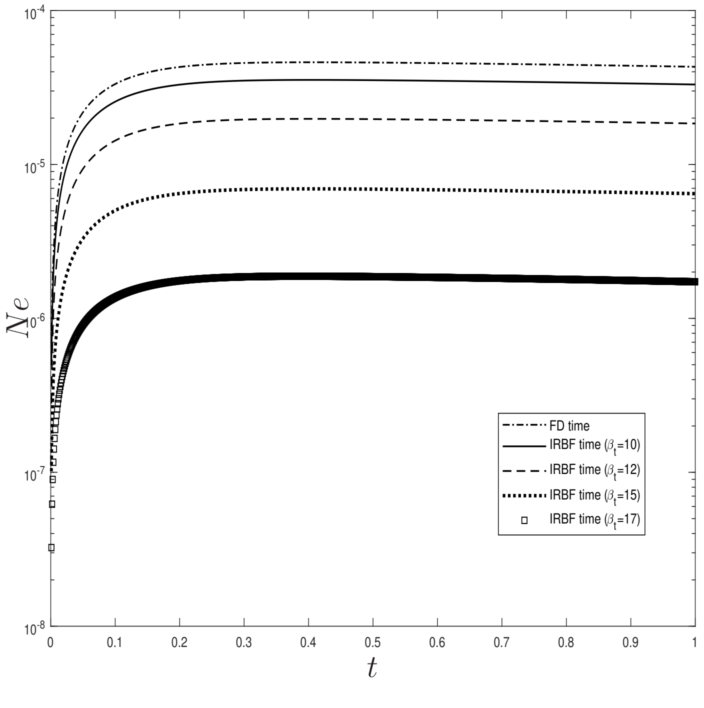


Figure 5.8: Example 2.1, 1D convection-diffusion equation, $\Delta t = 10^{-3}$: Effect of the temporal RBF width, represented through β_t (β_t= 10, 12, 15, 17), on the IRBF solution accuracy. Results by the conventional FD method are also included.

Two dimensional space

An unsteady convection-diffusion equation in two dimensional space for a variable f can be expressed as

$$\frac{\partial f}{\partial t}(x,y,t)+c_x\frac{\partial f}{\partial x}(x,y,t)+c_y\frac{\partial f}{\partial y}(x,y,t)=d_x\frac{\partial^2 f}{\partial x^2}(x,y,t)+d_y\frac{\partial^2 f}{\partial y^2}(x,y,t)+b(x,y,t)\,. \tag{5.44}$$

Here, we choose $c_x = c_y = 0.01$, $d_x = d_y = 1$ and

$$b(x,y,t) = 3\sin(x)\sin(y)r + 0.01r(\cos(x)\sin(y)+\cos(y)\sin(x))\,.$$

The domain of interest is of $[0,1]\times[0,1]$ and the initial and boundary conditions are given by

$$u(x,y,0) = \sin(x)\sin(y)\,, \tag{5.45}$$

$$u(0,y,t) = u(x,0,t) = 0, \tag{5.46}$$

$$u(1,y,t) = \sin(1)\sin(y)r, \tag{5.47}$$

$$u(x,1,t) = \sin(x)\sin(1)r, \tag{5.48}$$

where

$$r = 1 + t + \frac{t^2}{2} + \frac{t^3}{6} + \frac{t^4}{24} + \frac{t^5}{120}.$$

This problem has the following exact solution

$$f^e(x,t) = \sin(x)\sin(y)r. \tag{5.49}$$

The problem domain is represented by a Cartesian grid of 10 × 10. Other parameters employed are $\beta_s = 3.5$ and $\Delta t = 10^{-3}$. As shown in Figure 5.9, with the same spatial approximation employed, the proposed compact time scheme outperforms the conventional forward differences. Similar remarks to the case of one dimensional space can also be made here. In particular, the solution accuracy can be enhanced by changing the MQ width (β_t) without any additional computational cost.

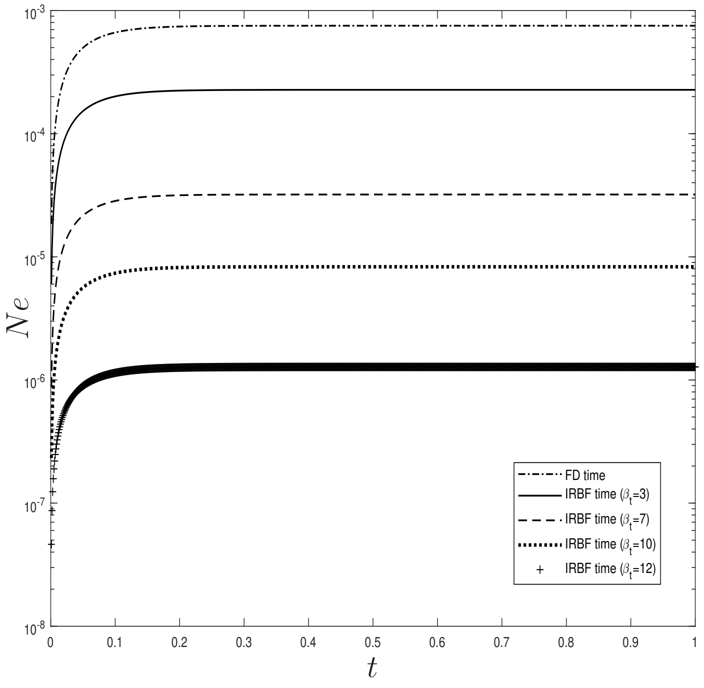


Figure 5.9: Example 2.2, 2D convection-diffusion equation, $\Delta t = 10^{-3}$: Effect of the temporal RBF width, represented through β_t (β_t= 3, 7, 10, 12), on the IRBF solution accuracy. Results by the conventional FD method are also included.

5.4.3 Example 3: Shallow water equations (SWEs)

In the case of problems concerning a thin layer (compared to its length scale) of fluid of constant density in hydrostatic balance, bounded from below by the bottom topography such as a sea floor and from above by a free surface, the shallow water equations (SWEs) are utilised. SWEs are a system of hyperbolic partial differential equations (PDEs) governing the flow of water in an area in which the horizontal dimension significantly exceeds the depth such as coastal regions, estuaries, rivers and channels, and horizontal velocity that dominates the flow field. The vertical momentum exchange is negligible and the vertical velocity component w is a lot smaller than the horizontal components u and v. These equations arise from the basic equations of fluid mechanics for an inviscid and incompressible fluid. For an incompressible fluid, the change in the density of a fluid is zero, which means that

$$\frac{d\rho}{dt} = 0. \tag{5.50}$$

The continuity equation becomes

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{5.51}$$

where velocity field $\vec{u}(\vec{x}, t) = (u(x, y, z, t), v(x, y, z, t), w(x, y, z, t))$.
Integrating this equation over the vertical extent of the fluid yields

$$\int_{z_b}^{z_b+h} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) dz = 0, \tag{5.52}$$

where z_b is the height of the bed surface.
Leibniz' Theorem

$$\frac{\partial}{\partial t}\int_{a(y,t)}^{b(y,t)} f(x, y, t)dx = \int_{a(y,t)}^{b(y,t)} \frac{\partial f}{\partial t}dx - f(a, y, t)\frac{\partial a}{\partial t} + f(b, y, t)\frac{\partial b}{\partial t}. \tag{5.53}$$

Applying this theorem, we have

$$\int_{z_b}^{z_b+h} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) dz = 0$$

$$\Rightarrow \int_{z_b}^{z_b+h} \frac{\partial u}{\partial x}dz + \int_{z_0}^{z_0+h} \frac{\partial v}{\partial y}dz + \int_{z_b}^{z_b+h} \frac{\partial w}{\partial z}dz = 0$$

$$\Rightarrow \frac{\partial}{\partial x}\int_{z_b}^{z_b+h} udz - u|_{z_b+h}\frac{\partial(z_b+h)}{\partial x} + u|_{z_b}\frac{\partial z_b}{\partial x}$$

$$+\frac{\partial}{\partial y}\int_{z_b}^{z_b+h} vdz - v|_{z_b+h}\frac{\partial(z_b+h)}{\partial y} + v|_{z_b}\frac{\partial z_b}{\partial y} + w|_{z_b}^{z_b+h} = 0$$

$$\Rightarrow \frac{\partial}{\partial x}\int_{z_b}^{z_b+h} udz + \frac{\partial}{\partial y}\int_{z_b}^{z_b+h} vdz - u|_{z_b+h}\frac{\partial(z_b+h)}{\partial x} - v|_{z_b+h}\frac{\partial(z_b+h)}{\partial y} + w|_{z_b+h}$$

$$+u|_{z_b}\frac{\partial z_b}{\partial x}+v|_{z_b}\frac{\partial z_b}{\partial y}-w|_{z_b}=0$$

$$\Rightarrow \frac{\partial}{\partial x}\int_{z_b}^{z_b+h} udz+\frac{\partial}{\partial y}\int_{z_b}^{z_b+h} vdz+\frac{\partial h}{\partial t}=0.$$

Defining the average velocity over the depth of vertical extent in x and y, we have

$$\int_{z_b}^{z_b+h} udz=\bar{u}h, \tag{5.54}$$

$$\int_{z_b}^{z_b+h} vdz=\bar{v}h. \tag{5.55}$$

The first of three two-dimensional depth-averaged shallow water wave equations is

$$\frac{\partial(\bar{u}h)}{\partial x}+\frac{\partial(\bar{v}h)}{\partial x}+\frac{\partial h}{\partial t}=0. \tag{5.56}$$

The momentum equation of motion can be used to obtain the remaining two dimensional shallow water equations. The momentum equations for a Newtonian fluid with constant density are

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial P}{\partial x}, \tag{5.57}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-\frac{1}{\rho}\frac{\partial P}{\partial y}, \tag{5.58}$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho}\frac{\partial P}{\partial z}-g. \tag{5.59}$$

Integrating the momentum equation in the x-direction over the vertical extent of the fluid yields

$$\int_{z_b}^{z_b+h}\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}+\frac{1}{\rho}\frac{\partial P}{\partial x}\right)dz$$

$$=\int_{z_b}^{z_b+h}\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)+\int_{z_b}^{z_b+h}\frac{1}{\rho}\frac{\partial P}{\partial x}dz$$

$$= A + B,$$

with $A = \int_{z_b}^{z_b+h}\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right)$ and $B = \int_{z_b}^{z_b+h}\frac{1}{\rho}\frac{\partial P}{\partial x}dz$

$$A = \int_{z_b}^{z_b+h}\frac{\partial u}{\partial t}dz + \int_{z_b}^{z_b+h}u\frac{\partial u}{\partial x}dz + \int_{z_b}^{z_b+h}v\frac{\partial u}{\partial y}dz + \int_{z_b}^{z_b+h}w\frac{\partial u}{\partial z}dz.$$

We use Leibniz theorem to time derivative term to obtain

$$\int_{zb}^{z_b+h}\frac{\partial u}{\partial t}dz = \frac{\partial}{\partial t}\int_{z_b}^{z_b+h}udz - u|_{z_b+h}\frac{\partial(z_b+h)}{\partial t} + u|_{z_b}\frac{\partial z_b}{\partial t}$$

$$\int_{z_b}^{z_b+h}u\frac{\partial u}{\partial x}dz = \frac{\partial}{\partial x}\int_{z_b}^{z_b+h}u^2dz - u^2\big|_{z_b+h}\frac{\partial(z_b+h)}{\partial x} + u^2\big|_{z_b}\frac{\partial z_b}{\partial x}$$

$$\int_{z_b}^{z_b+h}v\frac{\partial u}{\partial y}dz = \frac{\partial}{\partial y}\int_{z_b}^{z_b+h}uvdz - (uv)|_{z_b+h}\frac{\partial(z_b+h)}{\partial y} + (uv)|_{z_b}\frac{\partial z_0}{\partial y}$$

$$\Rightarrow A = \frac{\partial}{\partial t}\int_{z_b}^{z_b+h}udz + \frac{\partial}{\partial x}\int_{z_b}^{z_b+h}u^2dz + \frac{\partial}{\partial y}\int_{z_b}^{z_b+h}uvdz$$

$$-u|_{z_b+h}\frac{\partial(z_b+h)}{\partial t} - u^2\big|_{z_b+h}\frac{\partial(z_b+h)}{\partial x} - (uv)|_{z_b+h}\frac{\partial(z_b+h)}{\partial y} + (uw)|_{z_b+h}$$

$$+u|_{z_b}\frac{\partial z_b}{\partial t} + u^2\big|_{z_b}\frac{\partial z_b}{\partial x} + (uv)|_{z_b}\frac{\partial z_0}{\partial y} - (uw)|_{z_b}$$

$$= \frac{\partial}{\partial t}\int_{z_b}^{z_b+h}udz + \frac{\partial}{\partial x}\int_{z_b}^{z_b+h}u^2dz + \frac{\partial}{\partial y}\int_{z_b}^{z_b+h}uvdz$$

$$-u|_{z_b+h}\left(\frac{\partial h}{\partial t} + u|_{z_b+h}\frac{\partial(z_b+h)}{\partial x} + v|_{z_b+h}\frac{\partial(z_b+h)}{\partial y} - w|_{z_b+h}\right)$$

$$+u|_{z_b}\left(\frac{\partial z_b}{\partial t} + u|_{z_b}\frac{\partial z_0}{\partial x} + v|_{z_b}\frac{\partial z_0}{\partial y} - w|_{z_b}\right)$$

$$= \frac{\partial(uh)}{\partial t} + \frac{\partial(u^2h)}{\partial x} + \frac{\partial(uvh)}{\partial y}.$$

Any vertical elevation within the fluid pressure is given by

$$P = \rho g\left(z_b + h - z\right).$$

Integrated the term B to obtain

$$\int_{z_b}^{z_b+h} \frac{\partial P}{\partial x} dz$$

$$= \rho g \int_{z_b}^{z_b+h} \frac{\partial}{\partial x} (z_b + h - z)\, dz$$

$$= \rho g \left(\frac{\partial}{\partial x} \int_{z_b}^{z_b+h} (z_b + h - z)\, dz + (z_b + h - z_b) \frac{\partial z_b}{\partial x} - (z_b + h - (z_b + h)) \frac{\partial (z_b + h)}{\partial x} \right)$$

$$= \rho g \left(\frac{\partial}{\partial x} \int_{z_b}^{z_b+h} (z_b + h - z)\, dz + h \frac{\partial z_b}{\partial x} \right)$$

$$= \rho g \left(\frac{1}{2} \frac{\partial h^2}{\partial x} + h \frac{\partial z_b}{\partial x} \right),$$

$$\Rightarrow \frac{1}{\rho} \int_{z_b}^{z_b+h} \frac{\partial P}{\partial x} dz = \frac{1}{2} g \frac{\partial h^2}{\partial x} + gh \frac{\partial z_b}{\partial x}, \tag{5.60}$$

$$\Rightarrow \frac{\partial (uh)}{\partial t} + \frac{\partial (u^2 h)}{\partial x} + \frac{\partial (uvh)}{\partial y} + \frac{1}{2} g \frac{\partial h^2}{\partial x} + gh \frac{\partial z_b}{\partial x} = 0, \tag{5.61}$$

$$\Rightarrow \frac{\partial (uh)}{\partial t} + \frac{\partial (u^2 h)}{\partial x} + \frac{\partial (uvh)}{\partial y} + \frac{1}{2} g \frac{\partial h^2}{\partial x} = -gh \frac{\partial z_b}{\partial x}, \tag{5.62}$$

and

$$\frac{\partial (uh)}{\partial t} + \frac{\partial (uvh)}{\partial x} + \frac{\partial (v^2 h)}{\partial y} + \frac{1}{2} g \frac{\partial h^2}{\partial y} = -gh \frac{\partial z_b}{\partial y}. \tag{5.63}$$

These two equations in combination, constitute one form of the depth-averaged shallow water equation.

We will consider the general form of shallow water equations

$$\frac{\partial (uh)}{\partial x} + \frac{\partial (vh)}{\partial y} + \frac{\partial h}{\partial t} = 0, \tag{5.64}$$

$$\frac{\partial (uh)}{\partial t} + \frac{\partial (u^2 h)}{\partial x} + \frac{\partial (uvh)}{\partial y} + \frac{1}{2} g \frac{\partial h^2}{\partial x} = -gh \frac{\partial z_b}{\partial x}, \tag{5.65}$$

$$\frac{\partial (uh)}{\partial t} + \frac{\partial (uvh)}{\partial x} + \frac{\partial (v^2 h)}{\partial y} + \frac{1}{2} g \frac{\partial h^2}{\partial y} = -gh \frac{\partial z_b}{\partial y}. \tag{5.66}$$

Here, this is called shallow water bodies. The vertical momentum exchange is negligible and the vertical velocity component w is a lot smaller than the horizontal components u and v. These equations arise from the basic equations of fluid mechanics for an inviscid and incompressible fluid. The independent variables are time t, and space coordinates, x and y. The dependent variables are the fluid height or depth h, and two-dimensional fluid velocity field u and v. The force acting on the fluid is gravity, represented by the gravitational constant g. In order to write the equations in a compact form, one can introduce three vectors:

$$U = \begin{pmatrix} h \\ uh \\ vh \end{pmatrix}, \ F(U) = \begin{pmatrix} uh \\ u^2h + \frac{1}{2}gh^2 \\ uvh \end{pmatrix}, \ K(U) = \begin{pmatrix} vh \\ uvh \\ v^2h + \frac{1}{2}gh^2 \end{pmatrix}, \ S = \begin{pmatrix} 0 \\ -gh\frac{\partial z_b}{\partial x} \\ -gh\frac{\partial z_b}{\partial y} \end{pmatrix}.$$

The shallow water equations (5.64)-(5.66) reduce to an instance of a hyperbolic conservation law which can be seen as

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial K(U)}{\partial x} = S. \tag{5.67}$$

The continuity and momentum SWEs can be linearised as follows

$$\frac{\partial h}{\partial t}(x, y, t) + H\left(\frac{\partial u}{\partial x}(x, y, t) + \frac{\partial v}{\partial y}(x, y, t)\right) = 0, \tag{5.68}$$

$$\frac{\partial u}{\partial t}(x, y, t) + g\frac{\partial h}{\partial x}(x, y, t) = 0, \tag{5.69}$$

$$\frac{\partial v}{\partial t}(x, y, t) + g\frac{\partial h}{\partial y}(x, y, t) = 0, \tag{5.70}$$

where $g = 9.81\ m/s^2$. For convenience, the water depth h can be regarded as the sum of the mean water depth H and the water surface elevation ζ.

RBF methods have been applied to solve the shallow water equations (SWEs). Their solutions are reported using the global MQ approximation (Hon et al., 1999; Young et al., 2004, 2005), compactly supported RBF (CSRBF) (Wong et al., 2002) and local RBF differential quadrature (LRBFDQ) (Sun et al., 2013) methods. In these works, the time derivative term is approximated by conventional finite-difference schemes. For SWEs, there are two dependent variables, namely the water height in the z direction, denoted by h, and the velocity vector in the $x-y$ plane, denoted by (u, v). They are functions of space $\mathbf{x}$ and time t.

Consider a rectangular channel of length $L = 872\ km$ and width $W = 50\ km$ with the fluid being water as shown in Figure 5.10. The mean water depth is $H = 20\ m$.

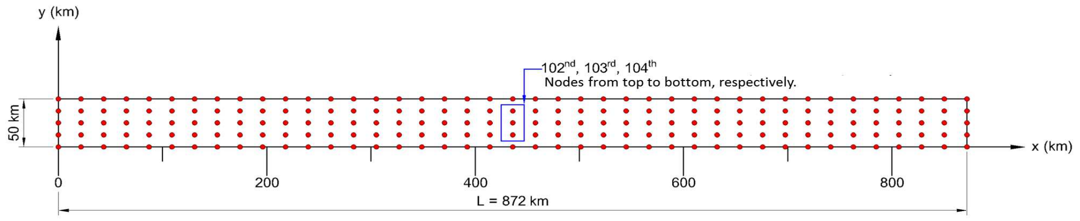


Figure 5.10: Example 3, shallow water flows: A rectangular channel and its Cartesian grid of 41 × 5. Numerical results obtained are compared at nodes 102, 103 and 104.

The boundary condition for the water surface elevation is specified as

$$\zeta(x, y, t) = \zeta_0 \cos at,$$

at $x = 0$, $0 \leq y \leq W$, $\zeta_0 = 1\ m$ and $a = 1.45444 \times 10^{-4} s^{-1}$, while the land boundary conditions are

$$u(x, y, t) = 0,$$

at $x = L$, $0 \leq y \leq W$ and

$$u(x, y, t) = 0,$$

at $y = 0$ and $y = W$, $0 \leq x \leq L$. The initial conditions are prescribed as

$$u(x, y, t = 0) = 0, \tag{5.71}$$

$$v(x, y, t = 0) = 0, \tag{5.72}$$

$$\zeta(x, y, t = 0) = \zeta_0 \cos\left(\frac{a}{\sqrt{gH}}(L - x)\right) / \cos\left(\frac{a}{\sqrt{gH}}L\right). \tag{5.73}$$

This fluid flow problem has the following exact solution

$$\zeta(x, y, t) = \zeta_0 \cos\left(\frac{a}{\sqrt{gH}}(L - x)\right) \cos at / \cos\left(\frac{a}{\sqrt{gH}}L\right), \tag{5.74}$$

$$u(x, y, t) = -\zeta_0 \sqrt{\frac{g}{H}} \sin\left(\frac{a}{\sqrt{gH}}(L - x)\right) \sin at / \cos\left(\frac{a}{\sqrt{gH}}L\right), \tag{5.75}$$

$$v(x, y, t) = 0. \tag{5.76}$$

As in (Sun et al., 2013), for comparison purposes, we also discretise the fluid domain using a set of 205 collocation points and employ a time step of 30 s. The results obtained from proposed method are shown in Table 5.1 together with those obtained with the global-MQ method (Young et al., 2005), CSRBF method (Wong et al., 2002) and LRBFDQ method employed with 9 (R_9) and 13 (R_{13}) local nodes per approximation (Sun et al., 2013). The temporal term is discretised by the Taylor method with second-order accuracy for the global-MQ and CSRBF methods, and full-implicit FD scheme for LRBFDQ. All the numerical results displayed in Table 5.1 are computed at $t = 43200$ s and at three particular points 102, 103, and 104 which are located at the centre of the basin (Figure 5.10). The units of water depth and velocity used are cm and cm/s, respectively. Errors for the water height and velocities are also measured by means of $RMSE$ and MAE defined in (5.33) and (5.34), respectively. It can be seen that the proposed method yields the most accurate results. Figure 5.11

shows the water free surfaces at two time levels ($t = 14400$ s and $t = 43200$ s) and the IRBF results look feasible when compared to the analytic solutions.

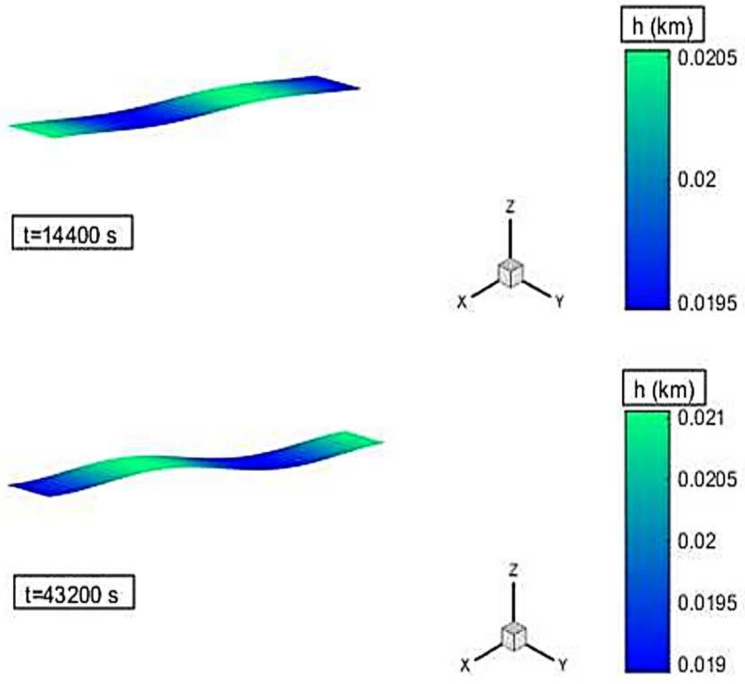


Figure 5.11: Example 3, shallow water flows: Water surfaces at $t = 14440$ s and $t = 43200$ s by the proposed method.

5.4.4 Example 4: Buoyancy-driven flows

In this example, natural convection between a heated inner circular cylinder of diameter D_i and a cooled square enclosure of side length D is considered (Figure 5.12). This problem has been investigated with both experimental and numerical works. The latter was conducted with a variety of numerical techniques such as the finite-difference methods (De Vahl Davis, 1983; Kuehn and Goldstein, 1976), finite-element methods (Manzari, 1999; Sammouda et al., 1999; Jin and Shen, 2016), finite-volume methods (Glakpe et al., 1986; Moukalled and Acharya, 1996), RBF-based methods (Šarler et al., 2004), lattice Boltz-

Table 5.1: Example 3, shallow water flows: Comparison of numerical errors at three nodes 102, 103, and 104 between the proposed method and the LRBFDQ, CSRBF and global MQ methods

Numerical error	Water depth (h) (cm)			Velocity (u) (cm/s)		
	102	103	104	102	103	104
Proposed method						
$RMSE$	0.007	0.007	0.0072	0.0080	0.0086	0.0080
MAE	0.030	0.031	0.031	0.029	0.027	0.029
LRBFDQ						
R_{13}						
$RMSE$	0.0080	0.0036	0.0076	0.016	0.020	0.016
MAE	0.30	0.13	0.29	0.61	0.74	0.60
R_9						
$RMSE$	0.0076	0.0044	0.0076	0.059	0.054	0.059
MAE	0.29	0.17	0.29	2.24	2.03	2.24
CSRBF						
$RMSE$	0.70	0.32	0.35	0.46	0.38	0.49
MAE	1.48	0.18	0.67	0.81	0.92	0.91
Global – MQ						
$RMSE$	0.49	0.71	1.01	0.63	1.0	1.48
MAE	1.19	1.51	1.76	1.06	2.33	2.74

mann methods (Wang et al., 2016; Ahrar and Djavareshkian, 2017) and spectral methods (Le Quere, 1991; Shu, 1999; Wang et al., 2015b).

The governing equations can be written in terms of the stream function (ψ), vorticity (ω) and temperature (T)

$$\nabla^2\psi = \omega, \tag{5.77}$$

$$\frac{\partial\omega}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\omega = \sqrt{\frac{Pr}{Ra}}\nabla^2\omega - \frac{\partial T}{\partial x}, \tag{5.78}$$

$$\frac{\partial T}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})T = \frac{1}{\sqrt{RaPr}}\nabla^2 T, \tag{5.79}$$

where $\mathbf{u}$ is the velocity vector ($u = \partial\psi/\partial y$ and $v = -\partial\psi/\partial x$), and Pr and Ra the Prandtl and Rayleigh numbers defined as $Pr = \nu/\gamma$ and $Ra = \beta g\Delta T D^3/\gamma\nu$, in which ν is the kinematic viscosity, γ the thermal diffusivity, β the thermal

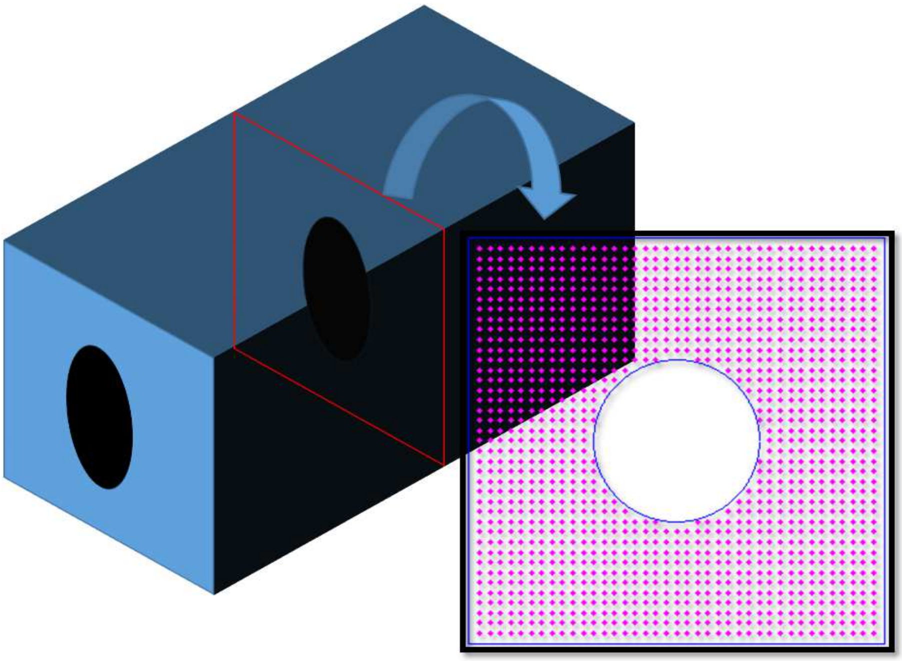


Figure 5.12: Example 4, natural convection: A domain of analysis and its Cartesian grid.

expansion coefficient and g the gravity.

We employ an aspect ratio of $D/D_i = 2.5$, $Pr = 0.71$ and $Ra = \{10^4, 5 \times 10^4, 10^5, 5 \times 10^5, 10^6\}$. Non-slip boundary conditions and the symmetry of flow about the vertical centreline lead to $\psi = 0$ and $\partial\psi/\partial n = 0$ (n - the normal direction) on the inner and outer boundaries. Following (Le-Cao et al., 2009), we derive boundary conditions for equation (5.78). The values of the vorticity at the boundary nodes on the x and y grid lines can be computed by

$$\omega_b = [1 + (\frac{y}{x})^2]\frac{\partial^2\psi_b}{\partial x^2}, \tag{5.80}$$

$$\omega_b = [1 + (\frac{x}{y})^2]\frac{\partial^2\psi_b}{\partial y^2}, \tag{5.81}$$

respectively. The boundary conditions for (5.79) are $T = 1$ and $T = 0$ on the inner and outer surfaces, respectively.

The fluid domain is discretised using a grid density of 30×30. The three equations (5.77)-(5.79) must be solved simultaneously; an iterative scheme, where the convection terms are treated explicitly, is employed to obtain a convergent solution with time. When the difference between two successive stream function fields can be negligible, the flow is considered to reach the steady state. Numerical experiments indicate that the proposed compact time stencil can work with larger time step than the conventional FD scheme, leading to a faster convergence as shown in Figure 5.13. The obtained velocity vector field and contour plots of the temperature are displayed in Figure 5.14, where 21 contour lines are used with their levels varying linearly between the minimum and maximum values. They look feasible when compared to existing results by other methods.

One important result of this type of flow is the local heat transfer coefficient

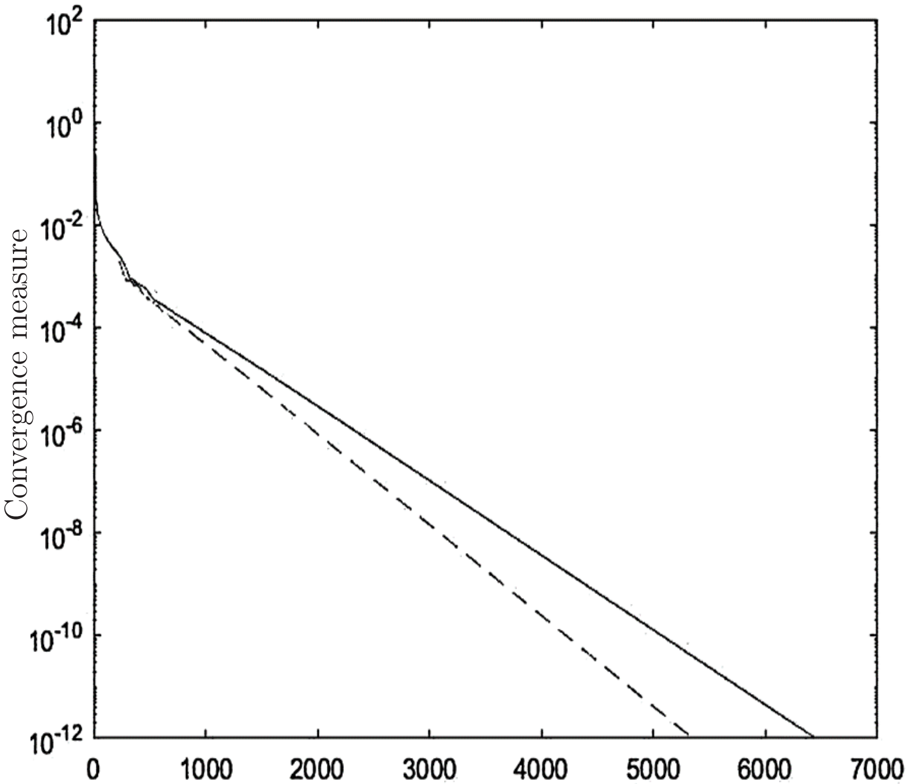


Figure 5.13: Example 4, natural convection, spatial compact IRBF stencils, $\Delta t = 0.02$ (IRBF) and $\Delta t = 0.014$ (FD), $Ra = 10^5$: The IRBF approximation with respect to time can work with a larger time step and its convergence ('——') is seen to be faster than that of the conventional FD one ('-')

defined as (Moukalled and Acharya, 1996)

$$\Theta = -l\frac{\partial T}{\partial n}, \tag{5.82}$$

where l is the thermal conductivity. The average Nusselt number (the ratio of the temperature gradient at the wall to a reference temperature gradient) is computed by

$$Nu = \frac{\overline{\Theta}}{l}, \tag{5.83}$$

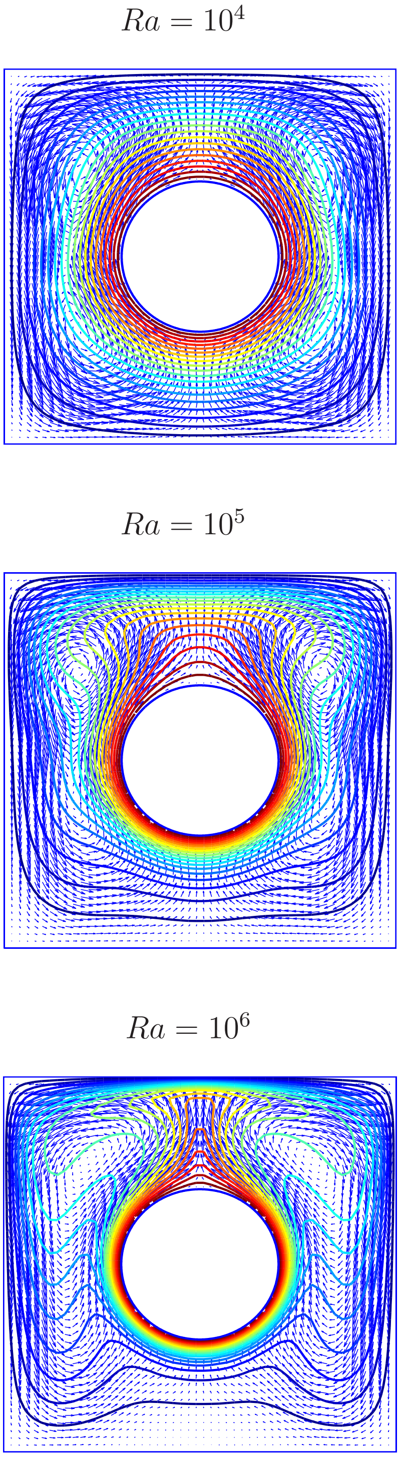
where $\overline{\Theta} = -\oint \frac{\partial T}{\partial n} ds$. Since the computational domain in (Moukalled and Acharya, 1996) is taken as one-half of the physical domain, values of Nu in the present work are divided by 2 for comparison purposes. Results concerning Nu for several values of Ra are shown in Table 5.2 along with those reported in (Moukalled and Acharya, 1996; Le-Cao et al., 2009; Shu and Zhu, 2002). It can be seen that they are in good agreement. Especially, for highly nonlinear solutions (e.g. $Ra = 10^6$), the result obtained from the proposed method is very close to that of the differential quadrature method (Shu and Zhu, 2002) but without the need to undertake coordinate transformation.

Table 5.2: Example 4, natural convection: Comparison of the average Nusselt number between the proposed method and some other methods for Ra in the range of 10^4 to 10^6

Ra	10^4	5×10^4	10^5	5×10^5	10^6
			Nu		
Proposed method	3.23	4.04	4.88	7.68	9.38
1D IRBF (Le-Cao et al., 2009)	3.22	4.04	4.89	7.43	8.70
FVM (Moukalled and Acharya, 1996)	3.24		4.86		8.90
MQ-DQ (Shu and Zhu, 2002)	3.33		5.08		9.37

5.5 Concluding remarks

In this chapter, a new approximation scheme for the time derivative term is proposed. The time stencil is based on two nodes over which IRBFs are employed to represent the field variable. In addition, apart from two nodal values of the field variable, its derivative value at the first node of the stencil is also included the approximation. When compared to conventional first-order FDs, numerical results indicate that larger time steps can be employed with the proposed time discretisation scheme. In this work, we combine the proposed time scheme with the space compact 5-point IRBF stencils, resulting in a numerical procedure, based on compact IRBF approximations only, for solving parabolic PDEs. The method is applied to simulate shallow water flows in large-scale domains and natural convection flows in multiply-connected domains, and produces accurate results using relatively large time steps.



$Ra = 10^4$

$Ra = 10^5$

$Ra = 10^6$

Figure 5.14: Example 4, natural convection: Velocity vector field and contour plots of the temperature for several values of *Ra* by the proposed method.

Chapter 6

Compact non-symmetric IRBF stencils and fictitious domains for complex fluid flows

This chapter presents a new non-boundary-fitted-grid numerical technique for simulation of incompressible viscous flows in multiply-hole domains. A multi-hole domain is converted into a simply-connected domain of rectangular or non-rectangular shape that is then discretised using a Cartesian grid. Compact radial basis function (RBF) stencils, which are presented in Chapter 3, are used to discretise the field variables. The imposition of inner boundary conditions is conducted by means of body forces that are derived from the local satisfaction of the governing equations and the prescribed boundary conditions. Salient features of the proposed method include: (i) simple pre-processing (Cartesian grid); (ii) high rates of convergence with respect to grid refinement achieved with compact integrated-RBF stencils and (iii) the system matrix kept unchanged for the case of moving holes. Several linear and nonlinear problems, including rotating-cylinder flows and buoyancy-driven flows in eccentric and concentric annuli, are simulated to verify the proposed technique.

6.1 Introduction

In solving partial differential equations (PDEs), multiply-connected domains (Figure 6.1) can be discretised using boundary-fitted and non-boundary-fitted grids/meshes. In the boundary-fitted-grid/mesh category, unstructured meshes are typically used. Boundary fitted grids/meshes can be used to represent a geometrically complex domain in an exact manner. Furthermore, an unstructured grid/mesh can be locally improved in selective regions to obtain refined information of the variable fields. However, generating an unstructured mesh is a time-consuming process. For the case of moving boundaries, the computational grid/mesh can be distorted. One needs to generate a new mesh and the variable field is then projected onto it, which are sophisticated tasks. Thus, the use of non-boundary-fitted grids/meshes to represent a multi-hole domain has received a great deal of attention (e.g. Parvizian et al., 2007, Husain et al., 2009, Mai-Duy and Tran-Cong, 2009, Buffat and Le Penven, 2011, Devendran and Peskin, 2012, Dechaume et al., 2010, Kang and Suh, 2011, Shi et al., 2012, Wang et al., 2017, Mo et al., 2018, Haji Mohammadi et al., 2019, Shankar et al., 2014). The basic idea of non-boundary-fitted-grid/mesh based methods is to extend the problem defined on a multiply-connected domain to that on a domain of simpler shape, where a regular grid/mesh and a fast algebraic solver can be used. In the case of moving interior holes, the grid/mesh may be kept unchanged. Consequently, the computational system matrix may be determined once and remains the same during the simulation process. In this category, special attention to the imposition of given boundary conditions is needed to match the solution on the extended domain with that on the physical domain. For this purpose, a body force is commonly introduced into the governing equations to describe the existence of the internal boundaries. Its main difficulty lies in a way used to obtain the body force field.

Many non-boundary-fitted-grid/mesh based methods have been reported in the literature. For example, Glowinski et al. (1994, 1998) proposed a class of

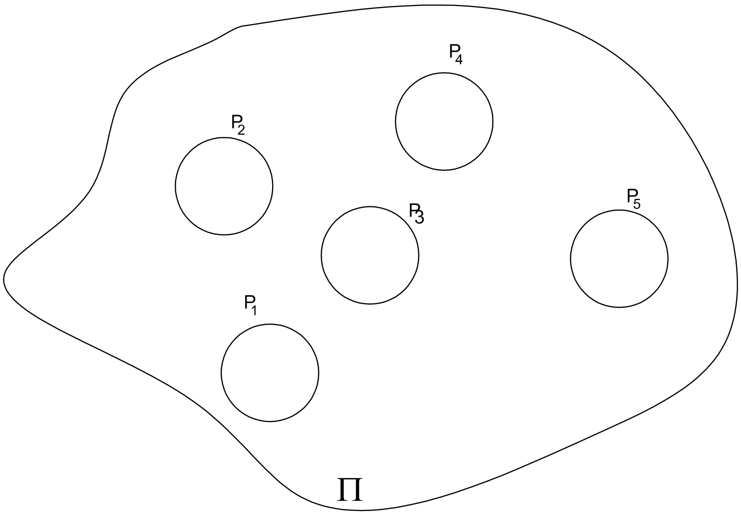


Figure 6.1: A typical multiply-connected domain.

fictitious-domain methods using Lagrange multipliers to enforce the inner conditions for simulating incompressible viscous flows. The methods were successfully implemented to solve practical problems such as rigid-body/fluid interactions (e.g. Patankar et al., 2000, Coesnon et al., 2008), fluid/flexible-body interactions (e.g. Yu, 2005, Van Loon et al., 2004, Shi et al., 2013), and particulate suspension flows (e.g. Glowinski et al., 1999, Wan and Turek, 2006, Dechaume et al., 2010).

Another approach is based on the immersed boundary method proposed in (Peskin, 1972). In this scheme, the body force was generated by the elasticity of the material and then "spread out" to grid nodes using Dirac delta functions. In (Fadlun et al., 2000), the body force was calculated based on the desired velocities at the interfaces. In (Uhlmann, 2005), the body force was first obtained on the immersed interfaces and was smoothly transferred to fixed grids by Dirac delta functions. In (Kim et al., 2001), an interpolation scheme for evaluating the velocities satisfying non-slip boundary conditions was proposed and the body force was then directly defined on grid nodes. Later on, Parvizian et al. (2007) introduced a finite cell method for solving problems of solid mechanics. Duster et al. (2008) extended the finite cell method to 3D-linear-elasticity problems. Maury (2001) proposed a fat boundary method (FBM) for solving PDEs in multi-hole domains. Vos et al. (2008) coupled the classic fictitious domain method and the FBM to constitute a method called the implicit FBM. Bertoluzza et al. Bertoluzza et al. (2005) implemented a semi-discrete FBM in the framework of FEM.

DRBFs and IRBFs were implemented with boundary-fitted grids for solution of the Navier-Stokes equation (Mai-Duy and Tran-Cong, 2008; Le-Cao et al., 2009; Le-Cao et al., 2011; Zhao et al., 2019; Xiao et al., 2015; Le et al., 2018). It should be pointed out that RBF system matrices are entirely populated and become ill-conditioned when a large number of nodes are used. Thus, recent RBF research has concentrated on solving these shortcomings. An efficient technique is to utilise local RBF stencils, where only a small subset of nodes are triggered for the approximation of a function at a given point, and compact local RBF stencils, where the approximations involve not only grid node function values, but also their derivative values. A sparse system matrix, which saves computer storage space and promotes the use of a much larger number of nodes, can be obtained with local schemes. Furthermore, the inclusion of derivative values can significantly improve accuracy of a local approximation scheme. Works reported in this research direction include (Le et al., 2018; Mai-Duy et al., 2017; Mai-Duy et al., 2018; Tien et al., 2015; Thai-Quang et al., 2012; Thai-Quang et al., 2013; Mai-Duy and Tran-Cong, 2013; Ahmad and Khaliq, 2017; Dehghan and Abbaszadeh, 2017; Lehto et al., 2017; Dehghan and Abbaszadeh, 2018; Pourbashash and Oshagh, 2018; Shu et al., 2003).

In this chapter, compact local IRBF stencils reported in Chapter 3 are incorporated into the non-boundary-fitted-grid (NBFG) framework for simulating fluid flows. Since compact local IRBF stencils can work on irregular grids, the desired velocities (i.e. the velocities take into account the inner boundaries) and the forcing terms can be evaluated directly at grid nodes without interpolation.

The solution procedure includes following steps:

- Step 1: Estimate fluid velocities in local domains near inner boundaries, where the forcing term is omitted.
- Step 2: Derive the forcing terms from the difference between the desired velocities and the estimated velocities obtained from Step 1. Note that the desired velocities are obtained by solving the governing equations subject to non-slip boundary conditions on local regions enclosing the inner boundaries.
- Step 3: Solve the governing equations in the extended domain with the obtained forcing terms.

The remainder of the chapter is organised as follows. In Section 6.2, the proposed IRBF-NBFG technique is described with emphasis placed on the formulation of forcing functions describing the influence of the interface on the solution. Details for IRBF discretisations of the governing equations in an extended (rectangular) domain are also included here. Numerical solutions are reported in Section 6.3. Section 6.4 concludes the chapter.

6.2 Proposed IRBF-NBFG technique

Consider a rectangular domain containing holes such as the one shown in Figure 6.2. The real domain $\mathbb{D}_R$ is extended to a regular domain $\mathbb{D}$ which is used for numerical simulation. The computational domain $\mathbb{D}$ thus comprises two sets of sub-region: the holes and the multiply-connected domain (i.e. original domain). A Cartesian grid is employed to discretise the extended/computational domain, and compact IRBF stencils are then applied for approximating the field variables. It is straight forward to implement the external boundaries. Nevertheless, appropriate schemes are needed to enforce the inner boundaries conditions as, generally, grid nodes do not lie on inner boundaries. The influence of the inner boundaries on the fluid flow is represented by forces $\mathbf{\Lambda}$ exerting on the fluid and correcting it to the velocity boundary conditions on the interfaces. Here, the momentum equations (2.11) are discretised in time by a first-order finite-difference scheme.

$$\begin{aligned}\boldsymbol{\nabla}\cdot\mathbf{u} &= 0, && (6.1)\\ \frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}+\boldsymbol{\nabla}p^{n} &= -\boldsymbol{\nabla}\cdot(\mathbf{u}^{n}\cdot\mathbf{u}^{n})+\frac{1}{Re}\boldsymbol{\nabla}^{2}\mathbf{u}^{n}+\boldsymbol{\Lambda}. && (6.2)\end{aligned}$$

where the superscript denotes the time level and $\boldsymbol{\Lambda}$ is the body force used to represent the influence of the inner boundaries on the fluid flow. It can be seen that, to solve equation (6.2), the forcing term $\boldsymbol{\Lambda}$ must be obtained in advance to force the flow solution to satisfy non-slip boundary conditions at the internal boundaries. The computation of $\boldsymbol{\Lambda}$ and the imposition of non-slip boundary conditions on the internal interfaces (e.g. holes' boundaries) will be presented in subsection 6.2.2.

In the remaining parts, we will use the notations:

- $\widetilde{[\ ]}$ to represent a vector/matrix $[\ ]$ which is associated with the whole computational domain $\mathbb{D}$,
- $\widehat{[\ ]}$ to represent a vector/matrix $[\ ]$ which is associated with a grid line of $\mathbb{D}$,
- $\overline{\overline{[\ ]}}_{|k}$ to represent a vector/matrix $[\ ]$ which is associated with the forcing domain $\mathbb{D}_{\mathbb{F}}$ of the kth hole,
- $\overline{[\ ]}_{|k}$ to represent a vector/matrix $[\ ]$ which is associated with a set of forcing points in a segment of $\mathbb{D}_{\mathbb{F}}$ of the kth hole,
- $([\ ])_{if|k}$ to denote the selected indexes (of the extended computational domain) which are associated with the set of forcing points of the kth holes.

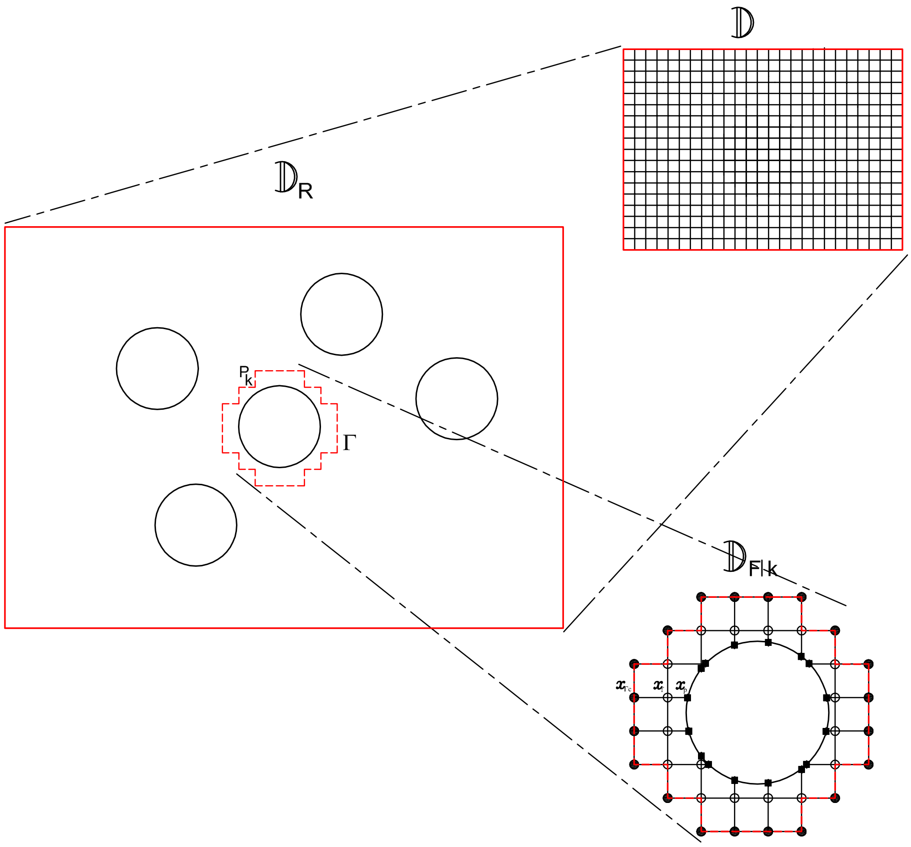


Figure 6.2: The physical domain ($\mathbb{D}_R$), the extended domain ($\mathbb{D}$) and the forcing domain of kth hole ($\mathbb{D}_{F|k}$). Location of forcing points and IRBF network calculation of desired velocities. Open circles ○ mark the forcing points of kth hole. Filled squares ■ indicate the inner boundaries ∂P_k. Filled circles ● indicate the boundary points of the frame $\Gamma_{|k}$

.

6.2.1 Compact IRBF stencils

Equations (2.15)-(2.16) involve the first and second-order derivative terms. Considering an x-grid line (Figure 6.3), one can make use of (2.27)-(2.27), $\partial^2 u/\partial x^2$ is expressed by

$$\frac{\partial^2 u(x)}{\partial x^2} = \sum_{i=1}^{m} w_i g_i(x) = \sum_{i=1}^{m} w_i \mathcal{I}_i^{(2)}(x), \tag{6.3}$$

Expressions for the first-order derivative and the function (field variable) are

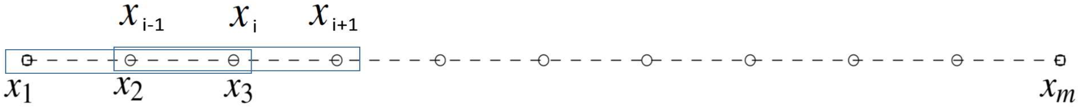


Figure 6.3: Points on a grid line of the extended domain $\mathbb{D}$.

then found by integration

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{m} w_i \mathcal{I}_i^{(1)}(x) + c_1, \tag{6.4}$$

$$u(x) = \sum_{i=1}^{m} w_i \mathcal{I}_i^{(0)}(x) + c_1 x + c_2, \tag{6.5}$$

Here, we implement the multiquadric (MQ) function which is

$$\begin{aligned}
\mathcal{I}_i^{(2)}(x) &= \sqrt{(x-c_i)^2 + a_i^2}, \\
\mathcal{I}_i^{(1)}(x) &= \frac{(x-c_i)}{2} A + \frac{a_i^2}{2} B, \\
\mathcal{I}_i^{(0)}(x) &= \left(\frac{-a_i^2}{3} + \frac{(x-c_i)^2}{6} \right) A + \frac{a_i^2 (x-c_i)}{2} B,
\end{aligned}$$

where c_i and a_i are the centre and the width of the ith MQ, respectively; $A = \sqrt{(x-c_i)^2 + a_i^2}$; and $B = \ln\left((x-c_i) + \sqrt{(x-c_i)^2 + a_i^2}\right)$. The set of collocation points are chosen to coincide with RBF centres. The influence domain here is a three-node stencil $[x_{i-1}, x_i, x_{i+1}]$ that is shifted along the grid line, where the index i runs from 2 to $(m-1)$. The IRBF approximations are based on three nodes rather than the whole set of nodes on the grid line. For compact stencils, second derivatives of the field variable obtained from the previous time level are incorporated into the approximation. We chose the width according to $a_i = \beta d_i$, where d_i is the shortest distance between c_i and its neighbours and β a scalar. Evaluation of (6.5) at x_{i-1}, x_i and x_{i+1}, and (6.3) at x_{i-1} and x_{i+1}

result in

$$\begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix} = \underbrace{\begin{bmatrix} \mathcal{I} \\ \mathcal{B} \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_{i-1} \\ w_i \\ w_{i+1} \\ c_1 \\ c_2 \end{pmatrix} \tag{6.6}$$

where

$$\mathcal{I} = \begin{bmatrix} \mathcal{I}_{i-1}^{(0)}(x_{i-1}) & \mathcal{I}_{i}^{(0)}(x_{i-1}) & \mathcal{I}_{i+1}^{(0)}(x_{i-1}) & x_{i-1} & 1 \\ \mathcal{I}_{i-1}^{(0)}(x_{i}) & \mathcal{I}_{i}^{(0)}(x_{i}) & \mathcal{I}_{i+1}^{(0)}(x_{i}) & x_{i} & 1 \\ \mathcal{I}_{i-1}^{(0)}(x_{i+1}) & \mathcal{I}_{i}^{(0)}(x_{i+1}) & \mathcal{I}_{i+1}^{(0)}(x_{i+1}) & x_{i+1} & 1 \end{bmatrix},$$

$$\mathcal{B} = \begin{bmatrix} \mathcal{I}_{i-1}^{(2)}(x_{i-1}) & \mathcal{I}_{i}^{(2)}(x_{i-1}) & \mathcal{I}_{i+1}^{(2)}(x_{i-1}) & 1 & 0 \\ \mathcal{I}_{i-1}^{(2)}(x_{i+1}) & \mathcal{I}_{i}^{(2)}(x_{i+1}) & \mathcal{I}_{i+1}^{(2)}(x_{i+1}) & 1 & 0 \end{bmatrix},$$

This system can be solved for the IRBF weights and two integration constants

$$\begin{pmatrix} w_{i-1} \\ w_i \\ w_{i+1} \\ c_1 \\ c_2 \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix} \tag{6.7}$$

where $\mathcal{C}^{-1}$ is the inverse of $\mathcal{C}$. Using (6.7), one can obtain the first and second derivatives of u at x_i as follows (6.4)

$$\frac{\partial u(x_i)}{\partial x} = \underbrace{\left[\begin{array}{ccccc} \mathcal{I}^{(1)}_{i-1}(x_i) & \mathcal{I}^{(1)}_{i}(x_i) & \mathcal{I}^{(1)}_{i+1}(x_i) & 1 & 0 \end{array} \right] \mathcal{C}^{-1}}_{\mathcal{D}_{1x}(x_i)} \begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix}, \quad (6.8)$$

and (6.3)

$$\frac{\partial^2 u(x_i)}{\partial x^2} = \underbrace{\left[\begin{array}{ccccc} \mathcal{I}^{(2)}_{i-1}(x_i) & \mathcal{I}^{(2)}_{i}(x_i) & \mathcal{I}^{(1)}_{i+1}(x_i) & 0 & 0 \end{array} \right] \mathcal{C}^{-1}}_{\mathcal{D}_{2x}(x_i)} \begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix}. \quad (6.9)$$

Similar to finite-difference and finite-element techniques, one will assemble these IRBF approximations to construct the global matrices $\widetilde{\mathcal{D}_{2x}}$ and $\widetilde{\mathcal{D}_{2y}}$. This task is fairly simple since the grid adopted here is regular. Expressions for computing derivative values of u at the interior grid nodes of extended domain can be written as

$\widetilde{\frac{\partial u}{\partial x}} = \widetilde{\mathcal{D}}_{1x}\widetilde{u} + \widetilde{k}_{1x}$, and $\widetilde{\frac{\partial^2 u}{\partial x^2}} = \widetilde{\mathcal{D}}_{2x}\widetilde{u} + \widetilde{k}_{2x}$, where the two vectors $\widetilde{k}_{1x}$ and $\widetilde{k}_{2x}$ are related to the boundary conditions and the imposed second derivative (compact components). In similar manner, one can obtain the compact IRBF discretisations for $\widetilde{\frac{\partial u}{\partial y}}$, $\widetilde{\frac{\partial^2 u}{\partial y^2}}$, $\widetilde{\frac{\partial v}{\partial x}}$, $\widetilde{\frac{\partial^2 v}{\partial x^2}}$, $\widetilde{\frac{\partial v}{\partial y}}$, $\widetilde{\frac{\partial^2 v}{\partial y^2}}$, $\widetilde{\frac{\partial p}{\partial x}}$, and $\widetilde{\frac{\partial p}{\partial y}}$.

6.2.2 Imposition of inner boundary conditions

To impose the boundary conditions on the inner boundaries, we use some iteration steps which are similar to those in the direct forcing immersed boundary

(Fadlun et al., 2000; Yusof, 1997). The inner boundary conditions are imposed by using the forcing terms $\mathbf{\Lambda}$ to force the solution in the extended domain $\mathbb{D}$ to match the solutions in the real domain $\mathbb{D}_R$. At the grid nodes, the forcing terms is defined as

$$\mathbf{\Lambda} = \frac{\mathbf{u}_d - \mathbf{u}_e}{\Delta t}, \tag{6.10}$$

where $\mathbf{u}_d = (u_d, v_d)$ the desired values of velocity when the non-slip boundary conditions at the inner boundaries are satisfied, $\mathbf{u}_e(u_e, v_e)$ the estimated values of velocity components which have not been taken into account the inner boundaries yet $\mathbf{u}_e$ can be obtained by

$$\mathbf{u}_e = \mathbf{u}^n - \Delta t \left(\boldsymbol{\nabla} p^n + \boldsymbol{\nabla} \cdot (\mathbf{u}^n \cdot \mathbf{u}^n) - \frac{1}{Re} \boldsymbol{\nabla}^2 \mathbf{u}^n \right). \tag{6.11}$$

To calculate $\mathbf{\Lambda}$, the desired velocities $\mathbf{u}_d$ must be determined in advance. For boundary points that coincide with the grid nodes (regular boundary points), one can apply (6.10) directly with $\mathbf{u}_d = \mathbf{u}_b$, where $\mathbf{u}_b(u_b, v_b)$ are given boundary values. Yet, in common, the position of boundary points do not match with the grid nodes and thus $\mathbf{u}_d$ are unknowns. A new computational scheme to resolve this problem is suggested as follows.

Imagine that a virtual staircase-shaped frame Γ encloses a kth hole as shown in Figure 6.2. The region lying between the kth hole and the frame Γ is considered to be a forcing domain, denoted by $\mathbb{D}_{F|k}$, which matches the estimated solution (in computational domain $\mathbb{D}$) with the real solution (in physical domain $\mathbb{D}_R$). In Figure 6.2, $\mathbf{x}_p(x_p, y_p)$ are boundary points of the kth hole ($\mathbf{x}_p \in \partial P_k$); $\mathbf{x}_f(x_f, y_f)$ denote for the coordinate of the forcing zone; and $\mathbf{x}_{\Gamma c}(x_{\Gamma c}, y_{\Gamma c})$ are boundary points of Γ. It can be seen that the forcing domain $\mathbb{D}_{F|k}$ has an irregular shape. Compact IRBF stencils can work with irregular grids and they were applied here to solve these problems. The centres of the new IRBFs collocating at $\mathbf{x}_p$

lie on the real boundary of kth hole. The centres of the new IRBFs collocating at $\mathbf{x}_f$ and $\mathbf{x}_{\Gamma c}$ coincide with the grid nodes of the Cartesian grid representing the extended domain $\mathbb{D}$.

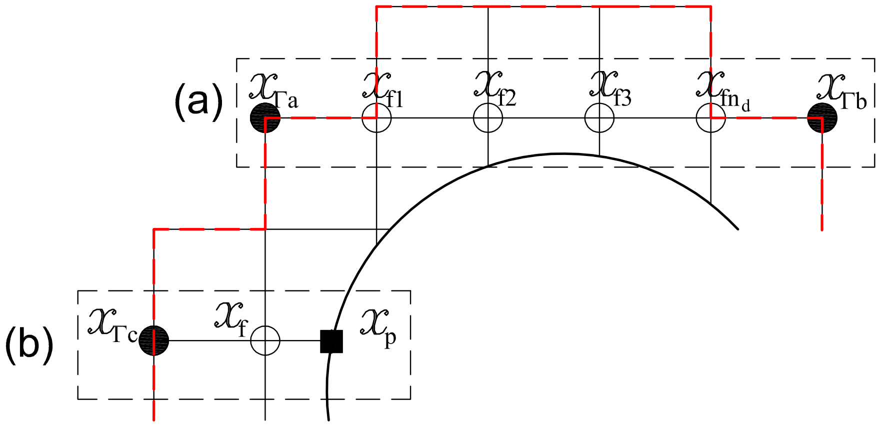


Figure 6.4: Points on a segment of forcing domain $\mathbb{D}_F$ of a holes.

Consider a segment that can be bounded by two faces of the frame (Figure 6.4a) or the boundary of the hole and the frame (Figure 6.4b). Assume a segment in the x direction and let consider u variable first. As demonstrated in Figure 6.4, a segment consists of two sets of points. The first set is n_d interior points that are also the forcing points (regular nodes). The desired values u_d at the forcing points ($\overline{x_f} = \left\{x_{f(i)}\right\}_{i=1}^{n_d}$) are unknown. The second set is constituted by the two boundary nodes x_{b1} and x_{b2}. Depending on how a segment is bounded, the boundary points x_{b1} and x_{b2} have specific locations. For example, one has ($x_{b1} \equiv x_{\Gamma a}$) and ($x_{b2} \equiv x_{\Gamma b}$) if a grid line is bounded by the two sides of the frame ($x_{b1} \in \Gamma$ and $x_{b2} \in \Gamma$), and ($x_{b1} \equiv x_{\Gamma c}$) and ($x_{b2} \equiv x_p$) if the bounding surfaces are the left side of Γ and the kth hole ($x_{b1} \in \Gamma$ and $x_{b2} \in \partial P_k$).

For the segment in Figure 6.4a, one can directly applied the IRBF approximation (6.8) and (6.9). However, for the segment in Figure 6.4b, one data point is not on regular grid nodes (generally). It is note that one has $n_d = 1$ in this case. Evaluation of (6.5) at collocation points $x_{\Gamma c}, x_f$ and x_p results in

$$\begin{bmatrix} u(x_{\Gamma c}) \\ u(x_f) \\ u(x_p) \end{bmatrix} = \overline{\varphi_{[0]}}_{|k} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ c_1 \\ c_2 \end{bmatrix}, \tag{6.12}$$

where

$$\overline{\varphi_{[0]}}_{|k} = \begin{bmatrix} \mathcal{I}_1^{(0)}(x_\Gamma) & \mathcal{I}_2^{(0)}(x_\Gamma) & \mathcal{I}_3^{(0)}(x_\Gamma) & x_\Gamma & 1 \\ \mathcal{I}_1^{(0)}(x_f) & \mathcal{I}_2^{(0)}(x_f) & \mathcal{I}_3^{(0)}(x_f) & x_f & 1 \\ \mathcal{I}_1^{(0)}(x_p) & \mathcal{I}_2^{(0)}(x_p) & \mathcal{I}_3^{(0)}(x_p) & x_p & 1 \end{bmatrix}.$$

It is emphasised that the function values at two boundary nodes are known (i.e. $u(x_{\Gamma c}) = u_e(x_{\Gamma c})$-calculated from (Equation 6.11) and $u(x_p) = u_b$-the given boundary condition).

The system (6.12) for the unknown vector of network weights can be calculated by

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ c_1 \\ c_2 \end{bmatrix} = \left(\overline{\varphi_{[0]}}\right)^{-1}_{|k} \begin{bmatrix} u(x_{\Gamma c}) \\ u(x_f) \\ u(x_p) \end{bmatrix}, \tag{6.13}$$

where $\left(\overline{\varphi_{[0]}}\right)^{-1}_{|k}$ is the inverse of $\overline{\varphi_{[0]}}_{|k}$.

Taking (6.13) into account, the values of the second derivative of u at the forcing point x_f are computed by (6.3)

$$\frac{\partial^2 u(x_f)}{\partial x^2} = \underbrace{\begin{bmatrix} \mathcal{I}_1^{(2)}(x_f) & \mathcal{I}_2^{(2)}(x_f) & \mathcal{I}_3^{(2)}(x_f) & 0 & 0 \end{bmatrix} \left(\overline{\varphi_{[0]}}\right)^{-1}_{|k}}_{\overline{\mathcal{D}}_{2x|k}=\begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix}} \begin{bmatrix} u(x_{\Gamma c}) \\ u(x_f) \\ u(x_p) \end{bmatrix}. \quad (6.14)$$

Since the values $u(x_{\Gamma c})$, and $u(x_p)$ are known, one can multiply with corresponding columns of the matrix $\overline{\mathcal{D}}_{2x|k}$ on the right hand side of equation (6.14) to form the the vectors of known quantities $\overline{d}_{2x|k}$. The approximated expression for second-order derivative of $u(x_f)$ are written in following form

$$\frac{\partial^2 u(x_f)}{\partial x^2} = D_2 u(x_f) + \underbrace{D_1 u(x_{\Gamma c}) + D_3 u(x_p)}_{\overline{d}_{2x|k}}. \quad (6.15)$$

The IRBF approximations for the derivatives are now expressed in terms of $u(x_f)$ nodal values and they now take into account the boundary conditions u_b. Hence, one only requires to put them to the governing equations. Assembling the obtained matrices on each segment for the whole forcing domain $\mathbb{D}_{F|k}$, one can obtain the following form for the kth hole.

$$\overline{\overline{\frac{\partial^2 u_d}{\partial x^2}}}_{|k} = \overline{\overline{\mathcal{D}}}_{2x|k} \overline{\overline{u}}_{d|k} + \overline{\overline{d}}_{2x|k}. \quad (6.16)$$

It is noted that $\overline{\overline{\frac{\partial^2 u_d}{\partial x^2}}}_{|k} = \frac{\partial^2 u(x_{(if|k)})}{\partial x^2}$.

The desired solution $\overline{\overline{u}}_{d|k}$ and $\overline{\overline{v}}_{d|k}$ around the kth hole is then determined by

$$\overline{\overline{u}}_{d|k} = \left(\overline{\overline{\mathcal{A}}}_{|k}\right)^{-1} \{(\widetilde{u}^n)_{if|k} - \Delta t \left(\widetilde{u}^n \widetilde{\frac{\partial u}{\partial x}}^n + \widetilde{v}^n \widetilde{\frac{\partial u}{\partial y}}^n\right)_{if|k} - \Delta t \left(\widetilde{\frac{\partial p}{\partial x}}^n\right)_{if|k} \ldots \tag{6.17}$$

$$-\frac{\Delta t}{Re}\left(\overline{\overline{d}}_{2x|k} + \overline{\overline{d}}_{2y|k}\right)\}, \tag{6.18}$$

and

$$\overline{\overline{v}}_{d|k} = \left(\overline{\overline{\mathcal{A}}}_{|k}\right)^{-1} \{(\widetilde{v}^n)_{if|k} - \Delta t \left(\widetilde{u}^n \widetilde{\frac{\partial v}{\partial x}}^n + \widetilde{v}^n \widetilde{\frac{\partial v}{\partial y}}^n\right)_{if|k} - \Delta t \left(\widetilde{\frac{\partial p}{\partial y}}^n\right)_{if|k} -\frac{\Delta t}{Re}\left(\overline{\overline{d}}_{2x|k} + \overline{\overline{d}}_{2y|k}\right)\}, \tag{6.19}$$

where $\overline{\overline{\mathcal{A}}}_{|k} = \left(\mathbf{1} - \frac{\Delta t}{Re}\left(\overline{\overline{\mathcal{D}}}_{2x|k} + \overline{\overline{\mathcal{D}}}_{2y|k}\right)\right)$. The desired values $\overline{\overline{u}}_{d|k}$ and $\overline{\overline{v}}_{d|k}$ are satisfied non-slip boundary condition at inner boundary ∂P_k and also the governing equations. In the same manner, one can gather the desired values $\left\{\overline{\overline{u}}_{d|k}\right\}_{k=1}^{np}$ and $\left\{\overline{\overline{v}}_{d|k}\right\}_{k=1}^{np}$ for np holes.

6.2.3 Solution Procedure

Step 1: The velocity fields $\widetilde{u}_e$ and $\widetilde{v}_e$ are estimated by Equation (6.11) for the extended domain.

Step 2: The forcing term $\mathbf{\Lambda}$ is calculated by

$$\widetilde{\Lambda_x} = \frac{\widetilde{u}_d - \widetilde{u}_e}{\Delta t}, \tag{6.20}$$

$$\widetilde{\Lambda_y} = \frac{\widetilde{v}_d - \widetilde{v}_e}{\Delta t}, \tag{6.21}$$

where $\widetilde{u_d}$ and $\widetilde{v_d}$ are obtained by assembling $\left\{\overline{\overline{u}}_{d|k}\right\}_{k=1}^{np}$ and $\left\{\overline{\overline{v}}_{d|k}\right\}_{k=1}^{np}$ in Equation (6.18) and (6.19). It is noted that the forcing term is zeros in $\mathbb{D} \setminus \mathbb{D}_F$.

Step 3: Calculate the velocity fields $\widetilde{u}^*$ and $\widetilde{v}^*$ by solving the momentum equations (6.2) with the obtained forcing term $\mathbf{\Lambda}$ in $\mathbb{D}$. To improve the stability, diffusion term $\mathbf{\nabla}^2\mathbf{u}$ is treated implicitly. It is noted that $\widetilde{u}^*$ and $\widetilde{v}^*$ are velocity components which have not satisfied (2.7) yet. The pseudo pressure variable is obtained by solving the following Poisson's equation

$$\left(\frac{\partial^2\widetilde{\phi}}{\partial x^2} + \frac{\partial^2\widetilde{\phi}}{\partial y^2}\right) = \frac{1}{\Delta t}\left(\widetilde{\frac{\partial u}{\partial x}}^* + \widetilde{\frac{\partial v}{\partial y}}^*\right). \tag{6.22}$$

The velocity variables are corrected by the pseudo pressure gradient term ϕ to satisfy the incompressibility constraint.

$$\widetilde{u}^{n+1} = \widetilde{u}^* - \Delta t\widetilde{\frac{\partial\phi}{\partial x}}, \tag{6.23}$$

$$\widetilde{v}^{n+1} = \widetilde{v}^* - \Delta t\widetilde{\frac{\partial\phi}{\partial y}}. \tag{6.24}$$

It is emphasised that, for problems with moving inner boundaries all the system matrices are remain unchanged during solving process.

6.3 Numerical examples

For all cases in this chapter, IRBF networks are performed with the MQ function. The solution accuracy is measured through the discrete relative L_2 norm of the error determined as

$$Ne = \frac{\sqrt{\sum_{i=1}^{n_{ip}}(u_i^{(e)} - u_i)^2}}{\sqrt{\sum_{i=1}^{n_{ip}}(u_i^{(e)})^2}}, \tag{6.25}$$

where n_{ip} is the number of interior points in the real domain, and $u^{(e)}$ and u are the exact and numerical solutions, respectively.

For grid refinement study, the convergence rate of the solutions is calculated by α in

$$Ne(h) \approx \beta h^{\alpha} = O(h^{\alpha}), \tag{6.26}$$

where β and α are exponential model's parameters and h is the average grid size. With a set of measurements, those parameters can be determined by the common linear least-squares method.

6.3.1 Example 1 - Poisson's equation

Two particular physical domain governed by a Poisson's equation are considered

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b(x, y), \tag{6.27}$$

where $b(x, y)$ is the driving function.

A domain with three holes

Here, we interest a square domain, $[0, 1] \times [0, 1]$, with three circular holes of radius $R = 0.2$ and their centres located at positions (0.65,0.4), (0.4,0.8) and (0.25,0.25). The exact solution to this example is

$$u^{(e)}(x, y) = \sin(2\pi x) \sin(2\pi y), \tag{6.28}$$

and the driving function $b(x, y)$ and Dirichlet boundary conditions can be calculated exactly. The interested domain is now embedded in a square one which can be effectively discretised by a uniform Cartesian grid. Both Dirac Delta

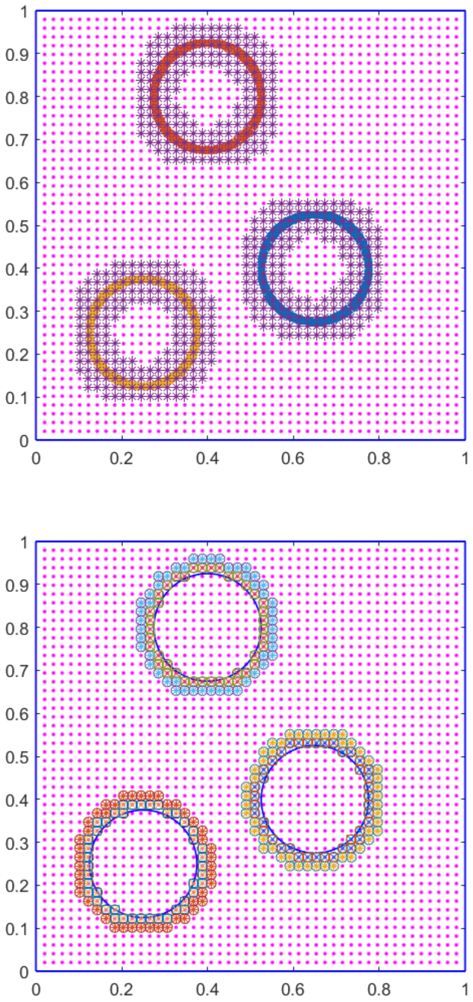


Figure 6.5: Example 1: Poisson's equation in three holes domain: Forcing point area by the present IRBF-NBFG technique method (bottom) and Peskin interpolation (top).

functions (Figure 6.5 Top) and RBFs (Figure 6.5 Bottom) are employed here for coupling the inner boundaries. Table 6.1 displays numerical accuracy obtained by the IRBF-NBFG method and Dirac Delta interpolations. The proposed scheme outperforms the Dirac Delta interpolations with respect to both convergence rate and accuracy. For example, to reach the accuracy of 8.10^{-4} the Dirac Delta interpolations needs a grid of 90×90 while only 20×20 with the IRBF-NBFG method. The proposed scheme generates a good convergence rate of $O(h^{3.40})$.

Table 6.1: Example 1: Poisson's equation in three holes domain: Numerical accuracy obtained by Peskin interpolation method and the proposed RBF-NBFG method. It is noted that $a(b)$ represents $a \times 10^b$.

nx	Peskin interpolation	Proposed IRBF-IBM
10	2.5641(-1)	7.3264(-3)
20	4.4228(-2)	8.5056(-4)
30	1.7652(-2)	1.4341(-4)
40	9.4371(-3)	5.8035(-5)
50	5.8697(-3)	2.8929(-5)
60	3.9919(-3)	1.6350(-5)
70	2.0810(-3)	1.0075(-5)
80	1.7256(-3)	6.6003(-6)
90	9.0921(-4)	4.5329(-6)
100	8.0924(-4)	1.2267(-6)

A domain with more than 3 holes

In this problem, the driving function $b(x, y) = -1$ is taken and homogeneous Dirichlet boundary conditions. The interested domain is a unit square with 9 holes of radius 0.0625. This example provides a good means of testing the ability of the IRBF-NBFG method in dealing with problems with multi-hole domains. It is known that these geometrically-complex-domains can be found in numerous practical situations such as fluid flows in a porous medium, particulate suspensions, or the thermal conductivity of composite materials, etc. A regular discretisation of the IRBF-NBFG scheme and that of FEM are displayed in Figure 6.6. It can be seen that the pre-processing process of the present technique is much simpler than that of FEM. We plot a visual comparison of the contour of the solution u between the IRBF-NBFG technique (grid of 100×100) and FEM obtained using the MATLAB PDE Toolbox as the exact solution is unavailable here. Figures (6.7-6.8) show that the two solutions have comparable variations.

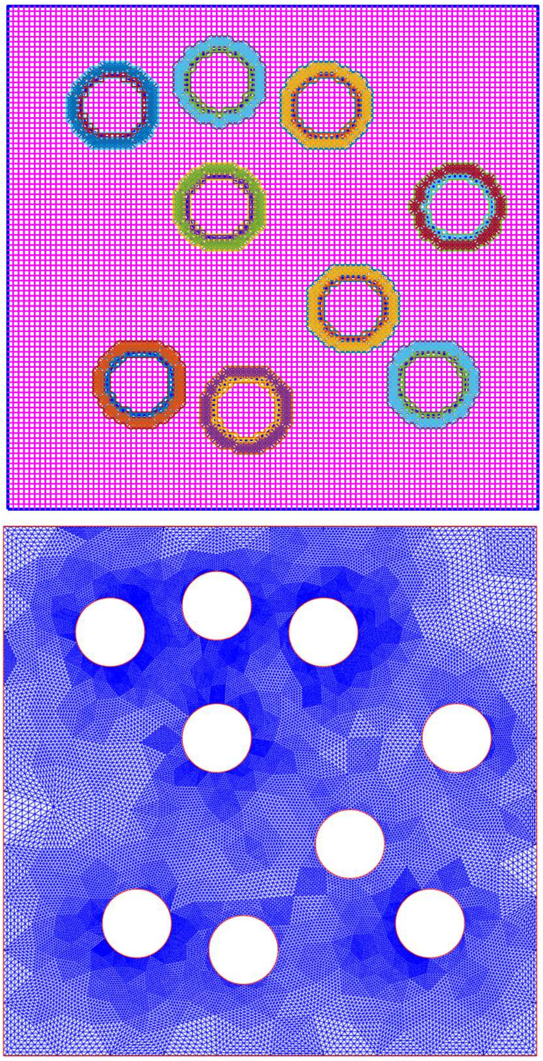


Figure 6.6: Example 1: Poisson's equation in multi holes domain: Discretisation by the present IRBF-NBFG technique method (top) and FEM (bottom).

6.3.2 Example 2 - Parabolic equation

Here, we interest in a problem governed by the parabolic PDE

$$\frac{\partial u}{\partial t}-\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=\sin(\pi x)\sin(\pi y)\left(1000e^{-1000t}+2k^2\pi^2\left(1-e^{-1000t}\right)\right), \tag{6.29}$$

in which k is a given value. Here, we choose $k=3$. Fig. 6.9 shows the problem domain which is the region between a circle with radius of 1 and three smaller

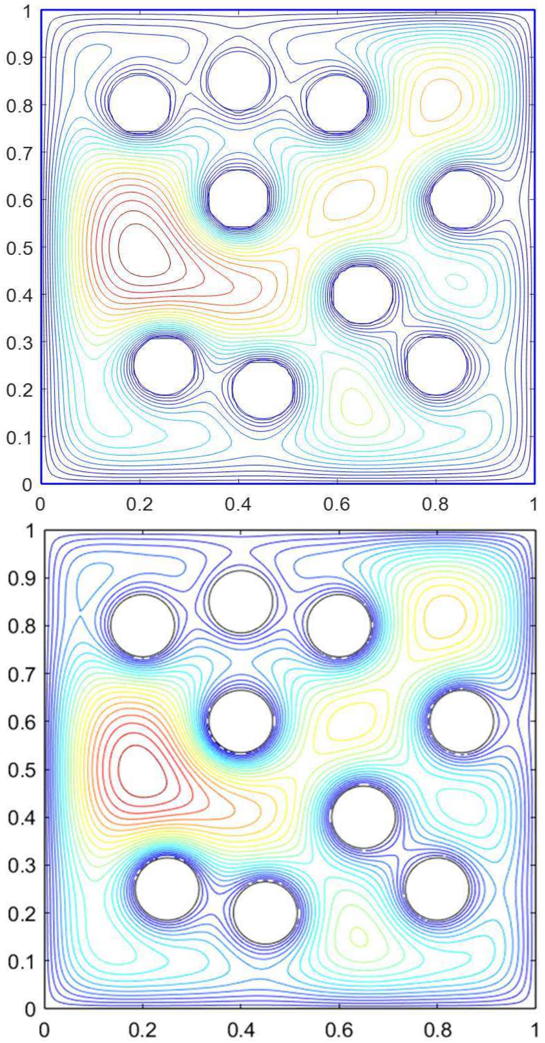


Figure 6.7: Example 1 (boundary value problem): A contour plot of u by the present IRBF-NBFG technique using grid of 100×100 (top) and FEM (bottom).

circles of radius 0.125.

The initial solutions, Dirichlet boundary conditions on the internal circular boundaries and Neumann boundary conditions on the external boundary can be computed exactly from the problem's solution given by

$$u^{(e)}(x,y,t)=\sin(k\pi x)\sin(k\pi y)t. \tag{6.30}$$

Results concerning Ne using a time step of 0.01 and the spatial discretisation of 20×20 to 80×80 are listed in Table 6.2. The system matrix condition numbers

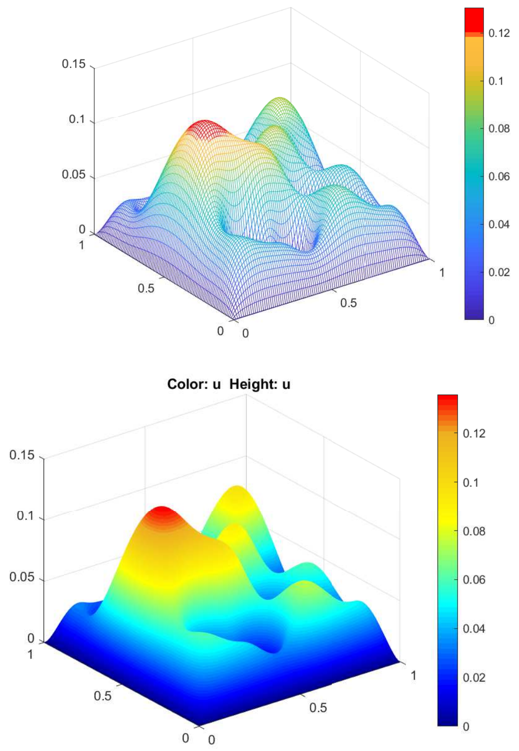


Figure 6.8: Example 1 (boundary value problem): A mesh of u by the present IRBF-NBFG technique using grid of 100×100 (top) and FEM (bottom).

are 3.5×10^3 for a grid of 40×40 and 5.4×10^3 for 80×80. It can be seen that the proposed IRBF-NBFG technique can accurately approximate the problem solutions.

6.3.3 Example 3: Cylinder-driven flows

Case 1: Rotating cylinder

This test problem is employed for the simulation of a steady incompressible viscous flow defined in an annulus between two concentric cylinders (the circular inner and square-shaped outer cylinders). The flow geometry is shown in Figure 6.10 and its discretisation is shown on Figure 6.11. The flows are induced by

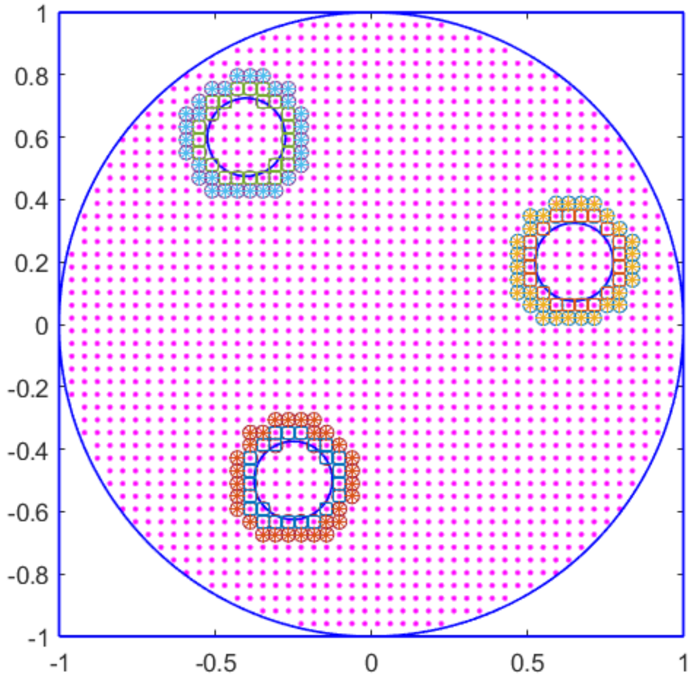


Figure 6.9: Example 2: parabolic equation in three holes domain: A typical discretisation using grid of 50×50.

assigning a constant angular velocity Ω to the inner cylinder. The values of u and v on the outer wall are simply fixed to zero, while the value of u and v on the inner wall are set as $u = -\Omega y$ and $v = \Omega x$.

The problem domain is extended to a rectangular one which is discretised by uniform Cartesian girds. The diffusion and convection terms can be treated implicitly and explicitly, respectively. Here, the first-order finite-difference scheme is used to discretise the solution concerning the time derivative. At the time $t = 0$, one requires to choose the initial values of all the variable fields (e.g. using a lower-Re solution). For the case of $Re = 100$, the initial values can commonly be set to zeros. Then, the problem solution is computed and updated till a steady-state is reached.

The governing equations (2.14)-(2.16) need be calculated simultaneously to determine the values of the two components of velocity field and pressure at the

Table 6.2: Example 2: Parabolic equation in three holes domain: Numerical accuracy obtained by proposed RBF-NBFG when refining mesh. It is noted that $a(b)$ represents $a \times 10^b$.

t	$nx = 20$	$nx = 30$	$nx = 40$	$nx = 50$	$nx = 60$	$nx = 70$	$nx = 80$
0.01	3.16(-4)	9.99(-5)	3.90(-5)	1.92(-5)	1.77(-5)	1.03(-5)	6.01(-6)
0.11	7.34(-5)	2.14(-5)	8.88(-6)	5.67(-6)	3.39(-6)	2.00(-6)	1.25(-6)
0.21	4.72(-5)	1.36(-5)	6.73(-6)	5.93(-6)	1.98(-6)	1.00(-6)	7.12(-7)
0.31	5.19(-5)	1.46(-5)	7.29(-6)	6.36(-6)	1.90(-6)	9.09(-7)	6.84(-7)
0.41	5.33(-5)	1.51(-5)	7.52(-6)	6.57(-6)	1.94(-6)	8.89(-7)	7.00(-7)
0.51	5.44(-5)	1.53(-5)	7.58(-6)	6.59(-6)	1.89(-6)	8.68(-7)	6.83(-7)
0.61	5.43(-5)	1.54(-5)	7.62(-6)	6.63(-6)	1.91(-6)	8.69(-7)	6.92(-7)
0.71	5.46(-5)	1.54(-5)	7.62(-6)	6.62(-6)	1.88(-6)	8.64(-7)	6.84(-7)
0.81	5.45(-5)	1.54(-5)	7.63(-6)	6.63(-6)	1.89(-6)	8.63(-7)	6.89(-7)
0.91	5.46(-5)	1.54(-5)	7.62(-6)	6.63(-6)	1.88(-6)	8.64(-7)	6.86(-7)

discrete points within the domain. First derivatives of the pseudo pressure on boundaries are utilised to derive Dirichlet boundary conditions for Poisson equation (6.22) (Thai-Quang et al., 2012). Consequently, all the boundary conditions of governing equations are Dirichlet. The projection method is employed for solving fluid variables. At each time interval, the solution procedure involves the following main steps:

a. Guessing the initial values of u, v and p

b. Discretising the equations (2.14)-(2.16) in time using a finite-difference scheme

c. Discretising the equations (2.14)-(2.16) in space using the compact IRBF stencils discretisation scheme. Because the differentiation matrices are identical for all variable fields, the matrix establishment process only requires to be done for one time. The system matrices which includes the IRBF approximations for the first and second differential terms of the governing equations, keep unchanged during the computational loop.

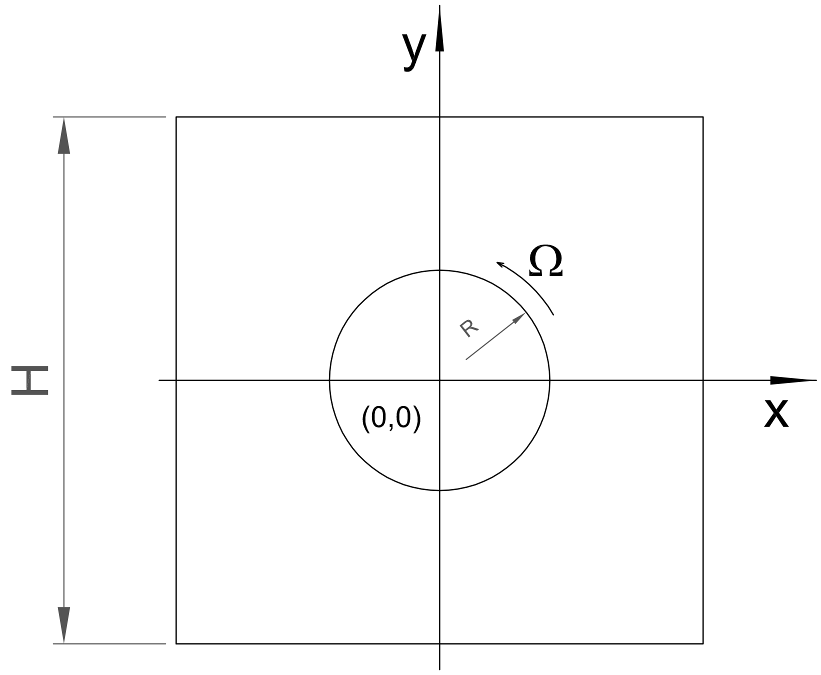


Figure 6.10: Example 3 (rotating cylinder): geometry.

d. Computing desired values u_d and v_d using equations (6.18)-(6.19), respectively.

e. Obtaining the forcing terms $\mathbf{\Lambda}$ by equations (6.20) and (6.21).

f. Solving the momentum equations with the obtained forcing term $\mathbf{\Lambda}$ to get $\widetilde{u}^*$ and $\widetilde{v}^*$.

g. Deriving the boundary conditions for ϕ and solving the pressure Poisson's equation (6.22) for ϕ.

h. Correcting velocity fields in equations (6.23)-(6.24) to satisfied the incompressibility constraint (2.7).

i. Checking the steady state by the convergence measure (CM) defined as follows: The maximum values of $CM[u], CM[v]$, and $CM[p]$ is chosen to

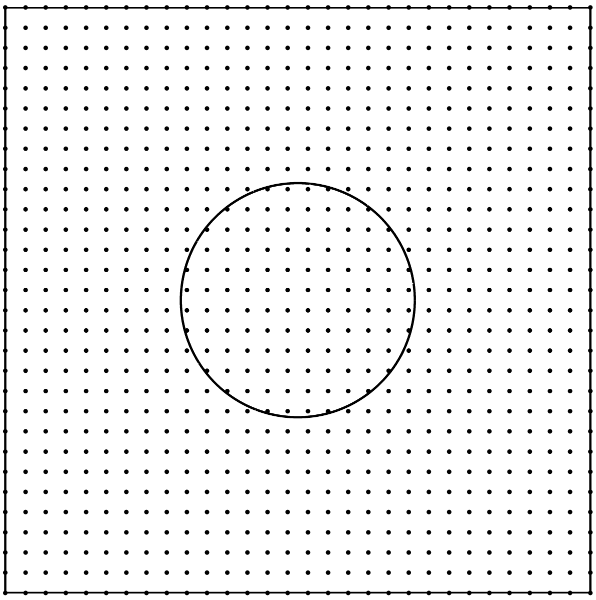


Figure 6.11: Computational domains and discretisations. It is noted that the real domain is the region between inner circular cylinder and outer square cylinder.

be CM.

$$CM[.] = \frac{\sqrt{\sum_{i=1}^{n_{ip}} \left([.]_i^{(l+1)} - [.]_i^{(l)}\right)^2}}{\sqrt{\sum_{i=1}^{n_{ip}} \left([.]_i^{(l+1)}\right)^2}}, \tag{6.31}$$

where [.] can be u, v, or p, n_{ip} is the number of interior points in the real domain, and l the time level. $CM < \epsilon$, where ϵ the tolerance (here, ϵ is chosen to be 10^{-10}).

In this example, the flow is simulated with $\Omega = 1.0$ using a uniform grid of 100×100. Several values of the Reynolds number, including $\{100, 200, 500, 700\}$, are studied. For comparison purposes, the stream function and vorticity variables can be derived by solving the following equation.

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{6.32}$$

Results concerning the maximum value of stream function ψ_{max} and vorticity ω_{max} calculated by the IRBF-NBFG scheme and the finite-difference scheme (Lewis, 1979) are displayed in Table 6.3-6.4, giving an adequate agreement. In Figure 6.12, the performance of the convergence measure CM versus the total number of time steps is provided. It can be observed that the calculation of high-Re number flows needs a higher number of steps. Plots of the velocity vector field and pressure field for the cases of $Re = \{200, 700\}$ are presented in Figure 6.13.

Table 6.3: Example 3 (rotating cylinder): Comparison of the maximum values of stream-function ψ, between the present IRBF-NBFG technique (grid of 100×100) and finite difference technique for several values of Re.

Re	100	200	500	700
		ψ		
Present	0.4520	0.4546	0.4553	0.4550
FDM (Lewis, 1979)	0.4656	0.4539	0.4465	0.4423

Table 6.4: Example 3 (rotating cylinder): Comparison of the maximum values of vorticity ω, between the present IRBF-NBFG technique (grid of 100×100) and finite difference technique for several values of Re.

Re	100	200	500	700
		ω		
Present	1.1154	1.2660	1.3717	1.3937
FDM (Lewis, 1979)	1.0186	1.2559	1.3430	1.3693

Case 2: Moving cylinder

The second example is the same as the first one, except that the cylinder is

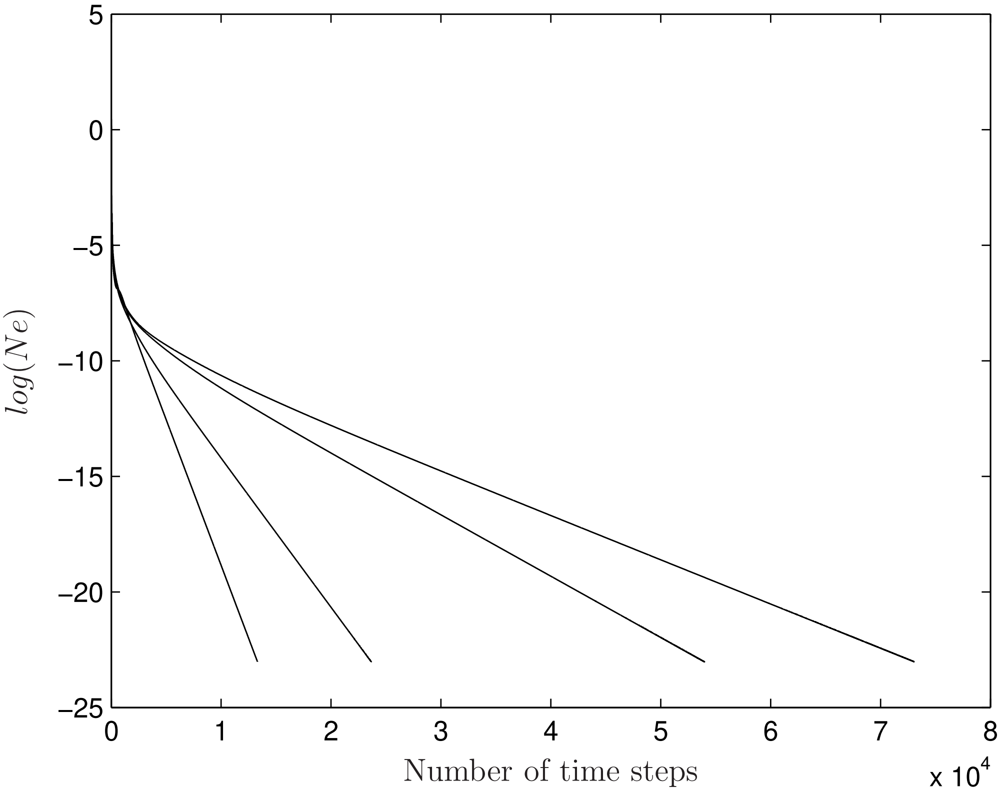
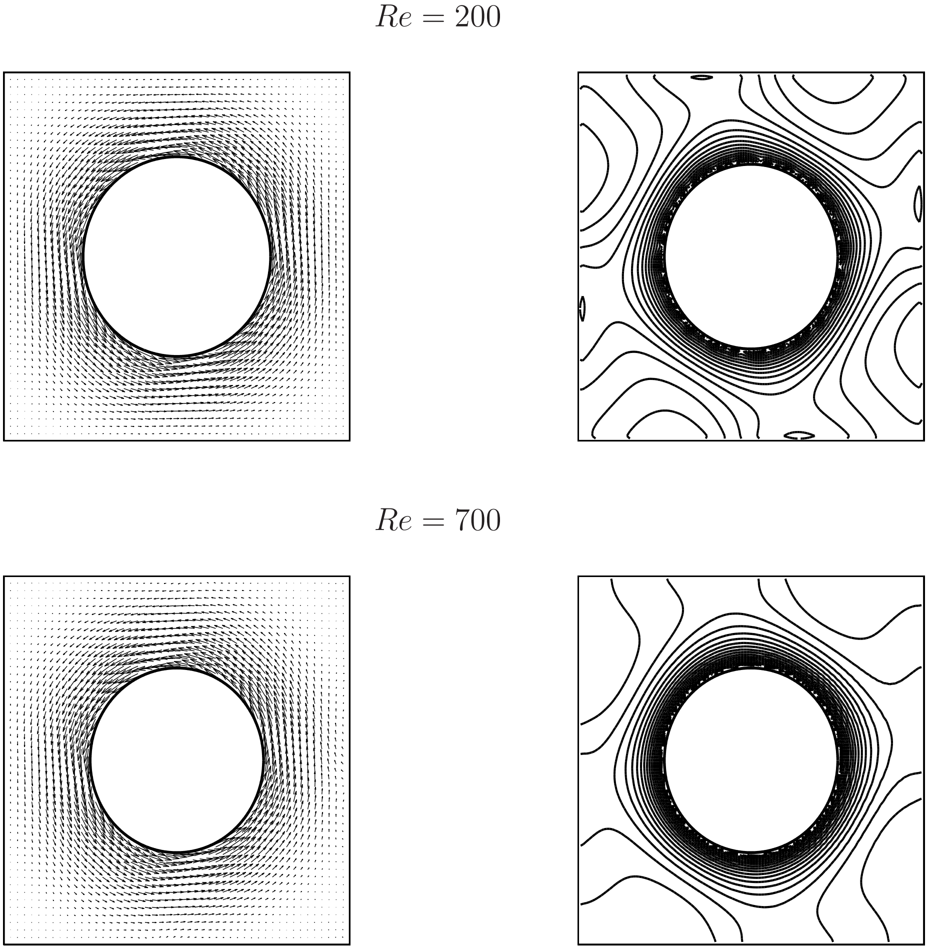


Figure 6.12: Example 3 (rotating cylinders): Iterative convergence. The values of CM become less than 10^{-10} when the numbers of iterations reach 13308, 23659, 53974, and 73086 for $Re = \{100, 200, 500, 700\}$, respectively. Using the last point on the curves as a positional indicator, from left to right the curves correspond to $Re = \{100, 200, 500, 700\}$.

repositioned after a certain number of time steps. The flow geometry is similar to Figure 6.10. The cylinder's radius is 0.1 and the angular velocity Ω is also given constant of 1. In this case the $Re = 10$ is considered. Figure 6.14 displays the velocity fields with four $y-$positions of the cylinder including $y_e = \{0.15, 0.1, 0.05, -0.02\}$.

Case 3: Multiple cylinders

This example is to verify the proposed technique in dealing with fluid flows in geometrically-complex-domains. The rotating cylinder problem is extended



$Re = 200$

$Re = 700$

Figure 6.13: Example 3 (rotating cylinder): Velocity vector field (left) and pseudo pressure field (right) for the flow at $Re = \{200, 700\}$.

to the case of multi-cylinder which are fixed and rotate at the same angular velocity. Here, the domain of interest is a unit square cylinder with 9 circular cylinders of radius 0.02. The cylinders are located at (0.15,-0.1), (-0.1,0.35), (-0.25,-0.25), (-0.1,+0.1), (-0.3,0.3), (-0.05,-0.3), (0.35,+0.1), (0.1,+0.3) and (0.3,-0.25). The discretisation of the IRBF-NBFG technique is similar to that of Example 1 Poisson's equation in multi holes domain (Figure 6.6). The pre-processing for this case is much more convenient, since these radii are uniform, one just updates the location of forcing points by an amount of eccentric (x_e, y_e). We present a visual the distribution of u and stream function ψ (grid of 100×100) on Figure 6.15 and 6.16, respectively.

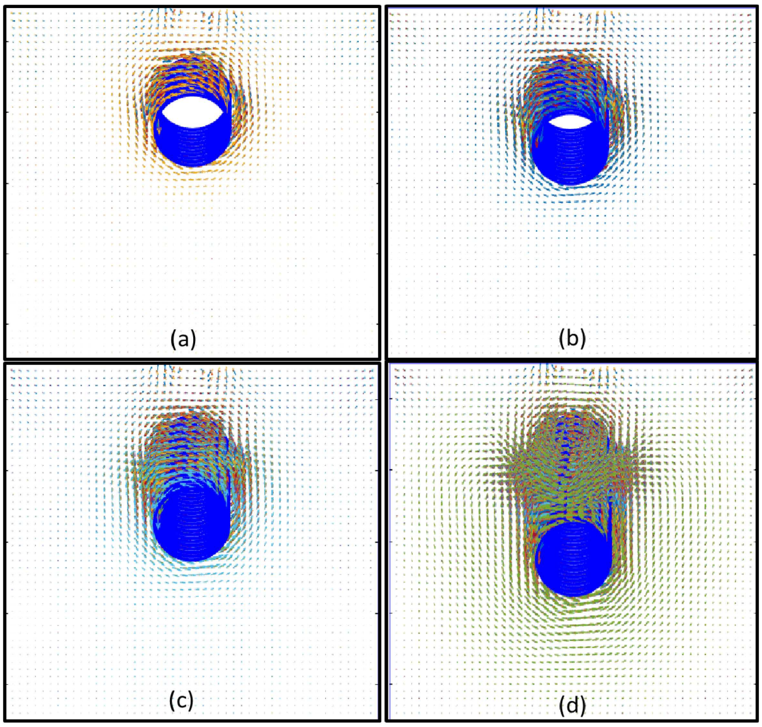


Figure 6.14: Example 3 (rotating cylinder): Velocity vector field of the viscous flow for the moving cylinder with predefined velocity.

6.3.4 Example 4: Buoyancy driven flows in double-connected domain

For this example, buoyancy driven flows between a heated internal circular cylinder and a cooled external square enclosure is studied. These flows have been widely investigated by experimental works as well as simulations. For the latter, many numerical methods were carried out such as FDM (e.g. Kuehn and Goldstein, 1976; Davis, 1983), FEM (e.g. Manzari, 1999; Sammouda et al., 1999), FVM (e.g. Glakpe et al., 1986; Moukalled and Acharya, 1996), RBFN (e.g. Šarler et al., 2004; Ho-Minh et al., 2009) and spectral techniques (e.g. Le Quere, 1991; Shu, 1999).

For non-isothermal flows, with the Boussinesq approximation (Ostrach, 1988),

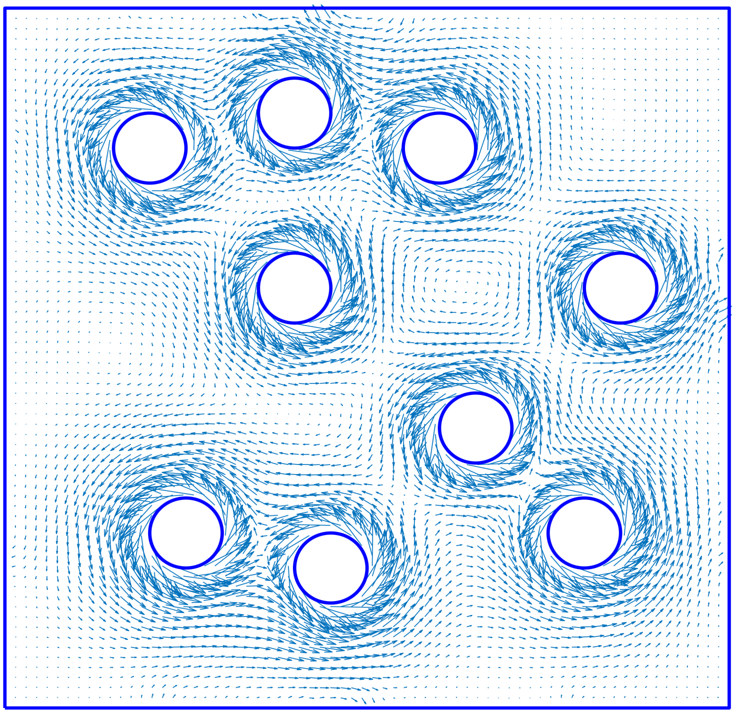


Figure 6.15: Example 3 (rotating cylinder): Velocity vector field for the multi-connected domains with 9 cylinders.

the governing equations in two dimensions can be written as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{6.33}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \sqrt{\frac{Pr}{Ra}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f_x, \tag{6.34}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \sqrt{\frac{Pr}{Ra}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + T + f_y, \tag{6.35}$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \frac{1}{\sqrt{RaPr}}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \tag{6.36}$$

where T is the temperature, $\boldsymbol{f} = (f_x, f_y)$ the body force vector. Ra and Pr are the Rayleigh and Prandtl numbers determined by $Ra = \kappa g \Delta T L^3/\alpha\nu$ and $Pr = \nu/\alpha$, respectively in which κ is the thermal expansion coefficient, α the thermal diffusivity coefficient, g the gravity, and ΔT and L the characteristic temperature difference and length, respectively. Here, the velocity scaling $U = \sqrt{gL\beta\Delta T}$ is used to balance the inertial and buoyancy forces.

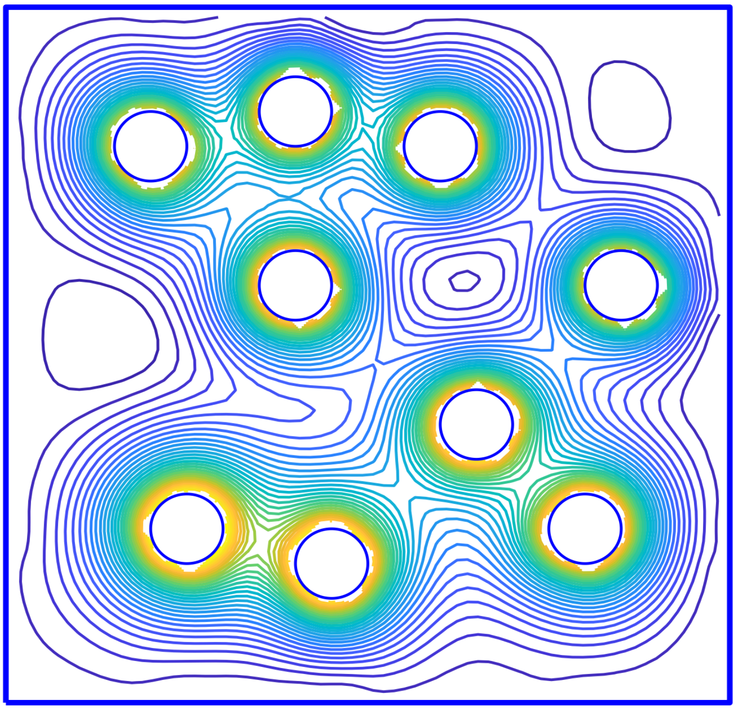


Figure 6.16: Example 3 (rotating cylinder): Stream function contours for the multi-connected domains with 9 cylinders.

Similar to Example 3, a Cartesian grid (Figure 6.11) is utilised to discretise the annulus domain. Parameters for the simulations include an aspect ratio of H/D_i (where D_i: the diameter of the internal hole and H: the length of the external square) and $Pr = 0.71$. The width of RBF is chosen a constant 5 for all simulations. For comparison purposes, the stream function can be derived by solving the following equation.

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{6.37}$$

Results and discussion

The obtained results are presented in the forms of velocity, pressure and temperature fields (Figures 6.18) with respects to three radii $H/D_i = 5, 2.5$ and 1.67. In Figure 6.19, stream function and temperature plots contain 21 contour lines which levels range linearly from the smallest to highest values for the ec-

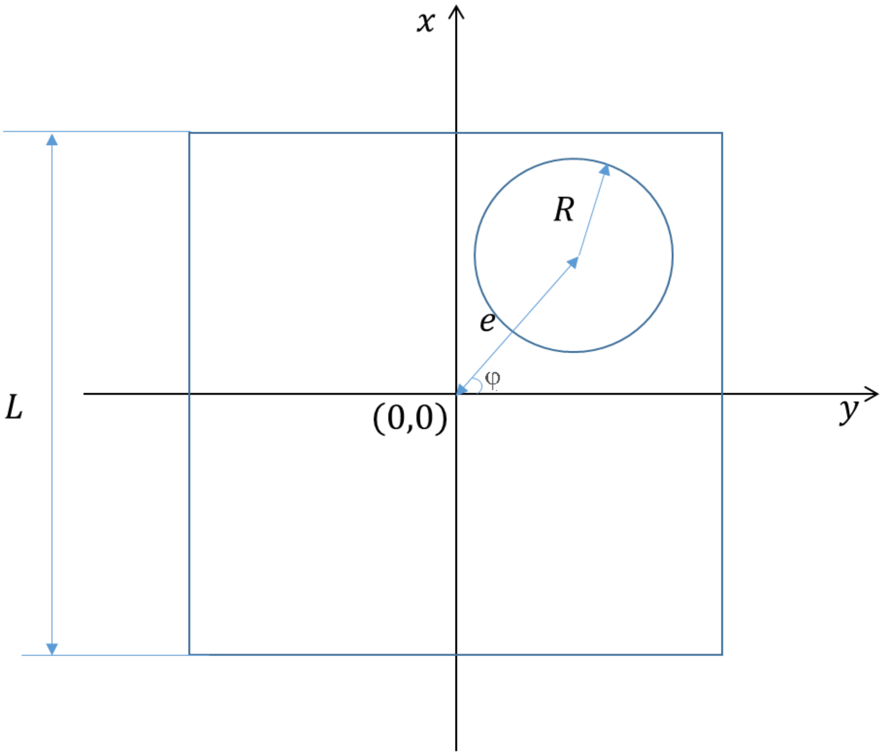


Figure 6.17: Example 4 (buoyancy flows in the square-circular annuli): geometry.

centric cases. The figure demonstrates that the current IRBF-NBFG results are in very good agreement with those presented in (Ding et al., 2005). Here, the local heat transfer coefficient and the average Nusselt number are determined by Moukalled and Acharya (1996)

$$\theta = -k\frac{\partial T}{\partial n}, \tag{6.38}$$

where k the thermal conductivity, and

$$Nu = \frac{\overline{\theta}}{k}, \tag{6.39}$$

where $\overline{\theta} = -\oint \frac{\partial T}{\partial n} ds.$

Results concerning the maximum value of stream function (ψ_{max}) and the average Nusselt number for five values of Ra , namely $\{1 \times 10^4, 5 \times 10^4$ (uniform grids of 60×60) and $1 \times 10^5, 5 \times 10^5, 1 \times 10^6$ (uniform grids of 84×84)}, with a time step of 1×10^{-4} are presented in Table 6.5 and Table 6.6, respectively. For $Ra = 10^4$, the initial solution is set as zero and for higher values of Ra, the initial solution is chosen by the solution at the most next lower Ra. These results agree well with those in (Le-Cao et al., 2009; Kuehn and Goldstein, 1976; Moukalled and Acharya, 1996; Shu and Zhu, 2002 and Ding et al., 2005). We also consider the shifting circular boundary, where the centre of the internal cylinder moves inside the external square. Varying amounts of position of the cylinder centre (e), $\{0.25, 0.5, 0.75 \text{ and } 0.95\}$, are considered. Results concerning ψ_{max} together with those of (Ding et al., 2005) for $Ra = 3 \times 10^5$ are displayed in Table 6.7. A good agreement between the results obtained by IRBF-NBFG scheme and those of the reported boundary fitted grid methods can be observed. The isotherms and streamlines of the solution flow for $Ra = 3 \times 10^5$ using a grid of 60×60 are plotted in Figures 6.18 and 6.19. Each plot comprises 24 contour lines which have levels varying linearly from the lowest to highest values.

Table 6.5: Example 4 (buoyancy flows in the square-circular annuli): Comparison of the maximum value of stream function ψ_{max} for Ra from 10^4 to 10^6 between the present technique and some other techniques.

Ra	10^4	5×10^4	10^5	5×10^5	1×10^6
			ψ_{max}		
Present method	1.04	5.13	8.34	19.94	24.29
MQ-DQ (Ding et al., 2005)	1.00		8.32		24.13
FVM(Moukalled and Acharya, 1996)	1.02		8.38		24.07

6.4 Concluding remarks

In this work, a new non-boundary-fitted-grid method is reported. Compact integrated RBF stencils are utilised to discretise the field variables on the computational domains, and the forcing terms are directly estimated from the local

Table 6.6: Example 4 (buoyancy flows in the square-circular annuli): Comparison of the average Nusselt number, Nu, for Ra from 10^4 to 10^6 between the present technique and some other techniques.

Ra	10^4	5×10^4	10^5	5×10^5	10^6
			Nu		
Present method	3.13	4.23	5.35	7.11	9.30
1D-IRBFN(Le-Cao et al., 2009)	3.22	4.04	4.89	7.43	8.70
DQM (Shu and Zhu, 2002)	3.24	4.02	4.86	7.53	8.90
FDM (Kuehn and Goldstein, 1976)	3.33		5.08		9.37

Table 6.7: Example 4 (buoyancy flows in the square-circular annuli): Comparison of the maximum stream-function values, ψ_{max}, for special cases $\varphi = \{-90^0, 90^0\}$ between the present technique and MQ-DQ technique.

e	0.25	0.5	0.75	0.95
φ		-90^0		
		ψ_{max}		
Present method	17.8	20.75	22.0	22.97
MQ-DQ (Ding et al., 2005)	18.64	21.29	23.52	
φ		90^0		
		ψ_{max}		
Present method	12.7	11.06	10.90	9.57
MQ-DQ (Ding et al., 2005)	12.39	11.38	10.09	

satisfaction of the governing equations. Unlike other immersed boundary methods, no interpolation between Lagrange and Euler grid is required here. The proposed method is successfully verified in several practical problems. Numerical results show that a high convergence rate is achieved and the matrix condition number is relatively small. These attractive features together with advantages of using non-boundary-fitted grids allow an efficient scheme to be developed for the numerical study of complex structure fluids such as particulate suspensions.

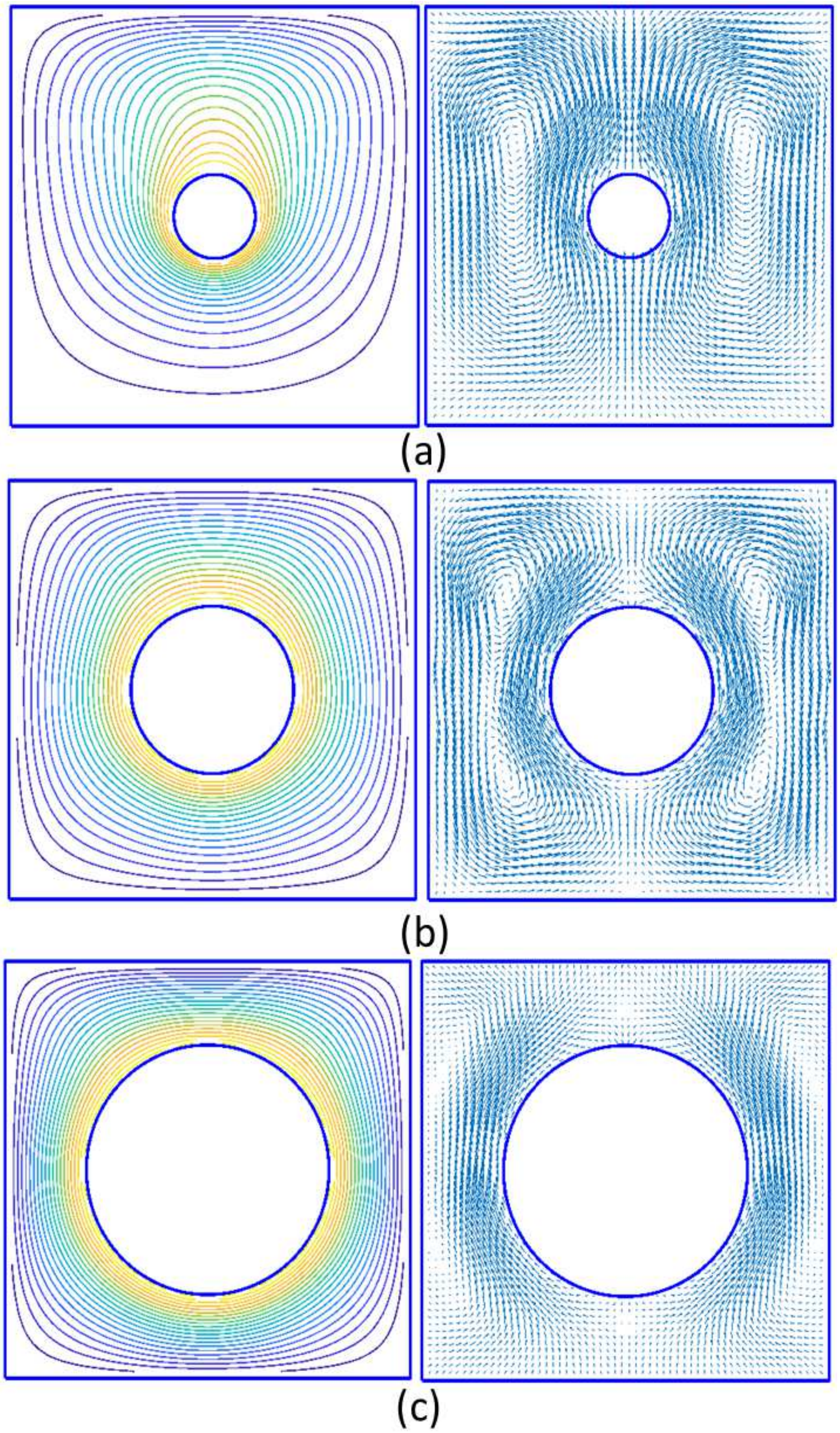


Figure 6.18: Example 4 (buoyancy flows in the square-circular annuli): three radii.

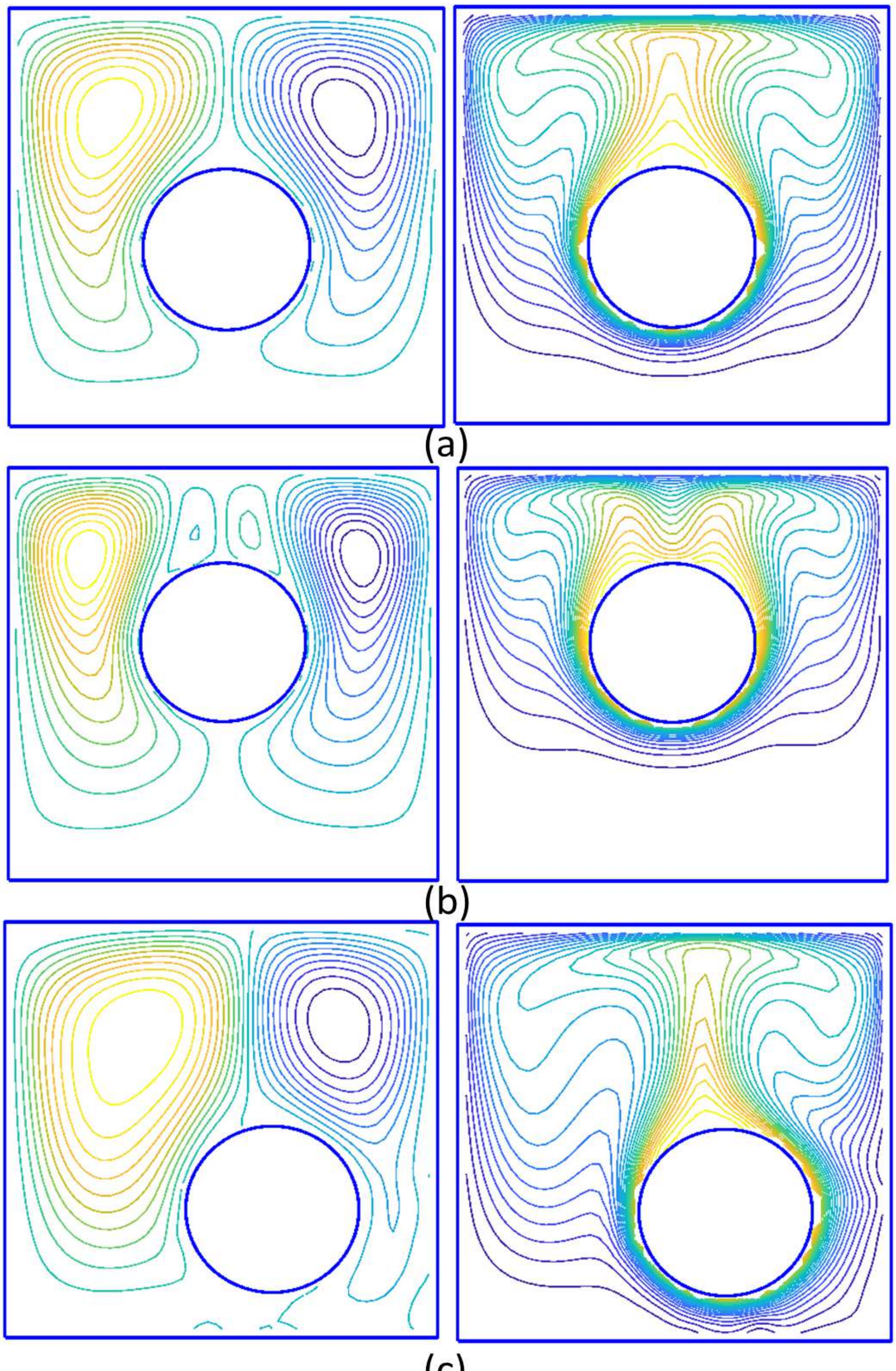


Figure 6.19: Example 4 (buoyancy flows in the square-circular annuli): eccentric.

Chapter 7

Conclusions

The aim of this PhD project was to further develop IRBF methods to produce improved RBF simulations of heat transfer and fluid flows. Its main contributions lie in the construction of new compact non-symmetric and symmetric IRBF stencils, where high degrees of solution accuracy and sparseness of the system matrix are achieved simultaneously. Below is a summary of the key achievements.

In Chapter 3, we have shown that the performance of compact integrated radial basis function (CIRBF) stencils over large values of the RBF width can be significantly improved with the use of extended precision, definite integrals, higher-order IRBFs and a minimum number of derivative equations. For the extended precision approach, accurate and stable solutions are achieved at the expense of higher computational costs and the need to use some specific computational tools such as function vpa in Matlab. For the other approaches, solution stability and accuracy are improved by reducing the size of the system matrices converting the RBF space into the physical space (the approaches based on definite integrals and a minimum number of derivative equations) and by integrating the MQ function four times instead of the usual twice (the approach based on higher-order IRBFs).

In Chapter 4, to produce symmetric and invertible interpolation matrices, we have introduced a compact symmetric IRBF stencil with Hermite interpolation for the numerical solution of ODEs/PDEs. Several schemes based on global and local approximations for rectangular and nonrectangular domains were presented. The extended precision approach is also utilised to extend the working range of the IRBF width for a given grid size, and a better accuracy is achieved. The local version is a preferred option for the handling of large-scale problems as it possesses several attractive features, including: (i) sparse system matrix; (ii) fast convergence rates (up to $O(h^{4.05})$) and (iii) the ability to also work with larger values of the RBF width with a relatively low computational cost.

In Chapter 5, we have developed a high-order approximation scheme based on IRBFs for time discretisations. The time stencil is based on two nodes over which: (i) IRBFs are employed to represent the field variable and (ii) the first nodal derivative value of the field variable is also included in the approximation. The proposed method, where both time and space terms are approximated using IRBFs, has been successfully applied to simulate shallow water flows in large domains and natural convection flows in multiply-connected domains. High levels of accuracy have been achieved using relatively large time steps. The results are comparable to those obtained by the differential quadrature method with respect to spatial discretisation and much more efficient (up to 1.42 times faster) than those by the finite difference method with respect to time discretisation.

In Chapter 6, we have presented a new domain embedding approach for the numerical simulation of complex-domain flows. In this method, the governing equations are modified to include the forcing terms and multiply-connected domains are transformed into a simply-connected domain that is simply represented by a fixed Cartesian grid. A new approach based on IRBFs and the governing equations to estimate the forcing term at grid nodes is proposed. The proposed technique has been verified successfully through several boundary-value and initial-value problems governed by the velocity-pressure formulation and the stream function-vorticity formulation in multi-connected domains. Numerical results have shown that a high convergence rate is achieved and the matrix condition number is relatively small.

Throughout this thesis, new local IRBF approximation-based schemes have been presented and their efficiency has been successfully validated in various test problems. However, the research in this thesis is limited to 2D problems and Newtonian inviscid and viscous fluids. The following works are suggested for possible further developments:

1. The proposed compact non-symmetric and symmetric IRBF stencils methods are presently developed for the simulation of 2D fluid flows. Extension of the methods to the 3D problems is straightforward. However, simulating fluid flow in 3D domains can thus result in high memory and time requirements. Since storage is proportional to the product of domain dimensions, the performance of simulation can be limited by memory capacity. To overcome this problem, one can utilise ADI schemes to decompose the 3D problems into 1D ones which save computer storage space. In addition, to achieve a higher level of accuracy with a relatively coarse grid very large values of the RBF width can be exploited. Here we suggest using extended precision (e.g. function VPA or variable precision arithmetic in MATLAB)-a straightforward way to handle ill-conditioned problems. It is noted that by defining a stencil on the unit length, one needs to compute the inversion of the conversion matrix only once and the result can be applied for any grid size to be employed.

2. The IRBF time stencil is introduced and presently formulated for the heat transfer and natural convection flows in Chapter 5. The method produces accurate results using a relatively large time step. Extension of this formulation to more complex fluids, i.e. shear thinning, viscoelastic fluids, is possible. For non-Newtonian fluid flows, the computational time

step is usually small and limited by the viscous of fluids. By discretising time derivatives using the IRBF time stencil, a larger time step may be employed which help to save the computational cost.

3. The proposed compact non-symmetric and symmetric IRBF stencils methods have limitation in solving 3D complex geometry and moving boundaries in time problems. Boundary points are generated by finding the intersection between $x-$ or $y-$grid lines and the geometry of boundaries. It requires generating a new Cartesian-grid at each time step due to moving boundaries. The problems can be solved by further developing the IRBF-NBFG approach proposed in Chapter 6 with a higher-order IRBF approximation on the forcing domains. In addition, an implementation of the proposed schemes in a parallel computing fashion would be desirable to increase the computational efficiency for large-scale problems.

References

Ahmad, I. and Khaliq, A. (2017). Local RBF method for multi-dimensional partial differential equations , *Computers and Mathematics with Applications* **74**(2): 292–324.

Ahrar, A. J. and Djavareshkian, M. H. (2017). Novel hybrid lattice Boltzmann technique with TVD characteristics for simulation of heat transfer and entropy generations of MHD and natural convection in a cavity, *Numerical Heat Transfer, Part B: Fundamentals* **72**(6): 431–449.

An, H., Ichikawa, Y., Tachikawa, Y. and Shiiba, M. (2011). A new iterative alternating direction implicit (iadi) algorithm for multi-dimensional saturated-unsaturated flow, *Journal of Hydrology* **408**(1-2): 127–139.

Ariathurai, R. and Krone, R. B. . (1976). Finite element model for cohesive sediment transport, *Journal of the Hydraulics Division* **102**(3): 323–338.

Bernal, F. and Kindelan, M. (2007). RBF meshless modeling of non-Newtonian Hele-Shaw flow, *Engineering Analysis with Boundary Elements* **31**(10): 863–874.

Bertoluzza, S., Ismail, M., Maury, B. Barth, T. J., Griebel, M., Keyes, D. E., Nieminen, R. M., Roose, D., Schlick, T., Barth, T. J., Griebel, M., Keyes, D. E., Nieminen, R. M., Roose, D., Schlick, T., Kornhuber, R., Hoppe, R., Paux, J., Pironneau, O.and Widlund, O. and Xu, J. E. (2005). The fat boundary method: Semi-discrete scheme and some numerical experiments, *Domain Decomposition Methods in Science and Engineering, Springer Berlin Heidelberg* **40**: 513–520.

Brunner, G., W. (2004). HEC-RAS reference manual, 3.1.2 ed., *Technical report*, US Army Corps of Engineers.

Buffat, M. and Le Penven, L. (2011). A spectral fictitious domain method with internal forcing for solving elliptic PDEs, *Journal of Computational Physics* **230**(7): 2433–2450.

Bustamante, C. A., Power, H., Sua, Y. H. and Florez, W. F. (2013). A global meshless collocation particular solution method (integrated Radial Basis Function) for two- dimensional Stokes flow problems, *Applied Mathematical Modelling* **37**(6): 4538–4547.

Canuto, C., Hussaini, M. Y., Quarteroni, A. and Zang, T. A. (2007). *Spectral methods: evolution to complex geometries and applications to fluid dynamics*, Springer Science and Business Media.

Carey, G. F. (1995). *Finite element modeling of environmental problems: surface and subsurface flow and transport*, Wiley.

Chen, C. S., Fan., C. M. and Wen., P. H. (2010). The method of approximated particular solutions for solving certain partial differential equations, *Numerical Methods for Partial Differential Equations* **28**(2): 506–522.

Chen, W., Ye, L. and Sun, H. (2010). Fractional diffusion equations by the Kansa method, *Computers and Mathematics with Applications* **59**(5): 1614–1620.

Chinchapatnam, P. P., Djidjeli, K. and Nair, P. B. (2007a). Domain decomposition for time-dependent problems using radial based meshless methods, *Numerical Methods for Partial Differential Equations* **23**: 38–59.

Chinchapatnam, P. P., Djidjeli, K. and Nair, P. B. (2007b). Radial basis function meshless method for the steady incompressible navierstokes equations, *International Journal of Computer Mathematics* **84**(10): 1509–1521.

Chinchapatnam, P. P., Djidjeli, K., Nair, P. B. and Tan, M. (2009). A compact RBF-FD based meshless method for the incompressible Navier-Stokes equations, *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment* **223**(3): 275–290.

Chorin, A. (1968). Numerical solution of the navier-stokes equations, *Mathematics of Computation* **22**(104): 745–762.

Coesnon, B., Heniche, M., Devals, C., Bertrand, F. and Tanguy, P. (2008). A fast and robust fictitious domain method for modelling viscous flows in complex mixers: The example of propellant make-down, *International journal for numerical methods in fluids* **58**(4): 427–449.

Collatz, L. (1966). *The Numerical Treatment of Differential Equations*, Springer-Verlag, Berlin.

Dai, W. and Nassar, R. (2002). Compact adi method for solving parabolic differential equations, *Numerical Methods for Partial Differential Equations* **18**(2): 129–142.

Dalrymple, R. A., Kirby, J. T. and Martin, P. A. (1994). Spectral methods for forward-propagating water waves in conformally-mapped channels, *Applied Ocean Research* **16**(5): 249–266.

Davis, G. D. V. (1983). Natural convection of air in a square cavity: a bench mark numerical solution, *International Journal for Numerical Methods in Fluids* **3**(3): 249–264.

De Vahl Davis, G. (1983). Natural convection of air in a square cavity: a bench mark numerical solution, *International Journal for Numerical Methods in Fluids* **3**(3): 249–264.

Dechaume, A., Finlay, W. and Minev, P. (2010). A two grid fictitious domain method for direct simulation of flows involving non interacting particles of a very small size, *International Journal for Numerical Methods in Fluid* **63**(11): 1241–1255.

Dehghan, M. and Abbaszadeh, M. (2017). The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations, *Journal of Computational Physics* **351**: 478–510.

Dehghan, M. and Abbaszadeh, M. (2018). An upwind local radial basis functions-differential quadrature (RBF-DQ) method with proper orthogonal decomposition (POD) approach for solving compressible Euler equation, *Engineering Analysis with Boundary Elements* **92**: 244–256.

Demirkaya, G., Soh, C. W. and Ilegbusi, O. J. (2008). Direct solution of navier-stokes equations by radial basis functions, *Applied Mathematical Modelling* **32**(9): 1848–1858.

Devendran, D. and Peskin, C. S. (2012). An immersed boundary energy-based method for incompressible viscoelasticity, *Journal of Computational Physics* **231**(14): 4613–4642.

DHI (2003). MIKE11 reference manual, *Technical report*, Danish Hydraulic Institute.

Ding, H., Shu, C., Yeo, K. S. and Lu, Z. L. (2005). Simulation of natural convection in eccentric annuli between a square outer cylinder and a circular inner cylinder using local MQ-DQ method, *Numerical Heat Transfer, Part A: Applications* **47**(3): 291–313.

Divo, E. and Kassab, A. J. (2005). A meshless method for conjugate heat transfer problems, *Engineering Analysis with Boundary Elements* **29**(2): 136 – 149.

Divo, E. and Kassab, A. J. (2006). Iterative domain decomposition meshless method modeling of incompressible viscous flows and conjugate heat transfer, *Engineering Analysis with Boundary Elements* **30**(6): 465–478.

Divo, E. and Kassab, A. J. (2007). An efficient localized radial basis function meshless method for fluid flow and conjugate heat transfer, *Journal of Heat Transfer* **129**(2): 124–136.

Divo, E. and Kassab, A. J. (2008). Localized meshless modeling of natural-convective viscous flows, *Numerical Heat Transfer, Part B: Fundamentals* **53**(6): 487–509.

Douglas, J. and Peaceman, D. . (1955). Numerical solution of two-dimensional heat flow problems, *American Institute of Chemical Engineering Journal* **1**: 505–512.

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

Duster, A., Parvizian, J., Yang, Z. and Rank, E. (2008). The finite cell method for three-dimensional problems of solid mechanics, *Computer Methods in Applied Mechanics and Engineering* **197**: 3768 – 3782.

Fadlun, E. A., Verzicco, R.and Orlandi, P. and Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *Journal of Computational Physics* **161**: 35–60.

Fan, C. M., Chien, C. S., Chan, H. F. and Chiu, C. L. (2013). The local rbf collocation method for solving the double-diffusive natural convection in fluid-saturated porous media, *International Journal of Heat and Mass Transfer* **57**(2): 500–503.

Fasshauer, G. (2007). *Meshfree Approximation Methods with Matlab*, Interdisciplinary Mathematical Sciences, World Scientific Publishers.

Fasshauer, G. E. (1997). *Solving partial differential equations by collocation with radial basis functions*, Vanderbilt University Press, Nashville.

Flyer, N., F. B. B. V. and Barnett, G. A. (2016). On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy, *Journal of Computational Physics* **321**: 21–38.

Flyer, N. and Wright, G. B. (2009). A radial basis function method for the shallow water equations on a sphere , *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 465 of *2106*.

Fornberg, B., L. E. and Flyer, N. (2011). Stable computations with Gaussian radial basis functions, *SIAM Journal Scientific Computing* **33**(2): 869–892.

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

Galland, J. C., Goutal, N. and Hervouet, J. M. (1991). TELEMAC: A new numerical model for solving shallow water equations, *Advances in Water Resources* **14**(3): 138–148.

Glakpe, E. K., Watkins Jr, C. B. and Cannon, J. N. (1986). Constant heat flux solutions for natural convection between concentric and eccentric horizontal cylinders, *Numerical Heat Transfer, Part A: Applications* **10**(3): 279–295.

Glowinski, R., Pan, T. W., Hesla, T. I. and Joseph, D. D. (1999). A distributed Lagrange multiplier/fictitious domain method for particulate flows, *International Journal of Multiphase Flow* **25**: 755–794.

Glowinski, R., Pan, T. W. and Periaux, J. (1998). Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, *Computer Methods in Applied Mechanics and Engineering* **151**: 181–194.

Glowinski, R., Pan, T. W. and Priaux, J. (1994). A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations., *Computer Methods in Applied Mechanics and Engineering* **112**: 133–148.

Haji Mohammadi, M., Sotiropoulos, F. and Brinkerhoff, J. (2019). Moving least squares reconstruction for sharp interface immersed boundary methods, *International Journal for Numerical Methods in Fluids* **90**(2): 57–80.

Hardy, R. L. (1975). Research results in the application of multiquadric equations to surveying and mapping problems, *Survg Mapp* **35**: 321–332.

Haykin, S. (1999). Multilayer perceptrons, *Neural Networks: A Comprehensive Foundation* **2**: 156–255.

Hejazi, K., Sami, S., Soltanpour, M. and Samsami, F. (2014). Numerical modeling of irregular wave on mud layer using spectral method, *Coastal Engineering Proceedings* **1**(34): 26.

Hejranfar, K. and Khajeh-Saeed, A. (2011). Implementing a high-order accurate implicit operator scheme for solving steady incompressible viscous flows using artificial compressibility method, *International Journal for Numerical Methods in Fluids* **66**(8): 939–962.

Ho-Minh, D., Mai-Duy, N. and Tran-Cong, T. (2009). A Galerkin-RBF approach for the streamfunction-vorticity-temperature formulation of natural convection in 2D enclosured domains, *CMES: Computer Modeling in Engineering & Sciences* **44**(3): 219–248.

Hon, Y. C., Cheung, K. F., Mao, X. Z. and Kansa, E. J. (1999). Multiquadric solution for shallow water equations, *Journal of Hydraulic Engineering* **125**(5): 524–533.

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

Huang, C.-S., Yen, H.-D. and Cheng, A. H.-D. (2010). On the increasingly flat radial basis function and optimal shape parameter for the solution of elliptic PDEs, *Engineering Analysis with Boundary Elements* **34**(9): 802–809.

Huang, L., Ng, C. O. and Chwang, A. T. (2006). A FourierChebyshev collocation method for the mass transport in a layer of power-law fluid mud, *Computer methods in applied mechanics and engineering* **195**(9): 1136–1153.

Husain, S. Z., Floryan, J. M. and Szumbarski, J. (2009). Over-determined formulation of the immersed boundary conditions method, *Computer Methods in Applied Mechanics and Engineering* **199**(1-4): 94 – 112.

Issa, R. I. (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting, *Journal of Computational Physics* **62**: 40–65.

Jin, L. and Shen, H. (2016). Projection-and characteristic-based operator-splitting simulation of mixed convection flow coupling heat transfer and fluid flow in a lid-driven square cavity, *Numerical Heat Transfer, Part B: Fundamentals* **70**(4): 354–371.

Kang, S. and Suh, Y. (2011). An immersed boundary finite volume method for direct simulation of flows with suspended paramagnetic particles, *International Journal for Numerical Methods in Fluids* **67**(1): 58–73.

Kansa, E. J. (1990). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - II. Solutions to parabolic, hyperbolic and elliptic partial differential equations, *Computers and Mathematics with Application* **19**(8-9): 147–161.

Kansa, E. J., Power, H., Fasshauer, G. E. and Ling, L. (2004). A volumetric integral radial basis function method for time-dependent partial differential equations: I. Formulation, *Engineering Analysis with Boundary Elements* **28**: 1191–1206.

Khattak, A. and Tirmizi, I. (2008). A meshfree method for numerical solution of KdV equation, *Engineering Analysis with Boundary Elements* **32**(10): 849–855.

Kim, J., Kim, D. and Choi, H. (2001). An immersed-boundary finite-volume method for simulations of flow in complex geometries, *ournal of Computational Physics* **171**: 132–150.

Kim, Y., Jun, S. and Lee, J. H. (2007). Meshfree point collocation method for the stream-vorticity formulation of 2D incompressible Navier-Stokes equations, *Computer methods in applied Mechanics and Engineering* **196**(33-34): 3095–3109.

Kosec, G. and Šarler, B. (2008a). Local RBF collocation method for Darcy flow, *CMES: Computer Modeling in Engineering & Sciences* **25**(3): 197–207.

Kosec, G. and Šarler, B. (2008b). Solution of thermo-fluid problems by collocation with local pressure correction, *International Journal of Numerical Methods for Heat & Fluid Flow* **18**(7-8): 868–882.

Kosec, G. and Šarler, B. (2009). Solution of phase change problems by collocation with local pressure correction, *CMES: Computer Modeling in Engineering & Sciences* **47**(2): 191–216.

Kuehn, T. H. and Goldstein, R. J. (1976). An experimental and theoretical study of natural convection in the annulus between horizontal concentric cylinders, *Journal of Fluid Mechanics* **74**(4): 695–719.

Larsson, E. and Fornberg, B. (2003). A numerical study of some radial basis function based solution methods for elliptic PDEs, *Computers and Mathematics with Applications* **46**: 891–902.

Le-Cao, K., Mai-Duy, N., Tran, C.-D. and Tran-Cong, T. (2010). Towards the analysis of shear suspension flows using radial basis functions, *CMES: Computer Modeling in Engineering & Sciences* **67**(3): 265–294.

Le-Cao, K., Mai-Duy, N., Tran, C.-D. and Tran-Cong, T. (2011). Numerical study of stream-function formulation governing flows in multiply-connected domains by integrated RBFs and Cartesian grids, *Computers & Fluids Journal* **44**: 32–42.

Le-Cao, K., Mai-Duy, N. and Tran-Cong, T. (2009). An effective integrated-RBFN Cartesian-grid discretization for the stream function vorticity temperature formulation in nonrectangular domains, *Numerical Heat Transfer, Part B: Fundamentals* **55**(6): 480–502.

Le Quere, P. (1991). Accurate solutions to the square thermally driven cavity at high Rayleigh number, *Computers and Fluids* **20**(1): 29–41.

Le, T. T. V., Mai-Duy, N., Le-Cao, K. and Tran-Cong, T. (2018). A time discretization scheme based on integrated radial basis functions for heat transfer and fluid flow problems, *Numerical Heat Transfer, Part B: Fundamentals* **74**(2): 498–518.

Lehto, E., Shankar, V. and Wright, G. (2017). A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces, *SIAM Journal on Scientific Computing* **39**(5): A2129–A2151.

Lewis, E. (1979). Steady flow between a rotating circular cylinder and fixed square cylinder, *Journal of Fluid Mechanics* **95**: 497–513.

Li, J. and 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**(8): 1085–1094.

Li, J. and Hon, Y. C. (2004). Domain decomposition for radial basis meshless methods, *Numerical Methods for Partial Differential Equations* **20**: 450–462.

Li, K., Huang, Q. B., Wang, J. L. and Lin, L. G. (2011). An improved localized radial basis function meshless method for computational aeroacoustics, *Engineering Analysis with Boundary Elements* **35**(1): 47 – 55.

Ling, L. and Trummer, M. R. (2004). Multiquadric collocation method with integral formulation for boundary layer problems, *Computers and Mathematics with Applications* **48**(5-6): 927–941.

Liu, G. R. (2009). *Meshfree methods: moving beyond the finite element method*, Taylor and Francis.

Liu, G. R. and Gu, Y. T. (2005). *An introduction to meshfree methods and their programming*, Springer Science & Business Media.

Liu, G. R., Yan, L., Wang, J. G. and Gu, Y. T. (2002). Point interpolation method based on local residual formulation using radial basis functions, *Structural Engineering and Mechanics* **14**(6): 713–732.

Liu, H., Ding, Y., Li, M., Lin, P., Yu, M. H. and Shu, A. P. (2015). A Hybrid Lattice Boltzmann Method Finite Difference Method Model for Sediment Transport and Riverbed Deformation, *River Research and Applications* **31**(4): 447–456.

Madych, W. R. (1992). Miscellaneous error bounds for multiquadric and related interpolators, *Computers and Biomedical Research Mathematics with Applications* **24**(12): 121–138.

Madych, W. R. and Nelson, S. A. (1990). Multivariate interpolation and conditionally positive definite functions, *II. Mathematics of Computation* **54**(189): 211–230.

Mai-Duy, N., Dalal, D., Le, T., Ngo-Cong, D. and Tran-Cong, T. (2018). A symmetric integrated radial basis function method for solving differential equations, *Numerical Methods for Partial Differential Equations* **34**(3): 959–981.

Mai-Duy, N., Le-Cao, K. and Tran-Cong, T. (2008). A Cartesian grid technique based on one-dimensional integrated radial basis function networks for natural convection in conccentric annuli , *International Journal for Numerical Methods in fluids* **57**: 1709–1730.

Mai-Duy, N., Le, T. T. V., Tien, C. M. T., Ngo-Cong, D. and Tran-Cong, T. (2017). Compact approximation stencils based on integrated flat radial basis functions, *Engineering Analysis with Boundary Elements* **74**: 79–87.

Mai-Duy, N. and Tran-Cong, T. (2001). Numerical solution of differential equations using multiquadric radial basis function networks, *Neural Networks* **14**(2): 185–199.

Mai-Duy, N. and Tran-Cong, T. (2003). Approximation of function and its derivatives using radial basis function networks, *Applied Mathematical Modelling* **27**(3): 197–220.

Mai-Duy, N. and Tran-Cong, T. (2007). A Cartesian-grid collocation method based on radial basis function networks for solving PDEs in irregular domains, *Numerical Methods for Partial Differential Equations* **23**(5): 1192–1210.

Mai-Duy, N. and Tran-Cong, T. (2008). A multidomain integrated-radial-basis-function collocation method for elliptic problems, *Numerical Methods for Partial Differential Equations* **24**(5): 1301–1320.

Mai-Duy, N. and Tran-Cong, T. (2009). An integrated-RBF technique based on Galerkin formulation for elliptic differential equations, *Engineering Analysis with Boundary Elements* **33**(2): 191–199.

Mai-Duy, N. and Tran-Cong, T. (2011). Compact local integrated-RBF approximations for second-order elliptic differential problems, *Journal of Computational Physics* **230**(12): 4772–4794.

Mai-Duy, N. and Tran-Cong, T. (2013). A compact five-point stencil based on integrated RBFs for 2D second-order differential problems, *Journal of Computational Physics* **235**: 302–321.

Manzari, M. T. (1999). An explicit finite element algorithm for convection heat transfer problems, *International Journal Numerical Methods for Heat and Fluid Flow* **9**(8): 860–877.

Maury, B. (2001). A fat boundary method for the Poisson problem in a domain with holes, *Journal of Scientific Computing* **16**: 319–339.

Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constructive Approximation* **2**: 11–22.

Mo, H., Lien, F., Zhang, F. and Cronin, D. (2018). An immersed boundary method for solving compressible flow with arbitrarily irregular and moving geometry, *International Journal for Numerical Methods in Fluids* **88**(5): 239–263.

Moukalled, F. and Acharya, S. (1996). Natural convection in the annulus between concentric horizontal circular and square cylinders, *Journal of Thermophysics and Heat Transfer* **10**(3): 524–531.

Mramor, K., Vertnik, R. and arler, B. (2013). Simulation of natural convection influenced by magnetic field with explicit local radial basis function collocation method, *CMES: Computer Modeling in Engineering & Sciences* **92**(4): 327–352.

Navarro, H. A., Cabezas-Gomez, L., da Silva, R. C. and Montagnoli, A. N. (2007). A generalized alternating-direction implicit scheme for incompressible magnetohydrodynamic viscous flows at low magnetic reynolds number, *Applied Mathematics and Computation* **189**(2): 1601–1613.

Ostrach, S. (1988). Natural convection in enclosures, *Journal of Heat Transfer* **110**: 1175–1190.

Park, J. and Sandberg, I. W. (1991). Universal approximation using radial basis function networks, *Neural Computation* **3**: 246–257.

Park, J. and Sandberg, I. W. (1993). Approximation and radial basis function networks, *Neural Computation* **5**: 305–316.

Parvizian, J., Duster, A. and Rank, E. (2007). Finite cell method h - and p -extension for embedded domain problems in solid mechanics, *Computational Mechanics* **41**: 121–133.

Patankar, N. A., Singh, P., Joseph, D. D., Glowinski, R. and Pan, T. W. (2000). A new formulation of the distributed lagrange multiplier/ fictitious domain method for particulate flows, *International Journal Multiphase Flow* **26**: 1509–1524.

Patankar, S. V. and Spalding, D. B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *International Journal of Heat and Mass Transfer* **15**: 1787–1806.

Peskin, C. S. (1972). Flow patterns around heart valves: A numerical method, *Journal of Computational Physics* **10**: 252–271.

Poggio, T. and Girosi, F. (1990). Networks for approximation and learning, *Proceedings of the IEEE* **78**: 1481–1497.

Pourbashash, H. and Oshagh, M. (2018). Local RBF-FD technique for solving the two-dimensional modified anomalous sub-diffusion equation, *Applied Mathematics and Computation,* **339**: 144–152.

Power, H. and Barraco, H. (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.

Power, H., Hernandez, A. and Rocca, A. L. (2007). Non-overlapping domain decomposition scheme for the symmetric radial basis function meshless approach with double collocation at the sub-domain interfaces, *WIT Transactions on Modelling and Simulation* **44**: 13–22.

Rashidinia, J., Fasshauer, G. E. and Khasi, M. (2016). A stable method for the evaluation of Gaussian radial basis function solutions of interpolation and collocation problems, *Computers and Mathematics with Applications* **72**(1): 178–193.

Rasulov, M., Aslan, Z. and Pakdil, O. (2005). Finite differences method for shallow water equations in a class of discontinuous functions, *Applied mathematics and computation* **160**(2): 343–353.

Roque, C. M. C., Ferreira, A. J. M. and Jorge, R. M. N. (2010). An optimized shape parameter radial basis function formulation for composite and sandwich plates using higher order formulations, *Journal of Sandwich Structures and Materials* **12**(3): 279–306.

Sammouda, H., Belghith, A. and Surry, C. (1999). Finite element simulation of transient natural convection of low-Prandtl-number fluids in heated cavity, *International Journal of Numerical Methods for Heat and Fluid Flow* **9**(5): 612–624.

Šarler, B. (2005). A radial basis function collocation approach in computational fluid dynamics, *CMES: Computer Modeling in Engineering and Sciences* **7**(2): 185–194.

Šarler, B. (2009). Solution of potential flow problems by the modified method of fundamental solutions: Formulations with the single layer and the double layer fundamental solutions, *Engineering Analysis with Boundary Elements* **33**(12): 1374–1382.

Šarler, B., Jelic, N., Kovacevic, I., Lakner, M. and Perko, J. (2006). Axisymmetric multiquadrics, *Engineering Analysis with Boundary Elements* **30**(2): 137 – 142.

Šarler, B., Kosec, G., Lorbicka, A. and Vertnik, R. (2010). A meshless approach in solution of multiscale solidification modeling, *Materials Science Forum* **649**: 211–216.

Šarler, B., Perko, J. and Chen, C. S. (2004). Radial basis function collocation method solution of natural convection in porous media, *International Journal of Numerical Methods for Heat and Fluid Flow* **14**(2): 187–212.

Šarler, B. and Vertnik, R. (2006). Meshfree explicit local radial basis function collocation method for diffusion problems, *Computers and Mathematics with Applications* **51**(8): 1269–1282.

Sarra, S. A. (2006). Integrated multiquadric radial basis function approximation methods, *Computers & Mathematics with Applications* **51**(8): 1283–1296.

Shankar, V., Wright, G., Fogelson, A. and Kirby, R. (2014). A radial basis function (RBF) finite difference method for the simulation of reaction-diffusion equations on stationary platelets within the augmented forcing method, *International Journal for Numerical Methods in Fluids* **75**(1): 1–22.

Shi, L., Pan, T. and Glowinski, R. (2012). Numerical simulation of lateral migration of red blood cells in Poiseuille flows, *International Journal for Numerical Methods in Fluids* **68**(11): 1393–408.

Shi, X., Lin, G., Zou, J. and Fedosov, D. (2013). A lattice Boltzmann fictitious domain method for modelling red blood cell deformation and multiple-cell hydrodynamic interactions in flow, *International Journal for Numerical Methods in Fluids* **72**(8): 895–911.

Shu, C. (1999). Application of differential quadrature method to simulate natural convection in a concentric annulus, *International Journal for Numerical Methods in Fluids* **30**(8): 977–993.

Shu, C., Ding, H. and Yeo, K. S. (2003). Local radial basis funcion-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering* **192**(7-8): 941–954.

Shu, C., Ding, H. and Yeo, K. S. (2005). Computation of incompressible navier-stokes equations by local rbf-based differential quadrature method, *CMES: Computer Modeling in Engineering & Sciences* **7**(2): 195–206.

Shu, C. and Wu, Y. L. (2007). Integrated radial basis functions-based differential quadrature method and its performance, *International Journal for Numerical Methods in Fluids* **53**(6): 969–984.

Shu, C. and Zhu, Y. D. (2002). Efficient computation of natural convection in a concentric annulus between an outer square cylinder and an inner circular cylinder, *International Journal for Numerical Methods in Fluids* **38**: 429–445.

Silberman, I. (1954). Planetary waves in the atmosphere, *Journal of Meteorology* **11**: 27–43.

Singh, S. and You, D. (2011). A multi-block adi finite-volume method for incompressible navierstokes equations in complex geometries, *Journal of Computational Physics* **230**(19): 7400–7417.

Skouras, E., Bourantas, G., Loukopoulos, V. and Nikiforidis, G. (2011). Truly meshless localized type techniques for the steady-state heat conduction problems for isotropic and functionally graded materials, *Engineering Analysis with Boundary Elements* **35**(3): 452–464.

Sun, C. P., Young, D. L., Shen, L. H., Chen, T. F. and Hsian, C. C. (2013). Application of localized meshless methods to 2D shallow water equation problems, *Engineering Analysis with Boundary Elements* **37**(11): 1339–1350.

Sun, X. (1994). Scattered Hermite interpolation using radial basis functions, *Linear Algebra and its Applications* **207**: 135–146.

Swarztrauber, P. N. (1996). Spectral transform methods for solving the shallow-water equations on the sphere, *Monthly Weather Review* **124**(4): 730–744.

Thai-Quang, N., Le-Cao, K., Mai-Duy, N., Tran, C. D. and Tran-Cong, T. (2013). A numerical scheme based on compact integrated-RBFs and Adams-

Bashforth/CrankNicolson algorithms for diffusion and unsteady fluid flow problems, *Engineering Analysis with Boundary Elements* **37**(12): 1653–1667.

Thai-Quang, N., Le-Cao, K., Mai-Duy, N. and Tran-Cong, T. (2012). A high-order compact local integrated-RBF scheme for steady-state incompressible viscous flows in the primitive variables, *CMES: Computer Modeling in Engineering & Sciences* **84**(6): 528–557.

ThaiQuang, N., Mai-Duy, N., Tran, C. D. and Tran-Cong, T. (2012). High-order alternating direction implicit method based on compact integrated-rbf approximations for unsteady/steady convection-diffusion equations., *CMES: Computer Modeling in Engineering & Sciences* **89**(3): 189–220.

Tian, Z., Liang, X. and Yu, P. (2011). A higher order compact finite difference algorithm for solving the incompressible navierstokes equations., *International Journal for Numerical Methods in Engineering* **88**(6): 511–532.

Tien, C. M. T., Thai-Quang, N., Mai-Duy, N., Tran, C.-D. and Tran-Cong, T. (2015). A three-point coupled compact integrated RBF scheme for second-order differential problems, *CMES: Computer Modeling in Engineering and Sciences* **104**(6): 425–469.

Tolstykh, A. I. and Shirobokov, D. A. (2003). On using radial basis functions in a finite difference mode with applications to elasticity problems, *Computational Mechanics* **33**(1): 68–79.

Tolstykh, A. I. and Shirobokov, D. A. (2005). Using radial basis functions in a finite difference mode, *CMES: Computer Modeling in Engineering & Sciences* **7**(2): 207–222.

Toro, E. F. (2001). *Shock-capturing methods for free-surface shallow flows*, Wiley.

Tsai, Y. M., Kuo, H. C., Chang, Y. C. and Tseng, Y. H. (2012). A new parallel domain-decomposed Chebyshev collocation method for atmospheric, *Terrestrial, Atmospheric and Oceanic Sciences* **23**(4).

Uhlmann, M. (2005). An immersed boundary method with direct forcing for the simulation of particulate flows, *Journal of Computational Physics* **209**: 448 – 476.

Van Loon, R., Anderson, P., De Hart, J. and Baaijens, F. (2004). A combined fictitious domain/adaptive meshing method for fluid-structure interaction in heart valves, *International Journal for Numerical Methods in Fluids* **46**(5): 533–544.

Vertnik, R. and Šarler, B. (2006). Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems, *International Journal of Numerical Methods for Heat and Fluid Flow* **16**(5): 617–640.

Vertnik, R., Založnik, M. and Šarler, B. (2006). Solution of transient direct-chill aluminium billet casting problem with simultaneous material and interphase moving boundaries by a meshless method, *Engineering Analysis with Boundary Elements* **30**(10): 847–855.

Vos, P., van Loon, R. and Sherwin, S. (2008). A comparison of fictitious domain methods appropriate for spectral/hp element discretisations, *Computer Methods in Applied Mechanics and Engineering* **197**: 2275 – 2289.

Wan, D. and Turek, S. (2006). Direct numerical simulation of particulate flow via multigrid fem techniques and the fictitious boundary method, *International Journal of Numerical Method Fluids* **51**: 531–566.

Wang, J. G. and Liu, G. R. (2002). A point interpolation meshless method based on radial basis functions, *International Journal for Numerical Methods in Engineering* **54**(11): 1623–1648.

Wang, Y., Shu, C., Teo, C. J. and Yang, L. M. (2016). A fractional-step lattice Boltzmann flux solver for axisymmetric thermal flows, *Numerical Heat Transfer, Part B: Fundamentals* **69**(2): 111–129.

Wang, Y., Shu, C., Yang, L. and Sun, Y., . (2017). On the immersed boundary lattice Boltzmann simulations of incompressible flows with freely moving objects, *International Journal for Numerical Methods in Fluids* **83**(4): 331–350.

Wang, Z. H., Huang, Z., Zhang, W. and Xi, G. (2015a). A meshless local radial basis function method for two-dimensional incompressible navier-stokes equations, *Numerical Heat Transfer, Part B: Fundamentals* **67**(4): 320–337.

Wang, Z., Huang, Z., Zhang, W. and Xi, G. (2015b). A multidomain chebyshev pseudo-spectral method for fluid flow and heat transfer from square cylinders, *Numerical Heat Transfer, Part B: Fundamentals* **68**(3): 224–238.

Waters, J. and Pepper, D. W. (2015). Global versus localized RBF meshless methods for solving incompressible fluid flow with heat transfer, *Numerical Heat Transfer, Part B: Fundamentals* **68**(3): 185–203.

Wong, S. M., Hon, Y. C. and Golberg, M. A. (2002). Compactly supported radial basis functions for shallow water equations, *Applied Mathematics and Computation* **127**(1): 79–101.

Wright, G. B. and Fornberg, B. (2006). Scattered node compact finite difference-type formulas generated from radial basis functions, *Journal of Computational Physics* **212**(1): 99–123.

Wu, W., Rodi, W. and Wenka, T. (2000). 3D numerical modeling of flow and sediment transport in open channels, *Journal of Hydraulic Engineering* **126**(1): 4–15.

Wu, Y. L. and Liu, G. R. (2003). A meshfree formulation of local radial point interpolation method (LRPIM) for incompressible flow simulation, *Computational Mechanics* **30**(5-6): 355–365.

Wu, Z. (1992). Hermite-Birkhoff interpolation of scattered data by radial basis functions, *Approximation Theory and its Applications* **8**(2): 1–10.

Xiao, D., Fang, F., Pain, C. and Hu, G. (2015). Non-intrusive reduced order modelling of the Navier-Stokes equations based on RBF interpolation, *International Journal for Numerical Methods in Fluids* **79**(11): 580–595.

You, D. (2006). A high-order pade adi method for unsteady convection-diffusion equations, *Journal of Computational Physics* **214**: 1–11.

Young, D. L. (1991). Finite element modeling of shallow water wave equations, *Journal of the Chinese Institute of Engineers* **14**(2): 143–155.

Young, D. L., Chen, C. S. and Wong, T. K. (2005). Solution of Maxwell's equations using the MQ method, *Computers, Materials & Continua* **2**: 267–76.

Young, D. L., Jane, S. C., Lin, C. Y., Chiu, C. L. and Chen, K. C. (2004). Solutions of 2D and 3D Stokes laws using multiquadrics method, *Engineering Analysis with Boundary Elements* **28**(10): 1233–1243.

Yu, Z. A. (2005). DLM/FD method for fluid / flexible-body interactions, *ournal of Computational Physics* **207**: 1–27.

Yun-Xin, Z. and Yong-Ji, T. (2006). Meshless schemes for unsteady Navier-Stokes equations in vorticity formulation using radial basis functions, *Journal of Computational and Applied Mathematics* **192**(2): 328–338.

Yusof, J. M. (1997). Combined Immersed Boundaries/B-Splines Methods for Simulations of Flows in Complex Geometries, *Annual Research Briefs, NASA Ames/Stanford University* .

Zahab, Z. E., Divo, E. and Kassab, A. J. (2009). A localized collocation meshless method (LCMM) for incompressible flows CFD modeling with applications to transient hemodynamics, *Engineering Analysis with Boundary Elements* **33**(8-9): 1045 – 1061.

Zerroukat, M., Power, H. and 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**: 1263–1728.

Zhang, M., Xu, Y., Hao, Z. and Qiao, Y. (2014). Integrating 1D and 2D hydrodynamic, sediment transport model for dam-break flow using finite volume method, *Science China Physics, Mechanics and Astronomy* **57**(4): 774–783.

Zhao, Z., Li, M., He, L., Shao, S. and Zhang, L. (2019). High-order curvilinear mesh generation technique based on an improved radius basic function approach, *International Journal for Numerical Methods in Fluids* **91**(3): 97–111.