2011

A New Cell-Centred Finite Difference Scheme for CFD Simulations

Ali Salih
University of Windsor

A NEW CELL-CENTRED FINITE DIFFERENCE SCHEME FOR CFD SIMULATIONS

by

Ali Salih

A Thesis
Submitted to the Faculty of Graduate Studies
through Mechanical, Automotive and Materials Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

2011

A New Cell-Centred Finite Difference Scheme for CFD Simulations

by

Ali Salih

APPROVED BY:

_____

Dr. Tirupati Bolisetti
Department of Civil and Environmental Engineering

_____

Dr. Biao (Bill) Zhou
Department of Mechanical, Automotive and Materials Engineering

_____

Dr. Ronald Barron, Advisor
Department of Mathematics and Statistics, and
Department of Mechanical, Automotive and Materials Engineering

_____

Dr. Bruce Minaker, Chair of Defense
Department of Mechanical, Automotive and Materials Engineering

June 20, 2011

AUTHOR'S DECLARATION OF ORIGINALITY

I, Ali Salih, hereby certify that I am the sole author of this thesis. This thesis includes original research that has been previously published/submitted for publication in peer reviewed conference proceedings, as follows,

Thesis Chapter	Publication title/full citation	Publication status
Chapter 3 & 4	Salih, A. and Barron, R.M. *A New Cell-Centred Finite Difference Scheme for CFD Simulations.* 19th Annual Conf. of the CFD Society of Canada, Montreal, April 2011.	*Presented*
Chapter 3 & 4	Salih, A. and Barron, R.M. *A Cell-Based Finite Difference Method for the Numerical Solution of PDEs.* Applied Mathematics, Modeling and Computational Science Conference, Waterloo, July 2011.	*Accepted*

I certify that the material contained in this thesis describes work completed during my registration as a graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, I certify that I have not included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act.

Provisional Patent Application 61/457,589 has been filed on April 22, 2011 for Patent Protection for: "C**ell-Centered Finite Difference Method for Numerical Solution of Partial Differential Equations on Arbitrary Mesh Systems"**

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted in any form for another degree or diploma to any other University or Institution.

ABSTRACT

The governing partial differential equations of fluid motion are usually numerically approximated using one of the three classical methods: Finite Difference (FD), Finite Volume (FV) or Finite Element (FE). In this thesis, a new cell-centred FD (CCFD) formulation is developed, in which the governing fluid flow equations are differenced over all the cell centres instead of grid points. The nodes (grid points) are then updated by averaging the property from all the cell centres that share that node. This feature, which is motivated by development of the FV method, allows the application of the proposed FD numerical formulation on unstructured grids. Several test cases are investigated here to illustrate this approach. To verify the results, the analytical solution for the test case is used if available. Otherwise, the numerical solution is compared to the traditional FD solution.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I owe my deepest gratitude to my supervisor, Dr. Ronald Barron, whose guidance, support and patience from the start to the final level of this project made this thesis possible.

Lastly, my regards to my colleagues for their interest and advice.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ω	relaxation factor (Ch. III)
ω	vorticity (Ch. V)
ψ	streamfunction
Δx	horizontal spacing
Δy	vertical spacing
$\emptyset$	property (e.g. temperature)
∂	partial derivative
P	point or node
opt	optimum

Superscripts

k	previous value
$k+1$	current value

Subscripts

n	north intersection
s	south intersection
e	east intersection
w	west intersection
cc	cell centre
T	top boundary
B	bottom boundary
R	right boundary
L	left boundary

CHAPTER I

INTRODUCTION

Finding practical solutions to the governing equations of fluid mechanics is one of the most challenging problems in engineering. These equations, in most cases, form a set of coupled non-linear partial differential equations (PDEs). In Computational Fluid Dynamics (CFD), the equations of fluid motion are usually approximated by algebraic expressions using one of several well-established numerical techniques. Currently, the most popular discretization methodologies in CFD are the Finite Difference (FD), Finite Volume (FV) and Finite Element (FE) methods.

The CFD field was dominated by the FD method in its early years. The underlying mathematics that forms the foundation of the FD method is relatively simple. This allowed researchers to carry out thorough analyses, such as stability and convergence studies, of the algorithms they were developing. Finite difference methods were initially developed for rectangular domains, and CFD researchers found themselves restricted to flow problems that could be approximated and modeled accordingly. For example, full-potential transonic flow over an airfoil could not be solved because the flow domain is non-rectangular. However, under the assumption of a thin airfoil which creates only small disturbances to the uniform free stream flow, researchers formulated the so-called transonic small-disturbance theory [1] and were able to obtain good numerical solutions that assisted engineers to design efficient airfoils for wing cross-sections. During the 1970's, several researchers developed very sophisticated numerical grid generation techniques which served as an important enabling technology for CFD

simulations [2-4]. However, research is about pushing the envelop, and CFD researchers soon began to place higher demands for the applications of their computer codes, in terms of both the physics that could be modeled and the geometrical complexity of the flow region. In particular, implementation of the FD formulation is restricted to structured grid systems, i.e. those grid systems that can be organized in such a way that an underlying logical connection between nodes can be defined. This is a serious limitation in modern-day CFD simulations, since most industrial applications of CFD involve highly complicated geometric structures and passages which cannot be easily or accurately represented by a structured grid system. Although multiblock methods have been developed to alleviate this problem with the application of the FD method, some issues still persist, such as complicated computer coding, loss of accuracy across block interfaces and the inability to design a highly automated numerical grid generation process.

In view of the limitation of the FD method to structured grids, extensive research on building new Finite Volume (FV) formulations was started in the late 1970's. Researchers in solid mechanics had been using the Finite Element (FE) method for many years and, in the course of their work, had developed fairly sophisticated mesh generation techniques. These mesh systems are referred to as unstructured since there is no logical connection between nodes in the mesh. Although more difficult to manage in terms of computer logic and storage requirements, unstructured meshes are very popular because of their capability to accurately represent highly complex domains. Furthermore, unstructured mesh generation has become highly automated, requiring much less human-computer interaction than structured grid generation. Primarily due to these features, the

FV approach, which can be implemented on either structured or unstructured meshes, has now become central to most of today's commercial CFD codes, such as CFX/ANSYS, FLUENT and STARCD, as well as many research codes. However, the FV formulation still faces many difficulties, such as those associated with grid arrangement [5]. The pressure-velocity coupling and the correct place to store their values (cell face or cell centre or a mix of both) constitutes a problem that continues to attract research in FV [6]. Also, high order FV methods cannot easily be formulated or implemented.

A new FD approach is presented in this thesis, which has the flexibility to be applied on an unstructured mesh. The governing equations are differenced over all the cell centres (control volume centres) instead of the grid points as would typically be done in a conventional FD approach, and hence the name Cell-Centred Finite Difference (CCFD). The nodes are then updated by averaging the property from all the cell centres that share that node.

Several simple test cases are investigated in this thesis to illustrate the development and application of this new method. The numerical results are compared with analytical solutions if available and/or traditional FD solutions. Even in many simple instances the analytical solution is not available, perhaps being restricted due to the boundary condition type. For example, an elliptic equation subject to Dirichlet boundary conditions (i.e. boundary values are specified) may have an analytical solution. However, the same equation subject to a Neumann boundary condition (i.e. normal derivative of the variable is specified) may not have an analytical solution. At the same time, the traditional FD solution may be cumbersome when the mesh topology is not a uniform distribution of the nodes with uniform spacings. In this case, a transformation is

needed to map the clustered grid points in the physical domain to a set of equally spaced grid points in the computational domain.

When the solution domain can be discretized to a structured grid with uniform spacing, as shown in Fig. 1.1a, traditional FD is more efficient than either FV or FE, i.e. it is more stable and needs less resources. However, for curvilinear coordinates or unequally spaced grid lines, as in Fig. 1.1b, the physical domain must be transformed to a computational domain, where the PDEs are solved after also being transformed. In the case of complex geometries and when hanging nodes exist (e.g. interior nodes that have three cells around it instead of four), as illustrated in Fig. 1.1c, the traditional FD method must be designed as a multiblock scheme. The traditional FD method cannot handle a mesh topology such as the one shown in Fig. 1.1d, while FV and FE have the ability to handle all the mesh topologies shown in Fig. 1.1.

The overall objective of this thesis research is to develop a finite difference based scheme for solving partial differential equations which can be implemented on both structured and unstructured mesh systems. The CCFD method developed in this research is designed to be applicable to any physical problem that can be mathematically modeled by PDEs with associated initial and/or boundary conditions. At each step in the development, a simple question is asked: “Does this assumption, or derivation, or decision restrict the general applicability of the method?”. Although this thesis concentrates on the development of the CCFD method for 2-dimensional problems, it is anticipated that, in future research, this method will be extended to 3-dimensional simulations.

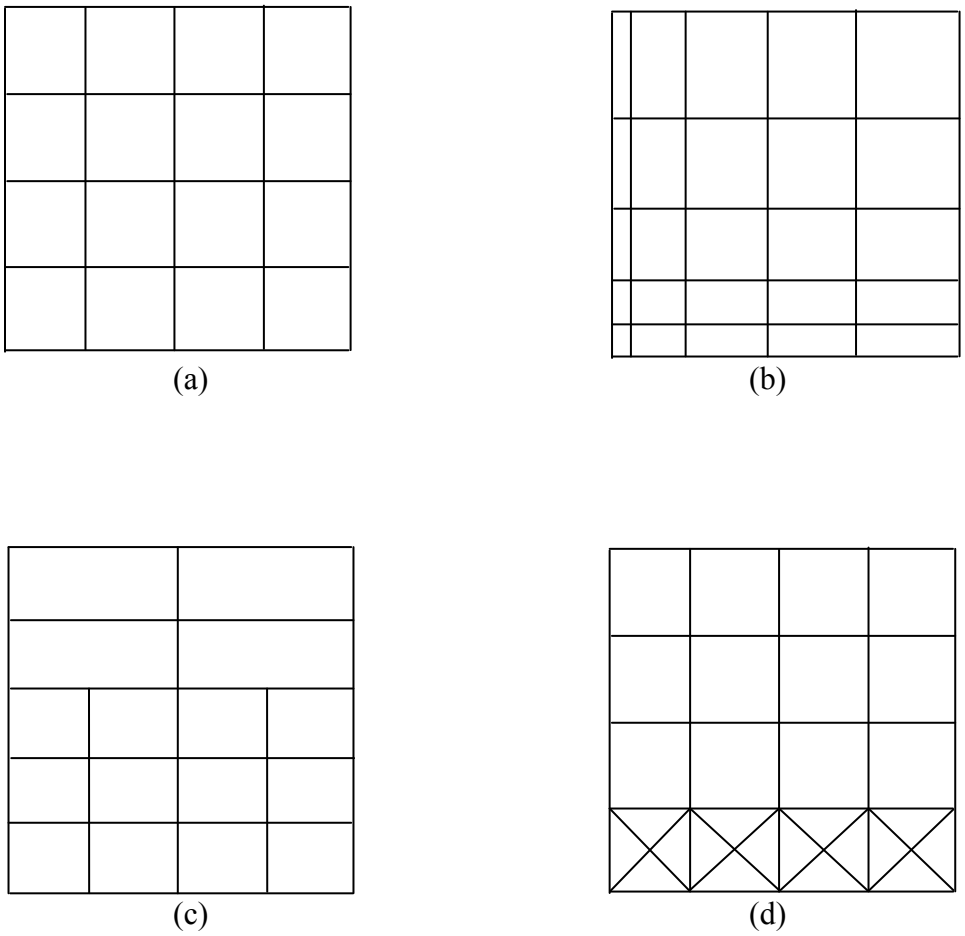


Figure 1.1: Different types of mesh topologies; (a) structured uniform mesh, (b) structured clustered mesh, (c) unstructured mesh with hanging nodes, (d) hybrid mesh

CHAPTER II

REVIEW OF LITERATURE

2.1 Introduction

There is a vast body of literature that discusses issues regarding the numerical solution of partial differential equations. Likewise, the literature on the numerical simulation of fluid flows is extensive. Although there has been much research in these fields, most of it does not directly impact on the research presented in this thesis. Rather, some fundamental ideas developed over many decades of research have informed the formulation of the Cell-Centred Finite Difference (CCFD) method. In this chapter, some specific literature that is relevant to the research in this thesis is reviewed.

2.2 Classical Numerical Schemes in CFD

Peiro and Sherwin [7] have presented the fundamental concepts of the FD, FV and FE methods. They point out that the integral formulation (i.e. FV and FE formulations) of the governing equations is more advantageous since it handles Neumann boundary conditions and discontinuous source terms in a more natural way. Moreover, the integral form deals with complex geometries better than the differential form (i.e. FD) as it doesn't rely on the type of the mesh, i.e. the mesh can be structured, unstructured or hybrid. The rules for assignment of weighting functions and the similarity between FV and FE discretization strategies are explained by Mattiussi [8]. Onata and Idelsohn [9] showed that a system of equations equivalent to FV can be derived from FE by setting weighting functions equal to unity inside the cell and zero outside the cell.

The fundamental concepts of FV and details of FD formulas are explained by Hoffmann and Chiang [10, 11], Smith [12], and in many other CFD textbooks. The main differences between the three classical numerical schemes include the following points. FD and FV generate the discretized equation at any node based on the corresponding values at neighbouring nodes. In the FE method, the discretized equation of each element is independent of the other elements. In FE, incorporating different types of boundaries (i.e. Dirichlet and Neumann) is simpler, because it will only effect the element equations. On the other hand, FD and FV formulations need to be modified for derivative boundaries. In FD and FV the coupling between the discretized equation setup and its solution is based on the cell type, while in FE, adding new cell types will only change the local cell equations and the final solution procedure through the global matrices doesn't need to be modified. In the CCFD formulation developed in this thesis, each cell is considered in much the same fashion as in the FE method. The effect of cell type and its geometrical details is translated into constant coefficients in the FD equation for that individual cell. However, unlike FE, in CCFD the global matrices do not need to be assembled, and the cell-centre values of adjacent cells are linked through the updating procedure of the physical mesh nodes. This part of the procedure is closer to techniques commonly used in FV formulations.

Gottlieb and Orzag [13] and Saleh [14] explain another method to solve differential equations through approximating the unknown functions by using Fourier series or a series of Chebyshev polynomials, which is known as the Spectral Method. Other methods, referred to as meshless methods, have also been in development over the past few years [15]. The main difference between the spectral method and the FD, FV

and FE methods is that the series is valid throughout the domain, whereas the solution is local in the mesh-based methods. Similarly, as implied by the name, meshless methods do not require a grid on which to discretize the governing equations.

2.3 Locating Variables and Related Issues in FV Method

As described by Patankar [16], who is regarded as one of the primary authors of the Finite Volume method, if pressure and velocity components of an incompressible fluid flow are stored in the same location (i.e. non-staggered or collocated grid), an oscillatory or "checkerboard" pressure field may occur. To avoid this problem, researchers developed the idea of a staggered grid, in which each of the flow variables is calculated and stored at a different node in the mesh [16]. In this type of grid system, the pressure-velocity coupling is usually handled through some variation of the SIMPLE algorithm [16], or by formulating a Poisson equation for pressure that simultaneously ensures that mass is conserved [10]. Generally speaking, staggered grids will be structured, so it is also possible to employ this type of grid system in a FD formulation. In a recent research conducted by Barron and Zogheib [17], a new numerical algorithm was developed for solving 2D incompressible flow equations on a staggered curvilinear grid by replacing some weaknesses in FD with equivalent strong parts of FV, especially the velocity-pressure coupling. One of the significant advantages of the FD approach in [17] is that it avoids the need to calculate fluxes at the cell faces, which is central to all FV formulations.

On the other hand, the non-staggered grid has the advantages of easy programming and easier incorporation of boundary conditions [18]. To achieve the

advantages associated with calculating all variables at the same location, researchers have devised schemes to solve the resulting non-realistic pressure fields. Rhie and Chow [19] performed a numerical study on 2D, incompressible, steady flow. They developed a "momentum interpolation" scheme on an ordinary (non-staggered) grid to suppress the pressure oscillations by adding a pressure gradient term into the pressure interpolation. Reggio and Camarero [20] developed a scheme that employs upwind differencing for the velocity gradients and downwind differencing for the pressure gradients to prevent the odd-even pressure field. In another research, Thiart [21] presented a differencing scheme that accounts for the effect of pressure on the velocity gradients. Peric et al. [5] and Demirdzic et al. [22] did a comparison between a staggered and a collocated grid arrangement for three test cases, and they indicated the advantages of a collocated scheme in terms of convergence speed and the ability to extend the implementation to non-orthogonal grids. Versteeg and Malalasekera [23] have also explained the issues related with the FV formulation on a curvilinear body-fitted structured grid, and the need for special correction expressions for the diffusion and convection fluxes in an unstructured grid. They have presented a special pressure-velocity coupling on a collocated grid which avoids the checkerboard pressure effect. They emphasize on the pressure interpolation practice developed in [19] and its success in collocated curvilinear body-fitted and unstructured grids. Majumdar [24] demonstrated the dependency of the momentum interpolation scheme on the under-relaxation factor, and proposed a new scheme for calculating the velocity value at cell faces which preserves a convergent solution, independent of the under-relaxation factor and without significant increases in computational time.

To maintain the advantageous features of a collocated grid, in this thesis a FD scheme is developed that locates or stores all the variables at the same location, the cell centre. A combination of the non-staggered FV philosophy with robust FD approximations is observed, and since it is a purely FD formulation, there is no need for approximating the fluxes across cell faces.

Other endeavors to develop a FD formulation similar to the CCFD formulation, at least in terms of confining the flow variables to the cell centre, can be seen in the MHD (Magneto-Hydro-Dynamics) field, as presented by Livne et al. [25], Stone and Mihalas [26], Gardiner and Stone [27], Mignone et al. [28] and Spekreijse [29]. Spekreijse [29] uses differencing stencils comprised of four cells, and shows that a nine-point 2D upwind stencil changes into a five-point block stencil. Other than the fact that these works are "cell-centred", they shared very little common features with the current CCFD method developed in this thesis.

2.4 Model Equation

Several benchmark test cases for incompressible flows can be found in the literature (eg. cf. [30] and [31]). New types of FV formulations are verified using these benchmark problems [32, 33]. In recent research, a detailed comparison between node-centred and cell-centred FV formulations was conducted by Diskin et al. [6]. Using the Poisson equation as their model equation, six second-order schemes were developed for six typical regular and irregular grids. They found that grid skewness plays a critical role in determining accuracy and stability of FV solutions, and that accuracies of both schemes (node-centred and cell-centred) are comparable at the same degree of freedom.

Maintaining a stable solution with a higher order of accuracy has always been one of the main objectives of new numerical methodologies. Sakai and Watabe [34] have proposed a new trial to increase the accuracy of numerical fluxes in the case of advection-diffusion equations. This further supports the conflicting issues in CFD and the motivation to re-assess the entire CFD approach.

2.5 Accelerating the Solution

The process of solving a physical engineering problem starts with defining the PDEs that model the problem. Then the PDEs are discretized, and the resulting algebraic equations, which can be assembled into a matrix equation, are solved over a discrete set of mesh points. For this purpose, direct and indirect/iterative methods are used. Iterative methods are more common in CFD because the matrices are very large and often have a well-defined sparse structure, and since iterative methods need less computational resources. It is well-known that using a so-called relaxation factor ω improves the convergence rate [10]. The effect of relaxation can be thought of as the addition of only a portion of the residuals, which is the difference in value between two successive iterations, to the calculated value for any node. In another words, it squeezes the gap between the initial and final solution faster. This influence is maximum at the first iterations and reduces as the solution approaches its final state within the pre-defined convergence criterion. There are no general directions for calculating the optimum ω value. Although ω_{opt} can be calculated for some applications, it will be limited to specific details like domain geometry and boundary type. Based on the relaxation parameter value, the iterative solution can be classified as under-relaxed ($0 < \omega < 1$),

over-relaxed ($1 < \omega < 2$) or the basic Gauss-Seidel ($\omega = 1$). In most cases the value of ω_{opt} increases with finer meshes. Versteeg and Malalasekera [23] state that ω_{opt} is mesh dependent and no guidelines exist for exact value computation. Attempts to set up a numerical formulation that doesn't rely on a relaxation parameter have been conducted, like the work of Majumdar [24]. However, current CFD simulations still rely heavily on the idea that relaxation can provide accurate and stable solutions over a wide range of applications.

2.6 Thesis Objectives

In this thesis, the CCFD approach reveals new vistas for research across the whole scope of CFD. A new FD scheme is presented for structured grid systems, with the flexibility to be extended to unstructured grids. As in all fundamental research in CFD, to introduce a new differencing scheme, a single PDE is used, e.g. [31]. The 2D Laplace equation is considered as the model equation in this study,

$$\frac{\partial^2 \emptyset}{\partial x^2} + \frac{\partial^2 \emptyset}{\partial y^2} = 0 \tag{2.1}$$

with Dirichlet and/or Neumann boundary conditions. To develop and explain the method, the mesh is initially constructed with rectangular cells of uniform size. The FD scheme is derived from the differential form of the conservation laws.

The thesis is organized as follows. The basic formulation of the CCFD method is described in Chapter III. Many of the critical aspects of any useful numerical method are explored in this chapter to ensure that the CCFD methodology meets the demands of a reliable, flexible, accurate and viable CFD code. Various tests are performed, using model equation (2.1) with Dirichlet boundary conditions. Further verifications are carried

out in Chapter IV, where the CCFD method is applied to a Poisson equation with Dirichlet boundary conditions, a Laplace equation with Neumann boundary conditions, and a model convection-diffusion equation with Dirichlet conditions. The CCFD method is then applied to a benchmark incompressible fluid flow problem, that is, the flow over a backward-facing step, in Chapter V. Chapter VI includes the conclusions from this research and suggests some avenues for further developments and future studies with the CCFD method.

CHAPTER III

DESIGN AND METHODOLOGY

3.1 The CCFD Scheme

Consider an arbitrary polygonal cell in 2D bounded by the line segments joining nodes *N1*, *N2*, *N3*, *N4* and *N5*, surrounded by neighbouring cells as shown in Fig. 3.1.

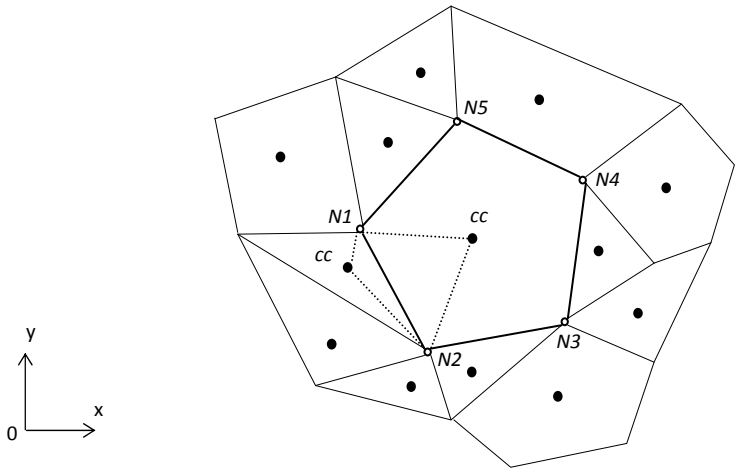


Figure 3.1: An arbitrary mesh topology

The coordinates of nodes *N1*,…,*N5* with respect to a fixed global Cartesian coordinate system Oxy are assumed to be known. An arrangement of nodes such as shown in Fig. 3.1 does not allow for a traditional FD formulation of the governing PDE since there is no orderly pattern to the placement of nodes. Traditional FD methods requires that all nodes (grid points) lie at the intersection of lines $x = x_i$ and $y = y_j$.

On the other hand, the FV and FE methods can be implemented on polygonal cells. For these methods, the governing PDE is first reformulated as an integral equation by performing a double integration of the PDE over the region contained within the cell. In the cell-centred FV method, the double integral is then converted into a line integral around the boundary of the cell using Green's theorem. This line integral is written as the

sum of line integrals along the straight lines joining the nodes, i.e. along the faces (or edges) of the cell. For example, consider the line integral along the face joining nodes *N1* and *N2*. To evaluate the integral exactly we need to know the variation of $\emptyset$ (or perhaps $\emptyset_x$ or $\emptyset_y$) along the face. Since this variation is unknown (otherwise we would already know the solution to the original PDE), it must be approximated. A simple approximation would be to take $\emptyset$ to be constant along the face, with value as the average of $\emptyset$ at the two end nodes *N1* and *N2*. Alternatively, we could approximate $\emptyset$ using the cell-centre values of the two cells sharing the face joining *N1* and *N2*. Applying this procedure to all cells in the domain, one can formulate a system of algebraic equations for the values of $\emptyset$ at all cell centres. This system is then solved, either by direct or iterative matrix solvers. If nodal values are required, e.g. at node *N1*, it can be obtained by taking a distance weighted average of cell-centre values for all cells sharing node *N1*.

In the FE method, the integrand is approximated over the cell, and then the double integration is performed exactly. The approximation will involve the unknown values of $\emptyset$ at the nodes of the cell, and must be a function which is integrable over the cell area. Assembling the equations obtained from each cell will lead to an algebraic system of equations for the unknown nodal values.

In the CCFD method, the PDE is applied at the centre of each cell, and the partial derivatives in the PDE are approximated by finite differences. To accomplish this, a finite difference stencil is placed inside the cell, with the centre of the stencil located at the centroid of the cell, as illustrated in Fig. 3.2. The arms of the stencil are positioned to be parallel to the global Cartesian coordinate axes. The end points of the stencil lie at the intersection of the stencil arms with the faces of the cell. These intersection points are

denoted by *w*, *e*, *n* and *s*. Standard finite difference formulae require uniform spacing between grid points, or, in the present context, that the distance from *s* to *cc* equals that from *cc* to *n*. Clearly, for an arbitrary cell topology, this will not be the case.

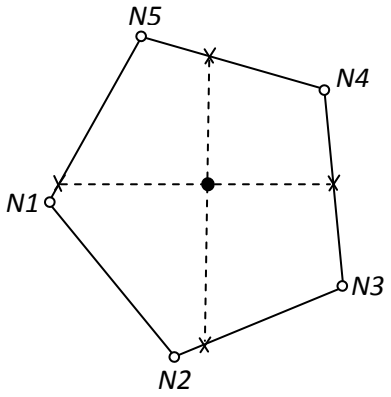


Figure 3.2: A finite difference stencil in an arbitrary pentagonal cell

One option would be to use finite difference approximations that account for variable grid spacing, but this degrades the accuracy of the approximation. A better approach, which preserves the accuracy of the standard formulae, is to map the non-uniform stencil to a uniform one, as demonstrated in Fig. 3.3.

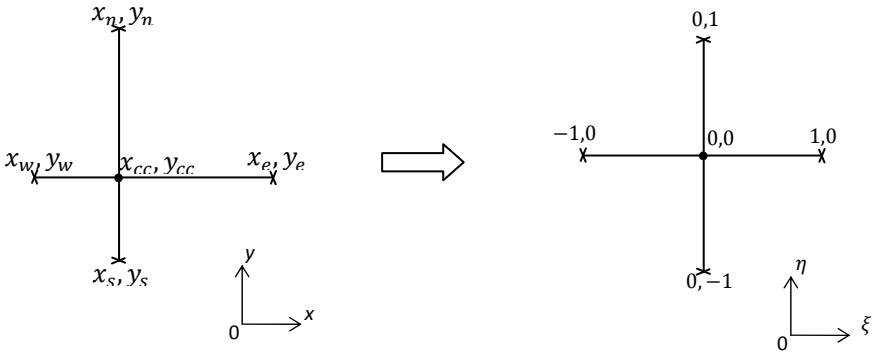


Figure 3.3: Mapping the physical stencil to a uniform computational stencil

The line segment joining *w* to *e* is mapped to $-1 \leq \xi \leq 1$, with *cc* mapped to $\xi = 0$. Similarly, the stencil arm joining *s* to *n* is mapped to $-1 \leq \eta \leq 1$, with *cc* mapped to $\eta = 0$. These mappings can be expressed as quadratic functions:

$$\begin{aligned} x &= a_2\xi^2 + a_1\xi + a_0 \\ y &= b_2\eta^2 + b_1\eta + b_0 \end{aligned} \tag{3.1}$$

Denoting the coordinates (with respect to the global system) of *w* by (x_w, y_w), coordinates of *e* by (x_e, y_e), etc., it can be shown that the coefficients in (3.1) are given by:

$$\begin{aligned} a_0 = x_{cc}\ ,\ \ a_1 = \tfrac{1}{2}(x_e - x_w)\ ,\ \ a_2 = \tfrac{1}{2}(x_e + x_w) - x_{cc} \\ b_0 = y_{cc}\ ,\ \ b_1 = \tfrac{1}{2}(y_n - y_s)\ ,\ \ b_2 = \tfrac{1}{2}(y_n + y_s) - y_{cc} \end{aligned} \tag{3.2}$$

Standard finite difference formulae can be used in the (ξ, η) system, in which $\Delta\xi = 1$, $\Delta\eta = 1$. To apply these standard approximations, the governing PDE must also be transformed to the (ξ, η) coordinates. Using chain rules, we get, for example,

$$\begin{aligned} \frac{\partial \emptyset}{\partial x} &= \frac{\partial \emptyset}{\partial \xi}\frac{d\xi}{dx} = \frac{1}{x'}\frac{\partial \emptyset}{\partial \xi} \\ \frac{\partial^2 \emptyset}{\partial x^2} &= \frac{1}{x'}\frac{\partial}{\partial \xi}\left(\frac{1}{x'}\frac{\partial \emptyset}{\partial \xi}\right) = \frac{1}{x'^2}\frac{\partial^2 \emptyset}{\partial \xi^2} - \frac{x''}{x'^3}\frac{\partial \emptyset}{\partial \xi} \end{aligned} \tag{3.3}$$

and similarly for $\frac{\partial \emptyset}{\partial y}$, $\frac{\partial^2 \emptyset}{\partial y^2}$ and higher order derivatives. In these expressions the metrics are given by $x' = \frac{dx}{d\xi} = 2a_2\xi + a_1$ and $x'' = \frac{d^2x}{d\xi^2} = 2a_2$. Since the partial derivatives are locally approximated at the cell centre, *x*′, *x*″, *y*' and *y*" are evaluated at $\xi = 0, \eta = 0$. Therefore, $x' = a_1,\ x'' = 2a_2,\ \ y' = b_1$ and $y'' = 2b_2$. Using eqn. (3.2), the metrics can be evaluated in terms of the coordinates of *w*, *e*, *n*, *s* and *cc*, i.e.

$$x' = \frac{1}{2}(x_e - x_w)\,, \quad x'' = (x_e + x_w) - 2x_{cc}$$
$$y' = \frac{1}{2}(y_n - y_s)\,, \quad y'' = (y_n + y_s) - 2y_{cc} \tag{3.4}$$

Note that the values of the metrics are cell specific. The governing PDE is not globally transformed under eqn. (3.1). It is uniquely transformed for each individual cell.

To illustrate the fundamental concepts of the CCFD method, consider the Laplace equation:

$$\nabla^2 \emptyset = 0 \tag{3.5}$$

Using transformation (3.1) and the relationships (3.3), the Laplace equation in the ξ, η coordinates becomes:

$$\frac{1}{x'^2}\frac{\partial^2 \emptyset}{\partial \xi^2} + \frac{1}{y'^2}\frac{\partial^2 \emptyset}{\partial \eta^2} - \frac{x''}{x'^3}\frac{\partial \emptyset}{\partial \xi} - \frac{y''}{y'^3}\frac{\partial \emptyset}{\partial \eta} = 0 \tag{3.6}$$

where x', x'', y' and y'' are given by eqn. (3.4). Now, suppose three-point central differencing is used to approximate the derivatives in eqn. (3.6), i.e.

$$\frac{\partial^2 \emptyset}{\partial \xi^2} \approx \emptyset_e - 2\emptyset_{cc} + \emptyset_w, \qquad \frac{\partial \emptyset}{\partial \xi} \approx \frac{1}{2}(\emptyset_e - \emptyset_w)$$
$$\frac{\partial^2 \emptyset}{\partial \eta^2} \approx \emptyset_n - 2\emptyset_{cc} + \emptyset_s, \qquad \frac{\partial \emptyset}{\partial \eta} \approx \frac{1}{2}(\emptyset_n - \emptyset_s) \tag{3.7}$$

Applying the approximations (3.7) to the eqn. (3.6), the resulting finite difference equation can be written as,

$$a_{cc}\emptyset_{cc} = a_w\emptyset_w + a_e\emptyset_e + a_s\emptyset_s + a_n\emptyset_n \tag{3.8}$$

where

$$a_{cc} = \frac{1}{(x_e - x_w)^2} + \frac{1}{(y_n - y_s)^2}$$

$$a_w = \frac{x_e - x_{cc}}{(x_e - x_w)^3}, \quad a_e = \frac{x_{cc} - x_w}{(x_e - x_w)^3}$$

$$a_s = \frac{y_n - y_{cc}}{(y_n - y_s)^3}, \quad a_n = \frac{y_{cc} - y_s}{(y_n - y_s)^3}$$

Following procedures similar to those used in the FV method, the values of $\emptyset_w$, $\emptyset_e$, $\emptyset_s$ and $\emptyset_n$ can be approximated in terms of cell nodal values and neighbouring cell centroid values. Equations for the nodal values can be constructed using the distance weighted average of cell-centre values for all cells sharing the node. Assembling the resulting equations at all cell centroids and all nodes leads to a system of linear algebraic equations which can be solved by standard numerical procedures.

In this thesis, rather than assemble the large matrix system, we use an iterative approach. An initial guess for $\emptyset$ at all cell centroids and nodes is made. A particular node *P* is selected and all cells sharing that node are identified. Then, for each of these cells, $\emptyset_w$, $\emptyset_e$, $\emptyset_s$ and $\emptyset_n$ are evaluated based on the initial guess. Equation (3.8) is used to update the value of $\emptyset_{cc}$ for each of the cells. The value of $\emptyset$ at the selected node *P* is then updated using

$$\emptyset_P = \frac{\sum_{i=1}^{N_P} \frac{\emptyset_{cci}}{L_{cci}}}{\sum_{i=1}^{N_P} \frac{1}{L_{cci}}} \tag{3.9}$$

where N_P is the number of cells sharing node *P,* $cc1$, ... , cc_{Np} are the centroids of these cells and L_{cci} is the distance from centroid cc_i to the node *P*.

The next node is selected and the above procedure is followed to update the nodal value, until all nodes in the mesh have been updated, completing the first iteration. The procedure is repeated until a prescribed convergence criterion is satisfied.

3.2 A Simple Test Problem

A simple test problem is chosen to demonstrate how the CCFD method is implemented. Consider the solution of the Laplace equation (3.5) on a square domain with four square cells, as shown in Fig. 3.4. The domain size is 1 unit by 1 unit, with equal grid spacing ($\Delta x = \Delta y$). Dirichlet conditions are applied, with all boundaries set to be zero except for the left boundary, which is taken to have the value 1.

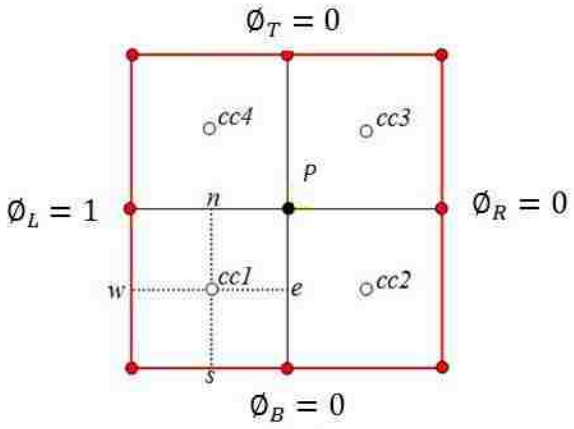


Figure 3.4: Simple domain with 4 identical cells

3.2.1 Solution procedure

In this example, the only node to be evaluated is the domain central node (*P*). The general procedure is as follows:

a. find the cells that share the current node (i.e. node *P*)

b. for each one of these cells;

   i. calculate the *cc* coordinates, and the coordinates of *w*, *s*, *e* and *n* intersections.

   ii. calculate $\emptyset_e$ by distance weighted averaging between the two centres of the cells that share the edge containing *e* (e.g. *cc1* and *cc2* in Fig. 3.4).

Similarly, evaluate $\emptyset_n$, $\emptyset_w$ and $\emptyset_s$. If the intersection lies on a boundary, use the corresponding boundary value.

iii. evaluate $\emptyset_{cc}$ from the discretized FD form of the model equation. Second order central differencing is used, leading to eqn. (3.8).

c. update node *P* by a distance weighted averaging from all adjacent cell centres, using eqn. (3.9).

The calculations start with an initial guess at *P* and all the cell centres, which are then updated iteratively until the convergence criterion is satisfied.

For the domain shown in Fig. 3.4, using the CCFD formulation with successive over-relaxation and a relative difference between iterates of 10E-9, $\emptyset_P$ was found to be 0.25 after 7 iterations (ω = 1.413). This is in excellent agreement with the exact analytical value of 0.25. The analytical solution for this test problem can be found in many resources, and is given by:

$$\emptyset(x,y) = \lim_{K\to\infty}\left(\frac{4\emptyset_L}{\pi}\sum_{n=odd}^{K}\frac{sinh\left(\frac{n\pi}{H}(L-x)\right)sin\,(\frac{n\pi}{H}y)}{n\,sinh(\frac{n\pi}{H}L)}\right) \tag{3.10}$$

where $\emptyset_L$ is the left boundary value, *L* and *H* are the length and the height of the domain, *K* is the number of terms in the summation and (*x, y*) are the Cartesian coordinates of any internal point in the domain (not necessarily a node).

For each cell in the CCFD formulation, the property value of the north, east, south and west intersections must be carefully calculated. The intersection may coincide on a node, in which case the nodal value is assigned to the intersection, whether the node is a

boundary or an internal one. This may occur in an unstructured grid with triangular cells, or a hybrid mesh (see Fig. 3.5f). Alternatively, the intersection may lie on a boundary face (a face is a line that joins two nodes). In this case the boundary value is directly assigned to the intersection, e.g. $\emptyset_w = 1$ for cell number one in Fig. 3.4. This means that the second part of Step b of the solution procedure is only applied when the intersection lies on an internal face, and is modified otherwise as explained above.

For square cells, instead of using the two adjacent cell centres, the intersection values can also be calculated by a distance weighted average between the two nodes forming the face on which the intersection lies. In this case, the differencing stencil incorporates the property values at the corners of the cell. This is equivalent to a nine-point formula of FV and FE which is derived in [8]. In other words, the CCFD scheme uses four "corner" values, while the traditional FD formulation does not. Other methods of calculating the intersection values are explained in section 3.4 of this chapter.

3.2.2 Mesh refinement and solution relaxation

With the same domain size and boundary conditions mentioned above in section 3.2, several types of (uniform) grid refinements and cell arrangements have been tested, without clustering. The grid arrangements and results for the domain central node value are shown in Figs. 3.5a-f. For cell aspect ratio ($\beta = \Delta x/\Delta y$) not equal to 1, i.e. non-square cells, the accuracy of the solution is reduced for coarse grids. This decrease in accuracy is also observed with traditional FD solutions for the same grids. Figures 3.5a and 3.5d show the results of refinement in the *x*-direction, corresponding to cell aspect ratios less than 1, i.e, $\beta = 0.5$ and 0.25, respectively. In both cases, the value of $\emptyset_P$ is less than the exact value of 0.25. This is due to the fact that, for these coarse grids, the solution is too

heavily influenced by the boundary conditions on the right, bottom and top of the domain (all zeros). The non-zero value on the left boundary, which drives the solution away from zero, only influences the solution through one node on the left boundary. The exact solution can be recovered by inserting additional nodes on the left boundary, as shown in Fig. 3.5c. The reverse occurs when the cell aspect ratio is taken greater than 1, as illustrated in Figs. 3.5b and 3.5e, where the non-zero left boundary dominates the solution. However, with CCFD, for any value of β, grid refinement improves the solution. Some typical results are tabulated in Table 3.1. M and N are the number of grid points in the *x*- and *y*-directions, respectively.

For a hybrid mesh (a mix of two different cell types in one domain), as shown in Fig. 3.5f, $\emptyset_P$ was found to be 0.2518. It should be mentioned here that eqns. (3.8) and (3.9) are solved for the property value at the cell centres and nodes respectively. This case provides a counterexample to the widely-held argument (cf., e.g. [6]), that the bottleneck of the FD formulation is that it cannot handle an unstructured mesh system over a complex geometry, because it requires a topologically square network of lines to discretize the PDEs. In fact, the present CCFD scheme is able to handle an arbitrary unstructured mesh made up of triangular cells or a hybrid mesh comprised of both quadrilateral and triangular cells.

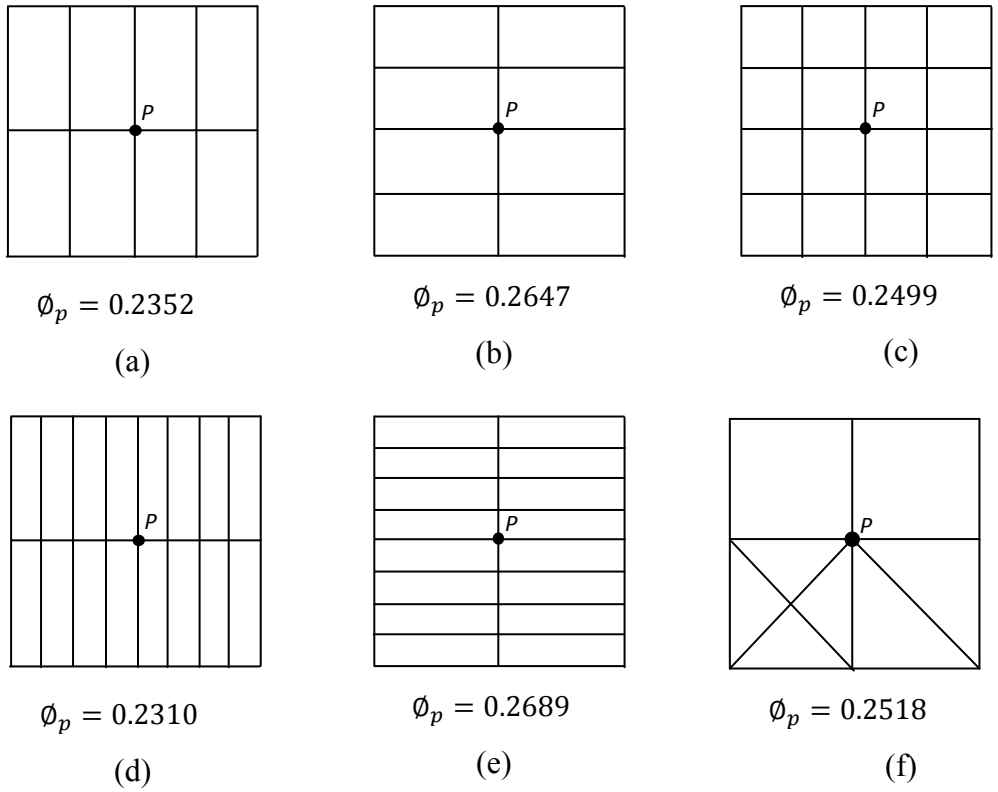


Figure 3.5: Different types of grid arrangement: (a) 5x3 grid, (b) 3x5 grid, (c) 5x5 grid, (d) 9x3 grid, (e) 3x9 grid, (f) hybrid grid

Table 3.1: Central node values for different aspect ratios and mesh refinements

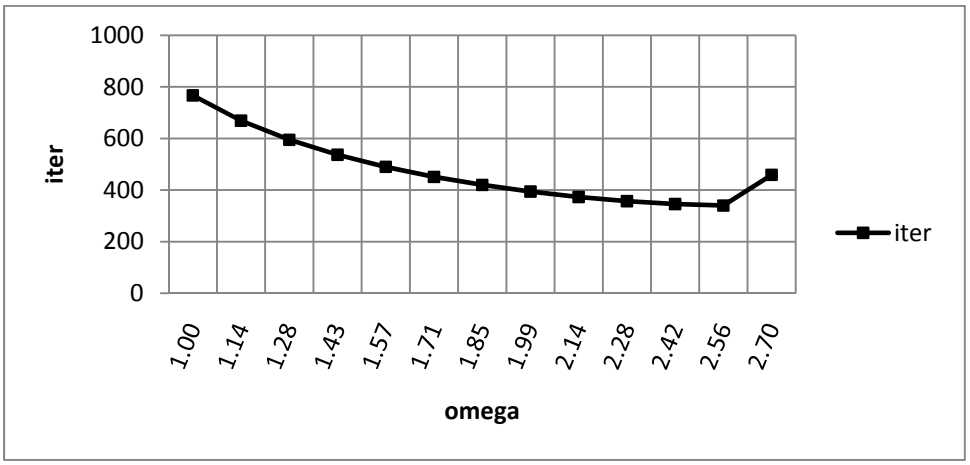
β	**MxN**	**ϕ_P**	**iter**
0.25	9x3	0.2310	63
0.25	49x13	0.2494	551
0.5	5x3	0.2353	20
0.5	25x13	0.2495	175
1	3x3	0.2500	7
1	13x13	0.2500	94
2	3x5	0.2647	16
2	13x25	0.2505	281
4	3x9	0.2690	63
4	13x49	0.2506	955

Table 3.1 indicates that a finer mesh recovers the loss in accuracy created by non-square cells, although more computation time will be associated with the higher number of cells. In this case, an accelerated solution will be desired for the iterative solver. Hoffmann and Chiang [10] state that introducing a relaxation factor ω into the discrete form of the FD approximation of the PDE should accelerate the solution (i.e. reduce the number of iterations), provided that $1 \leq \omega \leq 2$. In the current research, several grids (with rectangular cells) have been tested with a relaxed form of the CCFD scheme, which is expressed as:

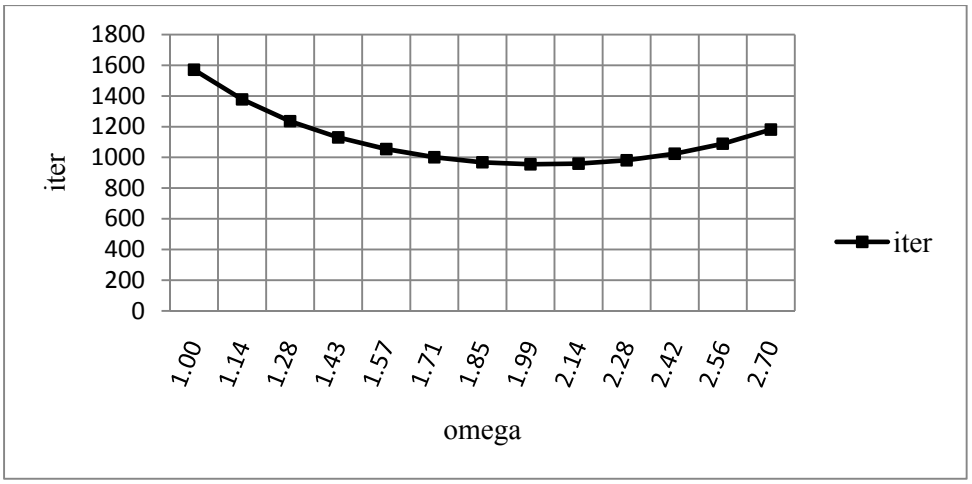
$$\varphi_{cc}^{n+1} = \varphi_{cc}^{n} + \omega(\tilde{\varphi}_{cc}^{n+1} - \varphi_{cc}^{n}) \tag{3.11}$$

where n+1 and n are the current and the previous iteration indices respectively, and the tilda indicates the cell-centre value calculated from eqn. (3.8).

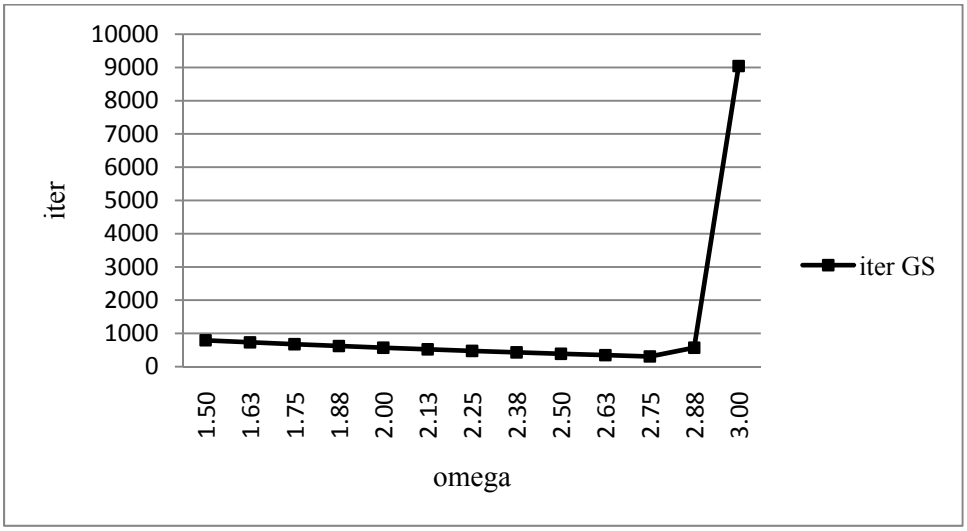
CCFD results for two grid systems are shown in Fig. 3.6, illustrating the relationship between the number of iterations required for convergence and the value of the relaxation factor. The optimum value for the relaxation factor was found to be 2.56 and 1.99 for 25x25 and 13x49 grid sizes respectively.



(a)



(b)



(c)

Figure 3.6: Effect of relaxation factor for CCFD scheme, (a) 25x25 grid, (b) 13x49 grid, (c) 49x13 grid

3.3 Solution of Test Problem

3.3.1 Uniform grid

A 25x25 grid has been selected to check many of the important aspects of the iterative numerical solution (optimum relaxation factor, number of iterations, average absolute error, etc.). A comparison between the exact analytical solution (eqn. (3.10)), the CCFD solution and a traditional FD point successive over-relaxation (PSOR) solution is

shown in Fig. 3.7. A relative difference between iterations is used for the convergence criterion, with tolerance less than or equal to 10E-9 at each node, for both the CCFD and PSOR solutions. The equations used for the PSOR scheme can be found in [10] (p.164, eqns. (5-18) through (5-20)). The optimum relaxation factor for the PSOR scheme was found to be 1.769, which made the solution converge after 86 iterations, while the optimum relaxation factor for the CCFD scheme was found to be 2.56 (in a tested range of 1.0 to 2.7) with 340 iterations to converge (see Fig. 3.6a). For most of the grid sizes that were tested, choosing ω beyond this range (1.0 to 2.7) leads to a sudden large increase in the number of iterations, as seen in Fig. 3.6c, or even divergence in some cases.

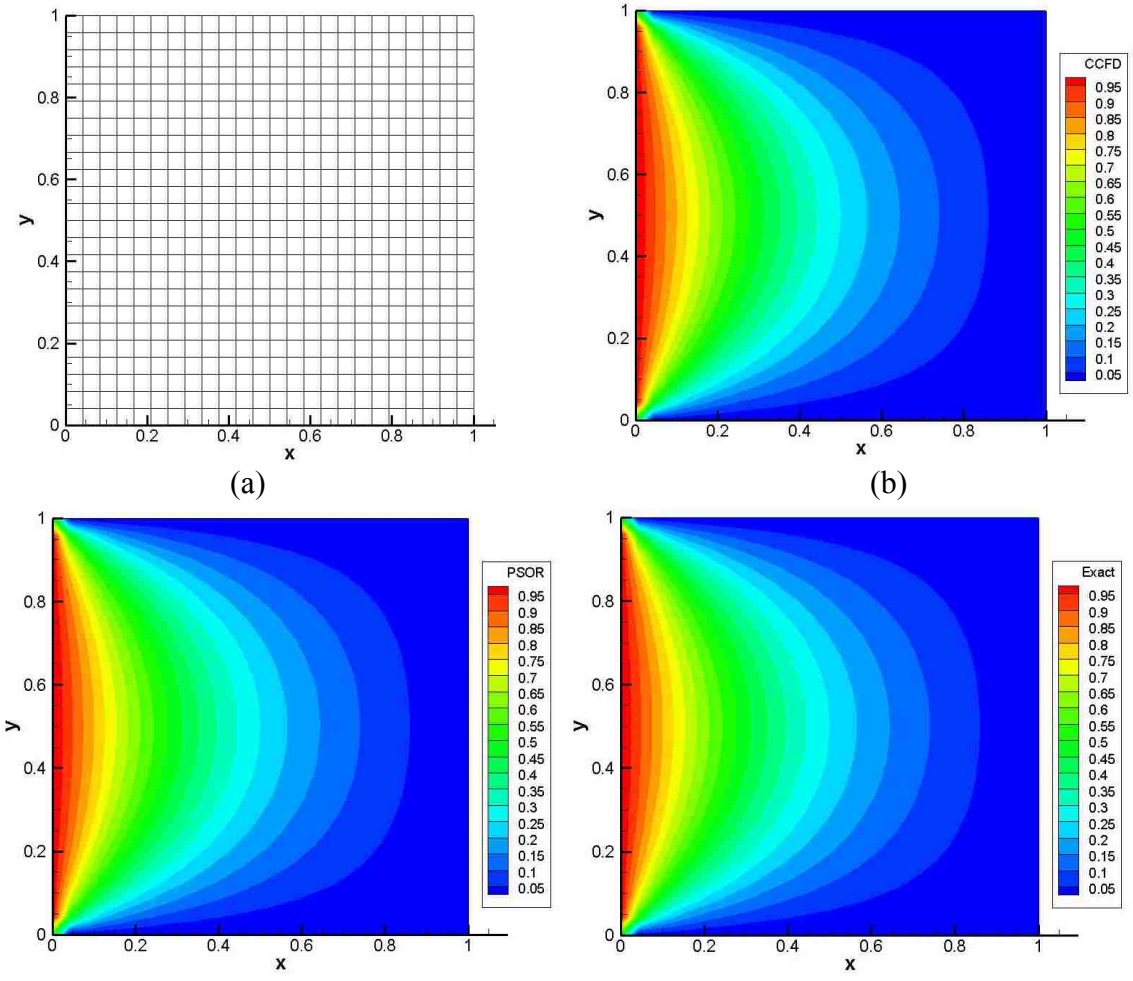
In the FV method, under-relaxation is normally used to control the oscillations in an iterative solution of this problem. Majumdar [24] proposed a FV formulation for calculating the velocity value at cell faces which preserves a convergent solution, independent of the under-relaxation factor and without significant increases in computational time. In another word, trials were made to untie available FV formulations with the classical relaxation bounds, i.e. ($0 < \omega < 1$). Such a decoupling may be possible in the CCFD method, but has not been explored in this thesis.

The average absolute error (for both CCFD and PSOR) over all the interior nodes is the difference between the numerical and exact analytical solution. However the "accuracy" of the exact solution itself, which is calculated from eqn. (3.10), depends on the number of terms (K) that are included in the summation. Figures 3.7d and 3.7e show the effect of a high or low number of terms. This leads to variations in the average error of CCFD and PSOR. Table 3.2 shows the decrease in the average error as we increase the

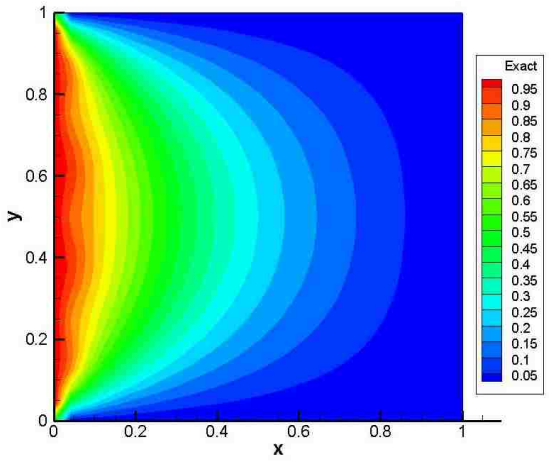
number of terms in the exact solution (i.e. in eq. (3.10)) without any modification to the grid. Naturally, the higher number of terms (K) in a relatively fine mesh (e.g. 49x49) is associated with increasing the processing time. However, most importantly, this example demonstrates that the CCFD method provides a solution with the same level of accuracy as the traditional FD method.

Table 3.2: Average absolute error for a 25x25 grid for different number of terms (K) in the infinite series solution

K	CCFD	PSOR
7	0.00215	0.00215
11	0.00098	0.00098
33	0.00041	0.00041



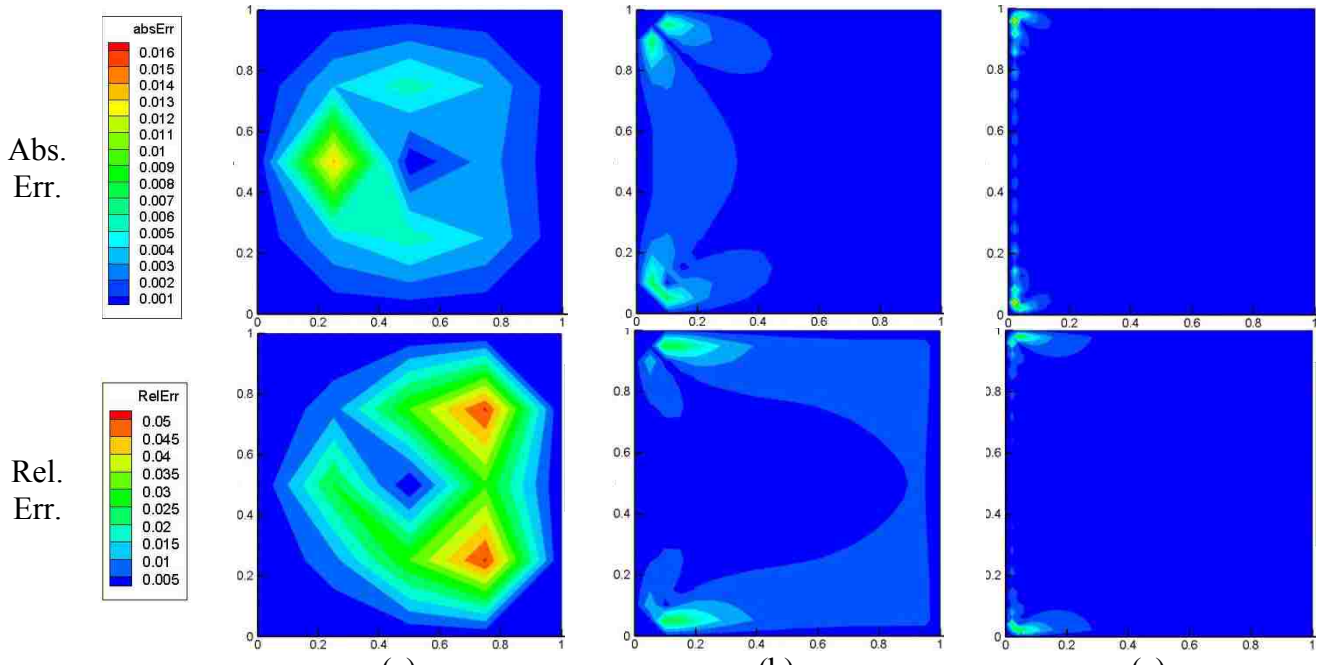
(c) (d)



(e)

Figure 3.7: Comparison of exact analytical, PSOR and CCFD schemes for a 25x25 uniform grid: (a) grid, (b) CCFD solution, (c) PSOR solution, (d) exact solution (K = 33), (e) exact solution (K = 7)

The effect of grid refinement on the solution accuracy is shown Fig. 3.8. Three structured grids are used for this purpose.



(a) (b) (c)

Figure 3.8: Effect of grid refinement on error distribution: (a) 5x5 grid, (b) 21x21 grid, (c) 49x49 grid

3.3.2 Clustered grid

To explore the effect of clustering grid lines toward different directions, first some simple cases of clustering are considered on coarse grids, then finer grids are considered. Again, the same domain size and boundary conditions mentioned in section 3.2 will be utilized. Figures 3.9a and 3.9b show the addition of nodes toward the bottom and the left boundaries, respectively. One may expect that adding nodes close to the left boundary would make the results more accurate, but this is not the case for coarse grids. To understand the reason for this, note that in Fig. 3.9a, two nodes lay on the left boundary, which is the non-zero boundary, and four nodes fall on the other three zero boundaries, giving a ratio of two to four, and a reasonably accurate solution $\emptyset_P = 0.2570$. On the other hand, in Fig. 3.9b the same ratio of nodes is one to five, and the solution $\emptyset_P = 0.2230$ is considerably less accurate. Comparing Fig. 3.9b and 3.9c makes the scope more clear, where the ratio of non-zero to zero boundary nodes is changed from 1 by 5 to 3 by 7, which brings the value of $\emptyset_P$ from 0.2230 to 0.2522. In another word, the number of nodes on any boundary reflects the effect of that boundary on interior nodes, implying that the number of grid points in each direction must be chosen so that all boundaries sufficiently influence the interior solution. However, this significant effect is true for coarse grids, and may or may not have the same impact in fine or dense grids.

With the grid clustered toward the left boundary, as shown in Figs. 3.10a and 3.10b, the CCFD method shows a close match with the analytical solution (see Figs. 3.10c and 3.10d). Two types of clustering procedures were used to design these grid systems, i) a logarithmic clustering function (Fig. 3.10a), and ii) a clustering scale factor

(Fig. 3.10b). Packing the grid to the left boundary using a scale factor α means that the ratio of adjacent cell lengths is α, i.e. $\Delta x_{i-1} = \alpha \Delta x_i$.

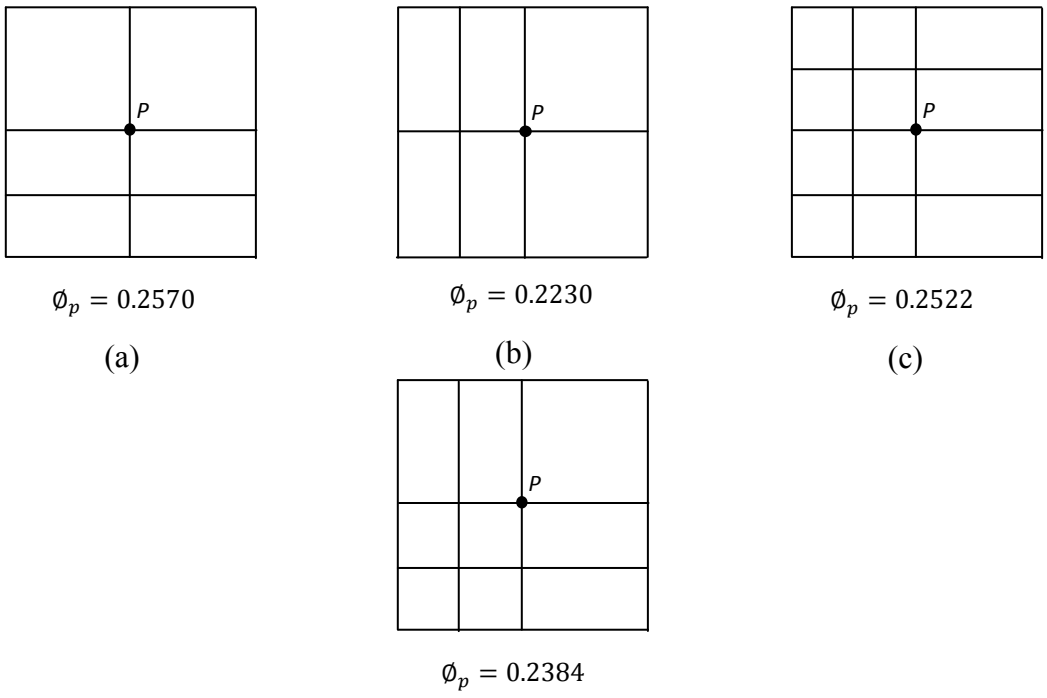


Figure 3.9: Clustered grids toward different boundaries: (a) bottom clustered, (b) left clustered, (c) left clustered and grid refined, (d) bottom and left clustered

The grid in Fig. 3.10b, which exhibits tight packing at the left boundary, was generated with α = 0.8. For some of these types of grids, the cell aspect ratio gets extremely small (β < 0.00014 for the 49x25 grid in Fig. 3.10b). Although most other numerical solution schemes have difficulty dealing with small (or large) aspect ratios, it is interesting to observe that the CCFD scheme can handle this extremely small aspect ratio, and produces a highly accurate solution. In this example, clustering also leads to an increase in the number of iterations for the CCFD scheme, from 340 iterations for the uniform grid with ω_{opt} = 2.56, to 372 iterations for the clustered grid with the same relaxation factor.

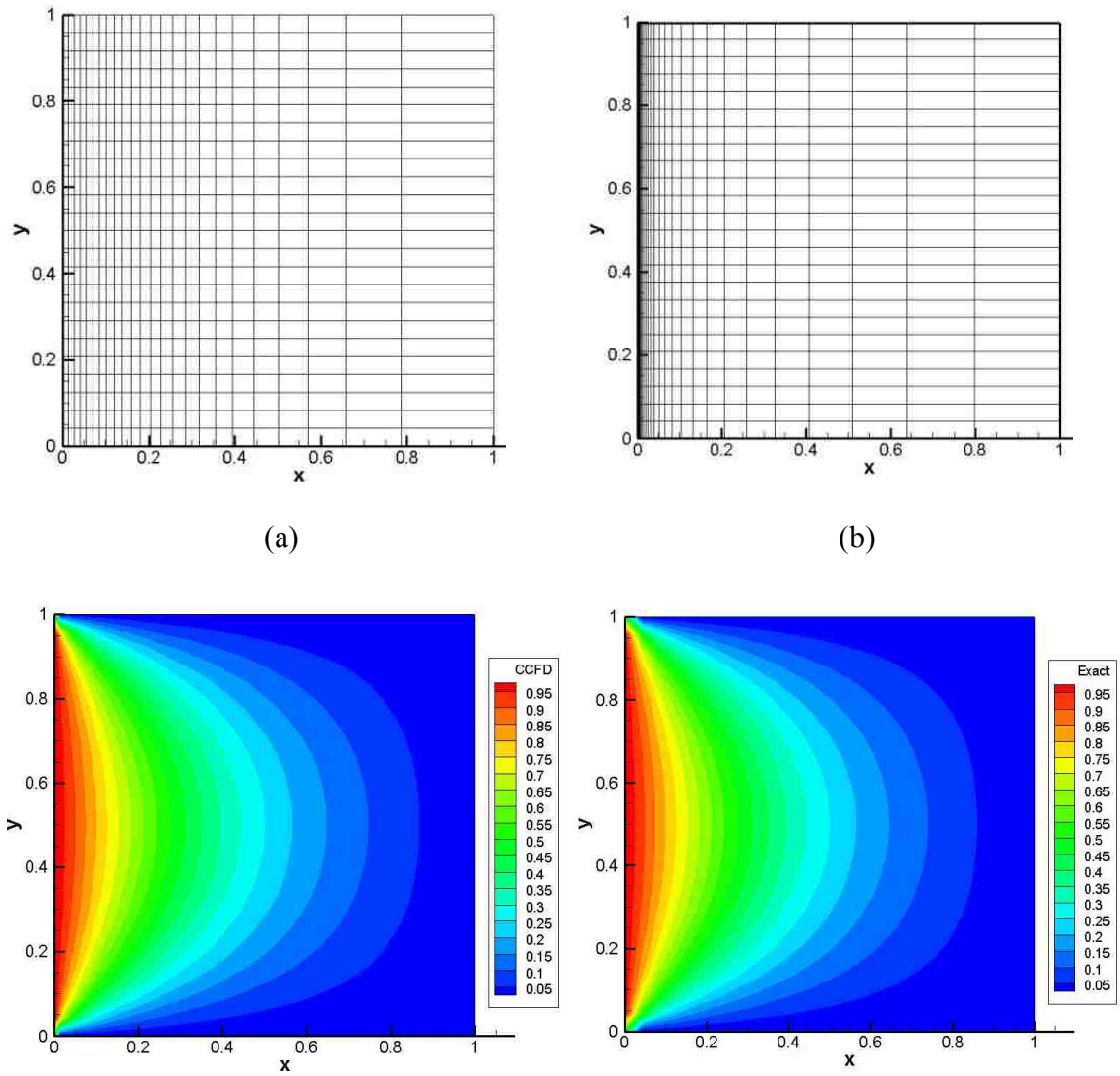


Figure 3.10: Clustering effect: (a) 25x25 clustered grid, (b) 49x25 clustered grid, (c) CCFD solution, (d) exact analytical solution (K = 33)

3.4 Alternative Methods to Calculate Intersection Values

There are other methods to calculate the property value at the four intersections (*n, s, e* and *w*) from cell centres and/or nodes. All of the ones discussed below involve distance weighted averaging from the face end nodes and/or cell centres to the

intersection. We illustrate these procedures by considering the evaluation of the east (e) intersection point in a cell.

3.4.1 Two cell centres

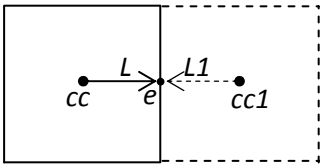


Figure 3.11: Two cell centres scheme

$$\emptyset_e = \frac{\frac{\emptyset_{cc}}{L} + \frac{\emptyset_{cc1}}{L1}}{\frac{1}{L} + \frac{1}{L1}} \tag{3.12}$$

where $L = |x_{cc} - x_e|$ and $L1 = |x_{cc1} - x_e|$.

3.4.2 Two end points (nodes) of the face

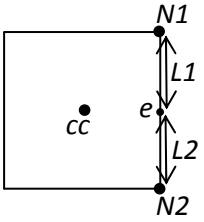


Figure 3.12: Two end nodes of the face scheme

$$\emptyset_e = \frac{\frac{\emptyset_{N1}}{L1} + \frac{\emptyset_{N2}}{L2}}{\frac{1}{L1} + \frac{1}{L2}} \tag{3.13}$$

where $L1 = |y_e - y_{N1}|$ and $L2 = |y_e - y_{N2}|$.

3.4.3 Two end nodes of the face and the cell centre

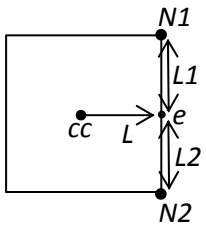


Figure 3.13: Two end nodes of the face and the cell centre scheme

$$\emptyset_e = \frac{\frac{\emptyset_{N1}}{L1} + \frac{\emptyset_{N2}}{L2} + \frac{\emptyset_{cc}}{L}}{\frac{1}{L1} + \frac{1}{L2} + \frac{1}{L}} \tag{3.14}$$

where $L = |x_{cc} - x_e|$, $L1 = |y_e - y_{N1}|$ and $L2 = |y_e - y_{N2}|$.

3.4.4 Control volume

In this case, we construct a control volume from the two face end points (nodes), the current cell centre and the centre of the cell adjacent to the east face.

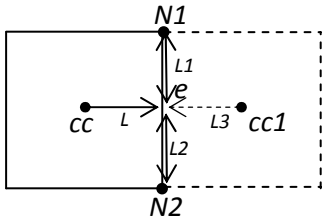


Figure 3.14: Control volume scheme

$$\emptyset_e = \frac{\frac{\emptyset_{cc}}{L} + \frac{\emptyset_{N1}}{L1} + \frac{\emptyset_{N2}}{L2} + \frac{\emptyset_{cc1}}{L3}}{\frac{1}{L} + \frac{1}{L1} + \frac{1}{L2} + \frac{1}{L3}} \tag{3.15}$$

where $L = |x_{cc} - x_e|$, $L1 = |y_e - y_{N1}|$, $L2 = |y_e - y_{N2}|$ and $L3 = |x_{cc1} - x_e|$.

3.4.5 Four vertices of the cell

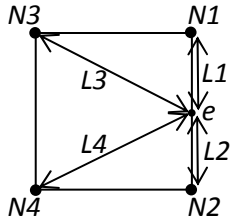


Figure 3.15: Four vertices scheme

$$\emptyset_e = \frac{\frac{\emptyset_{N1}}{L1} + \frac{\emptyset_{N2}}{L2} + \frac{\emptyset_{N3}}{L3} + \frac{\emptyset_{N4}}{L4}}{\frac{1}{L1} + \frac{1}{L2} + \frac{1}{L3} + \frac{1}{L4}} \tag{3.16}$$

where $L1 = |y_e - y_{N1}|$, $L2 = |y_e - y_{N2}|$,

$L3 = \sqrt{(x_e - x_{N3})^2 + (y_e - y_{N3})^2}$

and $L4 = \sqrt{(x_e - x_{N4})^2 + (y_e - y_{N4})^2}$

Schemes 3.4.2, 3.4.3 and 3.4.5 are more relevant to the core of the CCFD formulation, starting from the name CCFD, i.e. all the necessary information for the numerical evaluation of the calculated property at the cell centre comes from the current cell nodes and/or centre. This feature adheres to an important principle in the development of the CCFD method, to treat an unstructured mesh with its simplest formulation. In this context, simplest refers to two significant issues in the numerical solution of PDEs that govern physical processes. First, these schemes give more

flexibility to CCFD to handle an arbitrary number of cells and types that may share a single node. Also, it allows the CCFD method to handle cells that possess a number of nodes greater than its physical nodes at the cell vertices. These are commonly referred to as hanging nodes and occur on the two layers of cells that lay on the interface of a non-conformal mesh, as illustrated in Fig. 3.16.

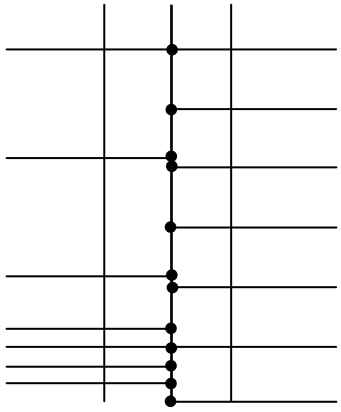


Figure 3.16: Node distribution along the vertical interface of a non-conformal mesh

Secondly, these schemes permit easier programming because no information from neighbour cells is needed. In the cases where data is needed from out of the cell, like the two cell centres method, the neighbouring cells should be identified. This can be done out of the solver part of the program, i.e. in the geometry part, in which case more computer memory is claimed. Alternatively, if it is kept within the PDE solver, then extensive processing time is required with larger meshes. The extra processing time originates from the searching process for the second cell that shares the same couple of nodes with the current cell, which must be repeated for every iteration of the solution process. To

imagine the size of searching, consider a structured grid of square cells with 49 by 49 nodes. Then,

No. of internal nodes = (49-2) X (49-2) = 2209

No. of cells to be evaluated = (No. of internal nodes) X (No. of cells adjacent to each node) = 2209 X 4 = 8836

No. of searches = (No. of cells to be evaluated) X (No. of intersections for each cell)

= 8836 X 4 = 35344

To illustrate this further, see Fig. 3.17 which shows the exponential increase in the number of searches as the number of internal mesh points increase.

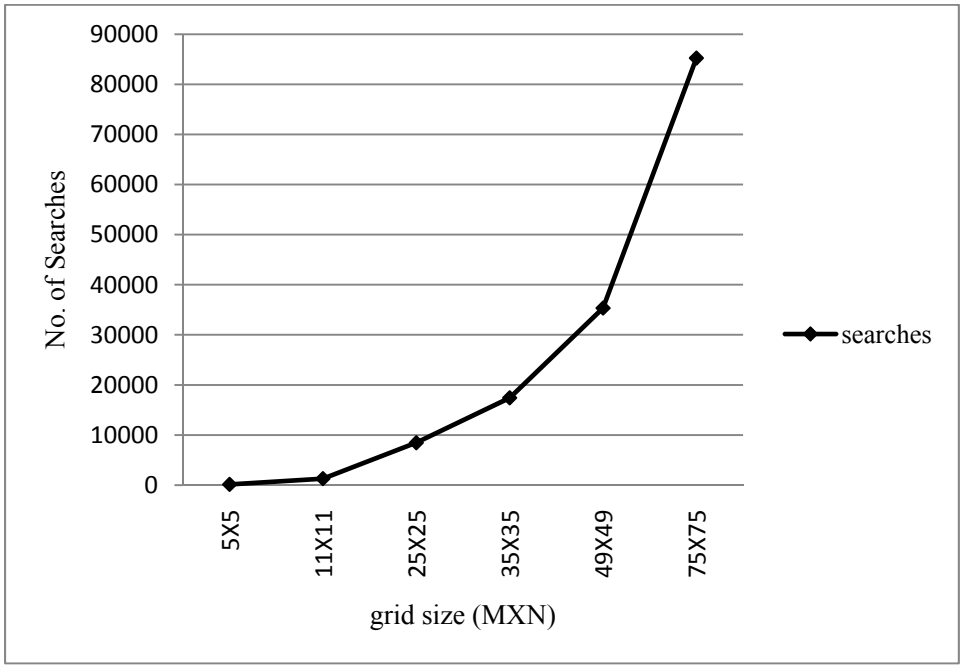
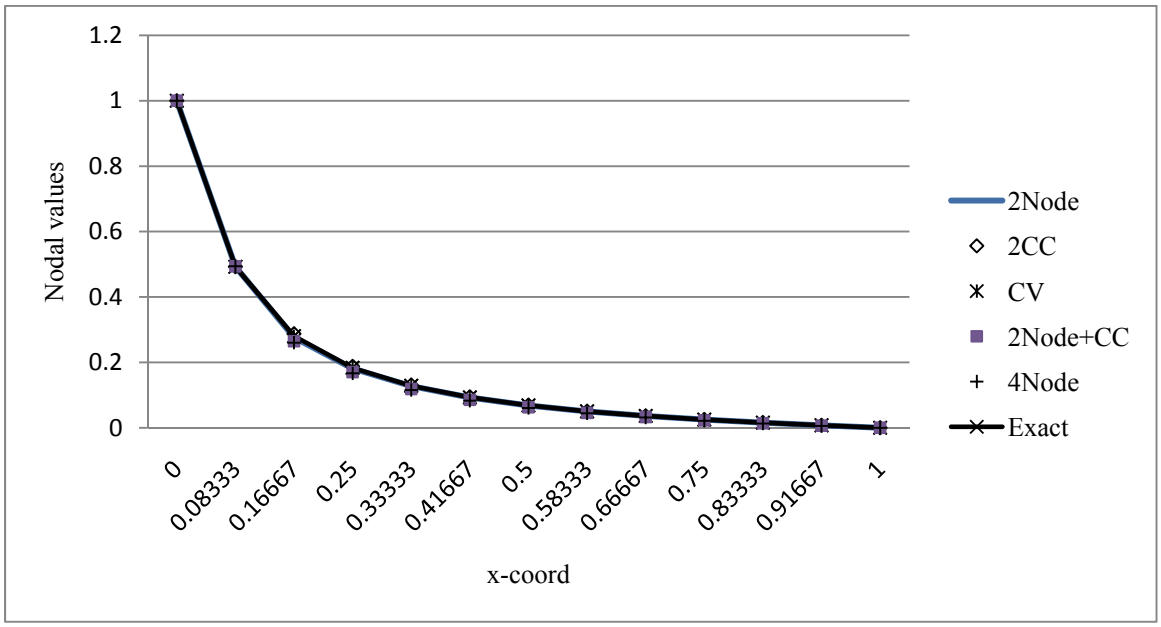


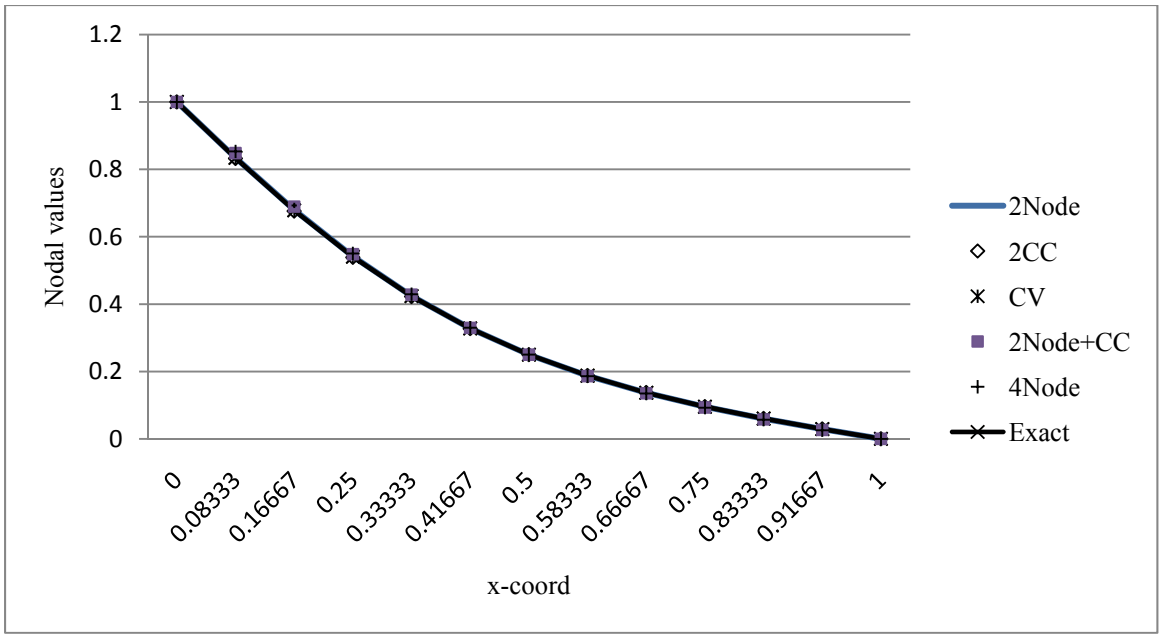
Figure 3.17: Number of searches for one iteration

A grid of 13x13 nodes was selected to compare the results of these five methods for evaluating the intersections values. The boundary value problem described in section 3.2 has been solved using each of these five schemes. For comparison to the exact solution,

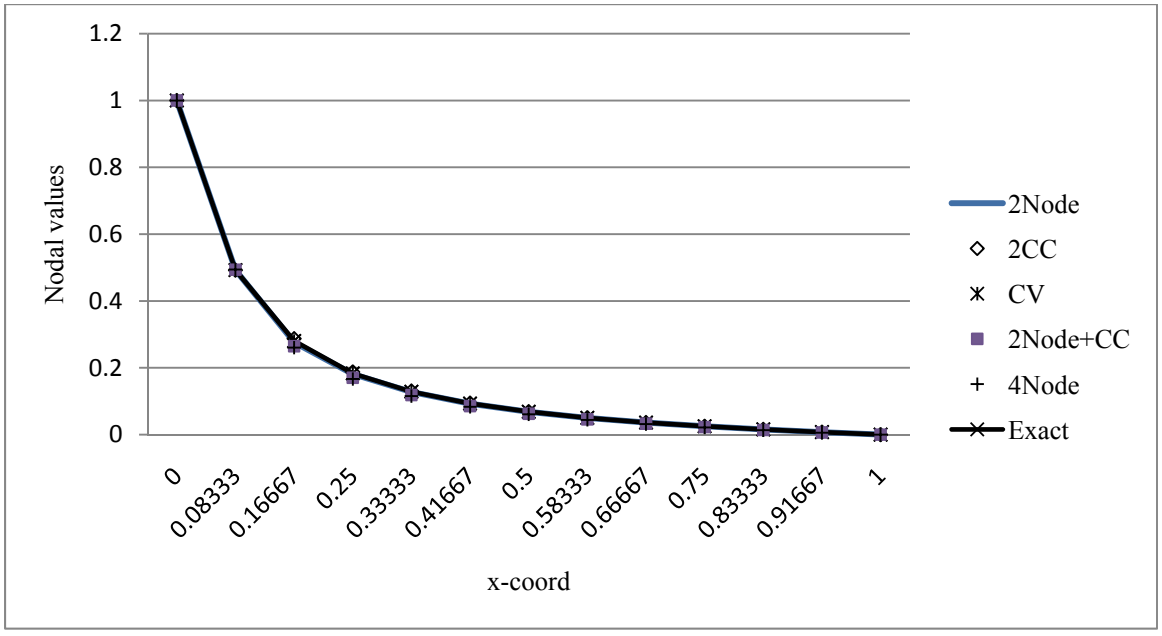
three horizontal rows of grid points (nodes) are selected. These three rows are: the second row of nodes from the lower boundary (y = 0.08333), in the middle of the domain (y = 0.5), and the second row of nodes from the top boundary (y = 0.91667). Property values from the left to the right boundary along these three rakes are shown in Fig. 3.18. For a uniform grid, corresponding to the data presented in Fig. 3.18, these graphs show that all five schemes have good accuracy (compared with the exact solution). However, further simulation tests have shown that, for grid systems with high and low cell aspect ratios, the two cell centres, two nodes and control volume schemes were more accurate than the other two schemes. As for convergence speed, the two cell centres scheme was always the fastest (i.e. converged with less number of iterations) compared with rest of the schemes. For this reason, the two cell centres scheme will be used to calculate the intersection values for all the problems in Chapter IV and Chapter V.



(a)



(b)



(c)

Figure 3.18: Comparison of five methods used to calculate the intersection values for a uniform 13x13 grid. Solution along horizontal lines: (a) y=0.08333, (b) y=0.5, (c) y=0.91667

CHAPTER IV

ANALYSIS OF FURTHER VERIFICATION TESTS

4.1 Introduction

In this chapter the CCFD method is implemented on three additional test problems that have been investigated by other researchers using other numerical methodologies. These three test cases are meant to explore the extendibility of the proposed CCFD scheme to other types of PDEs and boundary conditions. The investigations that have been reported in Chapter III are not all repeated for each problem. However, some aspects will be summarized in tables and/or presented in contour plots.

4.2 Poisson's Equation with Dirichlet Boundaries

Winslow [35] used Poisson's equation to test and explain a new numerical scheme over a triangle mesh. Therefore, to further test the CCFD method, a boundary value problem for the Poisson equation has been selected from [36]. The domain is a 1 unit x 0.5 unit rectangle, subject to a variable Dirichlet condition at the bottom boundary, and constant Dirichlet conditions on the rest of the boundaries. The PDE to be solved is

$$\frac{\partial^2 \emptyset}{\partial x^2} + \frac{\partial^2 \emptyset}{\partial y^2} = -\left(2 + \pi^2 x(1-x)\right) cos\ (\pi y) \tag{4.1}$$

with solution domain and boundary conditions as illustrated in Fig. 4.1. This boundary value problem has an exact analytical solution which is also provided in [36], given by

$$\emptyset(x, y) = x(1-x)\, cos(\pi y) \tag{4.2}$$

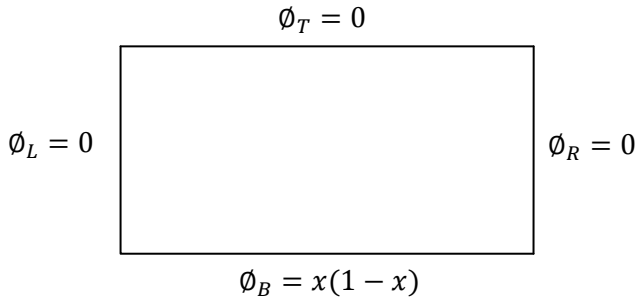


Figure 4.1: Poisson BVP

All the parameters and solution aspects investigated for the first test case in sections 3.2 and 3.3 can also be applied to this case. However, only the effects of grid refinement and relaxation factor are shown in Table 4.1. This problem has an exact analytical solution given by eqn. (4.2) and therefore both CCFD and PSOR (traditional FD method) solutions can be compared to the analytical solution. The last two pairs of columns in Table 4.1, absolute maximum error and relative maximum error, show that for all the tested grids, CCFD is more accurate than the traditional PSOR. As for the number of iterations, with the same convergence criterion, CCFD needs more iterations to converge.

Table 4.1: Effect of different grid sizes and relaxation factors for the Poisson equation test problem

MxN	Num. of Iter.		Omega Opt		AbsMaxErr		RelMaxErr	
	CCFD	PSOR	CCFD	PSOR	CCFD	PSOR	CCFD	PSOR
5x3	12	8	1.575	1.033	0.04930	0.07856	0.4161	0.8562
5x5	17	14	1.825	1.172	0.04822	0.07922	0.4706	1.1841
9x5	23	19	2.075	1.267	0.04248	0.07980	0.4368	1.3560
9x9	54	28	2.200	1.447	0.04297	0.08051	0.4492	1.4653
17x9	67	37	2.450	1.533	0.04111	0.08067	0.4526	1.5987
17x17	193	56	2.200	1.674	0.04155	0.08187	0.4555	1.6282

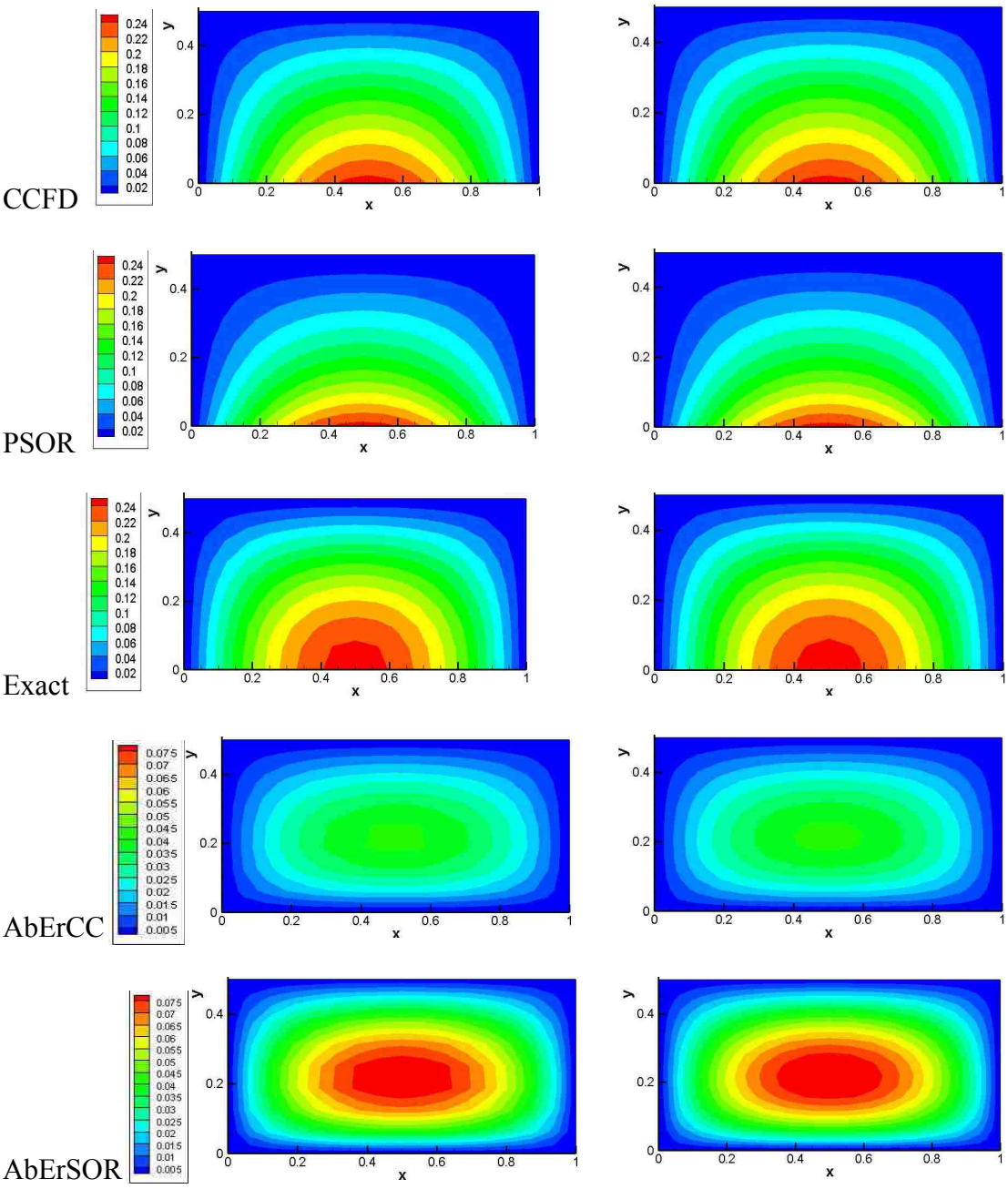


Figure 4.2: CCFD solution, PSOR solution, exact analytical solution, absolute error of CCFD and absolute error of PSOR, (a) 17x9 grid, and (b) 17x17 grid

Solution contours for two grid arrangements, 17x9 and 17x17, are shown in Fig. 4.2. Both grids demonstrate that the CCFD scheme generates solutions with a higher

degree of accuracy than the traditional FD method. This conclusion is typical of other grid arrangements applied to this BVP.

4.3 A Combination of Dirichlet and Neumann Boundaries

A worked example from [37] has been selected to further verify the CCFD scheme. In this example, the governing PDE is the Laplace equation and the solution domain has a derivative (Neumann) condition at the bottom boundary, while the rest of the boundaries are Dirichlet. Since there is no analytical solution available for this particular example, the CCFD solution will be compared with the PSOR (traditional FD) solution. Figure 4.3 shows the problem domain and boundary conditions.

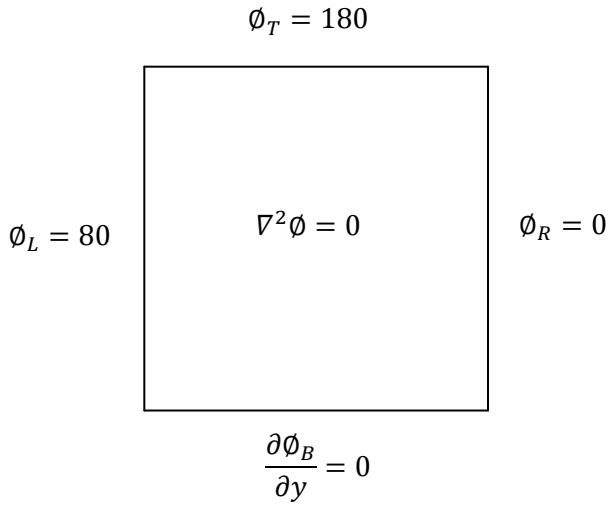


Figure 4.3: Combination of Dirichlet and Neumann boundaries

The domain is a 4 x 4 square. To handle the derivative condition along the lower boundary, a second-order forward differencing has been used to approximate $\emptyset_s$ for the layer of cells attached to the bottom boundary, i.e.,

$$\frac{\partial \emptyset_S}{\partial y} = \frac{3\emptyset_s - 4\emptyset_{cc} + \emptyset_n}{2\left(\frac{y_n - y_s}{2}\right)} \tag{4.3}$$

or $$\emptyset_s = \frac{1}{3}\left(4\emptyset_{cc} - \emptyset_n - \frac{\partial \emptyset_S}{\partial y}(y_n - y_s)\right) \tag{4.4}$$

The new calculated value of $\emptyset_s$ from eqn. (4.4) is used in eqn. (3.8), which approximates $\emptyset_{cc}$ using a second-order central differencing. Recall eqn. (3.8):

$$a_{cc}\emptyset_{cc} = a_w\emptyset_w + a_e\emptyset_e + a_s\emptyset_s + a_n\emptyset_n \tag{4.5}$$

Substituting eqn. (4.4) in (4.5), and re-arranging the equation for $\emptyset_{cc}$, the resulting finite difference equation can be written as,

$$\emptyset_{cc} = \frac{1}{\left(a_{cc} - \frac{4a_s}{3}\right)}\left(a_w\emptyset_w + a_e\emptyset_e + \left(a_n - \frac{a_s}{3}\right)\emptyset_n - \frac{a_s \Delta y \frac{\partial \emptyset_S}{\partial y}}{3}\right) \tag{4.6}$$

where the coefficients a_{cc}, a_w, a_e, a_n and a_s are given by eqn. (3,8). Equation (4.6) is solved for the cells attached to the bottom Neumann boundary. A typical Neumann boundary cell is shown in Fig. 4.4.

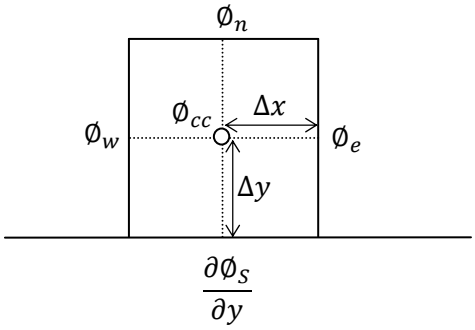


Figure 4.4: Typical cell adjacent to bottom Neumann boundary

The solution procedure for all the interior nodes is exactly the same as explained in section 3.2.1. After the solution is converged, the property value at the Neumann boundary nodes should be evaluated. For this evaluation, three different schemes have

been tested to approximate the nodal values, using two cell centre values that share that node, left-cell east value and the node on top, or simply two nodes inside. All three schemes are shown in Fig. 4.5. The last two methods are second-order forward differencing, while the first uses a weighted averaging from adjacent cell centre values.

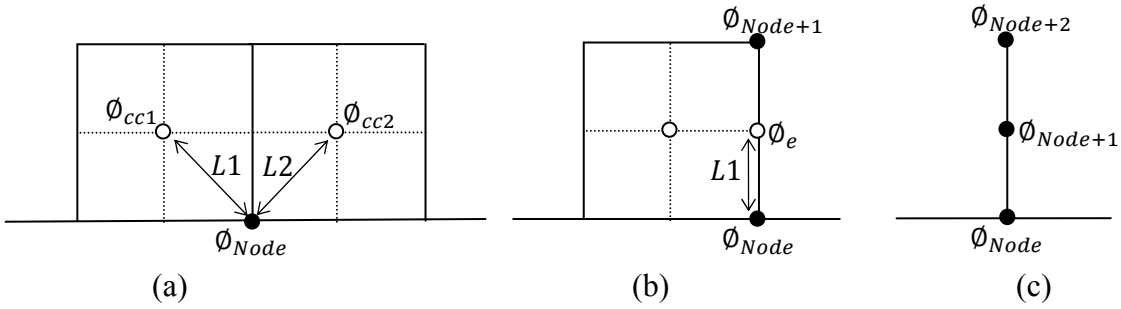


Figure 4.5: Three different methods for calculating Neumann nodes: (a) two cell centre method, (b) 2nd-order forward differencing within the cell, (c) 2nd-order forward differencing with two nodes

To compare these three methods, consider a 5x5 grid. Figure 4.6 shows the three bottom boundary nodes which are calculated with three different approximation schemes, using the converged or updated values for the interior locations. Both the two cell centre and two node methods show close results to the traditional FD formulation solved by the Gauss-Seidel (GS) method. However, the two node method is easier to compute and more logical or closer to the traditional FD scheme. Therefore, it will be used in any subsequent calculations where a Neumann boundary exists.

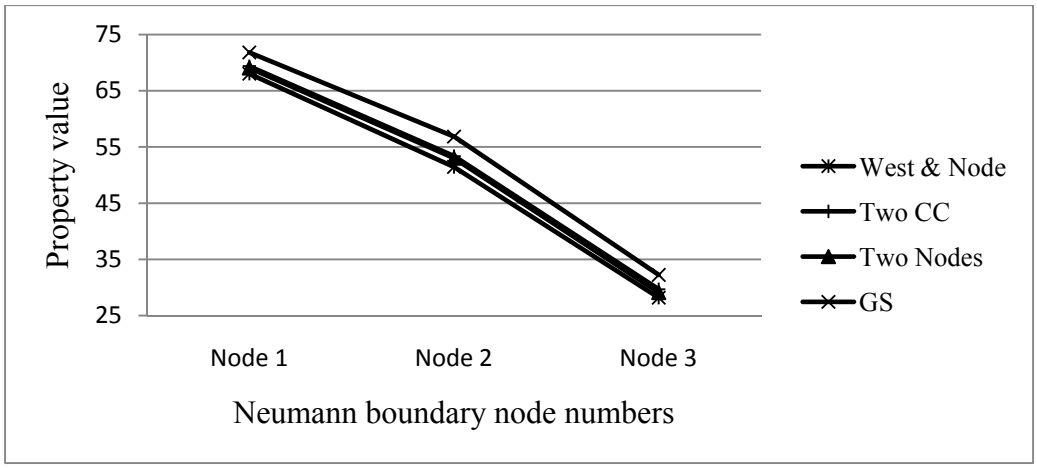
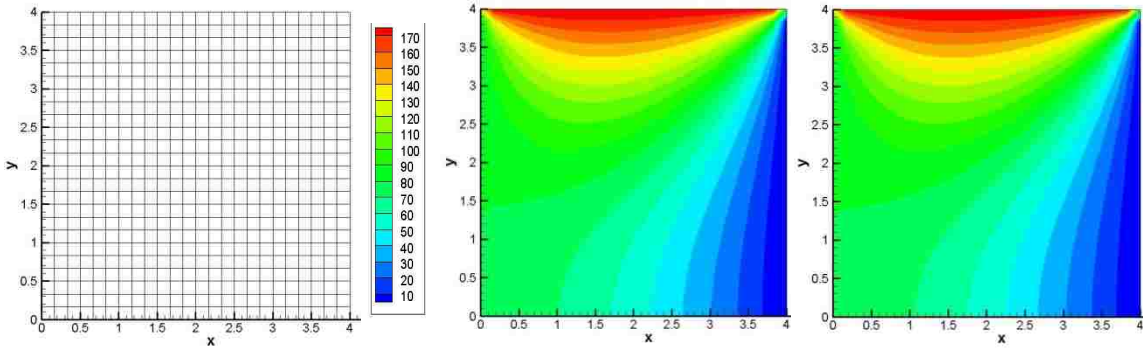


Figure 4.6: Results for three Neumann boundary nodes in a 5x5 grid, comparison of results between GS and three approximation methods

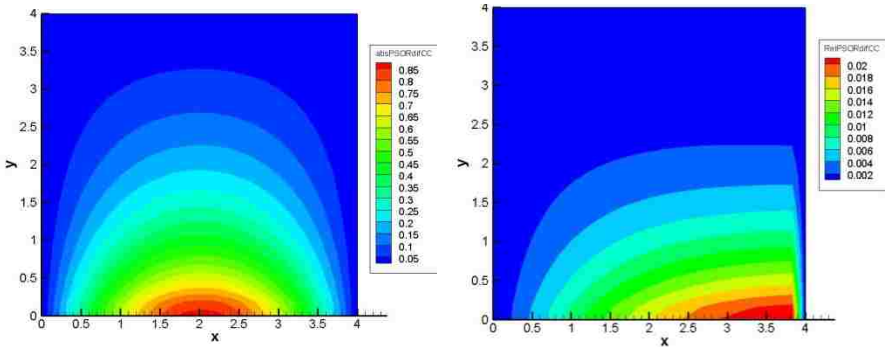
Different grids with uniform rectangular cells have been tested. They all show good agreement between the CCFD results and the traditional FD results. The absolute difference between CCFD and the traditional FD results, averaged by the number of interior nodes, reduces by refining the grid, as shown in Table 4.2. Figure 4.7 presents two grids with their corresponding results.

Table 4.2 : Average absolute difference between CCFD solution and traditional GS solution for different grid sizes

MxN	AveAbsDifCCFD_GS
5x5	0.36909
9x9	0.40683
17x17	0.24465
9x17	0.22978
9x33	0.11958

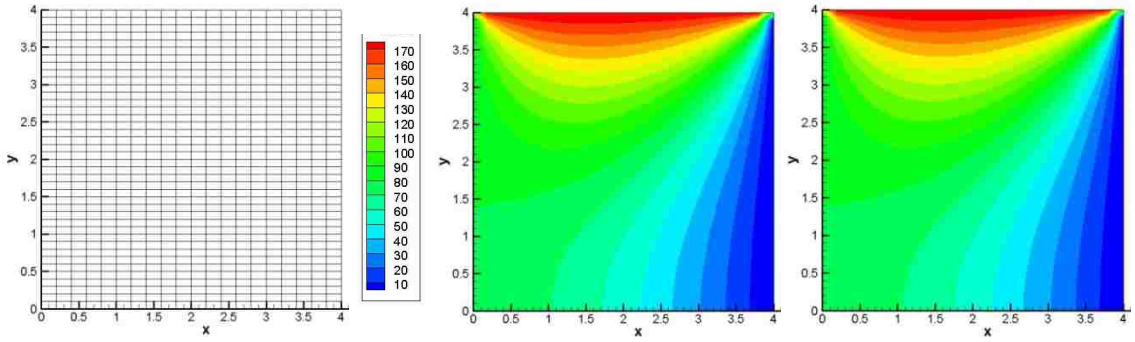


Grid CCFD GS

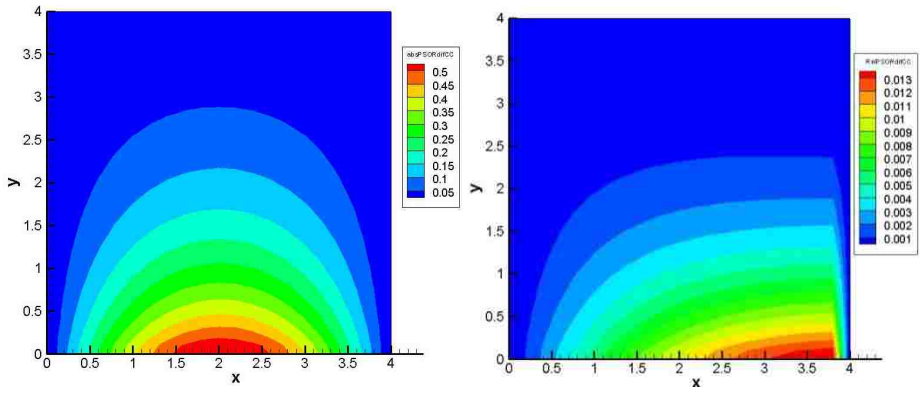


Absolute Diff. CCFD vs. GS Relative Diff. CCFD vs. GS

(a)



Grid CCFD GS



Absolute Diff. CCFD vs. GS Relative Diff. CCFD vs. GS

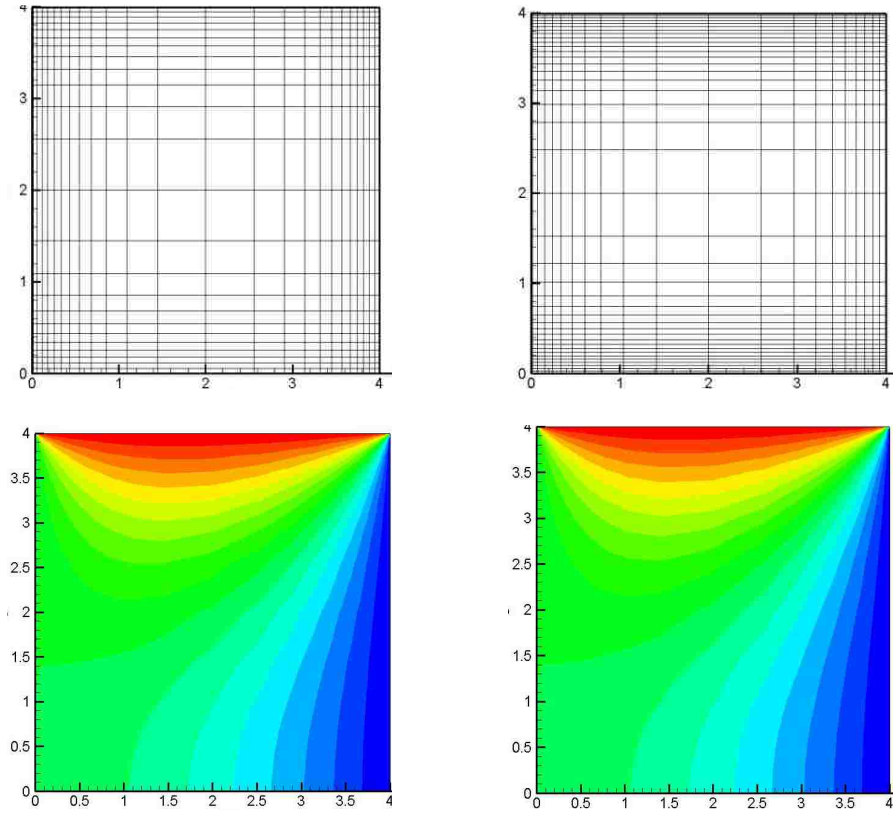
(b)

Figure 4.7: Grid, CCFD solution, GS solution, absolute and relative differences for CCFD vs. GS, (a) 25x25 grid, (b) 21x41 grid

From the two grids shown in Fig. 4.7 it was found that the maximum difference between CCFD and GS decreases from 0.85 for the 25x25 grid to 0.5 for the 21x41 grid. Similarly, the maximum relative difference drops from 0.02 to 0.013. This suggests that clustering the grid, particularly towards the bottom boundary, may improve the solution.

4.3.1 Grid clustering

Two non-uniform grids, 21x41 and 25x25, were selected to study the effect of grid clustering on the final solution, and also to check the limits of the CCFD scheme with highly clustered grids close to Neumann boundaries. Since the Dirichlet boundary values at the left, right and top boundaries are different, clustering grid lines toward all boundaries should yield better results. Figures 4.8a and 4.8b show both grids and the corresponding solution contours. It is important to note that the traditional FD method is not straightforward to apply here, because it requires a grid transformation to a uniformly spaced grid system to maintain the accuracy of the scheme.



(a) (b)

Figure 4.8: Clustered grid toward all boundaries: (a) 25x25 grid and its solution, (b) 21x41 grid and its solution

To check the CCFD bounds with clustering close to only Neumann boundaries, the 21x41 grid is used with a range of different clustering scale factors, i.e. α equal to 0.99, 0.97, 0.95, 0.92, 0.90, 0.87, 0.85, 0.82 and 0.8. The 0.99 factor generates almost a uniform grid, which doesn't show much change in the solution contours. However, for smaller values of the scale factor, like $\alpha = 0.82$, the solution contours tend to have sharper edges in the upper half of the domain, because only four rows (out of 41 rows) of grid points cover that region, i.e. the grid is really coarse in that part of the domain. Figure 4.9 shows the grids and their solution contours for scale factors 0.9 and 0.82.

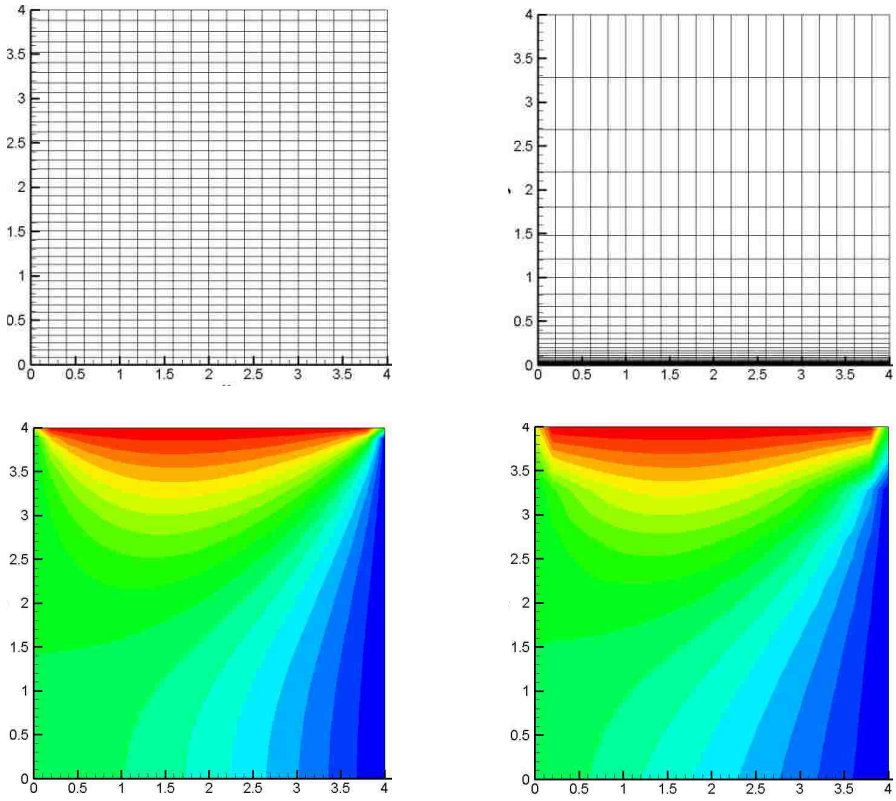


Figure 4.9: Clustered 21x41 grid toward the Neumann boundary: (a) α = 0.99 grid and its solution, (b) α = 0.82 grid and its solution

The computation time increases rapidly with decreasing value of the clustering scale factor. The number of iterations jumps from a couple hundred for α = 0.99 to more than ten thousand iterations for α = 0.82. The conclusion drawn from this numerical experiment is that excessive clustering close to Neumann boundaries does not necessarily improve the final CCFD solution and, in fact, it increases the processing time or may even cause divergence.

4.4 Convection-Diffusion PDE

A boundary value problem from [38] is selected to test the CCFD scheme to simulate a PDE that includes both convection and diffusion terms. The domain is a unit square, subject to Dirichlet boundaries all around. The boundary values are zero, except for the left boundary which has a non-zero parabolic profile. The problem is described by

$$\frac{\partial^2 \emptyset}{\partial x^2} + \frac{\partial^2 \emptyset}{\partial y^2} = P\ cos\theta\ \frac{\partial \emptyset}{\partial x} + P\ sin\theta\ \frac{\partial \emptyset}{\partial y} \tag{4.7}$$

with boundary conditions,

$$\emptyset(x,0) = 0, \qquad \emptyset(x,1) = 0, \qquad 0 \le x \le 1$$

$$\emptyset(0,y) = 4y(1-y), \qquad \emptyset(1,y) = 0, \qquad 0 \le y \le 1$$

This problem represents the convection of $\emptyset$ by a fluid moving with a uniform velocity at an angle θ to the *x*-axis, where $\emptyset$ is also allowed to diffuse with a constant diffusion coefficient. This problem has an exact solution, given by [38]

$$\emptyset(x,y) = e^{P(x\ cos\theta + y\ sind\theta)/2} \sum_{n=1}^{\infty} B_n\ sinh(\sigma_n(1-x))\ sin(n\pi y) \tag{4.8}$$

where $$\sigma_n{}^2 = n^2\pi^2 + P^2/4$$

and $$B_n = \frac{8}{sinh(\sigma_n)} \int_0^1 y(1-1) e^{-(P\ sin\theta\ y)/2} sin(n\pi y)\ dy$$

For this problem, the CCFD method uses second-order accurate central differencing to approximate the diffusion terms $\emptyset_{xx}$ and $\emptyset_{yy}$, and first-order accurate upwind differencing for the convective terms $\emptyset_x$ and $\emptyset_y$. Because the direction of flow is constant, i.e. the convective velocity always has positive components, the convective terms will be backward differenced at all cell centres.

In [38], numerical results calculated by UDS (upwind differencing scheme), CDS (central differencing scheme) and SCHOS (single cell high order scheme) are tabulated in

terms of the maximum absolute difference with the exact solution. For purposes of comparison, CCFD results will be based on the same input values for P, grid size and flow angle. However, to show the capabilities of the current scheme, only a 17x17 grid at two angles, 0 and $\pi/8$, are considered. The results are shown in Table 4.3.

Table 4.3: Comparison of accuracy of CCFD results with other numerical schemes for the convection-diffusion PDE with $P = 40$

θ	UDS	CDS	SCHOS	CCFD
0	0.1604	0.1532	0.01323	0.3681
$\pi/8$	0.2268	0.1286	0.01019	0.41

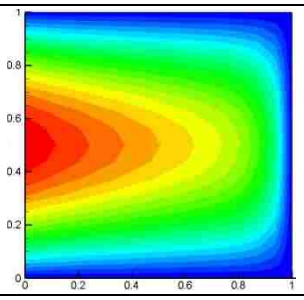
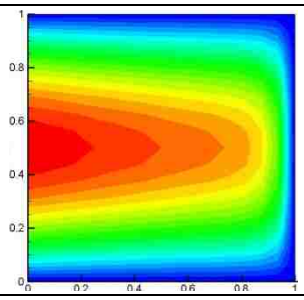
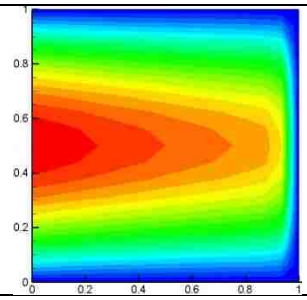
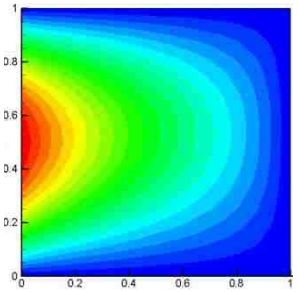
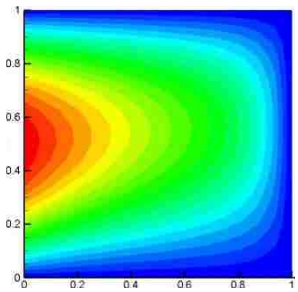
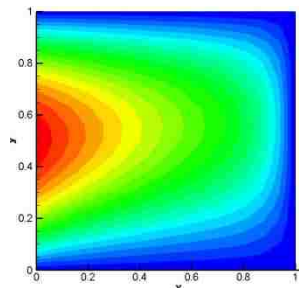
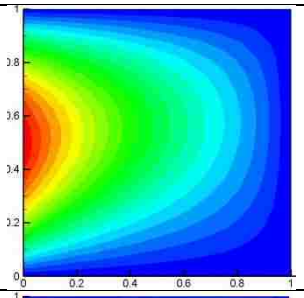
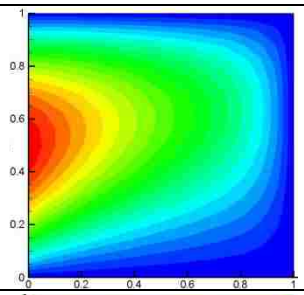
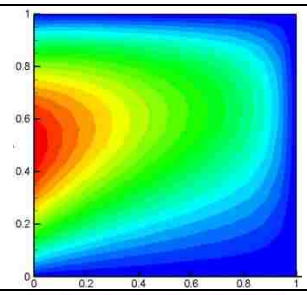
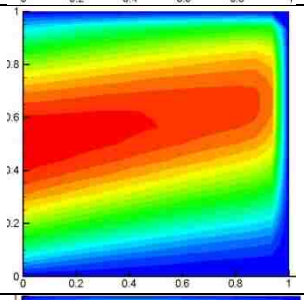
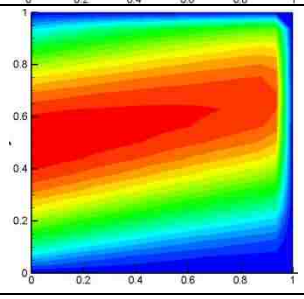
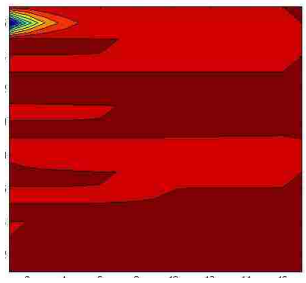
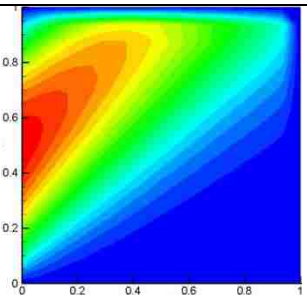
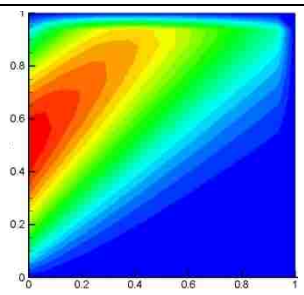
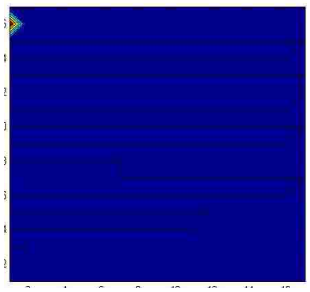
To further verify the CCFD formulation, and since Reference [38] doesn't provide information for other values of P or θ, CCFD solutions will be compared with the traditional FD method and the exact solution. For large angles, such as $\theta = 45^o$, and high P values such as $P = 500$, the exact solution represented by the infinite series (4.8) seems to be not accurate, or even not realistic, as seen in Tables 4.4 and 4.5. This is likely due to convergence difficulties with the infinite series [39]. In this case, the CCFD results should be directly compared with the traditional FD results.

Table 4.4: Absolute maximum difference between CCFD and exact analytical solution

θ	P=10	P=500
0^o	0.209	0.0768
10^o	0.211	exact failed
22.5^o	0.202	exact failed
45^o	0.173	exact failed

As illustrated in Tables 4.4 and 4.5, for high P values, even with small flow angles, the exact analytical equation does not present a correct solution. It should be mentioned that, even with low values of P and θ, the analytical solution assigns non-

Table 4.5: CCFD, GS and analytical solution contours for different values of P and θ

P	θ	CCFD	GS	Exact
40	0^o			
10	10^o			
	22.5^o			
500	10^o			
	45^o			

realistic values to the interior nodes when a large number of terms are taken in the infinite series (eqn. (4.8)). In another words, for each of the evaluated cases, the appropriate number of terms in the series must be determined by trial and error.

4.5 Summary of Results

The examples in this chapter have been chosen as test problems to represent important aspects of the numerical solution of boundary value problems, and determine the capability of the CCFD scheme to deal with these aspects. The results of this investigation provide verification that the CCFD method can simulate diffusion and convection-diffusion phenomena with Dirichlet and Neumann boundary conditions. The lessons learned in this study will serve as guidelines for the simulation of fluid flow considered in the next chapter.

CHAPTER V

INCOMPRESSIBLE FLOW OVER A BACKWARD-FACING STEP

5.1 Introduction

A well-known benchmark CFD problem, the steady incompressible laminar flow over a backward-facing step, will be investigated in this chapter using the CCFD approach. This is an important test case for any numerical scheme since, even though the geometry is simple, the flow physics is complex. In particular, the flow detaches at the sharp corner of the step, forms a large recirculating region behind the step, and then reattaches to the lower wall further downstream. As the flow Reynolds number increases, the reattachment point moves further away from the step and small separation pockets begin to form along the upper wall.

Armaly et al. [40] has conducted experiments on flow over the backward-facing step and shown that the laminar flow transitions to turbulent flow at a Reynolds number of around 1200. He has also demonstrated that, at the higher Reynolds numbers in the laminar regime between 400 and 1200, the adverse pressure gradient along the upper wall is sufficiently strong to promote the development of a secondary recirculation zone attached to the upper wall, which suppresses the growth of the lower eddy. Measurements and numerical predictions begin to deviate from each other at a Reynolds number of about 400. According to Armaly et al. [40], these deviations are due to the fact that the flow is inherently 3-dimensional for Reynolds number greater than 400, while the numerical simulations are all based on 2-dimensional models.

In this chapter, the flow over a backward-facing step is modeled using the vorticity-streamfunction formulation of the steady incompressible Navier-Stokes

equations. This formulation has been used by many researchers to test their numerical algorithms, primarily because it eliminates the problems associated with velocity-pressure coupling and the lack of a physical boundary condition for pressure on the walls.

5.2 The Governing Equations

Mathematically, vorticity is defined as the curl of the velocity field. In fluid mechanics, vorticity is the tendency of fluid elements to spin. Vorticity can be related to the amount of circulation in a fluid or, at a fluid point, it is twice the angular velocity. In vector form it can be written as

$$\vec{\omega} = \nabla \mathrm{X} \vec{V} \tag{5.1}$$

where $\vec{V} = (u, v)$ is the velocity vector.

In a 2-dimensional flow, only the component of vorticity in the direction normal to the plane of flow is non-zero, given by

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{5.2}$$

Now, consider the 2D steady incompressible Navier-Stokes equations in the dimensional, nonconservative form,

continuity eqn. $$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.3}$$

x-momentum eqn. $$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{-1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{5.4}$$

y-momentum eqn. $$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = \frac{-1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{5.5}$$

The pressure term in the momentum equations can be eliminated by cross-differentiation, i.e. differentiate eqn. (5.4) with respect to *y* and eqn. (5.5) with respect to *x*. Subtracting the resulting equations and substituting eqn. (5.2) and (5.3) yields an elliptic PDE known as the *vorticity transport equation*,

vorticity eqn. $$u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \nu\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) \tag{5.6}$$

The details of this derivation can be founded in many CFD textbooks.

For 2D steady flows, the streamfunction ψ can be introduced, which ensures conservation of mass by satisfying the continuity equation. It is especially convenient since it can be used to plot streamlines. The streamlines should hold constant values of streamfunction ψ, and the difference between any two streamfunction values gives the volumetric flow rate between the two streamlines corresponding to those values of the streamfunction. In Cartesian coordinates, the velocity components of an incompressible flow can be written in terms of streamfunction as

$$u = \frac{\partial \psi}{\partial y}\ , \quad v = -\frac{\partial \psi}{\partial x} \tag{5.7}$$

Direct substitution of (5.7) into (5.2) yields

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \tag{5.8}$$

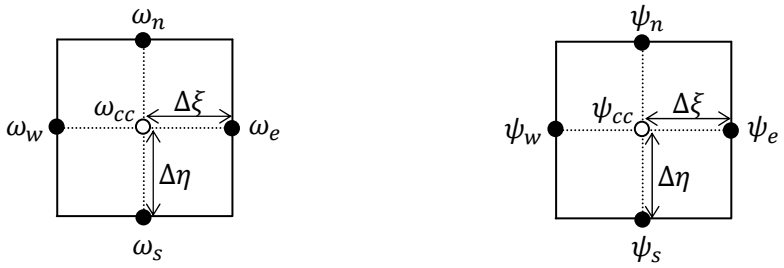
Equation (5.8) is known as the *streamfunction equation* and is classified as an elliptic (Poisson) PDE.

Developers of numerical algorithms for CFD often use the vorticity-streamfunction formulation. The main advantage of this formulation is that the pressure doesn't appear in either equation, which is due to introducing the new dependent

variables ψ and ω. This leads to recasting the steady Navier-Stokes equations into two elliptic equations, which can be solved sequentially by any iterative scheme to provide the velocity field. If the pressure field is desired, then a Poisson equation for pressure can be formulated and solved subsequently. A major drawback of this formulation is the difficulty to extend it to three dimensions. Also, in this formulation, the vorticity at the boundaries needs to be numerically calculated due to the lack of physical boundary conditions on vorticity.

5.3 Discretization of the Governing Equations

Going back to the essence of the CCFD scheme, which is approximation of the PDE at the cell centre, demands the visualization of the differencing stencil in a typical interior cell. For this purpose two model quadrilateral cells are shown in Fig. 5.1, one for vorticity and another for streamfunction. Both cells are in the (ξ, η) coordinates ($\Delta\xi = \Delta\eta = 1$), which takes care of any arbitrary cell shape, as explained in the first section of Chapter III. Therefore, the discretized FD equations will be valid for any cell within the physical domain, regardless of cell type or location.



(a) (b)

Figure 5.1: Differencing stencil for (a) vorticity, (b) streamfunction

First, the velocity components u and v need to be evaluated at the cell centre. Using eqn. (5.7) along with eqn. (3.3) (chain rule) and eqn. (3.7) (second-order central differencing for the derivatives) yields,

$$u_{cc} \approx \frac{1}{2y'}(\psi_n - \psi_s)$$
$$v_{cc} \approx -\frac{1}{2x'}(\psi_e - \psi_w) \tag{5.9}$$

where x' and y' are geometrical quantities given by eqn. (3.4).

For the vorticity transport equation (5.6), the convection terms $\frac{\partial \omega}{\partial x}$ and $\frac{\partial \omega}{\partial y}$ are approximated with upwind differencing. That is, the differencing is either backward or forward, based on the direction of the flow, i.e. the signs of the velocity components u and v. An upwind scheme simulates the physics of the problem more properly, because it uses a solution-sensitive FD stencil. Numerically, upwind schemes add artificial viscosity to the discretized PDE, which damps out oscillations and makes the solution more stable. In this study, first-order upwinding is employed. After mapping the convection derivatives to the (ξ, η) coordinates, one can combine both backward and forward approximations into a single upwinding equation [10]. For example,

$$\frac{\partial \omega}{\partial \xi} \approx (1 - \epsilon_x)\frac{\omega_e - \omega_{cc}}{2\Delta\xi} + (1 + \epsilon_x)\frac{\omega_{cc} - \omega_w}{2\Delta\xi} \tag{5.10}$$

where the value of ϵ_x is based on the sign of u-velocity. If u is positive, ϵ_x is set to 1 and backward differencing is employed. If u is negative, ϵ_x is set to -1 and forward differencing is employed. This means that eqn. (5.10) maintains the selective direction of differencing. A similar expression can written for $\frac{\partial \omega}{\partial \eta}$. Collecting the results thus far, the convection terms in the discretized form of eqn. (5.6) can be expressed as

$$u\frac{\partial \omega}{\partial x} \approx \frac{u_{cc}}{2x'}\left[(1-\epsilon_x)(\omega_e - \omega_{cc)} + (1+\epsilon_x)(\omega_{cc} - \omega_w)\right]$$
$$v\frac{\partial \omega}{\partial y} \approx \frac{v_{cc}}{2y'}\left[(1-\epsilon_y)(\omega_n - \omega_{cc)} + (1+\epsilon_y)(\omega_{cc} - \omega_s)\right] \tag{5.11}$$

The diffusion terms in the vorticity transport equation (5.6), ie. $\nabla^2\omega$, are discretized similar to the Laplace equation (3.5). The resultant finite difference equation for the vorticity can be written similar to eqn. (3.8). However, the coefficients now include both convection and diffusion terms. After simplification, the vorticity transport equation can be approximated as,

$$a_{ccVor}\omega_{cc} = a_{wVor}\omega_w + a_{eVor}\omega_e + a_{sVor}\omega_s + a_{nVor}\omega_n$$

where

$$a_{ccVor} = \frac{1}{4}\left(\frac{u_{cc}\epsilon_x}{x'} + \frac{v_{cc}\epsilon_y}{y'}\right) + \nu\, a_{cc}$$
$$a_{wVor} = \frac{1}{8}\left(\frac{u_{cc}(1+\epsilon_x)}{2x'}\right) + \nu\, a_w, \quad a_{eVor} = -\frac{1}{8}\left(\frac{u_{cc}(1-\epsilon_x)}{2x'}\right) + \nu\, a_e \tag{5.12}$$
$$a_{sVor} = \frac{1}{8}\left(\frac{v_{cc}(1+\epsilon_y)}{2y'}\right) + \nu\, a_s, \quad a_{nVor} = -\frac{1}{8}\left(\frac{v_{cc}(1-\epsilon_y)}{2y'}\right) + \nu\, a_n$$

and the coefficients a_w, a_e, a_s and a_n are given by eqn. (3.8).

Discretization of the streamfunction equation (5.8) is quite similar to the derivations in eqns. (3.5) through (3.8), except that the right hand side is $-\omega_{cc}$ instead of zero. Therefore, eqn. (3.8) can be recast for the streamfunction as,

$$a_{ccStr}\psi_{cc} = a_{wStr}\psi_w + a_{eStr}\psi_e + a_{sStr}\psi_s + a_{nStr}\psi_n - \left(\frac{-\omega_{cc}}{8}\right) \tag{5.13}$$

$$a_{ccStr} = \frac{1}{(x_e - x_w)^2} + \frac{1}{(y_n - y_s)^2}$$

$$a_{wStr} = \frac{x_e - x_{cc}}{(x_e - x_w)^3}, \quad a_{eStr} = \frac{x_{cc} - x_w}{(x_e - x_w)^3}$$

$$a_{sStr} = \frac{y_n - y_{cc}}{(y_n - y_s)^3}, \quad a_{nStr} = \frac{y_{cc} - y_s}{(y_n - y_s)^3}$$

5.4 Problem Geometry and Boundary Conditions

A typical backward-facing step is shown in Fig. 5.2.

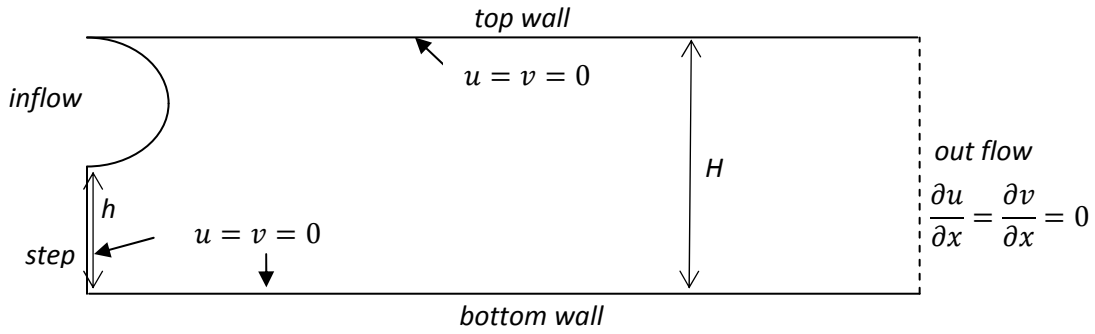


Figure 5.2: Backward-facing step domain and boundary conditions on primitive variables

Boundary conditions for the velocity components *u* and *v*:

- inflow: parabolic flow, $u = u(y), v = 0$.
- outflow: fully developed with Neumann boundary condition.
- step, bottom wall and top wall: no-slip stationary wall.

At the inlet, the volumetric flow rate can be defined as

$$\dot{Q} = (H - h)u_{ave} \tag{5.14}$$

where *H* and *h* are the height of the channel and the step respectively, as illustrated in Fig. 5.2, and u_{ave} is the average inlet *u*-velocity. The flow rate can be written in terms of Reynolds number as

$$\dot{Q} = \nu\, Re \tag{5.15}$$

From eqn. (5.14) and (5.15),

$$u_{ave} = \frac{\nu\, Re}{(H-h)} \tag{5.16}$$

This u_{ave} can be used in the fully developed parabolic profile at the inlet, which can then be written as

$$u_{inlet} = \frac{6\, u_{ave}}{(H-h)^2}(y-h)(H-y) \tag{5.17}$$

where y is the y-coordinate of any point at the inlet. If the channel length is long enough downstream, then the flow should return to the fully developed profile, given by,

$$u_{outlet} = \frac{6\,(H-h)u_{ave}}{H^3}\, y\,(H-y) \tag{5.18}$$

5.4.1 Boundary conditions for streamfunction

For the streamfunction at the inlet, $v = 0$ and $u = \psi_y$, or,

$$\psi_y = \frac{6\, u_{ave}}{(H-h)^2}(y-h)(H-y) \tag{5.19}$$

Integrating (5.19) from h to y yields,

$$\psi_{inlet} = \psi_h + \frac{6\, u_{ave}}{(H-h)^2}\left[-hHy + \frac{1}{2}(H+h)y^2 - \frac{y^3}{3}\right] + \frac{u_{ave}h^2(3H-h)}{(H-h)^2} \tag{5.20}$$

where ψ_h is the streamfunction value along the step and the bottom wall, and can be assigned any value. For the top wall, ψ_H needs to be calculated. This can be done by integrating (5.19) from h to H, which reduces eqn. (5.20) to

$$\psi_H = \psi_h + (H-h)u_{ave} \tag{5.21}$$

If we take $\psi_h = 0$, $H = 1$ and $h = H/2$, then,

$$\psi_H = 0.5u_{ave} \tag{5.22}$$

$$\psi_{inlet} = u_{ave}(2.5 - 12y + 18y^2 - 8y^3) \tag{5.23}$$

For Re = 50, the *u*-velocity and streamfunction profiles at the inlet are shown in Fig. 5.3.

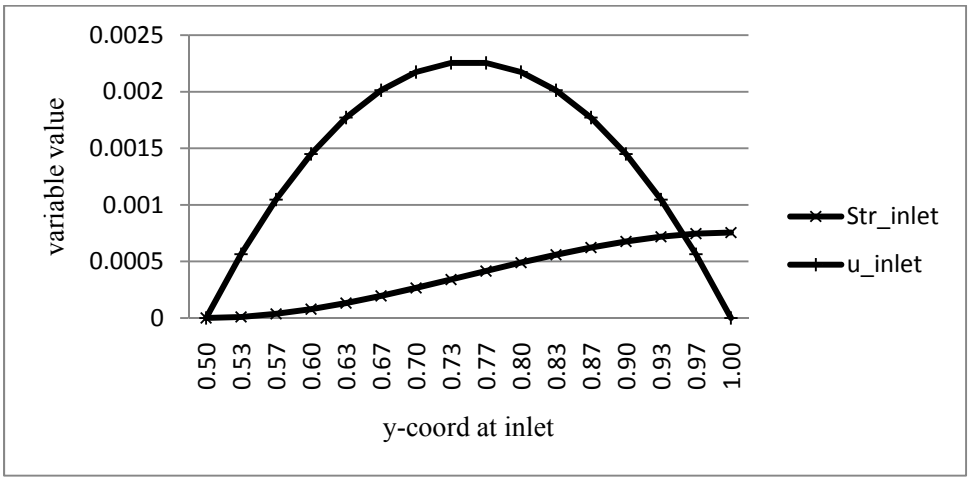


Figure 5.3: Velocity and streamfunction inlet profiles for Re = 50

At the outlet, since it is a Neumann boundary, the streamfunction value is numerically calculated using backward differencing to the inner nodes and/or cells. Figure 5.4 shows a summary of the streamfunction boundary conditions.

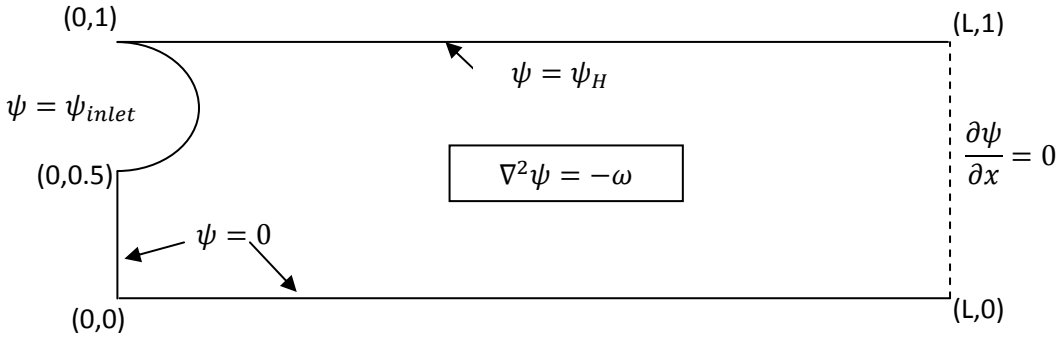


Figure 5.4: Streamfunction boundary conditions

5.4.2 Boundary conditions for vorticity

The vorticity is numerically evaluated at all boundaries by using vorticity definition, given by eqn. (5.2). At the inlet, u_y can be evaluated by differentiating the exact velocity profile at the inlet, eqn. (5.18), and v_x can be approximated using a first-order forward differencing to the inner cell centre or node. Since the CCFD scheme uses intersection points at the boundaries rather than the physical boundary nodes, therefore vorticity at the cell centres adjacent to the inlet will be used to calculate v_x for the inlet and the step. For H = 1 and h = 0.5, the resulting vorticity at the inlet for the west intersections can be written as,

$$\omega_{inlet} = 12u_{ave}(-3 + 4y) + \frac{v_{cc}}{\Delta x} \tag{5.24}$$

Recall that all walls are no-slip stationary walls, i.e. $u = v = 0$ on the walls. Along the step, u_y = 0, therefore eqn. (5.2) reduces to $\omega = v_x$, which can be handled similar to the inlet case, i.e.,

$$\omega_{step} = \frac{v_{cc}}{\Delta x} \tag{5.25}$$

For the top and bottom walls, v_x = 0, and $\omega = -u_y$. Then, for the north (top wall) and south (bottom wall) intersections, the vorticity is approximated by the cell-centre value of the cell adjacent to the boundary,

$$\omega_{top} = \frac{u_{cc}}{\Delta y}, \qquad \omega_{bottom} = -\frac{u_{cc}}{\Delta y} \tag{5.26}$$

At the outlet Neumann boundary, the vorticity is numerically calculated using backward differencing to the inner nodes and/or cells. Figure 5.5 shows a summary of the vorticity boundary conditions.

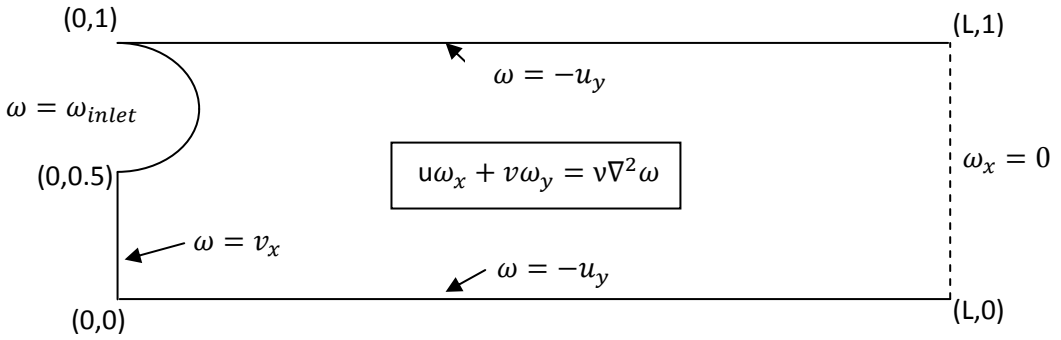


Figure 5.5: Vorticity boundary conditions

5.5 Solution Procedure and Results

The solution procedure starts with initializing ψ and ω at all cell centres. Then, for an individual cell:

a) compute ψ and ω at the intersections *e, w, n, s*.

b) compute u_{cc} and v_{cc} from eqn. (5.9).

c) compute ω_{cc} from eqn. (5.12).

d) compute ψ_{cc} from eqn. (5.13).

The above steps are repeated for all the cells around each node, and the nodal value is updated using eqn. (3.9). The same procedures are performed for all the interior nodes, completing the first iteration. Then the whole process is repeated until the convergence criterion is satisfied.

Results for Re = 50, 100 and 200 are compared with the experimental results of Armaly et al. [40], and the numerical results of Barton [41] and Zogheib and Barron [42]. All grids are structured with quadrilateral cells, with $H = 1$ and $h = 0.5$. The length of the channel (L) is extended with higher Reynolds numbers to ensure a fully developed profile. CCFD results are based on a relative difference $((\psi_{new} - \psi_{old})/\psi_{new}) < 10^{-6}$.

One of the key parameters to compare results, which is reported by both numerical and physical experimentalists, is the reattachment length, shown by X_r in Fig. 5.6.

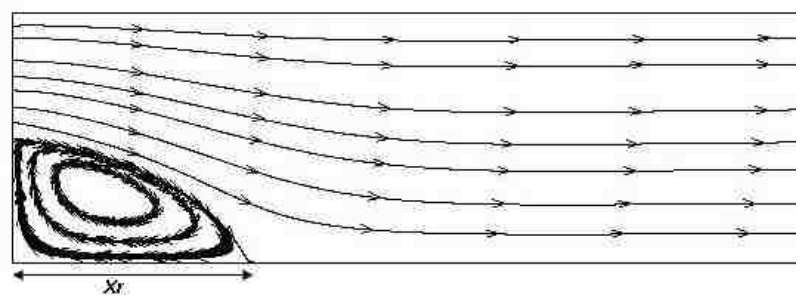


Figure 5.6: Reattachment length X_r

Table 5.1 and Fig. 5.7 show the results of the CCFD scheme along with other numerical and experimental results. The comparison of CCFD accuracy with other numerical schemes, for example Zogheib and Barron FOU[42] (which uses first-order upwind), and the experiments reported by Armaly et al. [40], demonstrate that the current scheme performs as well as other numerical schemes. Table 5.2 shows the ratio of reattachment length of these numerical schemes to the experimental lengths reported by Armaly et al. [40].

Table 5.1: Reattachment lengths for CCFD scheme compared to experimental data and other numerical methods

Re	50	100	200
Armaly et al. [40]	1.16	1.7	2.6
Barton[41]		1.8	2.6
Zogheib & Barron (FOU) [42]	0.99	1.6	2.5
Zogheib & Barron (SOU) [42]	1.02	1.7	2.7
CCFD	0.93	1.5	2.4

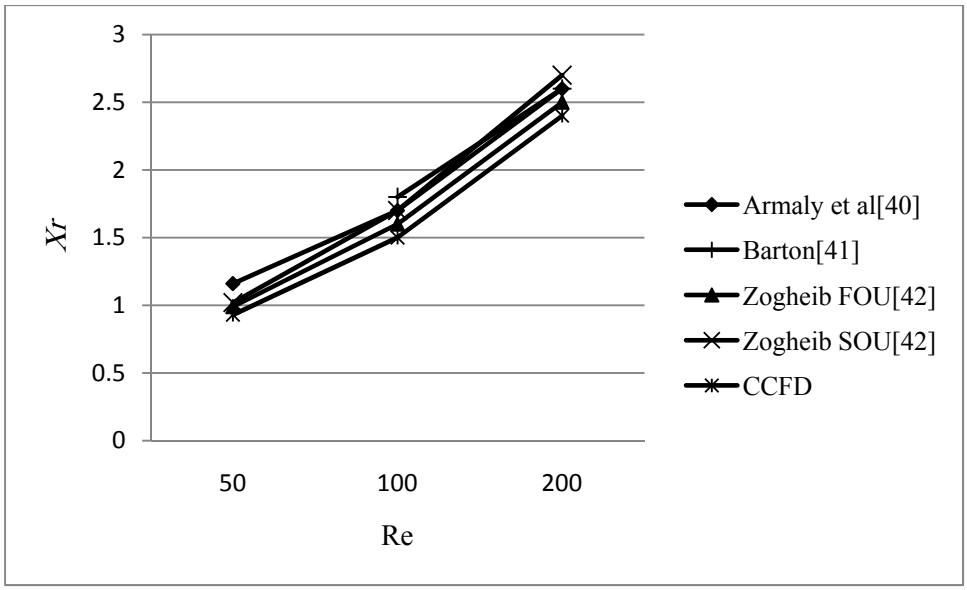


Figure 5.7: Reattachment length X_r as a function of Reynolds number

Table 5.2: Comparing the accuracy of the CCFD scheme

Re	50	100	200
Zogheib & Barron (FOU) [42]	0.8534	0.9412	0.9615
CCFD	0.8017	0.8824	0.9231

To confirm that the results improve with finer grids, a series of mesh refinements were performed for Re = 50. As illustrated in Fig. 5.8, after several refinements X_r improves slower or is less affected by further refinements. In another words, the CCFD scheme tends to become grid independent after a certain degree of numerical accuracy.

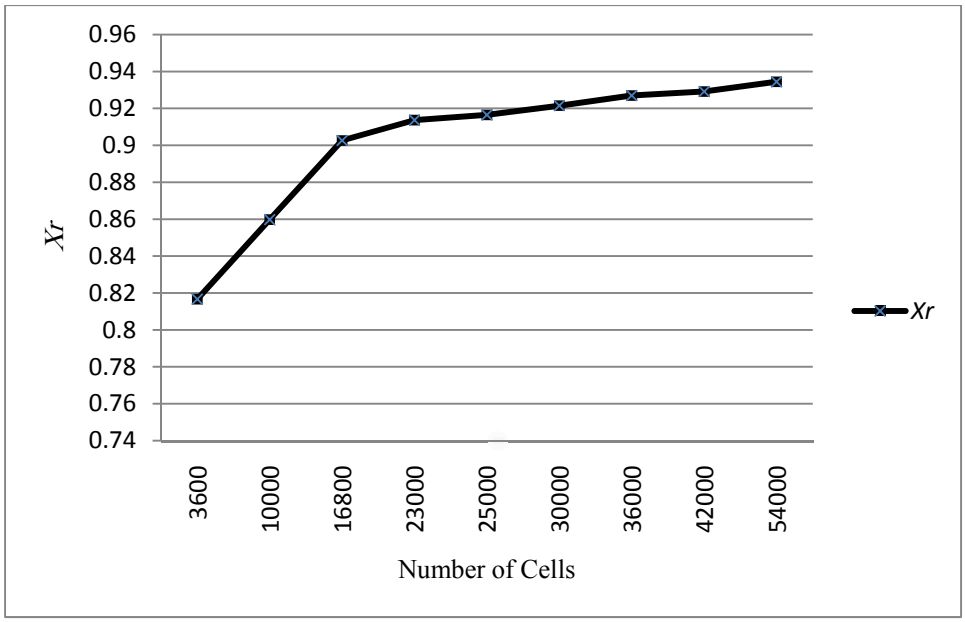
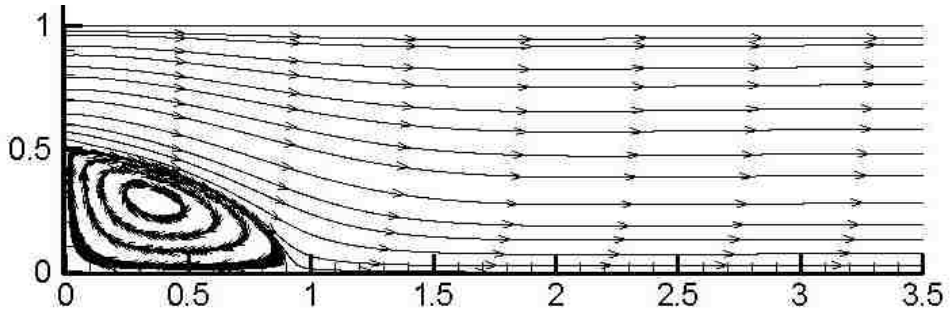


Figure 5.8: CCFD grid sensitivity for Re = 50

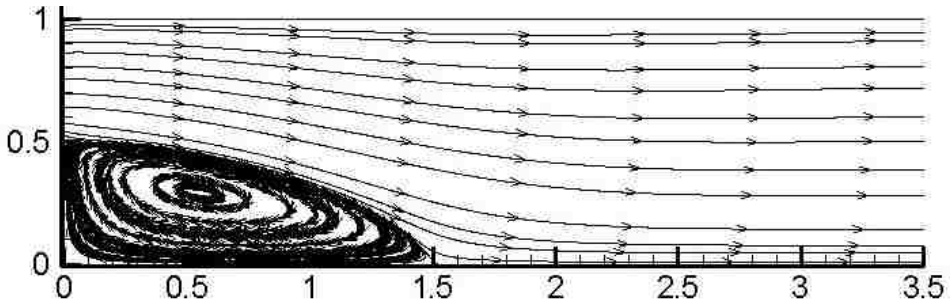
Table 5.3: Details of the grid refinement cases simulated by CCFD

Re	MxN	Length	num. cell /1000	X_r
50	121x31	3.75	3.6	0.817
50	201x51	3.75	10	0.860
50	211x81	3.75	16.8	0.903
50	231x101	3.75	23	0.914
50	251x101	3.75	25	0.917
50	251x121	3.75	30	0.922
50	301x121	3.75	36	0.927
50	351x121	3.75	42	0.929
50	451x121	3.75	54	0.934
100	121x31	5	3.6	1.249
100	211x81	5	16.8	1.440
100	251x101	5	25	1.468
100	451x101	5	45	1.498
200	151x31	7.5	4.5	1.918
200	501x101	7.5	50	2.414

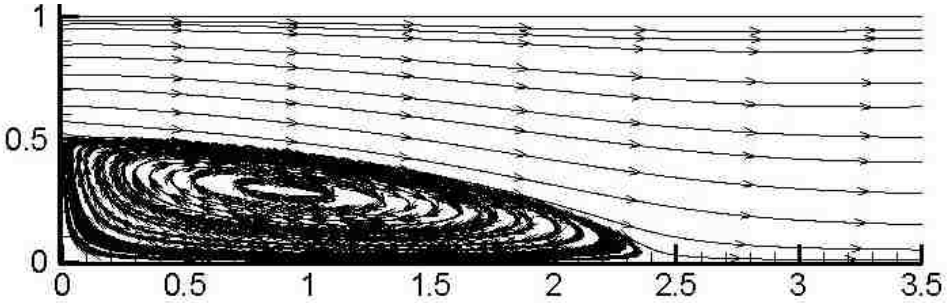
Table 5.3 includes reattachment lengths for the tested grid sizes and channel lengths for Re = 50, 100 and 200. The CCFD results listed in Table 5.1 are highlighted for reference (Re = 50, 100, 200). Further grid refinement is necessary for Re = 300 and 400 to capture the X_r more accurately.



(a)



(b)



(c)

Figure 5.9: Streamlines for: (a) Re = 50, (b) Re = 100, (c) Re = 200

The domain lengths listed in Table 5.3 are determined based on the ratio of the computed maximum u-velocity at the outlet to the exact maximum velocity at the outlet, which is calculated from eqn. (5.18). By ensuring this ratio close to one, the computational domain can be considered long enough to have fully developed outflow. For example, for Re = 100 and 200 these ratios were 0.9925 and 0.9996 respectively. Figure 5.9 shows the streamlines for each Reynolds number.

5.6 Summary

A system of coupled, nonlinear PDEs, modeling incompressible steady flow over a backward-facing step, has been solved with the CCFD method. The simulation results for low Reynolds numbers are in good agreement with the other numerical schemes and/or experimental results. It is expected that finer grids will capture the flow field more accurately for higher Reynolds number.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

A Cell-Centred Finite Difference (CCFD) method has been developed and initially applied to solve a model elliptic equation to investigate various important aspects of any new CFD algorithm. By comparing the CCFD results with analytical solution and/or traditional FD schemes, the proposed methodology shows good behaviour with grid refinement (even for high/low cell aspect ratios). Also, more accurate results are captured by clustering grid lines at regions where high gradients are expected. However, intensive clustering close to Neumann boundaries is not desirable due to the increase in computational time without significant accuracy improvement. Similar to most of the CFD codes, the CCFD method can be accelerated by introducing a relaxation factor, which should be larger with finer grids. The reported bounds of relaxation are based on numerical experiments. To verify these bounds, a stability analysis should be carried out. Five schemes are explored for the evaluation of intersection points. The two cell centres scheme was more reliable for all tested grid arrangements, besides being the fastest.

Four different cases have been tested, which are all single PDEs. Comparison between the available analytical solution and the CCFD approach demonstrates the validity of this method.

A final challenging test for the CCFD method was to solve a benchmark fluid flow problem, the 2D steady incompressible laminar flow over a backward-facing step. This problem involves extension of the CCFD algorithm to solve a system of coupled nonlinear PDE's. The preliminary simulation results confirm that the current

methodology is as good as equivalent numerical schemes (with similar degrees of accuracy).

6.2 Current Code Capabilities and Programming Issues

A code has been written in C/C++ language for the CCFD approach. The current version is able to read grid files with (.dat) or (.txt) extension. It is also able to build a structured grid (in a uniform domain) made of rectangular cells, refine the grid and/or cluster it in both *x*-direction and *y*-direction. The code can handle Neumann boundaries and Dirichlet boundaries, both constant and variable.

Programming issues can be divided into two parts: first are those related to storage requirements, and second are those related to the core of the geometry section of the CCFD scheme. Both parts are directly related to each other. The CCFD scheme uses a unique sorting of grid information, e.g. finding the cells that share each node and finding each pair of cells that share each face. This information is plugged instantly into the calling function from their allocation address in computer memory rather than being searched for through all the memory. However, this facilitating feature is accompanied with quick memory increase when the grid information is stored all together in a couple of multi-dimensional arrays, e.g. Array[365][24][60][60] needs 4Gb of memory. A neat programming architecture can break this array into three 2D arrays (Array[ ][ ]) that may do the same job of the 4D array and use just 200Mb of memory. Also, by constructing subroutines that will do specific calculations whenever needed, the memory requirement can be reduced. Inside these subroutines the required variables and arrays are declared to execute the appropriate equations for the specific problem being considered. This feature is not recommended in C/C++ reference textbooks, since it may lead to the use of one

variable in two different and simultaneous tasks, while the programmer has forgotten that this will overwrite the original values of these variables, causing a fatal or non-fatal logic error. Hence, in order to develop a good quality code and a good performance code, one of these goals is achieved at the expense of the other. With limited memory resources and the increasing number of calculation points (i.e. internal nodes), the programmer needs to decide his/her programming goals to do the job efficiently.

6.3 Future Work

This new method is applied in boundary cells in exactly the same manner as in interior cells and therefore there are no difficulties with the differencing formulas close to the boundaries. To extend this research, a more robust formulation can be implemented on unstructured grids, which should be rather straightforward. Since the FD approximation of the governing equation is totally constrained within the cell, differencing formulas with higher order accuracy can easily be applied. This method should also be implemented with the primitive variable formulation of the Navier-Stokes equations. Further extension of the methodology to 3-dimensional fluid flows and unsteady flows is needed.

The current version of the computer code (i.e. the C/C++ code) can also be improved in many aspects, such as parallel processing, building a graphical user interface (GUI), use of macros, …, etc.

REFERENCES

1. Murman, E.M. and Cole, J.D. Calculation of Plane Steady Transonic Flows. *AIAA Journal.* 9(1):114-121, 1971.

2. Thompson, J.F. *Numerical Grid Generation*. (ed.) North-Holland, New York, NY, 1982. (Also published as Vols. 10 and 11, *Applied Mathematics and Computation*, 1982).

3. Thompson, J.F. Grid Generation Techniques in Computational Fluid Dynamics. *AIAA Journal*. 22(11):1505-1523, 1984.

4. Knupp, P. and Steinberg, S. *Fundamentals of Grid Generation*. CRC Press, Boca Raton, FL, 1994.

5. Peric, M., Kessler, R. and Scheuerer, G. Comparison of Finite-Volume Numerical Methods with Staggered and Co-located Grids. *Computers and Fluids*. 16(4):389-403, 1988.

6. Diskin, B., Thomas, J.L., Nielsen, E.J., Nishikawa, H. and White, J.A. Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations. Part I: Viscous Fluxes. 47th *AIAA Aerospace Sciences Meeting,* AIAA 2009-597, Orlando, FL, 2009.

7. Peiro, J. and Sherwin, S. Finite Difference, Finite Element and Finite Volume Methods for Partial Differential Equations. In *Handbook of Materials Modeling. Volume I: Methods and Models.* Springer. 2415-2447, 2005.

8. Mattiussi, C. An Analysis of Finite Volume, Finite Element, and Finite Difference Methods Using Some Concepts from Algebraic Topology. *Journal of Computational Physics*. 133:289-309, 1997.

9. Onata, E. and Idelsohn, S.R. A Comparison Between Finite Element and Finite Volume Methods in CFD. *Computational Fluid Dynamics*. 1:93-100, 1992.

10. Hoffmann, K.A. and Chiang, S.T. *Computational Fluid Dynamics, Vol. 1*. 4th ed. Engineering Education System, 2000.

11. Hoffmann, K.A. and Chiang, S.T. *Computational Fluid Dynamics, Vol. 2*. 4th ed. Engineering Education System, 2000.

12. Smith, G.D. *Numerical Solution of Partial Differential Equations: Finite Difference Methods.* 3rd ed. Clarendon Press, Oxford, Oxfordshire, New York, 1985.

13. Gottlieb, D. and Orzag, S.A. *Numerical Analysis of Spectral Methods: Theory and Applications.* SIAM, PA, 1977.

14. Saleh, J.M. *Fluid Flow Handbook*. McGraw Hill Handbooks. NY, 1962.

15. Antoci, C., Gallati, M. and Sibilla, S. Numerical Simulation of Fluid-Structure Interaction by SPH. *Computers and Structures*. 85:879-890, 2007.

16. Patankar. S.V. *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1980.

17. Barron, R.M. and Zogheib, B. A Finite Difference Calculation Procedure for the Navier-Stokes Equations on a Staggered Curvilinear Grid. *World Academy of Science, Engineering and Technology*. 849-852, 2010.

18. Schneider, G.E. and Raw, M.J. Control Volume Finite-Element Method for Heat Transfer and Fluid Flow Using Co-located Variables: 1. Computational Procedure. *Numerical Heat Transfer*. 11:363-390, 1987.

19. Rhie, C.M. and Chow, W.L. Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation. *AIAA J*ournal 21(11):1525-1532, 1983.

20. Reggio, M. and Camarero, R. Numerical Solution Procedure for Viscous Incompressible Flows. *Numerical Heat Transfer*. 1:131-146, 1986.

21. Thiart, G.D. Finite Difference Scheme for the Numerical Solution of Fluid Flow and Heat Transfer Problems on Non-Staggered Grids. *Numerical Heat Transfer*. 17:43-62, 1990.

22. Demirdzic, I., Lilek, Z. and Peric, M. A Collocated Finite Volume Method for Predicting Flows at All Speeds. *International Journal for Numerical Methods in Fluids*. 16(12):1029-1050, 1993.

23. Versteeg, H.K. and Malalasekera, M. *An Introduction to CFD: The Finite Volume Method*. 2nd ed. Pearson Education Limited, 2007.

24. Majumdar, S. Role of Under Relaxation in Momentum Interpolation for Calculation of Flow with Non-Staggered Grids. *Numerical Heat Transfer*. 13:125-132, 1988.

25. Livne, E., Dessart, L., Burrows, A. and Meakin, C.A. A Two-Dimensional Magnetohydrodynamics Scheme for General Unstructured Grids. *Astrophysical Journal Supplement*. Vol. 170, 2007.

26. Stone, J.M. and Mihalas, D. Upwind Monotonic Interpolation Methods for the Solution of the Time Dependent Radiative Transfer Equation. *Journal of Computational Physics*. 100(2):402-408, 1992.

27. Gardiner, T.A. and Stone, J.M. An Unsplit Godunov Method for Ideal MHD via Constrained Transport. *Journal of Computational Physics*. 205(2):509-539, 2005.

28. Mignone, A., Tzeferacos, P. and Bodo, G. High-order Conservative Finite Difference GLM–MHD Schemes for Cell-Centered MHD. *Journal of Computational Physics*. 229(17):5896-5920 2010.

29. Spekreijse, S.P. Multigrid Solutions of Monotone Second-Order Discretization of Hyperbolic Conservation Laws. *Mathematics of Computation*. 49(179):135-155, 1987.

30. Bathe, K.J. and Ledezma, G.A. Benchmark Problems for Incompressible Fluid Flows with Structural Interactions. *Computers and Structures*. 85(11-14):628-644, 2007.

31. Oosterlee, C.W., Wesseling, P., Segal, A. and Brakkee, E. Benchmark Solutions for the Incompressible Navier-Stokes Equations in General Co-ordinates on Staggered Grids. *International Journal for Numerical Methods in Fluids*. 17(4):301-321, 1993.

32. He, P., Salcudean, M., Gartshore, I.S. and Nowak, P. Multigrid Calculation of Fluid Flows in Complex Geometries using Curvilinear Grids. *Computers and Fluids*. 25(4):395-419, 1996.

33. Choi, S.K., Nam, H.Y. and Cho, M. A Comparison of Higher-Order Bounded Convection Schemes. *Computer Methods in Applied Mechanics and Engineering*. 121(1-4):281-301, 1995.

34. Sakai, K. and Watabe, D. A Third-Order Scheme for Numerical Fluxes to Guarantee Non-Negative Coefficients for Advection-Diffusion Equations. *American Journal of Computational Mathematics*. 1(1):26-38, 2011.

35. Winslow, A.M. Numerical Solution of the Quasilinear Poisson Equation in a Non-uniform Triangle Mesh. *Journal of Computational Physics*. 135, Article CP975698:128-138, 1997.

36. Bradie, B.A. *A Friendly Introduction to Numerical Analysis*. Pearson Prentice Hall, NJ, p: 737-739, 2006.

37. Mathews, J.H. and Fink, K.D. *Numerical Methods Using Matlab*. Pearson Prentice Hall, NJ, p: 568-579, 2004.

38. Gupta, M.M., Manohar, R.P. and Stephenson, J.W. A Single Cell High Order Scheme for the Convection-Diffusion Equation with Variable Coefficients. *International Journal for Numerical Methods in Fluids*. 4:641-651, 1984.

39. Stubley, G.D., Private communication. University of Waterloo, Canada, June, 2011.

40. Armaly, B.F., Durst, F., Pereira, J.C.F., Schonung, B. Experimental and Theoretical Investigation of Backward-Facing Step Flow. *Journal of Fluid Mechanics.* 127:473–496, 1983.

41. Barton, I.E. A Numerical Study of Flow over Confined Backward-Facing Step. *International Journal for Numerical Methods in Fluids*. 21:653–665, 1995.

42. Zogheib, B. and Barron, R.M. Velocity–Pressure Coupling in Finite Difference Formulations for the Navier–Stokes Equations. *International Journal for Numerical Methods in Fluids*. 65:1096-1114, 2011.

VITA AUCTORIS

Ali Salih was born in 1983 in Kurdistan region, Iraq. He graduated from Barzani High School in 2000. From there he went to Salahaddin University where he obtained the B.Sc. in Mechanical Engineering in 2004. He then worked for local and international firms for five year before coming to Canada in 2009. Currently he is a candidate for the Masters of Applied Science degree in Mechanical, Automotive and Materials Engineering at the University of Windsor and hopes to graduate in Spring 2011.