UNIVERSITY OF SOUTHERN QUEENSLAND

MESHLESS RADIAL BASIS FUNCTION METHOD FOR UNSTEADY INCOMPRESSIBLE VISCOUS FLOWS

A dissertation submitted by

Lan Mai-Cao

B.Eng., HoChiMinh City University of Technology, Vietnam, 1991

M.Eng., Royal Melbourne Institute of Technology (RMIT), Australia, 1998

For the award of the degree of

Doctor of Philosophy

December 2008 (emended April 2009)

Dedication

To the memory of my grandparents & my father,
and to my mother, my wife & sons.

Notes to Readers

To facilitate the reading of this thesis, a number of files are included on the attached CD to provide colour presentation as well as animation of some numerical results in this thesis. The contents of the CD include:

1. thesis.pdf: An electronic version of this thesis with colour figures;
2. chap4-test4-1b.avi: An animation showing the numerical simulation of a bubble that moves and deforms in a shear flow (Test 4, Chapter 4);
3. chap4-test5-4b.avi: An animation showing the numerical simulation of four bubbles that move, deform and merge together in a shear flow (Test 5, Chapter 4);
4. chap5-test3-vc.avi: An animation showing the evolution of the velocity along the mid-vertical and horizontal lines in a lid-driven cavity flow (Test 3, Chapter 5);
5. chap5-test3-stream.avi: An animation showing the evolution of the stream-function in a lid-driven cavity flow (Test 3, Chapter 5);
6. chap5-test3-vort.avi: An animation showing the evolution of the vorticity in a lid-driven cavity flow (Test 3, Chapter 5);
7. chap6-test1-br.avi: An animation showing the numerical simulation of two bubbles rising upward in an interfacial flow (Test 1, Chapter 6).

Abstract

This thesis reports the development of new meshless schemes for solving time-dependent partial differential equations (PDEs) and for the numerical simulation of some typical unsteady incompressible viscous flows.

The new numerical schemes are based on the Indirect/Integrated Radial Basis Function Network (IRBFN) method which is fully meshless as no element-type mesh is required. The IRBFN method has been successfully applied to solve time-independent elliptic PDEs, some steady fluid flows and recently unsteady Navier-Stokes equations in streamfunction-vorticiy formulation using simple time integration methods (e.g. first-order backward Euler method). The main objective of the present research is to devise and implement meshless numerical schemes for unsteady problems in computational fluid dynamics where not only the accuracy but also the efficiency and stability of the numerical schemes are of primary concerns. In addition, the effects of different parameters of the IRBFN method on the accuracy, stability and efficiency of the proposed numerical schemes are extensively studied in this research.

As the first step in extending the IRBFN method to various types of time-dependent PDEs, two numerical schemes combining the IRBFN method with high-order time stepping algorithms are developed for solving parabolic, hyperbolic, and advection-diffusion equations. Sensitivity analysis of the method to point density, time-step size and shape parameter are extensively performed to study the influence of these parameters to the overall accuracy of the method.

A further extension of the IRBFN method for incompressible fluid flows with moving interfaces, especially for passive transport problems is accomplished in this research with a novel meshless approach in which the level set method is coupled with the the IRBFN method for capturing moving interfaces in an ambient fluid flow without any explicit computation of the actual front location.

Another contribution of this research is the development of two new meshless schemes based on the IRBFN method for the numerical simulation of unsteady incompressible viscous flows governed by the Navier-Stokes equations. In the new schemes, the splitting approach is used to deal with the momentum equation and the incompressibility constraint in a segregated manner. Numerical experiments on the new schemes in terms of accuracy and stability are performed for verification purposes.

Finally, a novel meshless hybrid scheme is developed in this research to numerically simulate interfacial flows in which the motion and deformation of the interface between the two immiscible fluids are fully captured. Unlike the passive transport problems mentioned above where the influence of the moving interface on the surrounding fluid is ignored, the interfacial flows are studied here with the surface tension taken into account. As a result, a two-way interaction between the moving interface and the ambient flow is fully investigated.

All numerical schemes developed in this research are verified through a wide range of transient problems including different kinds of time-dependent PDEs, typical passive transport problems and interfacial flows as well as unsteady incompressible viscous flows governed by Navier-Stokes equations.

Certification of Dissertation

I certify that the idea, experimental work, results and analyses, software and conclusions reported in this dissertation are entirely my own effort, except where otherwise acknowledged. I also certify that the work is original and has not been previously submitted for any other award.

Lan Mai-Cao, Candidate	Date

ENDORSEMENT

Prof. Thanh Tran-Cong, Principal supervisor	Date
Dr. Ruth Mossad, Associate supervisor	Date

Acknowledgments

I would like to express my deepest gratitude and appreciation to Professor Thanh Tran-Cong for his guidance and encouragement throughout this research. Without his continuous support this thesis would not have been possible.

I am very grateful to Ms. Ruth Hilton and Ms. Christine Bartlett formerly from the Office of Research and Higher Degrees for their invaluable help during my research. I would like to express my sincere thanks to Associate Professor David Buttsworth from the Faculty of Engineering and Surveying (FoES) for his kind supports. I wish to extend my thanks to Dr. Ruth Mossad for acting her role as the associate supervisor. Many thanks go to my colleagues and friends at FoES for their friendship and fruitful discussions.

I gratefully acknowledge the financial support from Moldflow Pty Ltd, the Postgraduate Scholarship from the University of Southern Queensland and the financial aids from the Faculty of Engineering and Surveying in the form of a Scholarship Supplementation.

Finally, special thanks go to my mother and my wife for their patience and supports during my study.

Papers Resulting from the Research

Book Chapters

1. N. MAI-DUY, L. MAI-CAO and T. TRAN-CONG (2006) "A new meshless RBF-based method for unsteady fluid flow analysis", Advances in Meshless Methods, Chapter XI, J. Sladek and V. Sladek (Eds), Tech Science Press, ISBN: 0-9717880-2-2, pp241-262.

Journal Papers

1. L.MAI-CAO and T.TRAN-CONG (2008) "A Meshless Approach to Capturing Moving Interfaces in Passive Transport Problems", CMES: Computer Modeling in Engineering and Sciences, 31(3), 157-188.
2. N. MAI-DUY, L. MAI-CAO and T. TRAN-CONG (2007) "Computation of transient viscous flows using indirect radial basis function networks". CMES: Computer Modeling in Engineering and Sciences, 18(1), 59-77.
3. L. MAI-CAO and T. TRAN-CONG (2005) "A meshless IRBFN-based method for transient problems", CMES: Computer Modeling in Engineering and Sciences, 7(2), 149-171.

4. L. MAI-CAO and T. TRAN-CONG (2008) "Meshless projection schemes for unsteady incompressible Navier-Stokes equations", in preparation.

5. L. MAI-CAO and T. TRAN-CONG (2008) "Meshless approach to interfacial flows", in preparation.

Conference Papers

1. L. MAI-CAO and T. TRAN-CONG (2006) "A meshless level-set scheme for interfacial flows", in Nguyen Quoc Son and Nguyen Dung (Eds), International Conference on Non-linear Analysis & Engineering Mechanics Today, Ho Chi Minh City, Vietnam, 11-14 December 2006, CD paper No 26.

2. L. MAI-CAO and T. TRAN-CONG (2005) "An IRBFN-based scheme for moving interfaces in an incompressible viscous flow", Keynote Lecture, in "Advances in Computational & Experimental Engineering and Sciences", SM Sivakumar, A Meher Prasad, B Dattaguru, S Narayanan, AM Rajendrean, SN Atluri (Eds), pp473-479.

3. L. MAI-CAO and T. TRAN-CONG (2004) "Element-Free Simulation for Non-Newtonian Flows", Keynote Lecture in Advances in Computational and Experimental Engineering and Sciences, S.N. Atluri and A.J.B. Tadeu (Ed), Tech Science Press, pp1332-1338.

4. L. MAI-CAO and T. TRAN-CONG (2003) "A meshless level set approach to interface capturing", Keynote Lecture, International Conference on Computational & Experimental Engineering and Sciences, July 24-29, Corfu, Greece, CD Chapter 17.

5. L. MAI-CAO and T. TRAN-CONG (2003) "Solving time-dependent PDEs with a meshless IRBFN-based method", Invited paper, International Workshop on Meshfree Methods 2003, July 21-23, Lisbon, Portugal, pp119-124.

Contents

Acronyms & Abbreviations

AB-CN	Adams Bashforth-Crank Nicolson
BDF	Backward Differentiation Formula
BEM	Boundary Element Method
DRBFN	Direct Radial Basis Function Network
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
GMRES	Generalized Minimal Residual Method
IPC	Incremental Pressure Correction
IPCPP	Incremental Pressure Correction with Pressure Prediction
IRBFN	Indirect/Integrated Radial Basis Function Network
MQ	MultiQuadric
NSE	Navier-Stokes Equation
PDE	Partial Differential Equation
RBF	Radial Basis Function
RK	Runge-Kutta
SL	Semi-Lagrangian
SVD	Singular Value Decomposition
TCN	Taylor-Crank-Nicolson
TE	Taylor-Euler
TPS	Thin Plate Splines

List of Tables

List of Figures

Chapter 1

Introduction

This chapter introduces an overall picture of the present research in which a brief overview of numerical methods for incompressible flows is first presented, followed by a review of meshless Radial Basis Function Network (RBFN) methods, particularly the Indirect/Integrated RBFN (IRBFN) method based on which all numerical schemes in this thesis are developed. An overview of the present research including the objectives and the methodology of the research as well as the outline of the thesis are then introduced.

1.1 Numerical Methods for Fluid Dynamics

Like many other physical processes in nature, the motion of fluids is governed by a set of partial differential equations (PDEs) known as the governing equations. These equations are based upon the fundamental conservation laws in physics postulating that mass, momentum and energy are conserved in any fluid motions. In most cases, analytical solutions to the governing equations cannot be obtained due to their complexity in nature. Numerical methods, therefore, have been widely used to find an approximate rather than exact solution to the

equations. Numerical solutions can be found by first discretizing the PDEs in space and time. As a result, one has a system of algebraic equations which can be linear or nonlinear. These equations are then to be solved for some physical properties of fluids (known as field variables) at discrete points within the computational domain. The common field variables include velocity, pressure and temperature.

The most commonly used numerical methods for solving the governing equations of fluid flows are finite difference (FDM), finite element (FEM), finite volume (FVM) and spectral method (SM). Fundamental backgrounds and applications in fluid dynamics can be found in (Peyret and Taylor, 1983; Tannehill et al., 1997; Roache, 1998) for the FDM, (Girault and Raviart, 1986; Zienkiewicz and Taylor, 2000; Gresho and Sani, 2000; Donea and Huerta, 2003) for FEM, (Patankar, 1980; Tannehill et al., 1997; LeVeque, 2002) for the FVM, and (Canuto et al., 1988; Karniadakis and Sherwin, 1999; Peyret, 2002) for the SM method.

1.2 Meshless Methods based on Radial Basis Functions

All the conventional methods mentioned above can be considered as mesh-based methods characterized by their reliance on a computation mesh with certain relationship between the nodes. Although these mesh-based methods are well established and efficient for various problems in engineering and sciences, they have the drawback that mesh generation is a non-trivial and time-consuming task. In addition, mesh distortion in large deformation problems can cause severe accuracy degradation. Although remeshing can improve the accuracy of the solution in large deformation problems, it requires complex, robust and adaptive algorithms to be developed, not to mention the burden associated

with a large number of remeshing operations. Meshless methods have been developed to avoid these difficulties by constructing the approximations entirely in terms of a set of points. The methods can be extended to multi-dimensional problems without any significant effort. Furthermore, boundary conditions can be handled naturally thanks to the unstructured feature of the methods.

In the last twenty years, meshless methods based on radial basis functions (RBFs) have increasingly attracted much attention from researchers not only for interpolating multivariate scattered data and approximating functions but also for solving PDEs. The idea of using RBFs for solving PDEs was first proposed by Kansa (1990a,b) where a global multiquadrics (MQ) scheme was used together with the collocation method to solve parabolic, hyperbolic and elliptic PDEs. This method is hereby referred to as Kansa's method or the DRBFN (Direct Radial Basis Function Network) method. Since its introduction, the method has been widely applied to various transient problems, particularly in computational fluid dynamics (CFD). Moridis and Kansa (1994) devised a hybrid scheme combining MQ with the numerical inversion of Laplace transforms for solving linear or linearized time-dependent PDEs. The applications of MQ and thin plate splines (TPS) to heat transfer problems and linear advection-diffusion equations were reported in (Zerroukat et al., 1998, 2000). RBFs-based methods were used for solving natural convection, porous media, and solid-liquid system problems in (Sarler et al., 2001; Perko et al., 2001; Kovacevic et al., 2003; Sarler et al., 2004). Hon et al. (1999) proposed a computational algorithm using MQ to solve the shallow-water equations. It is noted that, the global MQ and TPS are ranked to be the most accurate for scattered data approximation (Franke, 1982). The accuracy of MQ, however, is influenced by the so-called shape parameter whose optimal value depends on data (Carlson and Foley, 1991; Rippa, 1999).

In addition to the DRBFN method, other meshless approaches to solving PDEs based on RBFs have been developed in the last decade. Unlike the DRBFN

method where the RBF coefficient matrix is asymmetric, the Hermite-type collocation method proposed by Fasshauer (1996) yields a symmetric positive definite coefficient matrix which is guaranteed to be non-singular. Atluri and Zhu (1998b,a) developed the meshless local Petrov-Galerkin (MLPG) where compactly supported RBFs among others can be used as trial functions. Another approach investigated in (Chen et al., 1998; Golberg and Chen, 1999, 2001) is to combine the method of fundamental solutions (MFS) with techniques from the dual reciprocity method (DRM) where the MFS is employed to find the homogeneous solution and the approximation of particular solutions can be found in the context of the DRM using RBFs. Hon (2002) proposed the quasi-radial basis function method in which the quasi-interpolation (Beatson and Powell, 1990) and RBFs are combined so that the ill-conditioning problem resulted from a global RBFs scheme is eliminated.

1.3 Indirect/Integrated Radial Basis Function Networks Method

Recently, a new method, namely the Indirect/Integrated Radial Basis Function Networks (IRBFN) method was proposed by Mai-Duy and Tran-Cong (2001a, 2003) for approximating functions and their derivatives, and solving differential equations. In the DRBFN method, a function is first approximated by the radial basis functions, and its derivatives are then calculated by differentiating such closed form RBF approximation. Although the method has the ability to represent any continuous function to a prescribed degree of accuracy, the process of differentiation magnifies any errors that might arise from approximating the original function and thus result in inaccurate derivatives. For the IRBFN method, on the other hand, the highest order derivative is first decomposed into radial basis functions, its lower derivatives and the function itself are then obtained symbolically via the process of integration. In contrast to

the process of differentiation where any errors associated with the function approximation might be amplified, the integration has the effect of averaging out such errors. The IRBFN method, therefore, results in a better approximation accuracy than the usual approach as in the DRBFN method with the same numerical configuration (Mai-Duy and Tran-Cong, 2003). The IRBFN method has been successfully applied in solving elliptic PDEs and steady Navier-Stokes equations (Mai-Duy and Tran-Cong, 2001a,b).

1.4 About the Thesis

The aim of this research is to further develop the IRBFN method for time-dependent PDEs as well as for the numerical solutions of unsteady incompressible viscous flows. In particular, the objectives of the research are:

- to devise and implement meshless IRBFN-based schemes for solving time-dependent PDEs;
- to develop meshless IRBFN-based schemes for unsteady incompressible viscous flows;
- to devise and implement meshless IRBFN-based schemes for the numerical simulations of some moving boundary problems including passive transport problems and interfacial flows.

Preliminary studies led us to some specific choices as parts of the methodology for the current research:

- Devising numerical schemes based on the IRBFN method to solve time-dependent PDEs is a necessary step to set up building blocks for the applications of the method to unsteady fluid flows and moving boundary problems.

- Hardy's multiquadrics and Duchon's thin-plate splines are used as the basis functions in the IRBFN formulation thanks to their high accuracy in approximating functions and their derivatives;

- Since the IRBFN method is highly accurate in approximating functions and derivatives in space, the temporal discretization to be used for time-dependent problems should be of high order so that it does not degrade the whole accuracy of the new schemes;

- In addition to non-split methods where all equations of interest are solved for the unknowns simultaneously, the splitting approach need to be studied for transient problems to not only allow a decomposition of initially difficult problems into relatively easier substeps but also to reduce the size of the system of equations to be solved at each time step.

All fluid flows studied in this work are transient and incompressible. For moving boundary problems, the current research is limited to immiscible, non-reacting Newtonian fluids in laminar, isothermal, two-fluid flows without phase change.

1.5 Outline of the Thesis

The rest of the thesis is organized as follows: Chapter 2 presents the fundamental background of this research including the basic equations that govern the unsteady incompressible flows, the numerical methods for solving those equations, and the theoretical background of the meshless methods based on radial basis functions with full details on the IRBFN method.

Chapter 3 reports the derivation and implementation of two new meshless IRBFN-based schemes for time-dependent PDEs. In these schemes, high-order time integration methods are coupled with the IRBFN method to efficiently solve different kinds of PDEs, including parabolic PDEs, hyperbolic PDEs and

advection-diffusion equations. Extensive numerical experiments are performed to investigate the accuracy and the stability of the new schemes as well as the effects of different parameters on the accuracy and efficiency of the new schemes.

Chapter 4 presents a new meshless numerical approach to solving passive transport problems where material interfaces are forced to move, stretch and merge together in an ambient flow. Two new meshless schemes based on the semi-Lagrangian method and the Taylor series expansion in combination with the level set method are derived and implemented to solve such a special moving boundary problem.

Chapter 5 reports two novel numerical schemes for solving unsteady Navier-Stokes equations. In the new schemes, the projection method is used in the meshless framework of the IRBFN method to successively solve subproblems for the velocity field and pressure in a segregated manner. Numerical experiments on the new schemes in terms of accuracy and stability are performed for verification purposes.

Chapter 6 presents a new numerical scheme for the numerical simulation of interfacial flows. Unlike in chapter 4 where the influence of the moving interfaces on the ambient fluid flow is ignored, the meshless scheme in chapter 6 deals with a more realistic moving boundary problems in which the interaction between the two phases is fully captured with surface tension taken into account.

Finally, the conclusions of this thesis and suggestions for future work are presented in chapter 7.

Chapter 2

Fundamental Background

The background of the current research is presented in this chapter. The basic equations that govern the dynamic behavior of fluids in motion are first presented. The rest of this chapter is then devoted to the fundamentals of the numerical solutions to the governing equations. These include the space discretization with full details on the IRBFN method, time discretization especially focused on high order methods, and the solution of the resultant system of algebraic equations which can be linear or nonlinear.

2.1 Governing Equations of Fluid Dynamics

The governing equations of fluid dynamics are formulated with the assumption that the density of the fluid is high enough so that it can be approximated as a continuum. This implies that even an infinitesimally small (in the sense of differential calculus) element of the fluid still contains a sufficient number of particles, for which mean velocity and mean kinetic energy can be specified. The various properties of a fluid in motion can be therefore assumed to vary continuously with respect to space and time. The commonly encountered

physical properties of fluids include density, velocity, pressure and temperature.

The derivation of the governing equations of fluid dynamics is based on the conservation laws stating that mass, momentum and energy are conserved in any fluid motions. In order to depict mathematical formulations of the conservation laws, let's consider a control volume V in the fluid which is fixed in space and time in an Eulerian coordinate system. The boundary of V is a surface S with the unit vector $\mathbf{n}$ normal to S and pointing from the inside of V to the outside.

Continuity Equation

The mass conservation requires that the rate of accumulation of mass in V is balanced by the mass flux across S, yielding

$$\frac{\partial}{\partial t}\int_V \rho \,\mathrm{d}V + \int_S \rho\, \mathbf{n}\cdot\mathbf{u}\,\mathrm{d}S = 0, \tag{2.1}$$

where ρ is the density and $\mathbf{u}$ the velocity vector. By applying Gauss's divergence theorem, (2.1) can be rewritten as follows

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

Since V can be arbitrary, one has

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

or

$$\frac{D\,\rho}{Dt} + \rho\nabla\cdot\mathbf{u} = 0, \tag{2.4}$$

where $\frac{D}{Dt} = \frac{\partial}{\partial t} + (\mathbf{u}\cdot\nabla)$ is the material derivative. The equation (2.4) is referred to as the continuity equation. For the incompressible fluids, the density ρ is constant with respect to both space and time, and hence $\frac{D\rho}{Dt} = 0$. The continuity

equation for incompressible fluid is then reduced to

$$\nabla \cdot \mathbf{u} = 0. \tag{2.5}$$

Momentum Equation

The conservation of momentum states that the rate of accumulation of momentum in V plus the flux of momentum across S is equal to the rate of change in momentum due to body forces and surface stresses. Mathematically this postulation leads to

$$\frac{\partial}{\partial t}\int_V \rho\,\mathbf{u}\,\mathrm{d}V + \int_S \rho\,\mathbf{n}\cdot\mathbf{u}\mathbf{u}\,\mathrm{d}S = \int_V \rho\,\mathbf{f}\,\mathrm{d}V + \int_S \mathbf{n}\cdot\boldsymbol{\sigma}\,\mathrm{d}S, \tag{2.6}$$

where $\mathbf{f}$ is the body force vector per unit mass, and $\boldsymbol{\sigma}$ is the stress tensor

$$\boldsymbol{\sigma} = -p\,\mathbf{I} + \boldsymbol{\tau}, \tag{2.7}$$

in which p is the pressure, $\boldsymbol{\tau}$ is the deviatoric or extra stress tensor. Applying Gauss's divergence theorem to (2.6) yields

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

Since V is arbitrary, one has

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

Applying vector calculus and the continuity equation (2.4) to (2.9) yields

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

or

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

Equation (2.11) is referred to as the equation of motion or the momentum equation.

Energy Equation

The conservation of energy in the control volume V means that the rate of accumulation of energy plus the flux of energy across S is equal to the flux of heat coming in through S plus the rate of change in energy due to surface stresses. Mathematically this principle implies

$$\frac{\partial}{\partial t}\int_V \rho\, E \,\mathrm{d}V + \int_S \rho\, E(\mathbf{n}\cdot\mathbf{u})\,\mathrm{d}S = -\int_S \mathbf{n}\cdot\mathbf{q}\,\mathrm{d}S + \int_S \mathbf{n}\cdot(\boldsymbol{\sigma}\cdot\mathbf{u})\,\mathrm{d}S, \quad (2.12)$$

where $\mathbf{q}$ represents the heat flux vector which is related to temperature gradients via the Fourier law of diffusion

$$\mathbf{q} = -k\nabla T, \quad (2.13)$$

in which k is the thermal conductivity that may be a function of temperature T. In equation (2.12) E is the total specific energy given by

$$E = e + \frac{1}{2}\,\mathbf{u}^2 - \mathbf{f}\cdot\mathbf{u}, \quad (2.14)$$

in which e is the specific internal energy, $\frac{1}{2}\,\mathbf{u}^2$ is the specific kinetic energy, and $-\mathbf{f}\cdot\mathbf{u}$ is the specific potential energy. Applying Gauss's divergence theorem to (2.12) yields

$$\int_V \left[\frac{\partial}{\partial t}(\rho\, E) + \nabla\cdot(\rho\, E\,\mathbf{u})\right]\mathrm{d}V = \int_V \left[-\nabla\cdot\mathbf{q} + \nabla\cdot(\boldsymbol{q}\cdot\mathbf{u})\right]\mathrm{d}V. \quad (2.15)$$

Since V is arbitrary, one has

$$\frac{\partial}{\partial t}(\rho\, E) + \nabla\cdot(\rho\, E\,\mathbf{u}) = -\nabla\cdot\mathbf{q} + \nabla\cdot(\boldsymbol{q}\cdot\mathbf{u}). \quad (2.16)$$

Equation (2.16) can be further simplified as follows (Richardson, 1989)

$$\rho\frac{De}{Dt} = -\nabla\cdot\mathbf{q} - p\nabla\cdot\mathbf{u} + \boldsymbol{\tau} : \nabla\mathbf{u}, \tag{2.17}$$

where

$$\boldsymbol{\tau} : \nabla\mathbf{u} = \mathbf{tr}(\boldsymbol{\tau}\cdot\nabla\mathbf{u}). \tag{2.18}$$

Equation (2.17) is referred to as the energy equation.

Constitutive Equation

In addition to the continuity equation (2.4), momentum equation (2.11), and energy equation (2.17), a constitutive equation is needed to relate the extra stress tensor $\boldsymbol{\tau}$ to the deformation experienced by the fluid. In other words, the constitutive equation describes the material behavior of the fluid in motion. For Newtonian fluids, the stress tensor is linearly proportional to the rate of strain tensor. The constitutive equation for these fluids is given by

$$\boldsymbol{\tau} = 2\eta\mathbf{D}, \tag{2.19}$$

where η is the viscosity of the fluid, and $\mathbf{D}$ is the rate-of-strain tensor

$$\mathbf{D} = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right). \tag{2.20}$$

Initial and Boundary Conditions

In general, the unknowns to be found in the governing equations of incompressible flows are velocity, pressure, temperature. For a particular problem to be solved, initial and boundary conditions must be taken into account along with the governing equations presented above. Initial conditions define the values of the unknowns at the starting point in time. Boundary conditions describe the constraints on the fluid flow. Let's denote by $\mathbf{K}$ the viscous force per unit area on the boundary of interest, $\mathbf{K} \equiv \boldsymbol{\tau}\cdot\mathbf{n} = \eta\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\cdot\mathbf{n}$, some common

Table 2.1: Some common boundary conditions for Navier-Stokes equations

Boundary condition	Description
$\mathbf{u} = \mathbf{u}_0$	Inflow/Outflow velocity
$p = p_0$, $\mathbf{K} = 0$	Outflow/Pressure
$\mathbf{n} \cdot \mathbf{u} = 0$, $\mathbf{t} \cdot \mathbf{K} = 0$	Slip/Symmetry
$\mathbf{u} = 0$	No slip
$\mathbf{t} \cdot \mathbf{u} = 0$, $p = p_0$, $\mathbf{n} \cdot \mathbf{K} = 0$	Normal flow/Pressure

Note: $\mathbf{K} \equiv \boldsymbol{\tau} \cdot \mathbf{n} = \eta \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \cdot \mathbf{n}$: Viscous boundary force

boundary conditions are presented in Table (2.1).

2.2 Numerical Solutions of Time-dependent PDEs

The governing equations presented in the previous section are in forms of PDEs. As mentioned in the previous chapter, numerical methods are used to solve the equations since for most cases in practice, it is difficult, if not impossible, to find out analytical solutions.

Although there are various numerical methods for time-dependent PDEs available nowadays, many of them share a common path to the solutions of the equations: employing a separate discretization in space and time. This approach is known as the method of lines (MOL) (Hirt et al., 1974). In practice, there are two forms of MOL as follows (Knabner and Angermann, 2003)

- The vertical method of lines: The PDEs are first discretized with respect to the spatial variables (e.g. by means of the FDM, FEM, FVM or the meshless methods). This results in a system of ordinary differential equations (ODEs) which is known as a semidiscrete approximation to the PDEs. In a further step, the system is advanced in time using certain well-developed ODE solvers to obtain the unknowns for each time step.

- The horizontal method of lines: The time-dependent PDEs are first discretized with respect to the time variable to obtain a sequence of steady

problems. The semidiscrete approximation of the PDEs is then followed by a further discretization in space to obtain a full discretization which is in form of a system of algebraic equations.

In this thesis, both forms of the method of lines are used to solve time-dependent PDEs. The rest of this section will present the space discretization by the RBFNs-based methods and time discretization by high order time integration schemes. In addition, numerical methods for linear and nonlinear system of algebraic equations that result from the discretizations will also be discussed.

2.2.1 Space Discretization by RBFN Methods

Space discretization approximates unknown functions and their derivatives with respect to space appearing in the differential equations in such a way that the original continuous problem is reduced to a discrete form in space. Space discretization by meshless RBFN methods will be presented in this section.

The DRBFN Method

Let $u(\boldsymbol{x})$ be an unknown function continuously defined on a bounded domain $\Omega \subset \mathbf{R}^d$, $d = 1, 2, 3$, and $\{\boldsymbol{x}_j\}_{j=1}^M$ be a finite set of discrete points in Ω. The approximation $\hat{u}(\boldsymbol{x})$ to function $u(\boldsymbol{x})$ can be written as a linear combination of N radial basis functions

$$u(\boldsymbol{x}) \approx \hat{u}(\boldsymbol{x}) = \sum_{j=1}^{N} \omega_j \, g_j(\boldsymbol{x}), \tag{2.21}$$

where $\{g_j(\boldsymbol{x})\}_{j=1}^N$ is a given set of radial basis functions, and $\{\omega_j\}_{j=1}^N$ is the set of weight coefficients to be found. The derivatives of $u(\boldsymbol{x})$ up to second order with respect to space can be then approximated by successively differentiating

equation (2.21) as follows

$$\frac{\partial u}{\partial x} \approx \frac{\partial \hat{u}}{\partial x} = \sum_{j=1}^{N} \omega_j \frac{\partial g_j}{\partial x}, \tag{2.22}$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\partial^2 \hat{u}}{\partial x^2} = \sum_{j=1}^{N} \omega_j \frac{\partial^2 g_j}{\partial x^2}. \tag{2.23}$$

Among other radial basis functions, Hardy's multiquadrics (MQ) and Duchon's thin plate splines (TPS) are widely used for the ability to yield good approximation results (Franke, 1982). MQ and its derivatives up to second order are given by

$$g_j = \sqrt{r_j^2 + s_j^2}, \tag{2.24}$$

$$\frac{\partial g_j}{\partial x} = \frac{x - x_j}{\sqrt{r_j^2 + s_j^2}}, \tag{2.25}$$

$$\frac{\partial^2 g_j}{\partial x^2} = \frac{r_j^2 + s_j^2 - (x - x_j)^2}{(r_j^2 + s_j^2)^{1.5}}, \tag{2.26}$$

where $r_j = ||\boldsymbol{x} - \boldsymbol{x}_j||$ is the Euclidian norm, s_j is the shape parameter. Similarly, TPS and its derivatives up to second order are given by

$$g_j = r_j^{2m} \log(r_j) \tag{2.27}$$

$$\frac{\partial g_j}{\partial x} = (x - x_j)\, r_j^{2(m-1)} (2\, m\, \log(r_j) + 1), \tag{2.28}$$

$$\begin{aligned} \frac{\partial^2 g_j}{\partial x^2} = {} & 2(m-1)(x - x_j)^2 r_j^{2(m-2)} (2\, m\, \log(r_j) + 1) \\ & + r_j^{2(m-1)} (2\, m\, \log(r_j) + 1) + 2\, m\, r_j^{2(m-1)} (x - x_j)^2. \end{aligned} \tag{2.29}$$

While being ranked as the most accurate, the approximation quality of the MQ depends on the shape parameter s_j whose optimal value is found to be problem-dependent (Carlson and Foley, 1991; Rippa, 1999). TPS, on the other hand,

has no adjustable parameter, and hence is easier to use. Details on the Kansa's method and its applications to solving PDEs can be found in (Kansa, 1990a,b).

The IRFBN Method

Unlike the DRBFN method, the IRBFN method starts with the decomposition of the highest derivative (second order in this thesis) of the unknown function into a linear combination of RBFs. The lower derivatives and the function itself are then successively obtained by symbolic integrations. It is observed that, in contrast to the differentiation where any errors associated with the function approximation can be magnified in obtaining its derivatives, the process of integration implemented in the IRBFN method averages out such errors, and hence yields a more accurate results. For brevity, the IRBFN formulation for two-dimensional problems is recaptured as follows

$$\frac{\partial^2 u(x,y)}{\partial x^2} \approx \sum_{j=1}^{N} \hat{\omega}_j \, \hat{g}_j(x,y), \tag{2.30}$$

$$\frac{\partial u(x,y)}{\partial x} \approx \sum_{j=1}^{N} \hat{\omega}_j \int \hat{g}_j(x,y)\mathrm{d}x + C_1(y), \tag{2.31}$$

$$u(x,y) \approx \sum_{j=1}^{N} \hat{\omega}_j \iint \hat{g}_j(x,y)\mathrm{d}x\mathrm{d}x + C_1(y)x + C_2(y), \tag{2.32}$$

where $\{\hat{g}_j(x,y)\}_{j=1}^{N}$ is a set of radial basis functions such as multiquadrics

$$\hat{g}_j(x,y) = \sqrt{r_j^2 + s_j^2}, \quad j = 1, \ldots, N, \tag{2.33}$$

or thin plate splines

$$\hat{g}_j(x,y) = r_j^{2m} \log r_j, \quad j = 1, \ldots, N, \tag{2.34}$$

in which $r_j = \sqrt{(x-x_j)^2+(y-y_j)^2}$ is the Euclidian norm, s_j is the RBF shape parameter given by (Moody and Darken, 1989)

$$s_j = \beta\, d_j^{min}, \tag{2.35}$$

where β is the user-defined parameter and d_j^{min} is the distance from the j^{th} data point to its nearest neighboring point.

In Equation (2.32), $C_1(y)$, $C_2(y)$ are constants along the integration direction (x-direction) but are functions along the other direction (y-direction). Following the same rule as above, the constant of integration $C_1(y)$ can be approximated by the IRBFN method as follows

$$\frac{\mathrm{d}^2 C_1(y)}{\mathrm{d}y^2} = \sum_{j=1}^{P} \tilde{\omega}_j\, \tilde{g}_j(y), \tag{2.36}$$

$$\frac{\mathrm{d}C_1(y)}{\mathrm{d}y} = \sum_{j=1}^{P} \tilde{\omega}_j \int \tilde{g}_j(y)\mathrm{d}y + \tilde{C}_1, \tag{2.37}$$

$$C_1(y) = \sum_{j=1}^{P} \tilde{\omega}_j \iint \tilde{g}_j(y)\mathrm{d}y\mathrm{d}y + \tilde{C}_1 y + \tilde{C}_2, \tag{2.38}$$

where P is the number of centres needed to approximate the constant of integration; $\hat{g}_j(x,y)$ and $\tilde{g}_j(y)$ are the two-dimensional and one-dimensional radial basis functions such as Hardy's multiquadrics MQ

$$\hat{g}_j(x,y) = \sqrt{(x-x_j)^2+(y-y_j)^2+s_j^2}, \tag{2.39}$$

$$\tilde{g}_j(y) = \sqrt{(y-y_j)^2+s_j^2}. \tag{2.40}$$

Following the same rule as in equations (2.36)-(2.38), the constant of integration $C_2(y)$ can be approximated as

$$C_2(y) = \sum_{j=1}^{P} \bar{\omega}_j \iint \tilde{g}_j(y)\mathrm{d}y\mathrm{d}y + \bar{C}_1 y + \bar{C}_2, \tag{2.41}$$

It is noted that $\tilde{C}_1$, $\tilde{C}_2$ in (2.38), and $\bar{C}_1$, $\bar{C}_2$ in (2.41) are really constants whose values are also to be found. Once these constants of integration and the network weights $\hat{\omega}_j, j = 1, N$ are found, the derivatives up to second order and the function $u(x, y)$ itself can be approximated by equations (2.30-2.32).

2.2.2 Time Discretization

Time discretization approximates unknown functions and their derivatives with respect to time appearing in the time dependent PDEs in such a way that the original continuous problem is reduced into a discrete form in time. Time discretization, time stepping and time integration are used interchangeably in the CFD literature.

Vertical method of lines

In the vertical form of the method of lines as mentioned in the beginning of section 2.2, one first discretizes the original PDEs in space to obtain a system of ordinary differential equations (ODEs). A time discretization scheme is then used to solve the ODEs. There is a well-developed class of ODE solvers including Runge-Kutta (RK), Adams-Bashforth (AB), Backward Differentiation Formula (BDF), etc... In practice, these methods can be further categorized into One-step methods (e.g. RK) and Multi-step methods (e.g. AB, BDF) (Quarteroni et al., 2000). Consider a system of ODEs

$$\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t} = \boldsymbol{f}(t, \mathbf{y}(t)), \quad 0 < t < T, \tag{2.42}$$

$$\mathbf{y}(t = 0) = \mathbf{y}^0 \tag{2.43}$$

where $\boldsymbol{f}(t, y(t))$ is a set of given functions. The time interval $[0, T]$ is partitioned into N_t subintervals $[t^n, t^{n+1}]$ of length $\Delta t = T/(N_t - 1)$ with $t^0 = 0$ and $t^{N_t} = T$. Denote by $\mathbf{y}^n = \mathbf{y}(t^n)$ the value of $\mathbf{y}$ at $t = t^n$, some commonly used methods

for ODEs are presented as follows.

Runge-Kutta method

At time step $t = t^{n+1}$, four function evaluations of $\boldsymbol{f}$ are performed to update the value of $\mathbf{y}^{n+1} = \mathbf{y}(t^{n+1})$ as follows.

$$\begin{aligned}
\mathbf{R}_1 &= \Delta t\, \boldsymbol{f}\left(t^n, \mathbf{y}^n\right) \\
\mathbf{R}_2 &= \Delta t\, \boldsymbol{f}\left(t^{n+1/2}, \mathbf{y}^n + \frac{\mathbf{R}_1}{2}\right), \\
\mathbf{R}_3 &= \Delta t\, \boldsymbol{f}\left(t^{n+1/2}, \mathbf{y}^n + \frac{\mathbf{R}_2}{2}\right), \\
\mathbf{R}_4 &= \Delta t\, \boldsymbol{f}\left(t^{n+1}, \mathbf{y}^n + \mathbf{R}_3\right), \\
\mathbf{y}^{n+1} &= \mathbf{y}^n + \frac{\mathbf{R}_1 + 2\,\mathbf{R}_2 + 2\,\mathbf{R}_3 + \mathbf{R}_4}{6}.
\end{aligned} \tag{2.44}$$

BDF method

BDF method is mostly used as high-order time integration schemes for stiff ODEs. The general BDF method of order k can be described as follows.

$$\frac{1}{\Delta t}\sum_{j=0}^{k} a_j\, \mathbf{y}^{n+1-j} = \boldsymbol{f}(t^{n+1}, \mathbf{y}^{n+1}), \tag{2.45}$$

where $a_j, j = 0, \ldots, k$ are the coefficients whose values are given in Table (2.2).

Order	a_0	a_1	a_2	a_3
1	1	-1		
2	3/2	-2	1/2	
3	11/6	-3	3/2	-1/3

Table 2.2: Coefficients of the BDF methods of order $k = 1, 2, 3$.

Adams-Bashforth (AB) method

The general AB method of order k applied to Equation (2.42) is of the form

$$\frac{\mathbf{y}^{n+1} - \mathbf{y}^n}{\Delta t} = \sum_{j=0}^{k-1} b_j \, \boldsymbol{f}(t^{n-j}, \mathbf{y}^{n-j}) \tag{2.46}$$

where $b_j, j = 0, \ldots, k$ are the coefficients given in Table (2.3). Full detailed discussions on ODE solvers can be found in (Quarteroni et al., 2000).

Order	b_0	b_1	b_2
1	1		
2	3/2	-1/2	
3	23/12	-16/12	5/12

Table 2.3: Coefficients of the AB methods of order $k = 1, 2, 3$.

Horizontal method of lines

In the horizontal form of the method of lines, the time discretization is applied to the PDEs first to create a sequence of time-independent PDEs. For this case, time discretization methods can be classified into three groups: (a) explicit, (b) implicit, and (c) semi-implicit (or explicit-implicit) schemes.

Consider a bounded Lipschitz domain Ω with the boundary Γ from which $Q_{\Omega T} = \Omega \times (0, T),\ T > 0$ and its boundary $Q_{\Gamma T} = \Gamma \times (0, T)$ are defined. Given the functions $f : Q_{\Omega T} \to \mathbb{R}$, $g : Q_{\Gamma T} \to \mathbb{R}$, and $u_0 : \Omega \to \mathbb{R}$, the problem is to find a function $u : Q_{\Omega T} \to \mathbb{R}$ such that

$$\frac{\partial u}{\partial t} + \mathcal{S}u = f \quad \text{in } Q_{\Omega T}, \tag{2.47}$$

$$\mathcal{B}u = g \quad \text{on } Q_{\Gamma T}, \tag{2.48}$$

$$u = u_0 \quad \text{in } \Omega \times 0 \tag{2.49}$$

where $\mathcal{S}v$ and $\mathcal{B}v$ denote differential expressions for some function $v : \Omega \to \mathbb{R}$. Since this subsection is to review the time discretization, only the time dis-

cretization of Equation (2.47) is presented here whereas the detailed discussion on the boundary and/or initial conditions (2.48) and (2.49) will be described later in Chap.(3). Denoting by u^n,f^n and u^{n+1}, f^{n+1} the values of u and f at time $t = t^n$ and $t = t^{n+1}$, respectively, the explicit, implicit and semi-implicit schemes can be described as follows.

Explicit methods

In the explicit schemes, the unknowns can be calculated directly from those in previous time steps without solving any system of equations. Forward Euler method is an example of explicit methods when applied to Equation (2.47) as follows.

$$\frac{u^{n+1} - u^n}{\Delta t} = \mathcal{S}u^n + f^n. \tag{2.50}$$

The main drawback of these methods is stability issue which requires small time step size.

Implicit methods

In the implicit methods, the unknown at each time step depends on itself under the form of differential expressions. As a result, after the space discretization is applied to the semi-discrete formulation, a system of linear or nonlinear equations has to be solved for the unknowns at each time step as follows.

$$\frac{u^{n+1} - u^n}{\Delta t} = \mathcal{S}u^{n+1} + f^{n+1}. \tag{2.51}$$

In general, implicit schemes allow larger time step size and are suitable for those problems where stability is highly required.

Semi-implicit methods

A general semi-implicit method widely used in CFD is the θ method (Quarteroni

and Valli, 1997)

$$\frac{u^{n+1}-u^n}{\Delta t} = \theta\,\mathcal{S}u^{n+1} + (1-\theta)\mathcal{S}u^n + \theta\,f^{n+1} + (1-\theta)f^n, \tag{2.52}$$

where $0 \le \theta \le 1$. It is noted that $\theta = 0$ corresponds to the explicit forward Euler method, $\theta = 1$ is the implicit backward Euler method, whereas $\theta = 0.5$ is known as the Crank-Nicolson method. Semi-implicit methods have been widely used in solving PDEs of nonlinear convection-diffusion type to avoid nonlinearity in discretizing the equations where an implicit scheme is used for diffusion term and an explicit scheme is used for the convection term.

2.2.3 Solving Systems of Linear Equations

After space and time discretizations, one obtains a system of algebraic equations which can be linear or nonlinear. The numerical methods for the solution of linear systems are reviewed in this section, and an overview of the numerical methods for nonlinear systems will be presented in the next section.

A system of m linear equations in n unknowns can be written as

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, \ldots, m \tag{2.53}$$

or in matrix form as

$$\mathbf{A}\,\boldsymbol{x} = \mathbf{b}, \tag{2.54}$$

where $\boldsymbol{x} = \{x_j\}_{j=1}^{n}$ is the unknown vector, $\mathbf{A} = \{a_{ij}\}_{i=1,j=1}^{m,n}$ is the coefficient matrix, and $\mathbf{b} = \{b_i\}_{i=1}^{m}$ is the right-hand side vector.

If $m < n$, that is, the number of equations is smaller than the number of unknown variables, or if $m > n$ but $\mathbf{A}$ does not have a full rank (which means

that some equations are linear combinations of the other ones), the system is underdetermined and there are either no solution or infinitely many of them.

If $m > n$ and the matrix $\mathbf{A}$ has a full rank, that is, if the number of equations is greater than the number of unknown variables, there is generally no solution and the system is overdetermined.

If $m = n$ and the matrix $\mathbf{A}$ is nonsingular, the system (2.54) has a unique solution. Most of the methods presented below are suitable for this case.

Direct methods

Direct methods can give the solution in a predictable finite number of steps. Gauss elimination method belongs to this class. In addition, solving an equation system by means of matrix decompositions such as LU and QR factorizations can also be classified as direct methods.

Among others, LU factorization method is widely used for small-to-medium square system ($m = n$). This method is equivalent to Gauss elimination method where the matrix $\mathbf{A}$ is factorized into the product of two matrices, $\mathbf{A} = \mathbf{L}\,\mathbf{U}$, in which $\mathbf{L}$ is a lower triangular matrix and $\mathbf{U}$ is an upper triangular matrix. Once the matrices $\mathbf{L}$ and $\mathbf{U}$ have been computed, the solution to the linear system can be found by successively solving the two triangular systems

$$\mathbf{L}\,\mathbf{y} = \mathbf{b}, \quad \mathbf{U}\,\boldsymbol{x} = \mathbf{y}. \tag{2.55}$$

The orthogonal-triangular decomposition or QR factorization expresses any rectangular matrix $\mathbf{A}$ as the product of an orthogonal or unitary matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$. The linear system is then written as $\mathbf{Q}\,\mathbf{R}\,\boldsymbol{x} = \mathbf{b}$, or $\mathbf{R}\,\boldsymbol{x} = \mathbf{Q}^T\mathbf{b}$ (due to the fact that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$). The solving procedure is then

$$\mathbf{y} = \mathbf{Q}^T\mathbf{b}, \quad \boldsymbol{x} = \mathbf{R}^{-1}\mathbf{y}. \tag{2.56}$$

If $m > n$ and $\mathbf{A}$ does not have a full rank then any solution can be written as a

sum of a particular solution and a vector from the nullspace of $\mathbf{A}$. Alternatively, one can search for a solution $\boldsymbol{x}$ such that the Euclidian norm of $(\mathbf{A}\,\boldsymbol{x} - \mathbf{b})$ is minimized. In this case, one has a linear least-squares problem to be solved (Quarteroni et al., 2000).

Iterative methods

In many cases where the number of unknown variables is large, rounding errors can accumulate and prevent direct methods from yielding correct solutions. The same applies very much to systems with a nearly singular coefficient matrix. Iterative methods are good candidates for such cases as they are less sensitive to those problems and self-correcting at each iteration. Contrary to direct methods, iterative methods construct a series of solution approximations $x_i, i = 0, 1, 2, \ldots$ such that it converges to the exact solution of system (2.54). Theoretically, iterative methods yield the exact solution after an infinite number of steps. In practice, the iterative process is stopped at step k such that $||\mathbf{A}\,\boldsymbol{x}^{(k)} - \mathbf{b}|| < \epsilon$ or k exceeds a predefined maximum number of steps. The general algorithm for iterative methods can be expressed as follows.

$k = 0$

repeat

- $k = k + 1$
- Compute the approximate solution $\boldsymbol{x}^{(k)}$
- Compute the residual $r^{(k)} = \mathbf{A}\,\boldsymbol{x}^{(k)} - \mathbf{b}$

until $k \geq maxit$ or $||r^{(k)}|| \leq \epsilon$.

Among others, the GMRES (Generalized Minimal Residual) method proposed by Saad and Schultz (1986) is widely used for unsymmetric linear systems. The GMRES iterates are constructed as follows.

$$x^{(k)} = x^{(0)} + z_1 v^{(1)} + \ldots + z_k v^{(k)}, \tag{2.57}$$

where the coefficients $z_1, z_2, \ldots, z_k$ have been chosen to minimize the residual norm $||b-\mathbf{A}\,x^{(k)}||$, and $v_1, v_2, \ldots, v_k$ form the orthogonal basis of the Krylov subspace $\mathcal{K}_m(\mathbf{A}; \mathbf{v})$ which are calculated by Arnoldi orthogonal algorithm (Quarteroni et al., 2000).

2.2.4 Numerical Methods for the Solution of Nonlinear Systems

A system of nonlinear equations might result from space discretization of a nonlinear PDEs. At each time step then, the system to be solved can be written in matrix form as follows.

$$\mathbf{A}(\mathbf{u})\,\mathbf{u} = \mathbf{b}, \tag{2.58}$$

or

$$\mathbf{A}\,\mathbf{u} = \mathbf{b}(\mathbf{u}). \tag{2.59}$$

For these kinds of nonlinear systems, Picard and Newton-Raphson methods are commonly used.

Picard's method

The system (2.58) can be solved by Picard's method with relaxation as follows.

- given a guess $\mathbf{u}^0$ (frequently chosen as zero's),
- for $k = 0, 1, 2, \ldots$, until the prescribed criterion is satisfied (e.g. $||\mathbf{u}^{k+1} - \mathbf{u}^k||$ is sufficiently small)
  - solve the linear system $\mathbf{A}(\mathbf{u}^k)\,\mathbf{u}^* = \mathbf{b}$,
  - set $\mathbf{u}^{k+1} = \alpha\,\mathbf{u}^* + (1-\alpha)\,\mathbf{u}^k$, where $\alpha \in (0, 1]$ is a relaxation parameter.

The method is also known as *Successive Substitutions*. Similarly, the nonlinear system $\mathbf{A}\,\mathbf{u} = \mathbf{b}(\mathbf{u})$ can be solved by the method via a simple iteration

$$\mathbf{A}\,\mathbf{u}^* = \mathbf{b}(\mathbf{u}^k), \quad k = 0, 1, 2, \ldots \tag{2.60}$$

with the same update step for $\mathbf{u}^{k+1}$ as described above.

Newton's method

In this method, the nonlinear algebraic equations can be written as

$$\mathbf{F}(\mathbf{u}) = 0, \tag{2.61}$$

where $\mathbf{F}(u) = \mathbf{A}(\mathbf{u})\,\mathbf{u} - \mathbf{b}$, or $\mathbf{F}(\mathbf{u}) = \mathbf{A}\,\mathbf{u} - \mathbf{b}(\mathbf{u})$ corresponding to Equation (2.58) or (2.59). Vectors and/or matrices without argument are considered as constant, i.e., independent of $\mathbf{u}$. The approximation of $\mathbf{F}(\mathbf{u})$ is then written as

$$\mathbf{F}(\mathbf{u}) \approx \mathbf{F}(\mathbf{u}^k) + \mathbf{J}(\mathbf{u}^k) \cdot (\mathbf{u} - \mathbf{u}^k) \tag{2.62}$$

where $\mathbf{J} \equiv \nabla\mathbf{F}$ is the Jacobian of $\mathbf{F}$. Particularly, if $\mathbf{F} = [F_1, F_2, \ldots, F_m]^T$ and $\mathbf{u} = [u_1, u_2, \ldots, u_m]^T$, then the entry (i, j) of $\mathbf{J}$ is

$$J_{ij} = \frac{\partial F_i}{\partial u_j}, \quad i, j = 1, 2, \ldots, m.$$

The system (2.61) can be solved by Newton's method as follows.

- given a guess $\mathbf{u}^0$,
- for $k = 0, 1, 2, \ldots$, until the prescribed criterion is satisfied
  - solve the linear system $\mathbf{J}(\mathbf{u}^k)\,\delta\mathbf{u}^{k+1} = -\mathbf{F}(\mathbf{u}^k)$ for $\delta\mathbf{u}^{k+1}$,
  - set $\mathbf{u}^{k+1} = \mathbf{u}^k + \delta\mathbf{u}^{k+1}$.

The relevant termination criteria for the method are

$$||u^{k+1} - u^k|| \leq \epsilon_u, \quad \text{or} \quad ||\mathbf{F}(\mathbf{u}^{k+1})|| \leq \epsilon_r, \tag{2.63}$$

where ϵ_u and ϵ_r are given constants. Newton's method is quadratically convergent provided that the initial guess is sufficiently close to the solution and the Jacobian matrix is nonsingular (Quarteroni et al., 2000).

2.3 Concluding Remarks

The fundamental background of the current research has been presented in this chapter. Since the dynamic behavior of fluids in motion is governed by a set of partial differential equations, solving time-dependent PDEs is of primary interest in this thesis. In general, the numerical solution to such equations involves space and time discretizations, and solving the resultant system of linear or nonlinear equations. Basic tools commonly used in solving time-dependent PDEs have been summarized in this chapter and will be used in the rest of the thesis.

Chapter 3

Meshless IRBFN Schemes for Time-Dependent PDEs

This chapter presents numerical schemes based on the IRBFN method for time-dependent PDEs. Approximations of time-dependent functions and their derivatives with IRBFNs are first presented in section (3.2) from which two numerical schemes are derived in section (3.3). The IRBFN approximations are formulated such that the unknowns to be found are the nodal values of the functions rather than the weight coefficients as normally reported in the literature. These schemes can be used to solve a class of time-dependent PDEs such as parabolic, hyperbolic and advection-diffusion equations. Extensive numerical experiments are presented in section (3.4) to investigate the accuracy, the stability of the proposed schemes as well as the effects of the network parameters on the accuracy and efficiency of the schemes .

3.1 Introduction

The IRBFN method has been successfully applied to solving elliptic or time-independent PDEs (Mai-Duy and Tran-Cong, 2001a,b). In order to solve time-dependent PDEs, the method should be coupled with an appropriate time discretization to yield good results. Since the IRBFN method is a high-order space discretization, its counterpart with respect to time should be of high order as well to ensure that the ultimate accuracy of the numerical solution is not degraded. Among others, the standard θ-method (Quarteroni and Valli, 1997) is widely used for time discretization. It is noted that the θ-method corresponds to some well-known time-stepping methods with some special values of θ. For example, with $\theta = 0$ and $\theta = 1$, the method corresponds to the forward and backward Euler methods, respectively, whereas it is equivalent to the Crank-Nicolson method for $\theta = 1/2$. Another class of high-order methods for ordinary differential equations (ODEs) such as the one-step Runge-Kutta method and multistep methods such as Adams-Bashforth (AB), Backward Differentiation Formulation (BDF) (Quarteroni et al., 2000) can also be used for time discretization. In this chapter, two new numerical schemes for time-dependent PDEs are presented where the time discretization methods mentioned above and the IRBFN method are coupled together.

Furthermore, literature showed that the RBFN parameters have strong effects on the quality of the ultimate solution (Kansa, 1990b; Carlson and Foley, 1991; Rippa, 1999). They are the MQ's shape parameter s in Equation (2.24), the TPS's order m in Equation (2.27), and the point density. The effects of these parameters on the proposed IRBFN-based schemes are thoroughly studied in this chapter.

3.2 Approximation of Time-dependent Functions and their Derivatives

The IRBFN-based approximation of a function with respect to space has been reviewed in Chapter 2. For brevity, the IRBFN method for two-dimensional time-dependent problems is presented in this section. Extension to three-dimensional problems is straightforward.

Starting from the approximation of the second derivative in x-coordinate and following the same manner as presented in (2.30 - 2.32) where the network weights $\omega_j, j = 1, 2, \ldots$ are now considered to be functions with respect to time, the approximation of function $u(x, y, t)$ can be then calculated by the IRBFN method as follows.

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} \approx \sum_{j=1}^{N} \hat{\omega}_j(t)\, \hat{g}_j(x,y), \tag{3.1}$$

$$\frac{\partial u(x,y,t)}{\partial x} \approx \sum_{j=1}^{N} \hat{\omega}_j(t) \int \hat{g}_j(x,y)\mathrm{d}x + C_1(y), \tag{3.2}$$

$$u(x,y,t) \approx \sum_{j=1}^{N} \hat{\omega}_j(t) \iint \hat{g}_j(x,y)\mathrm{d}x\mathrm{d}x + C_1(y)x + C_2(y), \tag{3.3}$$

where $\{\hat{g}_j(x,y)\}_{j=1}^{N}$ is a set of radial basis functions such as multiquadrics

$$\hat{g}_j(x,y) = \sqrt{r_j^2 + s_j^2}, \quad j = 1, \ldots, N, \tag{3.4}$$

or thin plate splines

$$\hat{g}_j(x,y) = r_j^{2m} \log r_j, \quad j = 1, \ldots, N, \tag{3.5}$$

in which m is the TPS order, $r_j = \sqrt{(x - x_j)^2 + (y - y_j)^2}$ is the Euclidian

norm, and s_j is the RBF shape parameter given by (Moody and Darken, 1989)

$$s_j = \beta\, d_j^{min}, \tag{3.6}$$

where β is the user-defined parameter and d_j^{min} is the distance from the j^{th} data point to its nearest neighboring point. The constants of integration are approximated as follows.

$$\frac{\mathrm{d}^2 C_1(y)}{\mathrm{d}y^2} = \sum_{j=1}^{P} \tilde{\omega}_j(t)\, \tilde{g}_j(y), \tag{3.7}$$

$$\frac{\mathrm{d}C_1(y)}{\mathrm{d}y} = \sum_{j=1}^{P} \tilde{\omega}_j(t) \int \tilde{g}_j(y)\mathrm{d}y + \tilde{C}_1, \tag{3.8}$$

$$C_1(y) = \sum_{j=1}^{P} \tilde{\omega}_j(t) \iint \tilde{g}_j(y)\mathrm{d}y\mathrm{d}y + \tilde{C}_1 y + \tilde{C}_2, \tag{3.9}$$

$$C_2(y) = \sum_{j=1}^{P} \bar{\omega}_j(t) \iint \tilde{g}_j(y)\mathrm{d}y\mathrm{d}y + \bar{C}_1 y + \bar{C}_2, \tag{3.10}$$

where P is the number of centres needed to approximate the constants of integration in x-direction. Substituting Equations (3.7)-(3.10) into Equation (3.3) yields

$$u(x,y,t) \approx \sum_{j=1}^{N} \hat{\omega}_j(t) \iint \hat{g}_j(x,y)\mathrm{d}x\mathrm{d}x + \left[\sum_{j=1}^{P} \tilde{\omega}_j(t) \iint \tilde{g}_j(y)\mathrm{d}y\mathrm{d}y + \tilde{C}_1 y + \tilde{C}_2\right] x + \left[\sum_{j=1}^{P} \bar{\omega}_j(t) \iint \tilde{g}_j(y)\mathrm{d}y\mathrm{d}y + \bar{C}_1 y + \bar{C}_2\right], \tag{3.11}$$

or in the compact form

$$u(x,y,t) \approx \sum_{j=1}^{N+2P+4} \omega_j(t)\, g_j(x,y) = \boldsymbol{g}^T(x,y)\, \boldsymbol{\omega}(t) \tag{3.12}$$

where the basis function vector $\boldsymbol{g}(x,y)$ is given by

$$\begin{aligned}\boldsymbol{g}(x,y) \;=\; & \Big[\iint \hat{g}_1(x,y)\mathrm{d}x\mathrm{d}x, \iint \hat{g}_2(x,y)\mathrm{d}x\mathrm{d}x, \ldots, \iint \hat{g}_N(x,y)\mathrm{d}x\mathrm{d}x, \\ & x\iint \tilde{g}_1(y)\mathrm{d}y\mathrm{d}y, x\iint \tilde{g}_2(y)\mathrm{d}y\mathrm{d}y, \ldots, x\iint \tilde{g}_P(y)\mathrm{d}y\mathrm{d}y, xy, x, \\ & \iint \tilde{g}_1(y)\mathrm{d}y\mathrm{d}y, \iint \tilde{g}_2(y)\mathrm{d}y\mathrm{d}y, \ldots, \iint \tilde{g}_P(y)\mathrm{d}y\mathrm{d}y, y, 1\Big]^T \quad (3.13)\end{aligned}$$

As mentioned in section (2.2.1), the IRBFN formulation starts with the decomposition of the highest derivatives (second order in this thesis) of the unknown function into a linear combination of RBFs. In two-dimensional problems, this implies that there exists another alternative formulation starting from the second-order derivative in y-coordinate. Using the same rule as presented above, one gets the alternative IRBFN formulation for the approximation of function $u(x,y,t)$ and its derivatives as follows.

$$\begin{aligned}u(x,y,t) \approx \sum_{j=1}^{N} \hat{\omega}_j(t) \iint \hat{g}_j(x,y)\mathrm{d}y\mathrm{d}y + \left[\sum_{j=1}^{Q} \tilde{\omega}_j(t) \iint \tilde{g}_j(x)\mathrm{d}x\mathrm{d}x + \tilde{C}_1 x + \tilde{C}_2\right] y \\ + \left[\sum_{j=1}^{Q} \bar{\omega}_j(t) \iint \tilde{g}_j(x)\mathrm{d}x\mathrm{d}x + \bar{C}_1 x + \bar{C}_2\right], \quad (3.14)\end{aligned}$$

or in the compact form

$$u(x,y,t) \approx \sum_{j=1}^{N+2Q+4} \lambda_j(t)\, h_j(x,y) = \boldsymbol{h}^T(x,y)\, \boldsymbol{\lambda}(t) \quad (3.15)$$

where Q is the number of centres needed to approximate the constants of inte-

gration in y-direction. The basis function vector $\boldsymbol{h}(x,y)$ is given by

$$\begin{aligned}\boldsymbol{h}(x,y) \;=\; & \Big[\iint \hat{g}_1(x,y)\mathrm{d}y\mathrm{d}y, \iint \hat{g}_2(x,y)\mathrm{d}y\mathrm{d}y, \ldots, \iint \hat{g}_N(x,y)\mathrm{d}y\mathrm{d}y, \\ & y\iint \tilde{g}_1(x)\mathrm{d}x\mathrm{d}x, y\iint \tilde{g}_2(x)\mathrm{d}x\mathrm{d}x, \ldots, y\iint \tilde{g}_Q(x)\mathrm{d}x\mathrm{d}x, xy, y, \\ & \iint \tilde{g}_1(x)\mathrm{d}x\mathrm{d}x, \iint \tilde{g}_2(x)\mathrm{d}x\mathrm{d}x, \ldots, \iint \tilde{g}_Q(x)\mathrm{d}x\mathrm{d}x, x, 1\Big]^T \end{aligned} \tag{3.16}$$

Since the two formulations are written for the same function, the resulting approximations given by their applications must be identical. An explicit identity constraint was applied for this purpose in (Mai-Duy, 2001). However, this requires more storage and computation time. A different approach that was reported in (Mai-Cao and Tran-Cong, 2005; Mai-Duy, Mai-Cao and Tran-Cong, 2006) is presented here in which the above identity requirement is satisfied without any constraints. For doing so, equations (3.12) and (3.13) are first rewritten in a more compact form as follows.

$$u(\boldsymbol{x},t) \approx \sum_{j=1}^{\bar{N}} \omega_j(t) g_j(\boldsymbol{x}) = \boldsymbol{g}^T(\boldsymbol{x})\boldsymbol{\omega}(t) \tag{3.17}$$

where

$$\boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_{\bar{N}}(\boldsymbol{x})]^T \tag{3.18}$$

in which $\bar{N} = N + 2P + 4$, and the j^{th} component $g_j(\boldsymbol{x})$ is equal to that of (3.13). The new IRBFN formulations for time-dependent problems can be now established using the collocation method as follows.

Given a set of M data points and the corresponding nodal values of the function at certain point in time t, $\boldsymbol{u}(t) = [u_1(t), u_2(t), ..., u_M(t)]^T$, the application of the

collocation method by satisfying (3.12) at every data point yields

$$u(\boldsymbol{x}_i, t) \approx \sum_{j=1}^{\bar{N}} \omega_j(t)\, g_j(\boldsymbol{x}_i) = \boldsymbol{g}^T(\boldsymbol{x}_i)\, \boldsymbol{\omega}(t), \quad i = 1, 2, ..., M, \tag{3.19}$$

or in the compact form

$$\boldsymbol{u}(t) = \mathbf{G}\, \boldsymbol{\omega}(t), \tag{3.20}$$

where

$$\mathbf{G} = \begin{bmatrix} g_1(\boldsymbol{x}_1) & g_2(\boldsymbol{x}_1) & \dots & g_{\bar{N}}(\boldsymbol{x}_1) \\ g_1(\boldsymbol{x}_2) & g_2(\boldsymbol{x}_2) & \dots & g_{\bar{N}}(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(\boldsymbol{x}_M) & g_2(\boldsymbol{x}_M) & \dots & g_{\bar{N}}(\boldsymbol{x}_M) \end{bmatrix}. \tag{3.21}$$

The weight coefficients $\boldsymbol{\omega}(t)$ can be then determined by

$$\boldsymbol{\omega}(t) = \mathbf{G}^{-1}\, \boldsymbol{u}(t). \tag{3.22}$$

Substituting (3.22) into (3.12) yields the new IRBFN formulation starting from the second order derivative in x-coordinate

$$u(\boldsymbol{x}, t) \approx \boldsymbol{g}^T(\boldsymbol{x})\, \mathbf{G}^{-1}\, \boldsymbol{u}(t), \tag{3.23}$$

It is noted that the weight coefficients $\boldsymbol{\omega}(t)$ have been replaced by the nodal values of the function to be approximated in the new formulation. Applying the same procedure as above to (3.16) yields the alternative IRBFN formulation starting from the second order derivative in y-coordinate

$$u(\boldsymbol{x}, t) \approx \boldsymbol{h}^T(\boldsymbol{x})\, \mathbf{H}^{-1}\, \boldsymbol{u}(t), \tag{3.24}$$

where $\boldsymbol{h}(\boldsymbol{x})$ is defined in (3.16) and the matrix $\mathbf{H}$ is generated in a similar

manner as for $\mathbf{G}$ in (3.21) as follows.

$$\mathbf{H} = \begin{bmatrix} h_1(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_1) & \dots & h_{\bar{N}}(\boldsymbol{x}_1) \\ h_1(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_2) & \dots & h_{\bar{N}}(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\boldsymbol{x}_M) & h_2(\boldsymbol{x}_M) & \dots & h_{\bar{N}}(\boldsymbol{x}_M) \end{bmatrix}. \tag{3.25}$$

Equations (3.23) and (3.24) will be proved to yield the same resultant function approximation at every data points where $\boldsymbol{u}(t)$ is defined regardless of which radial basis function (e.g. multiquadrics or thin plate splines) is used. This is equivalent to prove

$$\boldsymbol{g}^T(\boldsymbol{x}_i)\boldsymbol{G}^{-1}\boldsymbol{u}(t) = \boldsymbol{h}^T(\boldsymbol{x}_i)\boldsymbol{H}^{-1}\boldsymbol{u}(t), \quad i = 1, 2, \dots, M \tag{3.26}$$

or equivalently the following system must be held

$$[g_1(\boldsymbol{x}_1), g_2(\boldsymbol{x}_1), \dots, g_{\bar{N}}(\boldsymbol{x}_1)]^T\boldsymbol{G}^{-1}\boldsymbol{u}(t) = [h_1(\boldsymbol{x}_1), h_2(\boldsymbol{x}_1), \dots, h_{\bar{N}}(\boldsymbol{x}_1)]^T\boldsymbol{H}^{-1}\boldsymbol{u}(t)$$

$$[g_1(\boldsymbol{x}_2), g_2(\boldsymbol{x}_2), \dots, g_{\bar{N}}(\boldsymbol{x}_2)]^T\boldsymbol{G}^{-1}\boldsymbol{u}(t) = [h_1(\boldsymbol{x}_2), h_2(\boldsymbol{x}_2), \dots, h_{\bar{N}}(\boldsymbol{x}_2)]^T\boldsymbol{H}^{-1}\boldsymbol{u}(t)$$

$\vdots$

$$[g_1(\boldsymbol{x}_M), g_2(\boldsymbol{x}_M), \dots, g_{\bar{N}}(\boldsymbol{x}_M)]^T\boldsymbol{G}^{-1}\boldsymbol{u}(t) = [h_1(\boldsymbol{x}_M), h_2(\boldsymbol{x}_M), \dots, h_{\bar{N}}(\boldsymbol{x}_M)]^T\boldsymbol{H}^{-1}\boldsymbol{u}(t)$$

or the following equation must be held

$$\boldsymbol{G}\boldsymbol{G}^{-1}\boldsymbol{u}(t) = \boldsymbol{H}\boldsymbol{H}^{-1}\boldsymbol{u}(t) \tag{3.27}$$

It is noted that in the statement above, the formulations of $\boldsymbol{G}$ and $\boldsymbol{H}$ in Equations (3.21) and (3.25) are used, respectively. Obviously, since $(\boldsymbol{G}\boldsymbol{G}^{-1})$ and $(\boldsymbol{H}\boldsymbol{H}^{-1})$ being identity matrices, Equation (3.27) is always held at every data points regardless of which radial basis function is used (since no such a specific requirement is used in the above proof). In other words, by replacing weight

coefficients $\boldsymbol{\omega}(t)$ with nodal values $\boldsymbol{u}(t)$ the resultant function approximation is unique regardless of which direction (x or y) the IRBFN formulation starts from. Without any additional constraints, the new IRBFN formulations significantly save computation time and storage in multi-dimensional problems.

To sum up, the IRBFN formulation for the approximation of a time-dependent function and its derivatives is written as follows.

$$\frac{\partial^2 u(\boldsymbol{x},t)}{\partial x^2} \approx \hat{\boldsymbol{g}}(\boldsymbol{x})^T \mathbf{G}^{-1} \boldsymbol{u}(t), \tag{3.28}$$

$$\frac{\partial u(\boldsymbol{x},t)}{\partial x} \approx \tilde{\boldsymbol{g}}(\boldsymbol{x})^T \mathbf{G}^{-1} \boldsymbol{u}(t), \tag{3.29}$$

$$u(\boldsymbol{x},t) \approx \boldsymbol{g}(\boldsymbol{x})^T \mathbf{G}^{-1} \boldsymbol{u}(t), \tag{3.30}$$

where

$$\begin{aligned} &\hat{g}_j(\boldsymbol{x}) = \varphi(||\boldsymbol{x}-\boldsymbol{x}_j||), \quad j = 1,\ldots,N, \\ &\hat{g}_j(\boldsymbol{x}) = 0, \qquad\qquad\qquad\quad j = N+1,\ldots,\bar{N}, \end{aligned} \tag{3.31}$$

$$\tilde{g}_j(\boldsymbol{x}) = \int \hat{g}_j(\boldsymbol{x}) \mathrm{d}x, \quad j = 1,\ldots,\bar{N}, \tag{3.32}$$

$$g_j(\boldsymbol{x}) = \iint \hat{g}_j(\boldsymbol{x}) \mathrm{d}x \mathrm{d}x, \quad j = 1,\ldots,\bar{N}, \tag{3.33}$$

in which $\varphi(||\boldsymbol{x}-\boldsymbol{x}_j||)$ are radial basis functions such as Hardy's multiquadrics or Duchon's thin-plate splines.

For a more compact form, the IRBFN formulation can be written as follows.

$$u_{xx}(\boldsymbol{x},t) \equiv \frac{\partial^2 u(\boldsymbol{x},t)}{\partial x^2} \approx \boldsymbol{\psi}_{\partial_{xx}}(\boldsymbol{x})^T \boldsymbol{u}(t), \tag{3.34}$$

$$u_x(\boldsymbol{x},t) \equiv \frac{\partial u(\boldsymbol{x},t)}{\partial x} \approx \boldsymbol{\psi}_{\partial_x}(\boldsymbol{x})^T \boldsymbol{u}(t), \tag{3.35}$$

$$u(\boldsymbol{x},t) \approx \boldsymbol{\psi}(\boldsymbol{x})^T \boldsymbol{u}(t), \tag{3.36}$$

where

$$\boldsymbol{\psi}_{\partial_{xx}}(\boldsymbol{x}) = \hat{\boldsymbol{g}}(\boldsymbol{x})^T \mathbf{G}^{-1}, \tag{3.37}$$

$$\boldsymbol{\psi}_{\partial_{x}}(\boldsymbol{x}) = \tilde{\boldsymbol{g}}(\boldsymbol{x})^T \mathbf{G}^{-1}, \tag{3.38}$$

$$\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x})^T \mathbf{G}^{-1}. \tag{3.39}$$

Let $\mathcal{S}$ be a certain differential operator in space that operates on the scalar function $u(\boldsymbol{x}, t)$ in $\Omega \in \mathbb{R}^d,\ d = 1, 2, 3$, then the IRBFN formulation above can be rewritten in the generic form for approximating function $u(\boldsymbol{x}, t)$ and/or its derivatives as follows.

$$\mathcal{S}u(\boldsymbol{x}, t) \approx \boldsymbol{\psi}_{\mathcal{S}}^T(\boldsymbol{x})\boldsymbol{u}(t), \tag{3.40}$$

where $\boldsymbol{\psi}_{\mathcal{S}}(\boldsymbol{x})$ is the vector whose components are the results of the application of operator $\mathcal{S}$ on the corresponding components of $\boldsymbol{\psi}(\boldsymbol{x})$,

$$\boldsymbol{\psi}_{\mathcal{S}}(\boldsymbol{x}) = [\mathcal{S}\psi_1(\boldsymbol{x}), \mathcal{S}\psi_2(\boldsymbol{x}), \ldots, \mathcal{S}\psi_M(\boldsymbol{x})]^T. \tag{3.41}$$

For a special case where $\mathcal{S}$ is the identity operator, $\mathcal{S} = \mathcal{I}$, one gets the approximation of function $u(\boldsymbol{x}, t)$. Otherwise, one obtains the corresponding derivative of the function. For example, if $\mathcal{S} = \frac{\partial}{\partial y} \equiv \partial_y$, one has the approximation of the first order derivative of $u(x, y, t)$ in y direction as follows.

$$\mathcal{S}u(x, y, t) = \frac{\partial}{\partial y}u(x, y, t) \approx \boldsymbol{\psi}_{\partial_y}^T(x, y)\boldsymbol{u}(t). \tag{3.42}$$

It is noted that in the case of the DRBFN method, the RBF must be sufficiently differentiable to satisfy the particular differential operator. The MQ is, therefore, always applicable as it is C^∞ continuous, whereas the TPS is only C^{2m-1} continuous, and m must be chosen appropriately to suit the differential operator. The IRBFN method, on the other hand, can use the first order TPS

($m = 1$) regardless of the order of the differential operator because the method starts from the highest derivative and the approximation of the corresponding function is obtained via successive integrations. The formulations of the first and second order antiderivatives of Hardy's multiquadrics and Duchon's thin plate splines are presented in Appendices A and B, respectively. Another note to be mentioned here is that, although the MQ and TPS are used with up to second order derivatives of a function in this work, the IRBFN method can be used with other kinds of RBFs and/or with higher order derivatives provided that symbolic integrations can be performed.

3.3 IRBFN-based Schemes for Time-Dependent PDEs

3.3.1 Problem statement

Consider a general two-dimensional initial boundary value problem

$$\frac{\partial u}{\partial t} + \mathcal{S}u = q \quad \text{in } Q_T := (0,T) \times \Omega,$$

subject to boundary and initial conditions

$$\mathcal{B}u = r \quad \text{on } \Sigma_T := (0,T) \times \partial\Omega, \tag{3.43}$$

$$u = u^0 \quad \text{on } \Omega, \text{ at } t = 0,$$

where Ω is a bounded domain in $\mathbb{R}^2$, with boundary $\partial\Omega$; $T > 0$ is a prescribed time-level; $q = q(x, y, t)$, $r = r(x, y, t)$ and $u^0 = u^0(x, y)$ are given functions. In the differential equation above, $\mathcal{S}$ is the differential operator in space which can

be one of the followings

$$\textit{Advection:} \quad \mathcal{S} \equiv \mathbf{v} \cdot \nabla, \tag{3.44}$$

$$\textit{Diffusion:} \quad \mathcal{S} \equiv -\nu \nabla^2, \tag{3.45}$$

$$\textit{Advection-Diffusion:} \quad \mathcal{S} \equiv \mathbf{v} \cdot \nabla - \nu \nabla^2, \tag{3.46}$$

where $\mathbf{v}$ is the given velocity field and ν is a given scalar. The operator $\mathcal{B}$ represents Dirichlet, Neumann or Robin boundary conditions as follows.

$$\textit{Dirichlet:} \quad \mathcal{B} \equiv \mathcal{I} \tag{3.47}$$

$$\textit{Neumann:} \quad \mathcal{B} \equiv \mathbf{n} \cdot \nabla \tag{3.48}$$

$$\textit{Robin:} \quad \mathcal{B} \equiv \mathbf{n} \cdot \nabla + \kappa \mathcal{I}, \tag{3.49}$$

in which $\mathcal{I}$ is the identity operator, $\mathbf{n}$ is the outward unit vector normal to the boundary, and κ is a given scalar on the boundary.

3.3.2 Fully discrete schemes

In fully discrete schemes, problem (3.43) is discretized with respect to both time and space variables using the horizontal method of lines as presented in Section (2.2.2). Firstly, the time interval $[0, T]$ is partitioned into N_T subintervals $[t^n,\, t^{n+1}]$ of length $\Delta t = T/N_T$ with $t^0 = 0$ and $t^{N_T+1} = T$. The discretization of the problem in time is then accomplished by a time-stepping scheme, followed by the spatial discretization based on the IRBFN method.

Among many possible time-stepping schemes, the standard θ-scheme (Quarteroni and Valli, 1997), $0 \leq \theta \leq 1$ is used in this work. It should be noted that the extreme cases $\theta = 0$ and $\theta = 1$ correspond to the well-known forward (fully explicit) and backward (fully implicit) Euler methods, respectively. The

scheme associated with the case $\theta = 1/2$ is equivalent to the (semi-implicit) Crank-Nicolson method which is second-order accurate. Applying the θ-scheme to problem (3.43) gives

$$\frac{\tilde{u}^{n+1} - \tilde{u}^n}{\Delta t} + \theta\, \mathcal{S}\tilde{u}^{n+1} + (1-\theta)\, \mathcal{S}\tilde{u}^n = \theta\, q^{n+1} + (1-\theta) q^n, \tag{3.50}$$

$$\mathcal{B}\tilde{u}^{n+1} = r^{n+1}, \tag{3.51}$$

where $t^{n+1} = t^n + \Delta t$, $\tilde{u}^{n+1} \approx u(x, y, t^{n+1})$, and $\tilde{u}^0 = u_0(x, y)$. The time discrete system (3.50)-(3.51) is then discretized in space by using the IRBFN method in (3.34)-(3.36) at time t^{n+1} and applied at every collocation point to obtain a system of equations to be solved for $\boldsymbol{u}^{n+1}$. Note that the domain is partitioned into M_I interior points and $M - M_I$ boundary points such that

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_I \\ \boldsymbol{u}_B \end{pmatrix} \tag{3.52}$$

The fully discrete version of (3.50) reads

$$\begin{aligned} \boldsymbol{\psi}^T(\boldsymbol{x}_i)\boldsymbol{u}_I^{n+1} + \theta\, \Delta t\, \boldsymbol{\psi}_{\mathcal{S}}^T(\boldsymbol{x}_i)\, \boldsymbol{u}_I^{n+1} &= \boldsymbol{\psi}^T(\boldsymbol{x}_i)\, \boldsymbol{u}_I^n - (1-\theta)\, \Delta t\, \boldsymbol{\psi}_{\mathcal{S}}^T(\boldsymbol{x}_i)\, \boldsymbol{u}_I^n \\ &\quad + \Delta t \left[\theta\, q^{n+1}(\boldsymbol{x}_i) + (1-\theta) q^n(\boldsymbol{x}_i)\right], \quad i = 1, \ldots, M_I, \end{aligned}$$

or

$$\begin{aligned} \left[\boldsymbol{\psi}^T(\boldsymbol{x}_i) + \theta\, \Delta t\, \boldsymbol{\psi}_{\mathcal{S}}^T(\boldsymbol{x}_i)\right]\, \boldsymbol{u}_I^{n+1} &= \boldsymbol{\psi}^T(\boldsymbol{x}_i)\, \boldsymbol{u}_I^n - (1-\theta)\, \Delta t\, \boldsymbol{\psi}_{\mathcal{S}}^T(\boldsymbol{x}_i)\, \boldsymbol{u}_I^n \\ &\quad + \Delta t\, \left[\theta\, q^{n+1}(\boldsymbol{x}_i) + (1-\theta) q^n(\boldsymbol{x}_i)\right], \quad i = 1, \ldots, M_I, \end{aligned} \tag{3.53}$$

where

$$\boldsymbol{\psi}_{\mathcal{S}}(\boldsymbol{x}_i) = [\mathcal{S}\psi_1(\boldsymbol{x}_i), \mathcal{S}\psi_2(\boldsymbol{x}_i), \ldots, \mathcal{S}\psi_{M_I}(\boldsymbol{x}_i)]^T, \quad i = 1, \ldots, M_I. \tag{3.54}$$

Similarly, the fully discrete versions of the boundary condition (3.51) can be

written in a general form

$$\boldsymbol{\psi}_{\mathcal{B}}^{T}(\boldsymbol{x}_i)\,\boldsymbol{u}_B^{n+1} = r(\boldsymbol{x}_i), \quad i = M_I + 1, \ldots, M, \tag{3.55}$$

where

$$\boldsymbol{\psi}_{\mathcal{B}}(\boldsymbol{x}_i) = [\mathcal{B}\psi_{M_I+1}(\boldsymbol{x}_i), \mathcal{B}\psi_{M_I+2}(\boldsymbol{x}_i), \ldots, \mathcal{B}\psi_{M}(\boldsymbol{x}_i)]^T, \quad i = M_I + 1, \ldots, M, \tag{3.56}$$

in which operator $\mathcal{B}$ is defined according to either (3.47), (3.48), or (3.49).

System (3.53) and (3.55) can be written in matrix form as follows.

$$\mathbf{A}\,\boldsymbol{u}^{n+1} = \mathbf{b}, \tag{3.57}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} + \theta\,\Delta t\,\boldsymbol{\Psi}_{\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_{\mathcal{B}} \end{bmatrix},$$

and

$$\mathbf{b} = \begin{bmatrix} \boldsymbol{u}_I^n - (1-\theta)\,\Delta t\,\boldsymbol{\Psi}_{\mathcal{S}}\,\boldsymbol{u}_I^n + \Delta t\,\left(\theta\,\boldsymbol{q}^{n+1} + (1-\theta)\boldsymbol{q}^n\right) \\ \boldsymbol{r} \end{bmatrix},$$

where $\boldsymbol{\Psi}_{\mathcal{S}}$ and $\boldsymbol{\Psi}_{\mathcal{B}}$ are the matrix representations of (3.54) and (3.56), respectively. The system (3.57) is solved at each time step for $\boldsymbol{u}^{n+1}$ until the prescribed time T is reached using the corresponding nodal values at the previous time $t = t^n$, $\boldsymbol{u}^n$, and the IRBFN coefficient matrices associated with function u. It is noted that an alternative system of equations in terms of the weight coefficients $\boldsymbol{w}$ can be set up by minimizing the sum-squared error (SSE) in the sense of the general least-squares principle (Mai-Duy and Tran-Cong, 2001a). In practice, however, the normal equation resulting from the SSE approach would raise the

condition number of the system matrix to the second power making it more ill-conditioned and difficult to solve. In addition, boundary conditions are not strictly satisfied within the least-squares context. As can be seen previously, the system (3.57) has been set up without using the SSE approach. The IRBFN fully discrete scheme was reported in (Mai-Cao and Tran-Cong, 2005).

3.3.3 Semi-discrete schemes

In semi-discrete schemes, problem (3.43) is first discretized in space while still continuously dependent on time. In other words, the schemes are derived on the basis of the vertical method of lines as presented in Section (2.2.2). By substituting (3.39) into (3.43), one obtains

$$\boldsymbol{\psi}^T(\boldsymbol{x})\frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} + \boldsymbol{\psi}_{\mathcal{S}}^T(\boldsymbol{x})\boldsymbol{u}(t) = q(\boldsymbol{x}), \tag{3.58}$$

$$\boldsymbol{\psi}_{\mathcal{B}}^T(\boldsymbol{x})\boldsymbol{u}(t) = r(\boldsymbol{x}). \tag{3.59}$$

Applying (3.58)-(3.59) at every collocation point then gives a system of ordinary differential equations (ODEs)

$$\boldsymbol{\psi}^T(\boldsymbol{x}_i)\frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} + \boldsymbol{\psi}_{\mathcal{S}}^T(\boldsymbol{x}_i)\boldsymbol{u}(t) = q(\boldsymbol{x}_i), \quad i = 1, \ldots, M_I, \tag{3.60}$$

$$\boldsymbol{\psi}_{\mathcal{B}}^T(\boldsymbol{x}_i)\boldsymbol{u}(t) = r(\boldsymbol{x}_i), \quad i = M_I + 1, \ldots, M, \tag{3.61}$$

where $\boldsymbol{\psi}_{\mathcal{S}}$ and $\boldsymbol{\psi}_{\mathcal{B}}$ are defined in (3.54) and (3.56), respectively. The above system can be written in the matrix form as follows.

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{\Psi}_{\mathcal{S}}\,\boldsymbol{u} = \mathbf{q}, \tag{3.62}$$

$$\boldsymbol{\Psi}_{\mathcal{B}}\,\boldsymbol{u} = \mathbf{r}, \tag{3.63}$$

where $\boldsymbol{\Psi}_{\mathcal{S}}$ and $\boldsymbol{\Psi}_{\mathcal{B}}$ are the matrix representations of (3.54) and (3.56), respectively. Solving the ODE system (3.62)-(3.63) for $\boldsymbol{u}$ with the initial conditions

$$\boldsymbol{\psi}^T(\boldsymbol{x}_i)\,\boldsymbol{u}(t=0) = u^0(\boldsymbol{x}_i), \quad i = 1, 2, ..., M,$$

yields the solution to the problem (3.43) at every data point within the time interval of interest. Various high-order ODE solvers presented in section (2.2.2) such as fourth-order Runge-Kutta, Adams-Bashforth (AB), Backward Differentiation Formulation (BDF) method can be applied to solve the system. The IRBFN semi-discrete scheme was reported in (Mai-Cao and Tran-Cong, 2005).

3.4 Numerical Results

For verification purposes, the IRBFN-based schemes presented in Section (3.3.2) and (3.3.3) are applied to solve five example problems namely the 1D and 2D diffusion equations, the 1D and 2D wave equations, and the 1D advection-diffusion equation. The results are then compared in terms of accuracy and efficiency to those from other methods such as finite difference, finite element, boundary element and the DRBFN methods. Furthermore, three network parameters including the user-defined parameter β described in (2.35), the point density and the time-step size are taken into account to investigate their influence on the accuracy of the solutions by the MQ-IRBFN method. Absolute/Norm errors at each time step and root-mean-squared-error throughout the time domain are used for accuracy comparison and sensitivity analysis. The root-mean-squared error is calculated by

$$RMSE = \sqrt{\frac{1}{N_T}\sum_{i=1}^{N_T}(u_i - \tilde{u}_i)^2}, \tag{3.64}$$

where $\tilde{u}$ and u are numerical and analytical solutions, respectively; N_T is the total number of time steps.

3.4.1 Test problem 1. One-dimensional heat equation

Two test problems are considered in this section where the numerical solutions by the present IRBFN method are compared to those by the FDM and the DRBFN method.

Test problem 1.1

Consider an 1D heat equation

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < 1, \quad t > 0,$$

subject to boundary and initial conditions

$$u(0,t) = u(1,t) = 0, \quad t > 0,$$

$$u(x,0) = \sin \pi x, \quad 0 \leq x \leq 1,$$

where $u(x,t)$ is the temperature at the position x at time t. The problem has the analytical solution $u(x,t) = e^{-\pi^2 t} \sin \pi x$.

The finite difference method (FDM) and the IRBFN method are used to solve the problem. For the FDM, Crank-Nicolson algorithm is used whereas the fully discrete scheme described in section (3.3.2) with θ=0.5 is used for the IRBFN method. The numerical solution by the IRBFN method is shown in Figure 3.1. Table 3.1 shows that the IRBFN method using either MQ or TPS RBFs gives more accurate result than the FDM with the same number of points $N = 11$ and the time step size $\Delta t = 0.01$. In particular, the present method yields the solution accurate to 4 significant digits with TPS and up to 5 significant digits with MQ. In Table 3.2, the IRBFN method shows its superiority over the FDM when the point density and the number of time steps used are only half as many as those used in the FDM. With such a "crude" discretization in both time and space domain, the IRBFN method still yields solutions more accurate than that of the FDM.

Table 3.1: Accuracy comparison of the solution at time $t = 0.5$ between the FDM and the IRBFN method (using either MQ or TPS) in Test problem 1.1. Both methods use the same discretization ($N = 11$, $\Delta t = 0.01$). The L_∞-norm of the error vector for FDM, TPS-IRBFN and MQ-IRBFN are $2.6766e-4$, $1.6887e-5$ and $8.3494e-6$, respectively.

		Numerical solution			Absolute Error		
X	Exact	FDM	TPS	MQ	FDM	TPS	MQ
0.1	0.002222	0.002305	0.002219	0.002224	8.271e-05	3.341e-06	1.908e-06
0.2	0.004227	0.004385	0.004218	0.004226	1.573e-04	8.856e-06	9.966e-07
0.3	0.005818	0.006035	0.005805	0.005815	2.165e-04	1.311e-05	3.066e-06
0.4	0.006840	0.007094	0.006824	0.006835	2.546e-04	1.591e-05	4.544e-06
0.5	0.007192	0.007460	0.007175	0.007187	2.677e-04	1.689e-05	5.035e-06
0.6	0.006840	0.007094	0.006824	0.006835	2.546e-04	1.592e-05	4.543e-06
0.7	0.005818	0.006035	0.005805	0.005815	2.165e-04	1.311e-05	3.064e-06
0.8	0.004227	0.004385	0.004218	0.004226	1.573e-04	8.861e-06	9.937e-07
0.9	0.002222	0.002305	0.002219	0.002224	8.271e-05	3.348e-06	1.912e-06

For the purpose of sensitivity studies for the MQ-IRBFN method, four point densities are chosen to be 5, 7, 9, 11 points, and the numbers of time steps of interest are 25 and 50. In addition, a wide range of β's values from 1 to 10 are used to study its influence on the solution accuracy. Figures 3.2a, 3.2b, 3.2c, 3.2d show the maximum absolute errors by the MQ-IRBFN method over the entire set of collocation points, and figures 3.3a, 3.3b, 3.3c, 3.3d present the root-mean-squared errors over the time domain of 25 time steps. It can be seen from the figures that the errors of the proposed method do not vary significantly for this problem where β's are in the range from 1 to 10. In particular, these errors are of the same order of magnitude. In addition, a "mesh" (or discretization) convergence can be observed from figures 3.2b, 3.2d and 3.3b, 3.3d where the accuracy of the numerical solution does not improve significantly when the number of points nx increases from 7 to 11.

Table 3.2: Accuracy comparison of the solution at time $t = 0.5$ between the FDM and the IRBFN method (using either MQ or TPS) in Test problem 1.1. Although the number of collocation points and time steps the IRBFN method needs (6 and 25) are only half as many as the FDM uses (11 and 50), the IRBFN method still yields more accurate solution. The L_∞-norm of the error vector for FDM, TPS-IRBFN and MQ-IRBFN are $2.5455e-4$, $4.3845e-5$ and $1.8211e-5$, respectively.

		Numerical solution			Absolute Error		
X	Exact	FDM	TPS	MQ	FDM	TPS	MQ
0.1	0.002222	0.002305	-	-	8.271e-05	-	-
0.2	0.004227	0.004385	0.004271	0.004245	1.573e-04	4.362e-05	1.821e-05
0.3	0.005818	0.006035	-	-	2.165e-04	-	-
0.4	0.006840	0.007094	0.006884	0.006852	2.546e-04	4.384e-05	1.176e-05
0.5	0.007192	0.007460	-	-	2.677e-04	-	-
0.6	0.006840	0.007094	0.006884	0.006852	2.546e-04	4.384e-05	1.176e-05
0.7	0.005818	0.006035	-	-	2.165e-04	-	-
0.8	0.004227	0.004385	0.004271	0.004245	1.573e-04	4.361e-05	1.821e-05
0.9	0.002222	0.002305	-	-	8.271e-05	-	-

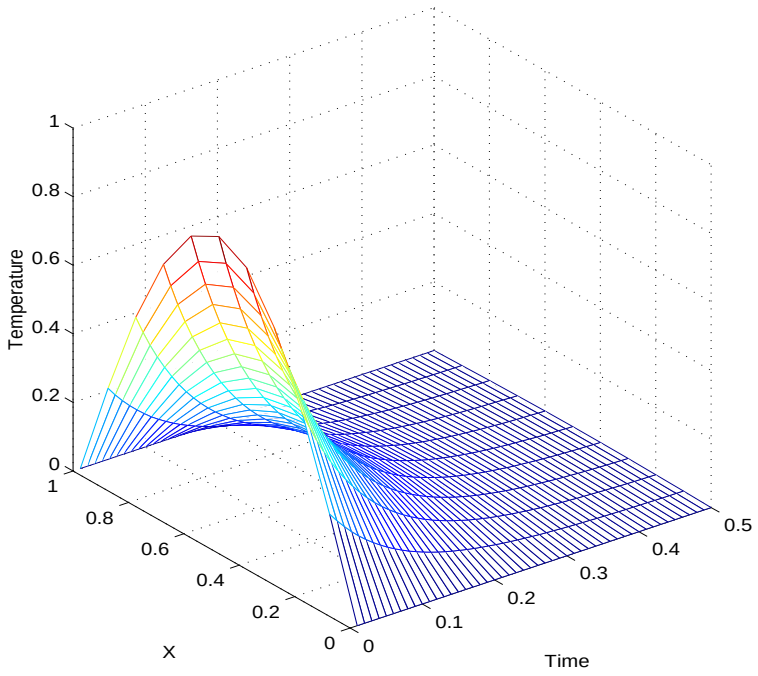

Figure 3.1: Solution profile in Test problem 1.1 by the MQ-IRBFN method.

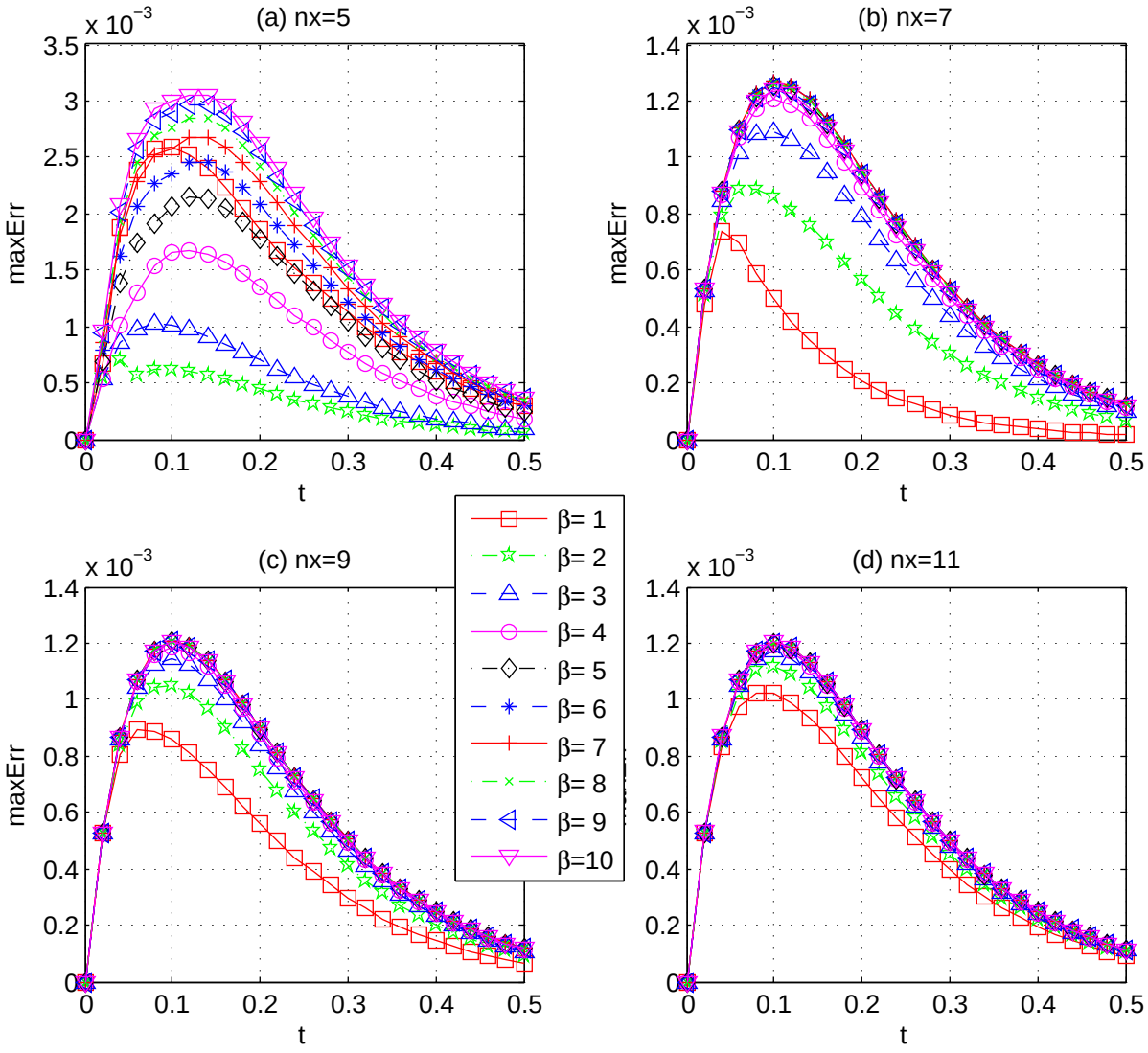

Figure 3.2: Sensitivity analysis of the MQ-IRBFN method with respect to the network parameters in Test problem 1.1 for the case where the number of time steps = 25. Figures (a),(b),(c),(d) show the maximum absolute error over the entire set of collocation points

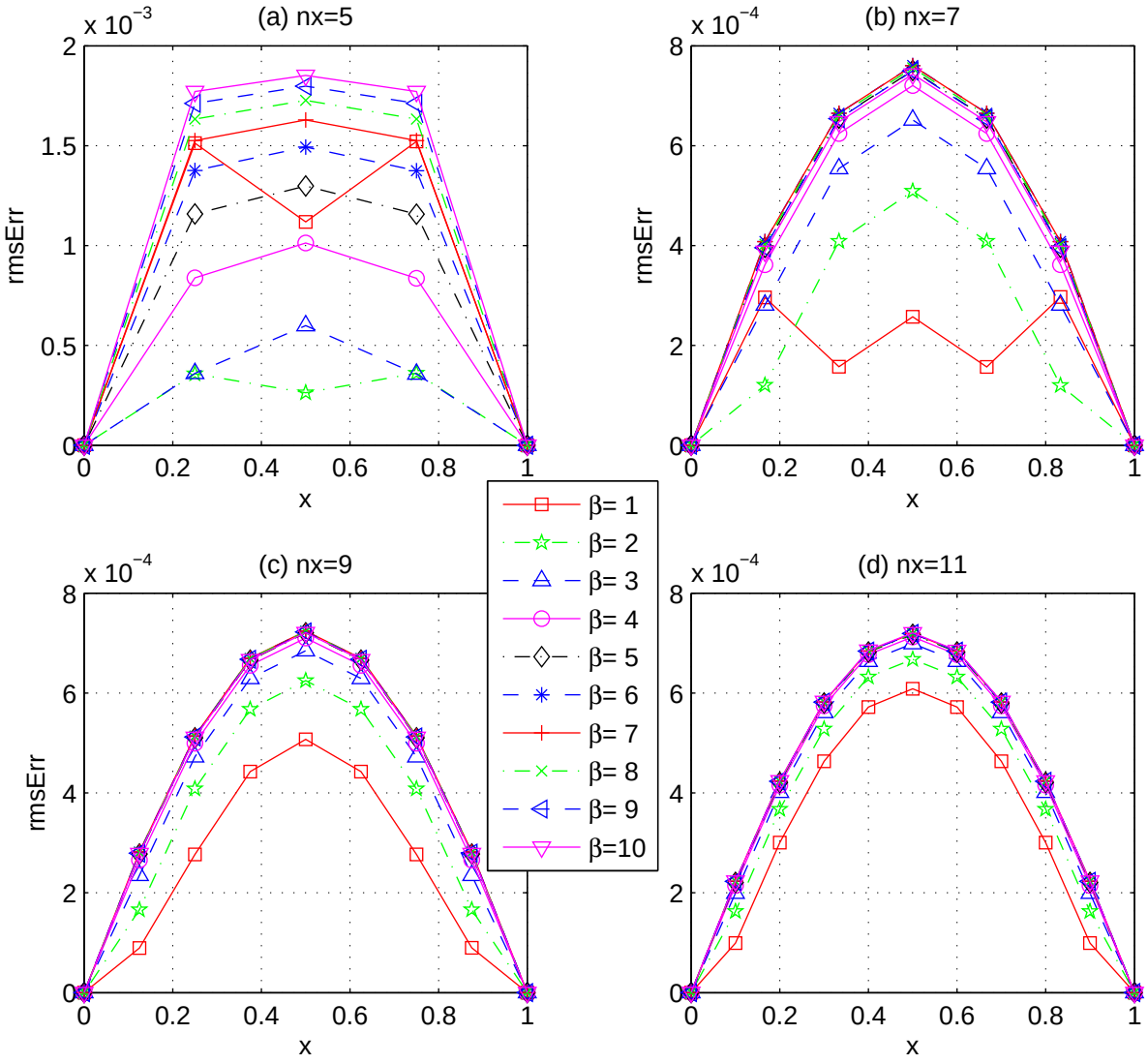

Figure 3.3: Sensitivity analysis of the MQ-IRBFN method with respect to the network parameters in Test problem 1.1 for the case where the number of time steps = 25. Figures (a),(b),(c),(d) show the root-mean-squared errors over the whole time domain.

Test problem 1.2

Consider the one-dimensional heat equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1,\ t > 0,$$

subject to Dirichlet boundary conditions

$$u(0,t) = u(1,t) = 0, \quad t \geq 0,$$

and initial conditions

$$u(\boldsymbol{x},0) = \begin{cases} \mu\, x, & 0 \leq x \leq 0.5, \\ \mu\,(1-x), & 0.5 \leq x \leq 1. \end{cases}$$

The analytical solution to the problem is given by

$$u(\boldsymbol{x},t) = \frac{4\,\mu}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \sin\frac{n\pi x}{2}\, e^{-\kappa t n^2 \pi^2}.$$

In this test, the problem is solved by both DRBFN and IRBFN methods. The solution by the DRBFN method is obtained from (Zerroukat et al., 1998) where the explicit scheme, namely EEX-MQ, was used. The scheme used explicit exponential formulation with multiquadrics as RBFs and was reported to yield the most accurate results among others. For the IRBFN method, a semi-discrete scheme based on Adams predictor-corrector method is used with either MQ or TPS to solve the problem. Figure 3.4 shows the solution profile of the problem given by the MQ-IRBFN method. Solutions at time $t = 1$ by both methods are shown in Tables 3.3 and 3.4. As can be seen from the tables, with the same number of collocation points, the IRBFN method using either MQ or TPS outperforms the DRBFN method using MQ while the number of time steps it uses ($Nts = 10$) is only one-tenth of that required by the DRBFN method ($Nts = 100$). In addition, the solutions by the IRBFN method near the boundary points are of high accuracy without any special treatment.

Table 3.3: Test problem 1.2: Accuracy comparison between the DRBFN method using explicit exponential scheme with MQ (Zerroukat et al., 1998) and the IRBFN using semi-discrete scheme with MQ. With the same number of collocation points, the IRBFN method yields more accurate solution while the number of time steps it uses ($Nts = 10$) is one-tenth of that required by the DRBFN method ($Nts = 100$). The L_∞-norm of the error vector for MQ-DRBFN and MQ-IRBFN methods are $5.6000e-4$ and $5.2220e-5$, respectively.

		Numerical solution		Absolute Error	
X	Exact	MQ-DRBFN	MQ-IRBFN	MQ-DRBFN	MQ-IRBFN
0.10	0.00483	0.00458	0.00484	2.5000e-004	1.0235e-005
0.30	0.01265	0.01218	0.01262	4.7000e-004	2.5465e-005
0.50	0.01564	0.01508	0.01560	5.6000e-004	4.2446e-005
0.60	0.01488	0.01434	0.01483	5.4000e-004	5.2220e-005
0.80	0.00919	0.00882	0.00918	3.7000e-004	1.2937e-005

3.4.2 Test problem 2. Two-dimensional diffusion equation

Consider the 2D diffusion equation

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

to be solved in the domain $0 \leq x \leq 1,\ 0 \leq y \leq 1$. The forcing function is given by

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

The initial and boundary conditions are appropriate to the analytical solution

$$u = \sin x \sin y \sin t.$$

In this example, the problem is solved by the BEM (Ingber and Phan-Thien, 1992) and the present IRBFN method. For the BEM, two numerical schemes, namely $M1$ and $M2$, were used where "generalized" forcing functions can be approximated by radial basis functions. For the IRBFN method, the fully

Table 3.4: Test problem 1.2: Accuracy comparison between the DRBFN method using explicit exponential scheme with MQ (Zerroukat et al., 1998) and the IRBFN using semi-discrete scheme with TPS. Although the number of time steps it uses ($Nts = 10$) is one-tenth of that required by the DRBFN method ($Nts = 100$), still the IRBFN method yields more accurate solution. The two methods use the same number of collocation points. The L_∞-norm of the error vector for MQ-DRBFN and TPS-IRBFN are $5.6000e-4$ and $1.6953e-4$, respectively.

		Numerical solution		Absolute Error	
X	Exact	MQ-DRBFN	TPS-IRBFN	MQ-DRBFN	TPS-IRBFN
0.10	0.00483	0.00458	0.00489	2.5000e-004	6.0989e-005
0.30	0.01265	0.01218	0.01275	4.7000e-004	1.0490e-004
0.50	0.01564	0.01508	0.01576	5.6000e-004	1.1617e-004
0.60	0.01488	0.01434	0.01505	5.4000e-004	1.6953e-004
0.80	0.00919	0.00882	0.00929	3.7000e-004	1.0127e-004

discrete scheme presented in section 3.3.2 is used with $\theta = 0.5$. For the purpose of comparing the two methods, the solution to the problem at point (0.8,0.8) throughout the whole time domain is considered. The solution by the BEM is obtained by scheme $M2$ (which was reported to be slightly more accurate than scheme $M1$) with a fine mesh of 17×17 boundary points and 9×9 interior points. As the IRBFN method uses a set of randomly generated points, the solution at the point of interest can be post-processed by simple function evaluation based on (3.30). Figure 3.5.a presents the set of randomly generated collocation points, and Figure 3.5.b shows the solution at point (0.8,0.8) by the IRBFN method. The results in this example show that the present IRBFN method yields accurate solution not only at grid points but also at arbitrary points in the domain. Table 3.5 shows that the IRBFN method using either MQ or TPS outperforms the BEM in terms of accuracy and efficiency. As can be seen from the table, the IRBFN method yields more accurate result even with a coarser discretization (12 boundary points and 13 interior points) than the BEM does with a much finer mesh (289 boundary points and 81 interior points). Furthermore, Figure 3.5(b) shows that the IRBFN method maintains its higher accuracy compared to the BEM throughout the whole time domain.

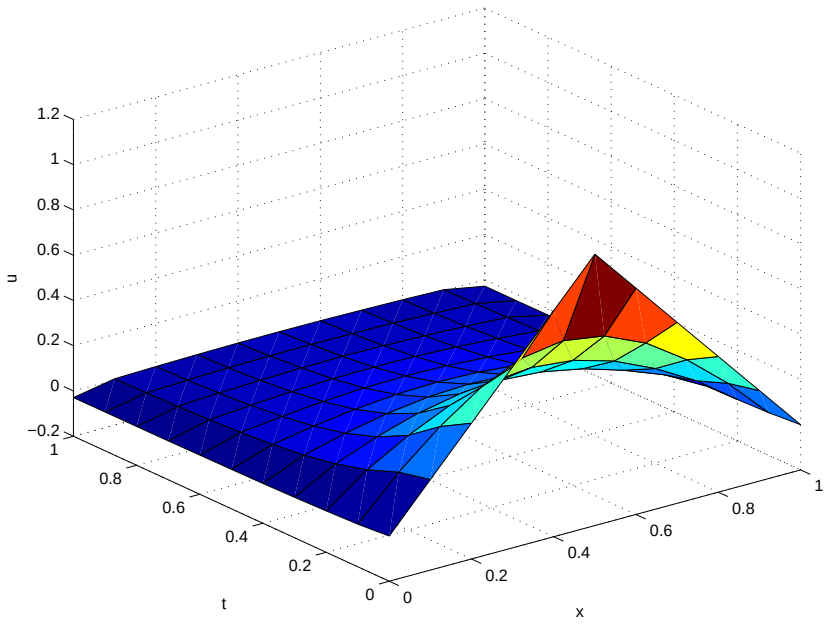

Figure 3.4: Numerical solution given by the MQ-IRBFN method for Test problem 1.2.

In this example, both methods use the same time step size Δt=0.25.

For the purpose of studying the sensitivity of the MQ-IRBFN method to the network parameters, four regular point densities (5×5), (7×7), (9×9) and (11×11) are used together with a set of β's values ranging from 1 to 10 and two different numbers of time steps (16 and 32). Figure 3.6 shows that the maximum absolute errors by the IRBFN method do not change significantly within a range of β's values from 1 to 10. Furthermore, it can be seen from the figures that while "mesh" convergence can be observed at a very coarse point density, the accuracy of the solution can be further improved by increasing the number of time steps. In this particular example, as the number of collocation points increases from Figure 3.6(a) or Figure 3.7(c) to more than 2 times denser in Figure 3.6(b) or Figure 3.7(d), the maximum errors do not decrease noticeably, indicating "mesh" convergence. However, as the time step size decreases from Figure 3.6(a)(b) to half the value in Figure 3.7(c)(d), the errors decrease by an order of magnitude. It can also be seen from Figures (3.8)-(3.9) that root-mean-squared errors by the IRBFN method in Test problem 2 do not vary significantly

Table 3.5: Solution to Test problem 2 at the interior point $x = 0.8$, $y = 0.8$. With a coarser point density (12 boundary points and 13 interior points), the IRBFN method (based on either MQ or TPS) still yields more accurate result compared to the BEM (Ingber and Phan-Thien, 1992) using a much finer mesh (289 boundary points and 81 interior points). Both methods use the same time step size Δt=0.25. The L_∞-norm of the error vector for BEM, TPS-IRBFN and MQ-IRBFN are $3.9589e-3$, $1.5474e-4$ and $1.1788e-4$, respectively.

		Numerical solution			Relative Error		
T	Exact	BEM	TPS	MQ	BEM	TPS	MQ
0.3	0.127314	0.126800	0.127159	0.127196	4.037e-03	1.215e-03	9.259e-04
0.5	0.246712	0.245300	0.246626	0.246655	5.724e-03	3.497e-04	2.326e-04
0.8	0.350771	0.348500	0.350624	0.350664	6.475e-03	4.187e-04	3.059e-04
1.0	0.433021	0.430000	0.432912	0.432952	6.976e-03	2.512e-04	1.583e-04
1.3	0.488347	0.484800	0.488221	0.488260	7.264e-03	2.584e-04	1.796e-04
1.5	0.513311	0.509400	0.513220	0.513258	7.619e-03	1.776e-04	1.027e-04
1.8	0.506359	0.502400	0.506276	0.506306	7.818e-03	1.639e-04	1.053e-04
2.0	0.467924	0.464100	0.467878	0.467905	8.173e-03	9.778e-05	4.108e-05
2.3	0.400396	0.397000	0.400373	0.400387	8.482e-03	5.801e-05	2.351e-05
2.5	0.307974	0.305200	0.307987	0.307995	9.006e-03	4.404e-05	7.055e-05
2.8	0.196403	0.194400	0.196443	0.196437	1.020e-02	2.049e-04	1.761e-04
3.0	0.072620	0.071500	0.072691	0.072679	1.543e-02	9.775e-04	8.141e-04
3.3	-0.055677	-0.055800	-0.055584	-0.055608	2.206e-03	1.667e-03	1.234e-03
3.5	-0.180513	-0.179700	-0.180401	-0.180430	4.504e-03	6.193e-04	4.609e-04
3.8	-0.294125	-0.292300	-0.294003	-0.294040	6.206e-03	4.152e-04	2.891e-04
4.0	-0.389450	-0.386800	-0.389325	-0.389363	6.805e-03	3.226e-04	2.251e-04

within a range of β's values from 1 to 10 for the case where the number of time steps Nts is equal to 16 as shown in Figure 3.8(a)-(g), and in Figure 3.9(h)-(n) where Nts is equal to 32.

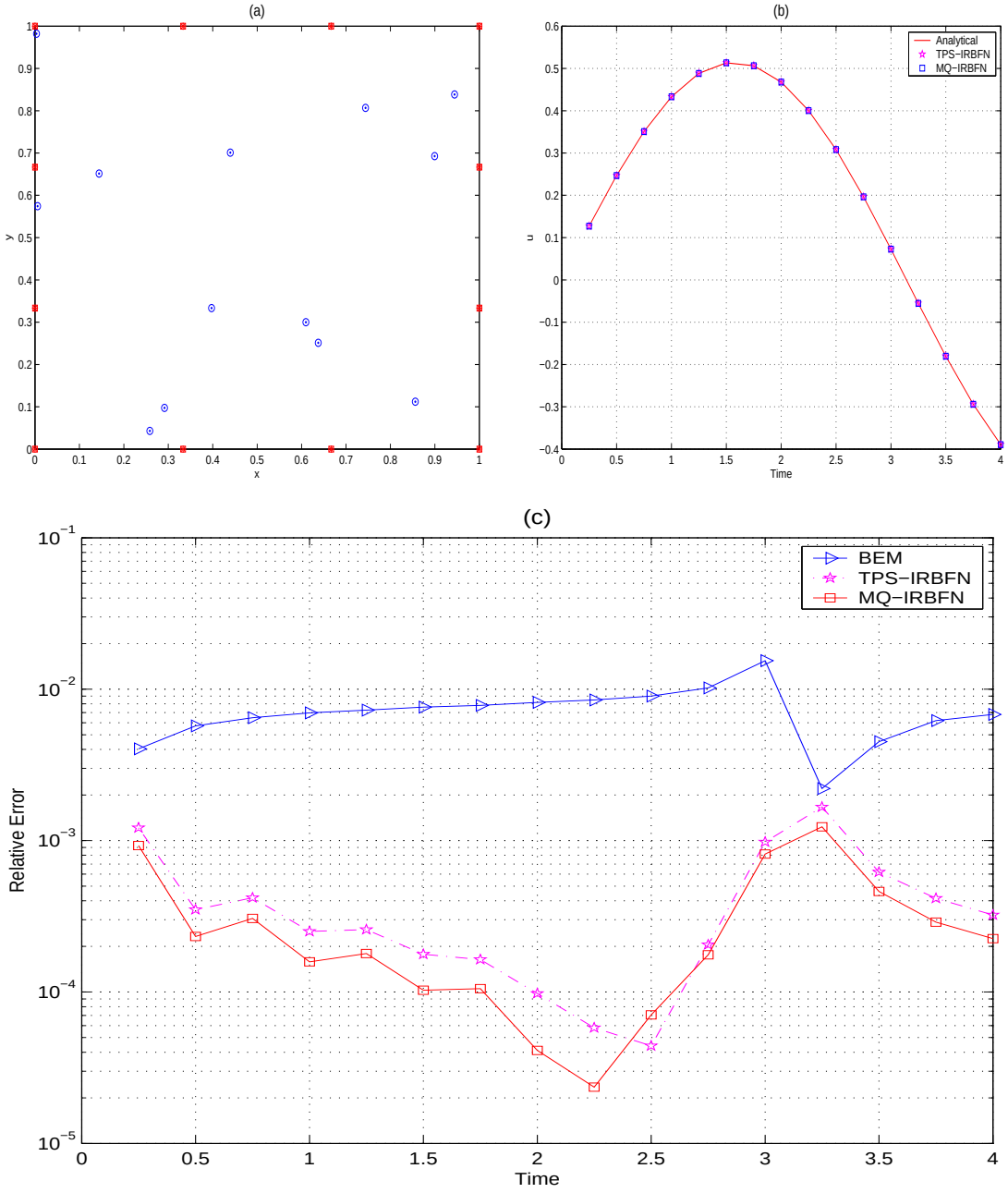

Figure 3.5: (a) A set of randomly generated points used by the IRBFN method; (b) Solution at point (0.8,0.8) by the IRBFN method; (c) An accuracy comparison between the IRBFN method and the BEM (Ingber and Phan-Thien, 1992) in Test problem 2. Using a coarser point density, the IRBFN method maintains its higher accuracy compared to the BEM throughout the whole time domain.

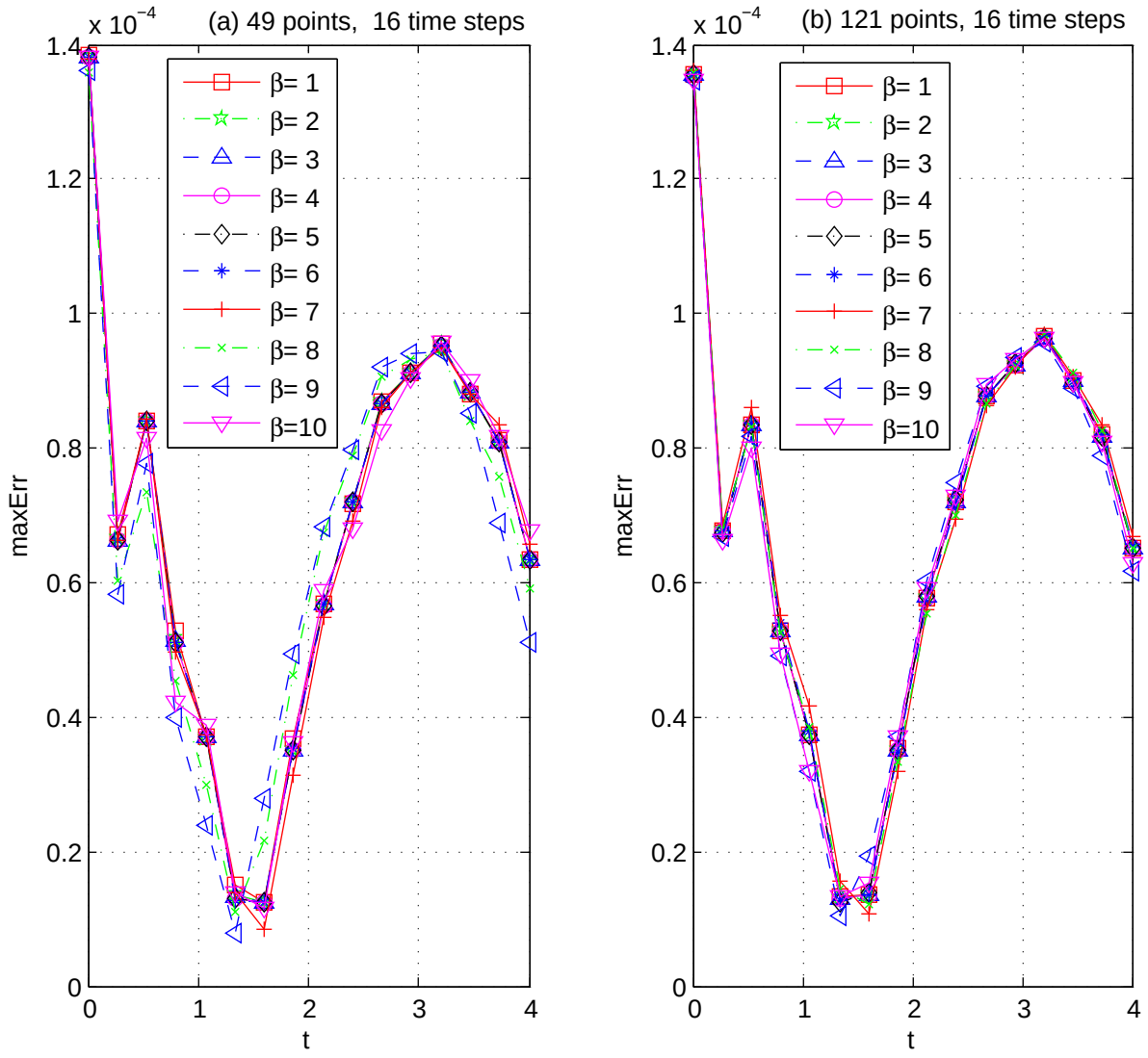

Figure 3.6: In Test problem 2, as the point density increases from figure (a) to more than 2 times denser in figure (b), the maximum errors by the MQ-IRBFN method do not decrease noticeably, indicating "mesh" convergence. In addition, the errors by the MQ-IRBFN method do not vary significantly within the range of β's values from 1 to 10.

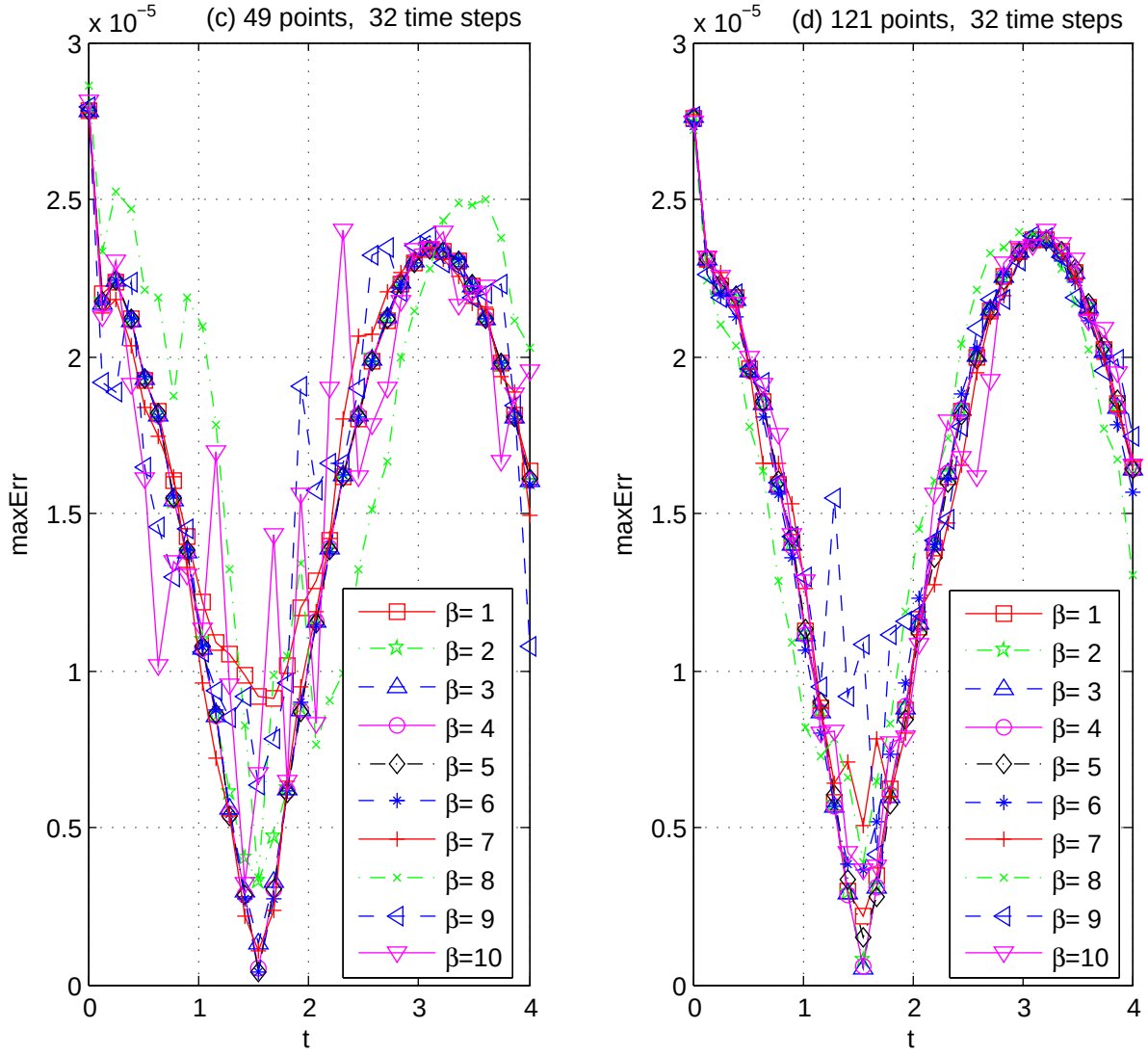

Figure 3.7: In Test problem 2, as the point density increases from figure (c) to more than 2 times denser in figure (d), the maximum errors by the MQ-IRBFN method do not decrease noticeably, indicating "mesh" convergence. The accuracy of the solution can be, however, improved by an order of magnitude by decreasing the time step size in Figure (3.6) to half the value in this figure. In addition, the errors by the MQ-IRBFN method do not vary significantly within the range of β's values from 1 to 10.

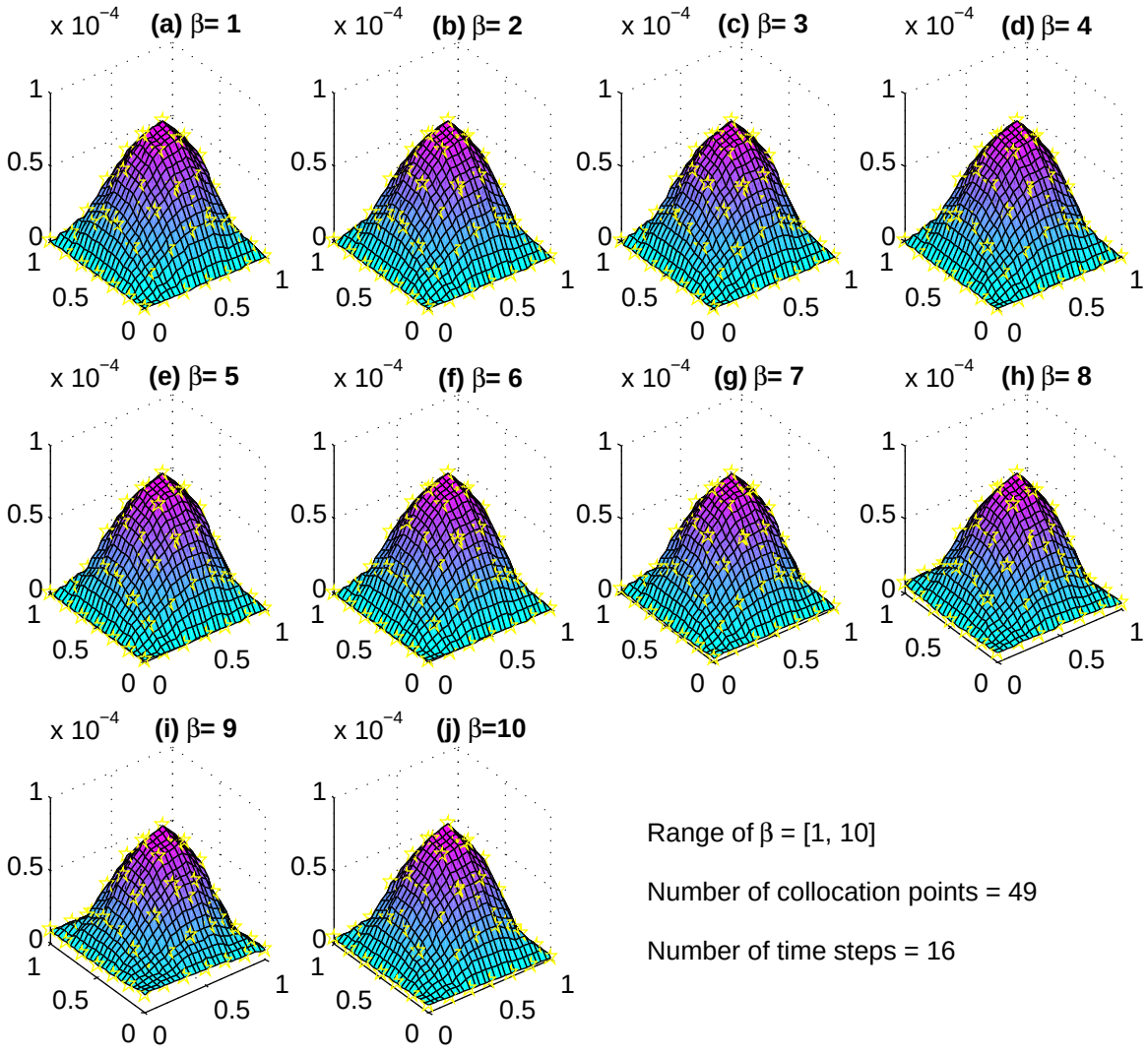

Figure 3.8: The root-mean-squared errors by the MQ-IRBFN method in Test problem 2 do not vary significantly within a range of β's values from 1 to 10 in figures (a)-(j) where the number of time steps $Nts = 16$.

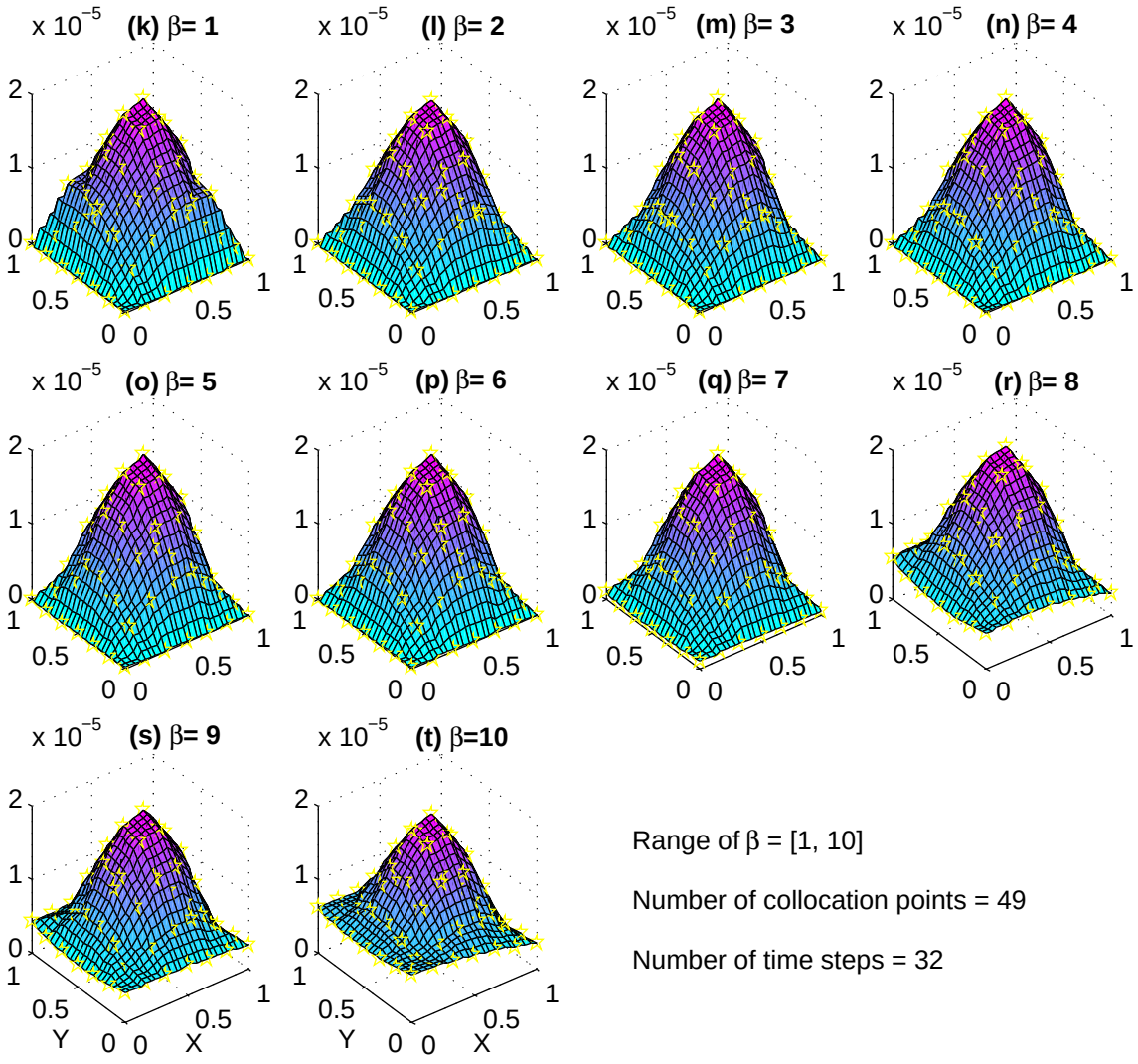

Figure 3.9: The root-mean-squared errors by the MQ-IRBFN method in Test problem 2 do not vary significantly within a range of β's values from 1 to 10 in figures (k)-(t) for $Nts = 32$. An increase of the number of time steps from 16 (Figure 3.8) to 32 (this figure) results in a noticeable decrease in the root-mean-squared errors.

3.4.3 Test problem 3. One-dimensional wave equation

Consider the 1D wave equation

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t) = 0, \quad 0 < x < 1, \quad t > 0,$$

subject to Dirichlet boundary conditions

$$u(0,t) = u(1,t) = 0, \quad t > 0,$$

and initial conditions

$$u(x,0) = f(\boldsymbol{x}), \quad 0 \le x \le 1,$$

$$\frac{\partial u}{\partial t}(x,0) = g(x), \quad 0 \le x \le 1.$$

With $f(x) = \sin \pi x$, $g(x) = 0$, $\alpha = 2$ the problem has the analytical solution

$$u(x,t) = \sin \pi x \, \cos 2\pi t.$$

The problem is solved by semi-discrete schemes described in section (3.3.3) using $N = 11$ collocation points in space. In order to apply the semi-discrete scheme, the problem is first reformulated as a system of first order ODEs

$$\frac{\partial u}{\partial t}(x,t) = v(x,t),$$

$$\frac{\partial v}{\partial t}(x,t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t),$$

with initial conditions

$$u(x,0) = f(x),$$

$$v(x,0) = g(x),$$

and subject to the same boundary conditions as in the original problem.

In this test problem, the one-step fourth-order Runge-Kutta (RK) with adaptive step control and the Adams predictor-corrector (PC) schemes are applied to solve the problem in the IRBFN framework. The solution profile is shown in Figure 3.10. Tables 3.6 and 3.7 show the solution and absolute errors by the two semi-discrete schemes using MQ and TPS, respectively. There is no significant difference in the solutions by RK and PC schemes using the same type of RBF (MQ or TPS). The predictor-corrector scheme, however, proved to be more efficient than Runge-Kutta for non-stiff problems with stringent tolerances and when the right-hand-side functions are particularly expensive to evaluate (Quarteroni et al., 2000). As can be seen from the tables, MQ-IRBFN with $\beta = 2$ yields more accurate solution than TPS-IRBFN in this example.

A range of the user-defined parameter β is used together with four point densities (5,7,9,11) and two different numbers of time steps (25 and 50) to investigate their influence on the accuracy of the numerical solution. Figures (3.11) and (3.12) show that the maximum errors and root-mean-squared errors by the IRBFN method do not change significantly within a range of β's values from 1 to 3. Also, it can be seen from the figures that while "mesh" convergence can be observed at a very coarse point density, the accuracy of the solution can be further improved by increasing the number of time steps.

Table 3.6: Accuracy comparison in Test problem 3 between the two semi-discrete schemes, Runge-Kutta (RK) and Predictor-Corrector (PC), based on MQ-IRBFN. The numerical and analytical solutions are shown at time t=1.0. The L_∞-norm of the error vector for MQ-IRBFN using RK and PC are $4.8788e-3$ and $4.9064e-3$, respectively.

		MQ-IRBFN		Absolute Error	
X	Exact	RK	PC	RK	PC
0.10	0.30901699	0.30901605	0.30899770	9.48658E-007	1.92943E-005
0.20	0.58778525	0.58776384	0.58779166	2.14102E-005	6.40824E-006
0.30	0.80901699	0.80902993	0.80902490	1.29361E-005	7.90858E-006
0.40	0.95105652	0.95106126	0.95104420	4.74844E-006	1.23145E-005
0.50	1.00000000	0.99997783	1.00001933	2.21743E-005	1.93331E-005
0.60	0.95105652	0.95106128	0.95104421	4.75879E-006	1.23049E-005
0.70	0.80901699	0.80902993	0.80902490	1.29347E-005	7.90928E-006
0.80	0.58778525	0.58776383	0.58779165	2.14231E-005	6.39342E-006
0.90	0.30901699	0.30901605	0.30899771	9.40765E-007	1.92876E-005

Table 3.7: Accuracy comparison in Test problem 3 between the two semi-discrete schemes, Runge-Kutta (RK) and Predictor-Corrector (PC), based on TPS-IRBFN. The numerical and analytical solutions are shown at time t=1.0. The L_∞-norm of the error vector for TPS-IRBFN using RK and PC are $5.2349e-3$ and $5.2791e-3$, respectively.

		TPS-IRBFN		Absolute Error	
X	Exact	RK	PC	RK	PC
0.10	0.30901699	0.30905179	0.30906749	3.47971E-005	5.04963E-005
0.20	0.58778525	0.58768745	0.58769273	9.78054E-005	9.25245E-005
0.30	0.80901699	0.80906449	0.80903911	4.74931E-005	2.21147E-005
0.40	0.95105652	0.95106175	0.95111521	5.23780E-006	5.86968E-005
0.50	1.00000000	0.99997721	0.99992776	2.27937E-005	7.22411E-005
0.60	0.95105652	0.95106176	0.95111531	5.24051E-006	5.87906E-005
0.70	0.80901699	0.80906450	0.80903899	4.75097E-005	2.19910E-005
0.80	0.58778525	0.58768744	0.58769281	9.78157E-005	9.24443E-005
0.90	0.30901699	0.30905177	0.30906750	3.47800E-005	5.05023E-005

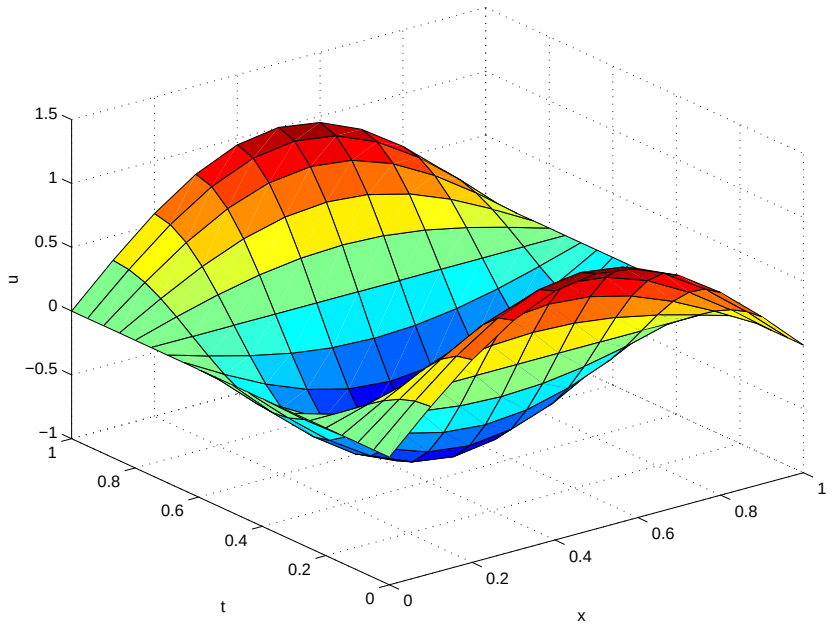

Figure 3.10: Solution profile of the problem in Test problem 3 by the MQ-IRBFN method.

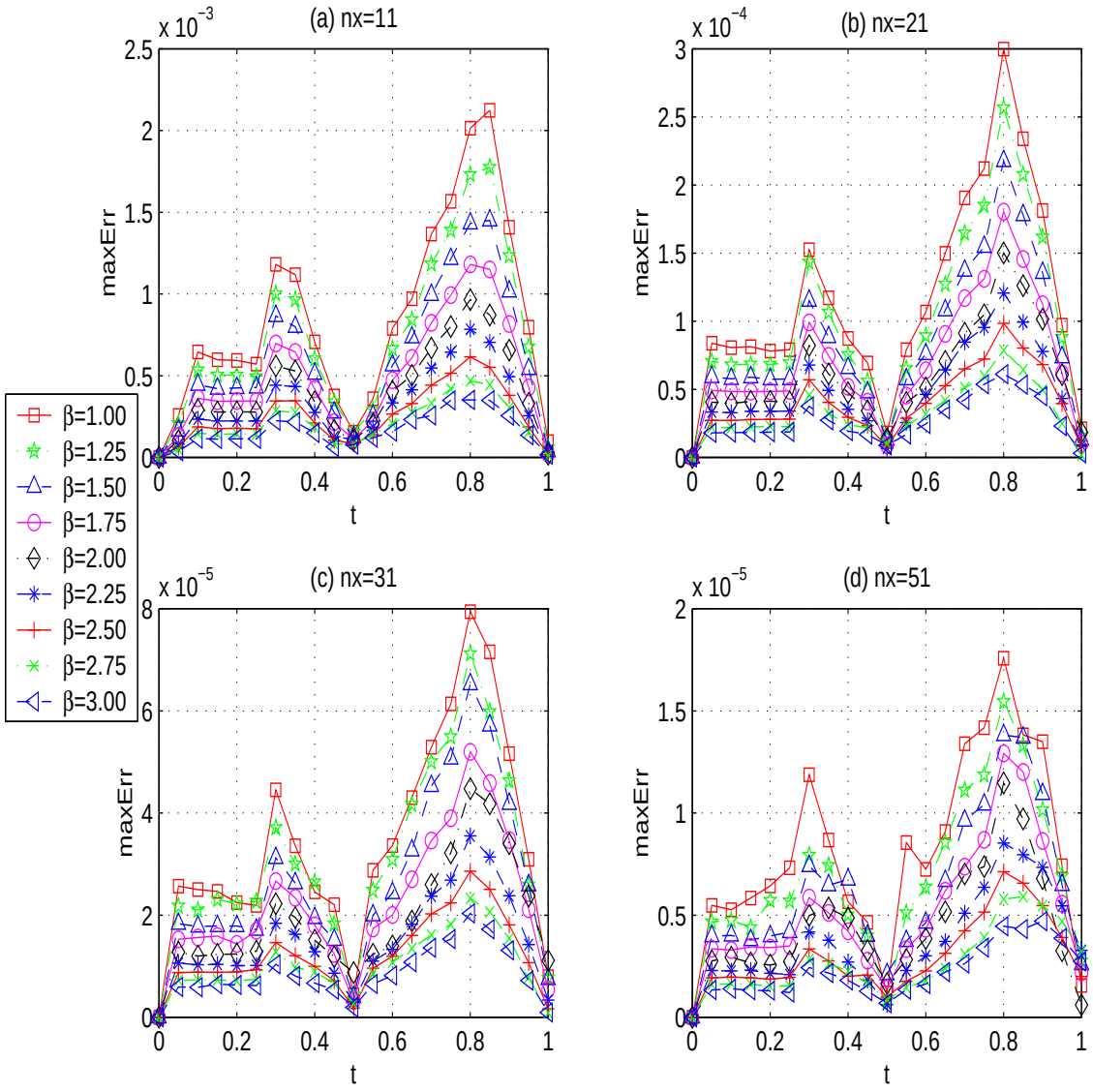

Figure 3.11: Sensitivity analysis of the MQ-IRBFN method for the problem in Test problem 3. The numerical solution yielded by the proposed method maintains its high accuracy with the maximum error remaining within an order of magnitude for the range of β's values from 1 to 3.

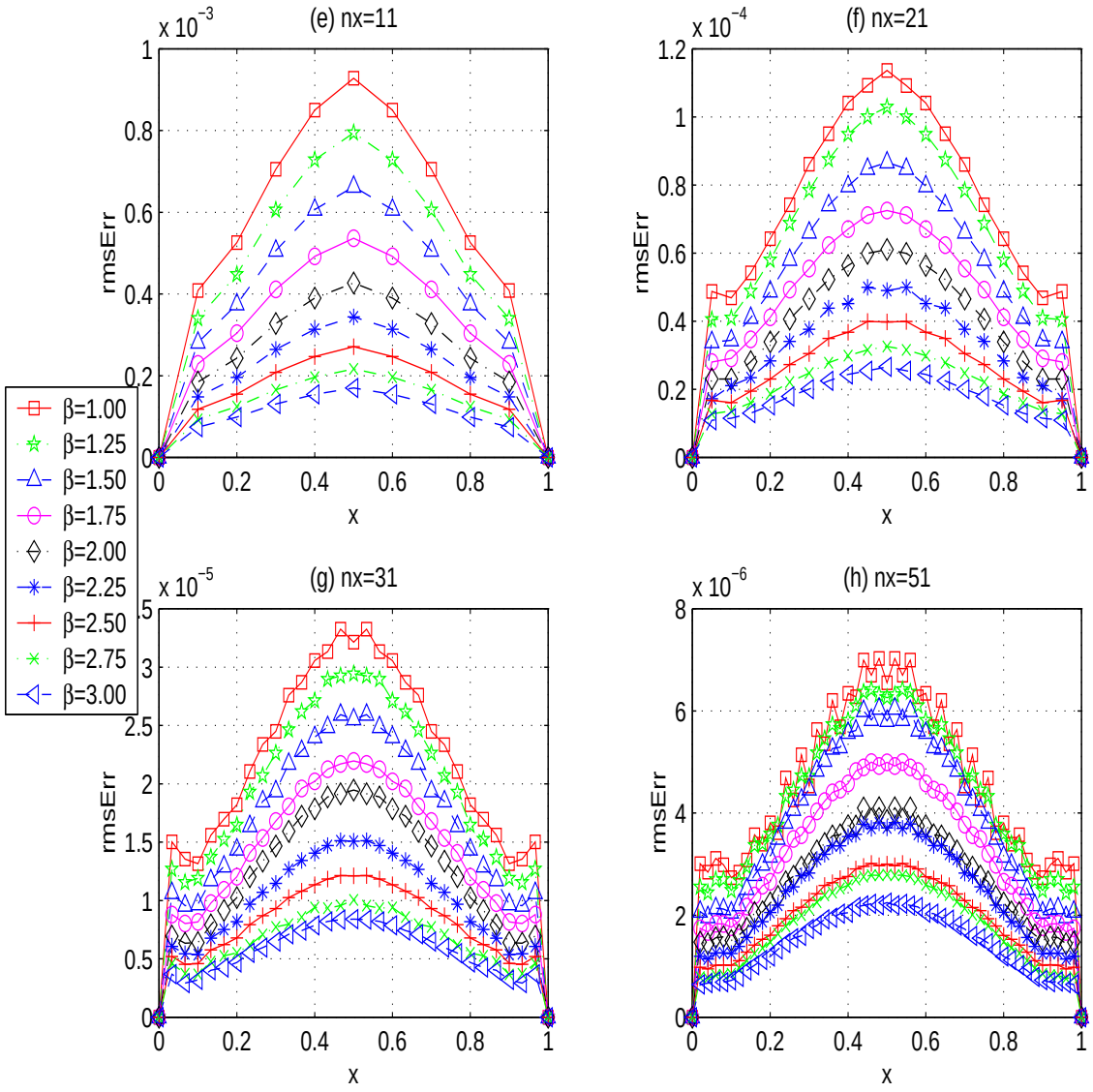

Figure 3.12: Sensitivity analysis of the MQ-IRBFN method for the problem in Test problem 3. The numerical solution yielded by the proposed method maintains its high accuracy with the root-mean-squared error remaining within an order of magnitude for the range of β's values from 1 to 3.

3.4.4 Test problem 4. Two-dimensional wave equation

Consider the 2D wave equation

$$\frac{\partial^2 u}{\partial t^2}(\boldsymbol{x},t) - \alpha^2 \nabla^2 u(\boldsymbol{x},t) = 0, \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^2, \quad t > 0,$$

subject to Dirichlet boundary conditions

$$u(\boldsymbol{x},t) = 0, \boldsymbol{x} \in \partial\Omega, \quad t > 0,$$

and initial conditions

$$u(\boldsymbol{x},0) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,$$

$$\frac{\partial u}{\partial t}(\boldsymbol{x},0) = g(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,$$

where $\Omega = \{(x,y) \,|\, 0 \le x \le a,\, 0 \le y \le b\}$. With $f(\boldsymbol{x}) = cx(a-x)y(b-y)$ and $g(\boldsymbol{x}) = 0$, the problem has the analytical solution

$$u(x,y,t) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \mathbf{B}_{mn} \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b} \cos\left(\alpha\pi t\sqrt{\frac{m^2}{a^2}+\frac{n^2}{b^2}}\right),$$

where

$$\mathbf{B}_{mn} = \frac{16a^2b^2c}{\pi^6 m^3 n^3}(1-\cos m\pi)(1-\cos n\pi).$$

The IRBFN semi-discrete scheme based on fourth-order Runge-Kutta scheme is used to solve the problem, and the numerical solution at time step = 5 is shown in Figure 3.13 where $\alpha^2 = 3$, $a = 1$, $b = 1$ and $c = 1$. It is noted that in order to apply the semi-discrete scheme, the problem is reformulated in to a system of first order ODEs as in Test problem 3. Comparisons of accuracy and efficiency between the FEM and the IRBFN method using both MQ and TPS are also performed in this example. With the FEM, a mesh of 177 nodes and 312 triangles is generated to solve the problem whereas for the IRBFN

Table 3.8: Comparison of accuracy and efficiency between the FEM and the MQ-IRBFN method in Test problem 4 at points (0.8,:) at time $t = 0.16$. The FEM uses 177 nodes and 312 triangles while the MQ-IRBFN method uses 121 collocation points. The L_∞-norm of the error vector for FEM and MQ-IRBFN are $7.9915e-4$ and $2.3697e-4$, respectively.

			Numerical solution		Absolute Error	
X	Y	Exact	FEM	MQ	FEM	MQ
0.80	0.10	0.001874	0.002213	0.001720	3.3933e-004	1.5402e-004
0.80	0.20	0.004222	0.004796	0.004229	5.7389e-004	6.9258e-006
0.80	0.30	0.007521	0.007719	0.007583	1.9771e-004	6.2266e-005
0.80	0.40	0.010127	0.010329	0.010149	2.0203e-004	2.2110e-005
0.80	0.50	0.010995	0.011122	0.011021	1.2658e-004	2.5854e-005
0.80	0.60	0.010127	0.010218	0.010149	9.1194e-005	2.2559e-005
0.80	0.70	0.007521	0.007961	0.007580	4.4046e-004	5.9521e-005
0.80	0.80	0.004222	0.004797	0.004231	5.7448e-004	8.3148e-006
0.80	0.90	0.001874	0.002177	0.001728	3.0315e-004	1.4563e-004

method, 121 collocation points are used. The accuracy comparison between the two methods is shown in Table 3.8 and 3.9 for the case of MQ and TPS, respectively. As can be seen from the tables, the IRBFN method gives more accurate result with a smaller number of points. It should be noted that for the FEM, a numerical interpolation is required to calculate the solution at some non-nodal points after obtaining the solution on the triangulated mesh. For the IRBFN method, with its unstructured configuration, particular points of interest can be inserted into the computational grid at early stages before solving the PDEs so that interpolation is not needed. This can be done with the IRBFN method at no additional cost. Alternatively, the solution at any point $\boldsymbol{x}$ at time t^n can be obtained simply by evaluating (3.30) after the solving process has been completed for time level t^n.

Table 3.9: Comparison of accuracy and efficiency between the FEM and the TPS-IRBFN method in Test problem 4 at points (0.8,:) at the point of time $t = 0.16$. The FEM uses 177 nodes and 312 triangles while the TPS-IRBFN method uses 121 collocation points. The L_∞-norm of the error vector for FEM and TPS-IRBFN are $7.9915e-4$ and $2.4562e-4$, respectively.

			Numerical solution		Absolute Error	
X	Y	Exact	FEM	TPS	FEM	TPS
0.80	0.10	0.001874	0.002213	0.001702	3.3933e-004	1.7164e-004
0.80	0.20	0.004222	0.004796	0.004245	5.7389e-004	2.2934e-005
0.80	0.30	0.007521	0.007719	0.007588	1.9771e-004	6.6888e-005
0.80	0.40	0.010127	0.010329	0.010176	2.0203e-004	4.8762e-005
0.80	0.50	0.010995	0.011122	0.011045	1.2658e-004	4.9373e-005
0.80	0.60	0.010127	0.010218	0.010169	9.1194e-005	4.1971e-005
0.80	0.70	0.007521	0.007961	0.007548	4.4046e-004	2.7258e-005
0.80	0.80	0.004222	0.004797	0.004247	5.7448e-004	2.5058e-005
0.80	0.90	0.001874	0.002177	0.001769	3.0315e-004	1.0419e-004

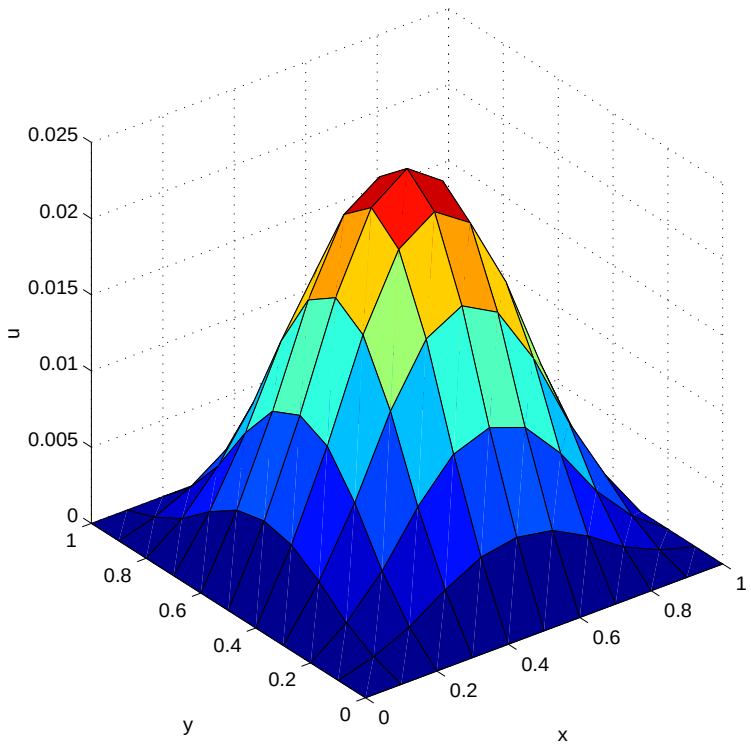

Figure 3.13: Numerical solution by the MQ-IRBFN method at time step 5 for the problem in Test problem 4.

3.4.5 Test problem 5. One-dimensional advection-diffusion equation

Consider the 1D advection-diffusion equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + v \frac{\partial u}{\partial x}, \quad 0 < x < 1, \quad t > 0,$$

subject to boundary and initial conditions

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

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

The analytical solution is given by

$$u(x,t) = ae^{bt-cx}, \text{ where } c = \frac{v \pm \sqrt{v^2 + 4\kappa b}}{2\kappa} > 0.$$

The IRBFN and DRBFN methods are used to solve the problem in this test. For comparison purposes, the solution to the problem by DRBFN method is obtained from (Zerroukat et al., 2000) where the implicit scheme, namely ICNTPS, was reported to be the most efficient among the other schemes. The ICNTPS scheme uses TPS for spatial discretization and θ-scheme with $\theta = 0.5$ for time integration. For the IRBFN method, a semi-discrete scheme based on the fourth-order Runge-Kutta scheme is used together with either MQ or TPS to solve this problem. Solutions at time $t = 1.0$ by direct and indirect methods are shown in Tables 3.10 and 3.11 where $\kappa = 0.1$, $b = 0.1$, $c = 1.61803$, $a = 1.0$, and $v = 0.1$. It can be seen from the tables that, with the same number of collocation points, the semi-discrete scheme using either MQ or TPS outperforms the ICNTPS scheme while the number of time steps used in the present IRBFN method is half as many as that required by the DRBFN method.

Table 3.10: Test problem 5: Comparison of accuracy and efficiency between the MQ-IRBFN method and the TPS-DRBFN method (Zerroukat et al., 2000) at $t = 1$. With the same number of collocation points, the MQ-IRBFN method outperforms the TPS-DRBFN method while the number of time steps it uses (Nts=25) is half as many as that required by the TPS-DRBFN method. The L_∞-norm of the error vector for TPS-DRBFN and MQ-IRBFN are $9.8935e-3$ and $5.9243e-5$, respectively.

		Numerical solution		Absolute Error	
X	Analytical	DRBFN	MQ-IRBFN	DRBFN	MQ-IRBFN
0.2500	0.737486	0.747380	0.737494	9.89e-003	7.76e-006
0.3125	0.666554	0.675000	0.666578	8.45e-003	2.38e-005
0.3750	0.602444	0.608810	0.602454	6.37e-003	9.92e-006
0.5000	0.492129	0.493910	0.492116	1.78e-003	1.22e-005
0.6250	0.402014	0.400190	0.401988	1.82e-003	2.59e-005
0.7500	0.328400	0.324740	0.328358	3.66e-003	4.19e-005
0.8750	0.268266	0.264260	0.268207	4.01e-003	5.92e-005

Table 3.11: Test problem 5: Comparison of accuracy and efficiency between the TPS-IRBFN method and the TPS-DRBFN method (Zerroukat et al., 2000) at $t = 1$. With the same number of collocation points, the TPS-IRBFN method outperforms the TPS-DRBFN method while the number of time steps it uses (Nts=25) is half as many as that required by the TPS-DRBFN method. The L_∞-norm of the error vector for TPS-DRBFN and TPS-IRBFN are $9.8935e-3$ and $1.2150e-3$, respectively.

		Numerical solution		Absolute Error	
X	Analytical	DRBFN	TPS-IRBFN	DRBFN	TPS-IRBFN
0.2500	0.737486	0.747380	0.737512	9.89e-003	2.50e-005
0.3125	0.666554	0.675000	0.666701	8.45e-003	1.47e-004
0.3750	0.602444	0.608810	0.602286	6.37e-003	1.58e-004
0.5000	0.492129	0.493910	0.491800	1.78e-003	3.28e-004
0.6250	0.402014	0.400190	0.401410	1.82e-003	6.03e-004
0.7500	0.328400	0.324740	0.327574	3.66e-003	8.26e-004
0.8750	0.268266	0.264260	0.267051	4.01e-003	1.22e-003

3.5 Concluding Remarks

Two new meshless numerical schemes, namely the fully discrete and semi-discrete IRBFN-based schemes, have been presented in this chapter. In these schemes, the IRBFN method is used for the spatial discretization whereas the θ-method or a class of ODE solver is used as the temporal discretization. With the new formulation where the unknowns are the values of the function to be solved rather than the weight coefficients, no additional constraints are needed. Numerical experiments show that the solutions given by the two proposed schemes are more accurate and efficient than those resulted from some traditional methods such as FDM, FEM, BEM and DRBFN even in the cases where the number of grid points and the number of time steps are smaller. In addition, numerical results show that the new schemes are less sensitive to the network parameters β for multiquadric basis functions and the first order TPS ($m = 1$) can be used regardless of the differential order in the PDEs under consideration.

Chapter 4

Meshless Approach to Passive Transport Problems

This chapter presents a new meshless numerical approach to solving a special class of moving interface problems known as the passive transport where an ambient flow characterized by its velocity field causes the interfaces to move and deform without any influences back on the flow. In the present approach, the moving interface is captured by the level set method at all time as the zero contour of a smooth function known as the level set function whereas one of the two new meshless schemes, namely the SL-IRBFN based on the semi-Lagrangian method and the Taylor-IRBFN scheme based on Taylor series expansion, is used to solve a convective transport equation for advancing the level set function in time. In addition, a mass correction is introduced after the reinitialization step to ensure mass conservation. Some basic tests are performed to verify the accuracy and stability of the new numerical schemes which are then applied to simulate bubbles moving, stretching and merging in an ambient flow to demonstrate the performance of the new meshless approach.

4.1 Introduction

Numerical methods for moving interface problems in general, or passive transport problems in particular, have been increasingly studied in recent years. A moving interface is the boundary $\Gamma(t)$ of $\Omega \subset \mathbf{R}^d, d = 1, 2, 3$, depending on time t that has an outward unit normal $\mathbf{n}$ and a normal velocity (also known as speed) F at each point. A passive transport problem is defined as to find $\Gamma(t)$ at time $t > 0$ moving in a given externally generated velocity field $\mathbf{v}$ such that $F = \mathbf{v} \cdot \mathbf{n}$. In such a problem, the influence of the moving interfaces on the velocity field is ignored. When it moves, the interface $\Gamma(t)$ might be subjected to topology changes such as shape splitting and/or merging.

There are two basic approaches to modeling the motion of the interfaces: moving-grid and fixed-grid methods. In the moving-grid methods, the interface is treated as the boundary of a moving surface-fitted grid (Floryan and Rasmussen, 1989). This approach allows a precise representation of the interface whereas its main drawback is the severe deformation of the mesh as the interface moves. The second approach which is based on fixed grids includes tracking and capturing methods. The tracking methods explicitly represent the moving interface by means of predefined markers (Unverdi and Tryggvason, 1992). In capturing methods, on the other hand, the moving interface is not explicitly tracked, but rather captured via a characteristic function. Examples of the capturing methods are phase field method (Jacqmin, 1999), volume-of-fluid method (Hirt and Nichols, 1981) and level set method (Osher and Sethian, 1988). The characteristic function used to implicitly describe the moving interface is the order parameter in the phase field method, volume fraction in the volume-of-fluid method and level set function in the level set method. For these capturing methods, no grid manipulation (e.g. rezoning/remeshing) is needed to maintain the overall accuracy even when the interface undergoes large deformation. In this research, the level set method is used to capture the moving interfaces.

The underlying idea of the level set method is to embed the moving interface $\Gamma(t)$ as the zero level set of a smooth (at least Lipchitz continuous) function $\phi(\boldsymbol{x}, t)$ known as the level set function (Osher and Sethian, 1988). The moving interface can be then captured at any time by locating the set of $\Gamma(t)$ for which $\phi(\boldsymbol{x}, t)$ vanishes. The level set function is advanced with time by a convective transport equation known as the level set equation. Usually, $\phi(\boldsymbol{x}, t)$ is initialized as a signed distance function to the interface (Sethian, 1999; Osher and Fedkiw, 2003). Due to numerical error, however, this feature is not necessarily held even after one time step. Reinitialization is therefore needed to make the level set function signed distance function after certain time steps which could be achieved by solving a time-dependent PDE to steady state (Sussman et al., 1994). It has been reported that such a reinitialization procedure could introduce some numerical diffusion which results in an inaccuracy of the interface location and some loss of mass (Tornberg, 2000). The procedure has been improved in (Chang et al., 1996; Sussman and Fatemi, 1999; Peng et al., 1999). In this work, an additional mass correction based on a well-known formula for the first variation of a volume integral (Cuvelier and Schulkes, 1990) is introduced after the reinitialization step to prevent any significant losses of mass.

The level set method has been applied widely in fluid dynamics (Sussman et al., 1994; Sussman and Smereka, 1997; Iafrati et al., 2001), to name just a few applications. Some conservative schemes have been used to solve the level set equation such as Lax-Friedrichs (Crandall and Lions, 1984), Essentially Non-Oscillatory ENO (Shu and Osher, 1989), and Godunov's schemes (Bardi and Osher, 1991). In this thesis, two new meshless numerical schemes, namely the SL-IRBFN scheme based on the semi-Lagrangian method and the Taylor-IRBFN motivated by the well-known Taylor-Galerkin method, are proposed to deal with the level set equation.

The semi-Lagrangian method can be considered as a hybrid approach between the Eulerian and the Lagrangian methods (Staniforth and Cote, 1991). An Eu-

lerian scheme, where the fluid motion is observed from a fixed point, retains the regularity of the mesh but requires small time steps in order to maintain stability. A Lagrangian scheme, on the other hand, is less restricted by stability requirements and allows larger time steps. However, since the fluid particles, initially regularly spaced, move with time, they usually become irregularly spaced as the system evolves. The semi-Lagrangian advection scheme combines the advantages of both schemes - the regularity of the Eulerian scheme and the enhanced stability of the Lagrangian scheme. The basic idea is to discretize the Lagrangian derivative of the transport quantity in time instead of the Eulerian derivative. It involves backward time integration along the characteristic curve to find the departure point of a fluid particle arriving at an Eulerian grid point. The solution at the departure points is then obtained by interpolation. Interested readers are referred to (Staniforth and Cote, 1991; Oliveira and Baptista, 1995; Behrens and Iske, 2002) and the references therein for details on semi-Lagrangian methods.

Another well-known numerical method, namely the Taylor-Galerkin method, is widely used for solving convective transport equations (Donea, 1984). This method is based on the Taylor series expansion about a point in time of a function including higher-order time derivatives. In general, by replacing the temporal derivatives in the Taylor series expansion with the corresponding spatial ones via the differential equation to be solved, the accuracy and the stability of the numerical solution can be improved (Donea and Huerta, 2003). For the Taylor-Galerkin method, the resultant semi-discrete equation is discretized in space using the standard Galerkin FEM method. For the Taylor-IRBFN scheme, on the other hand, the IRBFN method is used for spatial discretization.

The remaining of this chapter is organized as follows. Firstly the level set method, the meshless semi-Lagrangian SL-IRBFN and the Taylor-IRBFN schemes are introduced. The new meshless approach to solving passive transport problems is then presented with a detailed discussion on all “ingredients” mentioned

above, particularly the SL-IRBFN, Taylor-IRBFN schemes and the additional mass correction procedure. Finally, the individual schemes and the new approach are verified with some basic tests and demonstrated with some typical passive transport problems.

4.2 Level set method

In the level set method, the moving interface $\Gamma(t)$ which bounds an open region $\Omega \subset \mathbf{R}^d$ $(d = 2, 3)$ is embedded as the zero level set of a higher dimensional function $\phi(\boldsymbol{x}, t)$ such that

$$\Gamma(t) = \{\boldsymbol{x} \in \mathbf{R}^d | \ \phi(\boldsymbol{x}, t) = 0\}$$

Initially, ϕ is defined as the signed distance function from the front such that

$$\phi(\boldsymbol{x}, t) = \begin{cases} +d(\boldsymbol{x}, t) & \boldsymbol{x} \in \Omega^+ \\ 0 & \boldsymbol{x} \in \Gamma \\ -d(\boldsymbol{x}, t) & \boldsymbol{x} \in \Omega^- \end{cases} \tag{4.1}$$

where $d(\boldsymbol{x}, t)$ represents the Euclidean distance from $\boldsymbol{x}$ to the interface, Ω^- and Ω^+ are interior and exterior regions respectively. The interface can be then captured at any time by locating the set of $\Gamma(t)$ for which ϕ vanishes. In other words, instead of working with the interface, one evolves the level set with the following transport equation for ϕ,

$$\phi_t + \mathbf{v} \cdot \nabla\phi = 0, \quad \phi(\boldsymbol{x}, t = 0) = \phi_0(\boldsymbol{x}), \tag{4.2}$$

where $\phi_0(\boldsymbol{x})$ is a given function. This equation is also known as the kinematic boundary condition at the moving interface. Whenever needed, the moving interface can be extracted as the zero level of the level set function $\phi(\boldsymbol{x}, t)$. It is noted that while the level set function $\phi(\boldsymbol{x}, t)$ is initialized as a signed distance

function from the free surface, this is not necessarily true as time proceeds. In order to keep the numerical solution accurate, one needs to reinitialize $\phi(\boldsymbol{x}, t)$ to be the signed distance function from the evolving front $\Gamma(t)$ after certain period in time. This is accomplished by solving the following problem to steady state:

$$\phi_t = S_\epsilon(\bar{\phi})(1 - |\nabla\phi|), \quad \phi(x, y, t = 0) = \bar{\phi}(x, y) \tag{4.3}$$

where S_ϵ denotes the smoothed sign function

$$S_\epsilon(\bar{\phi}) = \frac{\bar{\phi}}{\sqrt{\bar{\phi}^2 + \epsilon^2}} \tag{4.4}$$

in which ϵ can be chosen to be the minimum distance from any data point to the others.

As mentioned earlier, due to numerical diffusion coming from the approximation of $sign(\phi)$ in solving equation (4.3), the reinitialization procedure presented above could move the interface location and cause some losses of mass. A mass correction which is added after the reinitialization step is described in section §4.5.4.

4.3 SL-IRBFN scheme for convective transport equations

Consider the transport equation with source term

$$\frac{\partial\phi}{\partial t} + \mathbf{v} \cdot \nabla\phi = f(\boldsymbol{x}, t), \tag{4.5}$$

where $\phi = \phi(\boldsymbol{x}, t)$ is a scalar quantity, and $\mathbf{v}(\boldsymbol{x})$ is a given convection velocity. The above equation can be written in the following Lagrangian form

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = f(\boldsymbol{x}, t) \tag{4.6}$$

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \mathbf{v}(\boldsymbol{x}, t) \tag{4.7}$$

The solving procedure for Equations (4.6) and (4.7) using semi-Lagrangian method is as follows.

- At each time step, track backward particles that arrive at the grid points over a single time step along characteristic curves (4.7) to their departure points;
- Compute the solution values at the departure points;
- Solve (4.6) for the current time step using the solution values at the departure points as the initial values;
- Advance to the next time step and repeat the above steps until the predefined time is reached.

Applying the semi-Lagrangian method to problem (4.6) in which the first-order backward Euler difference scheme is used for the time derivative, one obtains

$$\frac{\phi^{n+1} - \phi_d^n}{\Delta t} = f^{n+1}, \tag{4.8}$$

where ϕ_d^n is the value of ϕ at the departure points $\boldsymbol{x}_d$. ϕ_d^n is determined by first solving the following equation

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \mathbf{v}(\boldsymbol{x}, t), \quad \boldsymbol{x}^{n+1} = \boldsymbol{x}_a, \tag{4.9}$$

backward in one single step for the departure points $\boldsymbol{x}_d$ at time t^n with the initial condition $\boldsymbol{x}^{n+1} = \boldsymbol{x}_a$. ϕ_d^n can be then obtained via interpolation from ϕ^n

at grid points. In the above equation, $\boldsymbol{x}_a$ is the position of the arrival points which are the grid points at time t^{n+1}. Equation (4.9) can be solved by the explicit midpoint rule (Temperton and Staniforth, 1987) as follows.

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_a - \frac{\Delta t}{2}\mathbf{v}(\boldsymbol{x}_a, t^n), \tag{4.10}$$

$$\boldsymbol{x}_d = \boldsymbol{x}_a - \Delta t\, \mathbf{v}\left(\hat{\boldsymbol{x}}, t^n + \frac{\Delta t}{2}\right). \tag{4.11}$$

By letting

$$\boldsymbol{\delta} = \boldsymbol{x}_a - \boldsymbol{x}_d, \tag{4.12}$$

and substituting (4.10) into (4.11), one obtains

$$\boldsymbol{\delta} = \Delta t\, \mathbf{v}\left(\boldsymbol{x}_a - \frac{\Delta t}{2}\mathbf{v}(\boldsymbol{x}_a, t^n), t^n + \frac{\Delta t}{2}\right). \tag{4.13}$$

Once $\boldsymbol{\delta}$ is solved, the departure points $\boldsymbol{x}_d$ can be found via Equation (4.12). It is noted that the velocity field at $t = t^n + \frac{\Delta t}{2}$ in (4.13) can be determined by extrapolation using the Adams-Bashforth formula

$$\mathbf{v}(\boldsymbol{x}, t^n + \frac{\Delta t}{2}) = \frac{3}{2}\mathbf{v}(\boldsymbol{x}, t^n) - \frac{1}{2}\mathbf{v}(\boldsymbol{x}, t^n - \Delta t) + \mathcal{O}(\Delta t^2). \tag{4.14}$$

Alternatively, equation (4.9) can be solved by applying an implicit midpoint rule as follows.

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_a - \frac{\Delta t}{2}\mathbf{v}\left(\hat{\boldsymbol{x}}, t^n + \frac{\Delta t}{2}\right). \tag{4.15}$$

$$\boldsymbol{x}_d = \boldsymbol{x}_a - \Delta t\, \mathbf{v}\left(\hat{\boldsymbol{x}}, t^n + \frac{\Delta t}{2}\right). \tag{4.16}$$

In this case, one has to solve the following equation for $\boldsymbol{\delta}$,

$$\boldsymbol{\delta} = \Delta t\, \mathbf{v}\left(\boldsymbol{x}_a - \frac{\boldsymbol{\delta}}{2}, t^n + \frac{\Delta t}{2}\right). \tag{4.17}$$

Although equation (4.17) has to be solved iteratively, it converges after just a few iterations provided that $\max|\nabla \mathbf{v}|\Delta t$ is sufficiently small (Allievi and Bermejo, 2000). To enhance the accuracy of the integration, higher-order methods should be used. In this work, the IRBFN semi-discrete scheme presented in section (3.3.3) with the fourth-order Runge-Kutta method is used.

In general, the departure points $\boldsymbol{x}_d$ do not coincide with the grid points. The value of ϕ_d^n at those points is then obtained by interpolation. The IRBFN method with Duchon TPS basis functions is used for this purpose. Alternatively, the cubic spline interpolation can be used. After getting ϕ_d^n, the new value ϕ^{n+1} at time t^{n+1} can be obtained by equation (4.8). For convective transport equation with no source term, the semi-Lagrangian scheme reduces to $\phi^{n+1} = \phi_d^n$, i.e. the value of function ϕ remains constant on the characteristics and the old value is simply copied into its new position on the regular grid.

4.4 Taylor-IRBFN schemes for convective transport equations

Consider the two-dimensional pure convective transport equation

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} + v\frac{\partial \phi}{\partial y} = 0, \tag{4.18}$$

$$\phi(x, y, t = 0) = \phi^0(x, y), \tag{4.19}$$

where $\phi^0(x, y)$ is a given function, and $u = u(x, y)$ and $v = v(x, y)$ are the component of a given time-independent velocity field in x and y direction, respectively. For the sake of presentation using the same notations as in (Donea, 1984), Equation (4.18) is rewritten as follows.

$$\phi_t = -u\phi_x - v\phi_y, \tag{4.20}$$

where ϕ_t, ϕ_x, ϕ_y are the derivatives of ϕ in time, x and y direction, respectively.

In the remaining parts of this section, two formulations of the Taylor-IRBFN scheme, namely TE-IRBFN and TCN-IRBFN are derived to solve the problem under consideration. The two formulations differ on the way in which the first-order derivative in time is approximated. The former is based on the Euler difference formula whereas in the latter the Crank-Nicolson method is used.

4.4.1 The TE-IRBFN Scheme

The TE-IRBFN scheme for solving (4.18) is derived as follows.

Firstly, applying Taylor series expansion of ϕ forward about $t = t^n$ yields

$$\phi^{n+1} = \phi^n + \Delta t\phi_t^n + \frac{\Delta t^2}{2}\phi_{tt}^n + \frac{\Delta t^3}{6}\phi_{ttt}^n + \mathcal{O}(\Delta t^4), \tag{4.21}$$

or

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \phi_t^n + \frac{\Delta t}{2}\phi_{tt}^n + \frac{\Delta t^2}{6}\phi_{ttt}^n + \mathcal{O}(\Delta t^3). \tag{4.22}$$

Secondly, by differentiating Equation (4.20) successively up to the third order derivative in time and replacing the first order derivatives in time with the corresponding spatial derivatives in the right-hand side of the transport equation to be solved (4.20), one obtains

$$\phi_{tt} = \left[(uu_x + u_y v)\partial_x + u^2\partial_{xx} + (uv_x + vv_y)\partial_y + v^2\partial_{yy} + 2uv\partial_{xy}\right]\phi, \tag{4.23}$$

$$\phi_{ttt} = \left[(uu_x + u_y v)\partial_x + u^2\partial_{xx} + (uv_x + vv_y)\partial_y + v^2\partial_{yy} + 2uv\partial_{xy}\right]\phi_t, \tag{4.24}$$

where ∂_x and ∂_y denote the spatial differential operators in x and y directions, respectively. The last first-order time derivative in Equation (4.24) is kept to avoid high-order spatial derivatives in the resulting formulas.

By using the new differential operator notation $\partial_{\boldsymbol{\chi}}$ defined as

$$\partial_{\boldsymbol{\chi}} = \left[(uu_x + u_y v)\partial_x + u^2\partial_{xx} + (uv_x + vv_y)\partial_y + v^2\partial_{yy} + 2uv\partial_{xy}\right], \quad (4.25)$$

one has the simpler forms of Equations (4.23) and (4.24) as follows.

$$\phi_{tt} = \partial_{\boldsymbol{\chi}}\phi, \quad (4.26)$$

$$\phi_{ttt} = \partial_{\boldsymbol{\chi}}\phi_t. \quad (4.27)$$

Next, substituting (4.20), (4.26) and (4.27) into (4.22) yields

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \left(-u\partial_x - v\partial_y\right)\phi^n + \frac{\Delta t}{2}\partial_{\boldsymbol{\chi}}\phi^n + \frac{\Delta t^2}{6}\partial_{\boldsymbol{\chi}}\phi_t^n + \mathcal{O}(\Delta t^3). \quad (4.28)$$

Finally, by replacing ϕ_t^n in the above equation with the Euler difference formula

$$\phi_t^n = \frac{\phi^{n+1} - \phi^n}{\Delta t}, \quad (4.29)$$

and rearranging the terms, one obtains

$$\left(1 - \frac{\Delta t^2}{6}\partial_{\boldsymbol{\chi}}\right)\Delta\phi = \left[\Delta t\left(-u\partial_x - v\partial_y\right) + \frac{\Delta t^2}{2}\partial_{\boldsymbol{\chi}}\right]\phi^n + \mathcal{O}(\Delta t^3), \quad (4.30)$$

where $\Delta\phi = \phi^{n+1} - \phi^n$. For each time step, Equation (4.30) is solved for $\Delta\phi$, and the solution at $t = t^{n+1}$ is obtained via $\phi^{n+1} = \phi^n + \Delta\phi$.

Using the IRBFN method for the spatial discretization of Equation (4.30), the fully discrete TE-IRBFN formulation for problem (4.18) can be then derived as follows.

$$\left[1 - \frac{\Delta t^2}{6}\boldsymbol{\psi}_{\partial_{\boldsymbol{\chi}}}^T(\boldsymbol{x}_i)\right]\Delta\boldsymbol{\phi} = \left[\Delta t\left(-u_i\boldsymbol{\psi}_{\partial_x}^T(\boldsymbol{x}_i) - v_i\boldsymbol{\psi}_{\partial_y}^T(\boldsymbol{x}_i)\right) + \frac{\Delta t^2}{2}\boldsymbol{\psi}_{\partial_{\boldsymbol{\chi}}}^T(\boldsymbol{x}_i)\right]\boldsymbol{\phi}^n, \quad i = 1, 2, \ldots, M, \quad (4.31)$$

where $\boldsymbol{\psi}_{\partial_{\boldsymbol{\chi}}}$, $\boldsymbol{\psi}_{\partial_x}$ and $\boldsymbol{\psi}_{\partial_y}$ are the IRBFN approximations to the differential operator $\partial_{\boldsymbol{\chi}}$, ∂_x and ∂_y, respectively; u_i and v_i are the components of the velocity field $\mathbf{v}$ at position $\boldsymbol{x}_i$ in x and y directions, respectively.

4.4.2 The TCN-IRBFN Scheme

The TCN-IRBFN scheme for solving (4.18) is derived as follows.

Firstly, applying Taylor series expansions of function ϕ forward about $t = t^n$ and backward about $t = t^{n+1}$ yields

$$\phi^{n+1} = \phi^n + \Delta t\phi_t^n + \frac{\Delta t^2}{2}\phi_{tt}^n + \frac{\Delta t^3}{6}\phi_{ttt}^n + \mathcal{O}(\Delta t^4), \tag{4.32}$$

$$\phi^n = \phi^{n+1} - \Delta t\phi_t^{n+1} + \frac{\Delta t^2}{2}\phi_{tt}^{n+1} - \frac{\Delta t^3}{6}\phi_{ttt}^{n+1} + \mathcal{O}(\Delta t^4) \tag{4.33}$$

Subtracting (4.33) from (4.32) and rearranging terms result in

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \frac{(\phi_t^n + \phi_t^{n+1})}{2} + \frac{\Delta t}{4}(\phi_{tt}^n - \phi_{tt}^{n+1}) + \frac{\Delta t^2}{12}(\phi_{ttt}^n + \phi_{ttt}^{n+1}) + \mathcal{O}(\Delta t^3). \tag{4.34}$$

Next, substituting (4.20),(4.23),(4.24) into (4.34) and rearranging terms, one obtains

$$\begin{aligned}\frac{\phi^{n+1} - \phi^n}{\Delta t} = -\frac{1}{2}(u\partial_x + v\partial_y)\left(\phi^n + \phi^{n+1}\right) + \frac{\Delta t}{4}\partial\boldsymbol{\chi}\left(\phi^n - \phi^{n+1}\right) \\ + \frac{\Delta t^2}{12}\partial\boldsymbol{\chi}\left(\phi_t^n + \phi_t^{n+1}\right) + \mathcal{O}(\Delta t^3),\end{aligned} \tag{4.35}$$

where the differential operator notation $\partial_{\boldsymbol{\chi}}$ is defined in (4.25).

Finally, by applying the Crank-Nicolson time-stepping (Donea, 1984)

$$\frac{1}{2}\left(\phi_t^n + \phi_t^{n+1}\right) = \frac{\phi^{n+1} - \phi^n}{\Delta t}, \tag{4.36}$$

one has the semi-discrete form of (4.20) as follows.

$$\left[1+\frac{\Delta t}{2}(u\partial_x+v\partial_y)+\frac{\Delta t^2}{12}\partial\boldsymbol{\chi}\right]\Delta\boldsymbol{\phi}=-\Delta t(u\partial_x+v\partial_y)\phi^n, \tag{4.37}$$

where $\Delta\boldsymbol{\phi}=\phi^{n+1}-\phi^n$. For each time step, Equation (4.37) is solved for $\Delta\phi$, and the solution at $t=t^{n+1}$ is obtained via $\phi^{n+1}=\phi^n+\Delta\phi$.

Using the IRBFN method for the spatial discretization of Equation (4.37), the fully discrete TCN-IRBFN formulation for problem (4.18) can be then derived as follows.

$$\left\{1+\frac{\Delta t}{2}\left[u_i\boldsymbol{\psi}^T_{\partial_x}(\boldsymbol{x}_i)+v_i\boldsymbol{\psi}^T_{\partial_y}(\boldsymbol{x}_i)\right]+\frac{\Delta t^2}{12}\boldsymbol{\psi}^T_{\partial_{\boldsymbol{\chi}}}(\boldsymbol{x}_i)\right\}\Delta\boldsymbol{\phi}= \\ -\Delta t\left[u_i\boldsymbol{\psi}^T_{\partial_x}(\boldsymbol{x}_i)+v_i\boldsymbol{\psi}^T_{\partial_y}(\boldsymbol{x}_i)\right]\boldsymbol{\phi}^n,\quad i=1,\ldots,M, \tag{4.38}$$

where $\boldsymbol{\psi}_{\partial_{\boldsymbol{\chi}}}$, $\boldsymbol{\psi}_{\partial_x}$ and $\boldsymbol{\psi}_{\partial_y}$ are the IRBFN approximations to the differential operator $\partial_{\boldsymbol{\chi}}$, ∂_x and ∂_y, respectively; u_i and v_i are the components of the velocity field $\mathbf{v}$ at position $\boldsymbol{x}_i$ in x and y directions, respectively.

4.5 A new IRBFN-based approach to passive transport problems

The new meshless numerical approach to capturing moving interfaces in passive transport problems is built by bringing all ingredients previously presented together and consists of the following steps.

Step 1: Initialize the level set function $\phi(\boldsymbol{x})$ to be the signed distance to the interface as described by equation (4.1);

Step 2: Advance the level set function by solving the convective transport equation (4.2) for one time step using either SL-IRBFN or Taylor-IRBFN schemes

presented in section (4.3) and (4.4), respectively;

Step 3: Re-initialize the level set function that has just been calculated from the previous step to a signed distance function.

Step 4: The interface as the zero contour of the level set function has now been advanced one time step. Go back to step 2 for further evolution of the moving interface until the predefined time is reached.

4.5.1 Initialization

At time $t = 0$, the signed distance function in (4.1) is defined as the distance from the given collocation point $\boldsymbol{x}$ to the initial interface curve and the sign is chosen to be positive if the point is inside the curve, and negative if outside,

$$d(x_i, y_i, 0) = \pm min\|\boldsymbol{x} - \boldsymbol{x}_i\|, \boldsymbol{x}_i \in \Gamma_0, \tag{4.39}$$

where $\Gamma_0 = \Gamma(0)$ is the initial interface whose discrete representation is $\boldsymbol{x}_i$.

4.5.2 Advancing the level set function

The procedure for advancing the level set function with time by the SL-IRBFN scheme presented in section (4.3) consists of the following steps

Given $\mathbf{v}^0$, for any $\boldsymbol{x} \in \Omega$ and $n = 0, 1, \ldots, N$

1. Compute the departure points $\boldsymbol{x}_d$ at $t = t^n$ corresponding to the grid point $\boldsymbol{x} = \boldsymbol{x}_a$ at $t = t^{n+1}$ in (4.6,4.7) using the semi-discrete IRBFN-based scheme with Runge-Kutta method as described in section (3.3.3);

2. Calculate ϕ_d^n at the departure points $\boldsymbol{x}_d$ by interpolating the known values of $\phi(\boldsymbol{x}, t^n)$ at the grid points using the IRBFN method;

3. Advance $\phi(\boldsymbol{x}, t)$ one time step by assigning $\phi^{n+1} = \phi_d^n$

4.5.3 Calculation of ϕ at departure points

As mentioned earlier, since the departure points $\boldsymbol{x}_d$ do not coincide with the grid points, the values of ϕ_d^n at those points are obtained by interpolation. The IRBFN formulation (3.30) is used for this purpose as follows.

$$\boldsymbol{\phi}(\boldsymbol{x}, t) = \boldsymbol{g}^T(\boldsymbol{x})\, \mathbf{G}^{-1}\, \boldsymbol{\phi}(t) \tag{4.40}$$

where $\boldsymbol{\phi}(t)$ is the values of $\phi(\boldsymbol{x}, t)$ at all data points $\boldsymbol{x}$ at time t. The values of ϕ_d^n at the departure points $\boldsymbol{x}_d$ are obtained by IRBFN interpolation as follows.

$$\boldsymbol{\phi}(\boldsymbol{x} = \boldsymbol{x}_d, t = t^n) = \boldsymbol{g}^T(\boldsymbol{x} = \boldsymbol{x}_d)\, \mathbf{G}^{-1}\, \boldsymbol{\phi}(t = t^n). \tag{4.41}$$

It is noted that $\mathbf{G}^{-1}$ is calculated once, and thus only matrix-vector operations are performed at each time step for interpolation.

4.5.4 Re-initialization and mass correction

The reinitialization step is done by solving Equation (4.3) to steady-state using the semi-implicit IRBFN-based scheme with the fourth-order Runge-Kutta presented in 3.3.3. The mass correction is then performed to prevent any losses of mass as follows. Suppose that after advancing the level set function at time step $t = t^{n+1}$, one gets the moving interface Γ that bounds the domain $\Omega_2 = \boldsymbol{x} \in \Omega : \phi < 0$. To correct the area of Ω_2, one changes the zero level set to certain neighboring isoline based on the fact that it has almost the same shape since ϕ is a distance function. This can be done by simply moving the level set function upward or downward by an amount of c_ϕ, where $|c_\phi|$ is the distance

between old and new zero-level sets

$$\phi^{new} = \phi - c_\phi, \tag{4.42}$$

where ϕ^{new} is the new (raised or lowered) level set function, $\Omega_2^{new} = \{\boldsymbol{x} \in \Omega : \phi^{new} < 0\}$. The well-known formula for the first variation of a volume integral (Cuvelier and Schulkes, 1990) is then used to calculate c_ϕ as follows.

$$S_{exact} - S(\Omega) = \int_{\Omega_2^{new}} \mathrm{d}\Omega - \int_{\Omega_2} \mathrm{d}\Omega = \int_\Gamma (c_\phi \mathbf{n}) \cdot \mathbf{n} \mathrm{d}\Gamma + \mathcal{O}(c_\phi^2), \tag{4.43}$$

or

$$S_{exact} - S(\Omega_2) = c_\phi \int_\Gamma \mathrm{d}\Gamma + \mathcal{O}(c_\phi^2), \tag{4.44}$$

where S_{exact} is the given exact area of the region, and $S(\Omega_2)$ is the area of Ω_2. It follows that

$$c_\phi = \frac{S_{exact} - S_{\Omega_2}}{L(\Gamma)}, \tag{4.45}$$

in which $L(\Gamma)$ is the length of the interface Γ. It is noted from Equation (4.45) that if $S_{exact} > S(\Omega_2)$ then $c_\phi > 0$, and the level set function ϕ is to be lowered, meaning that the domain Ω_2 expands. Otherwise, the domain shrinks. In both cases, the level set function is corrected accordingly. In this way, the reinitialization procedure prevents an accumulation of numerical errors in a long run as shown in the numerical results.

4.6 Numerical results

Some numerical tests are performed in this section to verify the individual numerical schemes as well as the new meshless approach presented in the previous sections. The first test is for checking the capability of the SL-IRBFN and

Taylor-IRBFN schemes in dealing with shock wave propagation. The next two problems provide basic tests on the accuracy and efficiency for the new meshless approach to capturing moving interfaces of a solid circle that translates and rotates in a cavity. The present approach is then demonstrated with the simulation of more complicated passive transport problems in which bubbles are moving, stretching and merging together in a divergence-free shear flow.

4.6.1 Test 1 - Convective transport problems

Test problem 1.1

Consider the propagation of a cosine profile governed by the following convective transport equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad x \in \Omega \tag{4.46}$$

with the following initial condition

$$u(x,0) = \begin{cases} \frac{1}{2}(1+\cos(\pi(x-x_0)/\sigma)) & |x-x_0| \leq \sigma \\ 0 & \text{otherwise} \end{cases} \tag{4.47}$$

where $c = 1$ is the propagation speed.

The steep profile of the solution is well captured by the SL-IRBFN scheme as shown in Figure 4.1. In fact, with $N = 61$ regularly located points and $CFL = 0.5$, the numerical solutions are accurate up to 3 digits after the decimal point in the steep region whereas the absolute errors by the scheme can be of order 10^{-5} in the flat regions. In addition, it can be seen in Figure 4.1 that there are no severe errors found right before and after the shock as observed in Lax-Wendroff and second-order Taylor-Galerkin schemes (Donea and Huerta, 2003).

A comparison on accuracy and stability of the TCN-IRBFN scheme using MQ and TPS basis functions is shown in Figure 4.2. As can be seen from the figure, the MQ-based scheme is more accurate and stable than its counterpart TPS-based scheme. For this test, the β parameter of MQ-RBF is set to 1.0. Figure 4.3 shows a comparison on accuracy and stability of the TE-IRBFN schemes using MQ and TPS basis functions. It is observed from this test that TE-IRBFN scheme using MQ-RBF again yields better solution in terms of both accuracy and stability than its TPS-based counterpart.

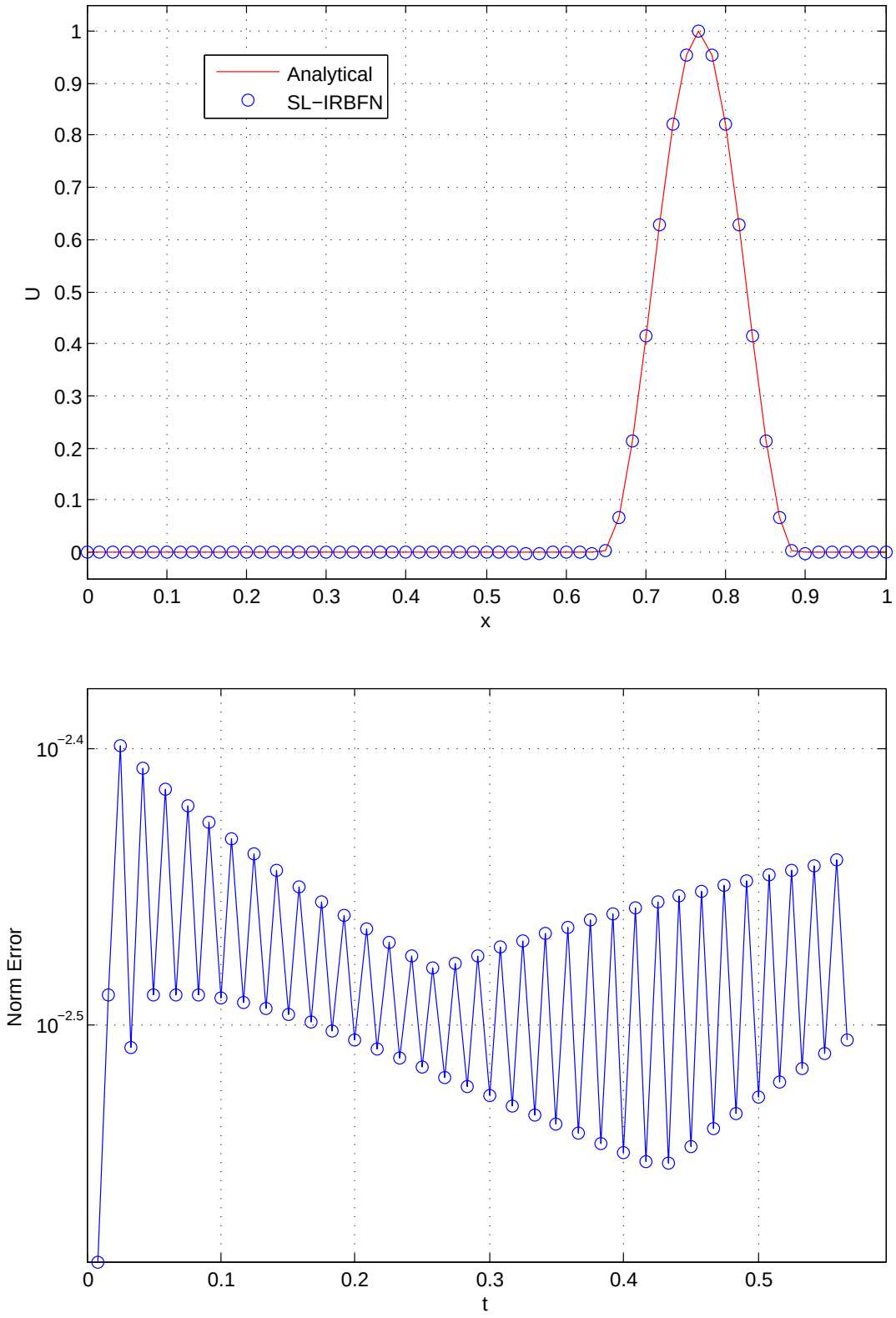

Figure 4.1: Numerical solution (top) and its norm error (bottom) by the SL-IRBFN scheme for Test 1.1.

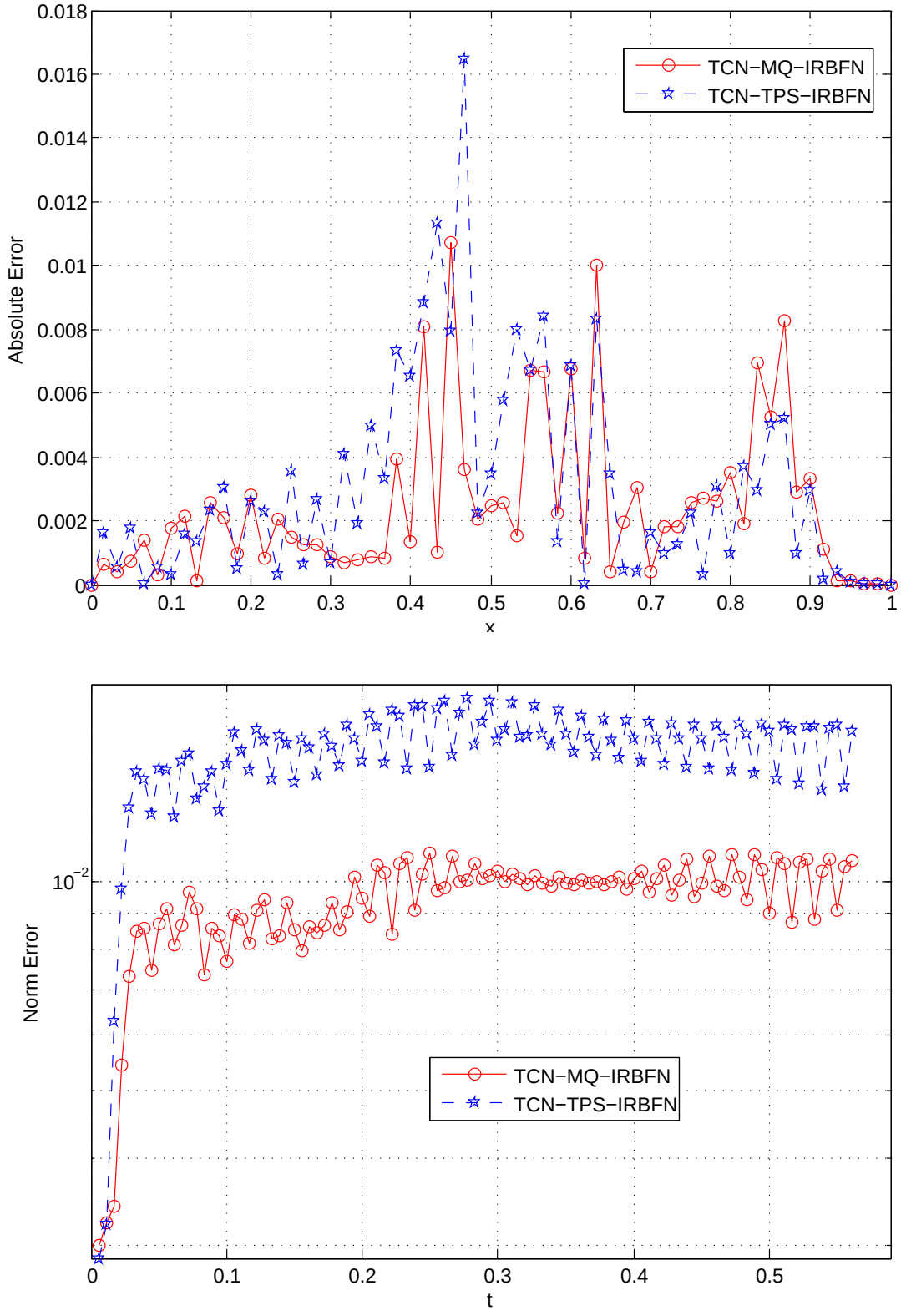

Figure 4.2: Comparison between the two variants of the TCN-IRBFN scheme (using MQ and TPS basis functions) in Test 1.1: (Top) Absolute error at the last time step; (Bottom) L_∞-norm error with respect to time.

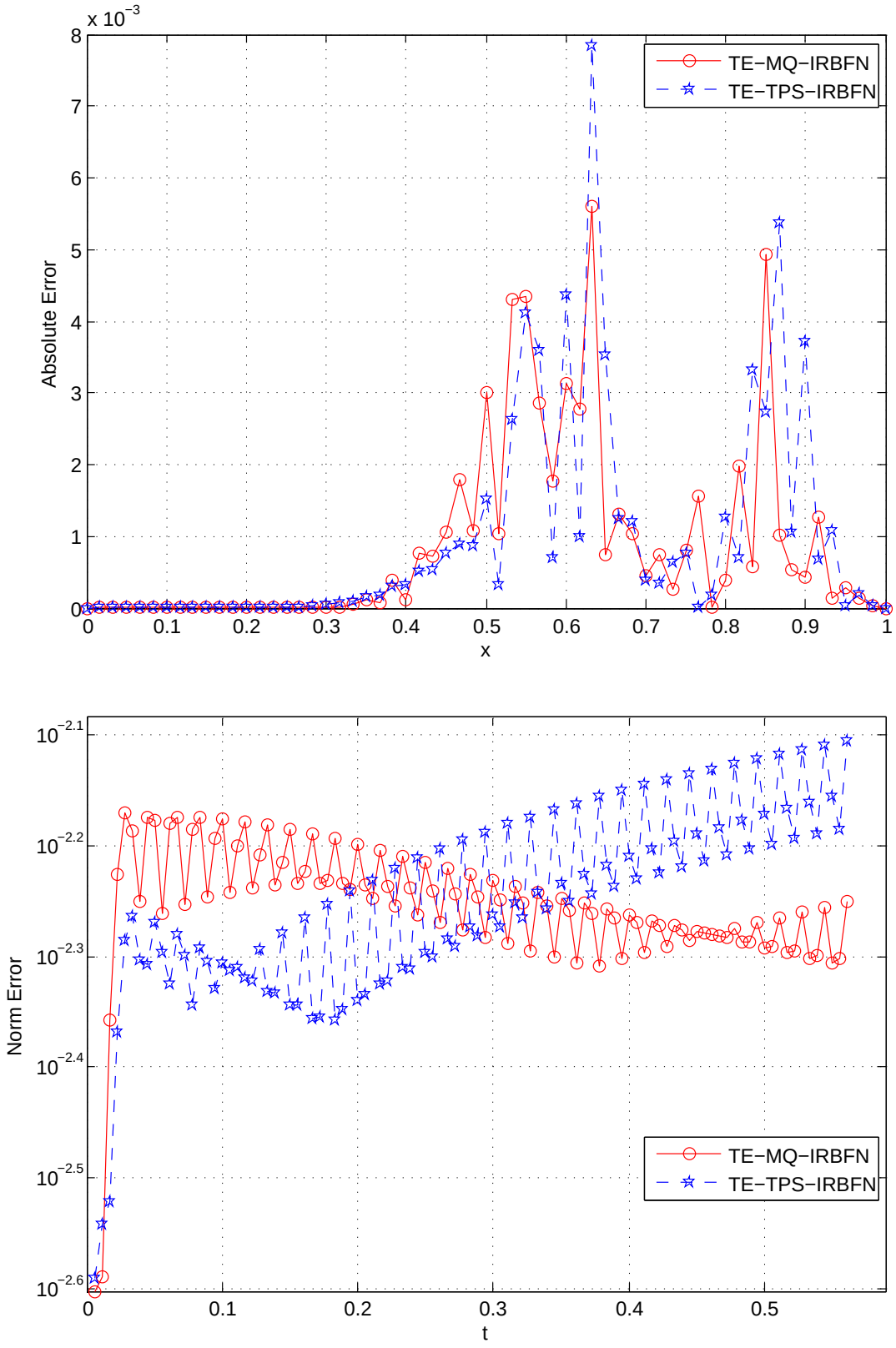

Figure 4.3: Comparison between the two variants of the TE-IRBFN scheme (using MQ and TPS basis functions) in Test 1.1: Absolute error at the last time step (top); L_∞-norm error with respect to time (bottom).

Test problem 1.2

Consider a convective transport equation

$$\frac{\partial u}{\partial t} - (\sin x)\frac{\partial u}{\partial x} = 0, \quad x \in \Omega,\ t \in (0, \frac{\pi}{2}], \tag{4.48}$$

subject to the initial condition

$$u(x, 0) = \sin x \tag{4.49}$$

The analytical solution to the problem is

$$u(x, t) = \sin\left(2\tan^{-1}\left(\exp^t \tan\frac{x}{2}\right)\right), \tag{4.50}$$

which develops sharp layers near the end points $x = 0$ and $x = 2\pi$. The problem is solved up to time $t = \pi/2$ by the Taylor-IRBFN scheme using regularly located points with point density as large as 1/10. Figure 4.4 shows the analytical and numerical solution to the convective transport equation in Test 1.2 by the TCN-IRBFN scheme. As can be seen from the figure, sharp layers near the end points are well resolved by the present scheme even with rather large time steps ($\Delta t = 0.1$ or CFL=1). In this case, the time-step size depends on the accuracy requirement, not on stability.

For the purpose of investigating the effect of CFL numbers on the accuracy and stability of the new numerical schemes, the test problem is solved by the SL-IRBFN and TE-IRBFN schemes using a set of different CFL numbers. The root mean square errors corresponding to the CFL numbers are calculated as follows.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{nt}(u_n - u_e)^2}{nt}}, \tag{4.51}$$

where u_n and u_e are the numerical and exact solutions, respectively; nt is the

total number of time steps. As can be seen in Figure 4.5, for various CFL number widely ranging from 0.5 to 4, the root mean square errors are bounded to $\mathcal{O}(10^{-3})$ for the SL-IRBFN scheme, and $\mathcal{O}(10^{-4})$ for the TE-IRBFN scheme. This verifies the accuracy and stability of the two new numerical schemes. It is observed from the test that on the one hand, the TE-IRBFN scheme is not sensitive to CFL number, meaning that the scheme works fine with high CFL number. On the other hand, it is also noted that unlike the SL-IRBFN scheme where the value of unknown at each time step can be found explicitly, the Taylor-IRBFN scheme requires a solution of a system of equations at each time step.

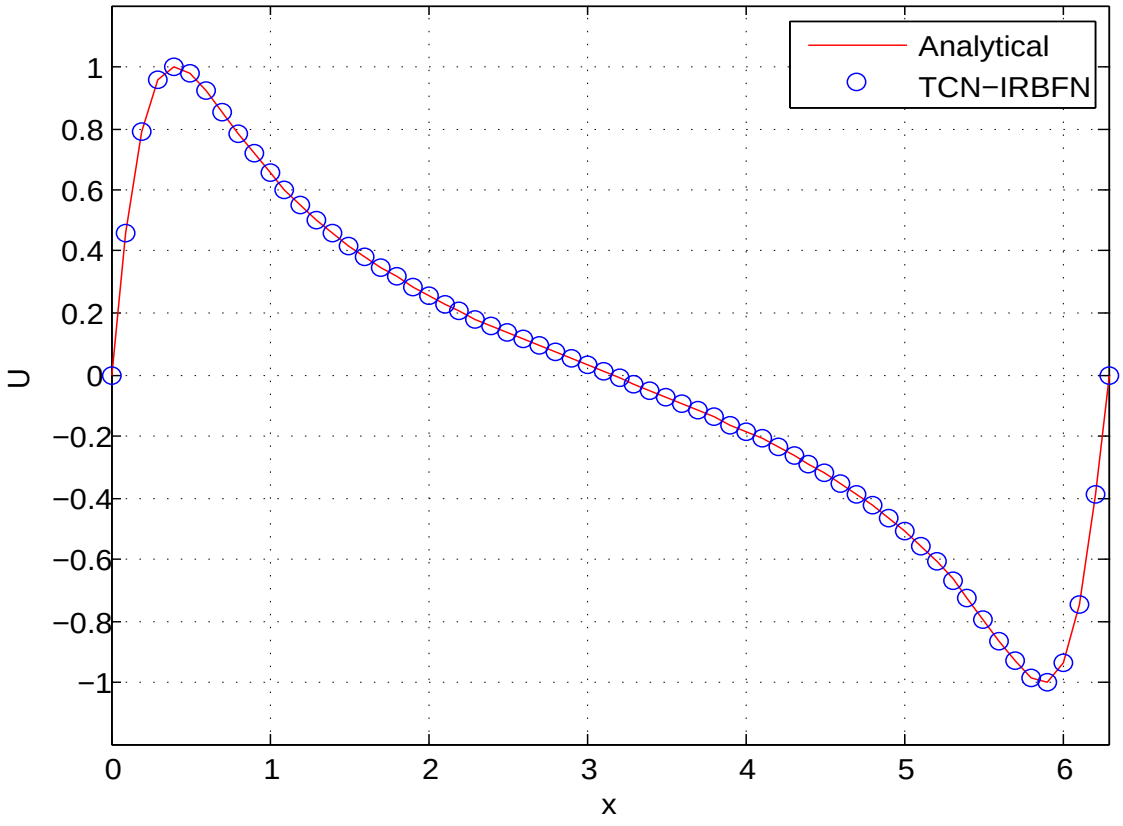

Figure 4.4: Numerical solution to Test 1.2 by the TCN-IRBFN scheme. The analytical (solid line) and numerical solution ("o" symbol) are plotted at $t = \pi/2$ with CFL=1 and $dx = 1/10$.

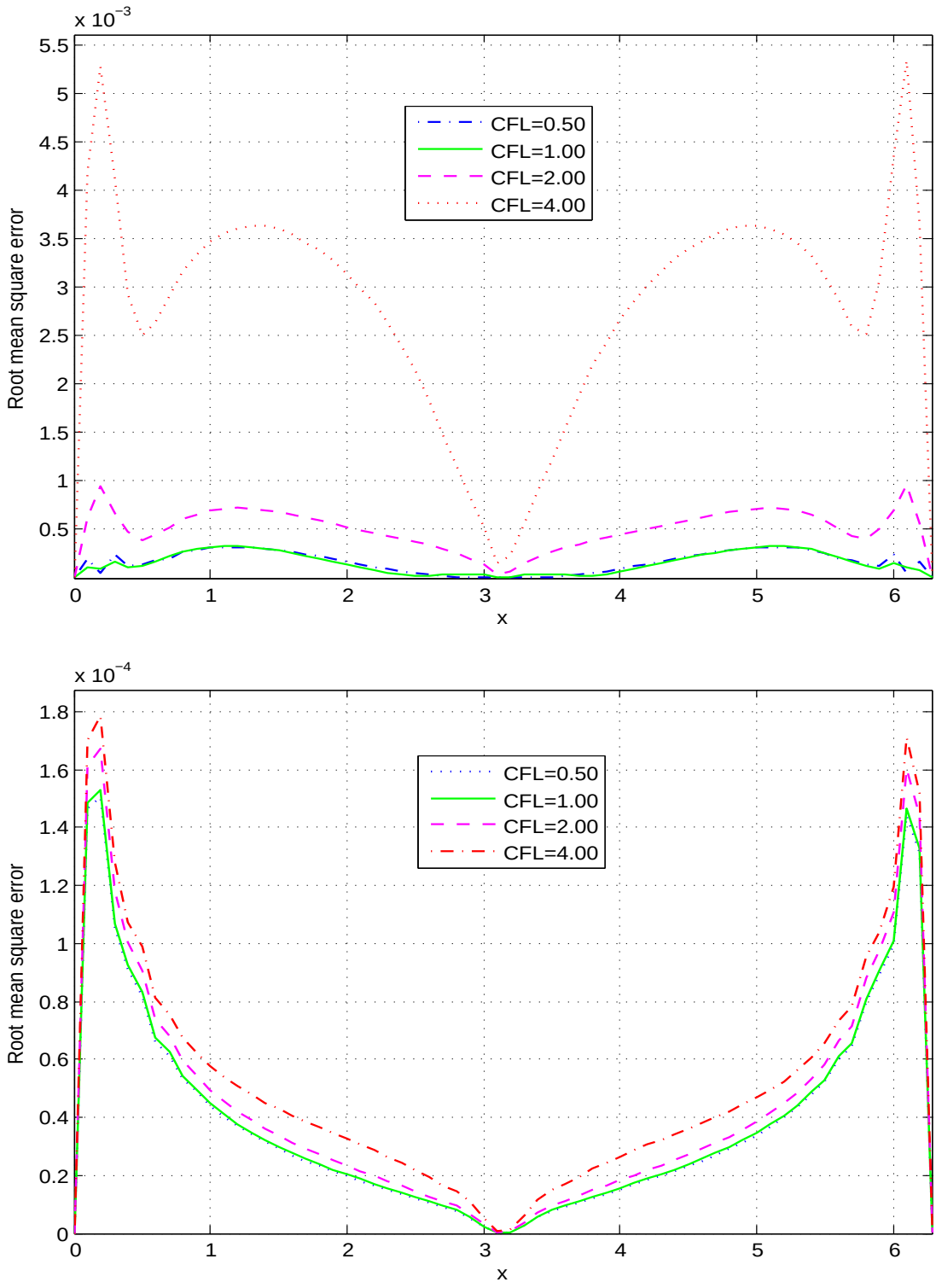

Figure 4.5: Numerical investigation on the accuracy and stability of the present numerical schemes for Test 1.2: Root mean square errors corresponding to various values of the CFL number within a rather wide range (0.5-4) are bounded to $\mathcal{O}(10^{-3})$ for the SL-IRBFN scheme (top), and $\mathcal{O}(10^{-4})$ for the TE-IRBFN scheme (bottom).

4.6.2 Test 2 - Solid body translation

In this test problem, a circle of radius 0.5, initially centered at (-0.75,0), translates to the right due to an external velocity field $\mathbf{v} = (u, v) = (1, 0)$. The objective of the test is to check the accuracy and stability of the new meshless approach in capturing the moving interface. The circle is translated until time $t = 1.0$, and the percentage change in the area is calculated for the verification purpose.

The problem is solved by the present meshless approach with the point density $dx = 1/15$ and time-step size $dt = 0.0667$. The level set function is advanced in time by the Taylor-IRBFN scheme.

Figure 4.6 shows zero contours of the level set function at different points in time by the TCN-IRBFN scheme. At each time step of interest, the zero contour of the level set function which is the moving interface is extracted using standard contouring algorithm, and the corresponding area of the circle at those time steps are calculated and compared to the exact area of the original circle. As can be seen from the figure, the circle is well captured by the present approach at different points in time.

Figure 4.7 shows the percentage change in area at different points in time of interest. It can be seen from the figure that the present approach with the TCN-IRBFN scheme is not only able to accurately capture the moving interface but also stable with the error bounded within $\mathcal{O}(10^{-5})$ over the computational time domain. The percentage changes in area at different points in time in this test show that with a coarser point density ($dx = 1/15$) and a larger time step ($dt = 0.0667$), the present meshless approach (using either TE-IRBFN or TCN-IRBFN scheme for solving the level set function) gives more accurate solutions ($\%error \sim \mathcal{O}(10^{-5}) - \mathcal{O}(10^{-3})$) than those resulted from the mesh-based level set scheme ($\%error \sim \mathcal{O}(10^{-2})$) with denser discretization in space ($dx = 1/80$) and time ($CFL = 0.9$) (Sethian, 1999). Figure 4.7 also shows that the TCN-

IRBFN scheme yields better result than the TE-IRBFN scheme for this test problem.

It is noted that for such a simple velocity field in this test, the reinitalization step is not needed. In fact, only a mass correction presented in Section § 4.5.4 is performed at each time step in this test. Without the reinitialization step, the present approach is still highly accurate and stable. This verifies the efficiency of the new meshless approach for this basic test problem.

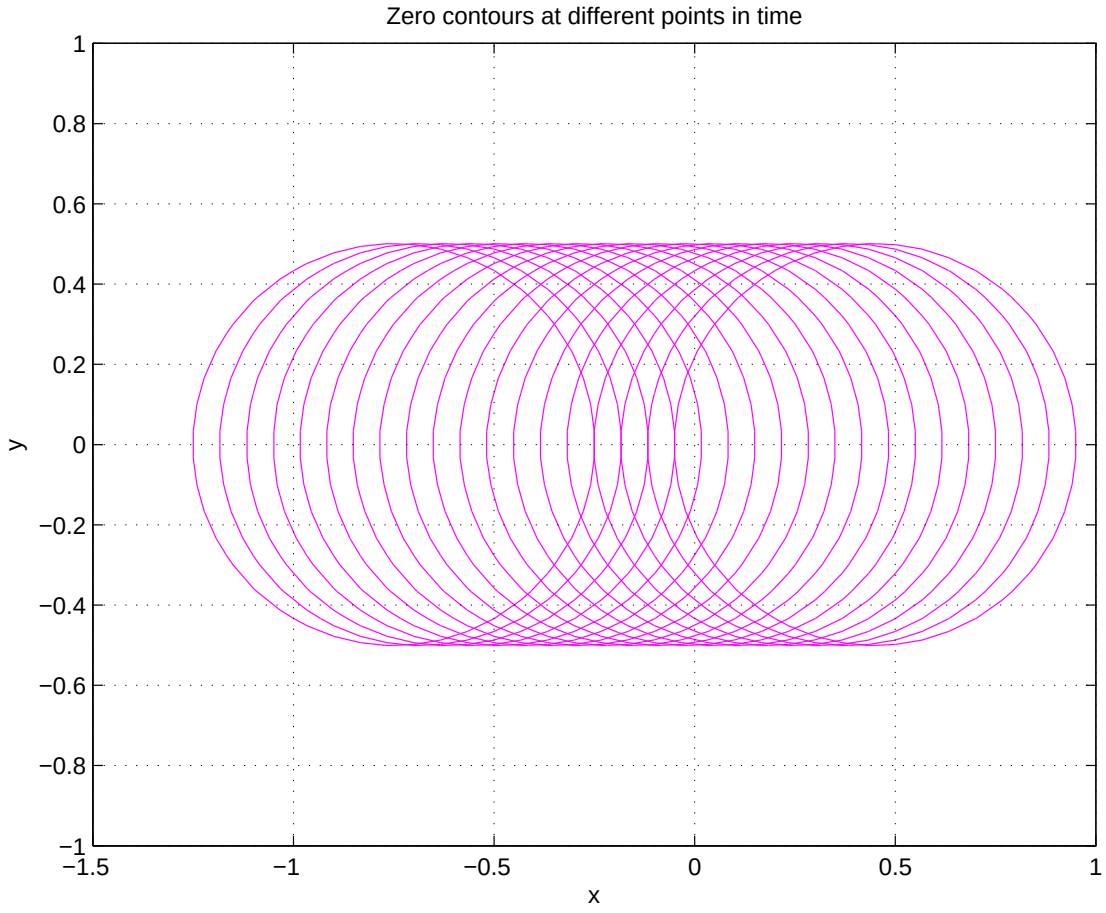

Figure 4.6: Zero contours of the level set function at different points in time by TCN-IRBFN scheme in Test 2. Although using a rather coarse point density ($dx = dy = 1/15$) and large time-step size ($dt = 0.0667$), the present approach is still able to exactly reconstruct the moving circle at the points in time of interest.

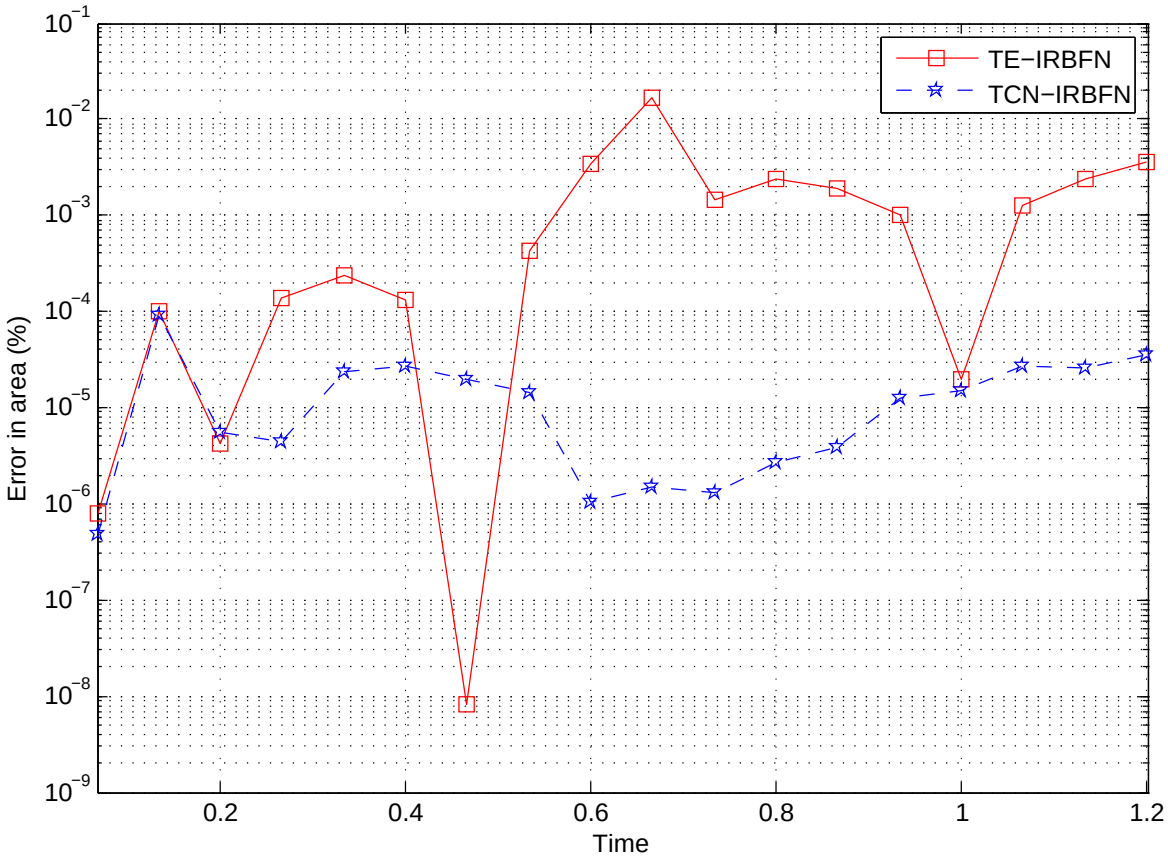

Figure 4.7: Percentage errors in area at different points in time in Test 2. With a coarser point density ($dx = 1/15$) and a larger time step ($CFL = 1.0$), the present meshless approach (using either using TE-IRBFN or TCN-IRBFN scheme for solving the level set function) gives more accurate solutions ($\%error \sim \mathcal{O}(10^{-5}) - \mathcal{O}(10^{-3})$) than those resulted from the mesh-based level set scheme ($\%error \sim \mathcal{O}(10^{-2})$) with denser grid points (81×81) and time ($CFL = 0.9$) (Sethian, 1999). The numerical result also shows that the TCN-IRBFN scheme is more accurate and stable than the TE-IRBFN scheme for this test problem.

4.6.3 Test 3 - Rotation of a solid body

Consider the rotation of a circle of radius $r = 0.5$ initially centered at $(-0.75, 0)$ in a vortex flow with velocity field $(u, v) = (-y, x)$. It is noted that with such a velocity field, the circle rotates around the coordinate's origin $(0, 0)$ without any deformations. In other words, the circle is considered to be a solid body. An half cycle of rotation is performed by the present meshless approach, and the percentage change in area of the circle during its motion is calculated.

The problem is solved by the present meshless approach with point density $dx = dy = 1/12$ and time-step size $dt = \pi/100 = 0.0314$. The level set function is advanced in time by the SL-IRBFN scheme (described in Section §4.3) in which the IRBFN semi-discrete scheme (Mai-Cao and Tran-Cong, 2005) with the fourth-order Runge-Kutta is used to track particles that arrive at the grid points backward to their departure points over a single time step. The function values at those departure points are then obtained by interpolation with TPS-IRBFN formulation.

Figure 4.8 shows the zero contours of the level set function at different points in time. Although using a rather coarse point density and normal time-step size, the present approach still exactly reconstructs the moving circle at the points in time of interest. Figure 4.9 presents the percentage errors in area at different points in time. With a very coarse point density ($dx = 1/12$) and a large time-step size ($dt = 0.0314$), the present meshless approach, using the SL-IRBFN scheme for solving the level set function, gives the solution after an half cycle of rotation with the change/error in area of 0.006970%. In (Sethian, 1999), the same test problem was performed with different grid sizes and the percentage error in area was reported to be 0.09758% with the grid size of 161×161. No conclusion on which (meshless or mesh-based scheme) is better is made for this particular test problem since there is no information about the time-step size used in (Sethian, 1999).

It is noted that the numerical solution for this test is obtained without the reinitialization step. In fact, only a mass correction step is performed at each time step. The numerical result shows that the present approach is accurate and stable with the error bounded within $\mathcal{O}(10^{-4}) - \mathcal{O}(10^{-3})$. It can be concluded that for such a simple velocity field like the one in this test or in Test 2, the reinitialization step is not required provided that a mass correction is performed at each time step. Since no PDEs are solved in the mass correction procedure, saving of computational time is achieved.

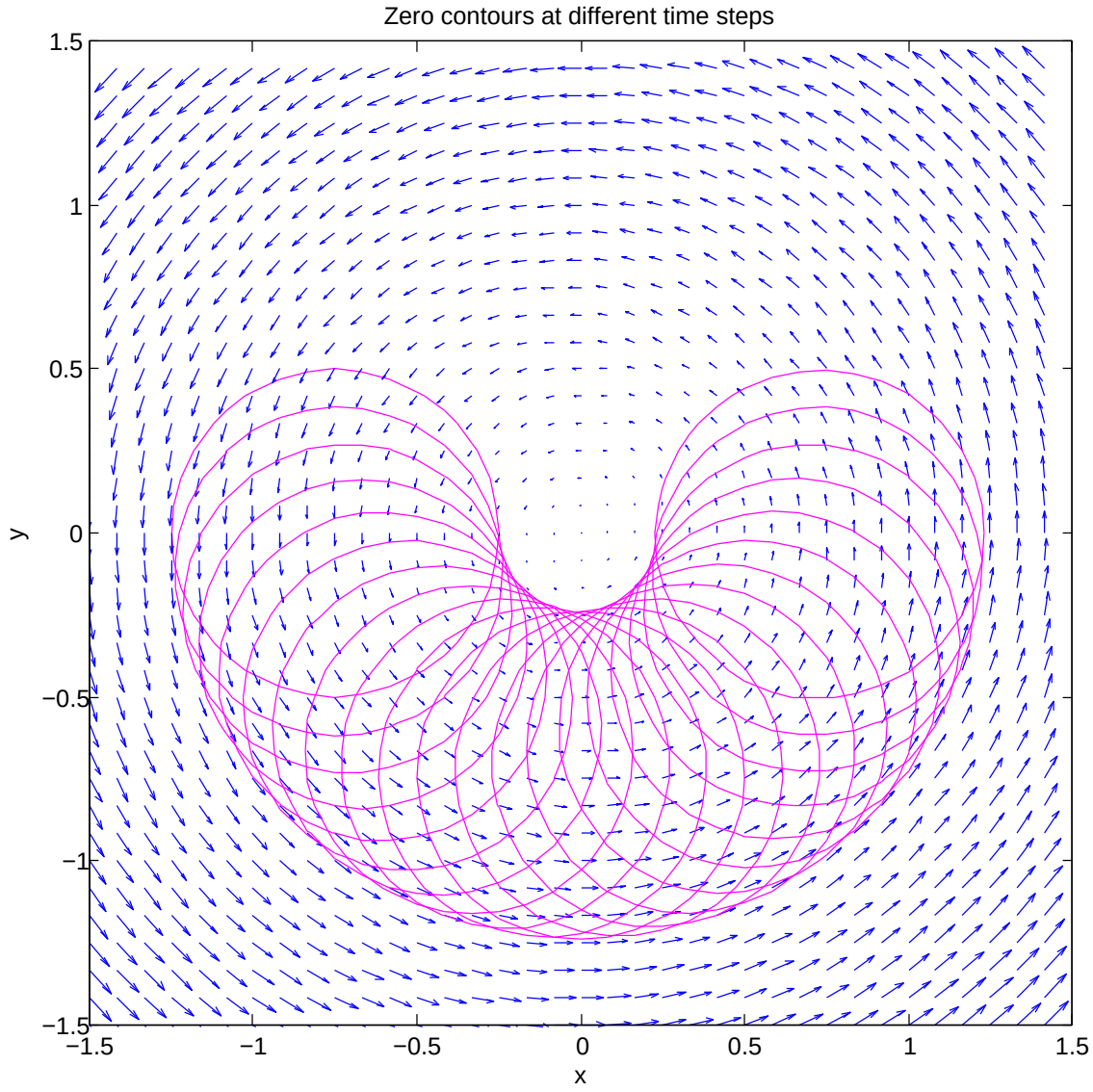

Figure 4.8: Zero contours of the level set function at different points in time in Test 3. Although using a rather coarse point density ($dx = dy = 1/12$) and normal time step ($dt = 0.0314$), the present approach still exactly reconstructs the moving circle at the points in time of interest.

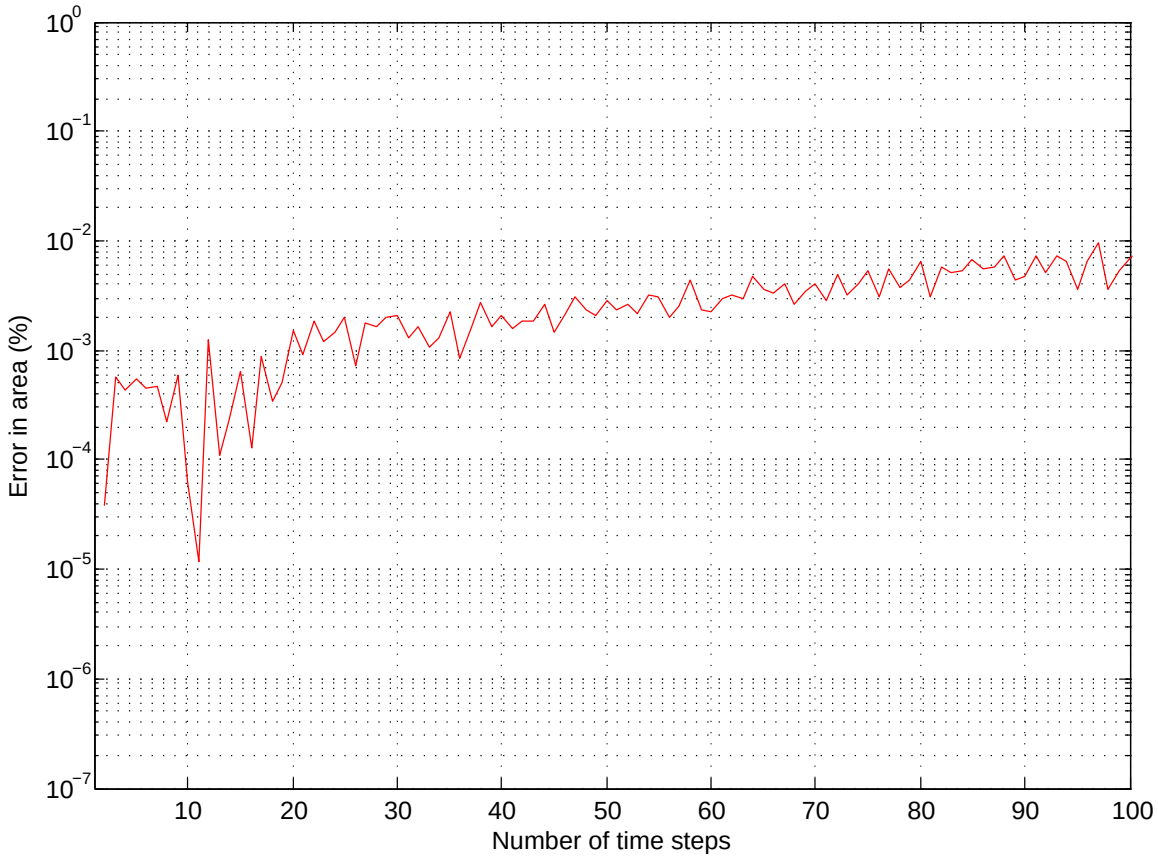

Figure 4.9: Percentage errors in area at different points in time in Test 3. With a rather coarse point density ($dx = dy = 1/12$) and normal time-step size ($dt = 0.0314$), the present meshless approach using the SL-IRBFN scheme for advancing the level set function) is quite accurate and stable.

4.6.4 Test 4 - Passive transport of a bubble in a shear flow

In this problem (this example and the following one represent more serious application of the present approach), a bubble with the radius of 0.15, initially centered at $(0.5, 0.7)$ moves and deforms in a shear flow with the divergence-free velocity field $\mathbf{v} = (u, v)$ defined as follows.

$$u = -\sin \pi x \cos \pi y, \quad 0 \leq x, y \leq 1, \ t \geq 0, \tag{4.52}$$

$$v = \cos \pi x \sin \pi y, \quad 0 \leq x, y \leq 1, \ t \geq 0. \tag{4.53}$$

The problem is solved by the present meshless approach with the point density $dx = dy = 1/50$ and time-step size $dt = 0.01$. The time-step size dt is chosen so as to satisfy the Courant-Friedreichs-Levy condition (Osher and Fedkiw, 2003).

$$\Delta t \times max \left\{ \frac{|u|}{dx} + \frac{|v|}{dy} \right\} = \text{CFL}, \tag{4.54}$$

where the CFL number is chosen to be unity.

In this problem, the level set function is advanced in time by the SL-IRBFN scheme in which the values of the level set function at departure points are obtained via interpolation by the TPS-IRBFN formulation. At the end of each time step, the reinitialization procedure is done by solving equation (4.3) to steady-state using the semi-implicit IRBFN-based scheme with the fourth-order Runge-Kutta (Mai-Cao and Tran-Cong, 2005). For the purpose of investigating the effect of the mass correction on the accuracy and stability of the present approach, the reinitialization procedure is done with and without mass correction. The area of the bubble in motion is calculated at each time step and compared to the original area (πR^2). The error in area of the bubble in motion throughout the simulation time is then used to check the stability of the present approach.

Figures 4.10-4.13 show the zero contours (left) and level set function (right) at different points in time.

Figure 4.14 shows a comparison on the percentage error in area of the bubble resulted from the present approach with and without mass correction for this problem. As can be seen from the figure, the accuracy of the numerical solution is improved significantly with mass correction. In addition, the percentage error of the bubble is bounded within $\mathcal{O}(10^{-5})$-$\mathcal{O}(10^{-4})$ indicating the good stability of the present approach with mass correction.

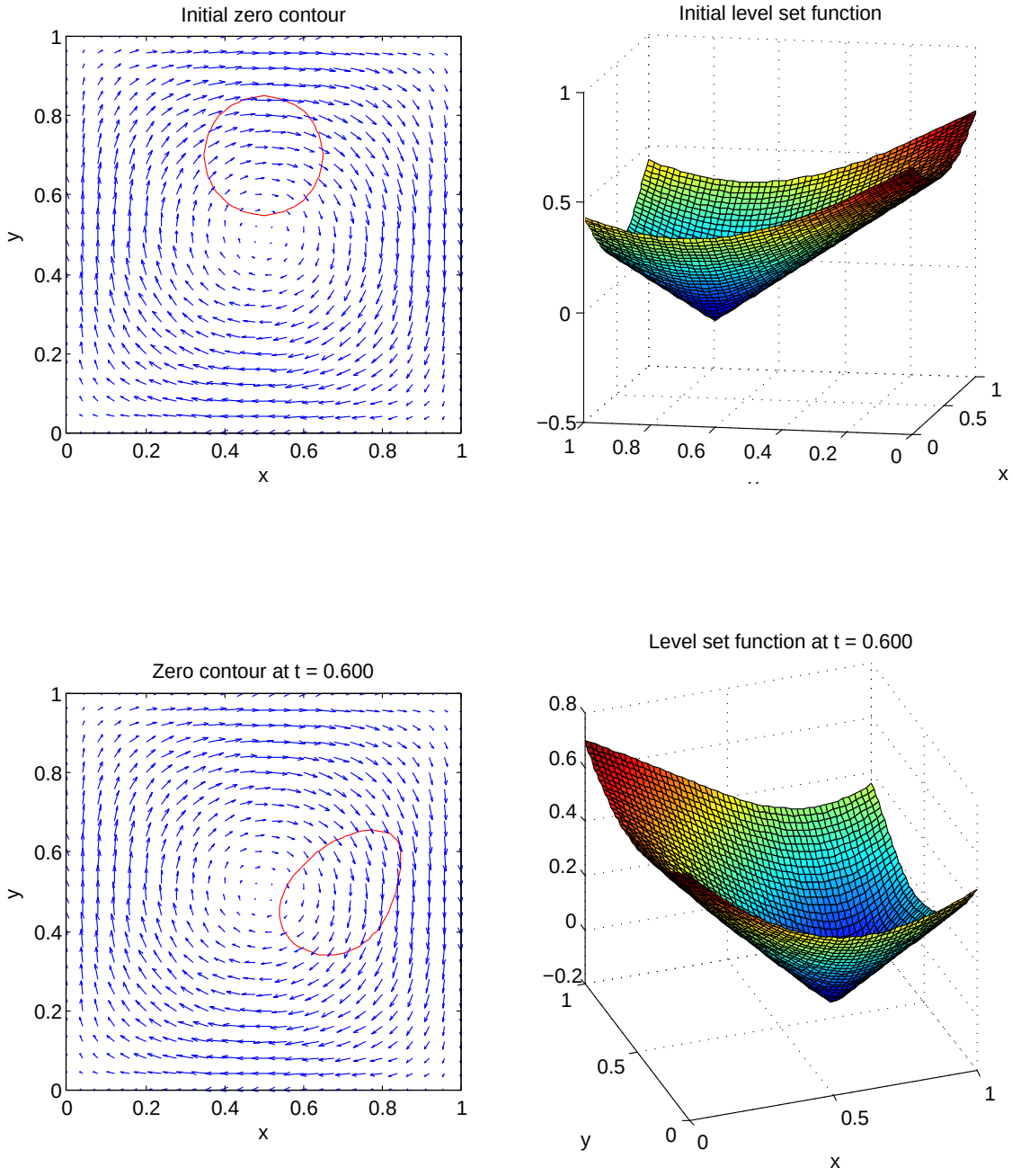

Figure 4.10: The zero contour and the level set function at $t = 0$ (top) and $t = 0.60$ (bottom) in Test 4.

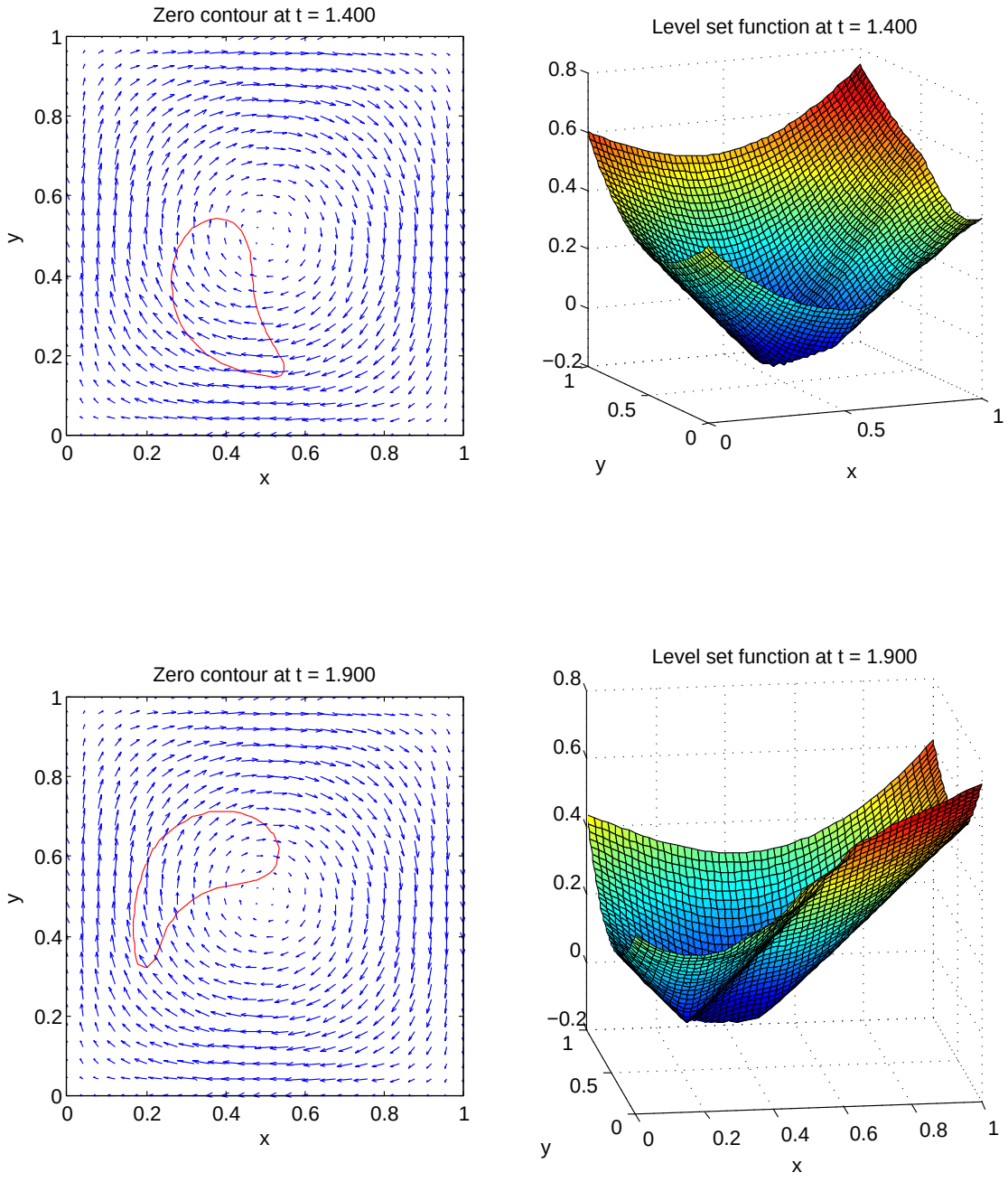

Figure 4.11: The zero contour and the level set function at $t = 1.40$ (top) and $t = 1.90$ (bottom) in Test 4.

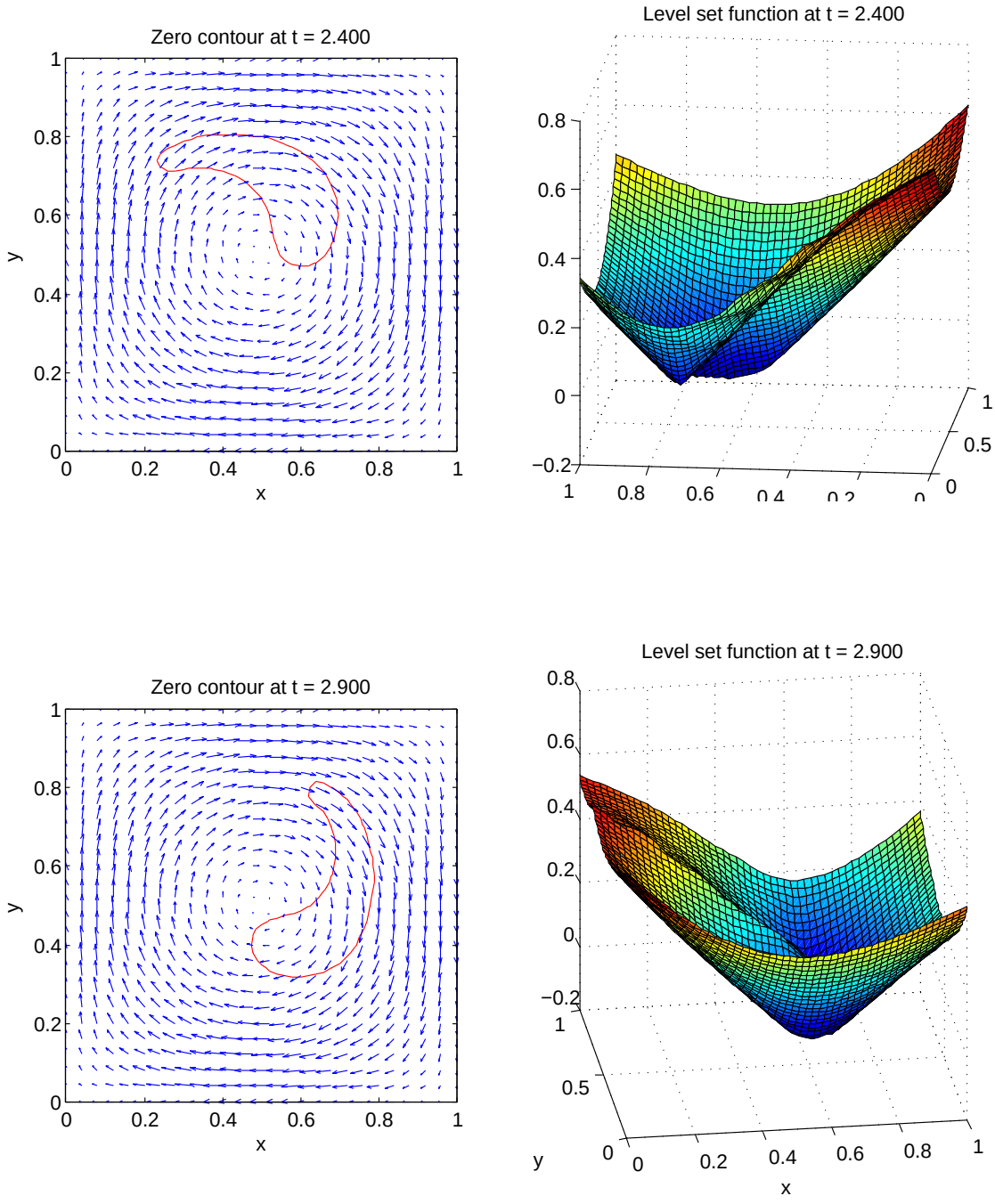

Figure 4.12: The zero contour and the level set function at $t = 2.40$ (top) and $t = 2.90$ (bottom) in Test 4.

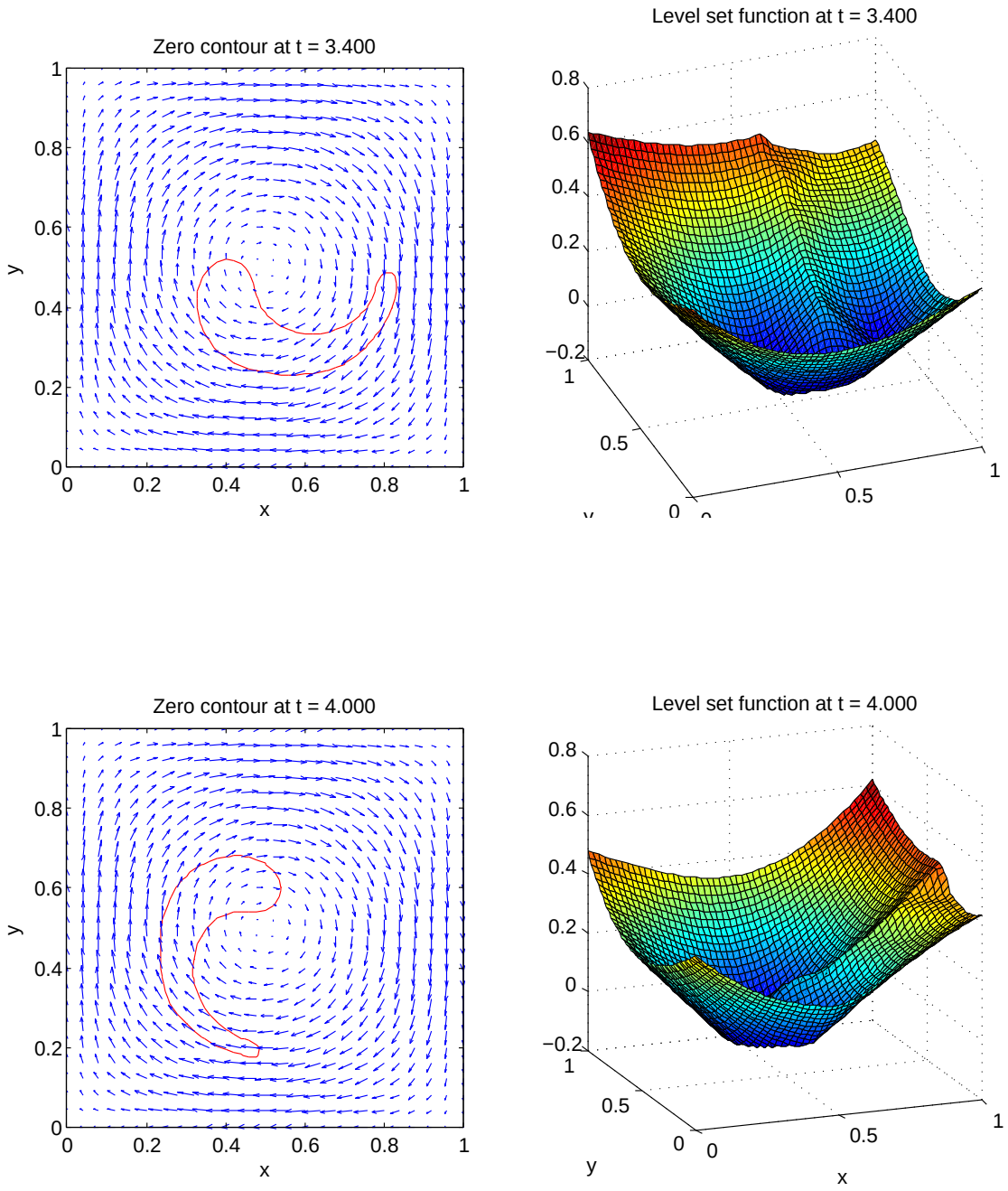

Figure 4.13: The zero contour and the level set function at $t = 3.40$ (top) and $t = 4.00$ (bottom) in Test 4.

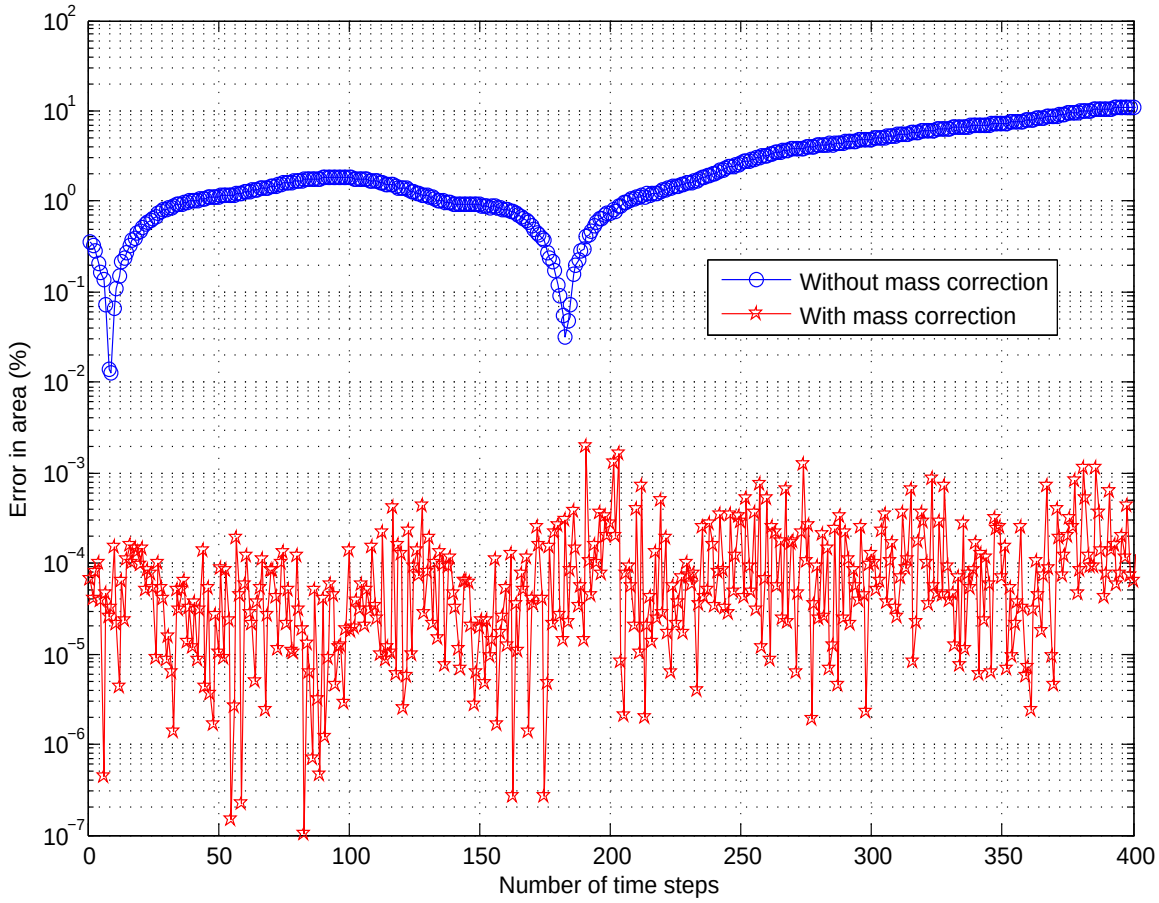

Figure 4.14: Comparison on the percentage error in area of the bubble resulted from the present approach with and without mass correction in Test 4. The accuracy of the numerical solution is improved significantly with mass correction. The error bounded within $\mathcal{O}(10^{-5})$-$\mathcal{O}(10^{-4})$ indicates that the present approach with mass correction is stable for this test problem.

4.6.5 Test 5 - Passive transport of four bubbles in a shear flow

The purpose of this example is to demonstrate the ability of the new meshless approach in dealing with the topology changes of interfaces in passive transport problems. Four bubbles, each having a radius of $R = 1/6$, are initially centered as shown in the top of Figure 4.15. The bubbles move in a domain of $(-1, 1) \times (-1, 1)$ where there exists a shear flow with the velocity field defined as follows.

$$u = -y \frac{max\left\{1 - (1 - x^2 - y^2)^4, 0\right\}}{8\left(x^2 + y^2\right)} \tag{4.55}$$

$$v = x \frac{max\left\{1 - (1 - x^2 - y^2)^4, 0\right\}}{8\left(x^2 + y^2\right)} \tag{4.56}$$

The problem is solved by the present meshless approach using SL-IRBFN scheme with the point density $dx = dy = 1/60$ and time-step size $dt = 0.0678$. The time-step size dt is so chosen to satisfy the Courant-Friedreichs-Levy condition as in the previous test problem. For the purpose of investigating the effect of the mass correction on the accuracy and stability of the present approach, the reinitialization procedure is done with and without mass correction. The total area of the bubbles in motion are calculated at each time step and compared to the original value ($a_0 = 4\pi R^2$).

Figures 4.15-4.17 show the zero contours (left) and level set function (right) at different points in time. Figure 4.18 shows a comparison on the percentage error in area of the bubble resulted from the present approach with and without mass correction for this test problem. As can be seen from the figure, the accuracy of the numerical solution is improved significantly with mass correction. In addition, the percentage error of the bubble is bounded within $\mathcal{O}(10^{-5})$-$\mathcal{O}(10^{-3})$ indicating the good stability of the present approach with mass correction.

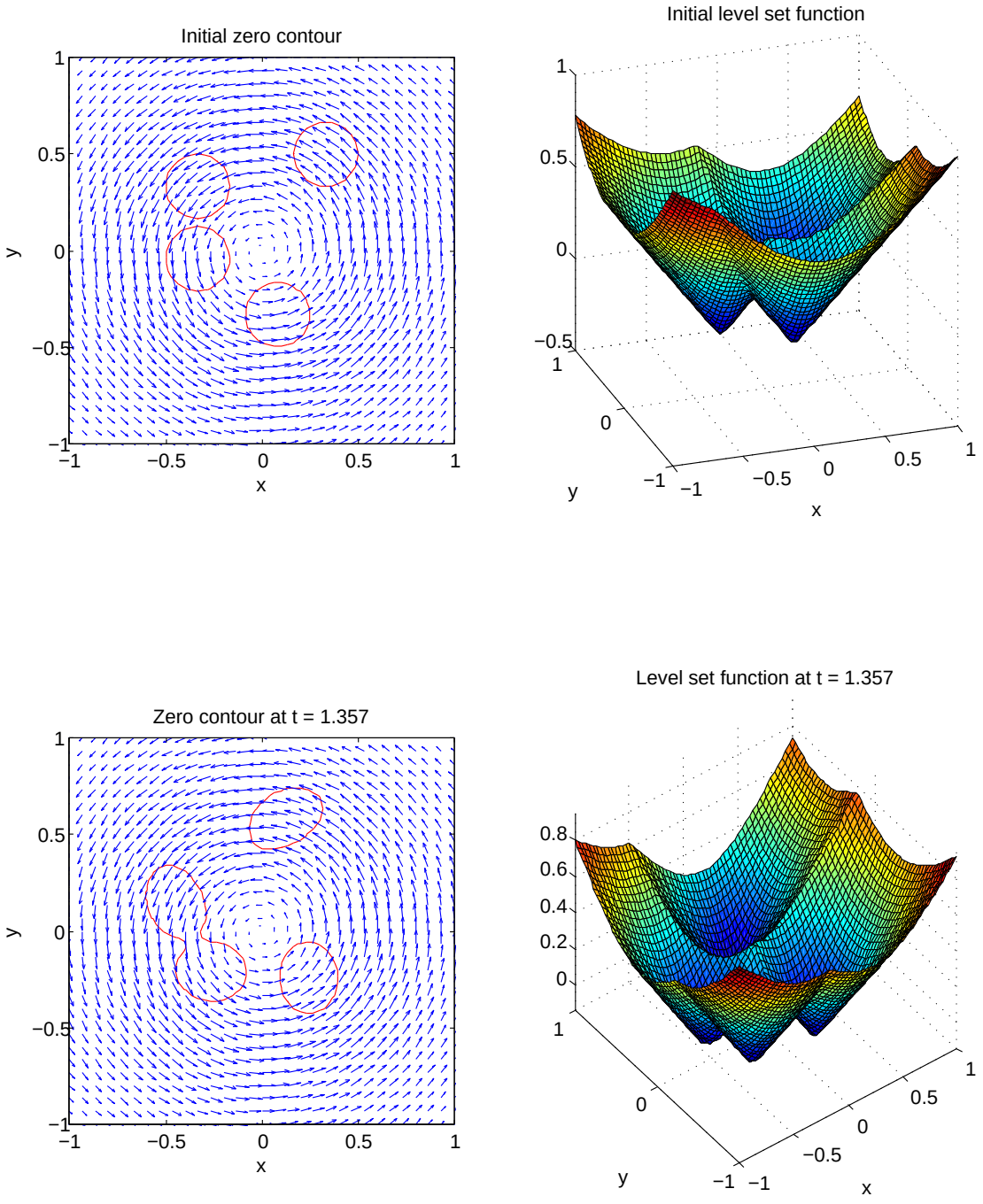

Figure 4.15: Zero contours and the level set function at t=0 (top) and t=1.357 (bottom) in Test 5.

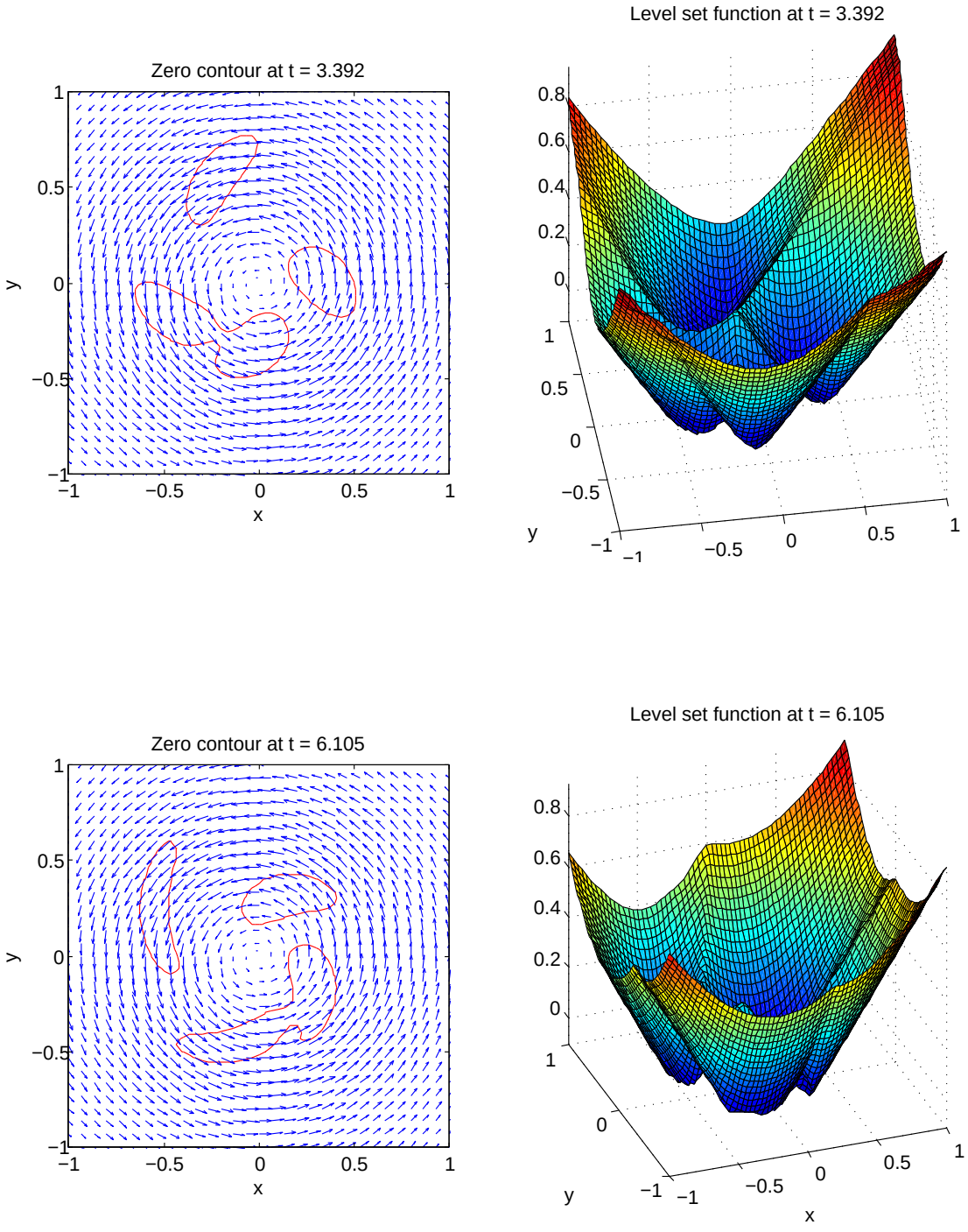

Figure 4.16: Zero contours and the level set function at t=3.392 (top) and t=6.105 (bottom) in Test 5.

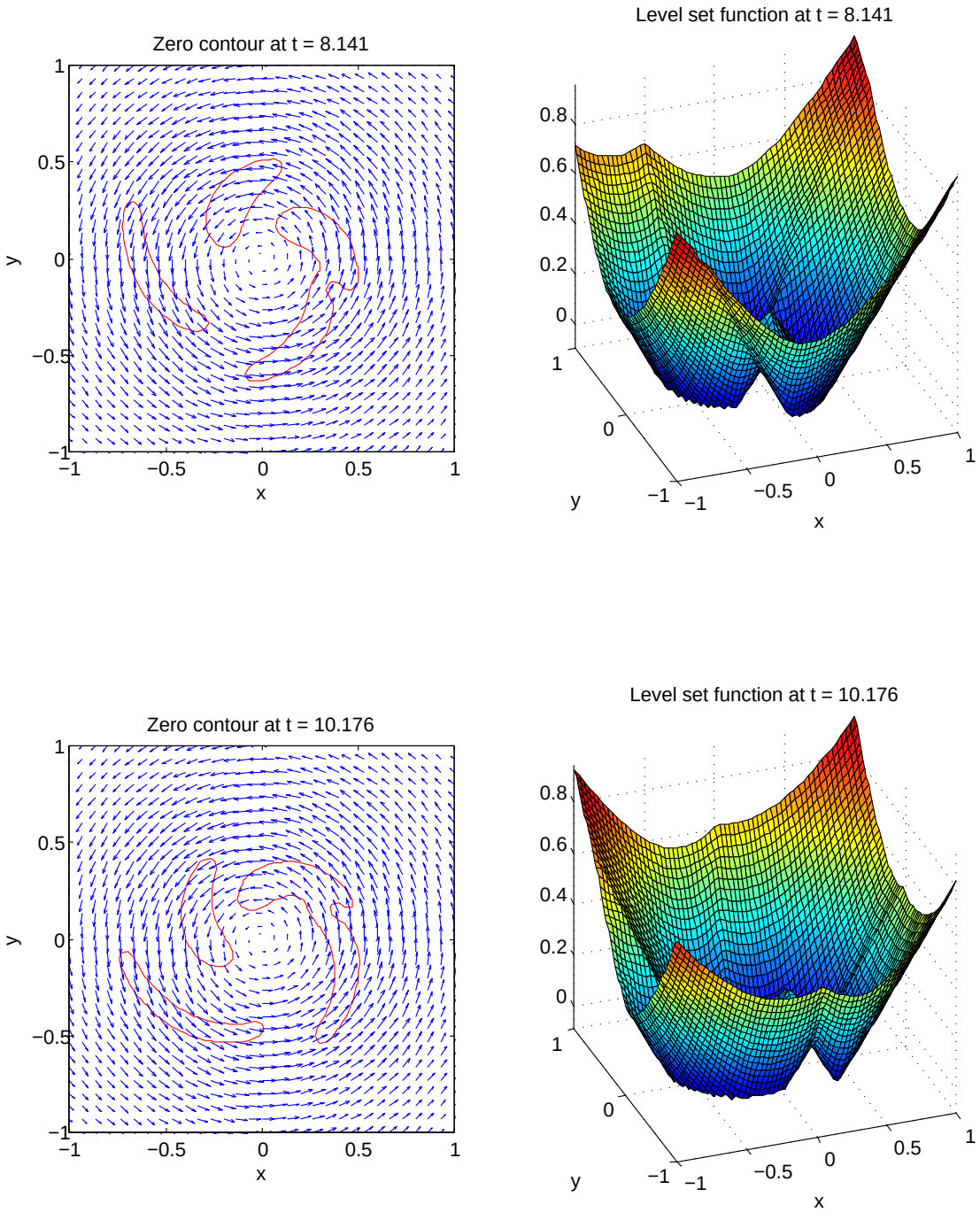

Figure 4.17: Zero contours and the level set function at t=8.141 (top) and t=10.176 (bottom) in Test 5.

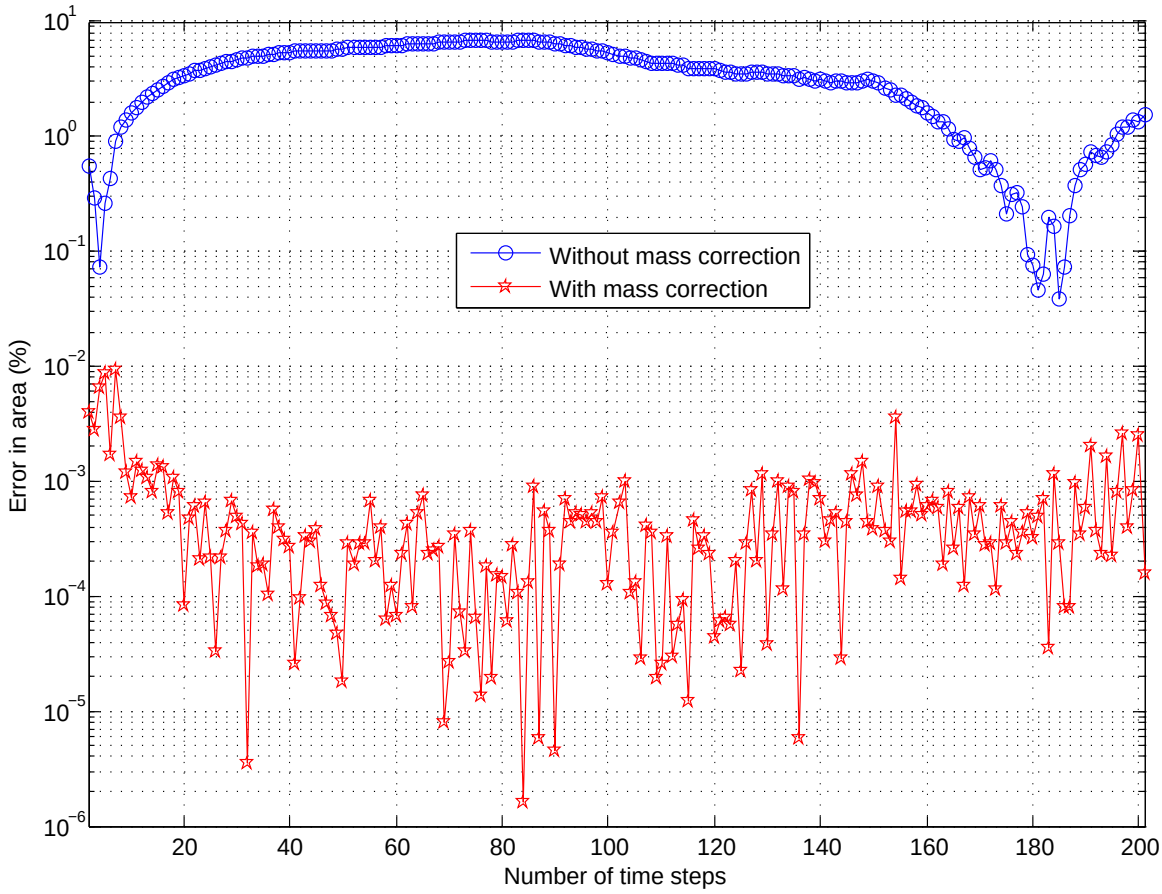

Figure 4.18: Comparison on the percentage error in area of the bubbles resulted from the present approach with and without mass correction in Test 5. The accuracy of the numerical solution is improved significantly for the latter case. The error in total area of the bubbles in motion bounded within $\mathcal{O}(10^{-5})$-$\mathcal{O}(10^{-3})$ indicates the good stability of the present approach with mass correction for this test problem.

4.7 Concluding Remarks

A new meshless approach to capturing moving interfaces in passive transport problems has been presented in this chapter where the motion and deformation of the moving interfaces are well captured by a unique procedure even with the presence of topology changes. The present approach brings the highly accurate IRBFN method for spatial discretization, the high-order time stepping methods based on semi-Lagrangian or Taylor series expansions and the level set method together for dealing with the moving interfaces in an accurate and efficient manner. In this work, a mass correction procedure is introduced at each time step to improve the accuracy of the interface reconstruction. The procedure can be used with or without reinitialization step. Numerical experiments, including some basic tests for the new numerical schemes and the simulation of one or more bubbles moving, stretching and merging in an ambient shear flow, show the good capability of the new approach for passive transport problems.

Chapter 5

Meshless Schemes for Unsteady Navier-Stokes Equations

This chapter reports the derivation and implementation of the two new meshless numerical schemes, namely IPC-IRBFN and IPCPP-IRBFN, for solving unsteady Navier-Stokes equations in which the two primary unknowns, velocity field and pressure, are coupled together via the momentum equation and the incompressibility constraint. In the proposed schemes, the projection method is used in the meshless framework of the IRBFN method to successively solve subproblems for the velocity field and pressure in a segregated manner. Numerical experiments on the new schemes in terms of accuracy and stability are performed for verification and demonstration purposes.

5.1 Introduction

Navier-Stokes equations are well known as the ones that govern incompressible viscous flows. There is a major difficulty for the numerical solution of these equations due to the imposition of the incompressibility constraint (Marion

and Temam, 1998). Particularly, the fact that the pressure does not appear in the continuity equation results in the saddle-point problem which is difficult for the numerical solution of the Navier-Stokes equations (Temam, 2001). Various numerical schemes based on the projection method have been developed to overcome this difficulty.

The underlying idea of the projection method is to treat the velocity and pressure in a segregated manner by solving at each time step a sequence of decoupled equations for each of these two quantities. In particular, the projection method consists of a predictor-corrector procedure between the velocity and pressure fields as follows. In the first substep, the momentum equation is solved using an initial approximation of the pressure to obtain an intermediate velocity field. This velocity in general does not satisfy the divergence-free constraint and must therefore be corrected. In the second substep, a pressure correction is obtained by solving a Poisson equation resulted from taking the divergence of the momentum equation and enforcing the incompressibility constraint. The new velocity field is then updated using the intermediate velocity and the pressure correction.

The projection method was first proposed in (Chorin, 1968) where the pressure is ignored in the first substep, and the pressure correction resulted from solving the Poisson equation is the new pressure itself. In this method, the gradient of pressure normal to the boundaries are forced to be zero on boundaries at all time. Such an inconsistent/artificial Neumann boundary condition gives rise to a numerical boundary layer that restricts the accuracy of the method to first order in time (Rannacher, 1992). Several improvements have been made in the past twenty years using one of the following approaches: (a) improving intermediate velocity boundary conditions (Kim and Moin, 1985); (b) improving accuracy order in time via pressure correction procedure (Goda, 1979; Van Kan, 1986; Bell et al., 1989; Gresho, 1990); (c) improving pressure boundary conditions (Orszag et al., 1986; Karniadakis et al., 1991).

By using old values of the pressure gradient in the first substep rather than completely ignoring them as in (Chorin, 1968), Goda (1979) reported numerical solutions with improved accuracy. This idea was then implemented by Van Kan (1986) using the semi-implicit Crank-Nicolson method as the temporal discretization and the finite difference method as the discretization in space. The accuracy of the numerical solutions obtained by such a method (known as the incremental pressure correction method) was reported to be of order two. Kim and Moin (1985) derived an appropriate boundary condition for the intermediate velocity in the first substep and obtained second-order accurate solutions in both space and time. In their work, the Crank-Nicolson method was used for the viscous part and an Adams-Bashforth method was used for the convective term as well as the forcing term. Timmermans et al. (1996) proposed a consistent Neumann boundary condition for the Poisson equation by taking into account the divergence of the intermediate velocity in the correction of the pressure term. This method is classified as the incremental pressure correction method in rotational form. Following an approach to improving the Neumann boundary conditions for pressure in the Poisson equation, Karniadakis et al. (1991) proposed the well-known class of stiffly stable schemes in which improved pressure boundary conditions of high order in time were introduced that minimize the effect of erroneous numerical boundary layer induced by the classical fractional-step/projection methods. The schemes exhibit higher accuracy in time with broader stability regions than the Adams-based schemes.

The remaining parts of this chapter are organized as follows. Section §5.2 presents the Navier-Stokes equations in velocity-pressure formulation. Two novel numerical schemes based on the projection method implemented within the meshless framework of the IRBFN method are then reported in Section §5.3. Numerical experiments on the proposed schemes for some test problems are carried out and analyzed in Section §5.4 for the purpose of verifying the new schemes in terms of accuracy and stability. This chapter ends with some concluding remarks in Section §5.5.

5.2 Mathematical Formulation

Consider a domain $\boldsymbol{\Omega} \subset \mathbb{R}^2$ with boundary $\boldsymbol{\partial\Omega}$. The Navier-Stokes equations that govern incompressible viscous flows are comprised of the momentum and continuity equations as presented in Chapter 2 and written in dimensionless form as follows.

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\,\mathbf{v} = -\nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{f}, \quad \text{in } \boldsymbol{\Omega}, \tag{5.1}$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \bar{\boldsymbol{\Omega}}, \tag{5.2}$$

where $\bar{\boldsymbol{\Omega}} = \boldsymbol{\Omega} \cup \partial\boldsymbol{\Omega}$, $\mathbf{v}(\mathbf{x}, t) = (u, v)^T$ is the velocity field, $p(\mathbf{x}, t)$ is the kinematic pressure, $f(\mathbf{x}, t)$ is the body force vector, and ν is the kinematic viscosity. The velocity field is subject to boundary and initial conditions as follows.

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t) \quad \text{on } \partial\boldsymbol{\Omega}, \tag{5.3}$$

$$\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0 \quad \text{at } t = 0, \tag{5.4}$$

where $\mathbf{v}_0(\mathbf{x})$ and $\mathbf{g}(\mathbf{x}, t)$ are given functions satisfying the following constraints for the Navier-Stokes equations to be well posed.

$$\nabla \cdot \mathbf{v}_0 = 0 \quad \text{in } \bar{\boldsymbol{\Omega}}, \tag{5.5}$$

$$\mathbf{n} \cdot \mathbf{v}_0 = \mathbf{n} \cdot \mathbf{g}(\mathbf{x}, 0) \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.6}$$

Since neither initial nor boundary conditions are prescribed for the pressure in the Navier-Stokes equations , p is determined up to an additive constant corresponding to the level of hydrostatic pressure. In addition, global mass conservation must be imposed through the boundary conditions, leading to the constraint (Marion and Temam, 1998)

$$\int_{\partial\boldsymbol{\Omega}} \boldsymbol{n} \cdot \boldsymbol{g} \mathrm{d}\boldsymbol{\Gamma} = 0. \tag{5.7}$$

5.3 Numerical Schemes

In this section, two new meshless projection schemes, namely IPC-IRBFN and IPCPP-IRBFN, are presented for solving the unsteady Navier-Stokes equations. The two schemes are of incremental pressure correction methods implemented in the meshless framework of the IRBFN method and coupled with the high-order multistep time integration BDF and AB methods (Section §2.2.2). Four different types of the projection methods are implemented in the two schemes, namely: (a) Standard IPC-IRBFN, a meshless incremental pressure correction scheme in the standard form inspired by (Van Kan, 1986); (b) Rotational IPC-IRBFN, a meshless incremental pressure correction scheme in the rotational form based on Timmermans et al. (1996); (c) Standard IPCPP-IRBFN, a meshless incremental pressure correction scheme in the standard form with pressure prediction suggested by Timmermans et al. (1996); and (d) Rotational IPCPP-IRBFN, a meshless incremental pressure correction scheme in the rotational form with pressure prediction suggested by Timmermans et al. (1996). It is noted that these numerical schemes are applicable to unsteady Stokes equations where the nonlinear convective term is not taken into account.

5.3.1 The IPC-IRBFN schemes

The derivation of the IPC-IRBFN schemes starts with considering the original projection method proposed by Chorin (1968) in which Equation (5.1) is first solved for the intermediate velocity field by using the backward Euler time stepping with the linearized convective term and without the pressure gradient

$$\frac{1}{\Delta t}\left(\tilde{\mathbf{v}}^{n+1}-\mathbf{v}^{n}\right)+\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n}=\nu\nabla^{2}\tilde{\mathbf{v}}^{n+1}+\boldsymbol{f}^{n+1}\quad\text{in }\boldsymbol{\Omega}, \tag{5.8}$$

$$\tilde{\mathbf{v}}^{n+1}=\boldsymbol{g}^{n+1}\quad\text{on }\partial\boldsymbol{\Omega}. \tag{5.9}$$

The new time-level (end-of-step) velocity $\mathbf{v}^{n+1}$ and pressure p^{n+1} are then obtained by solving

$$\frac{1}{\Delta t}\left(\mathbf{v}^{n+1} - \tilde{\mathbf{v}}^{n+1}\right) + \nabla p^{n+1} = 0 \quad \text{in } \boldsymbol{\Omega}, \tag{5.10}$$

$$\nabla \cdot \mathbf{v}^{n+1} = 0 \quad \text{in } \boldsymbol{\Omega}, \tag{5.11}$$

$$\mathbf{v}^{n+1} \cdot \mathbf{n} = \boldsymbol{g}^{n+1} \cdot \mathbf{n} \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.12}$$

Rather than simultaneously solving for the velocity and pressure, a Poisson pressure equation (PPE) is formulated from the above equations to solve for the new pressure separately. This is done by taking the divergence of Equation (5.10) and using the incompressibility constraint described in Equation (5.11)

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t}\nabla \cdot \tilde{\mathbf{v}}^{n+1} \quad \text{in } \boldsymbol{\Omega}, \tag{5.13}$$

The boundary condition for the PPE is obtained by taking the normal component of Equation (5.10) and taking into account the boundary conditions described in Equations (5.9,5.12)

$$\frac{\partial p^{n+1}}{\partial n} = 0 \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.14}$$

The end-of-step velocity field can be then updated using Equation (5.10)

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}}^{n+1} - \frac{1}{\Delta t}\nabla p^{n+1}. \tag{5.15}$$

In the incremental pressure correction methods, e.g. the one proposed by Van Kan (1986), the pressure gradient from the previous step is taken into account rather than ignored as in the original projection method. More specifically, the intermediate velocity field in this case can be found by solving the following equations

$$\frac{1}{\Delta t}\left(\tilde{\mathbf{v}}^{n+1} - \mathbf{v}^{n}\right) + \left[(\mathbf{v} \cdot \nabla)\,\mathbf{v}\right]^{n} = -\nabla p^{n} + \nu\nabla^2 \tilde{\mathbf{v}}^{n+1} + \boldsymbol{f}^{n+1} \quad \text{in } \boldsymbol{\Omega}, \tag{5.16}$$

$$\tilde{\mathbf{v}}^{n+1} = \boldsymbol{g}^{n+1} \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.17}$$

The end-of-step velocity and pressure can be then obtained by solving the following equations

$$\frac{1}{\Delta t}\left(\mathbf{v}^{n+1} - \tilde{\mathbf{v}}^{n}\right) + \nabla\left(p^{n+1} - p^{n}\right) = 0 \quad \text{in } \boldsymbol{\Omega}, \tag{5.18}$$

$$\nabla \cdot \mathbf{v}^{n+1} = 0 \quad \text{in } \boldsymbol{\Omega}, \tag{5.19}$$

$$\mathbf{v}^{n+1} \cdot \mathbf{n} = \boldsymbol{g}^{n+1} \cdot \mathbf{n} \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.20}$$

Let $q^{n+1} = p^{n+1} - p^{n}$ be the pressure increment. By taking the divergence of Equation (5.18), using Equation (5.19) and taking into account the boundary conditions in Equations (5.17,5.20), one has the Poisson equation for the the pressure increment q^{n+1} along with the boundary condition as follows

$$\nabla^2 q^{n+1} = \frac{1}{\Delta t}\tilde{\mathbf{v}}^{n+1} \quad \text{in } \boldsymbol{\Omega}, \tag{5.21}$$

$$\frac{\partial q^{n+1}}{\partial n} = 0 \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.22}$$

The end-of-step velocity and pressure are then given by

$$p^{n+1} = q^{n+1} + p^{n}, \tag{5.23}$$

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}}^{n+1} - \Delta t \nabla q^{n+1}. \tag{5.24}$$

The Standard IPC-IRBFN Scheme

On the basis of the incremental pressure correction method previously presented, the Standard IPC-IRBFN scheme can be now formulated with the following modifications motivated from (Karniadakis et al., 1991):

1. High-order BDF integration method is used for time stepping rather than

the first-order backward Euler. In particular, the temporal derivative is discretized in time as follows.

$$\left.\frac{\partial \mathbf{v}}{\partial t}\right|_{t=t^{n+1}} \approx \frac{1}{\Delta t}\left(\beta_0 \mathbf{v}^{n+1} - \sum_{k=1}^{J_v} \beta_k \mathbf{v}^{n+1-k}\right) \tag{5.25}$$

The values of coefficients $\beta' s$ corresponding to J_v are given in the next section.

2. High-order Adam-Bashforth extrapolation method is used to linearized the convective term. For this method, the convective term at a time level is calculated from multiple previous steps instead of just relying on the last value.

$$\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]|_{t=t^{n+1}} \approx \sum_{k=0}^{J_v-1} \alpha_k \left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-k} \tag{5.26}$$

The values of coefficients $\alpha' s$ corresponding to J_v are given in the next section.

3. Instead of just taking into account the value of the pressure from the last time step in solving for the intermediate velocity field, the Standard IPC-IRBFN scheme uses a pressure predictor which is extrapolated from multiple previous steps as follows.

$$\bar{p}|_{t=t^{n+1}} = \sum_{k=0}^{J_v-1} \alpha_k p^{n-k} \tag{5.27}$$

By taking the above modifications, the Standard IPC-IRBFN scheme consists of the following steps for each time level $t^{n+1} = (n+1)\Delta t, n = 0, 1, 2, \ldots$.

1. Calculate a predictor for the pressure, $\bar{p}^{n+1}$

$$\bar{p}^{n+1} = \sum_{k=0}^{J_p} \alpha_k p^{n-k} = \begin{cases} 0, & J_p = 0 \\ p^n, & J_p = 1 \\ 2p^n - p^{n-1}, & J_p = 2 \end{cases} \tag{5.28}$$

2. Compute a predictor for the velocity field, $\tilde{\mathbf{v}}^{n+1}$, by solving

$$\frac{1}{\Delta t}\left(\beta_0\tilde{\mathbf{v}}^{n+1}-\sum_{k=1}^{J_v}\beta_k\mathbf{v}^{n+1-k}\right)+\sum_{k=0}^{J_v-1}\alpha_k\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-k}=-\nabla\bar{p}^{n+1}+\nu\nabla^2\tilde{\mathbf{v}}^{n+1}+\boldsymbol{f}^{n+1}\quad\text{in }\boldsymbol{\Omega},\tag{5.29}$$

with the Dirichlet boundary condition

$$\tilde{\mathbf{v}}^{n+1}=\boldsymbol{g}^{n+1}\quad\text{on }\partial\boldsymbol{\Omega}.\tag{5.30}$$

where

$$\sum_{k=1}^{J_v}\beta_k\mathbf{v}^{n+1-k}=\begin{cases}\mathbf{v}^n, & J_v=1\ (\beta_0=1)\\ 2\mathbf{v}^n-\frac{1}{2}\mathbf{v}^{n-1}, & J_v=2\ (\beta_0=\frac{3}{2})\\ 3\mathbf{v}^n-\frac{3}{2}\mathbf{v}^{n-1}+\frac{1}{3}\mathbf{v}^{n-2}, & J_v=3\ (\beta_0=\frac{11}{6})\end{cases}\tag{5.31}$$

and

$$\sum_{k=0}^{J_v-1}\alpha_k\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-k}=\begin{cases}\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n}, & J_v=1,\\ 2\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n}-\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-1}, & J_v=2,\\ 3\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n}-3\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-1}+\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-2}, & J_v=3.\end{cases}\tag{5.32}$$

3. Calculate the pressure increment q^{n+1}

$$\nabla^2 q^{n+1}=\frac{\beta_0}{\Delta t}\nabla\cdot\tilde{\mathbf{v}}^{n+1}\quad\text{in }\boldsymbol{\Omega},\tag{5.33}$$

$$\frac{\partial q^{n+1}}{\partial n}=0\quad\text{on }\partial\boldsymbol{\Omega}.\tag{5.34}$$

4. Perform the correction step for pressure p^{n+1}

$$p^{n+1}=q^{n+1}+\bar{p}^{n+1}\tag{5.35}$$

5. Perform the correction step for velocity field $\mathbf{v}^{n+1}$

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}}^{n+1} - \frac{\Delta t}{\beta_0}\nabla q^{n+1}. \tag{5.36}$$

The Rotational IPC-IRBFN Scheme

In this scheme, a consistency requirement is explicitly posed on the numerical solutions stating that the end-of-step velocity and pressure, $\mathbf{v}^{n+1}$ and p^{n+1}, must numerically satisfy the momentum and continuity equations regardless of how the velocity predictor, $\tilde{\mathbf{v}}^{n+1}$, is calculated. More specifically, the momentum equation (5.1) and the continuity equation (5.2) must hold for $\mathbf{v}^{n+1}$ and p^{n+1} in the semi-discrete form in time as follows.

$$\frac{1}{\Delta t}\left(\beta_0\mathbf{v}^{n+1} - \sum_{k=1}^{J_v}\beta_k\mathbf{v}^{n+1-k}\right) + \sum_{k=0}^{J_v-1}\alpha_k\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-k} = -\nabla p^{n+1} + \nu\nabla^2\mathbf{v}^{n+1} + \boldsymbol{f}^{n+1} \quad \text{in } \boldsymbol{\Omega}, \tag{5.37}$$

$$\nabla\cdot\mathbf{v}^{n+1} = 0 \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.38}$$

The above equations are now used to derive the corresponding steps for the new Rotational IPC-IRBFN scheme as follows. First, subtracting Equation (5.37) from Equation (5.29) yields

$$\frac{\beta_0}{\Delta t}\left(\mathbf{v}^{n+1} - \tilde{\mathbf{v}}^{n+1}\right) = -\nabla(p^{n+1} - \bar{p}^{n+1}) + \nu\nabla^2(\mathbf{v}^{n+1} - \tilde{\mathbf{v}}^{n+1}), \tag{5.39}$$

By taking the divergence of Equation (5.39), one has

$$\frac{\beta_0}{\Delta t}\nabla\cdot\left(\mathbf{v}^{n+1} - \tilde{\mathbf{v}}^{n+1}\right) = -\nabla\cdot\left[\nabla(p^{n+1} - \bar{p}^{n+1})\right] + \nu\nabla\cdot[\nabla^2(\mathbf{v}^{n+1} - \tilde{\mathbf{v}}^{n+1})] \tag{5.40}$$

Using the identities (Gresho and Sani, 2000)

$$\nabla^2\mathbf{v} = \nabla\left(\nabla\cdot\mathbf{v}\right) - \nabla\times\left(\nabla\times\mathbf{v}\right), \tag{5.41}$$

$$\nabla \cdot (\nabla s) = \nabla^2 s \quad \forall \text{ scalar s} \quad \text{and} \quad \nabla \cdot (\nabla \times \mathbf{r}) = 0 \quad \forall \text{ vector } \mathbf{r}, \tag{5.42}$$

one can show that the divergence and Laplace operators commute

$$\nabla \cdot \left(\nabla^2 \mathbf{v}\right) = \nabla \cdot \left[\nabla \left(\nabla \cdot \mathbf{v}\right)\right] - \nabla \cdot \left[\nabla \times \left(\nabla \times \mathbf{v}\right)\right] = \nabla^2 \left(\nabla \cdot \mathbf{v}\right). \tag{5.43}$$

Using Equation (5.43) and imposing the incompressibility constraint on the end-of-step velocity, one can simplify Equation (5.40) to

$$-\frac{\beta_0}{\Delta t} \nabla \cdot \tilde{\mathbf{v}}^{n+1} = -\nabla^2 (p^{n+1} - \bar{p}^{n+1}) - \nabla^2 \left(\nu \nabla \cdot \tilde{\mathbf{v}}^{n+1}\right). \tag{5.44}$$

Rearranging terms in the above equation yields

$$\nabla^2 \left(p^{n+1} - \bar{p}^{n+1} + \nu \nabla \cdot \tilde{\mathbf{v}}^{n+1}\right) = \frac{\beta_0}{\Delta t} \nabla \cdot \tilde{\mathbf{v}}^{n+1} \tag{5.45}$$

By letting $q^{n+1} = p^{n+1} - \bar{p}^{n+1} + \nu \nabla \cdot \tilde{\mathbf{v}}^{n+1}$ be the pressure increment in this case, one obtains the Poisson equation for q^{n+1} as follows

$$\nabla^2 q^{n+1} = \frac{\beta_0}{\Delta t} \nabla \cdot \tilde{\mathbf{v}}^{n+1} \tag{5.46}$$

The boundary condition for the above equation that is consistent with global mass conservation can be obtained by integrating Equation (5.46) over $\mathbf{\Omega}$

$$\int_{\mathbf{\Omega}} \nabla^2 q^{n+1} \mathrm{d}\mathbf{\Omega} = \frac{\beta_0}{\Delta t} \int_{\mathbf{\Omega}} \nabla \cdot \tilde{\mathbf{v}}^{n+1} \mathrm{d}\mathbf{\Omega} \tag{5.47}$$

or equivalently

$$\int_{\Gamma} \frac{\partial q^{n+1}}{\partial n} \mathrm{d}\Gamma = \frac{\beta_0}{\Delta t} \int_{\Gamma} \tilde{\mathbf{v}}^{n+1} \cdot \mathbf{n} \, \mathrm{d}\Gamma = \frac{\beta_0}{\Delta t} \int_{\Gamma} \boldsymbol{g}^{n+1} \cdot \mathbf{n} \, \mathrm{d}\Gamma = 0. \tag{5.48}$$

where $\Gamma = \partial\mathbf{\Omega}$ is the boundary of the domain $\mathbf{\Omega}$. It is noted that the global mass conservation constraint (5.7) is used to derive the last equality in the above equation. Based on Equation (5.48), Timmermans et al. (1996) suggested a

homogeneous Neumann boundary condition for q^{n+1} as follows.

$$\frac{\partial q^{n+1}}{\partial n} = 0 \quad \text{on } \partial\mathbf{\Omega}. \tag{5.49}$$

The solving procedure in the Rotational IPC-IRBFN scheme is now summarized for each time level $t^{n+1} = (n+1)\Delta t, n = 0, 1, 2, \ldots$ as follows.

1. Calculate a predictor for the pressure, $\bar{p}^{n+1}$, using Equation (5.28);
2. Compute a predictor for the velocity field, $\tilde{\mathbf{v}}^{n+1}$, by solving Equations (5.29,5.30);
3. Calculate the pressure increment, q^{n+1}, by solving Equations (5.46,5.49) as in the Standard IPC-IRBFN scheme;
4. Perform the correction step for the new pressure p^{n+1}

$$p^{n+1} = q^{n+1} + \bar{p}^{n+1} - \nu\nabla \cdot \tilde{\mathbf{v}}^{n+1}. \tag{5.50}$$

5. Perform the correction step for velocity field, $\mathbf{v}^{n+1}$, using Equation (5.36) as in the Standard IPC-IRBFN scheme.

As can be seen from the solving procedure of the IPC-IRBFN schemes in both standard and rotational forms, the two forms of the IPC-IRBFN schemes differ in the manner that the pressure increment, q^{n+1}, is defined, and thus in the pressure correction step.

5.3.2 The IPCPP-IRBFN Schemes

Instead of extrapolating the pressure at the beginning of each time step as in the IPC-IRBFN schemes, the IPCPP-IRBFN schemes solve a Poisson equation with Neumann boundary condition (Gresho and Sani, 2000) for the pressure

predictor at each time step. By taking divergence of Equation (5.1) and making use of Equation (5.2), the Poisson equation for pressure is derived as

$$\nabla^2 p = -\nabla \cdot [(\mathbf{v} \cdot \nabla)\,\mathbf{v} - \boldsymbol{f}] \quad \text{in } \boldsymbol{\Omega}, \tag{5.51}$$

with the Neumann boundary condition being derived by taking the normal component of the momentum equation (5.1) as

$$\frac{\partial p}{\partial n} = \boldsymbol{n} \cdot \left[-\frac{\partial \mathbf{v}}{\partial t} - (\mathbf{v} \cdot \nabla)\,\mathbf{v} + \nu \nabla^2 \mathbf{v} + \boldsymbol{f} \right] \quad \text{on } \partial\boldsymbol{\Omega}. \tag{5.52}$$

Therefore, the pressure predictor in the IPCPP-IRBFN schemes is calculated by solving the above equations in which the implicit BDF method is used to discretize the temporal derivative and the forcing term with respect to time whereas the explicit AB method is used for the nonlinear convective term and the viscous term in Equations (5.51-5.52) as follows.

$$\nabla^2 \bar{p}^{n+1} = \nabla \cdot \left\{ \sum_{k=0}^{J_v - 1} \alpha_k \left[(\mathbf{v} \cdot \nabla)\,\mathbf{v} \right]^{n-k} + \boldsymbol{f}^{n+1} \right\} \quad \text{in } \boldsymbol{\Omega}, \tag{5.53}$$

$$\begin{aligned} \frac{\partial \bar{p}^{n+1}}{\partial n} = \boldsymbol{n} \cdot \Bigg\{ & -\frac{\beta_0 \mathbf{v}^{n+1} + \sum_{k=1}^{J_v - 1} \beta_k \mathbf{v}^{n+1-k}}{\Delta t} - \sum_{k=0}^{J_v - 1} \alpha_k \left[(\mathbf{v} \cdot \nabla)\,\mathbf{v} \right]^{n-k} \\ & + \nu \sum_{k=0}^{J_v - 1} \alpha_k \left[-\nabla \times (\nabla \times \mathbf{v}) \right]^{n-k} + \boldsymbol{f}^{n+1} \Bigg\} \quad \text{on } \partial\boldsymbol{\Omega}. \end{aligned} \tag{5.54}$$

where Dirichlet boundary condition on velocity, $\mathbf{v}^{n+1} = \boldsymbol{g}^{n+1}$, is applied to $\mathbf{v}^{n+1}$ in Equation (5.54). It is noted that in the Neumann boundary condition for the pressure prediction, the viscous term is decomposed into

$$\nabla^2 \mathbf{v} = \nabla (\nabla \cdot \mathbf{v}) - \nabla \times (\nabla \times \mathbf{v}) \tag{5.55}$$

and the incompressibility constraint is used accordingly (Gresho and Sani, 2000). The advantage of expressing the viscous term as in Equation (5.55) is

the improvement of the stability of the numerical scheme. Indeed, the boundary condition (5.54) requires an explicit treatment of the Laplace operator on the velocity. In general, such a treatment leads to instabilities. To improve the stability of the numerical scheme, Karniadakis et al. (1991) proposed to decompose the Laplace operator into 2 parts as presented in Equation (5.55). For incompressible fluid flows, we can treat the divergence part implicitly (in fact, this is equivalent to its removal by virtue of the incompressibility constraint, $\nabla \cdot \mathbf{v}^{n+1} = 0$) and treat the second part explicitly as normal.

Like the IPC-IRBFN schemes, the IPCPP-IRBFN schemes have implemented in both standard and rotational forms. The solving procedure in the IPCPP-IRBFN schemes is summarized for each time level $t^{n+1} = (n+1)\Delta t, n = 0, 1, 2, \ldots$ as follows.

- Step 1: Calculate the predictor for the pressure by solving Equations (5.53,5.54);
- Step 2 to Step 5: The same as in the IPC-IRBFN schemes

5.4 Numerical Results

This section presents the numerical results obtained by applying the IPC-IRBFN and IPCPP-IRBFN schemes to some unsteady problems in CFD. The first test problem is for verifying the accuracy and stability of the new meshless schemes in solving an unsteady Stokes problem that has an analytical solution. In this test, the nonlinear convective term does not exist. The second test problem is an unsteady Navier-Stokes problem with an analytical solution. The new schemes are used to solve the problem where the nonlinear convective term is discretized in time by the second-order Adams-Bashforth method. The two schemes are then demonstrated with the well-known lid-driven cavity flow

focusing on the unsteady behavior of the flow. In the test problems with analytical solutions, the norm error of function $\varphi(\boldsymbol{x}_j, t), \varphi = (u, v, p)$ is used to verify the accuracy of the numerical schemes at each time step and defined as follows.

$$RMSE(\varphi) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}[\varphi(\boldsymbol{x}_j, t) - \varphi_a(\boldsymbol{x}_j, t)]^2} \tag{5.56}$$

where N is the number of collocation points, and φ_a is the corresponding analytical solution. The stability analysis on the new schemes is carried out by checking the boundedness of the norm error over the time interval of interest.

5.4.1 Test 1: Unsteady Stokes equations with known analytical solution

Consider the Stokes equations governing an unsteady flow in a square domain $\boldsymbol{\Omega} = [-1, 1]^2$ with Dirichlet boundary conditions on the velocity. The analytical solution is given as follows

$$u(x, y, t) = \pi \sin(2\pi y) \sin^2(\pi x) \sin(t) \tag{5.57}$$

$$v(x, y, t) = -\pi \sin(2\pi x) \sin^2(\pi y) \sin(t) \tag{5.58}$$

$$p(x, y, t) = \cos(\pi x) \sin(\pi y) \sin(t) \tag{5.59}$$

The forcing term is obtained by the symbolic computing package Mathematica based on Equation (5.1) without the nonlinear convective term as follows.

$$\begin{aligned} fx = {} & \pi \cos(t) \sin^2(\pi x) \sin(2\pi y) - \pi \sin(t) \sin(\pi x) \sin(\pi y) \\ & - 2\pi^3 \sin(t) \cos^2(\pi x) \sin(2\pi y) + 6\pi^3 \sin(t) \sin^2(\pi x) \sin(2\pi y) \end{aligned} \tag{5.60}$$

$$fy = \pi \cos(\pi x)\cos(\pi y)\sin(t) + 2\pi^3 \sin(2\pi x)\cos^2(\pi y)\sin(t) - \pi\left(\cos(t) + 6\pi^2 \sin(t)\right)\sin(2\pi x)\sin^2(\pi y) \quad (5.61)$$

The test problem is solved by the IPC-IRBFN and IPCPP-IRBFN schemes with a point density of 31 along each direction. The multistep BDF and AB methods used in this test are of order two. Comparisons between the numerical and analytical solutions are shown in Figure 5.1 for the velocity field and in Figure 5.2 for the pressure field. The figures show that the numerical solutions are in very good agreement with the analytical ones.

Stability analysis of the new schemes on the velocity field is shown in Figures (5.3-5.4) where all rotational schemes have the norm errors bounded within $\mathcal{O}(10^{-4})$ over the computational time domain with the time-step size $\Delta t = (0.005, 0.01)$. As can be seen from the figures, the IPC-IRBFN scheme in rotational form is superior to its counterpart in standard form with respect to both accuracy and stability of the numerical solution. Although yielding more accurate solution of the velocity field, the IPCPP-IRBFN scheme is not as stable as the IPC-IRBFN scheme. In addition, it is observed that, the stability of the IPCPP-IRBFN scheme in both standard and rotational forms are almost identical, meaning that the divergence of the intermediate velocity does not have remarkable affects on the error of velocity field. This, however, is not observed in the stability analysis with respect to pressure as shown in Figures (5.5-5.6) where the difference in norm error of the pressure between the standard and rotational forms is noticeable.

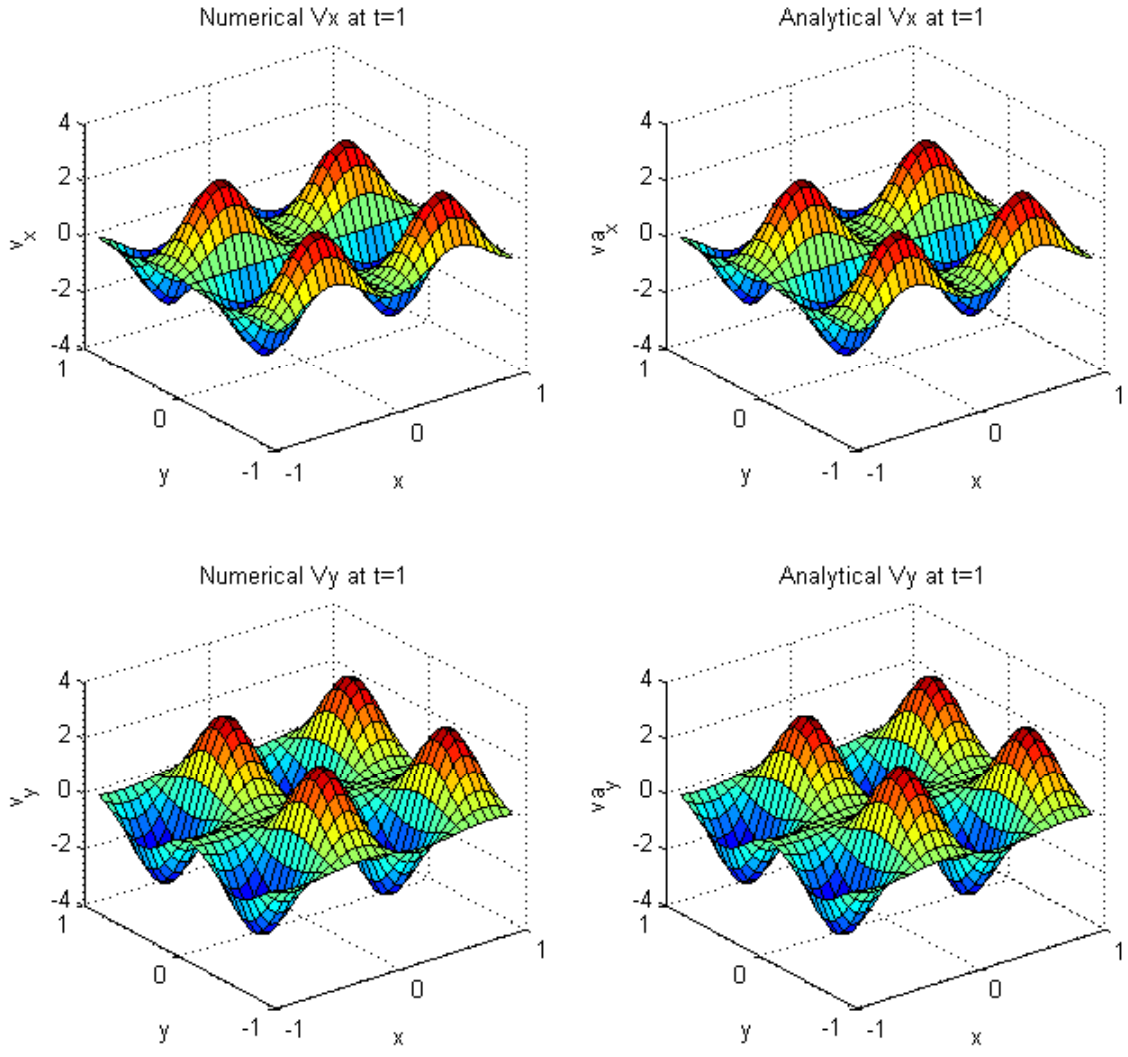

Figure 5.1: The numerical and analytical solutions of the velocity field with $\Delta t = 0.005$ in Test 1. Both IPC-IRBFN and IPCPP-IRBFN schemes yield accurate results compared to the analytical solution.

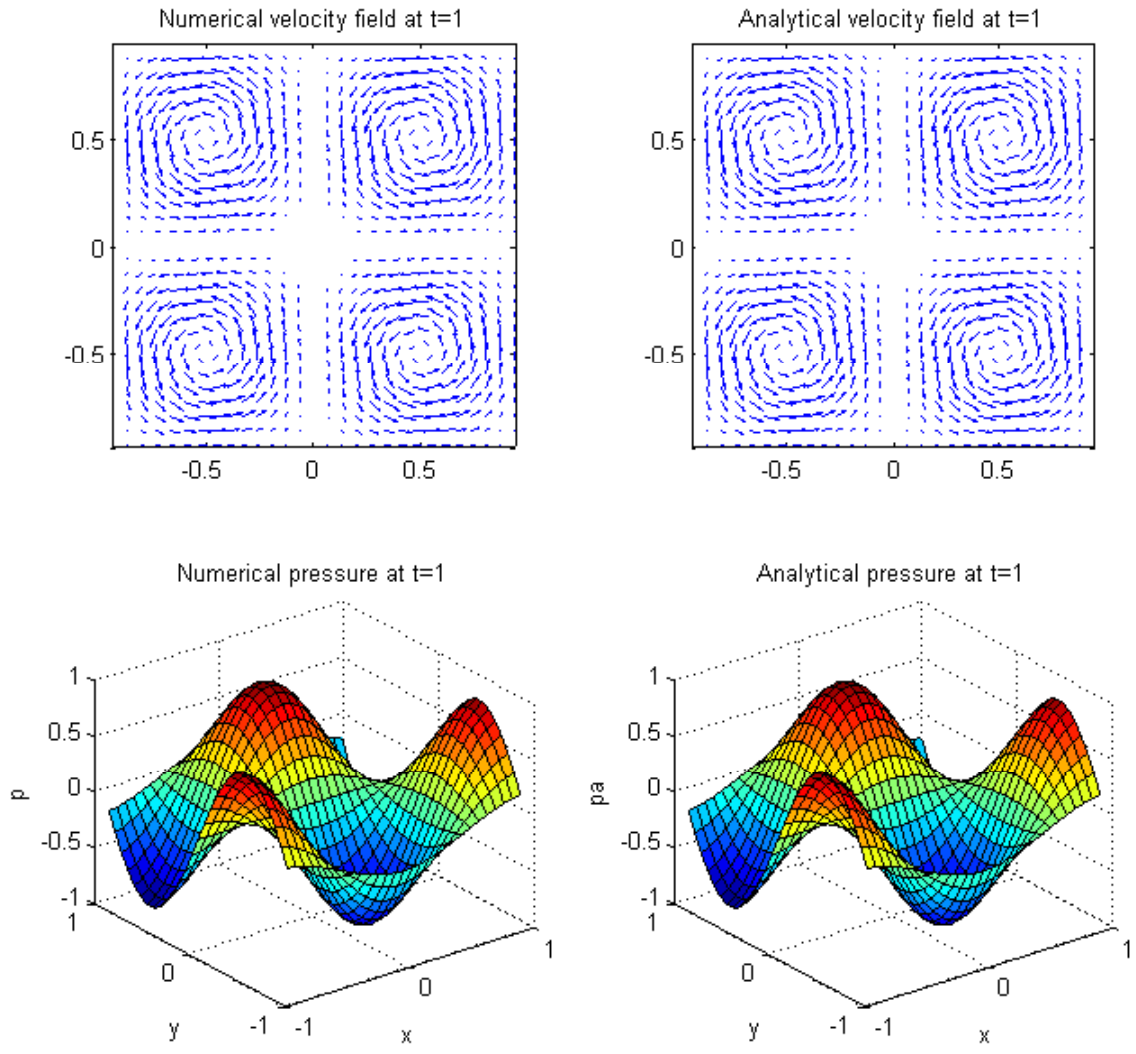

Figure 5.2: Analytical and numerical solutions of the pressure field with $\Delta t = 0.005$ in Test 1. Both IPC-IRBFN and IPCPP-IRBFN schemes yield accurate results compared to the analytical solution.

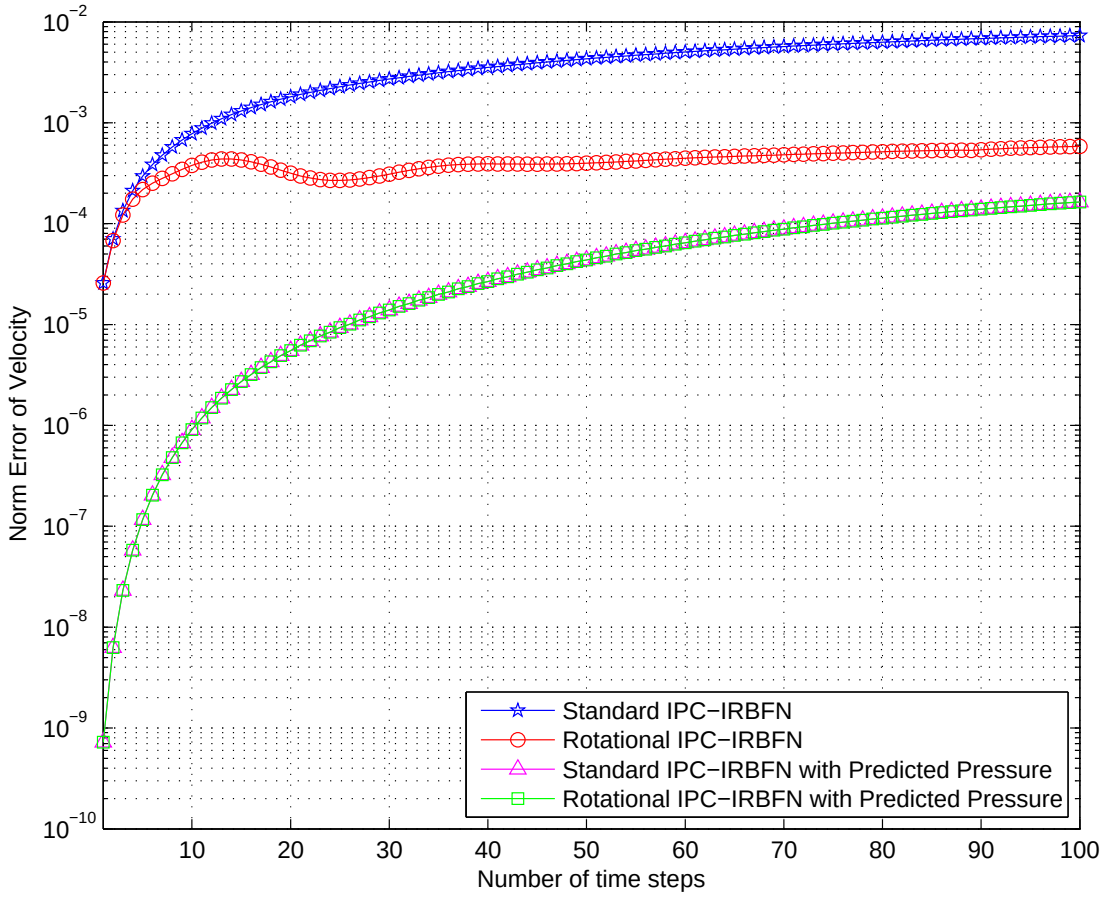

Figure 5.3: Stability analysis for the velocity field with $\Delta t = 0.01$ in Test 1. All rotational schemes have the norm errors bounded within $\mathcal{O}(10^{-4})$ in the time interval of interest with the time-step size $\Delta t = 0.01$.

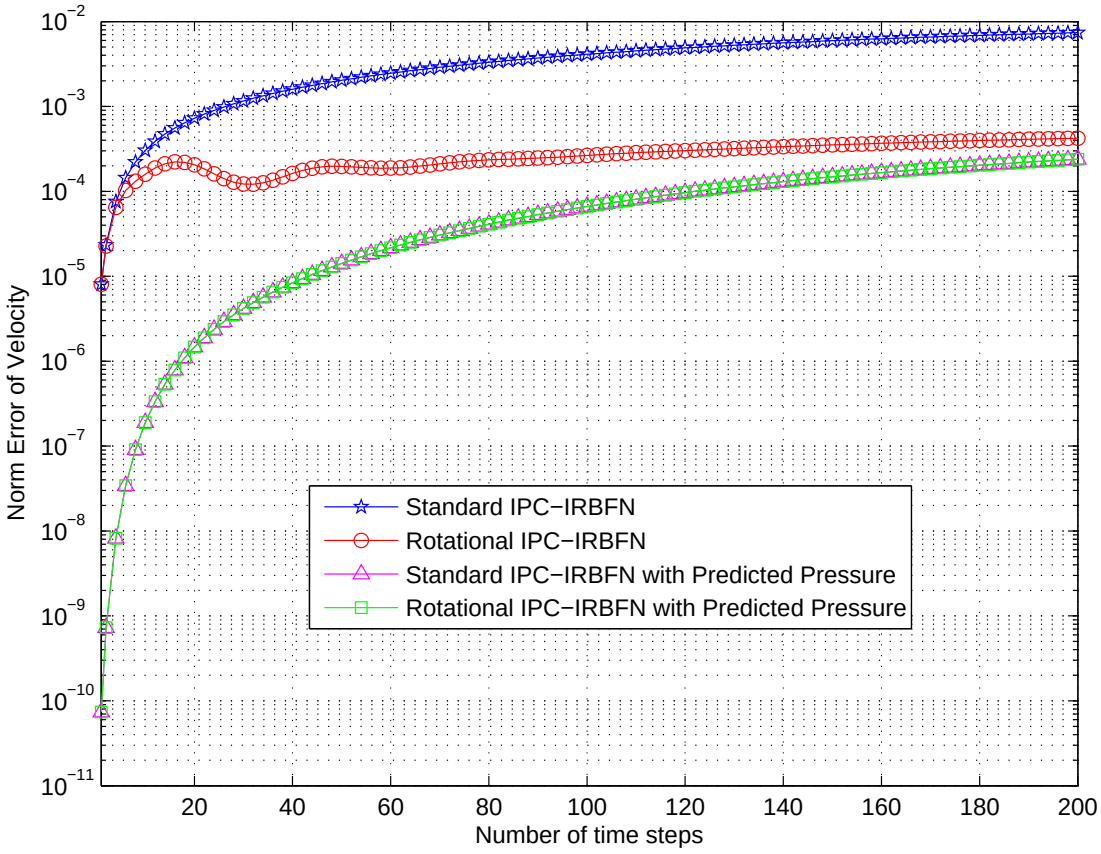

Figure 5.4: Stability analysis for the velocity field with $\Delta t = 0.005$ in Test 1. All rotational schemes have the norm errors bounded within $\mathcal{O}(10^{-4})$ in the time interval of interest with the time-step size $\Delta t = 0.005$.

Stability analysis of the new meshless schemes on the pressure is shown in Figures (5.5-5.6). For this test problem, the IPC-IRBFN scheme in rotational form shows its good accuracy where the norm error is bounded within $\mathcal{O}(10^{-3})$ in the time interval of interest with the time-step size Δt in the range of $(0.005-0.01)$. As can be seen from the figures, the rotational IPC-IRBFN scheme also shows its superiority over the other schemes with respect to the pressure stability.

5.4.2 Test 2: Unsteady Navier-Stokes equations with known analytical solution

Consider the Navier-Stokes equations governing an unsteady flow in a square domain $\mathbf{\Omega} = [0, \pi]^2$ with the analytical solution as follows (Kim and Moin, 1985).

$$u(x, y, t) = -\cos(x)\sin(y)\exp(-2t), \tag{5.62}$$

$$v(x, y, t) = \sin(x)\cos(y)\exp(-2t), \tag{5.63}$$

$$p(x, y, t) = -\frac{1}{4}\left(\cos(2x) + \cos(2y)\right)\exp(-4t). \tag{5.64}$$

The initial and boundary conditions as well as the forcing term are defined according to the analytical solution.

For this test problem, a point density of 31 grid points in each direction is used. Since the purpose of the analysis focuses on the temporal errors of the present schemes, the point density is chosen so that the error contributed from the spatial discretization does not affect on the ultimate error of the numerical schemes.

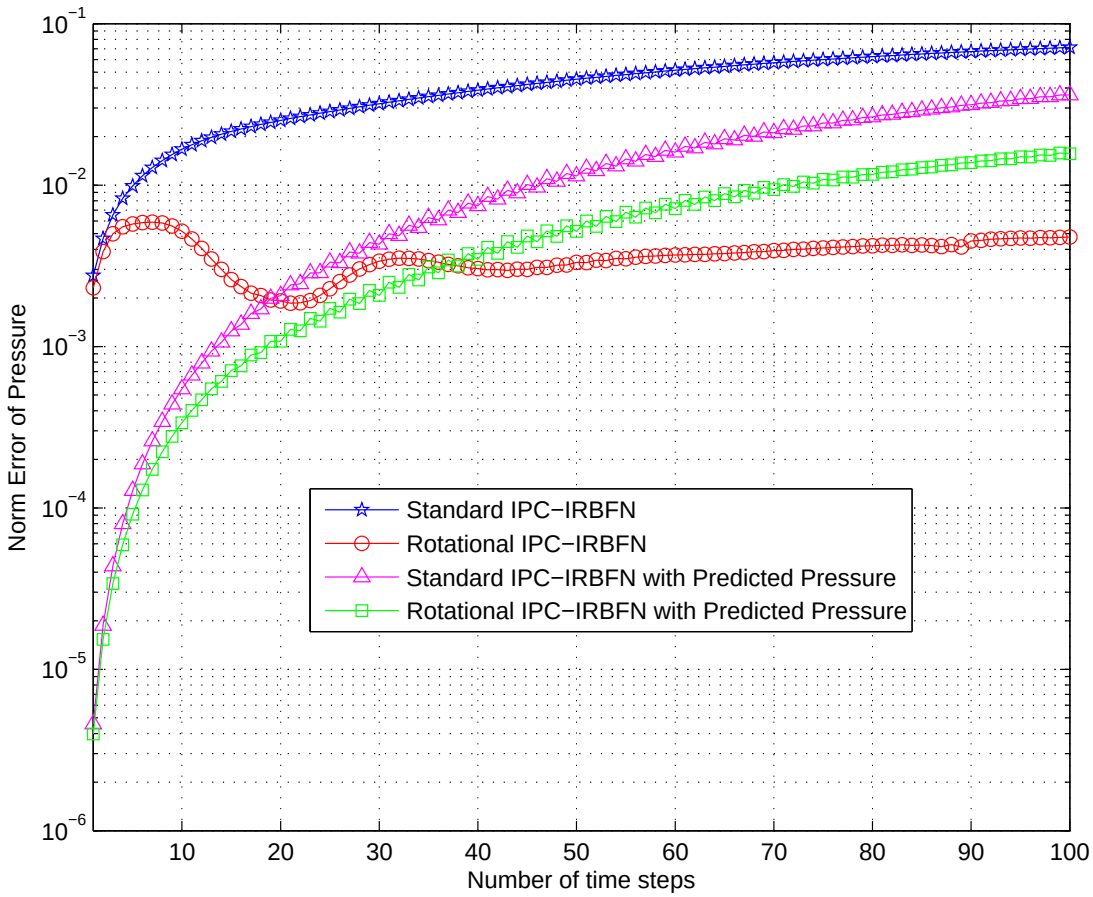

Figure 5.5: Stability analysis for the pressure field with $\Delta t = 0.01$ in Test 1. The IPC-IRBFN scheme in rotational form shows its superior to the other schemes with respect to the pressure stability as well as accuracy for this test problem.

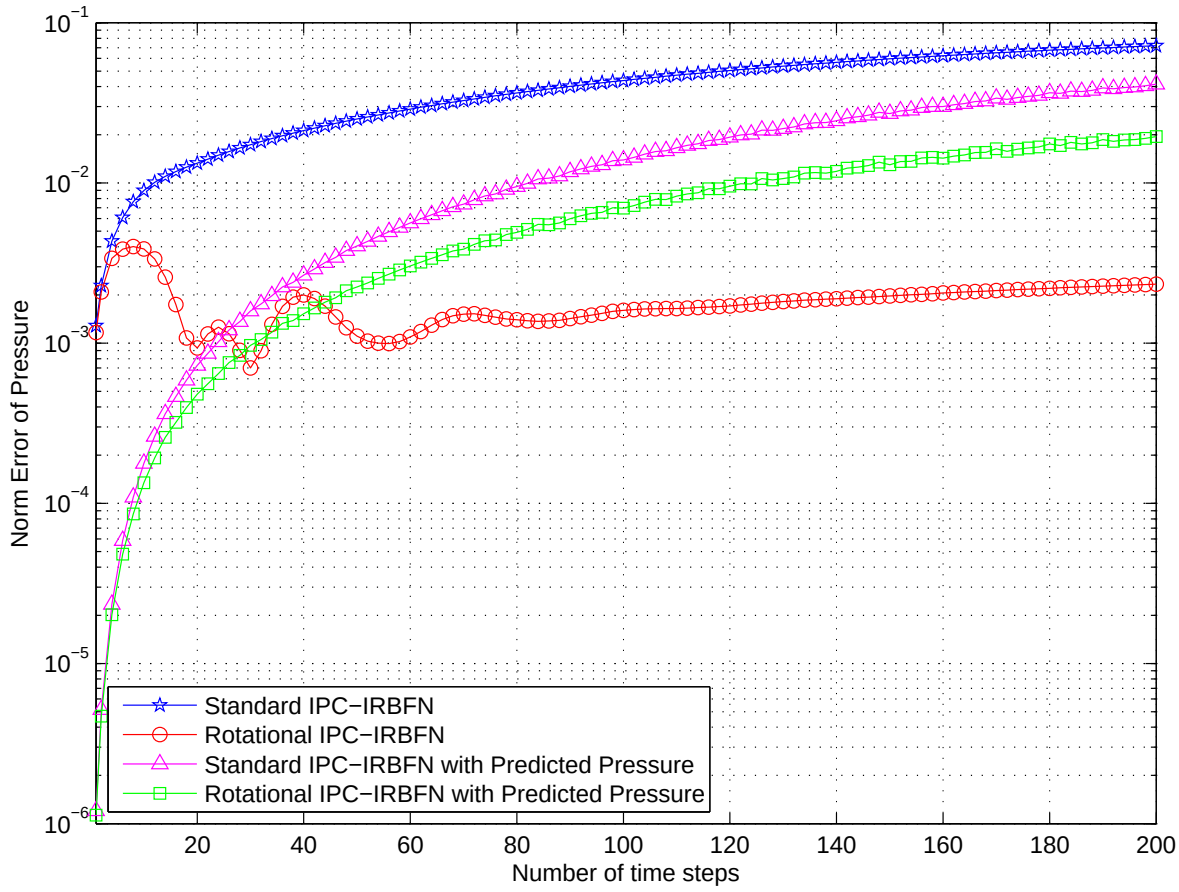

Figure 5.6: Stability analysis for the pressure field with $\Delta t = 0.005$ in Test 1. The IPC-IRBFN scheme in rotational form shows its superior to the other schemes with respect to the pressure stability as well as accuracy for this test problem.

In this second test problem, the multistep BDF and AB methods of order two are used from which the analysis on the accuracy and stability of the IPC-IRBFN and IPCPP-IRBFN schemes are accomplished. Figures (5.7-5.8) show the analytical and numerical solutions of Test 2. As can be seen from in Figure 5.7, the new schemes yield highly accurate solution. In fact, the norm error of the velocity is of $\mathcal{O}(10^{-5})$ for $\Delta t = 0.005$, and $\mathcal{O}(10^{-3})$ for $\Delta t = 0.01$. It is noted that the time-dependent boundary conditions of the velocity are well captured by the present schemes. For the pressure, errors on the boundaries associated with the numerical boundary layer (Karniadakis et al., 1991) is noticeable in Figure 5.8. The problem, however, is not severe (norm error of $\mathcal{O}(10^{-2})$ for this test with the IPC-IRBFN scheme in rotational form) thanks to the incremental pressure correction and the consistent boundary condition for pressure implemented in the two schemes.

The stability analysis of the two schemes in this test problem is shown in Figures (5.9-5.12). As can be seen from the figures, both IPC-IRBFN and IPCPP-IRBFN exhibit their good stability over the computational time domain. In particular, Figures (5.9-5.10) show that the numerical solutions of the velocity and pressure are highly stable over time just with a mild value of the time-step size ($\Delta t = 10^{-2}$). For this test problem which is the Navier-Stokes equations with time-dependent boundary conditions, the IPCPP-IRBFN schemes in both standard and rotational forms exhibit their good stability and accuracy. In fact, for $\Delta t = 0.005$, the norm error of the velocity is bounded within $\mathcal{O}(10^{-4})$ as shown in Figure 5.11, and the pressure error, with rather high value in a short interval of time, is quickly bounded within $\mathcal{O}(10^{-3})$ as can be seen in Figure 5.12.

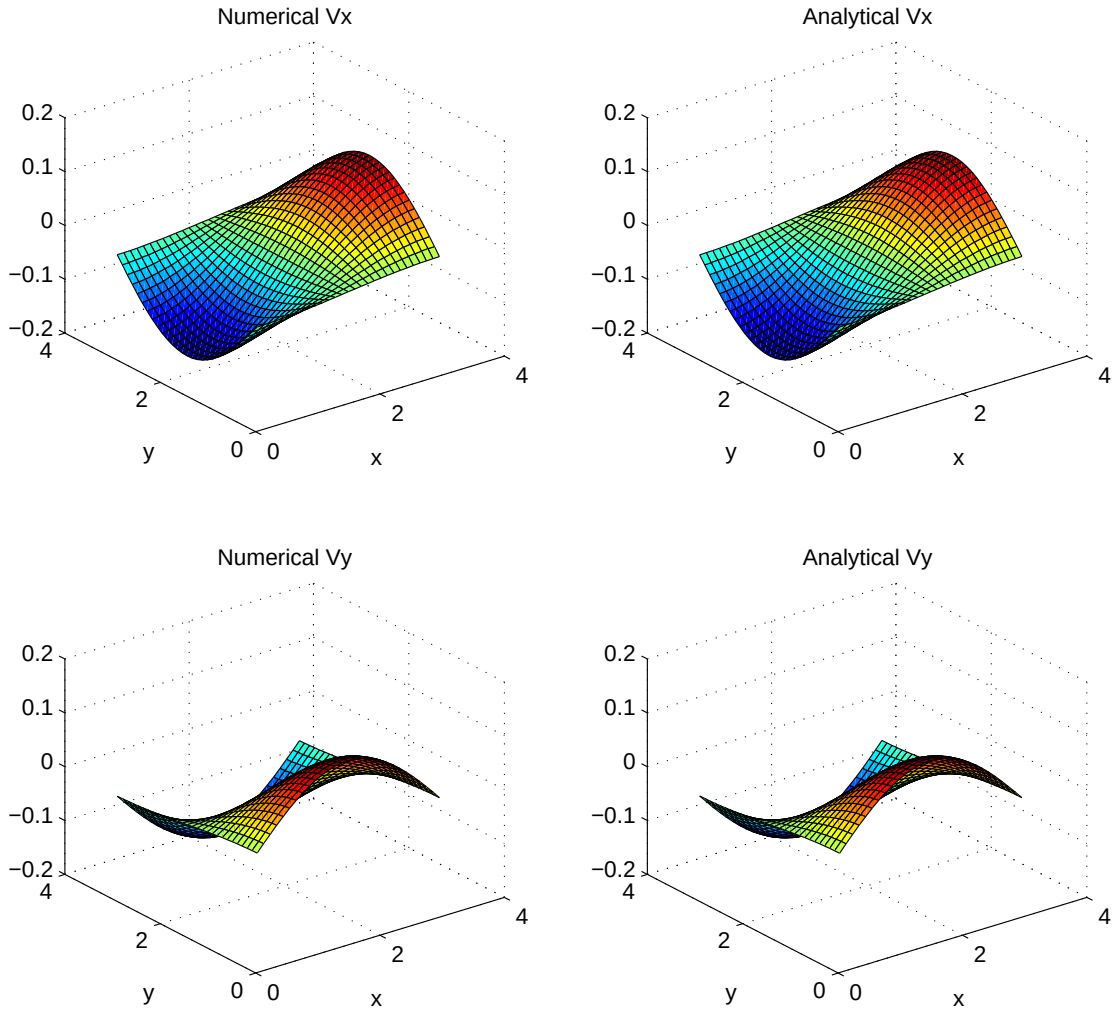

Figure 5.7: Analytical and numerical solutions of the velocity field in Test 2 by the IPCPP-IRBFN with $\Delta t = 0.005$ and the multistep time integration methods (BDF,AB) of order two.

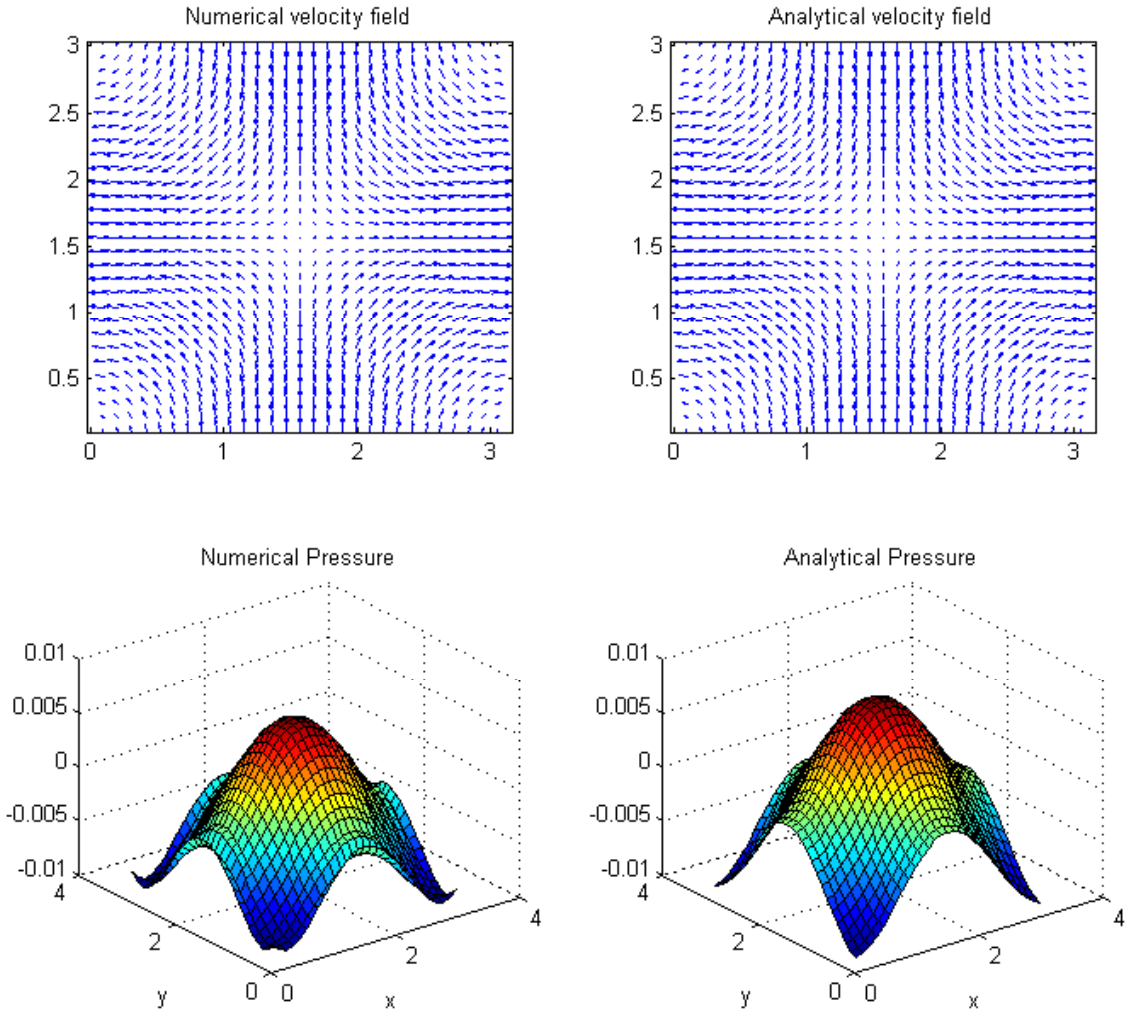

Figure 5.8: Analytical and numerical solutions of the pressure with dt=0.005 in Test 2.

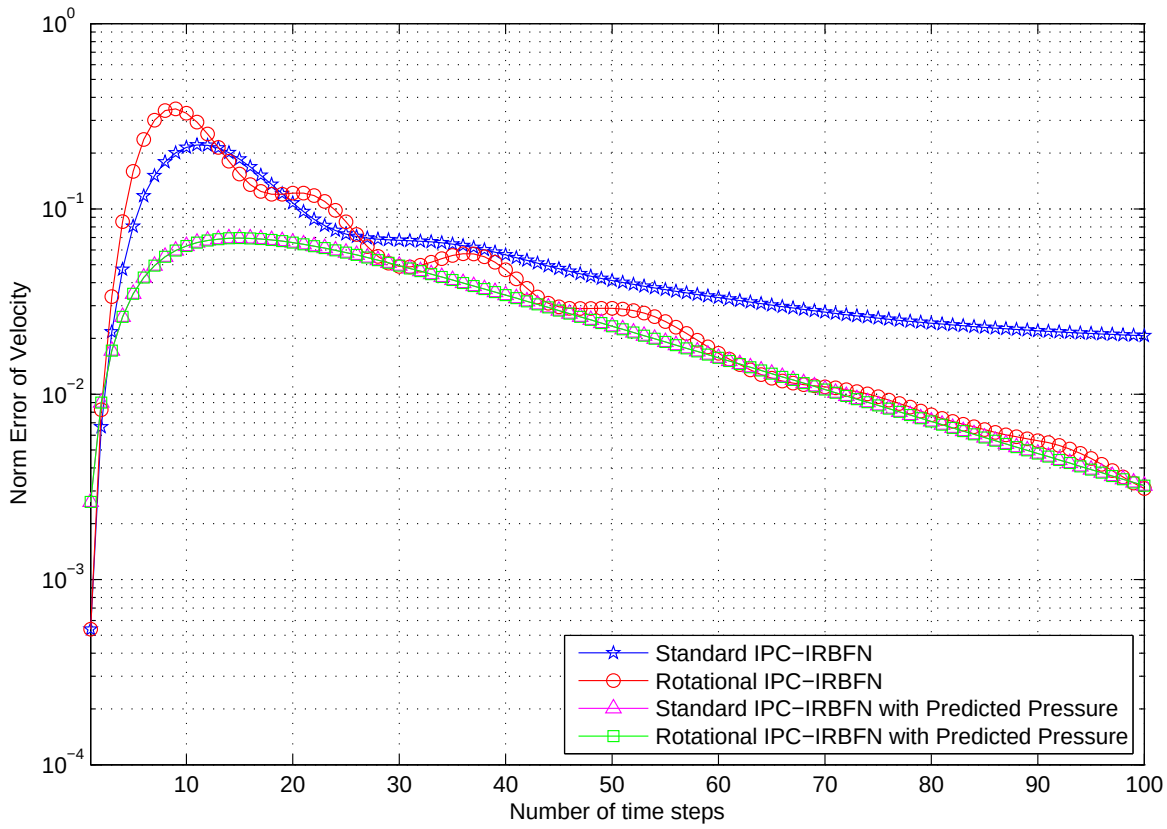

Figure 5.9: Stability analysis of the IPC-IRBFN and IPCPP-IRBFN schemes in terms of the velocity field with $\Delta t = 0.01$ in Test 2.

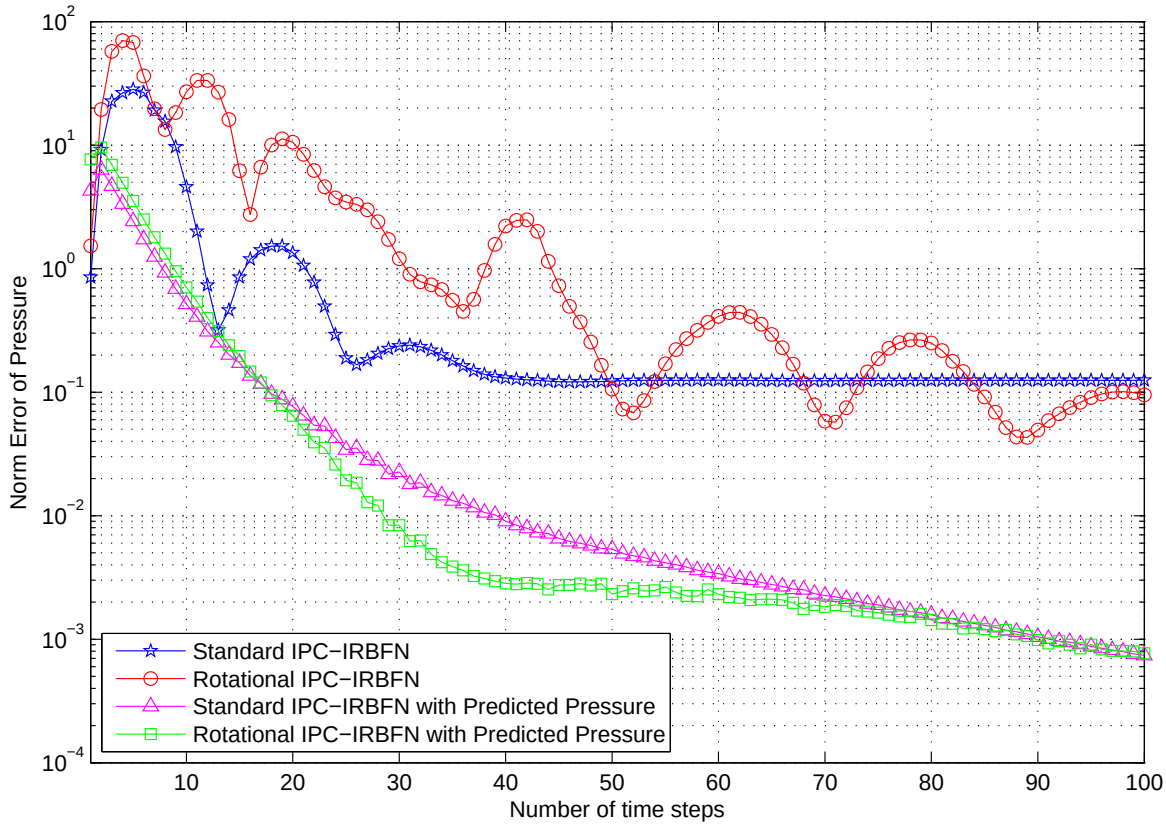

Figure 5.10: Stability analysis of the IPC-IRBFN and IPCPP-IRBFN schemes in terms of the pressure with $\Delta t = 0.01$ in Test 2.

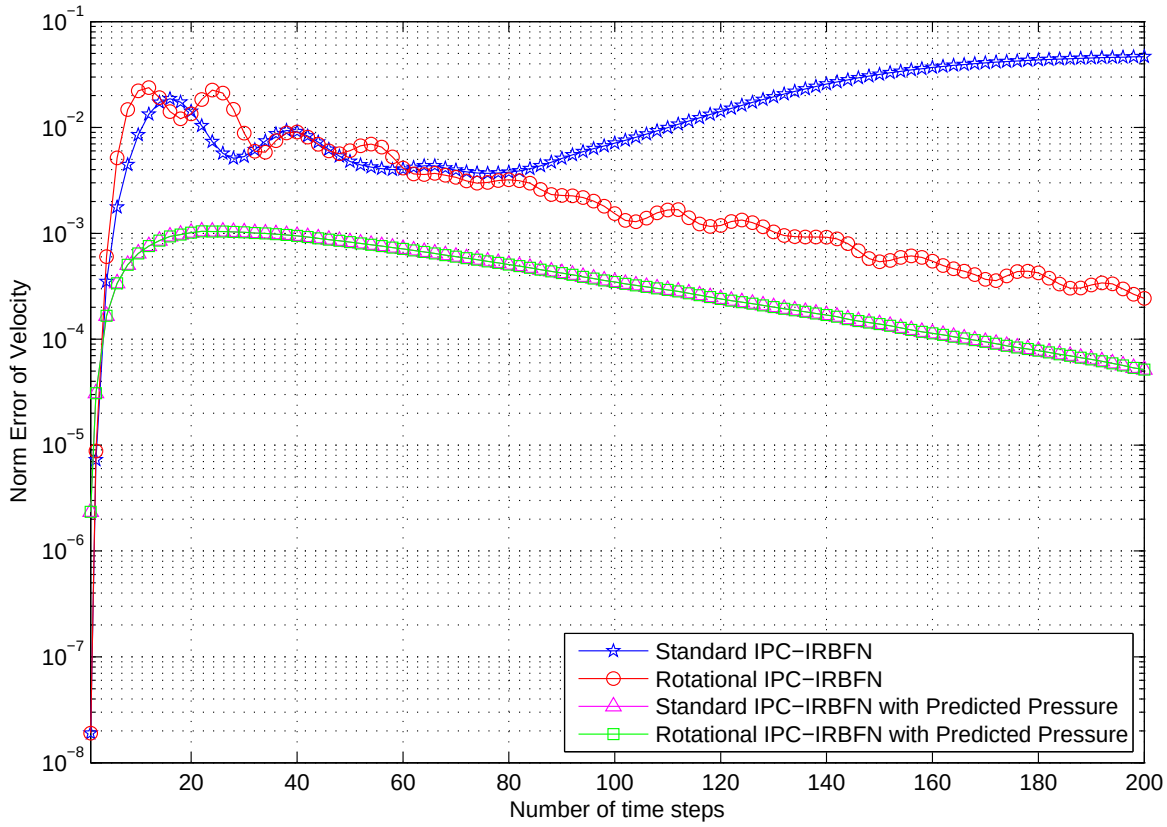

Figure 5.11: Stability analysis of the IPC-IRBFN and IPCPP-IRBFN schemes in terms of the velocity field with $\Delta t = 0.005$ in Test 2.

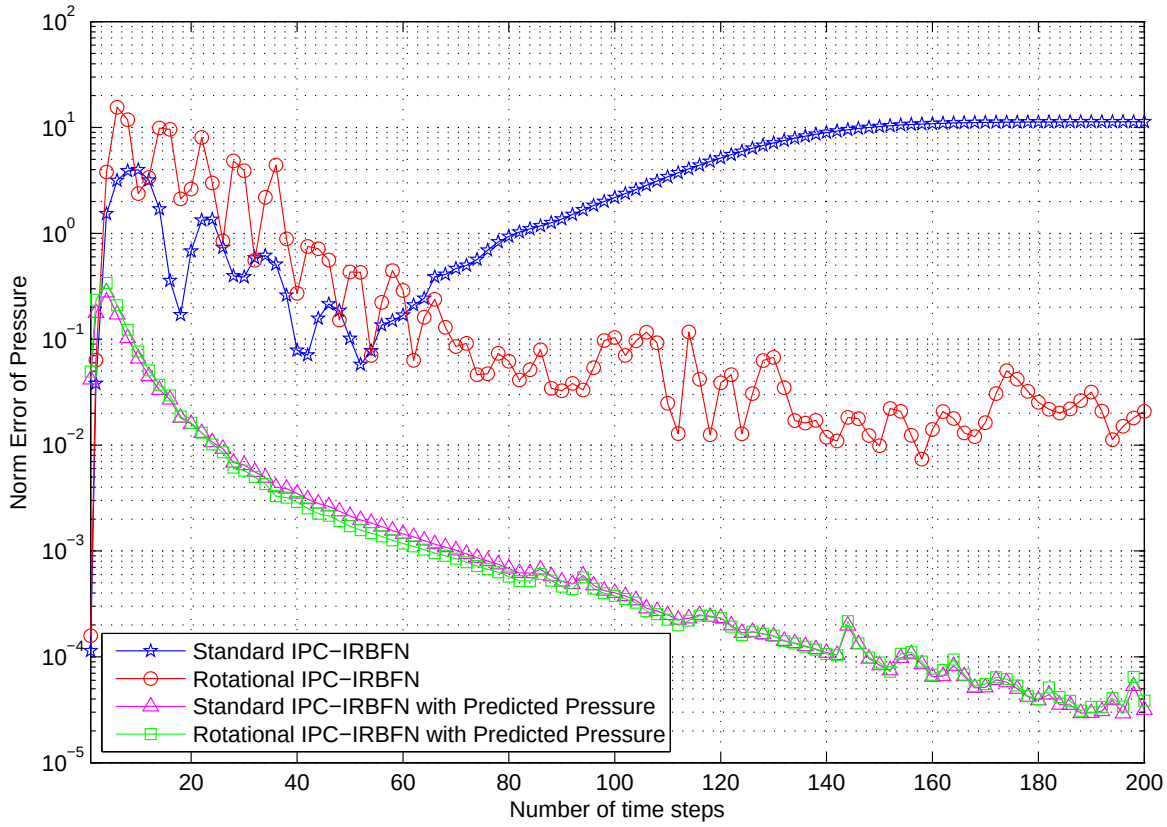

Figure 5.12: Stability analysis of the IPC-IRBFN and IPCPP-IRBFN schemes in terms of the pressure with $\Delta t = 0.005$ in Test 2.

5.4.3 Numerical analysis of the unsteady lid-driven cavity flow

Consider the lid-driven cavity flow in a unit square domain $\mathbf{\Omega} = [0, 1]^2$. The upper side of the cavity moves in its own plane at unit speed while the other sides are fixed. There is a discontinuity in the boundary conditions at the two upper corners of the cavity. There are two options in dealing with the discontinuity in numerical schemes for this problem: (a) The two upper corners are either considered as belonging to the moving upper side (leaky cavity); (b) they are assumed to belong to the fixed vertical walls (non-leaky). Option (a) is adopted in this work.

The objective of this analysis is to investigate the transient behavior of the lid-driven cavity flow, rather than on its steady-state solution. In particular, the influence of Reynolds number on the numerical solutions are of primary interest. Different values of the Reynolds number are used to study the effect of this dimensionless number on the numerical solution of such the flow. The IPC-IRBFN scheme in rotational form is used in this problem. A point density of 61 along each direction is used to well capture the vortices appearing near the cavity corners as well as the primary vortex. For this numerical analysis, the time-step size is fixed at $\Delta t = 0.001$.

Figures 5.13 shows the evolution of the velocity field along the mid-vertical line for the lid-driven cavity flow with Re=1000. It is observed that the rate of change in shape of the velocity along this line is rather fast in the beginning. This change slows down with time, and finally the velocity field reaches its steady state profile. Similarly, Figure 5.14 shows the evolution of the velocity along the mid-horizontal line.

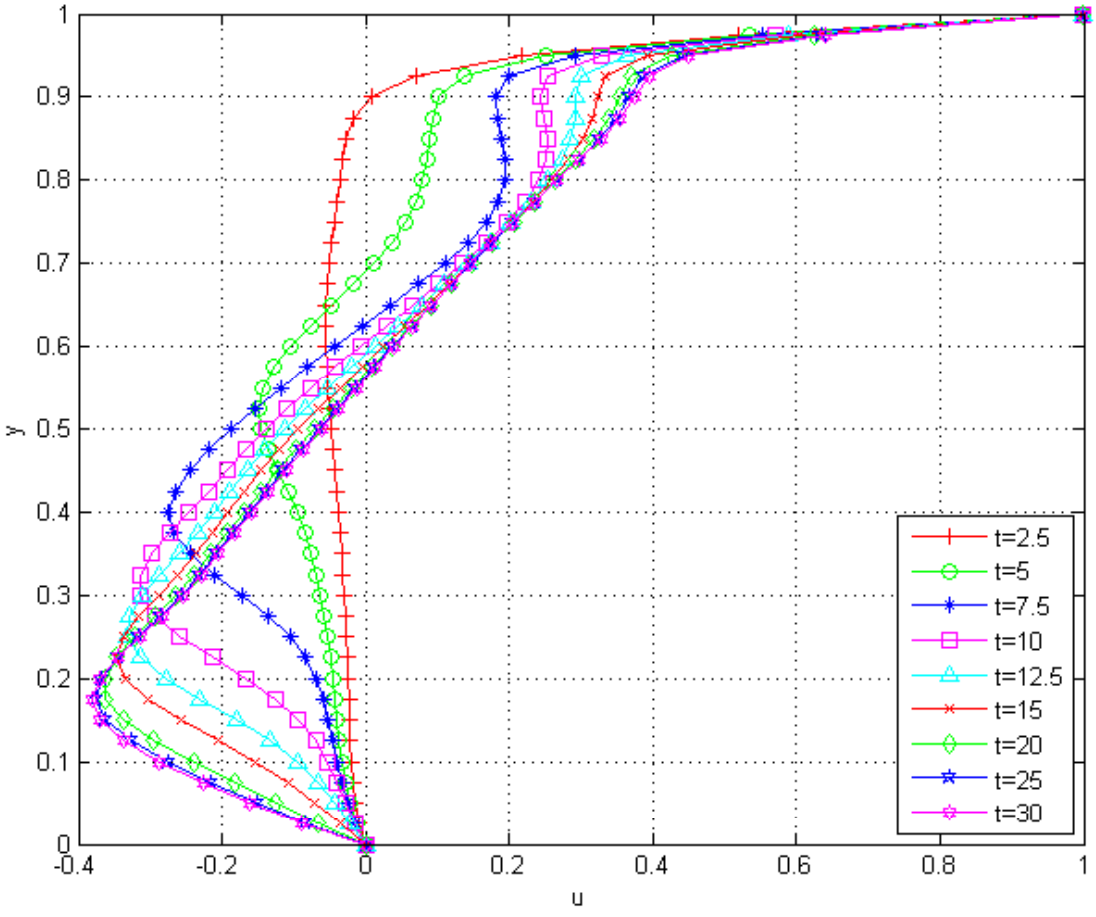

Figure 5.13: Convergence of the u-component velocity along the mid-vertical line in the lid-driven cavity flow.

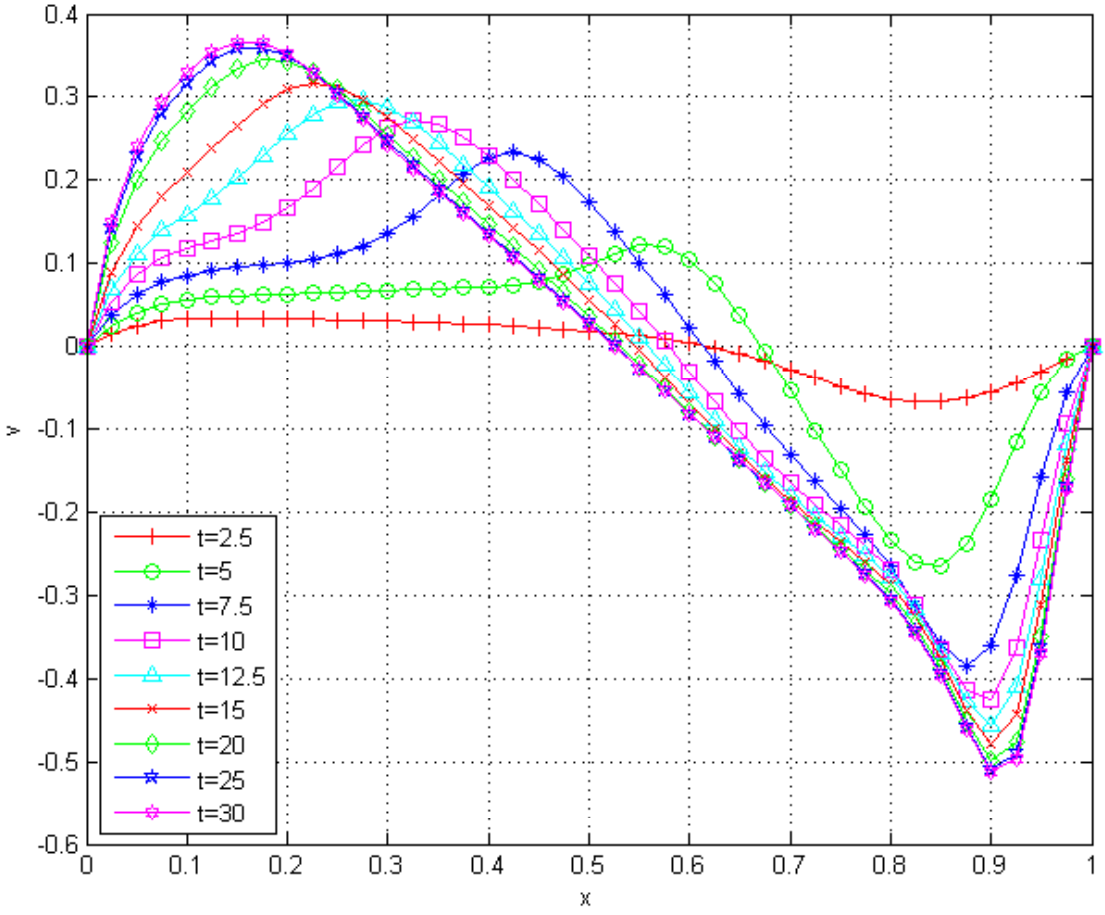

Figure 5.14: Convergence of the v-component velocity along the mid-horizontal line in the lid-driven cavity flow.

Table 5.1: Streamfunction ψ_c and vorticity ω_c at the center of the primary vortex at different time steps corresponding to Re=100,400 and 1000.

	Re=100	Re=400	Re=1000
t	ψ_c (ω_c)	ψ_c (ω_c)	ψ_c (ω_c)
1	-0.073624 (-3.668184)	-0.047149 (-5.825875)	-0.035008 (-8.523135)
2.5	-0.092665 (-3.551824)	-0.072358 (-4.363596)	-0.059933 (-6.84297)
5	-0.100557 (-3.247018)	-0.090033 (-3.441605)	-0.080071 (-4.62247)
7.5	-0.102477 (-3.181559)	-0.099643 (-2.819742)	-0.091677 (-3.596885)
10	-0.10298 (-3.147246)	-0.105461 (-2.55873)	-0.099589 (-3.049279)
12.5	-0.103111 (-3.147487)	-0.108959 (-2.427785)	-0.105255 (-2.710798)
15	-0.103145 (-3.147548)	-0.110951 (-2.366845)	-0.109306 (-2.484563)
17.5	-0.103154 (-3.147563)	-0.112037 (-2.327834)	-0.11212 (-2.333954)
20	-0.103156 (-3.147567)	-0.11262 (-2.305005)	-0.113977 (-2.231622)
25	-0.103157 (-3.147568)	-0.11309 (-2.296747)	-0.115874 (-2.115058)
30	-0.103157 (-3.147568)	-0.113224 (-2.291365)	-0.116652 (-2.062449)

In this work, the evolution of the streamfunction and the vorticity at the centre of the primary vortex at different points in time are analyzed with different Reynolds numbers. For any of the Reynolds numbers, the streamfunction values at the center of the primary vortex change stiffly in the beginning. The rate of change then slows down, and finally the streamfunction reaches its steady state. It can be seen in Figures (5.15-5.16) that the higher the Reynolds number is, the longer it takes for the two quantities to reach their steady state. The values of streamfunction and vorticity at the center of the primary vortex at different time step size captured by the IPC-IRBFN scheme are presented in Table 5.1. Figures (5.17-5.18) show the streamfunction contours at different points in time for the case with $Re = 1000$. The contours of vorticity are shown in Figures (5.19-5.20). As can be seen from the figures, at each time step, the new schemes well capture the primary vortex.

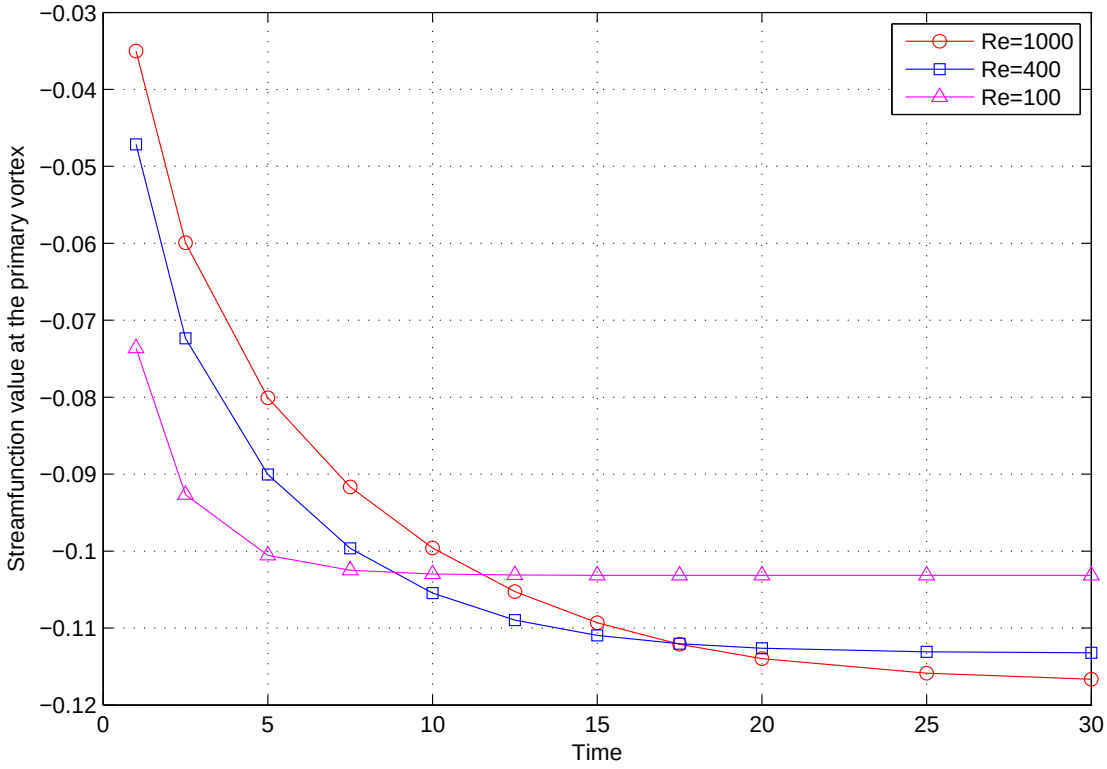

Figure 5.15: The evolution of the streamfunction at the centre of the primary vortex at different points in time with different Reynolds numbers. For any of the Reynolds numbers, the streamfunction values at the center of the primary vortex change stiffly in the beginning. The rate of change then slows down, and finally the streamfunction reaches its steady state.

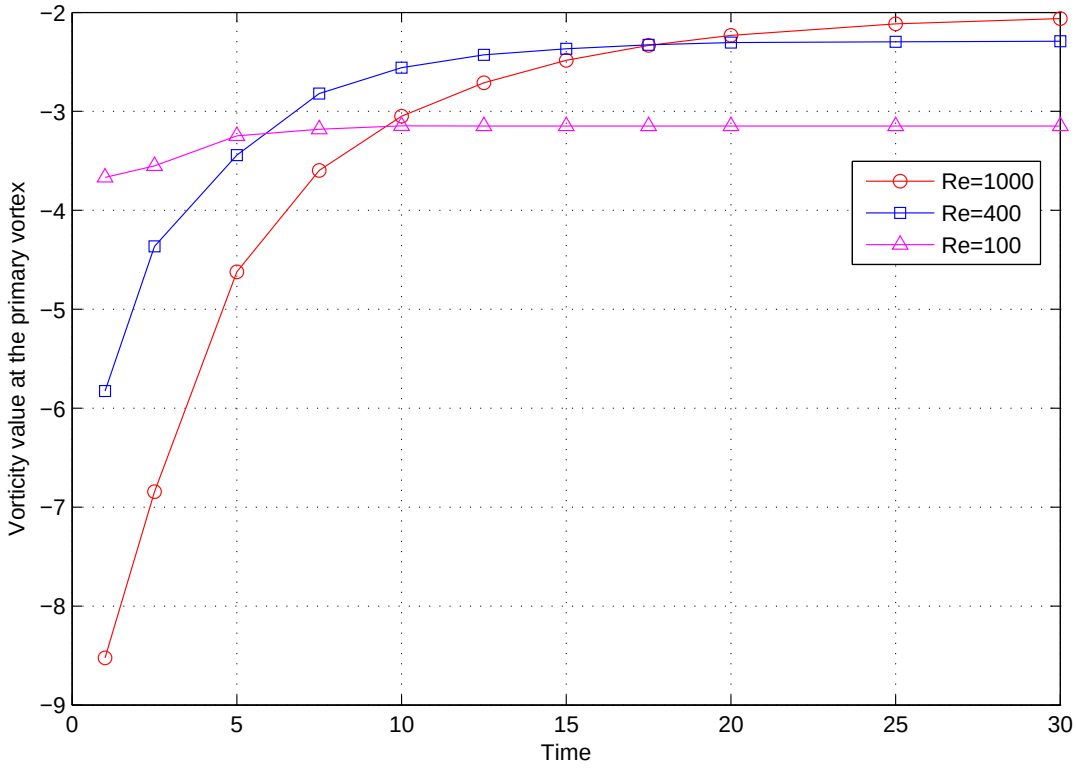

Figure 5.16: The evolution of the vorticity at the centre of primary vortex at different points in time with different Reynolds numbers. For any of the Reynolds numbers, the vorticity values at the center of the primary vortex change stiffly in the beginning. The rate of change then slows down, and finally the vorticity reaches its steady state.

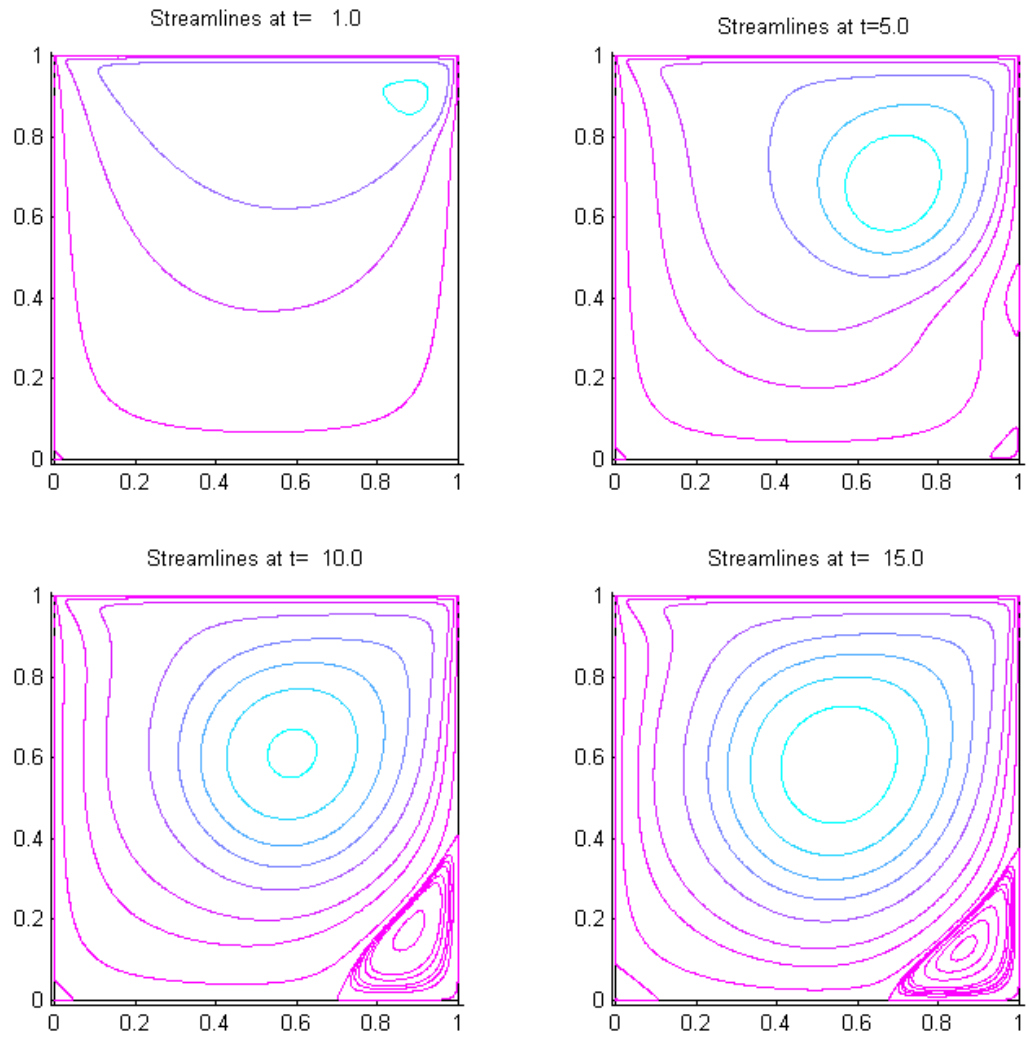

Figure 5.17: Streamlines at $t = 1, 5, 10, 15$ of the lid-driven cavity flow.

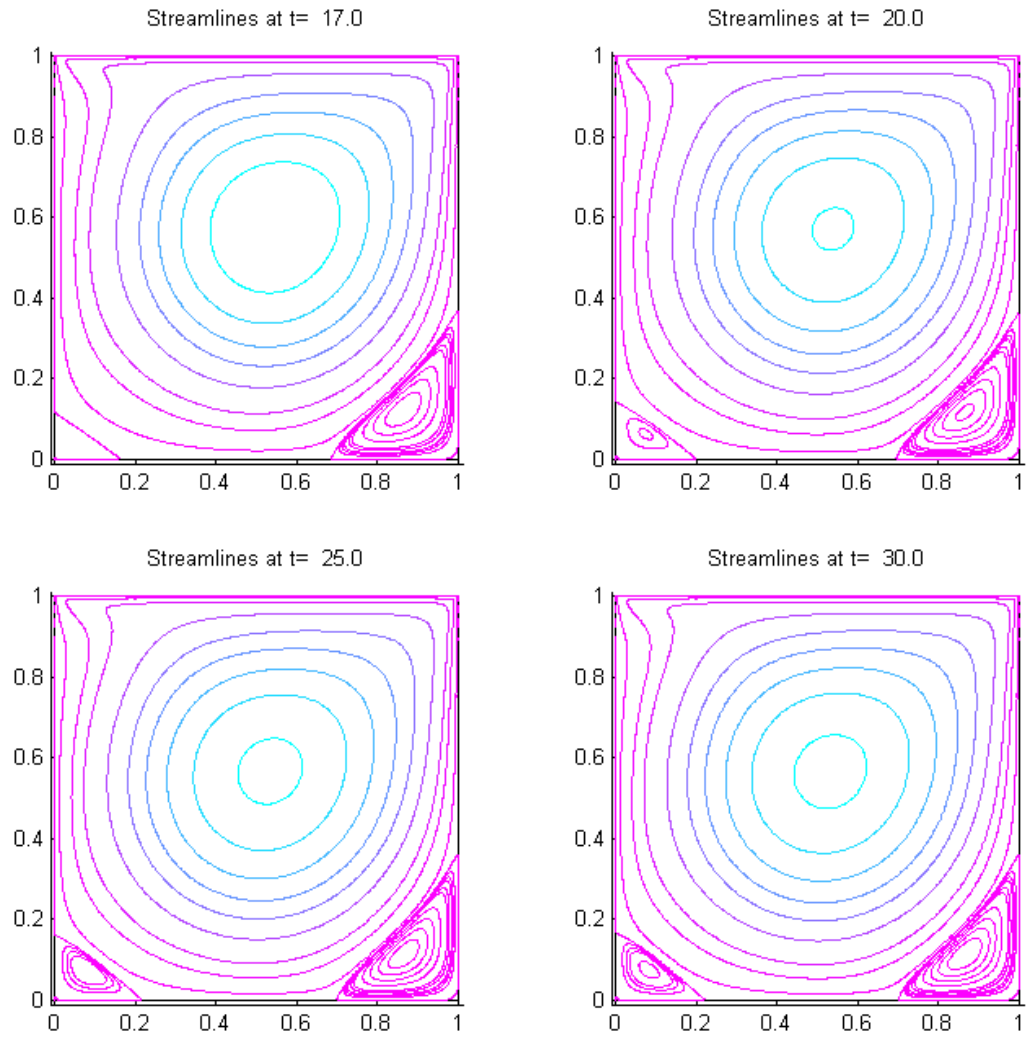

Figure 5.18: Streamlines at $t = 17, 20, 25, 30$ of the lid-driven cavity flow.

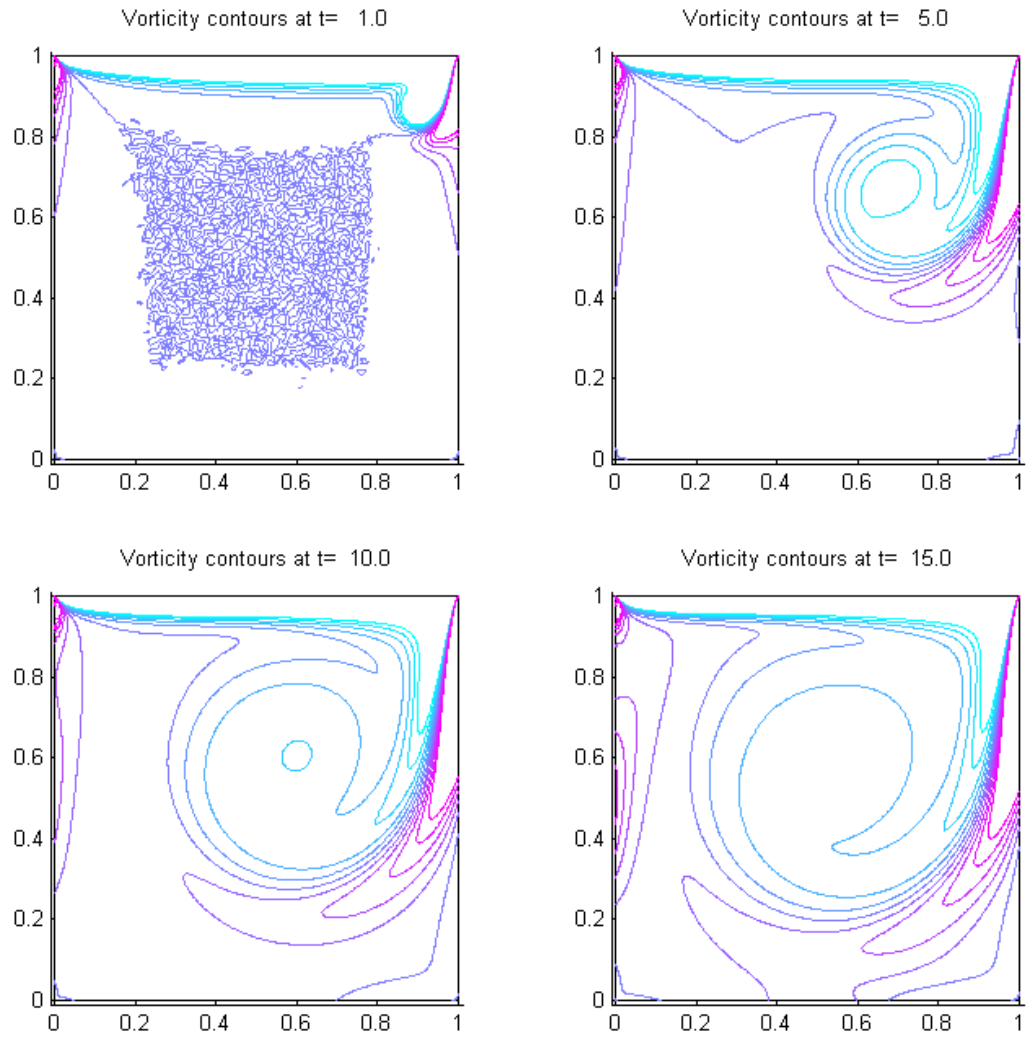

Figure 5.19: Contours of vorticity at $t = 1, 5, 10, 15$ of the lid-driven cavity flow.

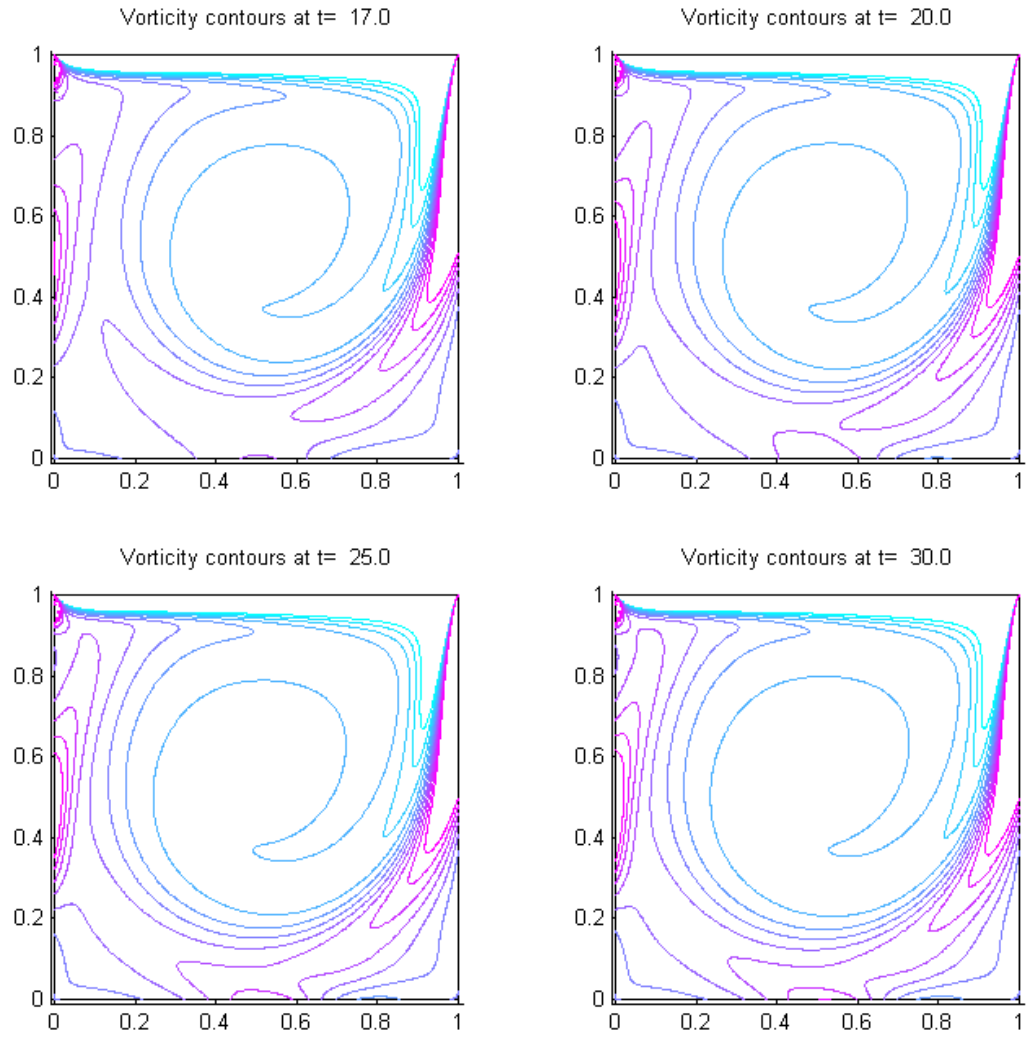

Figure 5.20: Contours of vorticity at $t = 17, 20, 25, 30$ of the lid-driven cavity flow.

Numerical solutions of unsteady lid-driven cavity flows are reported with Reynolds numbers up to 1000 which are widely reported in the literature. For larger Reynolds numbers, there are some notes to be mentioned as follows. Physically, the larger the Reynolds number is, the longer time it takes for the streamfunction and vorticity to reach steady state. This can be predicted from the analysis shown in Figures (5.15,5.16). Numerically, Reynolds number directly involves in Step 2 and Step 4 of the proposed schemes. In Equation (5.29) of Step 2, Re plays the role of a scaling factor for the discrete viscous term. This, however, has an insignificant effect on the condition number of the system matrix to be solved in this step. Indeed, a simple check shows that with $Re = (100, 1000, 3200)$, the condition number of the system matrix only changes in small range of $\mathcal{O}(10^{-2})$. Regarding Step 4 of the proposed schemes, as can be seen from Equation (5.50), the larger the Reynolds number is, the less contribution the divergence of the velocity predictor has to the pressure correction.

5.5 Concluding Remarks

Two novel meshless schemes, namely IPC-IRBFN and IPCPP-IRBFN, for the numerical solution of the Navier-Stokes equations have been presented in this chapter. The two schemes are based on the projection method in the combination with the IRBFN method for spatial discretization and high-order multistep time integration methods. By decoupling the treatment of the pressure and velocity field, the new schemes decompose the Navier-Stokes equations into a set of smaller and easier-to-solve subproblems. This well suits for RBF-based methods whose system matrices are generally dense. In fact, the proposed schemes significantly alleviates the storage requirement and computational time by solving smaller systems one at a time. Numerical experiments in solving unsteady Stokes, Navier-Stokes equations with known analytical solutions and the lid-driven cavity flow show the good capability of the two schemes for solving unsteady incompressible viscous flows.

Chapter 6

Meshless Approach to Interfacial Flows

This chapter reports a novel meshless approach to the numerical simulation of interfacial flows in which the motion and deformation of the interface between two immiscible fluids are fully investigated. Unlike the passive transport problems studied in Chapter 4 where the influence of the moving interface on the surrounding fluid is ignored, the interfacial flows are studied in this chapter with the surface tension taken into account. As a result, not only the position and shape of the moving interface but also the ambient flow variables (velocity field and pressure) change when the interface moves. In other words, a two-way interaction between the moving interface and the ambient flow is fully investigated. In this chapter, the moving interface is captured by the level set method and the Navier-Stokes equations (with the surface tension embedded into the momentum equation) are used to modelled the ambient incompressible viscous flow. A numerical simulation of two bubbles moving, stretching and merging in an incompressible viscous fluid is performed to demonstrate the working of the proposed approach.

6.1 Introduction

Fluid flows with free moving interfaces between immiscible fluids can be classified as interfacial flows. Numerical methods for such flows are faced with several intrinsic difficulties. First, large differences in density and viscosity of the two fluids across the interface require appropriate treatments for the sharp interface resolution. Next, the moving interface might be subject to topological changes such as interface folding, breaking and merging. These phenomena must be numerically modelled with reliable algorithms. In addition, mass conservation is of primary concerns in numerical modelling of interfacial flows due to the existence of the fluid-fluid boundary (i.e. the moving interface itself).

In general, a numerical approach to modelling interfacial flows consists of a flow modelling algorithm, an interface modelling method and a flow-interface coupling technique. A literature review of flow modelling algorithms and interface modelling methods for general moving boundary problems have been presented in Section §4.1. In this work, the level set method (Osher and Sethian, 1988) belonging to fixed grid capturing methods is used for interface modelling. Regarding flow modelling for interfacial flows of two immiscible fluids, the Navier-Stokes equations for each fluid are coupled together via an Heaviside function of the level set function (Sussman and Smereka, 1997) so that the resultant equations to be solved are in a similar form as in one-fluid flow problems (Chang et al., 1996).

With regard to flow-interface coupling in the numerical simulation of interfacial flows, the manner in which the surface tension is modelled is of primary concerns. Normally, the surface tension is taken into account in the computation of force balance at the interface where the difference in stress tensors of the two fluids in the direction normal to the interface is equal to the surface tension force on the interface (Floryan and Rasmussen, 1989). Naive implementations of such a interfacial boundary conditions in traditional methods have suffered

from difficulties in modelling topologically complex interfaces. A numerical model for surface tension that alleviates these interface topology constraints was presented by Brackbill et al. (1992). The proposed model, known as the continuum surface force (CSF) model interprets surface tension as a continuous, three-dimensional effect across an interface rather than as a boundary value condition on the interface. The advantage of this model is that the interface needs not to be explicitly described in order to apply the interfacial boundary condition. This well suits the interface capturing algorithm using the level set method which is used in this work.

Another numerical aspect of flow-interface coupling in modelling interfacial flows is how to update the flow variables and the interface. In an integrated manner the flow variables and the new interface position are sought simultaneously. This approach yields accurate and stable solutions but computationally expensive. In the segregated approach the flow is first computed with the previous information of the interface whose new position can be then found using the recently updated flow variables. In general, this approach is more efficient but less stable than its integrated counterpart.

6.2 Mathematical Formulation

6.2.1 The Two-Fluid Navier-Stokes Equations

Consider a domain Ω and its boundary $\partial\Omega$ containing two immiscible Newtonian fluids, both being incompressible. Let Ω_1 be the region containing fluid 1 at time t. Similarly, let Ω_2 be the region containing fluid 2 and bounded by the fluid interface Γ at time t. The governing equations describing the motion of the two fluid in their own regions are given by the Navier-Stokes equations,

$$\rho_1 \left(\frac{\partial \mathbf{v}_1}{\partial t} + \mathbf{v}_1 \cdot \nabla \mathbf{v}_1 \right) = -\nabla p_1 + \nabla \cdot (2\mu_1 \boldsymbol{\mathcal{D}}_1) + \rho_1 \boldsymbol{g}, \quad \boldsymbol{x} \in \Omega_1, \tag{6.1}$$

$$\rho_2 \left(\frac{\partial \mathbf{v}_2}{\partial t} + \mathbf{v}_2 \cdot \nabla \mathbf{v}_2 \right) = -\nabla p_2 + \nabla \cdot (2\mu_2 \boldsymbol{\mathcal{D}}_2) + \rho_2 \boldsymbol{g}, \quad \boldsymbol{x} \in \Omega_2, \tag{6.2}$$

with incompressibility constraints

$$\nabla \cdot \mathbf{v}_1 = 0, \quad \boldsymbol{x} \in \Omega_1, \tag{6.3}$$

$$\nabla \cdot \mathbf{v}_2 = 0, \quad \boldsymbol{x} \in \Omega_2, \tag{6.4}$$

where $\mathbf{v}_i$ is the velocity field, ρ_i is the density, $\boldsymbol{g}$ is the gravity, p_i is the pressure and μ_i is the viscosity. The rate of strain tensor $\boldsymbol{\mathcal{D}}_i$ is defined as

$$\boldsymbol{\mathcal{D}}_i = \frac{1}{2} \left(\nabla \mathbf{v}_i + \nabla \mathbf{v}_i^T \right), \quad i = 1, 2. \tag{6.5}$$

The subscript i in the above equation denotes the i^{th} fluid under consideration. The moving interface Γ is impermeable and assuming no mass transfer between the two fluids yields a continuous velocity condition at the interface

$$\mathbf{v}_1 = \mathbf{v}_2, \quad \boldsymbol{x} \in \Gamma. \tag{6.6}$$

The jump in normal stresses along the fluid interface is balanced with the surface tension as follows.

$$(2\mu_1 \boldsymbol{\mathcal{D}}_1 - 2\mu_2 \boldsymbol{\mathcal{D}}_2) \cdot \boldsymbol{n} = (p_1 - p_2 + \sigma\kappa)\, \boldsymbol{n}, \quad \boldsymbol{x} \in \Gamma, \tag{6.7}$$

where κ is the curvature of the interface, σ the surface tension coefficient, and $\boldsymbol{n}$ is the unit normal vector along the fluid interface pointing outwards from fluid 2 (bounded by the interface) into fluid 1 (the surrounding fluid). As mentioned in Section §6.1, the continuum surface force (CSF) model is used in this work to embed the surface tension into the momentum equation rather than imposing the above equations on the moving interface (Brackbill et al., 1992).

Let the fluid interface be the zero level of the level set function ϕ,

$$\Gamma = \boldsymbol{x} \mid \phi(\boldsymbol{x}, t) = 0, \tag{6.8}$$

where

$$\phi(\boldsymbol{x}, t) \begin{cases} > 0 & \boldsymbol{x} \in \Omega_1 \\ 0 & \boldsymbol{x} \in \Gamma \\ < 0 & \boldsymbol{x} \in \Omega_2 \end{cases} \tag{6.9}$$

The unit normal on the interface, drawn from the interior into the exterior region, and the curvature of the interface can be expressed in terms of $\phi(\boldsymbol{x}, t)$ as follows.

$$\boldsymbol{n} = \left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0} \quad \text{and} \quad \kappa = \nabla \cdot \left.\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right|_{\phi=0} \tag{6.10}$$

Let

$$\mathbf{v} = \begin{cases} \mathbf{v}_1 & \boldsymbol{x} \in \Omega_1 \\ \mathbf{v}_2 & \boldsymbol{x} \in \Omega_2 \end{cases} \tag{6.11}$$

be the fluid velocity continuous across the interface, since the interface moves with the fluid particles, the evolution of ϕ is then given by (Osher and Fedkiw, 2003).

$$\frac{\partial\phi}{\partial t} + \mathbf{v} \cdot \nabla\phi = 0. \tag{6.12}$$

By defining the Heaviside function $H(\phi)$

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < 0, \\ 1/2 & \text{if } \phi = 0, \\ 1 & \text{if } \phi > 0, \end{cases} \tag{6.13}$$

and the fluid properties

$$\rho(\phi) = \rho_2 + (\rho_1 - \rho_2)H(\phi), \tag{6.14}$$

$$\mu(\phi) = \mu_2 + (\mu_1 - \mu_2)H(\phi), \tag{6.15}$$

together with the CSF model (Brackbill et al., 1992), one obtains the Navier-Stokes equations for two immiscible fluids known as the one-fluid continuum formulation (Chang et al., 1996)

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\,\mathbf{v} = \frac{1}{\rho(\phi)}\left(-\nabla p + \nabla \cdot (2\mu(\phi)\boldsymbol{\mathcal{D}}) + \sigma\kappa(\phi)\delta(d)\boldsymbol{n}\right) + \boldsymbol{g}, \tag{6.16}$$

$$\nabla \cdot \mathbf{v} = 0. \tag{6.17}$$

where $\boldsymbol{f}_c = \sigma\kappa(\phi)\delta(d)\boldsymbol{n}$ is the surface tension, $\delta(d)$ is the Dirac delta function, d is the normal distance to the interface. For the numerical simulation of bubbles rising in a viscous fluid studied in this chapter, Equation (6.16) can be written in dimensionless form as follows.

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\,\mathbf{v} = \frac{1}{\rho(\phi)}\left(-\nabla p + \frac{1}{Re}\nabla \cdot (2\mu(\phi)\boldsymbol{\mathcal{D}}) + \frac{1}{B_o}\kappa(\phi)\delta(\phi)\nabla\phi\right) + \boldsymbol{g}_u \tag{6.18}$$

where the scaling factors are

$$p^* = \frac{p}{p_{ref}}, \quad \mathbf{v}^* = \frac{\mathbf{v}}{v_{ref}}, \quad \mathbf{x}^* = \frac{\mathbf{x}}{R},$$

$$t^* = \frac{t}{t_{ref}}, \quad \rho^* = \frac{\rho}{\rho_c}, \quad \mu^* = \frac{\mu}{\mu_c},$$

$$v_{ref} = (gR)^{1/2}, \quad p_{ref} = \rho_c v_{ref}^2, \quad t_{ref} = \frac{v_{ref}}{R}. \tag{6.19}$$

Fluid 1 is hereafter referred to as the fluid surrounding the bubble with density ρ_c and viscosity μ_c. Similarly, fluid 2 is referred to as the fluid inside the bubble, of initial radius R, that has the corresponding density ρ_b and viscosity μ_b. $\boldsymbol{g}_u$

is the unit gravitational vector pointing downward. The dimensionless groups in the above equation are the Reynolds number

$$Re = \frac{(2R)^{3/2}\sqrt{g}\rho_c}{\mu_c} \tag{6.20}$$

and the Bond number

$$Bo = \frac{4\rho_c g R^2}{\sigma} \tag{6.21}$$

The dimensionless density and viscosity in Equation (6.18) are defined as

$$\rho(\phi) = \lambda + (1-\lambda)H(\phi), \quad \text{and} \quad \mu(\phi) = \eta + (1-\eta)H(\phi), \tag{6.22}$$

where $\lambda = \rho_b/\rho_c$ is the density ratio, $\eta = \mu_b/\mu_c$ is the viscosity ratio.

6.3 Numerical Schemes

This section presents a new hybrid approach to interfacial flows in which the meshless schemes for capturing moving interfaces presented in Chapter 4 and the IRBFN-based IPC schemes for the Navier-Stokes equations in Chapter 5 are brought together to solve the problem in the meshless framework of the IRBFN method. The solving procedure consists of the following steps.

Step 0: Initialize the level set function $\phi(\boldsymbol{x})$ to be the signed distance to the interface as described by equation (4.1);

For each n^{th} time step, $n = 1, 2, ...$

Step 1: Compute the interface normal, curvature, and the density and viscosity of the fluids;

Step 2: Solve the two-fluid Navier-Stokes equations using the meshless projec-

tion schemes similar to the IPC-IRBFN schemes presented in Chapter 5 taking into account the interface dependence of density and viscosity as well as the surface tension;

Step 3: Advance the level set function from the previous step to the current one with the most updated velocity field calculated in Step 2;

Step 4: Re-initialize the level set function to a signed distance function at the current time step;

Step 5: Adjust the level set function by using the mass correction algorithm to ensure the mass conservation;

Step 6: The interface as the zero contour of the level set function has now been advanced one time step. Go back to step 1 for further evolution of the moving interface until the predefined time is reached.

The next sections describe in details each step listed above. For those already presented in the previous chapters, they are briefly recaptured here only for completeness.

6.3.1 Computing interface properties (normal and curvature) and fluid properties (density and viscosity)

The normal and curvature of the interface can be calculated by Equation (6.10) whereas the density and viscosity are given by Equation (6.22). For the computation of the above fluid properties, the Heaviside function is used. A naive implementation of Equation (6.13) poses numerical difficulty since large jumps in ρ and μ across the interface might cause numerical instabilities. In order to avoid this issue, it is common to introduce an interface thickness to smooth the density and viscosity at the interface. This can be done by replacing the Heav-

iside function in Equation (6.13) with a smoothed Heaviside function $H_\epsilon(\phi)$ defined as (Chang et al., 1996)

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ (\phi+\epsilon)/(2\epsilon) + \sin(\pi\phi/\epsilon)/(2\pi) & \text{if} |\phi| \leq \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \tag{6.23}$$

The above Heaviside function defines the smoothed Dirac delta function δ_ϵ as follows.

$$\delta_\epsilon(\phi) = \begin{cases} 1/2(1+\cos(\pi\phi/\epsilon))/\epsilon & \text{if } |\phi| < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{6.24}$$

6.3.2 Solving the two-fluid Navier-Stokes equations

The IPC-IRBFN scheme in Section §5.3 is applied for solving the two-fluid Navier-Stokes equations as follows. First, the equations are rewritten in such a way that the surface tension and gravity forces in (6.18) are treated as the forcing term $\bar{\boldsymbol{f}}$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\,\mathbf{v} = -\frac{1}{\rho(\phi)}\nabla p + \frac{1}{\rho(\phi)Re}\nabla\cdot(2\mu(\phi)\boldsymbol{\mathcal{D}}(\mathbf{v})) + \bar{\boldsymbol{f}} \quad \text{in } \Omega \tag{6.25}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{on } \Omega, \tag{6.26}$$

$$\mathbf{v} = \mathbf{v}_b \quad \text{on}\partial\Omega. \tag{6.27}$$

where $\mathbf{v}_b$ is the Dirichlet boundary condition for velocity, and $\bar{\boldsymbol{f}}$ is given by

$$\bar{\boldsymbol{f}} = \boldsymbol{g}_u + \frac{1}{\rho(\phi)B_o}\kappa(\phi)\delta(\phi)\nabla\phi. \tag{6.28}$$

The IPC-IRBFN scheme is applied to solve the two-fluid Navier-Stokes equations as follows.

Calculating the velocity predictor

$$\beta_0\tilde{\mathbf{v}}^{n+1}-\frac{1}{\rho(\phi^n)Re}\nabla\cdot\left(2\mu(\phi^n)\boldsymbol{\mathcal{D}}(\tilde{\mathbf{v}}^{n+1})\right)=\sum_{k=1}^{J_v}\beta_k\mathbf{v}^{n+1-k}-\sum_{k=0}^{J_v-1}\alpha_k\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right]^{n-k} - \frac{\Delta t}{\rho(\phi^n)}\nabla\bar{p}^{n+1}+\Delta t\bar{\mathbf{f}}^{n+1} \quad \text{in } \Omega \tag{6.29}$$

$$\tilde{\mathbf{v}}^{n+1}=\mathbf{v}_b^{n+1} \quad \text{on } \Omega. \tag{6.30}$$

Calculating the pressure

$$\nabla\cdot\left(\frac{1}{\rho(\phi^n)}\nabla q^{n+1}\right)=\frac{\beta_0}{\Delta t}\nabla\cdot\tilde{\mathbf{v}}^{n+1} \quad \text{in}\Omega, \tag{6.31}$$

$$\frac{\partial q^{n+1}}{\partial n}=0 \quad \text{on } \partial\Omega, \tag{6.32}$$

$$p^{n+1}=q^{n+1}+\bar{p}^{n+1}. \tag{6.33}$$

Updating the velocity

$$\mathbf{v}^{n+1}=\tilde{\mathbf{v}}^{n+1}-\frac{\beta_0\Delta t}{\rho(\phi^n)}\nabla q^{n+1}. \tag{6.34}$$

In the above equations, the dimensionless density and viscosity $\rho(\phi^n)$ and $\mu(\phi^n)$ are given by

$$\rho(\phi^n)=\lambda+(1-\lambda)H_\epsilon(\phi^n), \quad \lambda=\rho_b/\rho_c, \tag{6.35}$$

$$\mu(\phi^n)=\eta+(1-\eta)H_\epsilon(\phi^n), \quad \eta=\mu_b/\mu_c. \tag{6.36}$$

where the smoothed Heaviside function $H_\epsilon(\phi^n)$ is calculated from Equation (6.23) with the corresponding value of the level set function ϕ^n.

6.3.3 Advancing the level set function

The level set function is advanced by solving the convective transport equation

$$\phi_t + \mathbf{v} \cdot \nabla\phi = 0, \quad \phi(\boldsymbol{x}, t = 0) = \phi_0(\boldsymbol{x}), \tag{6.37}$$

for one time step using either SL-IRBFN or Taylor-IRBFN schemes presented in Section §4.3 and §4.4, respectively;

6.3.4 Re-initializing the level set function

Due to numerical error, the level set function is not necessarily a distance function as desired even after one time step. Reinitialization is therefore needed to make the level set function signed distance function after certain time steps which could be achieved by solving the following time-dependent PDE to steady state (Sussman et al., 1994).

$$\phi_t = S_\epsilon(\bar{\phi})(1 - |\nabla\phi|), \quad \phi(x, y, t = 0) = \bar{\phi}(x, y) \tag{6.38}$$

where S_ϵ denotes the smoothed sign function

$$S_\epsilon(\bar{\phi}) = \frac{\bar{\phi}}{\sqrt{\bar{\phi}^2 + \epsilon^2}} \tag{6.39}$$

in which ϵ can be chosen to be the minimum distance from any data point to the others. Equation (6.38) can be solved to steady-state using the semi-implicit IRBFN-based scheme with the fourth-order Runge-Kutta presented in Section §3.3.3, Chapter 3.

6.3.5 Adjusting the level set function with the mass correction algorithm

The reinitialization procedure might introduce some numerical diffusion which results in an inaccuracy of the interface location and some loss of mass (Tornberg, 2000). The mass correction is then performed to prevent any losses of mass. After advancing the level set function at time step $t = t^{n+1}$, one gets the moving interface Γ that bounds the domain $\Omega_2 = \boldsymbol{x} \in \Omega : \phi < 0$. To correct the area of Ω_2, one changes the zero level set to certain neighboring isoline based on the fact that it has almost the same shape since ϕ is a distance function. This can be done by simply moving the level set function upward or downward by an amount of c_ϕ, where $|c_\phi|$ is the distance between old and new zero-level sets

$$\phi^{new} = \phi - c_\phi, \tag{6.40}$$

where ϕ^{new} is the new (raised or lowered) level set function, $\Omega_2^{new} = \{\boldsymbol{x} \in \Omega : \phi^{new} < 0\}$. Details of the mass correction was described in Section §4.5.4.

6.4 Numerical results

This section reports the application of the new meshless approach to simulate the motion and deformation of the two bubbles in an interaction with the surrounding fluid flow. In this numerical experiment, a rectangular cavity is filled up with two immiscible fluids where the heavier one settles at the bottom and the lighter one at the top. Two bubbles, containing the same light fluid as in the top layer, are initially embedded in the heavier fluid at the bottom, one above the other. The bubbles are then released from rest and allowed to rise by buoyancy force. The five primary parameters are chosen as follows: Reynolds number, $Re = 10$; Bond number, $Bo = 5$; Density ratio, $\lambda = 1/10$; Viscosity

ratio, $\eta = 1$. It is noted that the density ratio λ indicates that the fluid inside the bubbles and in the top layer is ten times lighter than the heavier fluid. For this numerical simulation, the new hybrid approach presented in Section §(6.3) is used on an regular computational grid with 21 points in x-direction and 41 points in y-direction.

As the bubbles are lighter than the surrounding fluid, they will rise with time. The two bubbles start moving upwards from the bottom of the cavity due to the gravity force as can be seen in Figure 6.1. During the move, the bubbles merge together and continuously affect on the surrounding fluid flow indicated by the change in direction and magnitude of the velocity around the bubbles, as shown in Figures (6.2-6.4). The merging bubbles finally reach the free surface in the upper part of the cavity and totally diffuse into the body of fluid of the top layer as shown in Figures (6.5-6.10).

As can be observed from the figures, although having the same density and viscosity, the lower bubble moves faster than the upper one. This can be explained by the wake formation below the upper bubble. As the time evolves, the lower one becomes entrapped into the wake region identified by the large magnitude of the velocity field below the upper bubble making the lower one move faster.

When the bubbles get closer to the free surface as in Figures 6.3, due to its surface tension, the free surface tends to prevent the upward motion of the bubble making them flatten remarkably. In their turn, the bubbles in keeping their trend to move upward make the free surface bend upwards more significantly. The figures clearly show the effect of the surface tension in keeping the kinematic equilibrium on the free surface.

In addition, the presence of the vorticities in Figures 6.7 indicates the effect of the surface tension along the free surface on the velocity field even when the bubbles completely diffuse into the surrounding fluid.

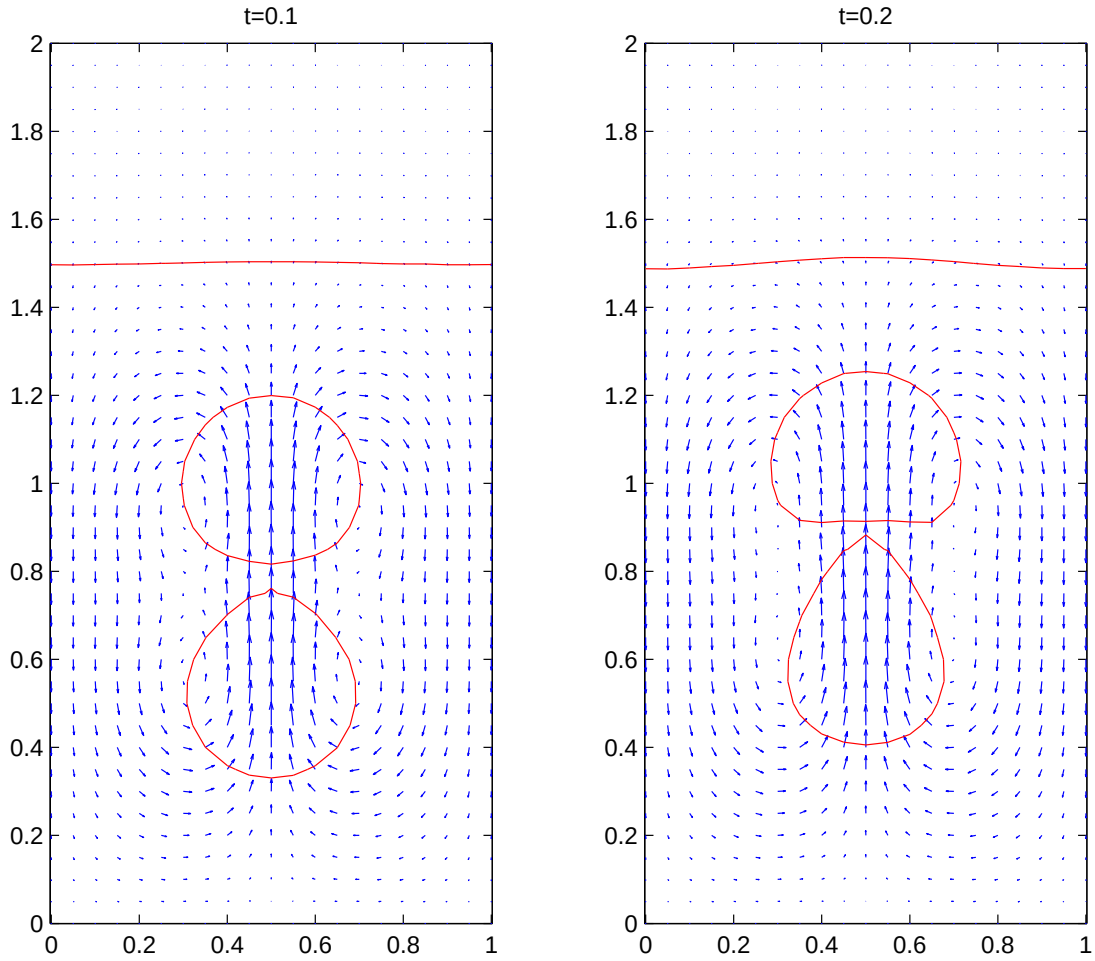

Figure 6.1: Numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 0.1$ and $t = 0.2$. Initially separated from the upper one, the lower bubble moves faster due to the wake formation below the upper one.

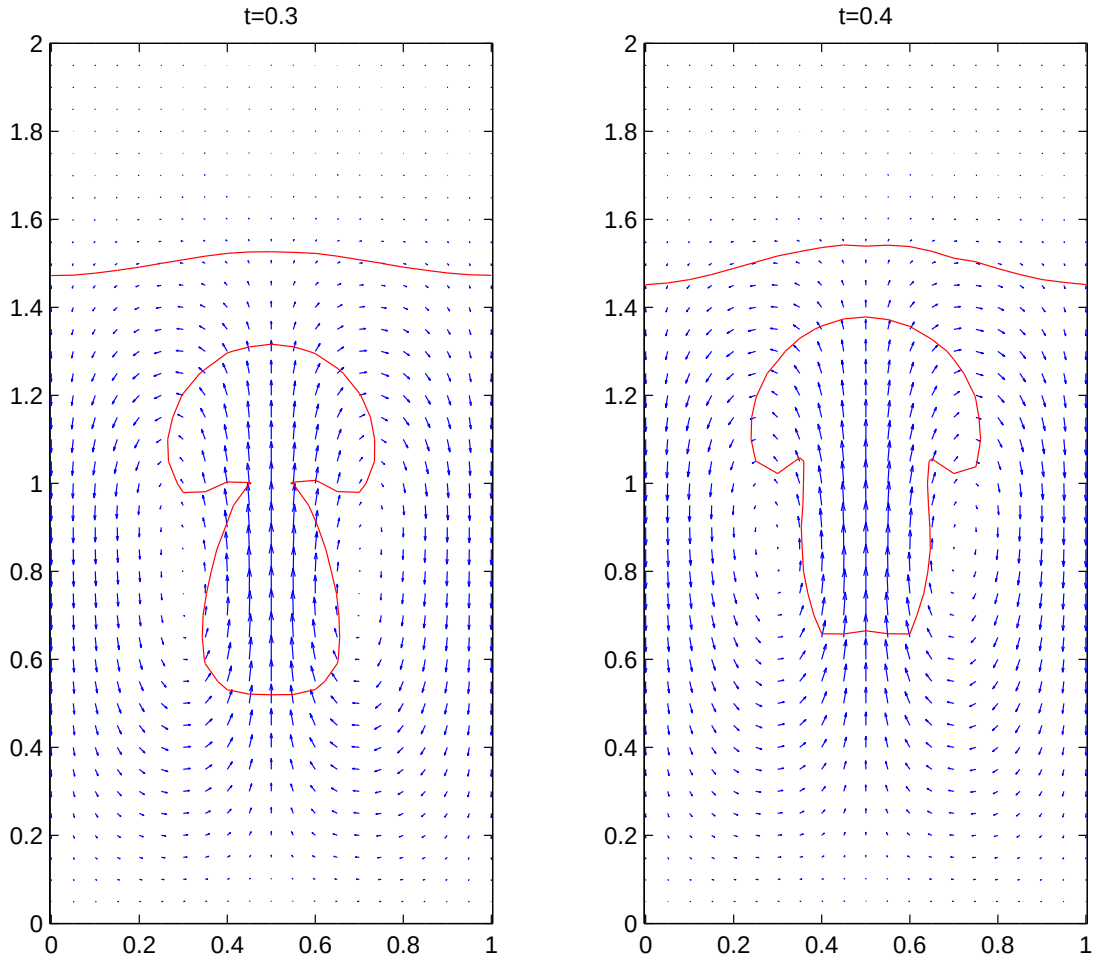

Figure 6.2: Numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 0.3$ and $t = 0.4$. The two bubbles merge together and continue to move upwards.

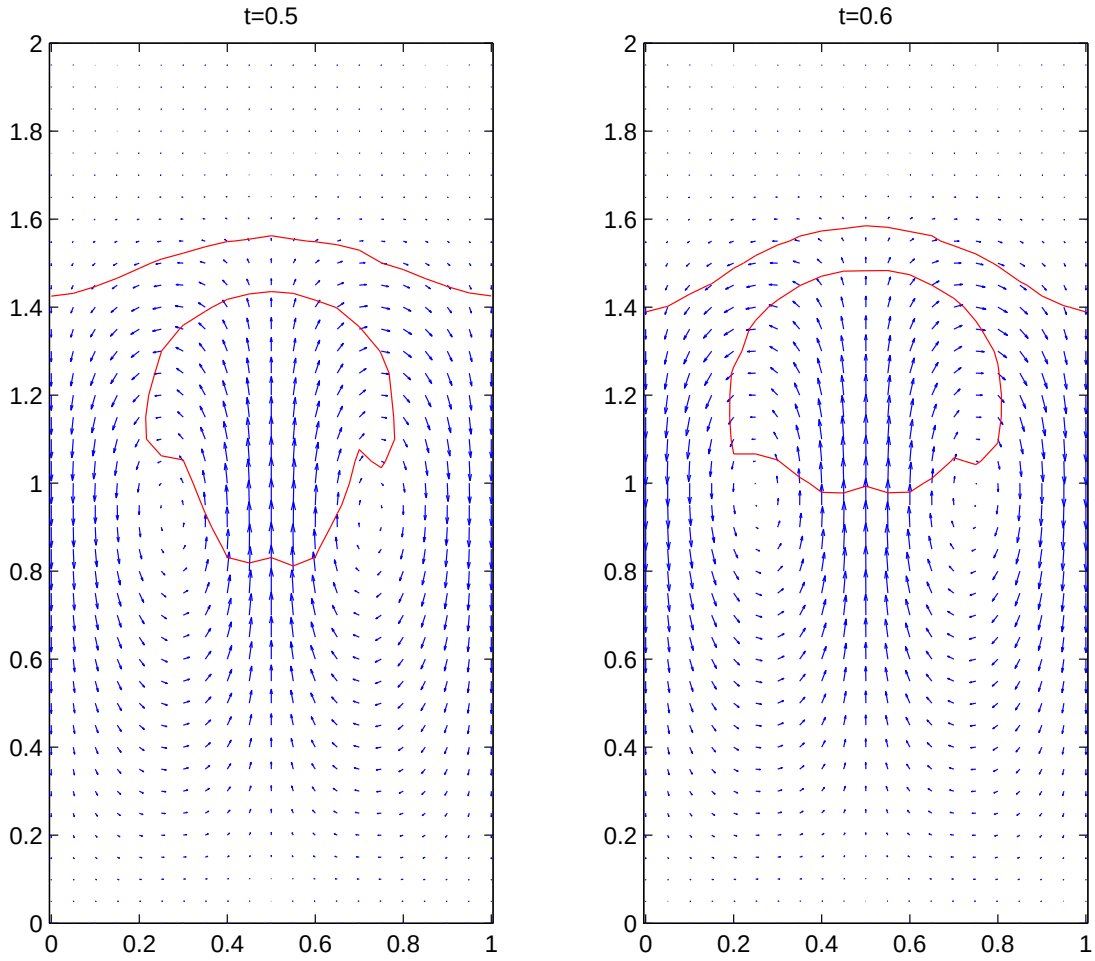

Figure 6.3: Numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 0.5$ and $t = 0.6$. The bubbles reach the free surface on the top part of the cavity. The curvature of the free surface shows the effect of the surface tension in keeping the kinematic equilibrium on the free surface.

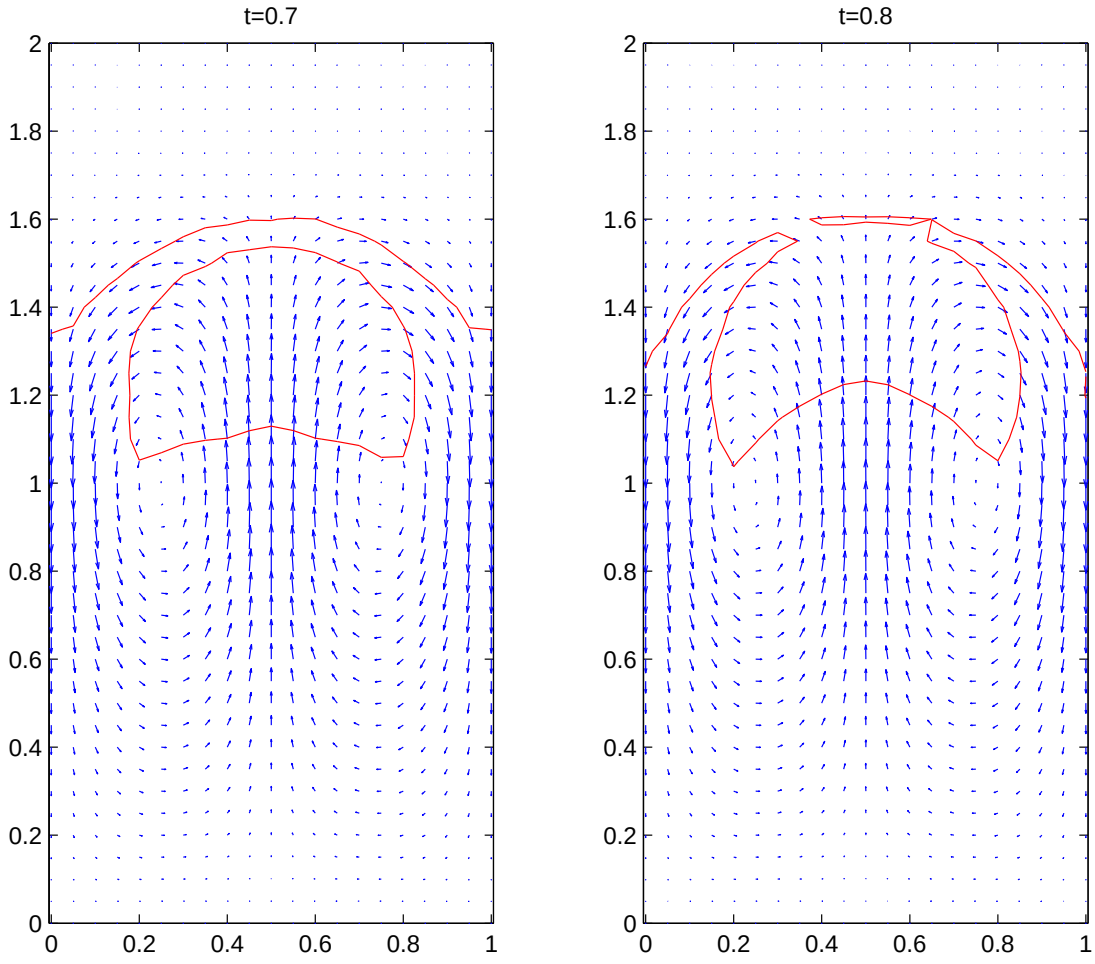

Figure 6.4: Numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 0.7$ and $t = 0.8$. The free surface is finally broken by the moving bubbles. This also causes the bubbles to diffuse themselves into the surrounding fluid.

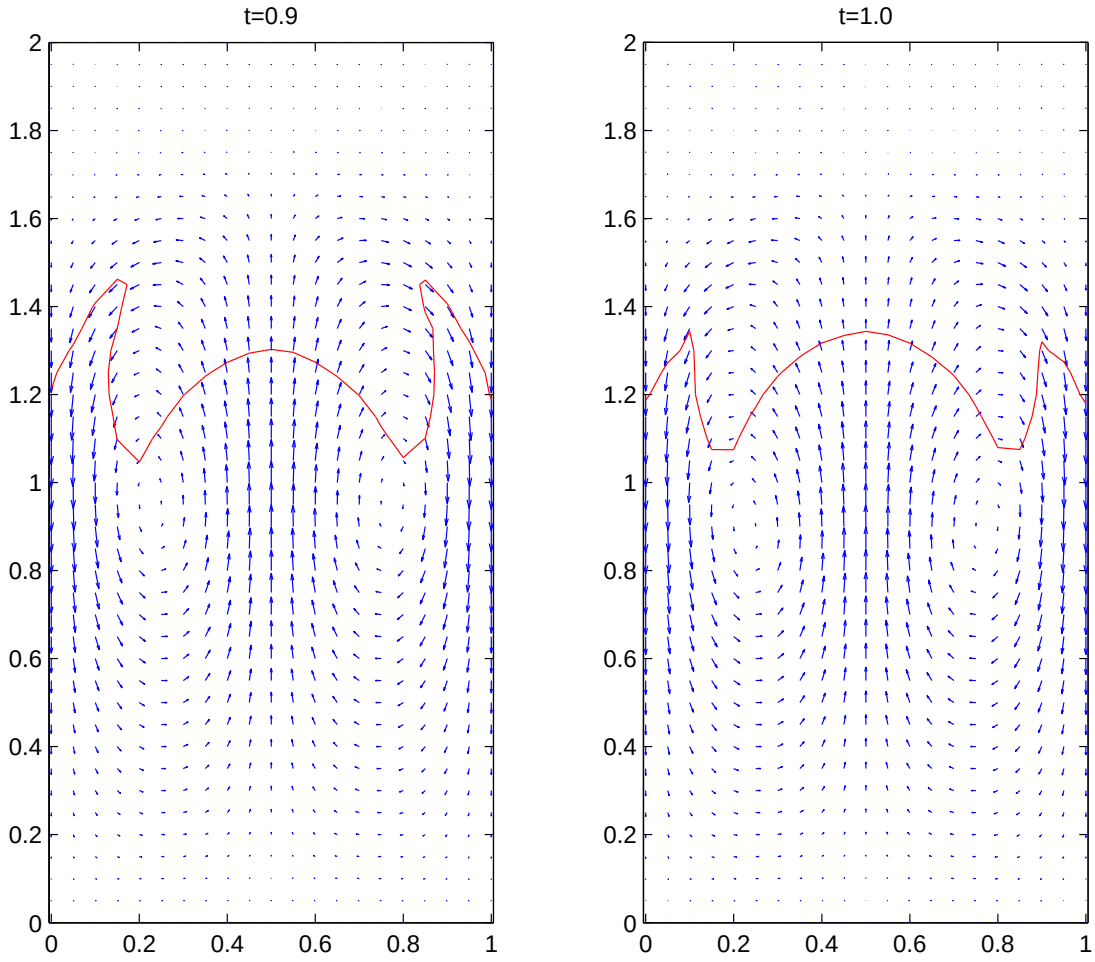

Figure 6.5: Numerical simulation of two bubbles rising up in a buoyancy-driven flow. The shape of the free surface after broken and the velocity field at time $t = 0.9$ and $t = 1.0$.

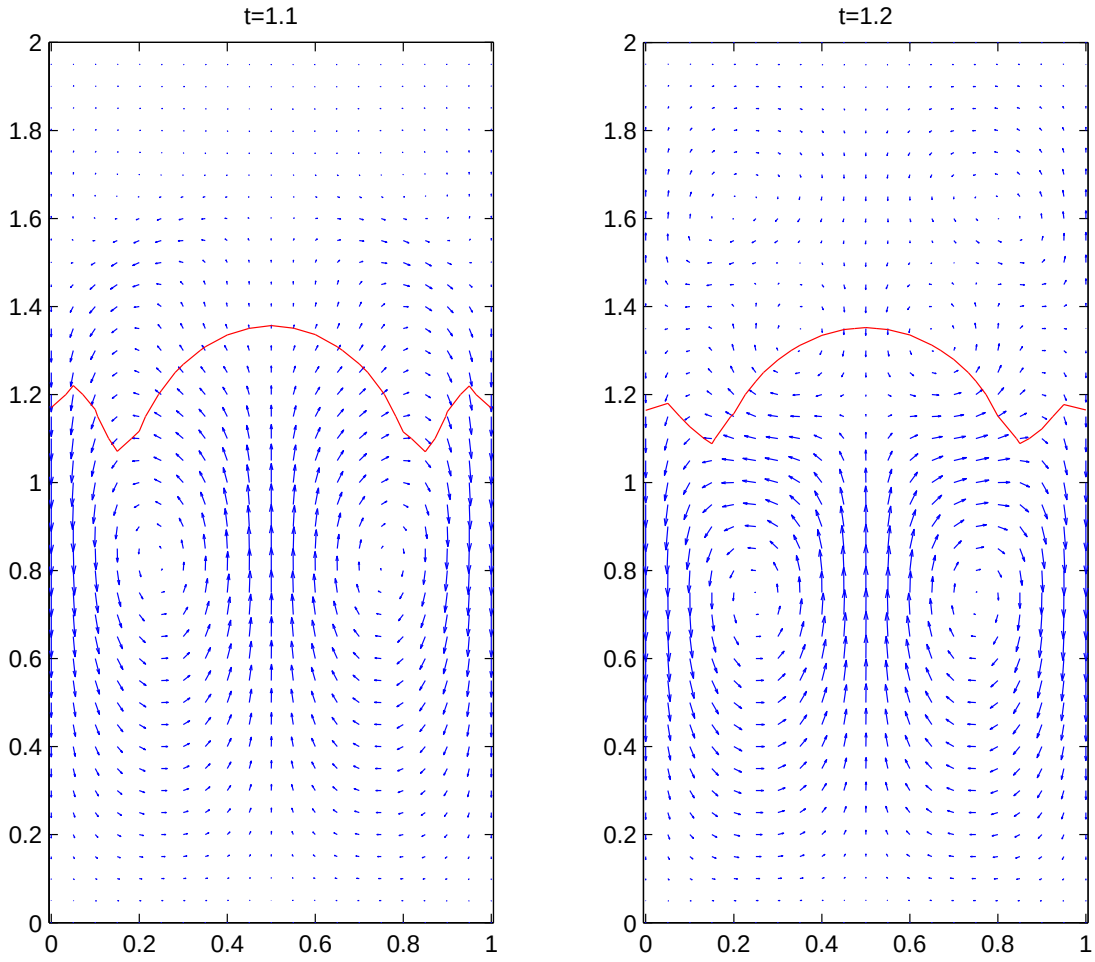

Figure 6.6: Numerical simulation of two bubbles rising up in a buoyancy-driven flow. The shape of the free surface after broken and the velocity field at time $t = 1.1$ and $t = 1.2$.

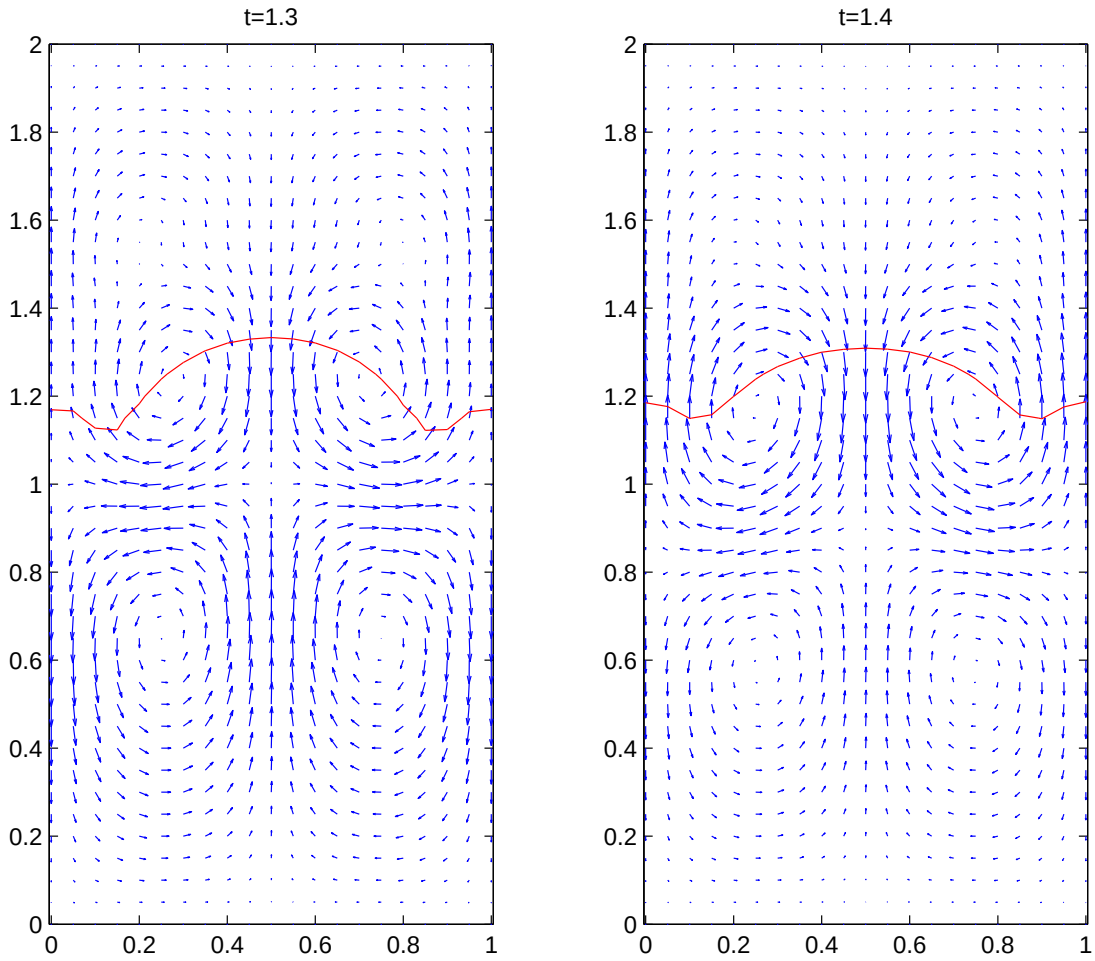

Figure 6.7: Velocity field in the numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 1.3$ and $t = 1.4$. The presence of the vorticities indicates the effect of the surface tension along the free surface on the velocity field even when the bubbles completely diffused into the surrounding fluid.

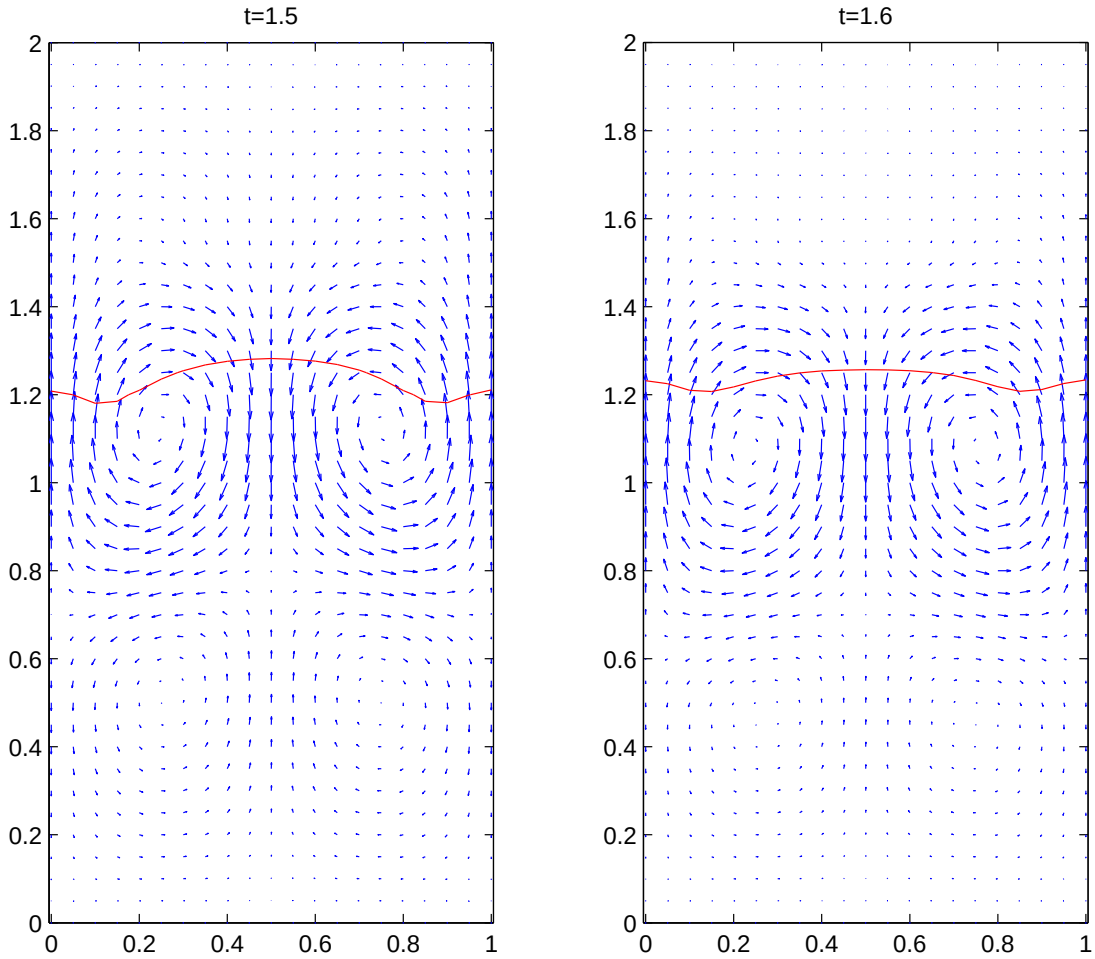

Figure 6.8: Numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 1.5$ and $t = 1.6$. The disappearance of the two upper vortices (see Figures 6.7) corresponds to the decrease in curvature of the free surface.

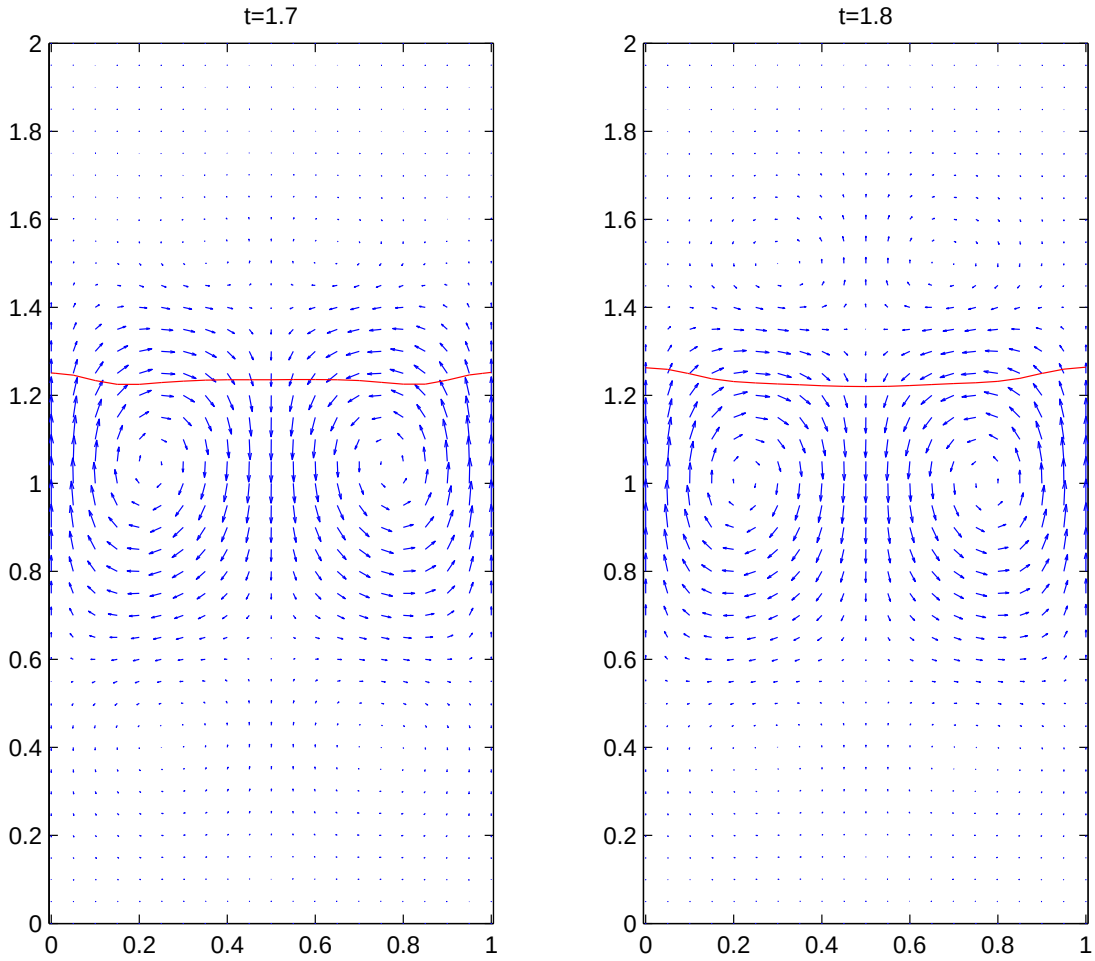

Figure 6.9: Numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 1.7$ and $t = 1.8$. The free surface is on the way to setup a new equilibrium.

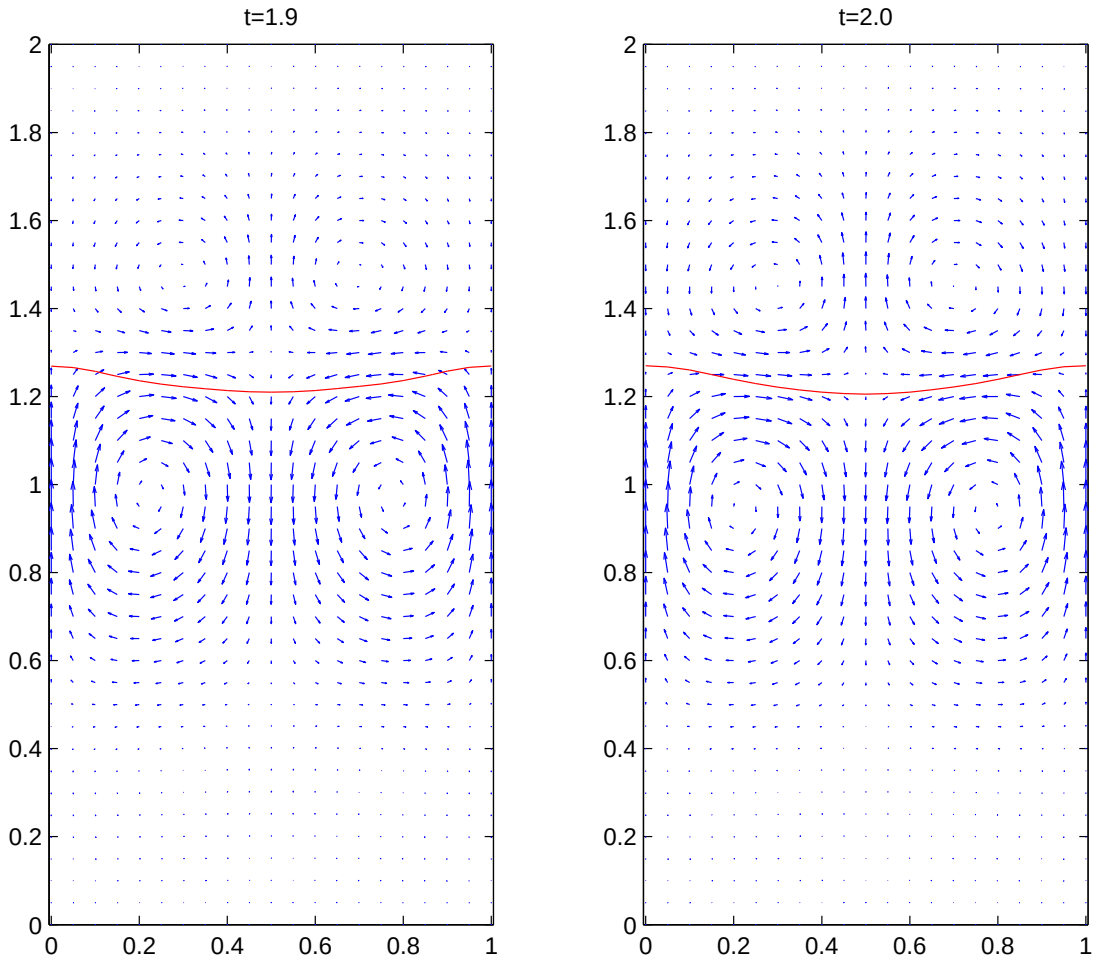

Figure 6.10: Velocity field in the numerical simulation of two bubbles rising up in a buoyancy-driven flow at time $t = 1.9$ and $t = 2.0$. A new equilibrium is about to be set for the free surface.

6.5 Concluding Remarks

A novel meshless approach has been reported in this chapter for the numerical simulation of interfacial flows in which the motion and deformation of the interface as well as the interaction between the moving interface and the surrounding fluid are fully captured. The new approach consists of (a) the flow modelling scheme based on the IPC-IRBFN scheme with modifications; (b) the interface modelling scheme from Chapter 4; and (c) the flow-interface coupling model based on the CSF model proposed by (Brackbill et al., 1992). Bringing those "ingredients" together requires appropriate modifications as well as adaptations so as to suit the numerical simulation of the interfacial flows. All of these aspects have been reported in this chapter.

Regarding the flow modelling scheme, the IPC-IRBFN scheme, which shows its good capability to solve the unsteady incompressible Navier-Stokes equations in Chapter 5, is used with modifications in this chapter for solving two-fluid Navier-Stokes equations with variable density and viscosity. With regards to the interface modelling, the meshless approach to capturing moving interfaces reported in Chapter 4 is used with the application of the level set formulation based on smoothed Heaviside and Dirac delta functions. This helps avoid numerical instabilities in solving the two-fluid Navier-Stokes equations. Finally, the flow-interface coupling model based on the CSF model make the new approach easier to implement thanks to the fact that no explicit description of the moving interfaces is needed to impose the kinematic equilibrium conditions on the interface at each time step.

The new meshless approach has been applied to the numerical simulation of the interfacial flows of two immiscible fluids. The numerical results show that the new approach is capable of capturing primary phenomena of flows such as the deformation and topological change of the moving interfaces as well as the interaction between the interface and the surrounding fluid.

Conclusions and Future Work

The main contributions of the current research include the new meshless numerical schemes for solving time-dependent PDEs as well as for the numerical simulation of some typical unsteady incompressible viscous flows that have been reported throughout this thesis. These contributions are highlighted in this chapter. In addition, recommendations for improvements and extensions to the results of this research are also made.

Two meshless numerical schemes have been reported in Chapter 3 for solving time-dependent PDEs such as parabolic, hyperbolic and convection-diffusion equations. In these meshless schemes, the θ-method or a class of ODE solver is used together with the IRBFN method. Numerical results show that the two proposed schemes are highly accurate and efficient. In addition, extensive experiments on the the two schemes show that the two schemes are not sensitive to the network parameter β for multiquadric basis functions.

A new meshless approach to capturing moving interfaces in passive transport problems has been presented in Chapter 4 in which the motion and deformation with or without topological changes of the moving interfaces are well captured by a unique procedure. In the new approach, the high-order time stepping methods either based on semi-Lagrangian or Taylor series expansions and the level set method are coupled within the meshless framework of the IRBFN method for dealing with the moving interfaces in an accurate, stable and efficient manner. Numerical results show the good capability of the new meshless approach

for passive transport problems.

Two novel meshless schemes, namely IPC-IRBFN and IPCPP-IRBFN, have been reported in Chapter 5 for solving the unsteady incompressible Navier-Stokes equations. The two schemes are based on the projection methods that decouple the treatment of the pressure and velocity field. By decomposing the Navier-Stokes equations into a set of smaller and easier-to-solve subproblems, the two schemes well suit for RBF-based methods whose system matrices are generally dense. Numerical results obtained by applying the new schemes to unsteady Stokes, Navier-Stokes equations with known analytical solutions as well as the lid-driven cavity flow show that the two schemes are highly stable, accurate and suitable for solving unsteady incompressible viscous flows.

Finally, a new meshless approach to the numerical simulation of interfacial flows has been reported in Chapter 6 in which the motion and deformation of the interface between the two immiscible fluids as well as the interaction between the moving interface and the surrounding fluid are fully captured. The new approach brings the flow modelling scheme based on the IPC-IRBFN scheme, the interface modelling scheme based on the level set method, and the flow-interface coupling scheme based on the CSF model together to deal with interfacial flows in an accurate and stable manner. Numerical results show the good capability of the new approach to capture primary phenomena of the interfacial flows of two immiscible fluids.

Throughout this thesis, the performance of the proposed schemes have been numerically verified in various test problems. A rigorous theoretical analysis of the schemes would be beneficial for further improvements of the results from this research. In addition, an implementation of the proposed schemes in this thesis in a parallel computing fashion would be desirable to increase the computational efficiency for large-scale problems. Finally, an extension of the results from this research to unsteady non-Newtonian fluid flows would have to be undertaken so as to be able to simulate non-ideal fluid flows in industries.

Appendix A

The first and second order antiderivatives of Hardy's multiquadrics

This appendix presents the symbolic integrations of 2D MQ-RBF

$$\varphi_i = \sqrt{r_i^2 + s_i^2},$$

where $r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ and s_i is a shape parameter associated with the i^{th} center. Only the antiderivatives are described here.

$$\int \varphi_i \, dx = \frac{1}{2}(x - x_i)\sqrt{r_i^2 + s_i^2} + \frac{1}{2}\left((y - y_i)^2 + s_i^2\right) \ln\left((x - x_i) + \sqrt{r_i^2 + s_i^2}\right),$$

$$\int \varphi_i \, dy = \frac{1}{2}(y - y_i)\sqrt{r_i^2 + s_i^2} + \frac{1}{2}\left((x - x_i)^2 + s_i^2\right) \ln\left((y - y_i) + \sqrt{r_i^2 + s_i^2}\right),$$

$$\iint \varphi_i \, dx \, dx = \frac{1}{6} \left((x - x_i)^2 - 2(y - y_i)^2 - 2s_i^2 \right) \sqrt{r_i^2 + s_i^2} + \frac{1}{2} (x - x_i) \left((y - y_i)^2 + s_i^2 \right) \ln \left((x - x_i) + \sqrt{r_i^2 + s_i^2} \right),$$

$$\iint \varphi_i \, dy \, dy = \frac{1}{6} \left((y - y_i)^2 - 2(x - x_i)^2 - 2s_i^2 \right) \sqrt{r_i^2 + s_i^2} + \frac{1}{2} (y - y_i) \left((x - x_i)^2 + s_i^2 \right) \ln \left((y - y_i) + \sqrt{r_i^2 + s_i^2} \right),$$

$$\begin{aligned} \iint \varphi_i \, dx \, dy = & - \frac{1}{18} (y - y_i) \left((y - y_i)^2 + 6 \, s_i^2 - 6 \, (x - x_i) \sqrt{r_i^2 + s_i^2} \right) \\ & + \frac{1}{3} s_i^3 \left(\tan^{-1} \frac{y - y_i}{s_i} - \tan^{-1} \frac{(x - x_i)(y - y_i)}{s_i \sqrt{r_i^2 + s_i^2}} \right) \\ & + \frac{1}{6} (y - y_i) \left((y - y_i)^2 + 3 \, s_i^2 \right) \ln \left((x - x_i) + \sqrt{r_i^2 + s_i^2} \right) \\ & + \frac{1}{6} (x - x_i) \left((x - x_i)^2 + 3 \, s_i^2 \right) \ln \left((y - y_i) + \sqrt{r_i^2 + s_i^2} \right) \end{aligned}$$

Appendix B

The first and second order antiderivatives of Duchon's thin plate splines

This appendix presents the symbolic integrations of 2-dimensional first order TPS-RBF

$$\varphi_i = r_i^2 \ln(r_i),$$

where $r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$. Only the antiderivatives are described here.

$$\begin{aligned}\int \varphi_i \, dx = &\frac{2}{3}(y - y_i)^3 \tan^{-1}\left(\frac{x - x_i}{y - y_i}\right) - \frac{1}{9}(x - x_i)\left((x - x_i)^2 + 6(y - y_i)^2\right) \\ &+ \frac{1}{3}(x - x_i)\left((x - x_i)^2 + 3(y - y_i)^2\right)\ln(r_i),\end{aligned}$$

$$\begin{aligned}\int \varphi_i \, dy = &\frac{2}{3}(x - x_i)^3 \tan^{-1}\left(\frac{y - y_i}{x - x_i}\right) - \frac{1}{9}(y - y_i)\left(6(x - x_i)^2 + (y - y_i)^2\right) \\ &+ \frac{1}{3}(y - y_i)\left(3(x - x_i)^2 + (y - y_i)^2\right)\ln(r_i),\end{aligned}$$

$$\begin{aligned}\iint \varphi_i \, dx \, dx = &- \frac{1}{144} (x - x_i)^2 \left(7 (x - x_i)^2 + 78 (y - y_i)^2\right) \\ &+ \frac{2}{3} (x - x_i) (y - y_i)^3 \tan^{-1}\left(\frac{x - x_i}{y - y_i}\right) \\ &- \frac{1}{3} (y - y_i)^4 \ln\left(1 + \frac{(x - x_i)^2}{(y - y_i)^2}\right) \\ &+ \frac{1}{12} r_i^2 \left((x - x_i)^2 + 5 (y - y_i)^2\right) \ln(r_i),\end{aligned}$$

$$\begin{aligned}\iint \varphi_i \, dy \, dy = &- \frac{1}{144} (y - y_i)^2 \left(78 (x - x_i)^2 + 7 (y - y_i)^2\right) \\ &+ \frac{2}{3} (x - x_i)^3 (y - y_i) \tan^{-1}\left(\frac{y - y_i}{x - x_i}\right) \\ &- \frac{1}{3} (x - x_i)^4 \ln\left(1 + \frac{(y - y_i)^2}{(x - x_i)^2}\right) \\ &+ \frac{1}{12} r_i^2 \left(5 (x - x_i)^2 + (y - y_i)^2\right) \ln(r_i),\end{aligned}$$

$$\begin{aligned}\iint \varphi_i \, dx \, dy = &\frac{1}{6} (x - x_i)^4 \tan^{-1}\left(\frac{y - y_i}{x - x_i}\right) + \frac{1}{6} (y - y_i)^4 \tan^{-1}\left(\frac{x - x_i}{y - y_i}\right) \\ &+ \frac{1}{18} (x - x_i) (y - y_i) r_i^2 (6 \ln(r_i) - 5) .\end{aligned}$$

References

Allievi, A. and Bermejo, R. (2000). Finite element modified method of characteristics for the Navier-Stokes equations, *International Journal for Numerical Methods in Fluids* **32**: 439–464.

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

Atluri, S. and Zhu, T. (1998b). A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computational modeling and simulation, *Modeling Simulation in Engineering* **3**: 187–196.

Bardi, M. and Osher, S. (1991). The nonconvex multidimensional Riemann problem for Hamilton-Jacobi equations, *SIAM Journal on Mathematical Analysis* **22**: 344–351.

Beatson, R. and Powell, M. (1990). Univariate multiquadric approximation: Quasi-interpolation to scattered data, *Technical Report DAMTP 1990/NA7*, University of Cambridge.

Behrens, J. and Iske, A. (2002). Grid-free adaptive semi-Lagrangian advection using radial basis functions, *Computers and Mathematics with applications* **43**: 319–327.

Bell, J., Colella, P. and Glaz, H. (1989). A second-order projection method for the incompressible Navier-Stokes equations, *Journal of Computational Physics* **85**: 257–283.

Brackbill, J., Kothe, D. and Zemach, C. (1992). A continuum method for modeling surface tension, *Journal of Computational Physics* **100**: 335–354.

Canuto, C., Hussaini, M., Quarteroni, A. and Zang, T. (1988). *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York.

Carlson, R. and Foley, T. (1991). The parameter r^2 in multiquadric interpolation, *Computers and Mathematics with Applications* **21**(9): 29–42.

Chang, Y., Hou, T., Merriman, B. and Osher, S. (1996). A level set formulation of Eulerian interface capturing method for incompressible fluid flows, *Journal of Computational Physics* **124**: 449–464.

Chen, C., Golberg, M. and Rashed, Y. (1998). A meshfree method for linear diffusion equations, *Numerical Heat Transfer, Part B* **33**: 469–486.

Chorin, A. (1968). Numerical solution of the Navier-Stokes equations, *Math. Comp.* **22**: 745–762.

Crandall, M. and Lions, P. (1984). Two approximations of solutions of Hamilton-Jacobi equations, *Math. Comput.* **43**: 1–19.

Cuvelier, C. and Schulkes, R. (1990). Some numerical methods for the computation of capillary free boundaries governed by the Navier-Stokes equations, *SIAM Review* **32**(3): 355–423.

Donea, J. (1984). A Taylor-Galerkin method for convective transport problems, *International Journal for Numerical Methods in Engineering* **20**: 101–119.

Donea, J. and Huerta, A. (2003). *Finite Element Methods for Flow Problems*, John Wiley & Sons Ltd., England.

Fasshauer, G. (1996). Soving partial differential equations by collocation with radial basis functions, *in* A. Le Mehaute, C. Rabut and L. Schumaker (eds), *Proceedings of Chamonix*, Vanderbilt University Press, Nashville, pp. 1–8.

Floryan, J. and Rasmussen, H. (1989). Numerical methods for viscous flows with moving boundary, *Applied Mechanics Reviews* **42**(12): 323–341.

Franke, R. (1982). Scattered data interpolation: Test of some methods, *Mathematics of Computation* **38**(157): 181–200.

Girault, V. and Raviart, P. (1986). *Finite Element Methods for Navier-Stokes Equations. Theory and Algorithms*, Springer-Verlag, Berlin.

Goda, K. (1979). A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows, *Journal of Computational Physics* **30**: 76–95.

Golberg, M. and Chen, C. (1999). *The Method of Fundamental Solutions for Potential, Helmholtz and Diffusion Problems*, WIT Press and Computational Mechanics Publications, Boston, Southampton, chapter 4, Boundary Integral Methods: Numerical and Mathematical Aspects, pp. 105–176.

Golberg, M. and Chen, C. (2001). An efficient mesh-free method for nonlinear reaction-diffusion equations, *Computer Modeling in Engineering and Sciences* **2**(1): 87–95.

Gresho, P. (1990). On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory, *International Journal for Numerical Methods in Fluids* **11**(5): 587–620.

Gresho, P. and Sani, R. (2000). *Incompresible Flow and the Finite Element Method; Vol. 1: Advection-Diffusion. Vol. 2: Isothermal Laminar Flow*, John Wiley & Sons, Chichester.

Hirt, C., Amsden, A. and Cook, J. (1974). An arbitrary Lagrangian-Eulerian computing method for all speeds, *Journal of Computational Physics* **14**: 227–253.

Hirt, C. and Nichols, B. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics* **39**(1): 201 –225.

Hon, Y. (2002). A quasi-radial basis functions method for American options pricing, *Computer and Mathematics with Applications* **43**(3/5): 513–524.

Hon, Y., Cheung, K., Mao, X. and Kansa, E. (1999). Multiquadric solution for shallow water equations, *ASCE Journal of Hydraulic Engineering* **125**(5): 524–533.

Iafrati, A., Mascio, A. and Campana, E. (2001). A level set technique applied to unsteady free surface flows, *International Journal for Numerical Methods in Fluids* **35**: 281–297.

Ingber, M. and Phan-Thien, N. (1992). A boundary element approach for parabolic differential equations using a class of particular solutions, *Applied Mathematical Modelling* **16**: 124–132.

Jacqmin, D. (1999). Calculation of two-phase Navier-Stokes flows using phase-field modelling, *Journal of Computational Physics* **155**: 96–127.

Kansa, E. (1990a). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics–I. Surface approximations and partial derivative estimates, *Computers and Mathematics with Applications* **19**(8/9): 127–145.

Kansa, E. (1990b). 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 Applications* **19**(8/9): 147–161.

Karniadakis, G., Israeli, M. and Orszag, S. (1991). High-order splitting methods for the incompressible Navier-Stokes equations, *Journal of Computational Physics* **97**: 414–443.

Karniadakis, G. and Sherwin, S. (1999). *Spectral /Hp Element Methods for CFD*, Oxford University Press, New York.

Kim, J. and Moin, P. (1985). Application of a fractional step method to incompressible Navier-Stokes equations, *Journal of Computational Physics* **59**: 308–323.

Knabner, P. and Angermann, L. (2003). *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*, Springer-Verlag, New York.

Kovacevic, I., Poredos, A. and Sarler, B. (2003). Solving the Stefan problem with the radial basis function collocation method, *Numerical Heat Transfer, B Fundamentals* **44**: 1–24.

LeVeque, R. (2002). *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge.

Mai-Cao, L. and Tran-Cong, T. (2005). A meshless IRBFN-based method for transient problems, *Computer Modeling in Engineering & Sciences* **7**(2): 149–171.

Mai-Duy, N. (2001). *Mesh-Free Radial Basic Function Networks Methods for Some Problems in Continuum Mechanics*, PhD thesis, University of Southern Queensland, Australia.

Mai-Duy, N., Mai-Cao, L. and Tran-Cong, T. (2006). *A New Meshless RBF-Based Method for Unsteady Fluid Flow Analysis*, Tech Science Press, chapter XI, pp. 241–262.

Mai-Duy, N. and Tran-Cong, T. (2001a). Numerical solution of differential equations using multiquadric radial basis function networks, *Neural Networks* **14**: 185–199.

Mai-Duy, N. and Tran-Cong, T. (2001b). Numerical solution of navier-stokes

equations using multiquadric radial basis function networks, *International Journal for Numerical Methods in Fluids* **37**: 65–86.

Mai-Duy, N. and Tran-Cong, T. (2003). Approximation of function and its derivatives using radial basis function networks, *Applied Mathematical Modelling* **27**: 197–220.

Marion, M. and Temam, R. (1998). *Navier-Stokes Equations: Theory and Approximation*, Vol. VI of *Handbook of Numerical Analysis*, North-Holland, Amsterdam, pp. 503–688.

Moody, J. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units, *Neural Computations* **1**: 281–294.

Moridis, G. and Kansa, E. (1994). The Laplace transform multiquadradric method: A highly accurate scheme for the numerical solution of linear partial differential equations, *Journal of Applied Science and Computations* **1**(2): 375–407.

Oliveira, A. and Baptista, A. (1995). A comparison of integration and interpolation Eulerian-Lagrangian methods, *International Journal for Numerical Methods in Fluids* **21**: 183–204.

Orszag, S., Israeli, M. and Deville, M. (1986). Boundary conditions for incompressible flows, *Journal of Scientific Computing* **1**: 75–111.

Osher, S. and Fedkiw, R. (2003). *Level Set Methods and Dynamic Implicit Surfaces*, Vol. 153 of *Applied Mathematical Sciences*, Springer, New York.

Osher, S. and Sethian, J. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics* **79**: 12–49.

Patankar, S. (1980). *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishihg Corporation, USA.

Peng, D., Merriman, B., Osher, S., Zhao, H. and Kang, M. (1999). A PDE-based fast local level set method, *Journal of Computational Physics* **155**: 410–438.

Perko, J., Chen, C. and Sarler, B. (2001). A polygon-free numerical solution of steady natural convection in solid-liquid systems, *in* B. Sarler and C. Brebbia (eds), *Moving Boundaries VI: Computational Modelling of Free and Moving Boundary Problems, (Computational and Experimental Methods, Vol. 4)*, WIT Press, Boston, pp. 111–122.

Peyret, R. (2002). *Spectral Methods for Incompressible Viscous Flow*, Applied Mathematical Sciences 148, Springer-Verlag, New York.

Peyret, R. and Taylor, T. (1983). *Computational Methods for Fluid Flow*, Springer-Verlag, New York.

Quarteroni, A., Sacco, R. and Saleri, F. (2000). *Numerical Mathematics*, Vol. 37 of *Texts in Applied Mathematics*, Springer-Verlag, New York, USA.

Quarteroni, A. and Valli, A. (1997). *Numerical Approximation of Partial Differential Equations*, Springer-Verlag, New York.

Rannacher, R. (1992). *On Chorin's Projection Method for the Incompressible Navier-Stokes Equations*, Vol. 1530 of *Lecture Notes in Mathematics*, Springer Berlin, Heidelberg, pp. 167–183.

Richardson, S. (1989). *Fluid Mechanics*, Hemisphere Publishing Corp., New York.

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

Roache, P. (1998). *Fundamentals of Computational Fluid Dynamics*, Hermosa, USA.

Saad, Y. and Schultz, M. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific and Statistical Computing* **7**: 856–869.

Sarler, B., Perko, J. and Chen, C. (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.

Sarler, B., Perko, J., Chen, C. and Kuhn, G. (2001). A meshless approach to natural convection, *in* S. Atluri, T. Nishioka and M. Kikuchi (eds), *Advances in Computational Engineering & Sciences, ICES'01: [Proceedings of the International Conference on Computational Engineering & Sciences, 19-25 August 2001, Puerto Vallarta, Mexico]*, Tech Science Press, Palmdale, CA.

Sethian, J. (1999). *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, New York.

Shu, C. and Osher, S. (1989). Efficient implementation of essentially non-oscillatory shock capturing schemes II, *Journal of Computational Physics* **83**: 32–78.

Staniforth, A. and Cote, J. (1991). Semi-lagrangian integration schemes for atmospheric models - A review, *Monthly Weather Review* **119**: 2206–2223.

Sussman, M. and Fatemi, E. (1999). An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow., *SIAM Journal on Scientific Computing* **20**(4): 1165–1191.

Sussman, M. and Smereka, P. (1997). Axisymmetric free boundary problems, *Journal of Fluid Mechanics* **341**: 269–294.

Sussman, M., Smereka, P. and Osher, S. (1994). A level set approach for computing solutions to incompressible two-phase flow, *Journal of Computational Physics* **114**: 146–159.

Tannehill, J., Anderson, D. and Pletcher, R. (1997). *Computational Fluid Mechanics and Heat Transfer*, Taylor & Francis, USA.

Temam, R. (2001). *Navier-Stokes Equations : Theory and Numerical Analysis*, AMS Chelsea Publishing, Providence, RI.

Temperton, C. and Staniforth, A. (1987). An efficient two-time level semi-Lagrangian semi-implicit integration scheme, *Quart. J. Roy. Meteor. Soc.* **113**: 1025–1039.

Timmermans, L., Minev, P. and Van De Vosse, F. (1996). An approximate projection scheme for incompressible flow using spectral elements, *International Journal for Numerical Methods in Fluids* **22**: 673–688.

Tornberg, A. (2000). *Interface Tracking Methods with Application to Multiphase Flows*, PhD thesis, Royal Institute of Technology, Stockholm.

Unverdi, S. and Tryggvason, G. (1992). A front-tracking method for viscous, incompressible, multi-fluid flows, *Journal of computational physics* **100**: 25–37.

Van Kan, J. (1986). A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM Journal on Scientific and Statistical Computing* **7**(3): 870–891.

Zerroukat, M., Djidjeli, K. and Charafi, A. (2000). Explicit and implicit meshless methods for linear advection-diffusion-type partial differential equations, *International Journal for Numerical Methods in Engineering* **48**: 19–35.

Zerroukat, M., Power, H. and Chen, C. (1998). A numerical method for heat transfer problems using collocation and radial basis functions, *International Journal for Numerical Methods in Engineering* pp. 1263–1278.

Zienkiewicz, O. and Taylor, R. (2000). *The Finite Element Method - Volum 3: Fluid Dynamics*, Butterworth-Heinemann, Oxford, UK.