Vertex-Based Discretisation Methods for Thermo - Fluid Flow in a Finite Volume-Unstructured Mesh Context

Diane McBride

Centre for Numerical Modelling and Process Analysis
School of Computing and Mathematical Sciences
the University of Greenwich
London

A thesis submitted in partial fulfillment of the requirements of the University of Greenwich for the Degree of Doctor of Philosophy

July 2003

Abstract

The main aim of this research project is to investigate techniques to improve the resolution of flow variables on unstructured skewed meshes whilst working within a Finite Volume (FV) context. A three-dimensional vertex-based FV algorithm for the solution of thermo - fluid flow problems has been developed and integrated within a multi-physics FV framework PHYSICA. Currently PHYSICA employs a cell-centred discretisation technique for fluid mechanics problems and a vertex-based discretisation technique for solid mechanics problems. The vertex-based discretisation approach is validated for a variety of heat transfer problems and comparisons are made with cell-centred solutions. A coupled thermo-mechanical problem, including solidification and radiation, is simulated using vertex-based and cell-centred techniques. Results, run-time and memory requirements are compared.

Hybrid vertex-based/cell-centred discretisation of the hydrodynamic variables is also investigated. The components of velocity are solved vertex-based with pressure cell-centred or conversely pressure is solved vertex-based with velocity cell-centred. The methods are applied to flow in a lid-driven cavity and solutions are obtained on a number of distorted meshes. Comparisons are made with the benchmark solutions. The hybrid discretisation enables solutions on distorted meshes where purely cell-centred techniques fail. The hybrid methods produce final solutions containing errors due to mesh distortion.

The co-located vertex-based flow solutions obtained on the distorted meshes are comparable to solutions obtained on a uniform Cartesian mesh. Having a good resolution of the flow field on distorted meshes enables the solution of other transported variables using cell-centred techniques. Finally, this hybrid vertex-based/cell-centred technique is applied to thermally driven flow, turbulent flow, and three-dimensional flow over an aircraft wing.

Acknowlegements

I would like to thank my supervisors Prof. Mark Cross and Dr. Nick Croft for their invaluable support throughout this research project. It is their encouragement, constructive advice and helpful discussions that have made completion of this research project possible. I would also like to acknowledge the advice given by Dr. Nick Croft through his extensive knowledge of PHYSICA.

Thanks go to my research colleagues for their friendship, suggestions and enjoyable working environment. Thanks also to the academic and technical staff of the school for their support.

This research would not have been possible without the financial support of the Engineering and Physical Sciences Research Council and the University of Greenwich.

Especial thanks to my family, my husband Conrad and sons Jamie and Simon for their patience and understanding throughout my academic studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational Fluid Dynamics (CFD) modelling of 'real-life' processes often requires solutions in complex three-dimensional geometries, which can result in meshes, aspects of which are badly distorted. A numerical solution technique is required that accurately models the physical process on a mesh that matches the true geometry of the physical domain. Employing an unstructured mesh allows physical processes with increasingly complex geometries to be modelled. Significant advances have been made in the development of numerical methods designed to yield accurate solutions on unstructured meshes. Various solution methods have been tried but the two major strategies that have emerged involve Finite Volume (FV) and Finite Element (FE) approaches. The FV principles of applying conservation laws locally to control volumes lend themselves to easy physical interpretation. It is these physically meaningful concepts that have lead to the FV method being the preferred method for the simulation of flow processes and is used in most of the commercially available CFD codes, e.g. PHEONICS, FLUENT, FLOW-3D, STAR-CD, CFX. There are a number of FV approaches, although the most widely used is the cell-centred approach. This technique is computationally efficient, uses simple approximations to discretise the terms in the transport equation and has low memory requirements and fast simulation times. It handles orthogonal unstructured meshes well but is not robust on a non-orthogonal grid. The FE method, for complex flow problems, has increased in popularity over recent years and is now being used in commercial CFD codes, e.g. FIDAP. A vertex-based FV technique that utilises the flexibility of FE, element based piecewise linear shape functions, and the local conservation principles of the FV method allows skewed meshes to be handled with ease. However, this method is computationally far more expensive than cell-centred techniques.

At the Centre of Numerical Modelling and Process Analysis, University of Greenwich, a 3D unstructured FV framework for multi-physics modelling [1] (PHYSICA) has been developed. PHYSICA currently employs a cell-centred Eulerian discretisation technique for fluid dynamics problems and a vertex-based Lagrangian discretisation technique for solid mechanics problems. The meshes employed can be constructed of a variety of elements, hexahedral, pentahedral or tetrahedral. Problems are encountered when solving cell-centred on distorted non-orthogonal meshes. Fitting a highly orthogonal mesh to a complex geometry can be one of the most time consuming aspects of the modelling process. Coupling between different physical phenomena, such as flow and stress problems, can result in mesh distortion occurring during the solution process. The cell-centred approach, though fast and efficient on orthogonal meshes, requires the addition of correction terms to account for mesh skewness. These correction terms can introduce errors into the solution process and lead to difficulties with convergence, [2]. As a result the CFD group at the University of Greenwich is interested in identifying a discretisation technique that handles non-orthogonality in a mesh and can be efficiently incorporated within the multi-physics environment. Is there a method that can capture the efficiency of the cell-centred method, whilst retaining the geometric flexibility of a vertex-based discretisation procedure?

The overall aim of this research project is to investigate techniques to improve the resolution of flow variables on unstructured skewed meshes whilst working within a Finite Volume (FV) context. As part of this investigation a vertex-based Eulerian discretisation technique, utilising element based piecewise linear shape functions, is reviewed and compared with the cell-centred discretisation method. A three-dimensional vertex-based FV algorithm for the solution of the general transport equation has been developed and integrated within PHYSICA. The vertex-based method is further extended to the solution of flow, employing the revised SIMPLER solution algorithm. Hybrid vertex-based/cell-centred solution methods are investigated and a method is sought that enables solutions on non-orthogonal distorted meshes whilst utilising the computational efficiency of the cell-centred technique.

1.1 Overview of Finite Volume Methods

The finite volume method (FVM) was originally developed as a special class of the finite difference formulation, but has since become the most common and preferred approach for CFD analysis. The FVM was first introduced by McDonald [3] in 1971 for the solution of two dimensional time dependent Euler equations, and was extended to three-dimensional flows in the early seventies by Patankar and Spalding [4].

In the finite volume method the solution domain is divided into a number of finite control volumes. The conservation equation is integrated in physical space over the control volume and a solution sought which makes each of these integrals equal to zero. The resulting expression maintains exact conservation of the relevant properties for each finite cell volume. The finite volume method traditionally uses finite difference type approximations to discretise the transport equation and can be applied to cell-centred, cell-vertex or vertex-centred control volumes. The control volume and location of the solved variable for the different discretisation methods is shown in Figure 1.1

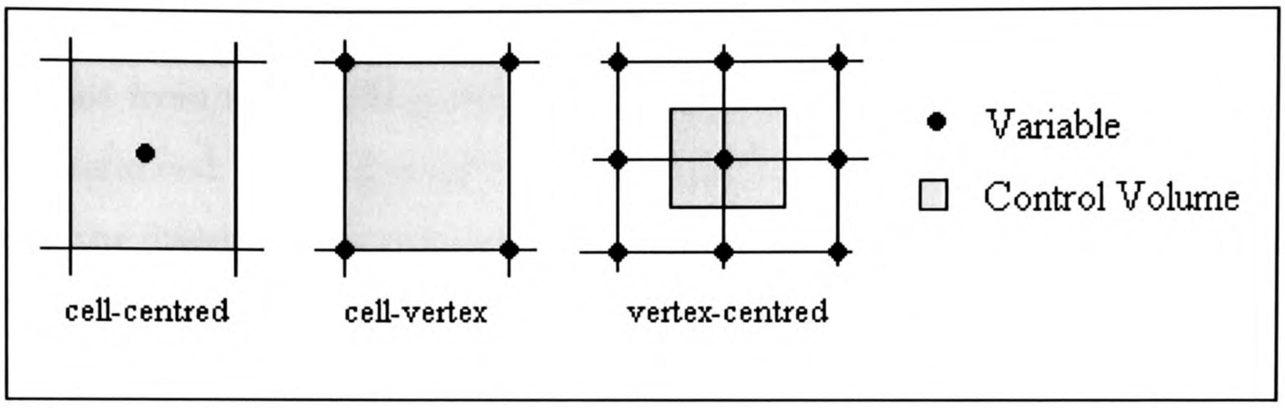


Figure 1.1: Control volumes and location of solved variable

◇ Cell-Centred

The control volume is associated with the mesh element. The governing equation is integrated over the mesh cell and the unknown variable is solved and stored at the cell centre.

The cell-centred finite volume method was developed in the early 1970's for structured meshes by Patankar and Spalding [4, 5]. A staggered grid was employed to solve the velocity components at the cell faces. A standard work for the cell-centred finite volume approach to heat transfer and fluid flow is given by Patankar [6]. The method was extended to unstructured meshes in the 1980's. The components of velocity and pressure were co-located at the cell centre [7, 8].

◇ Cell-Vertex

The control volume is again associated with the mesh cell. The governing equation is integrated over the mesh cell but in contrast to the cell-centred method the unknown variable is solved and stored at the cell vertices.

The cell-vertex finite volume method describes a variety of techniques where the variable is stored at the vertices of the mesh cell. The transport equations are integrated, either as surface or volume integrals, for the element cell and distributed among the vertices. The conservation equation for a mesh vertex includes contributions from surrounding element cells giving a system of equations that are over-determined. Having more equations than unknown's means that the task of solving the discretised equations is more complex than systems with equal numbers of equations and unknowns. A general overview is given by Hirsh [9]. McDonald [3] applied this technique to flow on a two-dimensional structured mesh in the early 1970's.

◇ Vertex-Centred

The vertex-centred finite volume methods can be viewed as particular cases of the cell-vertex finite volume method producing a system of algebraic equations with an equal number of unknowns and equations. Vertex-centred methods can have overlapping control volumes consisting of all surrounding element cells with a vertex at a common node. Fluxes are estimated across element cell faces from the face corner values. Internal cell face fluxes cancel so only external faces of the element cells contribute to the flux balance of the vertex control volume.

Non-overlapping control volume methods, as shown in Figure 1.1, construct a dual cell centred on the mesh vertex. The dual cell is constructed by dividing each element cell between its vertices, normally the element is divided by lines connecting the mid-face and element centre. In contrast to the cell vertex method the fluxes are integrated along the perimeter of the dual control volume. Consistent expressions can then be applied across adjacent control volume faces.

1.1.1 Development of the Finite Volume Method

The cell-centred discretisation method has remained the most preferred finite volume technique. Using simple finite difference type approximations to discretise the various terms in the governing equations, combined with comparatively low storage requirements, have made this approach favoured by many of the finite volume commercial codes. The earliest finite volume approach used fully orthogonal Cartesian meshes and employed a staggered grid for the velocity components. The idea is to evaluate scalar variables, i.e. pressure, temperature, on the ordinary Cartesian mesh, but to evaluate velocity components on staggered grids centred on the control volume faces. Harlow and Welsh [12, 13] first introduced grid staggering in 1965 for finite difference methods. Patankar and Spalding [4, 6] applied staggered grid arrangements to structured finite volume methods in the 1970's. Grid staggering was introduced as a means of overcoming spurious oscillating pressure and velocity fields that can be encountered when employing equal order co-located methods. Difficulties encountered with equal-order co-located methods are a consequence of the fact that only gradients of pressure appear in the momentum equations. If pressure is interpolated linearly, only pressure differences between alternate solution points are involved in the overall system of equations and a checkerboard pressure field would give the same zero pressure gradient as a uniform pressure field. In the mid-1970's the FVM was being successfully applied to complex flow phenomena, but was limited to fully orthogonal structured meshes [4, 5, 10, 11]. The method was fast and efficient, but lacked the ability to model physically realistic domains.

In an attempt to fit structured meshes to real life geometries, cell blocking, curvilinear grids and multi-block techniques were employed. To sculpt structured meshes to the physical geometry, elements can be fully or partially blocked [14]. The use of Body Fitted Co-ordinates [15] to [20], can enable solutions when a strict Cartesian mesh is unsuitable. The solution grid and associated transport equations are mapped onto their topologically equivalent Cartesian mesh. This approach considerably increases the storage requirements and adds to the complexity of the discretisation process. Errors accumulate during the mapping process and become increasingly significant if the physical domain is substantially different from the Body Fitted Mesh [21]. Solving realistic geometries on a Cartesian mesh can be further improved by the use of multi-block techniques [22]. The solution domain is divided into a number of blocks, each block being mapped onto its equivalent Cartesian mesh. Multi-block techniques allow quite complex geometries to be modelled, external surfaces can be represented accurately but internal complexity still presents problems due to the lack of flexibility of using different element types.

In the early 1980's the Finite Volume technique was extended to unstructured meshes for a number of CFD applications. Cell-centred techniques were employed by Pan *et al* [23] to solve laminar flow problems using an unstructured mesh of triangular elements. Chow [24] employed polygonal elements, including triangles, squares and octagons, to solve simple flow and heat transfer problems. Simple flow problems were solved by many authors, including [25, 26]. Swirling, turbulent, reacting flows have been solved by Croft *et al* [1, 2] and many others. Unstructured techniques permitting mixed element types allow complex internal and external domains to be represented accurately. Research has focused on non-staggered techniques as staggered grid arrangements do not extend easily to unstructured meshes, difficulties are encountered when defining staggered velocity and pressure control volumes since the grid lines are generally not parallel to the co-ordinate lines. Staggering also requires a large increase in storage of geometry information. Specialized interpolation schemes were developed for co-located non-staggered arrangements. Several momentum interpolation methods were proposed in the early 1980's, with small variations, by Hsu [7], Rhie and Chow [8] Peric [27] and Burns [28]. In general, co-located formulations prevent checker-boarding by expressing the face velocity in terms of adjacent velocity and pressure values. An additional term often referred to as 'added dissipation' is added to the face velocity. The most widely used scheme is that of Rhie and Chow, employed by nearly all the recent unstructured grid methods [1, 29, 30, 31]. Other procedures investigated include:

- Reggio and Camarero [32], who adopted forward differencing for the mass flow rate and backward differencing for the pressure gradient.

- Shih and Tan [33] used a penalty formulation with several grids and control volume locations.
- Majumdar [34] examined the role of under relaxation of momentum interpolation.
- Chen [35] investigated a partially staggered grid system using cell-centred and vertex-centred techniques and a non-staggered system using simple averaging as a means of suppressing the pressure oscillations.

The success of the Cartesian staggered mesh method of Harlow and Welsh [36], which does not suffer from spurious pressure modes, has lead to an interest in developing unstructured staggered methods. Hall *et al* [37] and Nicholaides [38] independently generalised the staggered mesh method of Harlow and Welsh to unstructured triangular meshes. This method is based on the theory that every unstructured mesh has a local orthogonal dual mesh associated with it, Figure 1.2. The components of velocity share the same set of dual control volumes, unlike the structured-staggered method, where each of the components of velocity have a different set of control volumes associated with them. The momentum equations are integrated over the vertex-centred control volume and velocities are solved on the control volume face. Velocities are now located at the element face. Pressure is solved cell-centred and mass is conserved over the element cell. These techniques are being investigated by a number of authors including Perot *et al* ,[39] to [43], using tetrahedral meshes and Hyman *et al*[44, 45] using quadrilateral elements.

Several authors, [46] to [47], have investigated the use of unstructured vertex techniques. Vertex-centred methods provide better discretisation on complex geometries as nodes/vertices can easily be positioned at critical locations. They also have an advantage in that, unknown quantities are resolved directly at boundary locations. However, conflicts can occur at the vertex of more than one boundary. The construction of vertex-centred control volumes requires extra computational effort and greater storage requirements making it computationally far more expensive than

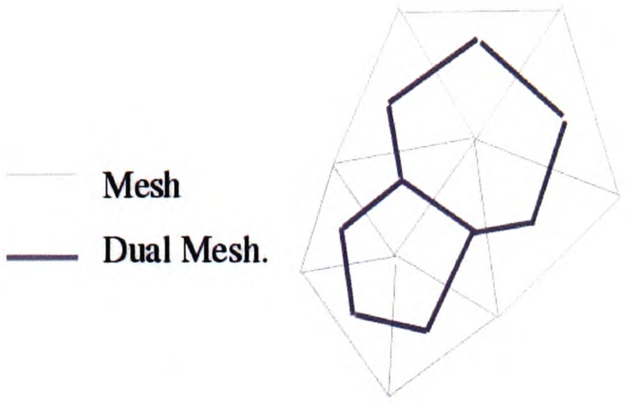


Figure 1.2: Delaunay mesh and associated dual mesh

cell-centred techniques. A variety of vertex-centred techniques have been applied in the field of aerodynamics to model flow past complex aerodynamic shapes. Jameson *et al* [48] used vertex-centred techniques to calculate inviscid transonic flow over a complete aircraft. Chakrabartty [49, 50] employed a vertex-centred finite volume scheme for flow past complex aerospace geometries.

Although much work has been done on developing unstructured finite volume techniques capable of obtaining flow solutions on complex meshes, problems are still encountered when solving on highly non-orthogonal meshes. Overcoming spurious pressure fields can still be a problem. There are unresolved problems with the Rhie and Chow scheme. Croft [2] showed that the Rhie Chow interpolation method can lead to non-physical cell centred values despite satisfying continuity. Majumdar [34] and Choi [51] reported that the converged velocity field is slightly dependent on relaxation factor and time step size. Recently, Kawaguchi *et al* [52] reported that Rhie Chow interpolation could cause checkerboard pressure predictions. Unstructured staggered mesh methods are still in the development stage and are limited by the difficulty in defining an orthogonal dual mesh.

1.2 The Finite Element Method

Both the FVM and Finite Element Method (FEM) can be viewed as subsets of the method of weighted residuals, Finlayson [53]. The main differences between the methods are the finite space over which the equations are integrated and the weighting functions used. In the FVM the weighting functions can be regarded as equal to 1. The governing equations are multiplied by a set of weighting functions, integrated, and residuals minimised. In FEM's a variety of weighted residual methods have been investigated, the optimum method is generally accepted as being the Bubnov-Galerkin weighted residual approach.

In the FEM the domain is meshed using a number of non-overlapping elements. Simple piecewise shape functions valid on local elements are used to describe the local behaviour of a variable ϕ within an element. The solution domain is meshed using any type of element for which a shape function exists. This gives the advantage of allowing complex geometries to be meshed using fully unstructured meshes. The shape functions approximating ϕ are then substituted into the governing equations. These approximate functions to the governing equations will not hold exactly and a residual is defined to measure the errors. The residuals are minimised by multiplying the governing equations by a set of weighting functions and integrating over the mesh element. The weighting functions W_i are local in that they are non-zero over element i, but zero everywhere else in the domain. The Bubnov-Galerkin weighted residual approach requires that the weighting and shape functions be the same. The residuals are only required to be zero in some weighted sense, the conservation principle is not enforced locally but global conservation is ensured. Finite element methods bring a high degree of accuracy to arbitrary meshes, allowing complex geometries to be modelled. Any type of element, for which a shape function exists, can be employed. The finite element method was developed initially for the solution of stress and displacement calculations in structural analysis and was first applied to fluid flow problems in the early 1970's [54, 55]. It has since been developed for all fluid flow problems and a standard work for fluid applications is given by Zienkiewicz and Taylor [56]. Textbooks by Baker [57] and Chung [58] are devoted exclusively to the subject of fluid flow computation by finite elements. Other examples of finite elements being applied to convection-diffusion and flow problems are given in, [59] to [64] amongst others.

In finite element methods checkerboard pressure fields have been eliminated by a variety of techniques, including:

- Obtaining a pressure field using a Poisson equation, Schneider *et al* [65].
- Using a mixed interpolation approach as introduced by Hood and Taylor [66].
- Comparison of various mixed interpolation methods, Huyakorn *et al* [67].
- Special techniques to filter out spurious pressure or introducing pressure smooth ing methods, Sani *et al* [68] and Hughes *et al* [69].
- Penalty function formulations that transform the governing equations that are constrained by the continuity equation into unconstrained equations. Pressure can then be eliminated from the equations as done by Bercovier and Engelman [70] and Heinrich and Marshall [71], or semi-implicitly coupled by Uszawa's algorithm, Fortin and Fortin [72]. Several variants of the penalty formulation have been proposed, among them [73, 74].

Finite element methods are increasingly used to solve complex flow problems due to their versatility in discretising complex geometries. They have the advantage of handling distorted meshes with ease, but require extensive storage/topological information and are computationally expensive. Unlike finite volume methods which ensure local conservation, finite elements are only globally conservative.

1.3 Mixed Finite Volume and Finite Element Techniques

Finite volume and finite element methods have emerged as the dominant CFD analysis techniques. Although there are advantages and disadvantages associated with both the FVM and FEM, finite volume is the preferred method for processes involving flow. However, the application of the finite element method to flow problems has improved over the decades and is favoured by some in the simulation of complex fluid flow problems due to its versatility when modelling complex geometries. Since there are merits with either method, much research has been done over the years to combine aspects from both methodologies. The first example of the amalgamation of ideas borrowed from FEM and FVM is given in the works of Winslow [75] and Williamson [76]. Feistauer *et al* [77] modelled flows involving shock waves using a combination of FV discretisation and FE approximations. A combined finite volume - finite element method employing finite volume discretisation to the convection term and employing the finite element method for diffusion have been used by [78], and other authors. Combined FV and FE approaches have been applied to flow problems by , [79] to [83], among others. A finite volume format utilising aspects from FE's was developed for structural mechanics by Onate *et al* [85]. It has also been employed to model linear elastic problems, Fryer *et al* [86, 87], Bailey and Cross [88] for non linear elastic problems. Taylor [89] used a Vertex-Based Finite Volume discretisation scheme employing finite element type approximations for stress-strain analysis.

The Control Volume-Finite Element Method (CV-FEM) for fluid flow problems emerged in the early 1980's. This method combines the flexibility of finite elements to discretise complex geometry with the conservation formulation of the Finite Volume Method. This method can be viewed in a finite volume context as a vertex-based FVM. The local variation of a variable ϕ within an element is described by simple piecewise polynomial functions, as used in the FEM. The terms in the governing equations can be integrated as volume integrals or surface integrals. Compos Silva and Moura [90] applied surface integrals to the momentum equations and volume integrals for the pressure term and continuity equation. Baliga and Patankar [91, 92] originally used an unequal-order method to overcome checkerboard velocity and pressure fields. The momentum and continuity equations were discretised over different sets of control volumes, termed macro- and micro-elements. Effectively, the pressure is discretised over a coarser grid (macro-elements) and computed at fewer grid points than velocity. There are a number of disadvantages with this method. Mass conservation is not strictly satisfied over the momentum control volumes. Accuracy is reduced when pressure varies strongly over the domain. Separate solvers are required in the solution of pressure and velocity. Finally, extensive storage or specification of geometrical or topological information is required. In 1985 Prakash and Patankar [93] proposed an equal-order method which does not suffer from spurious checkerboard pressure fields. The key idea is similar to using a staggered grid. A new face velocity field is defined which is dependent on the pressure difference between adjacent nodal points. This new velocity field is substituted into the discretised continuity equation and hence a checkerboard pressure field can not emerge.

The CV-FEM has been applied to flow problems by Baliga [94], Ramadhyani [95], Prakash [96, 97], Raw [98], Hookey [99] among others. Various mesh element types have been investigated for two-dimensional flow, Baliga and Patankar [100] using triangular elements. Saabas and Baliga [101, 102] used triangular and tetrahedral elements for two- and three-dimensional flow problems. Raw and Schneider [103, 104] utilised quadrilateral elements. A nine-node element was employed to solve two-dimensional heat conduction problems by Raw *et al* [105]. Banaszek [106] compared six-node and nine-node elements for diffusion problems. Wintruff *et al* [107] used unstructured triangular grids to solve two-dimensional melting and solidification problems. Reyes *et al* [108] simulated turbulent flow in irregular geometries using triangular elements.

1.4 Outline of Contents

The finite volume method is described in chapter 2. Details of the cell-centred and vertex-based discretisation methods investigated in this research are presented. Problems associated with the use of cell-centred techniques on skewed meshes are discussed. The various types of mesh skewness and the need for correction terms when solving cell-centred on a distorted mesh are shown. The methods used in the discretisation of each term in a general transport equation for a scalar variable are presented. A number of discretisation schemes in the evaluation of the convective flux are presented and discussed.

Chapter 3 presents a set of standard validation test cases for convection-diffusion of a scalar variable. The first test case is for transient-diffusion, the vertex-based discretisation technique used in the discretisation of the transient term is investigated. Steady-state heat conduction is used to illustrate the errors introduced in the cell-centred discretisation of the diffusion term on non-orthogonal meshes. The ability of the vertex-based technique to handle non-orthogonalty in a mesh is shown. Finally, a pure convection test case is employed to compare cell-centred and vertex-based upwind schemes.

Chapter 4 introduces the coupling of multi-physics problems. The vertex-based method is coupled with existing PHYSICA physics modules. The performance of the vertex-based and cell-centred techniques applied to three-dimensional heat conduction plus solidification/melting phase change in a multi-material geometrical domain are compared. The simulation of the casting of a turbine blade is performed using a thermal only and a thermo-mechanical analysis. A moving boundary condition is applied to approximate the withdrawal of the assembly from a furnace. Results, simulation times and memory requirements are discussed for both methods.

Algorithms specific to the solution of the hydrodynamic variables are presented in chapter 5. The SIMPLE, SIMPLER and revised SIMPLER solution algorithms employed in the coupling of pressure with the velocity components are described. Problems of decoupling and 'checker-boarding' results are discussed. Methods employed to avoid spurious pressure results are presented. A detailed account of the cell-centred and vertex-based solution procedure is given. Hybrid discretisation of the hydrodynamic variables are investigated. The cell-centred, vertex-based and hybrid solution procedures are applied to a test case involving the simulation of a lid-driven cavity. Comparisons are made with benchmark solutions on meshes with different degrees of skewness. The ability of the methods to handle distorted meshes are discussed.

A method combining vertex-based flow solutions with cell-centred discretisation techniques for other transported variables is outlined in chapter 6. The method is applied to a test case combining heat transfer and fluid flow in the simulation of buoyancy driven flow in a cavity. The simulations are carried out for a variety of Rayleigh numbers. Results are compared with benchmark solutions for purely cell-centred, purely vertex-based and combined vertex-based flow - cell-centred temperature solutions. The simulations employ uniform and distorted meshes for the vertex-based and combined solutions. A measure of the error on the distorted meshes is given and mass conservation at an element control volume level is investigated for the combined vertex-based - cell-centred method.

The combined vertex-based - cell-centred method is extended to turbulent flow in chapter 7. Models used in the solution of turbulent flow are presented. The $k - \epsilon$ two equation turbulence model is employed in the solution of turbulent flow. The cell-centred discretisation technique is applied to the turbulent variables k and ϵ. The method is coupled with the vertex-based solution methods for hydrodynamic variables and compared with purely cell-centred solutions. Both methods are applied to flow over a backward facing step problem. Results, simulation times and memory requirements are compared. In the second case the method is applied to three-dimensional turbulent flow over an aircraft wing. Results are compared on uniform meshes. Finally the mesh is distorted and solutions obtained using combined vertex-based / cell-centred solution techniques.

In Chapter 8, the conclusions of this study, including suggestions for future work, are given.

Chapter 2

Finite Volume Method

The cell-centred finite volume approach and vertex-based discretisation technique being investigated in this thesis employ the finite volume principles given below:

- The solution domain is divided into a number of finite non-overlapping control volumes.
- The governing equation is integrated over each control volume to yield a discretised equation at discrete points in the solution domain.
- Conservation is enforced locally by means of consistent expressions for fluxes through the cell faces of adjacent control volumes.
- Guaranteeing local conservation also ensures global conservation for the entire solution domain.

A general conservation equation that highlights the common features of the various transport processes can be employed to describe the transport of a general scalar variable ϕ, for example temperature or pollutant concentration. The conservation equation can be expressed by the following equation:

$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{transient} + \underbrace{div(\rho\phi\mathbf{u})}_{convection} = \underbrace{div(\Gamma grad\phi)}_{diffusion} + \underbrace{S_\phi}_{source} \tag{2.1}$$

The general equation is integrated over each control volume as well as over time. Through the discretisation procedure each term is approximated to produce a system of linear equations of the form $[A][\phi] = [B]$, where $[\phi]$ is a vector of values of ϕ at a number of finite points in the solution domain.

2.1 Cell-Centred Approach

In the cell-centred approach the control volume is associated with the mesh element. The general variable ϕ is solved and stored at the cell centre (element centroid), Figure 2.1 shows the control volume in 2 dimensions.

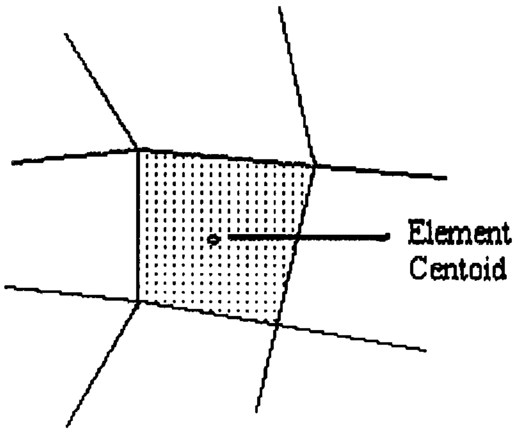


Figure 2.1: Cell-centred control volume

The terms in the general transport equation are discretised using finite-difference type approximations to describe how ϕ varies between solution points. These approximations assume that adjacent element centre points lie on a line, perpendicular to the element face, that passes through the face centroid. These approximations are sufficient for orthogonal meshes but non-orthogonal meshes require corrections to be made to the usual discretisation process.

2.1.1 Non-Orthogonal Mesh

Fitting a mesh to a complex geometry frequently leads to some degree of non-orthogonality in the generated mesh. Extra terms need to be included, when discretising the transport equations, to accurately model the transfer of conserved quantities in the skewed mesh. The diffusive flux across a face requires the inclusion of a secondary gradient term and this can lead to face fluxes that are no longer computed in terms of neighbouring cell values. Most of the terms in the discretised transport equation require face values of ϕ, for orthogonal meshes this is achieved through interpolation of adjacent cell values, for non-orthogonal meshes an extra term is required based on the gradients of ϕ. The inclusion of these correction terms allow solutions on meshes with some degree of skewness. However, approximations made in calculating the corrections can introduce significant errors into the solution procedure, leading to oscillatory results, difficulties in convergence and longer simulation times. On highly skewed meshes a high percentage of the solution can comprise of these correction terms and the solution may diverge. The correction terms used in PHYSICA are correct to first order and are categorised in terms of two aspects of mesh skewness, [2]. The corrections required are presented in the remainder of this section as a) non-orthogonality, the line connecting element centre points does not lie along the face normal (affects the diffusion term only) and b) non-conjunctionality, the line connecting element centre points does not pass through the face centroid.

2.1.1.1 Non-Orthogonality

The diffusion term requires the estimation of the derivative of ϕ with respect to the face normal. Using finite difference approximations gives the derivative of ϕ with respect to the face normal only when the line connecting the adjacent cell centres is parallel to the face normal vector, see Figure 2.2. This is not the case for non-orthogonal control volumes and a correction term is required.

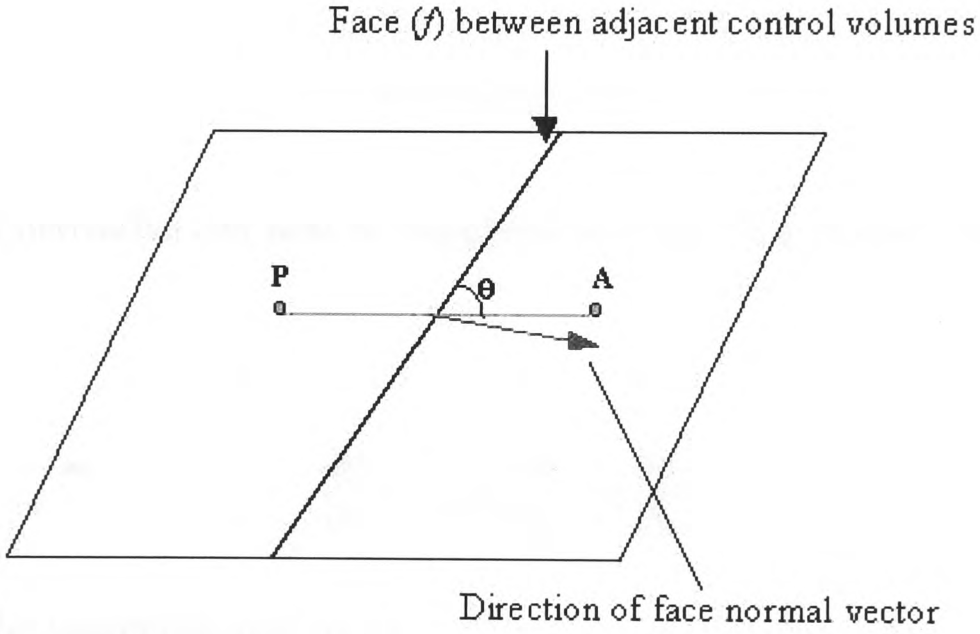

Face (*f*) between adjacent control volumes
P
θ
A
Direction of face normal vector

Figure 2.2: Adjacent non-orthogonal control volume's

In the case of orthogonal meshes the angle θ, shown in Figure 2.2, is 90 degrees. Let v represent the vector along the line connecting the cell centroids P and A:

$$\frac{\partial \phi}{\partial v} = \frac{\phi_A - \phi_P}{d_{AP}} \tag{2.2}$$

This can be viewed as the primary gradient of the diffusion term. For orthogonal meshes this vector is in the direction of the face normal and is sufficient:

$$\frac{\partial\phi}{\partial n} = \frac{\partial\phi}{\partial v} = \frac{\phi_A - \phi_P}{d_{AP}} \tag{2.3}$$

In Figure 2.2 the mesh is considered to be non-orthogonal because the line joining the cell centroids P and A is not perpendicular to the face (f) and the angle θ is no longer 90 degrees. In this case the derivative can not be written purely in terms of (2.3), an additional secondary gradient is required to compute $\partial\phi/\partial n$. The face normal vector can be written in terms of the v vector plus a vector ξ which is tangential to the face:

$$\underline{n} = (\underline{v}.\underline{n})\underline{v} + \underline{\xi} \tag{2.4}$$

The normal derivative can now be expressed in terms of a primary and secondary gradient,

$$\frac{\partial\phi}{\partial n} = (\underline{v}.\underline{n})\frac{\partial\phi}{\partial v} + \xi\frac{\partial\phi}{\partial t} \tag{2.5}$$

where t is the tangential unit vector.

The secondary gradient term can easily be expressed in terms of its Cartesian components, giving:

$$\frac{\partial\phi}{\partial n} = (\underline{v}.\underline{n})\frac{\partial\phi}{\partial v} + \xi_x\frac{\partial\phi}{\partial x} + \xi_y\frac{\partial\phi}{\partial y} + \xi_z\frac{\partial\phi}{\partial z} \tag{2.6}$$

The problem of computing the secondary gradient is now reduced to the problem of computing the Cartesian derivatives of ϕ on the control volume faces. Linear interpolation of nodal derivatives can be employed to compute face derivatives of ϕ. The Cartesian derivative of ϕ at a node can be computed by integrating the derivative over the control volume associated with the node and applying the divergence theorem. This allows the nodal values of the derivatives to be calculated in terms of the sum of the face values of ϕ:

$$\begin{aligned} \int_{cv} \frac{\partial \phi}{\partial x} dv &= \int_S \phi n_x dS \\ &\approx \sum_f A_f \phi_f n_x \end{aligned} \tag{2.7}$$

The inclusion of the secondary gradient in non-orthogonal meshes can significantly improve the accuracy of the solution. For most good quality meshes this secondary term is relatively small but for highly non-orthogonal meshes the secondary term can dominate. This term enters the equations as an extra source term which can lead to instabilities in the solution procedure.

2.1.1.2 Non-Conjunctionality

In terms requiring the calculation of control volume face values, the average value of the variable is taken at the centre of the face and presumed to prevail over the whole face. The face value is estimated using interpolation between the values at the nodes at the element centres either side of the face. As shown in Figure 2.3, on skewed meshes the line connecting the adjacent element centroids does not pass through the element face centroid and a correction term is required to estimate the value of the variable at the face centre.

The value at the intersection point I, where the line connecting the adjacent element centroids passes through the control volume face, is interpolated from adjacent values as for orthogonal meshes. The face centroid value is extrapolated from the intersection quantity plus the gradient of the quantity from point I to f,

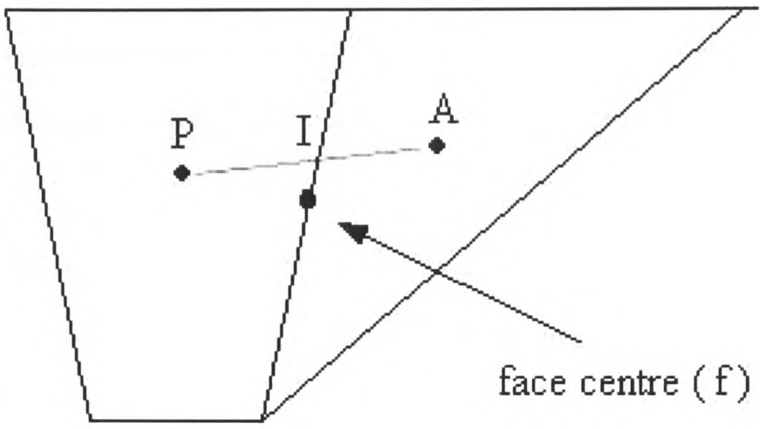


Figure 2.3: Adjacent non-conjunctional control volume's

$$\phi_f = \phi_I + \underline{d}_{If}.grad\phi_f \tag{2.8}$$

where $\underline{d}_{If}$ is a vector from the intersection point to the face centre.

To calculate the gradients of ϕ the divergence theorem can be employed, as in (2.7). The problem is then encountered that Equation (2.7) requires knowledge of ϕ_f. Substituting (2.8) into (2.7), gives

$$\frac{\partial \phi}{\partial x} = \frac{1}{V}\sum_f A_f n_x \left(\phi_I + \sum_j \left[\frac{\partial \phi}{\partial x_j}\right]_f (d_{If})_i\right) \tag{2.9}$$

where $[\partial\phi/\partial x_j]_f$ is interpolated from values in adjacent elements. To calculate $[\partial\phi/\partial x_j]_f$ we need to know $[\partial\phi/\partial x_j]$ which requires knowledge of ϕ_f. The simplest and most commonly used solution is to use previous iteration values of the gradients ϕ on the right hand side of equation (2.9).

2.2 Vertex-Based Approach

The previous paragraphs have highlighted the problems with the unstructured cell centred method on non-orthogonal meshes. The vertex-based technique, outlined in this section, allows the direct computation of the derivatives of ϕ in the required direction. The value of ϕ can also be calculated directly at any required point within the control volume, thus negating the need for correction terms on skewed meshes.

The vertex-based control volume is constructed around the vertices of the mesh element. For a two-dimensional mesh quadrilateral or triangle elements are used and for a three-dimensional mesh the mesh elements can be hexahedral, pentahedral or tetrahedral. Each element is divided into a number of sub-control volumes by connecting the element centroid to the element face centres, as illustrated in Figure 2.4 for a two-dimensional quadrilateral mesh element.

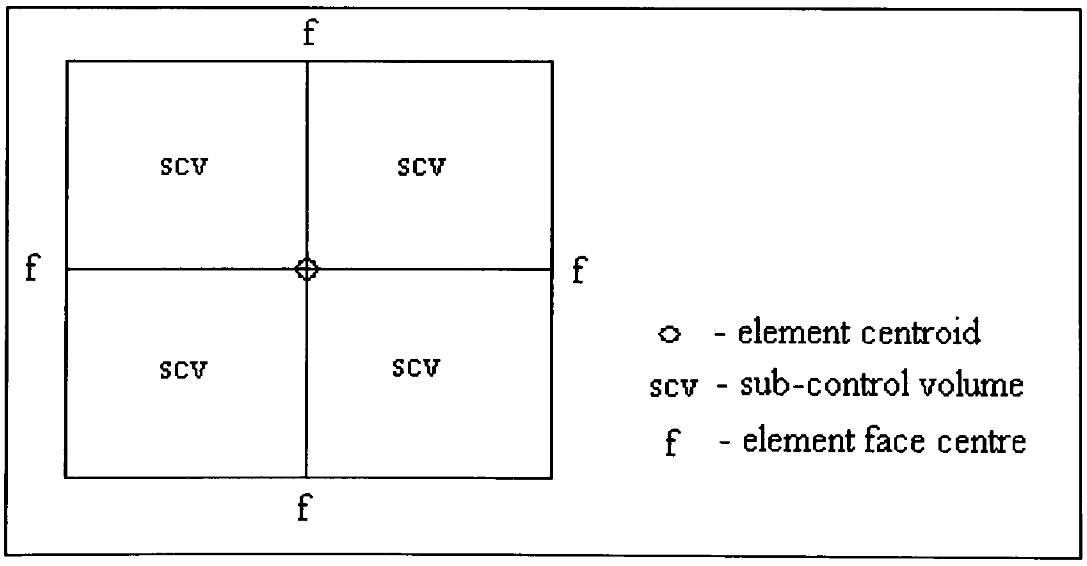


Figure 2.4: Quadrilateral element and sub-control volumes

The control volume consists of a number of sub-control volumes constructed around the mesh vertices. The internal surfaces of the sub-control volumes within the mesh element define the control volume, as illustrated in Figure 2.5 for a two-dimensional mesh. The general variable ϕ is solved and stored at the vertices of the mesh elements.

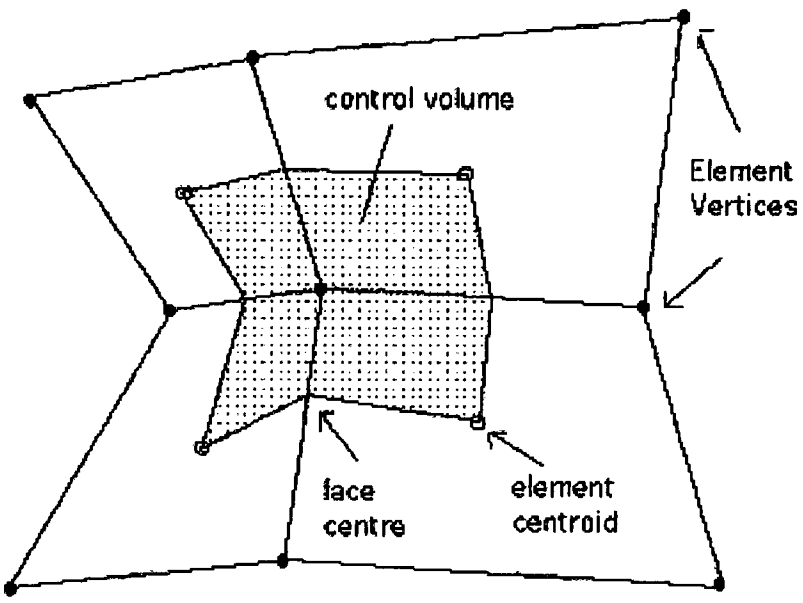


Figure 2.5: Vertex-based control volume

The terms in the conservation equation are approximated working in local co-ordinates. This allows for all elements of a certain type to be treated identically regardless of how distorted any element may be in terms of global co-ordinates. The local co-ordinate system for a quadrilateral element is shown in Figure 2.6 and for triangular, hexahedral, pentahedral and tetrahedral elements in Appendix A.

The integration points are located at the centre of the sub-control faces which form the vertex-based control volume, as illustrated in Figure 2.7 for a quadrilateral mesh element. The local co-ordinates of the integration points located on the sub-control volume faces for triangular, hexahedral, pentahedral and tetrahedral elements are given in Appendix A.

The terms in the general transport equation are discretised using simple linear shape function approximations. The shape functions used on each element type and their

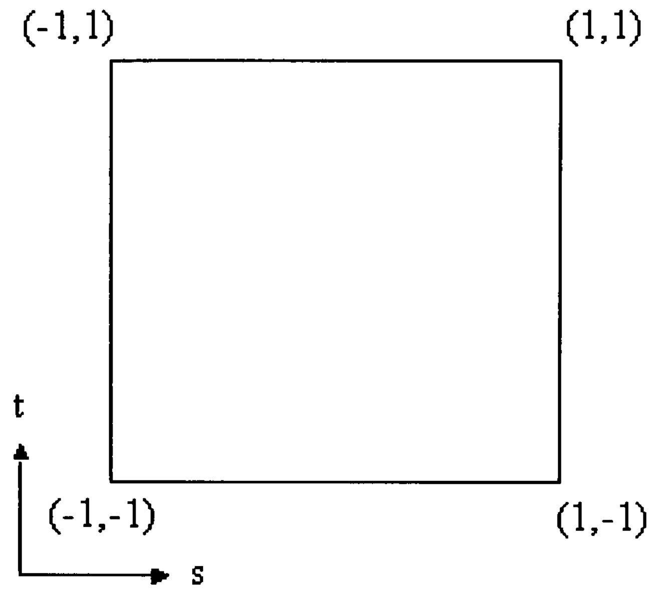


Figure 2.6: Normalised quadrilateral element

associated derivatives are defined in local co-ordinates and given in Appendix B.

The local co-ordinate system is transformed into global co-ordinates, the co-ordinate transformation is performed using the same shape functions. The elements described here are isoparametric, allowing for both the co-ordinate transformations and the variable approximation within the element to be described by the same shape functions. The local to global transformations, the variation of a variable within the element and the partial derivatives of the variable with respect to local co-ordinates are described in Appendix C.

2.3 Comparison of Approaches

On a one-dimensional mesh the vertex-based discretisation of the domain will comprise of one extra solution point more than the cell-centred approach. The approx-

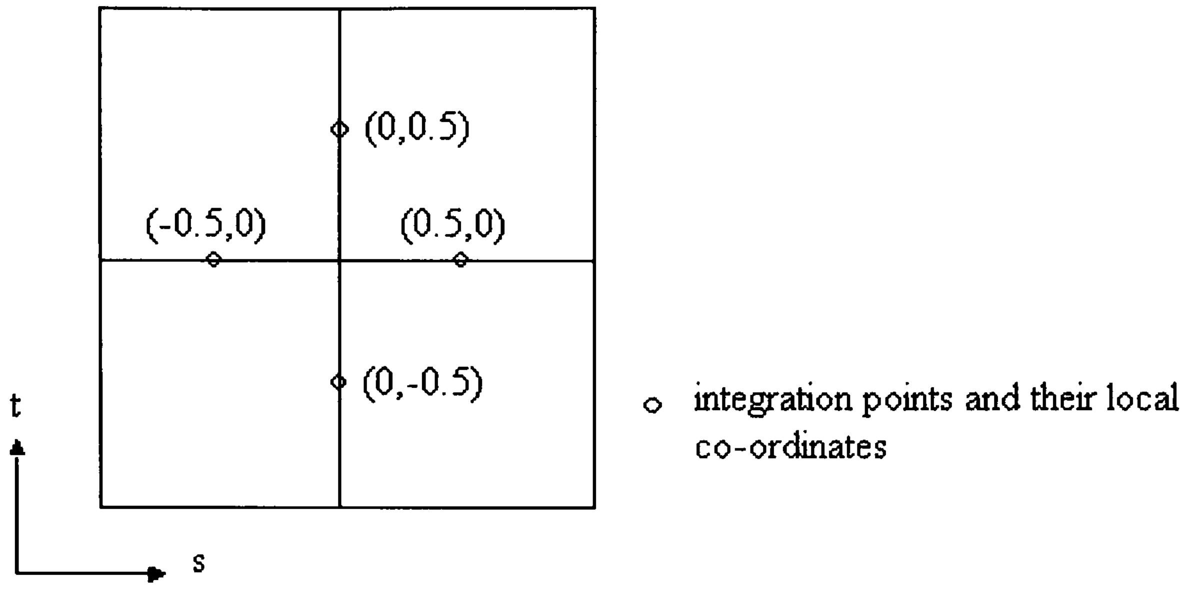


Figure 2.7: Integration points for quadrilateral element

imations made during the discretisation process are equivalent for internal control volumes with evenly spaced solution points. For two- or three-dimensional meshes the size of the resulting set of equations depends upon the choice of method, vertex or cell based. On a mesh constructed of quadrilaterial or hexahedral elements there are more vertices than elements in the mesh. For triangular mesh discretisation there are approximately twice as many elements as vertices and for a tetrahedral mesh five to six times as many elements as vertices.

For two- and three-dimensional meshes of the same type the vertex-based control volume comprises of a greater number of faces and the discretised equations have a larger number of possible non-zero coefficients. Figure 2.8 shows the control volumes associated with node P when using the cell-centred and vertex-based approaches for a mesh constructed of quadrilateral elements. In the cell-centred case P has four neighbours N,S,E and W giving a possible five non-zero coefficients in the discretised equation. The vertex-based control volume P is constructed of eight faces with neighbours N, NE, E, SE, S, SW, W and NW leading to a possible nine non-zero coefficients in the discretised equation. For a mesh comprising of hexahedral elements the cell-centred discretised control volume would contain six faces and a possible seven non-zero coefficients in the discretised equation. A vertex-based control volume would comprise of twenty four faces giving a discretised equation with a possible twenty seven non-zero coefficients.

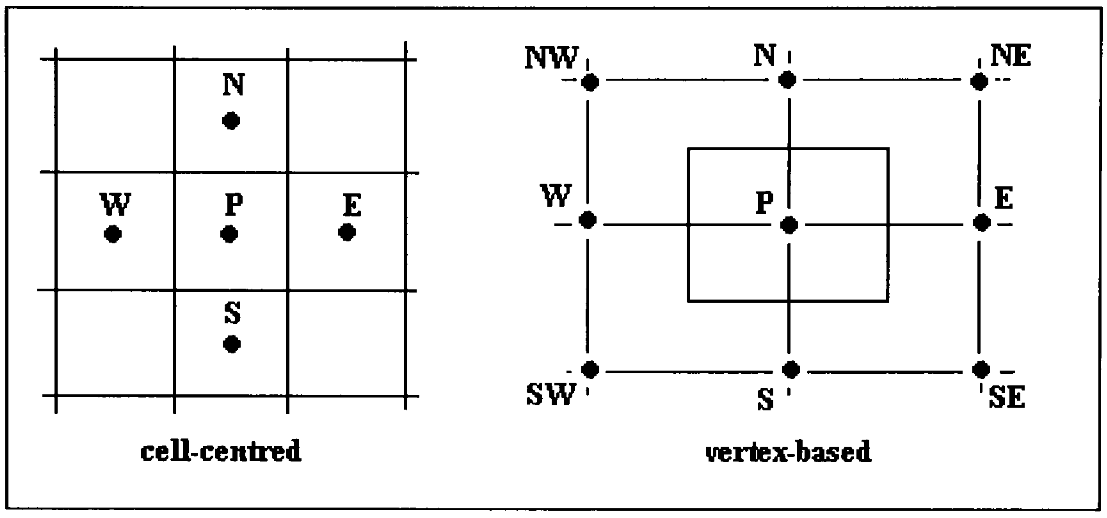


Figure 2.8: Control volume P and adjacent neighbours

Control volumes comprising of a larger number of faces should give greater accuracy when calculating the fluxes across the control volume boundary. This improved discretisation requires greater computational effort and produces a solution matrix with a much larger bandwidth than the cell-centred approach. Improved communication between surrounding nodes should lead to less solver iterations required for convergence. The cell-centred solution matrix is diagonally dominant and can be solved using nearly any iterative solvers. The vertex-based method produces a symmetric positive-definite solution matrix for sub-control volumes of equal size and an asymmetric positive-definite matrix for unequal sub-control volumes. Thus the resulting system of equations can require specialist solvers, such as the conjugate gradient solver [110, 111]. The solution matrix contains a larger number of non-zero's making it computationally more time consuming per solver iteration. The vertex-based method gives direct computation of unknown values at a boundary wall. However, problems can arise at intersection boundary vertices where boundary conditions conflict. In the cell-centred method boundary values are extrapolated from element based values which may give rise to errors.

2.3.1 System of Discretised Equations

The discretised equations at each solution point (p), represent a set of algebraic equations of the form,

$$a_p\phi_p = \sum a_{nb}\phi_{nb} + S_\phi \tag{2.10}$$

where $\sum$ indicates summation over all neighbouring nodes and a_{nb} are the neighbouring coefficients.

This set of equations are normally solved using iterative solution techniques. For converging realistic solutions the set of equations should satisfy certain conditions. The conditions required for cell-centred and vertex-based discretisation are as follows,

Cell-Centred

The cell-centred method uses techniques to ensure that the matrix of coefficients is diagonally dominant,

$$\frac{\sum \mid a_{nb} \mid}{\mid a_p \mid} \begin{cases} \leq & 1 \text{ at all nodes} \\ < & 1 \text{ at one node at least} \end{cases} \tag{2.11}$$

Scarborough [112] has shown this to be a sufficient condition for convergent iterative methods. Another condition that should be satisfied is 'boundedness', in the absence of sources the internal nodal values of ϕ should be bounded by its boundary values. Boundedness is satisfied if all coefficients of the discretised equation are positive.

Vertex-Based

The vertex-based method results in a matrix of coefficients which is symmetric for equally sized sub-control volumes or asymmetric for unequal sub-control volumes, and positive definite,

$$[X^T][A][X] > 0 \tag{2.12}$$

where $[X]$ is a vector of the values of ϕ and $[A]$ is a matrix of coefficients.

Unlike the cell-centred method where all neighbouring coefficients, a_{nb}, are positive the vertex-based method contains some negative neighbouring coefficients. However, for converging solutions the sum of the neighbouring coefficients, $\sum a_{nb}$, should always be positive, [93]. If $\sum a_{nb}$ becomes negative the solution becomes 'unbounded' or divergence is encountered.

2.4 Discretisation of the General Transport Equation

This section covers the discretisation techniques used to approximate terms of the general transport (2.1). The vertex-based approach uses linear shape functions to approximate the variable ϕ and its partial derivatives as shown in Appendix B. The cell-centred approach uses finite difference type approximations. The methods described here can be applied to the transport of any scalar quantity.

2.4.1 Transient Term

The first step is to integrate the transient term over the control volume and over a time interval t to $t + \Delta t$:

$$\int_{t}^{t+\Delta t} \int_{cv} \frac{\partial(\rho\phi)}{\partial t} dV dt \tag{2.13}$$

On a stationary mesh the volume, V, does not change with time and the order of the integrals can be reversed,

$$\int_{V} (\rho\phi - \rho^0\phi^0) dV \tag{2.14}$$

For a control volume (cv) this equation is approximated by,

$$V_{cv}(\rho_{cv}\phi_{cv} - \rho_{cv}^0\phi_{cv}^0) \tag{2.15}$$

where the superscript 0 indicates values at the previous time step and V is the volume of the control volume.

On a moving mesh [109] if the relative velocity is used in the convection term, $\underline{v} - \underline{v}_m$ where $\underline{v}_m$ is the mesh velocity, then the integrated form of the transient term becomes,

$$\int_{t}^{t+\Delta t} \frac{d}{dt} \int_{V} \rho\phi dV dt \tag{2.16}$$

which is equal to,

$$\int_V \rho\phi dV\Big|_t^{t+\Delta t} \tag{2.17}$$

giving a discretised equation for a control volume as:

$$V_{cv}\rho_{cv}\phi_{cv} - V_{cv}^0\rho_{cv}^0\phi_{cv}^0 \tag{2.18}$$

Using the fully implicit scheme all other terms in the equation are multiplied by Δt. Dividing by Δt means that the dependence on integration over time can be ignored in the remaining terms:

$$\frac{\rho_{cv}\phi_{cv}V - \rho_{cv}^0\phi_{cv}^0 V_0}{\Delta t} \tag{2.19}$$

Cell-Centred

Assuming that the value at the centre of a control volume is a representative of the average value in the control volume, the discretisation is complete.

Vertex-Based

For vertex-based discretisation Equation (2.19) is sufficient. However, since the control volume is made up of a number of sub-control volumes, the discretised transient term can be further written as:

$$\frac{1}{\Delta t}\sum_{scv}(\rho_{scv}V_{scv}\phi_{scv} - \rho_{scv}^0 V_{scv}^0\phi_{scv}^0) \tag{2.20}$$

The chosen integration point could be any position within the sub-control volume. The position is assumed to be representative of the average value in the sub-control volume and presumed to prevail over the whole sub-control volume. The value of ϕ

and position in local co-ordinates can be described by local shape functions giving the discretised term as,

$$\frac{1}{\Delta t}\sum_{scv}\sum_{i=1}^{n}(\rho_{scv}V_{scv}N_i\phi_i - \rho_{scv}^0 V_{scv}^0 N_i\phi_i^0) \tag{2.21}$$

where n is the number of element nodes. N_i is the shape function associated with node i and ϕ_i is the value of the variable at node i.

The logical integration point is the centre point of a sub-control volume. Care must be taken when using this form of discretisation since adding to neighbouring coefficients can lead to $\sum a_{nb}$ becoming negative resulting in 'unbounded' or diverging solutions. An investigation into discretising the transient term as Equation (2.21) is undertaken in Chapter 3. The resulting system of equations is shown in Appendix D.

2.4.2 Source Term

To speed up the convergence of a solution the source term may be expressed in linearised form [6],

$$S_\phi = S_c - S_p\phi \tag{2.22}$$

where S_c and S_p can be functions of ϕ or any stored value. For stability and convergence the value of S_p must be positive.

An example of a linearised temperature source of the form $T_d - T$, where T_d is a stored value, is given as:

$$S_c = T_d \qquad S_p = 1.0 \tag{2.23}$$

Placing a portion of the source in the linear S_p part, method (2.23), should potentially speed up convergence.

Cell-Centred

The linearised source term for a cell-centred control volume where all terms are evaluated at the centre of the element is discretised as:

$$V_p(S_c - S_p\phi_p) \tag{2.24}$$

Vertex-Based

Equation (2.24) can be applied to a vertex-based control volume. Ensuring S_p is positive adds to a_p satisfying positive-definite and ensures boundedness requirements. Alternatively (2.24) could be applied to the vertex-based sub-control volumes,

$$(S_\phi)_{cv} = \sum_{scv}(S_\phi)_{scv} \tag{2.25}$$

Giving the linearised source term as:

$$(S_\phi)_{scv} = (S_c)_{scv} - (S_p)_{scv}.\phi_{scv} \tag{2.26}$$

Assuming linear variation of ϕ over an element, the source term for a vertex-based control volume, can be written as,

$$\sum_{scv}\sum_{i=1}^{n} V_{scv}\left((S_c)_{scv} - (S_p)_{scv}N_i\phi_i\right) \tag{2.27}$$

where all terms are evaluated at an elemental level and subscript i refers to element nodal values. If the source term is large this form of discretisation can result in diverging solutions that do not satisfy boundedness, since it contains the potential for $\sum a_{nb}$ to become negative.

2.4.3 Diffusion Term

The first step is to integrate the diffusion term over the control volume.

$$\int_{cv} div(\Gamma_{\phi} grad\phi)dV \tag{2.28}$$

The volume integral is rewritten as an integral over the entire bounding surface of the control volume using Gauss' divergence theorem.

Gauss' divergence theorem states, for a vector $\underline{F}$,

$$\int_{V} div(\underline{F})dV = \int_{S} \underline{F}.\underline{n}dS \tag{2.29}$$

where S is the surface enclosing V. $\underline{n}$ is the outward normal vector to the surface.

Giving:

$$\int_{V} div\left(\Gamma_{\phi} grad(\phi)\right) dV = \int_{S} \Gamma_{\phi} grad(\phi).\underline{n}dS \tag{2.30}$$

The bounding surface of the control volume consists of a set of faces. Consequently the equation can be written as a sum of surface integrals over each face bounding the control volume.

Cell-Centred

Equation (2.30) is simplified by recognising that $grad(\phi).\underline{n}$ is equal to the gradient of ϕ in the direction of the face normal:

$$\sum_f \int_f \Gamma_\phi \frac{\partial \phi}{\partial n} dS \tag{2.31}$$

For an orthogonal mesh the line connecting the cell-centred values across a control volume face f is parallel to the face normal. The normal gradient can be approximated by,

$$\frac{\partial \phi}{\partial n} \simeq \frac{\phi_{adj} - \phi_{cv}}{d_f} \tag{2.32}$$

where d_f is the distance across the control volume face between the control volume centroid and adjacent control volume centroid.

This gives a discretised equation for the control volume,

$$\sum_f (\Gamma_\phi)_f \left(\frac{\phi_{adj} - \phi_{cv}}{d_f} \right) A_f \tag{2.33}$$

where Γ_f on the face is calculated using a harmonic mean, thus:

$$(\Gamma_\phi)_f = \frac{(\Gamma_\phi)_{adj} (\Gamma_\phi)_{cv}}{\alpha_f (\Gamma_\phi)_{cv} + (1 - \alpha_f)(\Gamma_\phi)_{adj}} \tag{2.34}$$

$$\alpha_f = \frac{d_{adj-f}}{d_{adj-f} + d_{f-cv}} \tag{2.35}$$

Vertex-Based

Equation (2.30) can be rewritten, using tensor notation, as a sum of the bounding faces of the control volume, thus,

$$\sum_f \sum_{i=xyz} \Gamma_f A_f \left(\frac{\partial \phi}{\partial x_i} . n_i \right)_f \tag{2.36}$$

where A_f is the area of face f.

The partial derivatives of ϕ are calculated at integration points situated at the face centres and can be described by partial derivatives of the shape functions. Hence the discretised form of the equation is,

$$\sum_f \Gamma_f A_f \sum_{i=1}^{3} \sum_{j=1}^{n} \phi_j (\frac{\partial N_j}{\partial x_i} . n_i)_f \tag{2.37}$$

where N_j is the local shape function associated with node j on the element that contains the face f and ϕ_j is the value of the variable at node j.

The discretised diffusion term for internal solution points leads to a positive indefinite solution matrix, shown in Appendix D, where

$$[X^T][A][X] \geq 0 \tag{2.38}$$

giving a solution matrix which is near singular and extremely susceptible to rounding error. The addition of boundary conditions stabilises the matrix and fulfils positive definite requirements.

The vertex-based discretised diffusion term, Equation (2.37), is equally valid for all meshes. The cell-centred discretised diffusion term, Equation (2.33), assumes an orthogonal mesh. Non-orthogonality in the mesh requires the addition of correction terms, given in 2.1.1, for cell-centred discretisation.

2.4.4 Convection Term

The following discretisation of the convection term assumes that the velocity field has been resolved. The components of velocity are likely to be stored as either element centred or nodal values. This requires velocity components to be estimated on the control volume faces, a number of methods are discussed in chapter 5. For the present it is assumed that velocity values have been obtained at the required location.

The convection term is integrated over a control volume:

$$\int_{cv} div(\rho \underline{u} \phi) dV \tag{2.39}$$

As with the diffusion term the volume integral is rewritten as an integral over the entire bounding surface of the control volume using Gauss' divergence theorem, giving:

$$\int_{S} \rho(\underline{u}.\underline{n}) \phi dS \tag{2.40}$$

Again as with the diffusion term the equation can be written as a sum of surface integrals over each face bounding the control volume,

$$\sum_f \rho_f A_f (\underline{u}.\underline{n})_f \phi_f \tag{2.41}$$

where A_f is the area of face f and n_f is the outward normal vector to face (f).

In the vertex-based procedure ρ_f is simply taken to be the value of ρ in the element that contains the face. Using cell-centred discretisation ρ_f takes the upwind element value, thus:

$$\begin{aligned} \rho_f &= \rho_{cv} \qquad \text{if}(\underline{u}.\underline{n})_f > 0.0 \\ \rho_f &= \rho_{adj} \qquad \text{if}(\underline{u}.\underline{n})_f < 0.0 \end{aligned} \tag{2.42}$$

To complete the discretisation of the convection term, values of the transported property ϕ_f need to be estimated at the cell face. Methods employed to do this are discussed in the following section.

2.5 Convection Discretisation Schemes

The principle problem in discretising the convection term is obtaining good approximations of the value of ϕ at the control volume faces. Using linear differencing, as used in the diffusion term, does not take into account that unlike diffusion, which effects the distribution of ϕ along its gradients in all directions, the convection influence is only in the flow direction. Any expressions formulated to approximate ϕ_f should satisfy certain fundamental properties, [113]. The most important ones are:

- **Conservativeness** - To ensure conservation of ϕ over the whole solution domain the flux of ϕ through a common face must be represented by consistent expressions.

- **Boundedness** - In the absence of sources the converged solution of the internal properties of ϕ should be bounded by its boundary values. In cell-centred discretisation this is achieved by ensuring that all neighbouring coefficients are positive. The vertex-based discretisation requires the sum of the neighbouring coefficients to be positive.

- **Transportiveness** - The relative strengths of convection and diffusion are taken into account. For strongly convective flows ϕ_f is increasingly influenced by conditions upstream to the direction of flow.

2.5.1 Central Differencing

The cell-centred central differencing approach is a two-point scheme which employs a linear fit between adjacent element centre values. The control volume face value is obtained using simple averaging of the values in the elements either side of the face under inspection. The vertex-based approach of using linear shape functions to model the variation of ϕ within an element, allows a larger number of surrounding values to be taken into account when estimating ϕ_f. Tetrahedral elements give a four-point scheme, pentahedral elements allow a six-point scheme and hexahedral elements an eight-point scheme. Face values of ϕ, defined in local co-ordinates, can be approximated thus,

$$\phi_f = \sum_{i=1}^{n} N_i \phi_i \tag{2.43}$$

where ϕ_i is the variable described at node i of the element which contains face (f), giving the discretised convection term as:

$$\sum_f \rho_f A_f(\underline{u}.\underline{n})_f \sum_{i=1}^{n} N_i\phi_i \tag{2.44}$$

In discretising the convection term over a control volume, only surrounding nodal values of ϕ appear in the equation, terms connecting ϕ at the control volume node cancel out. This produces an positive-definite system of equations with the leading diagonal containing zero coefficients, shown in Appendix D. For convection only problems, using iterative solution methods, no solution is obtained. The solution of the system matrix requires division by the leading coefficients, division by zero is encountered. Since diffusion always occurs along-side convection in nature, the problem of zero leading coefficients is overcome by coupling the convection and diffusion terms. This method produces stable and accurate solutions when the strength of convection relative to diffusion is low. When convection dominates the sum of the neighbouring coefficients can become negative. The boundedness criteria is violated and the solution fails to converge or produces physically unrealistic oscillating results. In cell-centred techniques, if convection dominates, central differencing violates the requirement of diagonal dominance and positive neighbouring coefficients, Patankar [6].

2.5.2 Upwind Formulation

One of the major inadequacies of using central differencing is that ϕ_f is influenced by ϕ at the surrounding nodes without regard to the strength of convection relative to diffusion or the direction of the flow. As the strength of convection increases the value of ϕ at the cell face will be increasingly influenced by ϕ at nodes upstream and less by downstream conditions. A solution is to use an upwind formulation, first suggested for finite difference formulations by Courant, Isaacson and Rees [114].

Using an upwind formulation the direction of flow is ascertained on each cell face.

The value of ϕ on the cell face is taken as the nodal value in the upstream control volume.

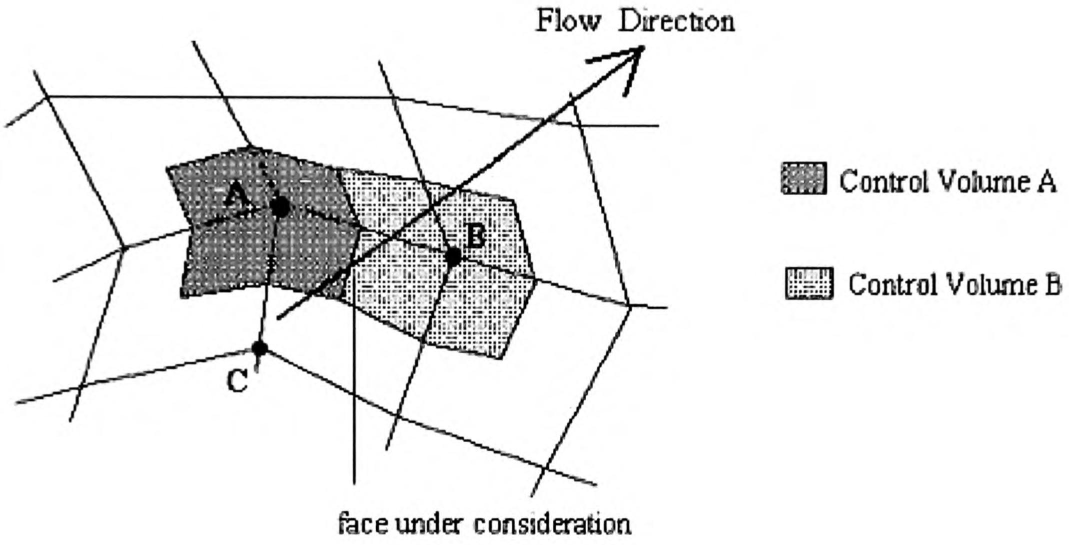


Figure 2.9: Determining value of ϕ on face of adjacent control volumes

Figure 2.9 shows adjacent control volumes for a two-dimensional case. If flow on a face of control volume A, with adjacent control volume B, is flowing from A into B. The face value of ϕ is taken as the nodal value in control volume A. Conversely if flowing from B into A, the face value of ϕ is taken as the nodal value in control volume B. This method produces a system of equations which is well conditioned, contains non-zero leading coefficients, and gives stable and converging solutions for convection and convection-diffusion problems. The upwind discretisation satisfies transportiveness requirements but is only first order accurate which makes it prone to numerical diffusion [115]. This numerical smearing reduces with grid refinement. The vertex-based approach allows more flexibility in selecting upwind values. A vertex-based control volume comprises a far greater number of faces than a cell-centred control volume. As each face is upwinded the combination of different upwind values for a vertex-based control volume is greater than for a cell-centred control volume. The upwind nodal value is also influenced by a greater number of neighbouring values, shown in Figure 2.10 for hexahedral mesh. The indirect influence of neighbouring values of ϕ on ϕ_f is much increased using vertex-based techniques.

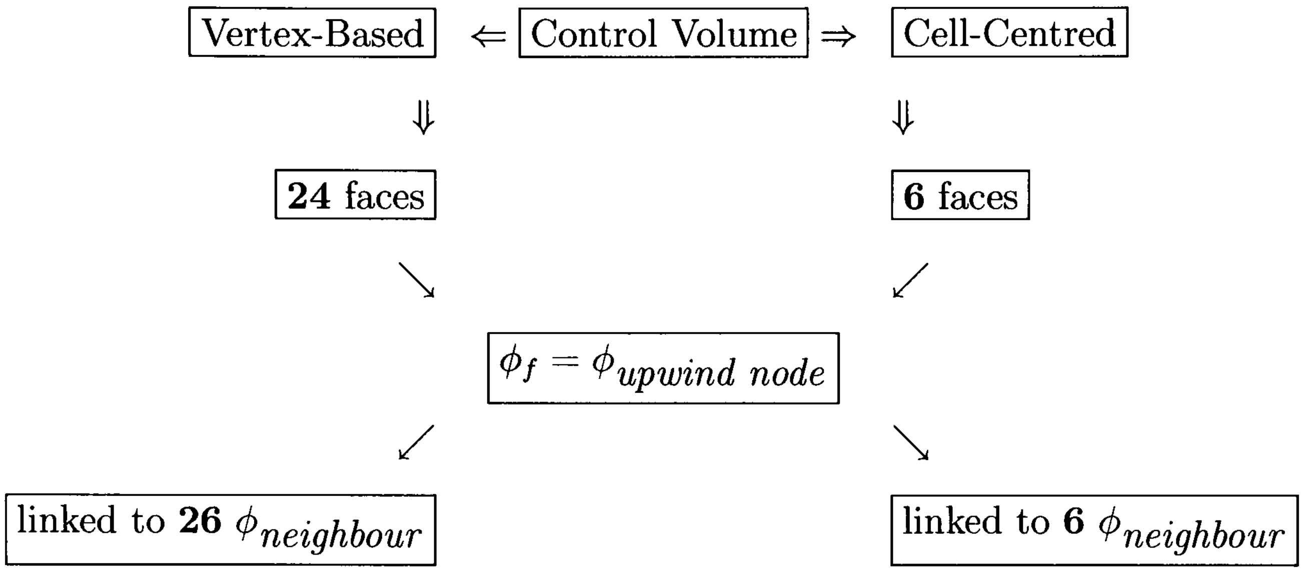

Vertex-Based
Control Volume
Cell-Centred
24 faces
6 faces
$\phi_f = \phi_{upwind\ node}$
linked to **26** $\phi_{neighbour}$
linked to **6** $\phi_{neighbour}$

Figure 2.10: Influence of nodal values on face values for hexahedral elements

2.5.3 Hybrid Scheme

Spalding [116] developed the hybrid-differencing scheme for finite-difference formulations, based on a combination of central and upwind differencing schemes. It is applied here using a combination of linear shape function interpolation and upwind formulations. The hybrid formulation exploits the beneficial aspects of both methods.

A measure of the relative strengths of convection and diffusion can be defined by a non-dimensional cell Peclet number (Pe), shown below,

$$Pe = \frac{\rho u \delta x}{\Gamma} \tag{2.45}$$

where δx is the element width across the centre point of the face in the direction of the face normal vector. u is the resultant velocity at the face centre.

The local Pe number is computed on each cell face, employing linear shape function interpolation for small Peclet numbers ($|Pe| < 2$) and the upwind formulation for large Peclet number ($|Pe| > 2$).

$$\begin{array}{ll} |Pe| < 2 & \phi_f = \sum_{i=1}^{n} N_i \phi_i \\ |Pe| > 2 & \phi_f = \phi_{\text{adjacent upwind cv}} \end{array} \tag{2.46}$$

For strongly convective flows ($|Pe| > 2$), the face value of ϕ is more strongly influenced by the relevant upstream value. As $|Pe|$ increases, $\partial\phi/\partial x$ tends to zero, indicating that the diffusive effect becomes negligible. Hence, for large $|Pe|$ values inclusion of the diffusion term in the upwind formulation leads to diffusion becoming overestimated, [6].

The hybrid scheme is the default differencing scheme in many CFD codes due to its stability and robustness. It gives good approximations to exact solution when flow aligns closely with grid lines and sources are small. Numerical accuracy is degraded due to the introduction of artificial numerical diffusion whenever flow streamlines are at an angle to grid lines . Solutions can suffer from a jump at points where the local $|Pe| = 2$ due to the suppression of physical diffusion.

2.5.4 Higher Order Schemes

A number of authors [117] to [126] have formulated higher order schemes in an effort to reduce false diffusion. These schemes attempt to reduce the error by bringing in a wider influence from surrounding neighbouring points. Three point differencing schemes utilise points located upwind and downwind of the face and an extra point located on the upstream side of the face. The quadratic upstream interpolation scheme (QUICK) was one of the earlier higher order schemes developed in 1979 by Leonard [120]. This scheme uses a three point upstream-weighted quadratic interpolation for cell faces and has been widely used for cell-centred discretisation. The face value of ϕ is obtained from a quadratic function passing through the two adjacent points, located at either side of the face, and a point located at the upstream side of the face. Using consistant quadratic profiles makes this scheme conservative and transportiveness is built-in as the quadratic function is based on two upstream and one downstream point value. Unfortunately the scheme can become unstable and unbounded producing minor under- and over-shoots in complex flow solutions. As a consequence the QUICK scheme has been re-formulated in several ways by a number of authors using weighting techniques in an attempt to alleviate stability problems, including Hayase *et al* [126], who derived a stable and fast converging variant.

A literature review and general study of three-point upwind schemes was published by Waterson and Deconinck [127] in 1995. The authors classify schemes as linear, i.e. coefficients are not a direct function of the convected variable, and non-linear schemes. Linear schemes can suffer from unboundedness problems and are not conditionally stable. Non-linear schemes attempt to adapt the discretization to fulfill particular boundedness criteria usually either Total Variation Diminishing (TVD), where the variation of the solution does not increase with time, or positivity. The non-linear SMART [117] scheme is the equivalent of a bounded QUICK algorithm. The authors express face values of ϕ in the general format,

$$\phi_f = \phi_w + 0.5B(\phi_w - \phi_{ww}) \tag{2.47}$$

where B is a function of r:

$$r = \frac{(\phi_p - \phi_w)}{(\phi_w - \phi_{ww})} \tag{2.48}$$

The location of ϕ_p, ϕ_w and ϕ_{ww} is shown in Figure 2.11 for a control volume face (f). Table 2.1 shows Waterson's generalization of the definition of $B(r)$ for a selection of linear schemes and their bounded non-linear alternatives.

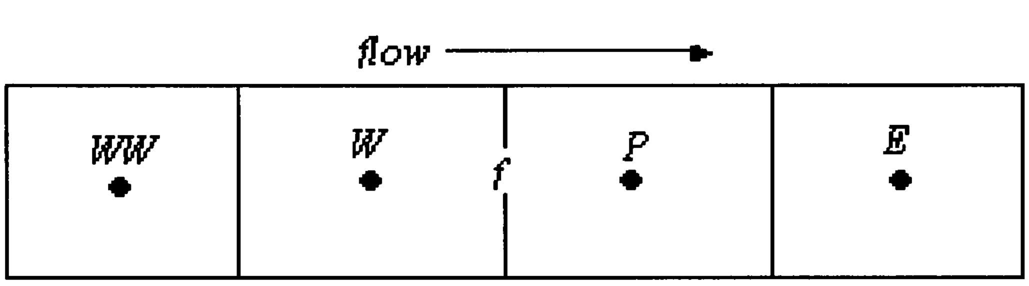


Figure 2.11: Three-point upwind schemes

2.5.5 Flow Oriented Schemes

The cell-centred higher order discretisation schemes can be reformulated into directional schemes. This is achieved by selecting the downstream point values as points located in the downstream flow direction and not necessarily upwind along grid lines. Referring to Figure 2.9, applying the upwind method, the face value is taken as the value at node A. A more representative face value would include some contribution from the value at a node upwind to the direction of flow, in this case node C. Upwind procedures designed to better account for the direction of the flow, thus

FROMM	$B(r) = 0.5((1+k)r + (1-k)$, $k = 0$
CUBIC	$B(r) = 0.5((1+k)r + (1-k)$, $k = 1/3$
QUICK	$B(r) = 0.5((1+k)r + (1-k)$, $k = 1/2$
SMART *(Bounded QUICK, positive)*	$B(r) = max(0, min(2 * r, 0.75 * r + 0.25, 4))$
KOREN *(Bounded CUBIC, TVD)*	$B(r) = max(0, min(2 * r, 2 * r/3 + 1/3, 2))$
VAN LEER *(Bounded FROMM, TVD)*	$B(r) = max(0, min(2 * r, 0.5 + 0.5 * r, 2))$
UMIST *(Bounded QUICK, TVD)*	$B(r) = max(0, min(2 * r, 0.25 + 0.75 * r,$ $0.75 + 0.25 * r, 2))$

Table 2.1: Waterson's generalization of $B(r)$

reducing false diffusion, tend to be computationally expensive and can potentially lead to spatially oscillating, physically unrealistic, distributions of the transported quantity. First-order flow-oriented schemes that attempt to follow streamlines, such as CUPID [128] applied to scalar convective transport and SUCCA [129] which possesses stability problems when applied to momentum transport, incorporate the influence of downstream corner cells into the discretisation process. Patel *et al* investigated a number of flow-oriented schemes for reducing 'false diffusion', [130].

Flow-oriented interpolation functions have been developed for the control-volume finite-element method. The most commonly used are the flow-oriented interpolation method (FLO) of Baliga and Patankar [100], the flow-oriented interpolation with source effects (FLOS) proposed by Hookey *et al* [131]. Both these methods employ interpolation functions that respond appropriately to the direction of the average velocity vector in an element and the element Peclet number (Pe). The FLO and FLOS schemes can lead to unbounded solutions especially when large velocity gradients exist within an element and when elements are distorted. Employing linear shape function profiles allows higher point schemes to be formulated. Numerical diffusion can be added in the flow direction using upwind first order principles but higher order accuracy can be maintained in the cross stream direction where no numerical diffusion is added. A mass weighted upwind scheme (MAW) was developed by Saabas and Baliga [101] for triangular and tetrahedral elements. This scheme has been employed on elements with sub-control volumes constructed of two internal faces, as shown is Figure 2.12. The value of ϕ on the face is estimated as follows,

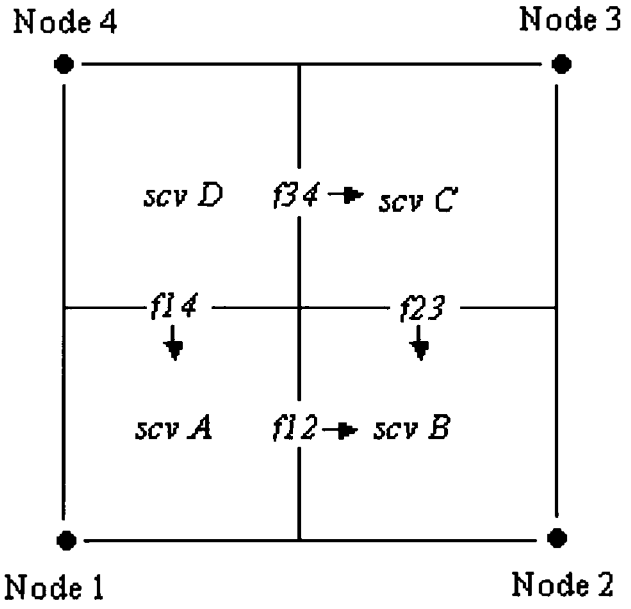


Figure 2.12: Implementation of weighted upwind scheme

For face (f_{12}) in Figure 2.12,

$$\begin{aligned} &\text{If } m_{f12} > 0 && \phi_{f12} = f\phi_{f14} + (1-f)\phi_1 \\ &\text{where} && f = min[max(m_{f14}/m_{f12}, 0), 1] \end{aligned} \tag{2.49}$$

$$\begin{aligned} &\text{If } m_{f12} < 0 && \phi_{f12} = f\phi_{f23} + (1-f)\phi_2 \\ &\text{where} && f = min[max(-m_{f23}/m_{f12}, 0), 1] \\ &\text{AND} && \\ & && \phi_f = \sum_{i=1}^{n} N_i\phi_i \end{aligned} \tag{2.50}$$

where the mass flow rate across a control volume face is estimated as:

$$m_f = \rho_f(\underline{u}.\underline{n})_f \tag{2.51}$$

This scheme maintains consistent expressions for ϕ_f and builds in transportiveness. The MAW scheme is usually sufficient to maintain positivity in the leading diagonal coefficients. The equations, (2.49) maintain that the convective transport of ϕ out of the sub-control volume is not greater than the convective transport of ϕ into the sub-control volume. This gives a rough approximation for the influence of the flow direction but the numerical predicted solutions still suffer from some degree of false diffusion.

2.6 Closure

The cell-centred and vertex-based discretisation methods have been presented in this chapter. On a non-orthogonal mesh the cell-centred method requires additional correction terms which can cause problems with convergence. The use of elemental linear shape functions for vertex-based discretisation allows the direct computation of fluxes in the required direction even on a non-orthogonal mesh. In the vertex-based approach control volumes are comprised of a larger number of faces and each solution point is connected to a greater number of neighbouring points. This extra discretisation requires additional computational effort and produces a positive-definite solution matrix. In the vertex-based discretisation of the transient and source terms care must be taken to ensure the 'boundedness' requirements. The upwind differencing scheme is the most stable of the convection schemes reviewed. The vertex-based hybrid method employs a higher point scheme when diffusion dominates. In strongly convective flows the scheme reverts to the upwind method which will introduce some degree of numerical diffusion. However, the stability of the higher order schemes on highly distorted meshes is questionable. In this investigation solutions are required on skewed meshes, the hybrid scheme will be employed due to its stability and robustness.

Chapter 3

Convection-Diffusion Test Cases

The aim of these test cases is to validate the vertex-based code for a transported scalar variable. Temperature is used as the transported quantity for all the test cases. The general equation governing heat transfer is,

$$\frac{\partial(\rho c T)}{\partial t} + div(\rho c \underline{u} T) = div(k \nabla(T)) + S_T \tag{3.1}$$

where ρ is the density, c is the specific heat and k is the thermal conductivity of the material.

The results obtained using the vertex-based approach are validated against analytical solutions. The transient-diffusion test case investigates the vertex-based technique used in the discretisation of the transient term. The results are shown for two cases, 1) Discretising the transient term over the vertex-based control volume, taking the average value at the mesh node. 2) Sum of the discretised transient term over the sub-control volumes, taking the average value at the centre of the sub-control volume. In the steady state heat conduction test case the mesh elements are arranged to give some non-orthogonal connections. The first case uses pentahedral elements and the second case hexahedral elements. The vertex-based results are compared with the results obtained from PHYSICA, which uses a cell centred approach. Results from PHYSICA are shown with and without the use of correction terms for mesh skewness on the pentahedral mesh. On the distorted hexahedral mesh no solution could be obtained using the correction terms as divergence was encountered. Lastly a convection only test case shows the degree of smearing that occurs when flow is at an angle to a Cartesian grid.

3.1 Transient Test Case

The test case for unsteady diffusion is transient temperature distribution on a thin plate. This is a one-dimensional unsteady conduction example, modelled taking the y and z directions equal to 1.

A thin plate of thickness $L = 2cm$, Figure 3.1, is initially at a uniform temperature of 200^0C. At a certain time the temperature of the east side of the plate is suddenly reduced to 0^0C. The other surface is insulated. The plate is modelled using a row of five hexahedral elements.

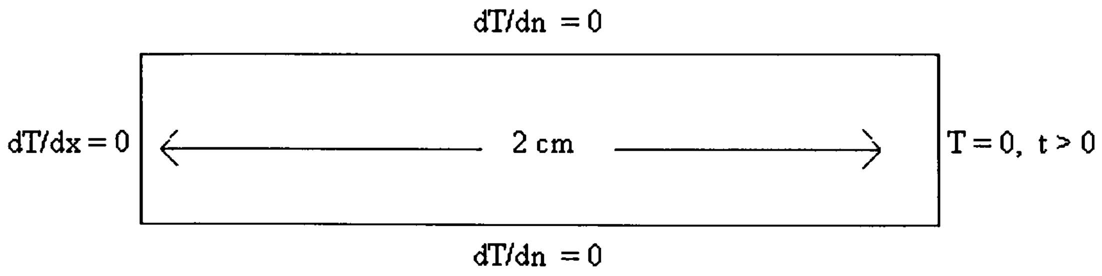


Figure 3.1: Transient heat conduction on a thin plate

The material properties are $k = 10W/m/K$ and $\rho c = 10^7 J/m^3/K$. The initial conditions are $T = 200$ at $t = 0$ and the boundary conditions are:

$$\begin{aligned} \frac{\partial T}{\partial x} &= 0 \quad at \quad x = 0, t > 0 \\ T &= 0 \quad at \quad x = L, t > 0 \end{aligned}$$

The analytical solution is given in Ozisik [133] as:

$$\frac{T(x,t)}{200} = \frac{4}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{2n-1}exp(-\alpha\lambda_n^2 t)cos(\lambda_n x) \tag{3.2}$$

$$\text{where } \lambda_n = \frac{(2n-1)\pi}{2L} \text{ and } \alpha = k/\rho c$$

Time-dependant problems require knowledge of the value of ϕ, in this case temperature, within a control volume. An average value, usually at the centre of the control volume, is chosen to represent the control volume temperature. If temperature is varying strongly over the solution domain a better approximation may be obtained by sub-dividing the control volume. Since in the vertex-based technique a control volume comprises of a number of sub-control volumes, the control volume temperature can be calculated as a sum of the sub-control volume temperatures. The most logical point to take as the average value is the centre of a sub-control volume, but any point within the sub-control volume could be chosen, Figure 3.2. This test-case compares the results using the two discretisation methods:

Method 1: Discretising the transient term as,

$$\frac{\rho_{node}c_{node}T_{node}V - \rho_{node}^0 c_{node}^0 T_{node}^0 V_0}{\Delta t} \tag{3.3}$$

This method takes the nodal value to be a representative value for the control volume.

Method 2: Discretising the transient term as,

$$\frac{1}{\Delta t}\sum_{scv}\sum_{i=1}^{n}(\rho_{scv}c_{scv}V_{scv}N_iT_i - \rho^0_{scv}c^0_{scv}V^0_{scv}N_iT^0_i) \tag{3.4}$$

This method takes a representative value as a weighted average of values at the centre the sub-control volumes.

In cell-centred techniques the element centroid value is taken to be a representative average value.

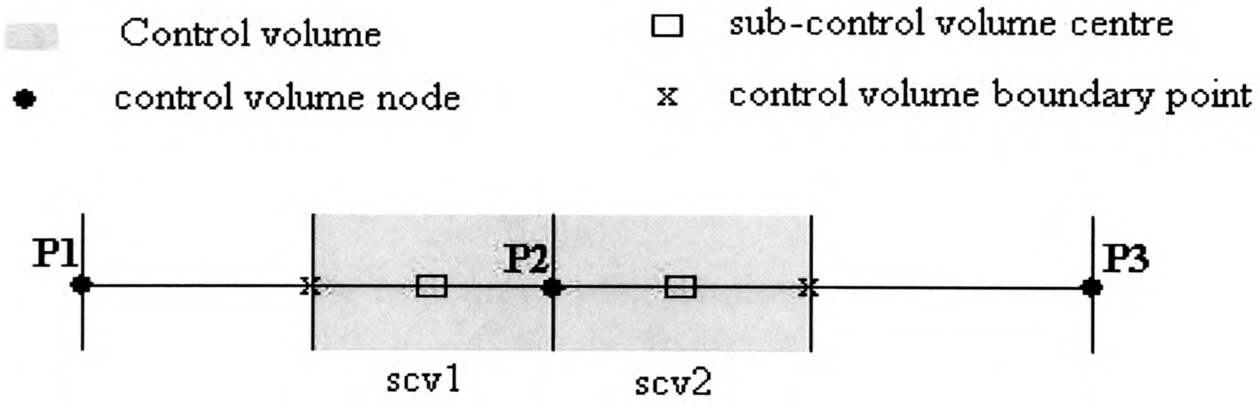


Figure 3.2: Control volume value represented by nodal value or an average of scv values

The domain was divided into a number of equal sized elements and simulations were carried out employing various time step sizes. It was found that method 2 produced unbounded results when the contribution from the transient term (T) was greater than the diffusion term (D),

$$\frac{T}{D} > 1 \tag{3.5}$$

where,

$$\begin{aligned} T &= \frac{\rho c V_{scv}}{\Delta t} \\ D &= \frac{k}{dx} \end{aligned}$$

giving the condition for bounded solutions as:

$$\frac{\rho c V_{scv} dx}{\Delta t k} < 1 \tag{3.6}$$

On investigation it was found that although method 2 produced a positive definite system of equations, the system became ill-conditioned if the transient term dominated, i.e. $\sum a_{nb} < 0$. In the simulations undertaken, when Equation (3.6) was satisfied, method 2 produced more accurate results on a coarser mesh than method 1.

Method 1 does not suffer from unbounded solutions. Coupling of the transient term with the diffusion term, using method 1, produces a positive-definite system which is well conditioned in all cases investigated.

To illustrate the problem the domain is divided into five equal sized elements and simulations carried out using a time step of 1 second and 2 seconds giving $T/D > 1$ and $T/D < 1$ respectively. The simulations were run for 40 seconds and the results are shown along with cell-centred and analytical solutions. The nodal values from cell-centred solutions are extrapolated from element based results. Figure 3.3 to 3.7 shows the results obtained at 5 locations, $x = 0$, $x = 0.004$, $x = 0.008$, $x = 0.012$ and $x = 0.016$. For a time step of 2 second, method 2 gives results which are in good agreement with the analytical solution at all solution points, with a maximum error at $x = 0$ of 0.07%. Method 1 gives similar results to cell-centred solutions with a maximum error at $x = 0$ of 2.25%. Employing a time step of 1 second method 2 gives unbounded solutions at $x = 0.012$ and $x = 0.016$ in the early time steps. The cell-centred and method 1 solutions remain unchanged. If the mesh is refined T/D reduces, the stability of method 2 is ensured and smaller time steps can be employed. Refining the mesh reduces the error in the results obtained using method 1 or cell-centred solutions. Figure 3.8 gives the results for $x = 0$ on a mesh comprising of 40 elements. The percentage error with the analytic solution has reduced to 0.45%

In cases where $T/D < 1$, method 2 appears to give more accurate results. The sub-division of the transient term links the rate of change in a control volume to the rate of change in neighbouring control volumes. For the control volume shown in Figure 3.2, assuming $\rho_1 = \rho_2 = \rho_3$ and $V_1 = V_2 = V_3$, Equation (2.21) gives the discretised transient term for P2 as,

$$\frac{1}{\Delta t}\rho V\left((N_1 + N_2)(\phi - \phi^0)_2 + N_1(\phi - \phi^0)_1 + N_2(\phi - \phi^0)_3\right) \tag{3.7}$$

thus improved communication is obtained between solution points. Method 2 allows the control volume temperature to be more accurately represented when temperature does not vary linearly throughout the solution domain. Again referring to Figure 3.2 and assuming linear variation between solution points, at time $t - 1$ let P1=90, P2=80 and P3=60, the temperature gradient between P2 and P3 is greater than the temperature gradient between P1 and P2. In this case taking an average of sub-control values will give a more representative control volume average than the value at node P2.

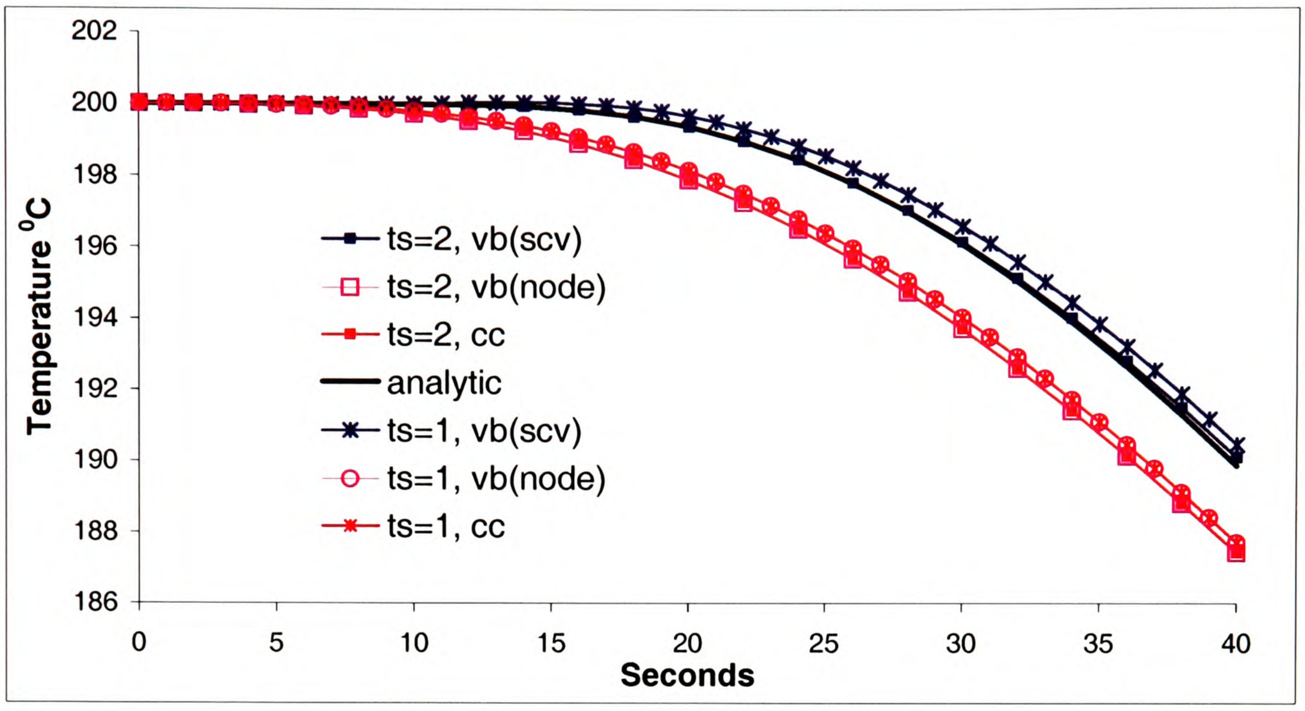


Figure 3.3: Temperature over 40 seconds at $x = 0$

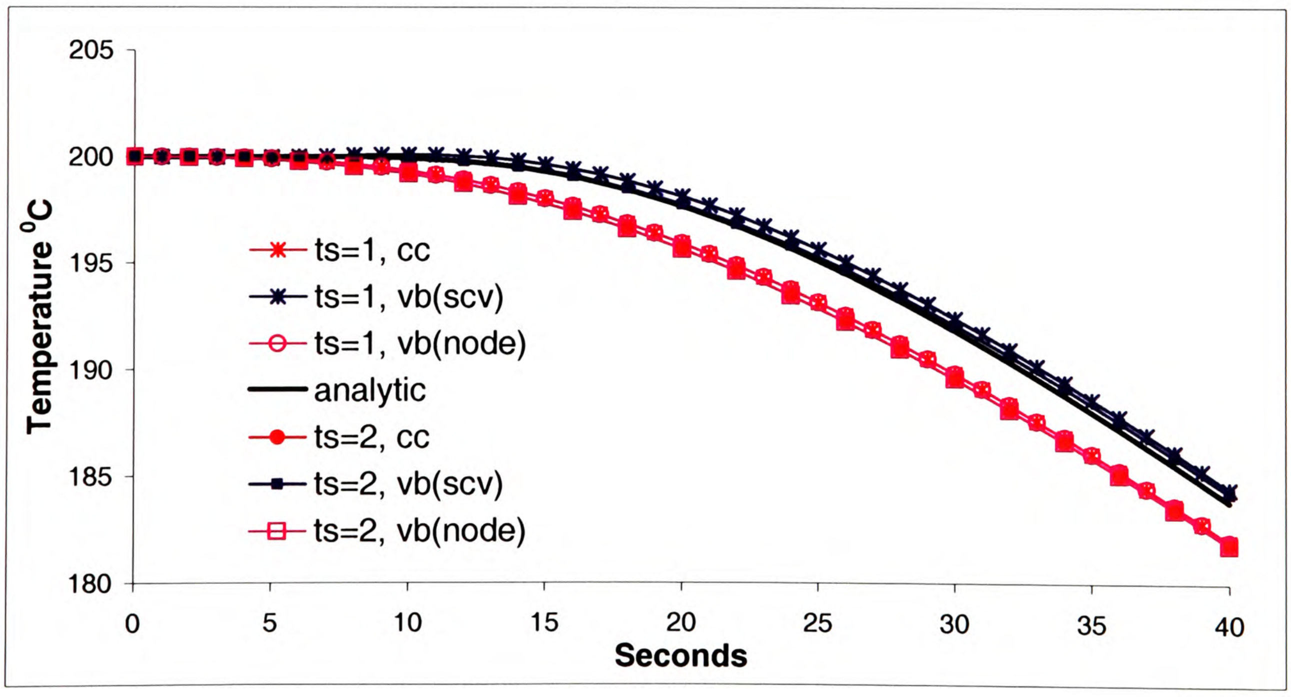


Figure 3.4: Temperature over 40 seconds at $x = 0.004$

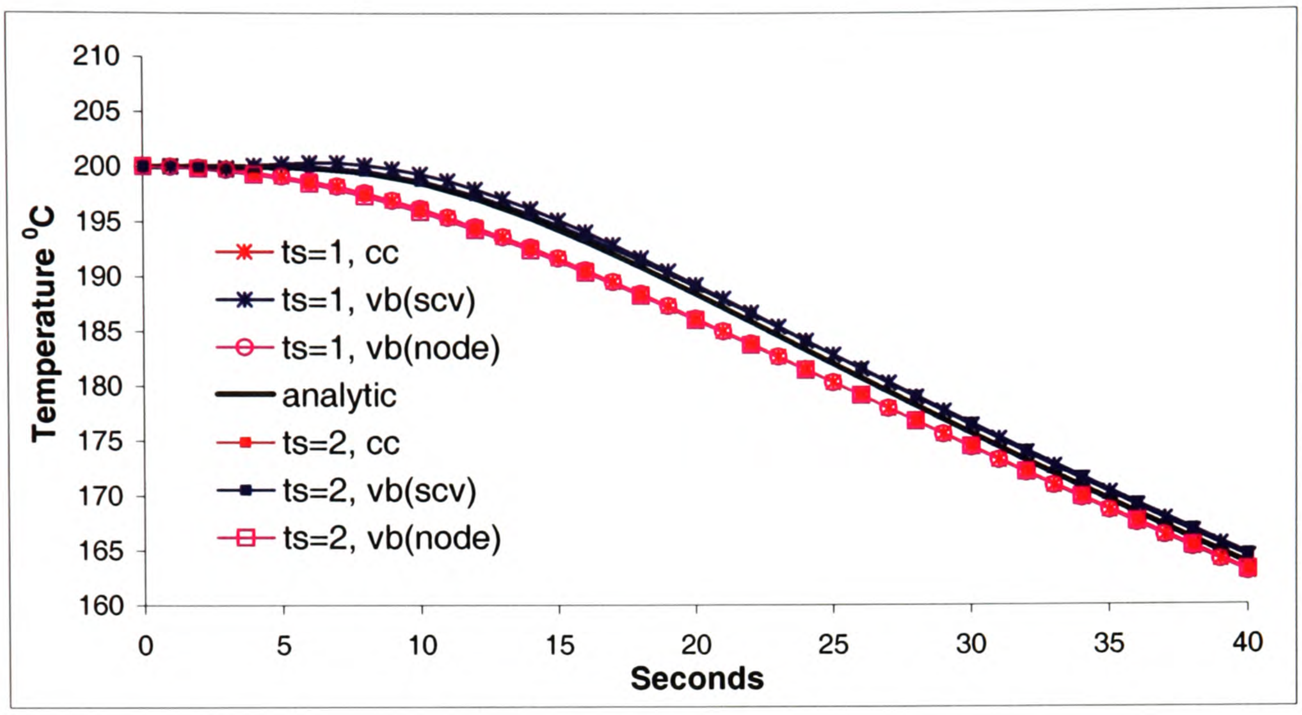


Figure 3.5: Temperature over 40 seconds at $x = 0.008$

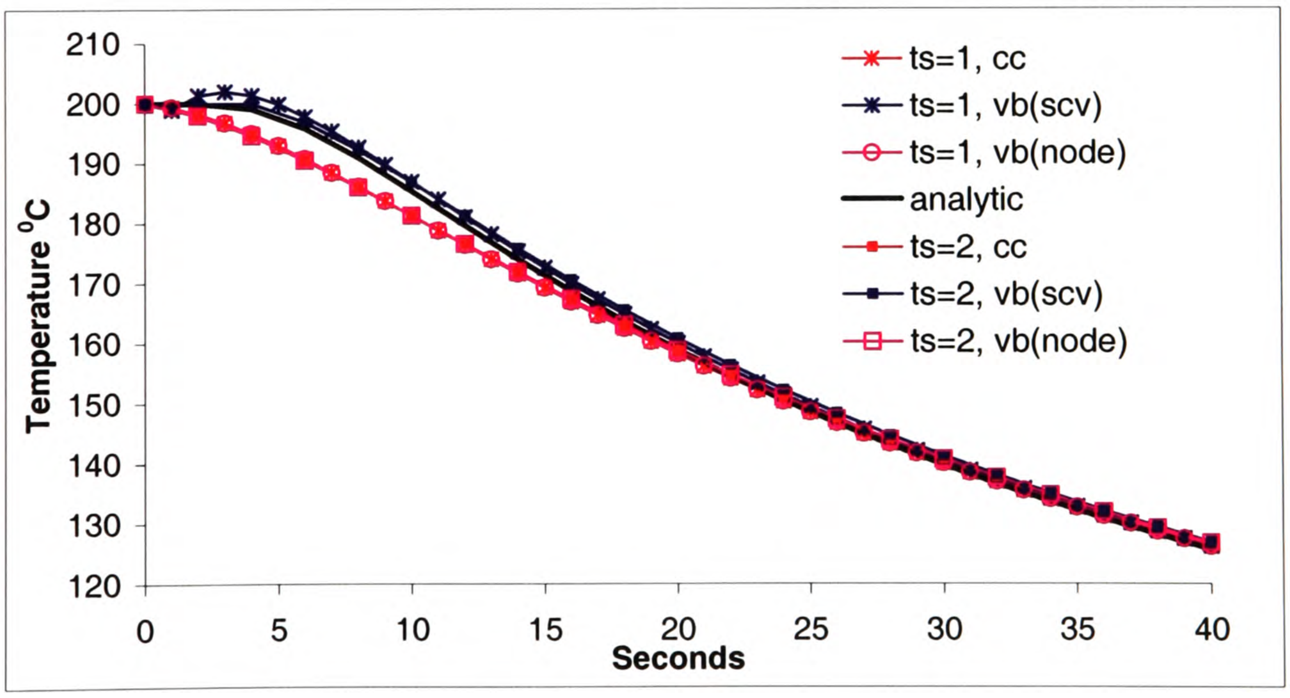


Figure 3.6: Temperature over 40 seconds at $x = 0.012$

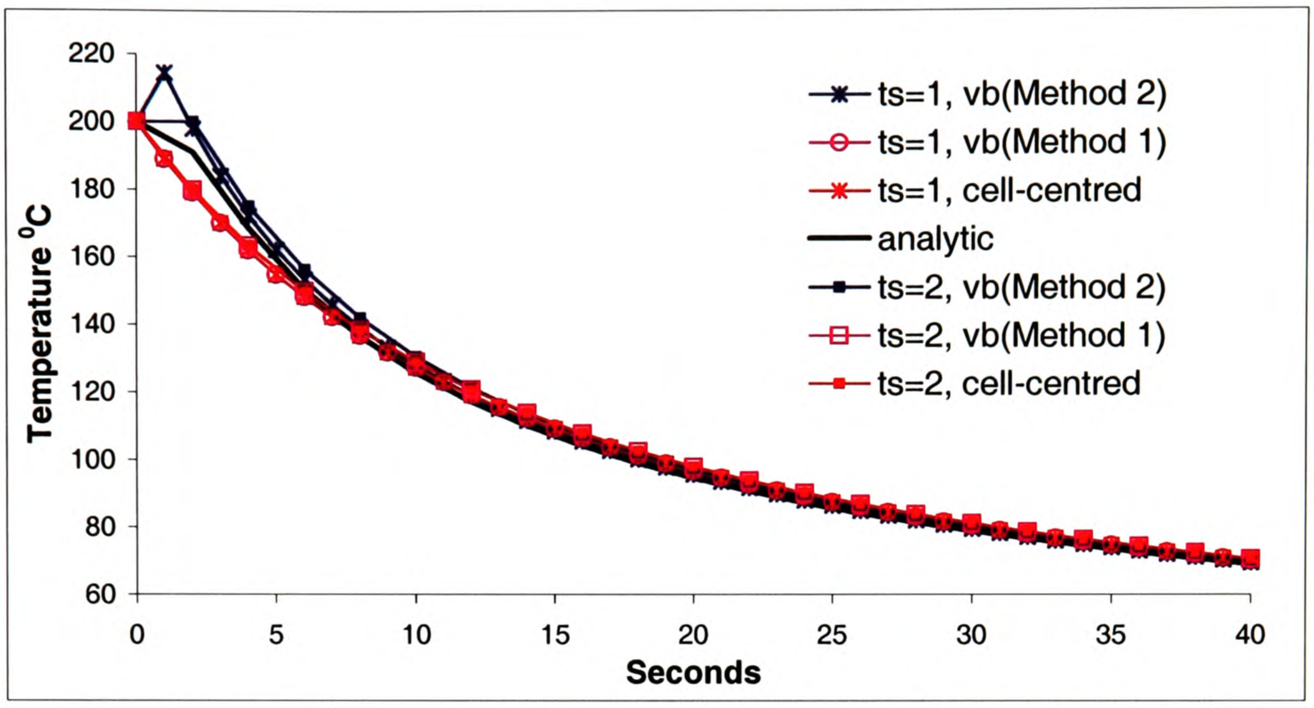


Figure 3.7: Temperature over 40 seconds at $x = 0.016$

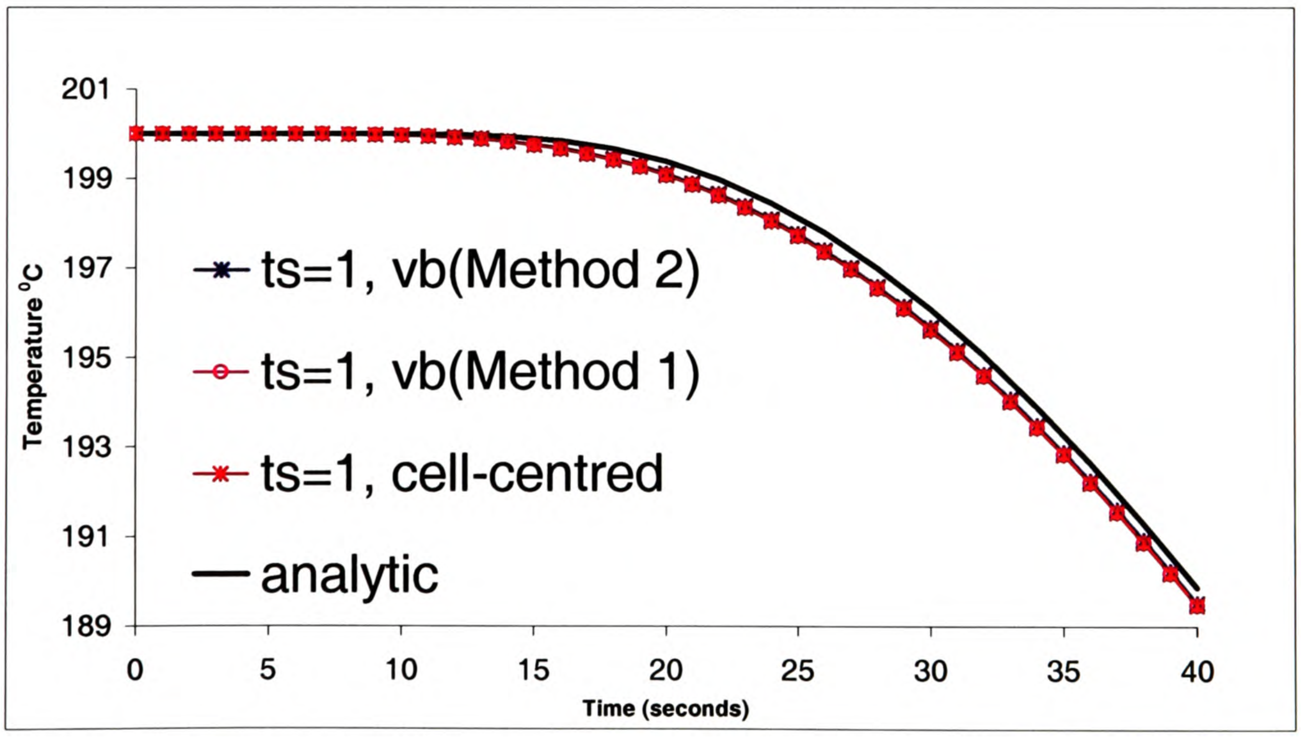


Figure 3.8: Temperature at $x = 0$ on a 40 element mesh

3.2 Steady-State Diffusion

The test case used for steady-state diffusion is simple one-dimensional heat conduction in a three-dimensional cube. The cube is meshed using pentahedral elements, with elements arranged to give some non-orthogonal connections, as shown in Figure 3.9. This case illustrates the error in the solution when cell-centred techniques are used with no non-orthogonal correction terms, and the ability of the vertex-based technique to handle non-orthogonal meshes.

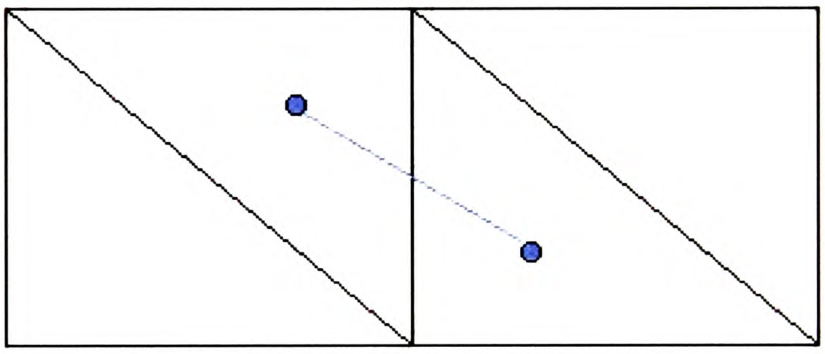


Figure 3.9: Mesh with some degree of non-orthogonality

One side of the cube is at a temperature of 0^0C, the opposite side is 1^0C. The other sides are adiabatic. The dimensions of the cube are $1m^3$. The analytic answer is simply a function of the x distance, $T = 1 - x$. Taking a cross-section of the cube at a distance $0.6m$ from the side at 0^0C the analytical solution throughout the cross section is a constant temperature of 0.6^0C. The graph in Figure 3.10 shows the results obtained using the vertex-based approach and the cell-centred approach, both with and without the correction terms for mesh skewness. The solution obtained using cell-centred techniques, without the correction terms for mesh skewness, took 350 solver iterations to converge but appears to have some dependency on the y direction. Using the correction terms gives exact solutions. As cell-centred solutions are not obtained at the boundary, boundary values have been extrapolated from element values, this has given rise to the error in the plotted boundary results. The number of iterations required for convergence increased to 500 solver iterations. The results obtained using the vertex-based approach are exact throughout and the solution converged in 100 solver iterations.

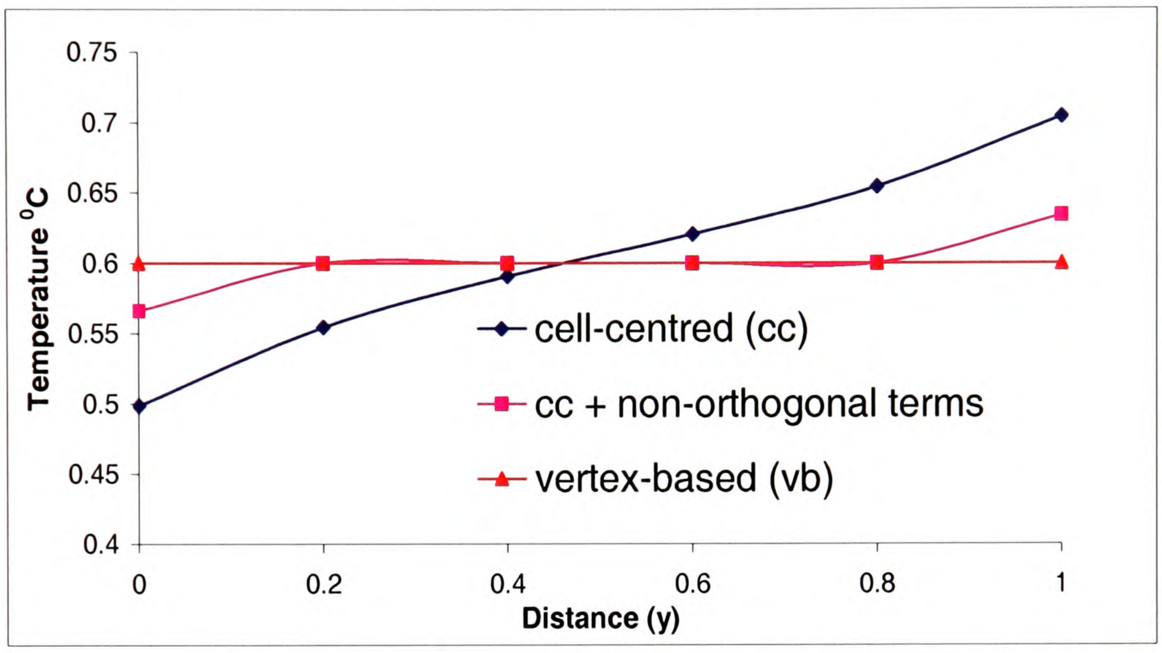


Figure 3.10: Cross section: Temperature results for heat transfer on non-orthogonal mesh

The cell-centred simulation results illustrate the error in the discretisation of the diffusion term caused by the absence of correction terms. Inclusion of non-orthogonal and conjunctionality corrections are required to obtain solutions matching analytic results.

The simulation was repeated on a mesh of arbitrary skewness, Figure 3.11. Inclusion of the cell-centred non-orthogonal terms resulted in divergence and solutions failed. The cell-centred solutions shown in Figure 3.12 do not include any correction terms. Figure 3.12 shows the temperature values along the central horizontal line of the cube. Again the vertex-based solutions performed well, giving 1.9% maximum percentage error at a distance $x = 0.5$. Without the inclusion of correction terms the cell-centred solutions were poor, obtaining a maximum percentage error of 15.5%

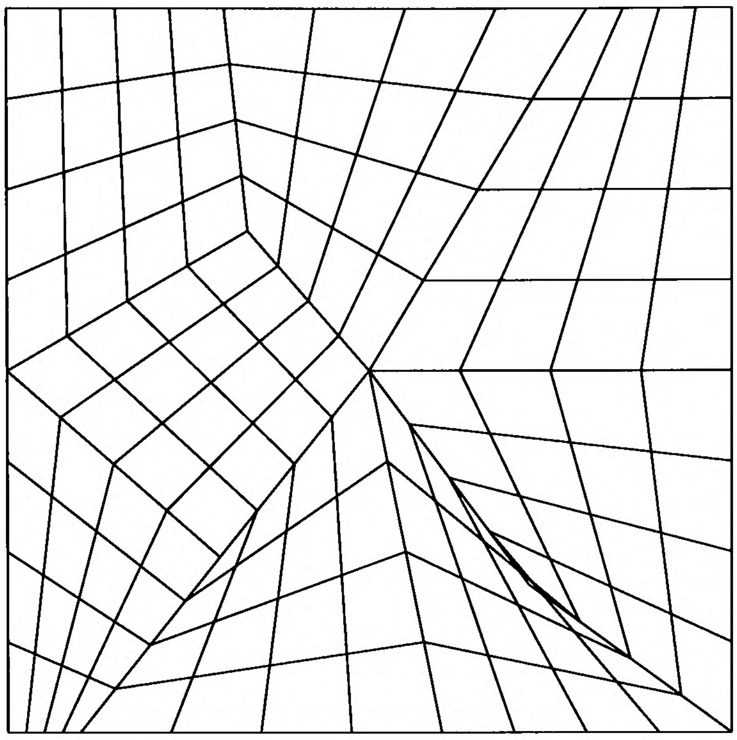


Figure 3.11: Arbitrary distorted mesh

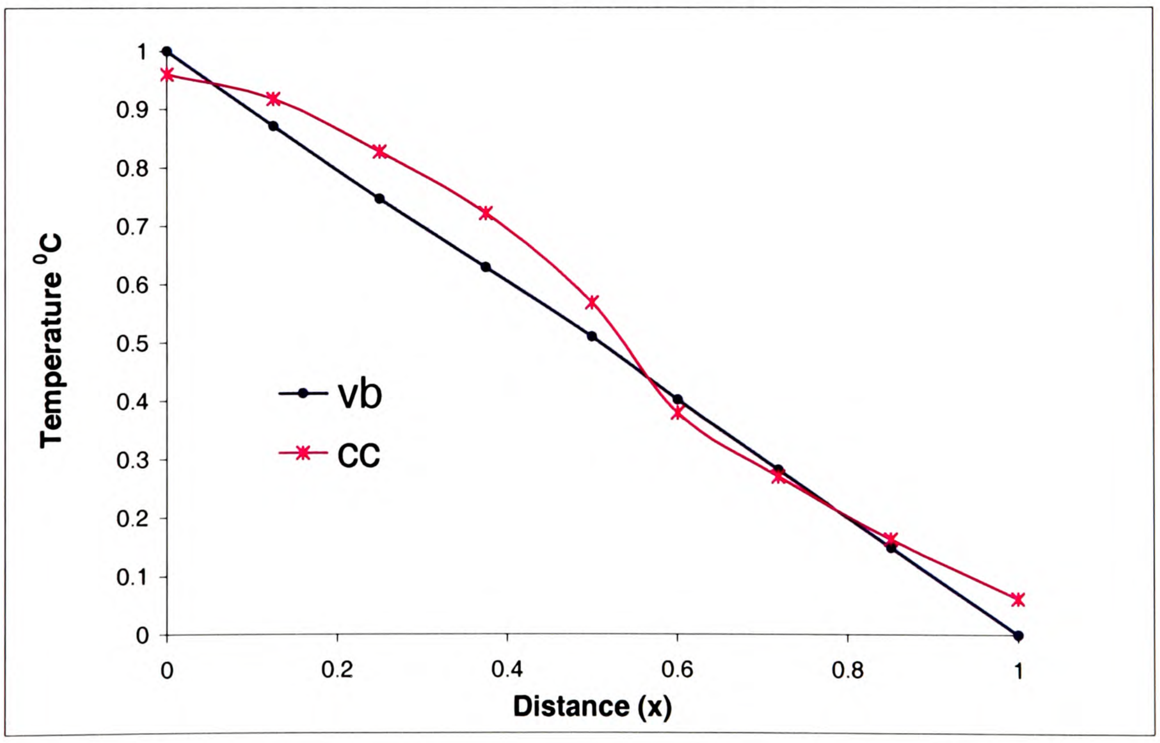


Figure 3.12: Temperature results on arbitrary distorted mesh

x-distance (cm)	analytic	cc	vb
0	1.0	0.96 (3.95%)	1.0 (0.0%)
0.125	0.875	0.918 (4.90%)	0.871 (0.46%)
0.25	0.75	0.828 (10.35%)	0.747 (0.45%)
0.375	0.625	0.722 (15.47%)	0.629 (0.63%)
0.5	0.5	0.569 (13.72%)	0.510 (1.90%)
0.601	0.399	0.380 (4.84%)	0.402 (0.68%)
0.719	0.281	0.271 (3.73%)	0.282 (0.42%)
0.851	0.149	0.164 (9.85%)	0.149 (0.24%)
1.0	0.0	0.0627 (6.27%)	0.0 (0.0%)

Table 3.1: Solutions and percentage error with analytic solutions on a arbitrary skewed mesh.

at a distance $x = 0.375$. The solutions and analytic results are shown in Table 3.1. The percentage error with respect to the analytical solutions are given in brackets.

In many meshes of arbitrary skewness inclusion of mesh correction terms leads to diverging solutions. Longer convergence times are encountered on meshes where solutions are obtained using correction terms. In the vertex-based technique the diffusion term is discretised directly, each node is linked to a greater number of neighbouring nodes. Communication is increased between solution points and information is spread faster throughout the solution domain. This improved communication leads to less solver iterations required to convergence but increased computational time per iteration.

3.3 Convected Scalar

This test case involves convection of a scalar in a domain where the flow is not aligned with the grid lines. The upwind differencing scheme is used to discretise the convection term in both the vertex-based and cell-centred code. The resulting amount of numerical diffusion due to the upwind scheme is compared for both methods on a Cartesian grid. To identify the false diffusion a pure convection process is considered without physical diffusion. There are no source terms and a steady state solution is sought. The same case is then solved on a skewed mesh to illustrate the stability of the upwind scheme when solving on highly distorted meshes.

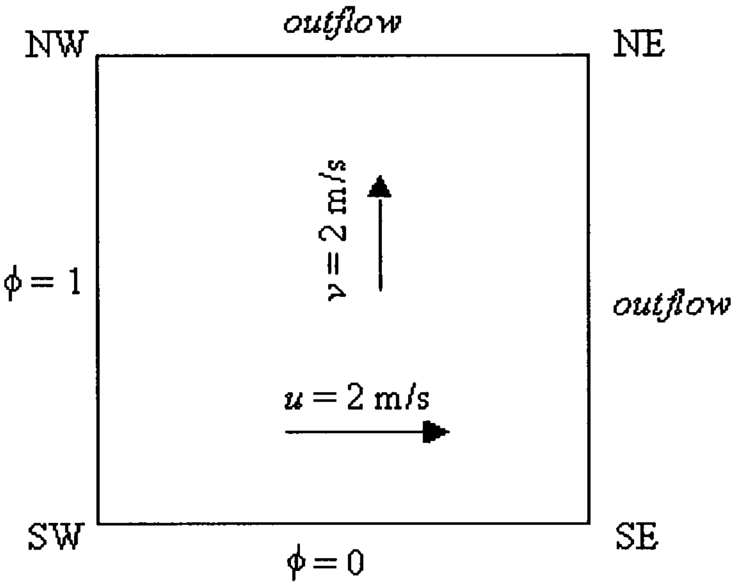


Figure 3.13: Flow domain for pure convection

The solution domain is a $1m^3$ cube, meshed with hexahedral elements, and $u = v = 2ms^{-1}$ everywhere so that the velocity field is uniform and at an angle to the grid lines. The boundary conditions are $\phi = 0$ along the south boundary and $\phi = 1$ along the west boundary. At the south-western node where the boundaries intersect, a value of 0.5 is assigned to the property ϕ. The north and east boundaries are open outflow boundaries. The flow domain is illustrated in Figure 3.13. Since there is no physical diffusion the exact solution exhibits a step change of ϕ from 0 to 1 along the diagonal SW to NE.

The degree of false diffusion along the cross flow diagonal of the domain (NW to SE) is shown in Figure 3.14 for meshes containing 100 elements and 2500 elements. The results from the vertex-based and cell-centred method are similar. Grid refinement reduces the numerical diffusion, but the degree of grid refinement required to eliminate the error would not be computationally practical. The number of solver iterations required to achieve convergence when using the vertex-based method was 800 on a 10x10 element mesh and 1200 on a 50x50 element mesh. Cell-centred solutions required significantly more iterations, 4000 and 20000 on the same meshes. The convection only simulation was also run on a highly skewed mesh to test the stability of the upwind scheme when the mesh is distorted. Vertex-based and cell-centred solutions gave similar results, Figure 3.15 shows the temperature contours obtained.

3.4 Closure

The transient-diffusion test case illustrated the problem that can be encountered when discretising the transient term using Equation (2.21). In cases were the boundedness criteria is satisfied, employing vertex-based discretisation, Equation (2.21), can considerably improve the transient solution on a coarse mesh. In order to ensure stable and converging solutions in the remainder of this research the transient term will be discretised using Equation (2.19). The ability of the vertex-based method to handle distorted meshes was shown in the diffusion test case. Solving cell-centred on a distorted mesh without non-orthogonal corrections degrades the solution. The inclusion of the correction terms improves the solution but divergence is often encountered. The upwind differencing scheme was shown to handle distorted meshes well, but suffers from numerical diffusion. The vertex-based discretisation, although computationally more expensive, required considerably less solver iterations than cell-centred solutions.

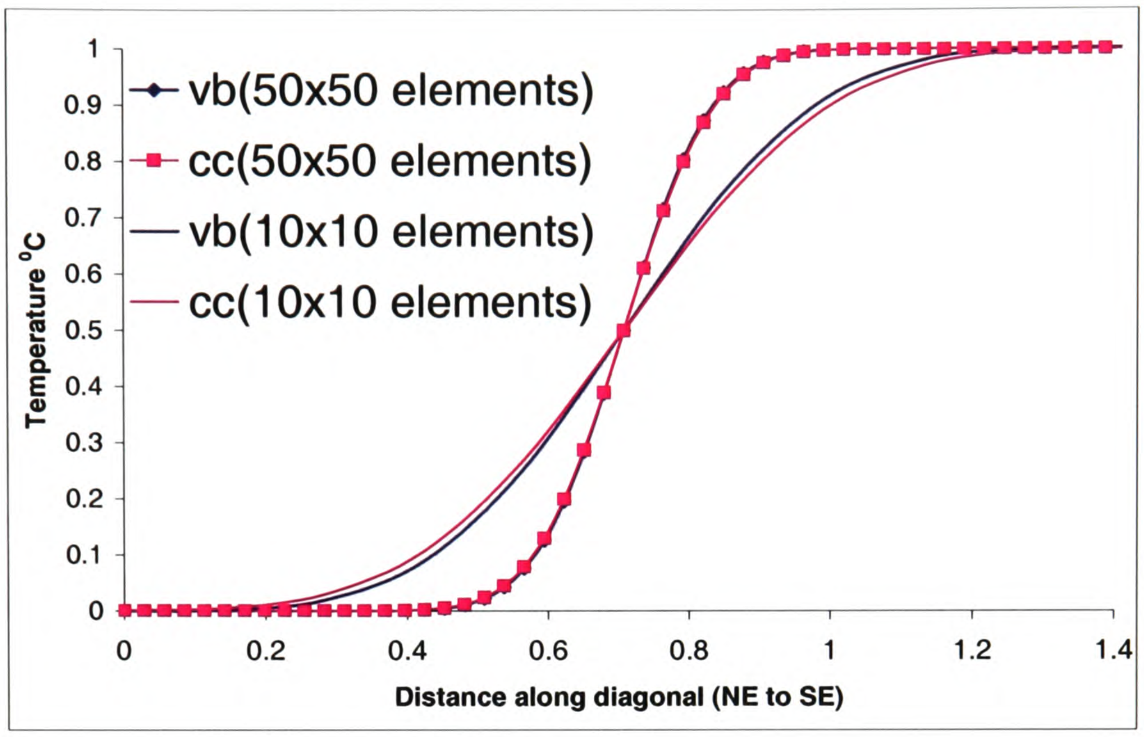


Figure 3.14: Temperature values across domain

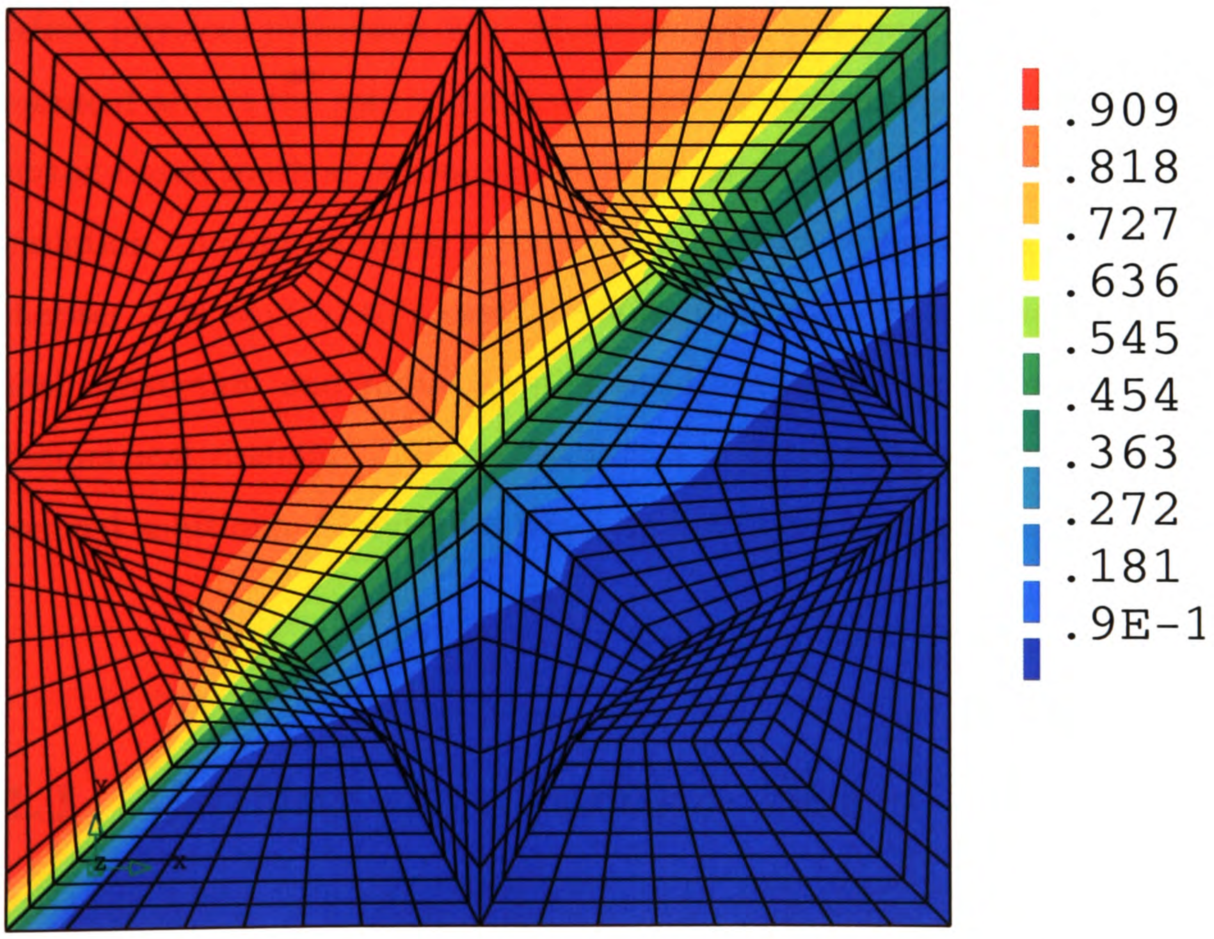


Figure 3.15: Upwind convection on a Skewed Mesh

Chapter 4

Coupled Physical Phenomena

The objective of this chapter is to apply the vertex-based discretisation technique to a complex multi-physics heat transfer problem. The vertex-based method is embedded within the modular framework of PHYSICA and coupled with other well established physical processes, e.g. phase-change, thermal stress. The PHYSICA environment allows the user to implement additional physics through user routines. The vertex-based method is coupled with the user module to allow additional physics, i.e radiation, moving boundary, source terms. The vertex-based technique, outlined in chapter 2, is applied to a case involving heat transfer by conduction, losses by natural convection and radiation plus solidification phase-change in a multi-material geometrical domain. The vertex-based code is validated against PHYSICA cell-centred solutions. Simulation times and memory requirements are compared.

4.1 Casting of a Turbine Blade

Turbine blades located in engine hot sections have to withstand extreme temperatures. The process of casting the turbine blade is critical to the strength and stability of the finished blade. The casting and solidification process affects the pattern and type of grain growth, shrinkage and porosity. Many turbine blades rely on internal cooling passages to circulate air, cooling the metal in the turbine hot sections. Ceramic cores, installed during the creation of the mold, form the interior surfaces of the blade during the casting process. An outer ceramic shell is constructed in layers to a uniform thickness, ensuring even thermal conductivity throughout the solidification of the alloy. Filling the mold in a vacuum keeps impurities from affecting the properties of the alloy. Creep is the principal failure of a blade, occurring ultimately at the grain boundaries. The blade is weakest when grain boundaries are normal to the stress axis. The strength of turbine blades has been improved with the development of directionally solidified casting. Directional solidified casting restricts the path of grain growth so that all the grain boundaries are parallel to the stress axis, thus improving the strength and life of the blade. Both the mold and the metal are kept at similar temperatures. A hot mold, above the melting point of the alloy, sits on a water-cooled copper chill. The chill plate and mold are withdrawn from the hot zone into the cold zone. The water-cooled cooper chill extracts heat energy ensuring the alloy solidifies from the bottom up.

Simulating the casting of a turbine blade requires the coupling of several physical phenomena. The process involves heat transfer by conduction and radiation, phase change and non-linear solid mechanics. The assembly is made up of four components, a ceramic shell and core, an alloy blade and copper chill. Initially the assembly is in a furnace with radiative heat transfer between the shell and the furnace. The assembly is withdrawn from the furnace at a speed of 6.35×10^{-5} m/s. A moving boundary condition, employing an ambient temperature profile simulates the withdrawal of the assembly from the furnace. As the assembly is withdrawn the combined effect of the moving boundary condition and the water-cooled copper chill cools the geometry. During the cooling process the alloy blade starts to solidify and a gap is formed between the shell and alloy. When a gap is formed it is assumed that a vaccum exists and heat is transferred across the gap by radiation. The contraction of the metal causes thermal stress in the blade. The geometry specifications, material data and boundary conditions are given in the following sections. The equations employed to describe the physical processes are given in section 4.2. The simulation results for a thermal only and a coupled thermo-mechanical analysis are shown in section 4.3. The results are shown for cell-centred and vertex-based thermal analysis. The mechanical analysis uses vertex-based techniques.

4.1.1 The Geometry

The core is assumed to be located centrally in the y-direction, thus allowing a symmetric mesh to be employed. The symmetric geometry is meshed using hexahedral elements. The total number of nodes is 53647 and the total number of elements is 45156. The main features of the four components of the assembly are shown in Figures 4.1 to 4.4.

The core, Figure 4.1, is enclosed within the metal blade geometry, Figure 4.2. The bottom of the core is located 90mm from the bottom of the metal blade and sits at the level before the blade begins to contract. Connected to the bottom of the metal blade is the copper chill, Figure 4.3. The shell shown in Figure 4.4 surrounds the metal blade and copper chill with a width of approximately 7mm.

4.1.2 Material Data

The material data for the thermal analysis are given in Tables 4.1 for the ceramic (core and shell), copper (chill) and Table 4.2 for the alloy (blade). For thermo-elastic analysis the Young's modulus and Poisson ratio for the alloy blade are given in Table 4.3 and 4.4 respectively, values are temperature dependant and use piecewise linear assumptions between data points. All other materials are given constant material

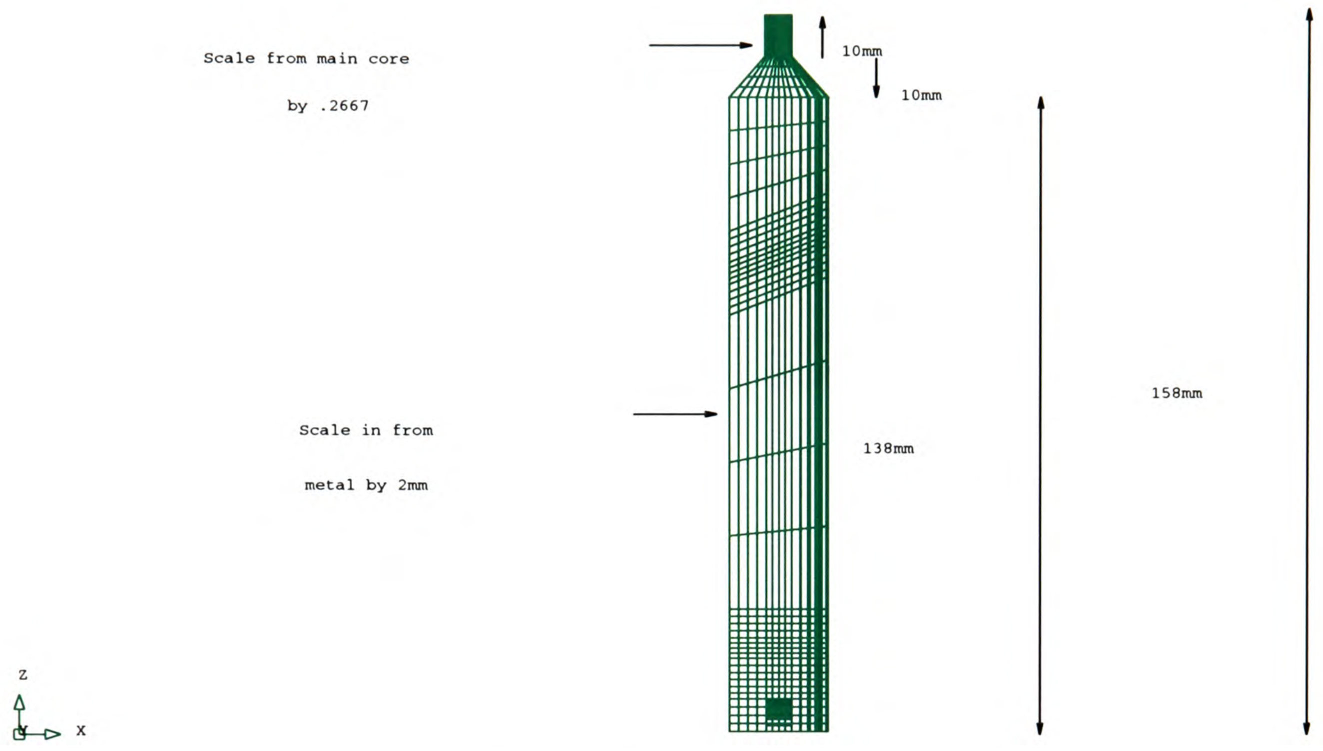


Figure 4.1: The core geometry

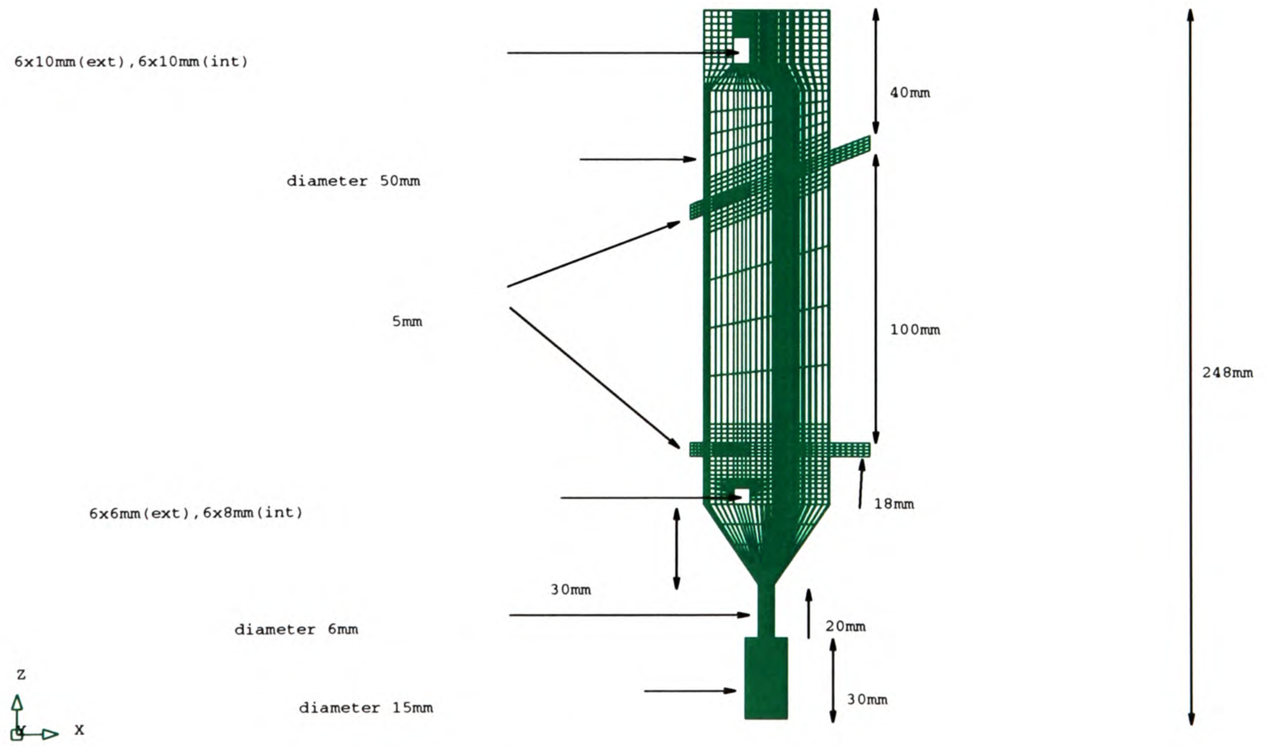


Figure 4.2: The blade geometry

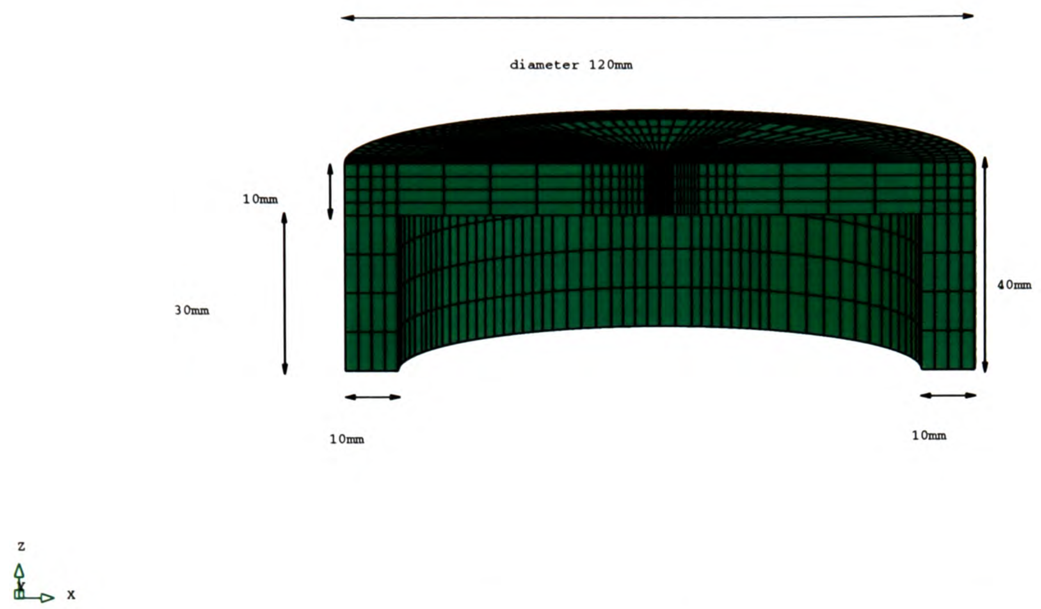


Figure 4.3: The copper chill geometry

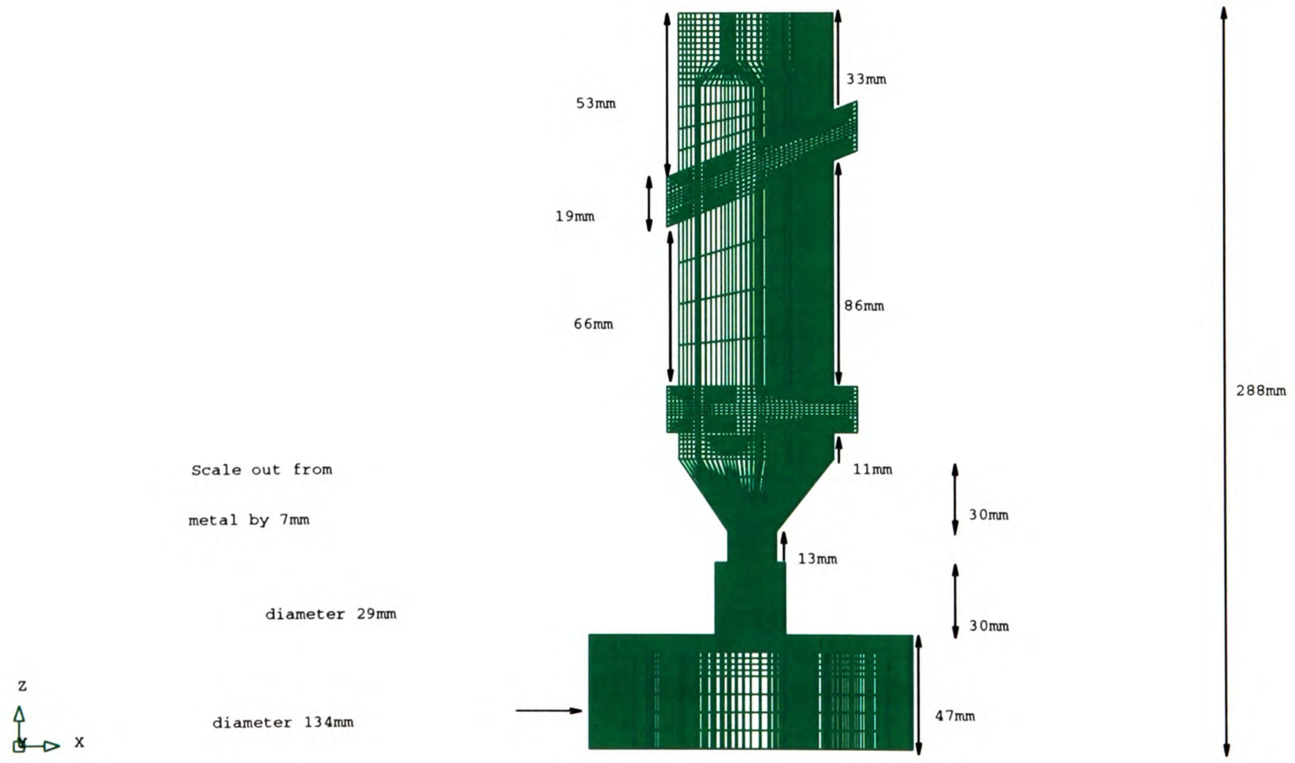


Figure 4.4: The shell geometry

assumptions of Young's modulus = 1.0×10^6 Pa and Poisson's ratio = 0.3. The thermal expansion coefficient for the alloy blade is $1.5 \times 10^{-5} K^{-1}$ and zero for all other materials. All materials are assumed to remain elastic throughout the solution and a value of 1.0×10^{20}Pa was set for the yield stress.

	Ceramic	Copper
Density (kg/m^3)	3400	8960
Thermal Conductivity (W/mK)	2.2	401
Specific Heat (J/kgK)	1000	385

Table 4.1: Thermal data for ceramic and copper

Density	$8200 kg/m^3$
Thermal Conductivity	$30 W/mK$
Specific Heat	$770 J/kgK$
Latent Heat	$2.65 \times 10^5 J/kg$
Solidus Temperature	$1673 K$
Liquidus Temperature	$1674 K$

Table 4.2: Material data for alloy

4.1.3 Boundary Conditions

The initial temperature of the ceramic shell and core, and alloy blade was set to $1773K$ and the copper chill to $293K$. Newton cooling type boundary conditions were applied to the base of the shell and the external face of the chill. The heat transfer coefficient for the shell base was $10W/mK$ and for the chill $6000W/mK$.

A radiative boundary condition was applied externally to the shell. The ambient temperature of the geometry still in the furnace should be $1773K$ and of the geometry not in the furnace should be $293K$. The chill/alloy interface is located at

Temperature (K)	Young's Modulus (Pa)
0.0	2.1×10^{11}
400	1.9×10^{11}
1000	2.0×10^{10}
1673	2.0×10^{9}
1674	1.0×10^{6}
3000	1.0×10^{6}

Table 4.3: Young's modulus for alloy

Temperature (K)	Poisson's Ratio
0	0.3
500	0.3
1673	0.49
3000	0.49

Table 4.4: Poisson's ratio for alloy

$z = -0.11m$. A linear ambient temperature profile is initially applied to the region $-0.258 < z < -0.11$ and moves at a speed of $6.35 \times 10^{-5} m/s$ to simulate the withdrawal of the geometry from the furnace. The ambient temperature, T_A, is given as a function of position and time,

1. The ambient temperature T_A above the initial chill/alloy interface, i.e the geometry still in the furnace, should be $1773K$, $(z > z_a, T_A = 1773)$
2. The ambient temperature T_A of the geometry not in the furnace should be $293K$, $(z < z_b, T_A = 293)$
3. Linear profile for $z_b \leq z \leq z_a$, $T_A = 293 + \frac{z-z_a}{z_b-z_a}(1773 - 293)$

where $z_a = a + V_e t$, $z_b = b + V_e t$, V_e is the extraction speed, 6.35×10^{-5}, t is time (seconds), $a = -0.11m$ and $b = -0.258m$.

For the thermo-mechanical analysis the shell, core and chill were fixed with only the alloy being allowed to deform. It was assumed that the alloy and core materials remained attached together with x, y and z being fixed at the interface. The movement of the alloy within the shell was restricted by fixing the values of z. These assumptions assume that the alloy would only deform in the x and y direction and no gap would develop between the alloy and core. In reality the alloy may deform in all directions and may become detached from the core.

4.2 Models of Physical Processes

The equations used to model the physical processes that occur during the casting of a rotary blade are given in this section. The PHYSICA modules for solidification and solid mechanics are employed in the simulation. The modules have been slightly adjusted to allow coupling with vertex-based heat transfer solutions. The vertex-based adjustments are given in the relevant section. The user module was employed to implement heat transfer by radiation when a gap formed between the ceramic shell and alloy blade.

4.2.1 Solidification

The PHYSICA solidification module employs a source-based method [134, 135]. During the phase change of a material energy is released. The evolution of the latent heat of solidification is accounted for by definition of a source term in the equation governing heat transfer (3.10). The source term for non-convection problems is,

$$-L\frac{\partial(\rho f)}{\partial t} \tag{4.1}$$

where L is the latent heat of solidification and f is the liquid fraction.

The principle behind the source-based method is to allow the phase change to be identified by a local liquid fraction f that keeps track of the phase change as it moves through the solution domain. The advantage of this approach is that it allows a solution without the need for a mesh that moves with the solidification front. The liquid fraction is assumed to be a function of temperature $f = F(T)$, however liquid fraction could be a function of a number of variables. In the evaluation of the liquid fraction a linear relationship is assumed giving,

$$f = \begin{cases} 1 & : \; T > T_L \\ (T - T_S)/(T_L - T_S) & : \; T_S \leq T \leq T_L \\ 0 & : \; T < T_S \end{cases} \tag{4.2}$$

where T_L is the temperature above which the material is in liquid state, and T_S is the temperature below which the material is in solid state. This system contains a linear transition phase, a solid/liquid region over which the latent heat of solidification evolves. In the transition from liquid to solid state a large amount of energy can be released causing the temperature to rise and an oscillating liquid-solid-liquid cycle can be encountered. In order to avoid oscillations a correction Δ is applied to update the liquid fraction f. A number of corrections have been proposed, [134] to [137]. The Voller-Prakash [137] and Voller-Brent [134] methods calculate a correction equal to,

$$\Delta = [T - F^{-1}(f)]/\delta h \tag{4.3}$$

where the equation for δh are as follows,

$$\text{Voller Prakash} \qquad \delta h = \frac{L}{c} + T_L - T_S \tag{4.4}$$

$$\text{Voller Brent 1} \qquad \delta h = \frac{L\rho V}{Ta_P \Delta t} \tag{4.5}$$

$$\text{Voller Brent 2} \qquad \delta h = \frac{L\rho V}{Ta_P \Delta t} + T_L - T_S \tag{4.6}$$

where c is the specific heat and Ta_P is the diagonal cell of the discretised temperature equation.

The Voller and Prakash method underestimates the correction and convergence can be slow, but it has the advantage of requiring less under-relaxation than the other methods. The Voller and Brent methods are faster, method 2 is the fastest but frequently requires under-relaxation, method 1 may also require some relaxation.

The vertex-based discretisation procedure allows the liquid fraction f to be estimated for each sub-control volume. This effectively means that the liquid fraction is being calculated over a much finer mesh than in the cell-centred procedure. The solution procedure in each time-step is as follows,

1. Obtain material properties for current temperature.
2. Solve the discretised temperature equation.
3. Calculate the temperature in the sub-control volumes
4. Correct the liquid fraction in each sub-control volume.

4.2.2 Radiation across Mesh Gap

When a gap is formed between two materials and a vacuum is assumed to exist, heat is transferred across the gap by thermal radiation. Stefan-Boltzmann law can be applied to obtain the maximum flux, Q_{max}, at which radiation may be emitted from a surface,

$$Q_{max} = \sigma T_s^4 \tag{4.7}$$

where σ is the Stefan-Boltzmann constant and has the value $\sigma = 5.67 \times 10^{-8}$. To obtain the real flux emitted from the surface, the right hand side is multiplied by emissivity of the surface property,

$$Q_s = \epsilon \sigma T_s^4 \tag{4.8}$$

where the value of emissivity ϵ is in the range $0 \leq \epsilon \leq 1$.

The net range of radiation heat exchange between two surfaces may be written in the form:

$$Q_r = \epsilon \sigma (T_{s_1}^4 - T_{s_2}^4) \tag{4.9}$$

This expression can more conveniently be expressed in the following form,

$$Q_r = k_r (T_{s_1} - T_{s_2}) \tag{4.10}$$

where the radiation heat transfer coefficient k_r is given as:

$$k_r = \epsilon \sigma (T_{s1} + T_{s2})(T_{s1}^2 + T_{s2}^2) \tag{4.11}$$

Since in the vertex-based method, temperature is solved directly on the gap boundary, T_{s1} and T_{s2} are known and the above equations are sufficient to estimate the heat transferred across the gap by radiation. In the cell-centred method the heat transfer from cell-centre to gap boundary needs to be included, as illustrated in Figure 4.5.

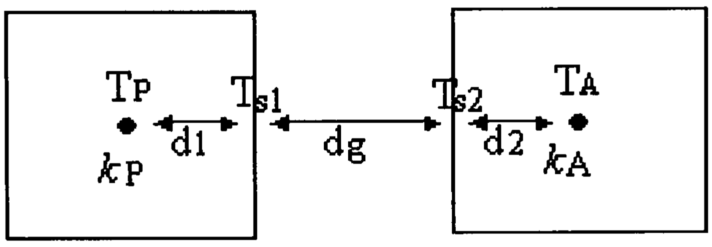


Figure 4.5: Formation of mesh gap

Equations for the heat flux from the cell centre to gap boundary are as follows:

$$Q_{P-S1} = \frac{k_P}{d_1}(T_{s1} - T_P) \tag{4.12}$$

$$Q_{A-S2} = \frac{k_A}{d_2}(T_A - T_{s2}) \tag{4.13}$$

Adding equations (4.12) and (4.13) to equation (4.11) gives, after some rearrangement,

$$Q = h(T_A - T_P) \tag{4.14}$$

where

$$h = \frac{\epsilon\sigma(T_{s1} + T_{s2})(T_{s1}^2 + T_{s2}^2)k_A k_P}{\epsilon\sigma(T_{s1} + T_{s2})(T_{s1}^2 + T_{s2}^2)(d_1 k_A + d_2 k_P) + k_P k_A d_g^2} \tag{4.15}$$

Equation (4.15) requires the values of T_{s1} and T_{s2}. Since values at points P and A are known, the assumption is made that $T_{s1} \simeq T_P$ and $T_{s2} \simeq T_A$.

4.2.3 Solid Mechanics Equations

The solid mechanics PHYSICA module allows for thermo-elastic, thermo-elasto-plastic and thermo-elasto-visco-plastic analysis. In the simulation of the casting of a rotary blade only thermo-elastic analysis was performed. The equations relevant to the thermo-elastic analysis are given in this section. The solid mechanics module employs vertex-based discretisation techniques. PHYSICA employs cell-centred temperature solutions and the vertex-based temperatures required in the thermo-elastic analysis are normally extrapolated from these element based values. Using fully vertex-based solutions the temperatures are located at the required position for the stress analysis. A detailed account of the discretisation and solution procedure is given by Taylor [89].

For infinitesimal strain problems, with strains typically less than 1%, the displacement variation is assumed to be linear. The linear elastic equilibrium equations are given in matrix form as,

$$[L]^T\{\triangle\sigma\} = 0 \tag{4.16}$$

where [L] is the differential operator and $\{\triangle\sigma\}$ is the Cauchy stress incremental yield. For isotropic homogeneous material the stress is related to the elastic incremental strain $\triangle\varepsilon_e$ such that,

$$\{\triangle\sigma\} = [D]\{\triangle\varepsilon_e\} \tag{4.17}$$

where $[D]$ is the elasticity matrix, defined in terms of Young's modulus of elasticity, the proportion of longitudinal strain that is equal to the longitudinal stress, and Poisson's ratio, a measure of the amount the material contracts sideways as it extends. The total incremental strain $\triangle\varepsilon$ is the sum of the elastic incremental strain

$\triangle\varepsilon_e$ and the thermal incremental strain $\triangle\varepsilon_{th}$ giving:

$$\{\triangle\sigma\} = [D]\left(\{\triangle\varepsilon\} - \{\triangle\varepsilon_{th}\}\right) \tag{4.18}$$

The thermal strain rate is given by,

$$\varepsilon_{ij}^{th} = \alpha\dot{T}\delta_{ij} \tag{4.19}$$

where α is the linear coefficient of thermal expansion, $\dot{T}$ is the rate of change of temperature, and δ_{ij} is the Kronecker delta function:

$$\delta_{ij} = \begin{cases} 0 & \text{when} \quad i \neq j \\ 1 & \text{when} \quad i = j \end{cases} \tag{4.20}$$

4.3 Simulation Results

A thermal only analysis was performed on the withdrawal of the assembly from the furnace. The simulation was run using a time-step of 15 seconds. Figures 4.6 and 4.7 give cross-sections showing temperature contours and the liquid fraction for vertex-based and cell-centred results respectively, at 30 minute intervals up to 120 minutes. Their are no significant differences in either the temperature or liquid fraction results obtained cell-centred or vertex-based. The blade solidifies from the base upwards and after about approximately 70 minutes the whole blade has solidified The geometry also cools as it is withdrawn from the furnace and continues to cool from the base upwards after withdrawal is complete. The withdrawal from the furnace takes approximately 75 minutes to complete and a further 75 minutes until the whole geometry is completely cool. After 120 minutes the temperature of the geometry is between $293K$ and $442K$.

A thermal-elastic analysis was performed on the same mesh with a time-step of 15 seconds. The elastic analysis was performed every 24 time-steps with the thermal analysis taking place every time-step. Figure 4.8 shows the temperature contours at 30 minute intervals up to 120 minutes. There are no significant differences in the vertex-based and cell-centred solutions for temperature and liquid fraction. Figures 4.9 and 4.10 show the cross-sectional effective stress and displacement contours for the vertex-based and cell-centred methods respectively. The resultant displacement of the alloy within the shell is similar for both the cell-centred and vertex-based results. The maximum displacement is 8.09×10^{-3} mm (cell-centred) and 9.67×10^{-3} mm (vertex-based). The results show the stress in the alloy to be more concentrated around the base of the alloy, at the alloy/chill interface. This is probably due to the assumption that the alloy and chill remain attached together with x, y and z being fixed at the interface. The maximum effective stress value is located at this interface, being 3.57×10^{10} Pa (vertex-based) and 3.26×10^{10} Pa (cell-centred).

4.3.1 Run-time - Memory Requirements

The analysis was carried out on a DEC alpha 466 MHz processor. The thermal analysis required a computational time of approximately 22 hours (vertex-based) and 14 hours (cell-centred). The vertex-based method required less iterations per time-step to achieve the same level of convergence as the cell-centred method. This gives a computational time per solution point of approximately 1.46 seconds (vertex-based) and 1.11 seconds (cell-centred). The vertex-based method required 1.6 times the computational time of the cell-centred method for the thermal simulation, but this reduces to 1.3 per solution point. The additional computational time required for the mechanical analysis was approximately 24 hours in both cases. The memory requirements for the thermal analysis was 45.78 megabytes and 8.18 megabytes for vertex-based and cell-centred solutions respectively. The stress analysis required an extra 38.25 megabytes.

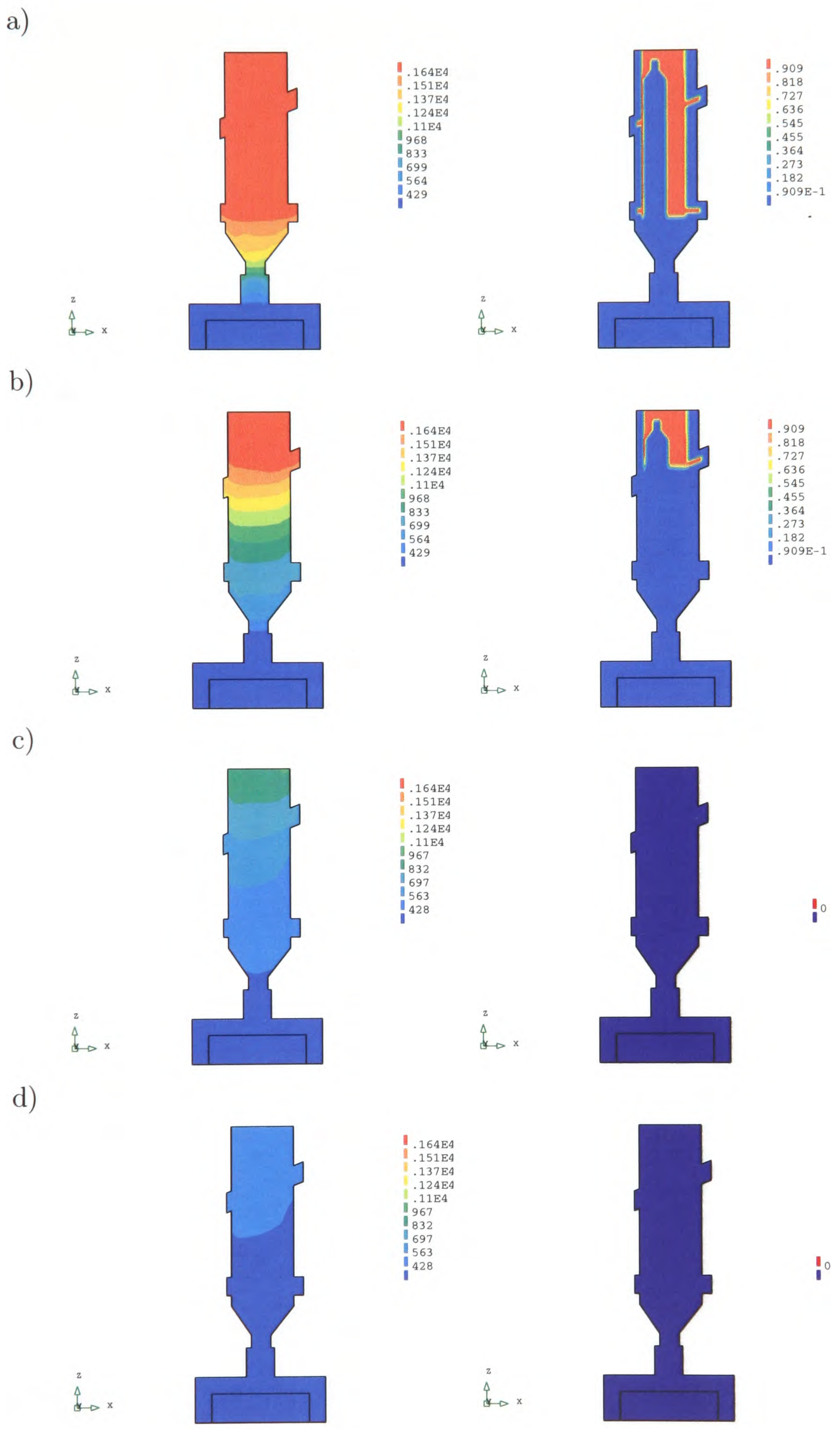


Figure 4.6: Vertex-Based results, Cross-sections showing the temperature contours (left) and liquid fraction (right), after a)30 mins, b)60 mins, c)90 mins and d)120 mins

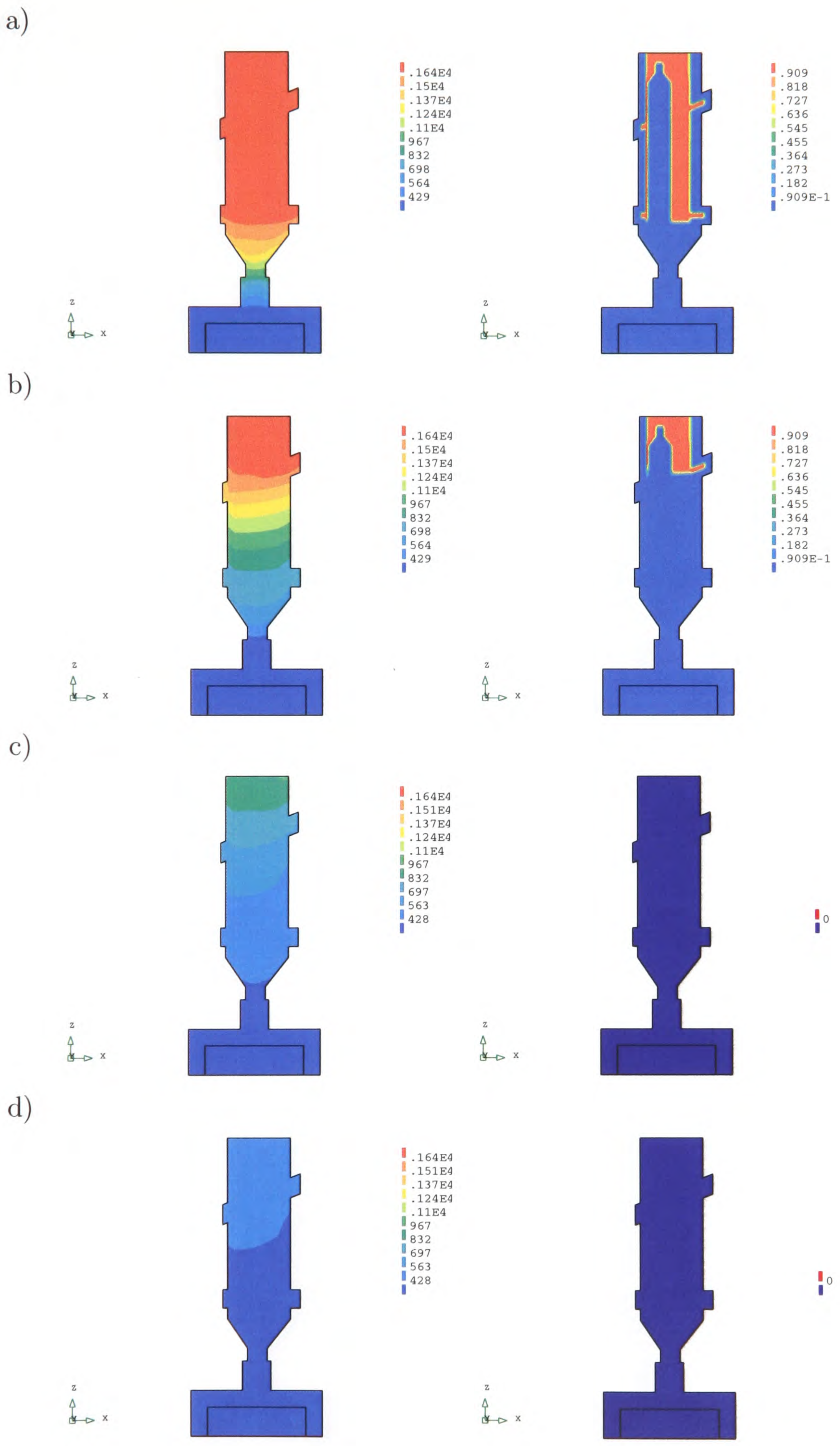


Figure 4.7: Cell-Centred results, Cross-sections showing the temperature contours (left) and liquid fraction (right), after a)30 mins, b)60 mins, c)90 mins and d)120 mins

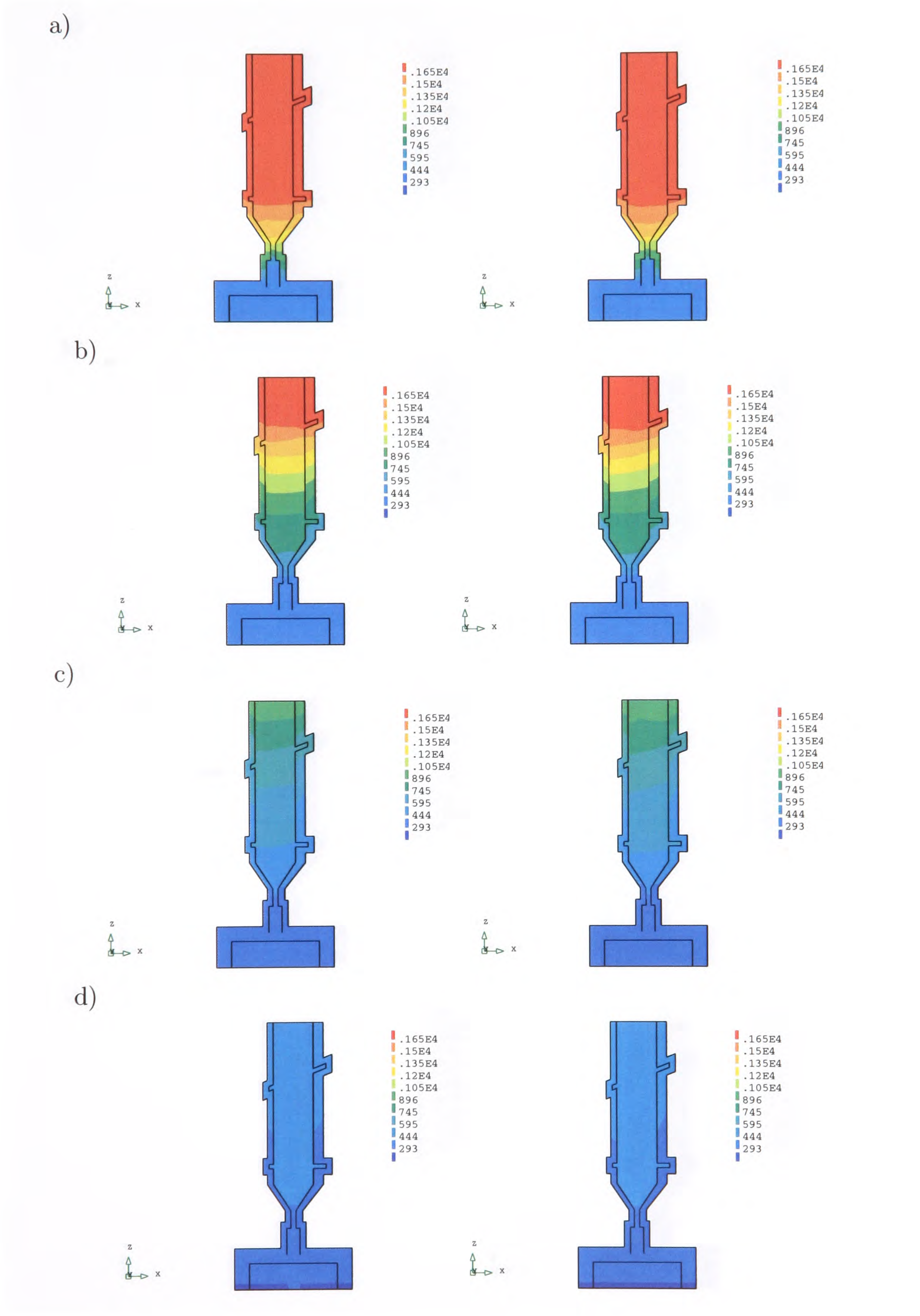


Figure 4.8: Thermal-elastic analysis, Cross-sections showing the temperature contours, vertex-based (left) and cell-centred (right), after a)30 mins, b)60 mins, c)90 mins and d)120 mins

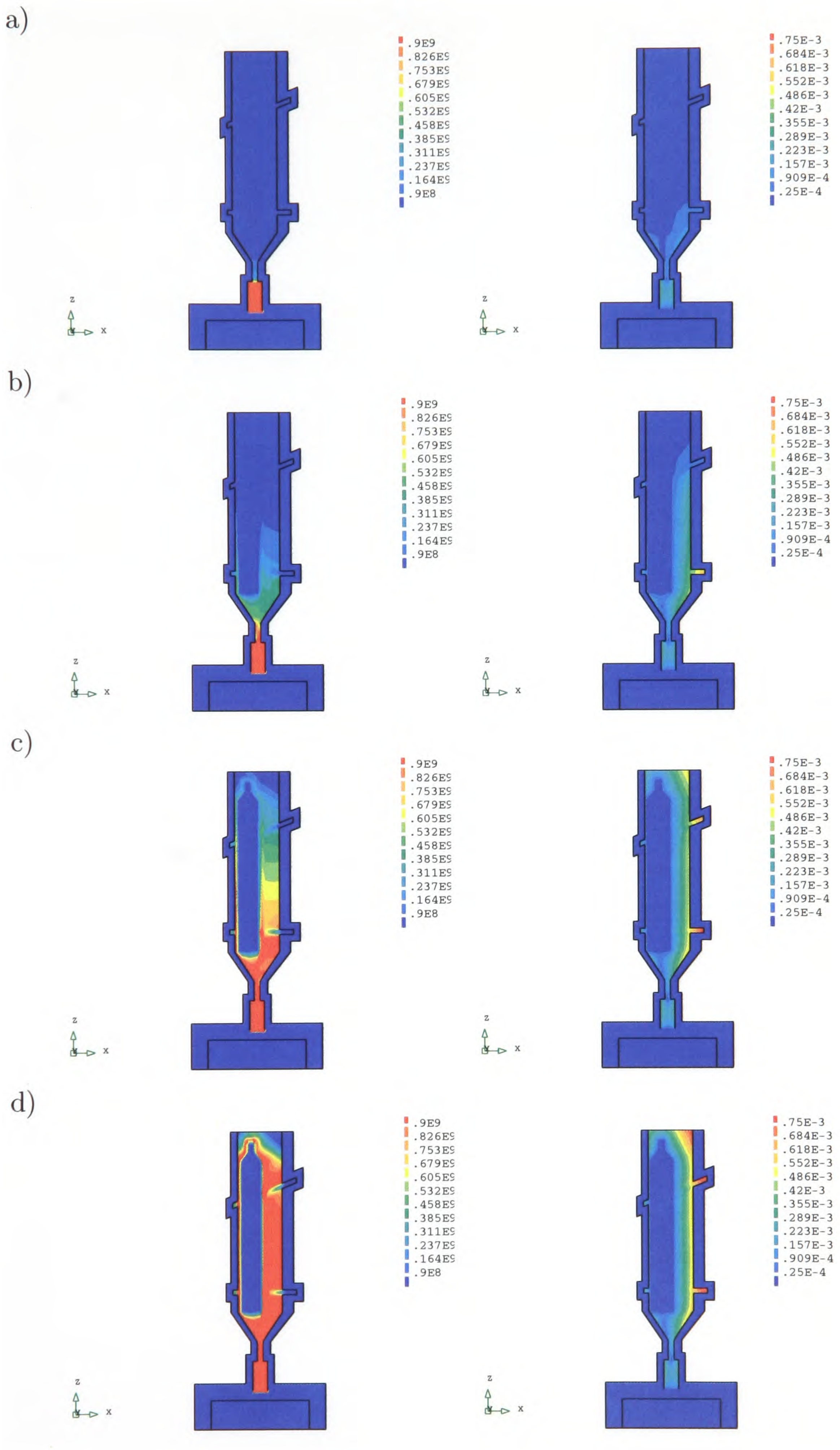


Figure 4.9: Vertex-Based results, Cross-sections showing the effective stress contours (left) and displacement (right), after a)30 mins, b)60 mins, c)90 mins and d)120 mins

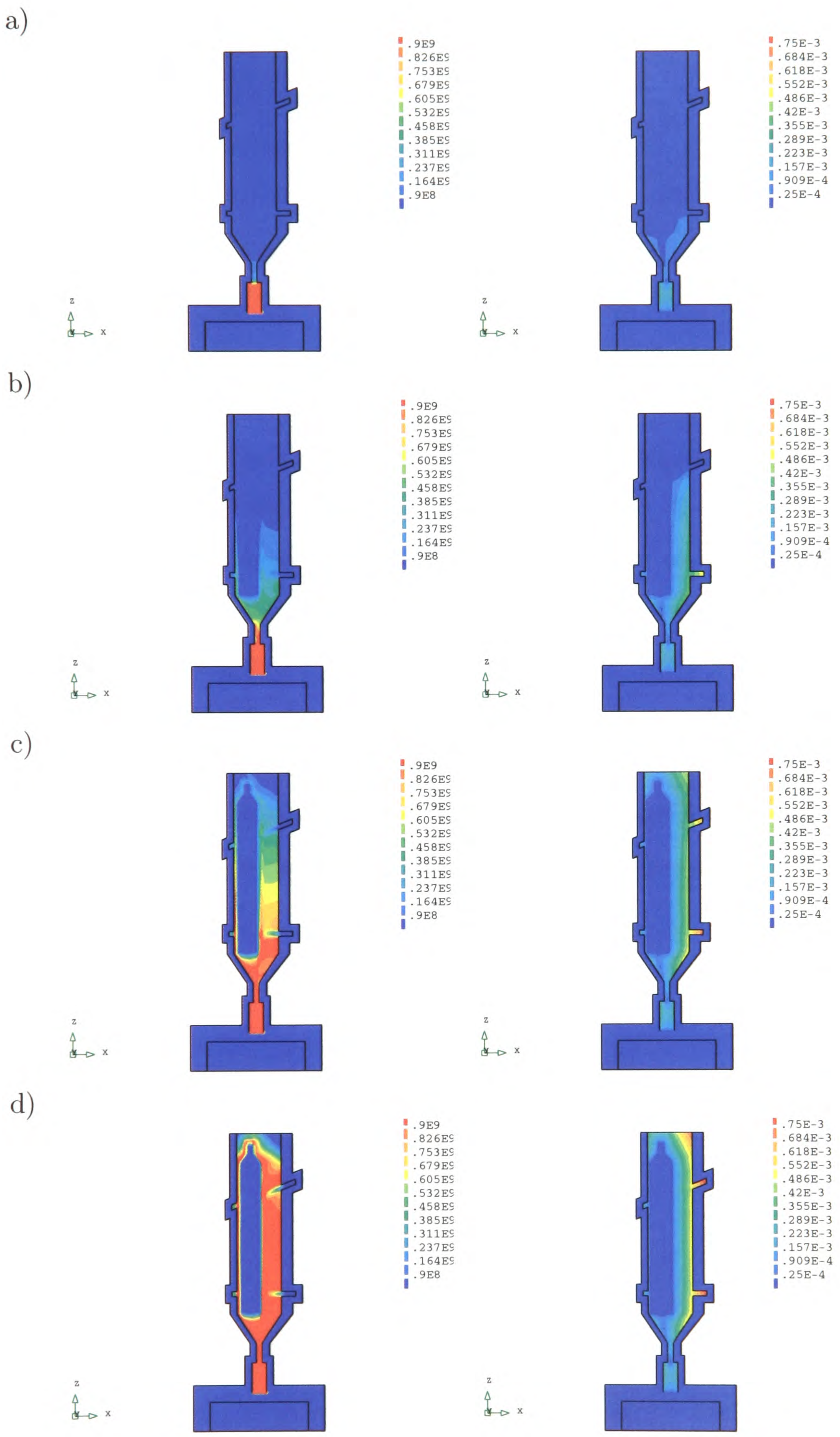


Figure 4.10: Cell-Centred results, Cross-sections showing the effective stress contours (left) and displacement (right), after a)30 mins, b)60 mins, c)90 mins and d)120 mins

4.4 Closure

The vertex-based discretisation method has been successfully embedded with in the multi-physics framework of PHYSICA. The casting of the rotary blade involved heat transfer, solidification, radiation and stress analysis. Vertex-based solutions were in good agreement with the cell-centred solutions. The results indicate that the vertex-based method can be successfully coupled with other physical phenomena utilising existing PHYSICA modules. Computational requirements significantly increase when using vertex-based discretisation. On an nearly orthogonal mesh, as employed in the simulation of the casting a rotary blade, the cell-centred method is computationally less expensive and there would appear to be no benefit in using the vertex-based method. However, if during the simulation the mesh deformed, introducing non-orthogonality into the mesh, problems could be encountered with cell-centred discretisation. As discussed in section 2.1.1 non-orthogonal corrections would need to be included which could lead to difficulties in convergence and longer simulation times. It is in this scenario that it could be beneficial to employ vertex-based discretisation as it does not suffer from non-orthogonal issues.

Chapter 5

Solution of Fluid Flow

In the work presented earlier in chapter 2, the solution of a conserved variable ϕ assumed that the velocity field was somehow known. In general the flow field is not known, but emerges as part of the solution procedure. The velocity field must be obtained by solving a number of conservation equations. The momentum transport equations can be written in the same form as the general transport equation (2.1), with $\phi = u, v$ or w and $\Gamma = \mu$. The pressure gradient term that forms the main momentum source term is written separately:

$$\frac{\partial(\rho u_i)}{\partial t} + div(\rho \underline{u} u_i) - div(\mu \nabla u_i) = S_{u_i} - \nabla_{x_i} p \tag{5.1}$$

The velocity field must also satisfy mass conservation:

$$\frac{\partial \rho}{\partial t} + div(\rho \underline{u}) = S_m \tag{5.2}$$

The non-linearity of the convection term can be handled by an iterative solution method. If the pressure field is known there is little difficulty in obtaining a solution of the velocity components. In general the pressure field in flow computations is not known and is calculated as part of the solution.

If flow is compressible the continuity equation may be used as a transport equation for density and the energy equation as a transport equation for temperature. The pressure may then be obtained from the equation of state $p = p(\rho, T)$. If flow is incompressible the difficulty arises on how to resolve the role played by pressure. A pressure gradient source term appears in all three momentum equations but there is no other equation linked to pressure. Pressure and velocity are intrinsically coupled, if the correct pressure field is applied in the momentum equations the resulting velocity field will satisfy the continuity equation. The problem of pressure-velocity coupling can be overcome by means of an iterative guess and correct solution procedure, such as the SIMPLE or SIMPLER algorithm outlined in sections 5.1.1 and 5.1.2. Decoupling of pressure and velocity components can occur if the face velocity component is not dependant on adjacent pressure values. This problem is addressed in section 5.2 which deals with pressure and velocity checkerboarding. Initially a general outline of the solution algorithms are given in the following sections. A more detailed account of the procedures used for the cell-centred and vertex-based discretisation methods is given in sections 5.3 and 5.4 respectively.

5.1 Pressure - Velocity Coupling

As a starting point for the solution procedure, the system of algebraic equations obtained from the discretised momentum equations can be expressed in the following form:

$$a_{u_i} u_i = \sum a_{nb} u_{nb} + b_{u_i} - \nabla_{x_i} p \tag{5.3}$$

The velocity field is also subject to the constraint that it must satisfy the continuity equation (5.2). The discretised form of the continuity equation for a control volume (cv) is:

$$\frac{\rho_{cv} V_{cv} - \rho^0_{cv} V^0_{cv}}{\triangle t} + \sum_f \rho_f A_f (\underline{u}.\underline{n})_f = V_{cv} S^m_{cv} \tag{5.4}$$

5.1.1 SIMPLE

The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE), was originally developed by Patankar and Spalding [?] for staggered grid arrangements. It is a guess and correct solution procedure, initially the discretised momentum equations are solved using a guessed pressure field (p^*) to yield velocity components u^*, v^* and w^*:

$$a_{u_i} u^*_i = \sum a_{nb} u^*_{nb} + b_{u_i} - \nabla_{x_i} p^* \tag{5.5}$$

The aim is to improve the guessed pressure p^* successively, thus improving the solution of the velocity components u^*_i. Substitution of the correct pressure field p into the momentum equations will yield the correct velocity field. A correction term $p^{'}$ is defined as the difference between the correct pressure field p and the guessed pressure field p^*:

$$p = p^* + p^{'} \tag{5.6}$$

Similarly velocity correction terms $u^{'}_i$ are defined to relate u^*_i to the correct velocities u_i:

$$u_i = u_i^* + u_i^{'} \tag{5.7}$$

Subtraction of (5.5) from (5.3) yields an expression linking the correction formulae:

$$a_{u_i} u_i^{'} = \sum a_{nb} u_{nb}^{'} - \nabla_{x_i} p^{'} \tag{5.8}$$

It is at this point that the main approximation of the SIMPLE algorithm is made. The first term on the right hand side of (5.8) is omitted, giving:

$$u_i^{'} = \frac{-\nabla_{x_i} p^{'}}{a_{u_i}} \tag{5.9}$$

Inclusion of the omitted term would lead to a system of equations where pressure corrections were dependant on neighbouring corrections. Omission of this term simplifies the resulting pressure correction equation. This approximation does not affect the final solution since at convergence this term would be zero.

The simplified equation (5.9) for the velocity and pressure correction can be substituted into (5.7) to give an expression linking the correct velocity u_i to the guessed velocity u_i^* and the pressure correction $p^{'}$:

$$u_i = u_i^* - \frac{-\nabla_{x_i} p^{'}}{a_{u_i}} \tag{5.10}$$

Equation (5.10) gives an expression for updating the velocity field when a pressure correction value has been obtained.

From the conservation principle of the control volume formulation, the velocity field at a point on the face of a control volume, must also have a discretised momentum equation of the same form as (5.3). Using the same principles as applied to obtain (5.10), a similar equation can be written for the face velocity component:

$$(u_i)_f = (u_i^*)_f - \frac{-(\nabla_{x_i} p')_f}{(a_{u_i})_f} \tag{5.11}$$

Equation (5.11) can now be substituted into the discretised continuity equation (5.4) to obtain an equation for the pressure correction p'. The face velocity components and their related coefficients are interpolated from control volume nodal values.

5.1.1.1 SIMPLEC

A variant of SIMPLE is the SIMPLEC algorithm of Van Doormal and Raithby, [138]. The SIMPLEC procedure proceeds along the same lines as the SIMPLE algorithm, section 5.1.1, with the difference that the velocity correction equations omit terms that are less significant than those omitted in SIMPLE. This method assumes that $u'_{nb} \simeq u'_i$. Enhancing the correction procedure improves the convergence behaviour, producing savings in computational time.

The velocity correction equation is,

$$u'_i = \frac{-\nabla_{x_i} p'}{a_{u_i} - \sum a_{u_{nb}}} \tag{5.12}$$

giving an expression for updating the velocity field as,

$$u_i = u_i^* - \frac{-\nabla_{x_i} p'}{a_{u_i} - \sum a_{u_{nb}}} \tag{5.13}$$

and a similar equation for the face velocity component:

$$(u_i)_f = (u_i^*)_f - \frac{-(\nabla_{x_i} p')_f}{(a_{u_i} - \sum a_{u_{nb}})_f} \tag{5.14}$$

The sequence of operations is identical to that of SIMPLE, with Equation (5.14) being substituted into the discretised pressure correction equation.

5.1.2 SIMPLER

In the SIMPLER (SIMPLE Revised) algorithm of Patankar [6] the discretised continuity equation (5.4) is used to derive a discretised equation for pressure, instead of a pressure correction equation as in SIMPLE. The pressure field is obtained directly, without the use of corrections. The velocity components are still obtained through the velocity corrections (5.10), as derived in the SIMPLE algorithm.

The momentum equations (5.3) are rearranged as follows:

$$u_i = \frac{\sum a_{nb} u_{nb} + b_{u_i}}{a_{u_i}} - \frac{\nabla_{x_i} p}{a_{u_i}} \tag{5.15}$$

Pseudo-velocities $\hat{u}, \hat{v}, \hat{w}$ and a pressure gradient coefficient d^{u_i} for each control volume can be defined as,

$$\hat{u}_i = \frac{\sum a_{nb} u_{nb} + b_{u_i}}{a_{u_i}} \tag{5.16}$$

$$d^{u_i} = \frac{V}{a_{u_i}} \tag{5.17}$$

Substituting (5.16) and (5.17) in to equation (5.15) gives:

$$u_i = \hat{u}_i - d^{u_i}\frac{\partial p}{\partial x_i} \tag{5.18}$$

As in the SIMPLE algorithm, applying conservation principles, a similar expression can be derived for the face velocity component.

$$(u_i)_f = (\hat{u}_i)_f - (d^{u_i})_f\left(\frac{\partial p}{\partial x_i}\right)_f \tag{5.19}$$

An equation for pressure is obtained by substituting (5.19) into the discretised continuity equation (5.4).

If consistent discretisation and interpolation techniques are applied, the coefficients of the derived pressure equation (5.20) will be the same as the coefficients of the pressure correction equation (5.21) of SIMPLE.

$$[A][p] = [\hat{b}] \tag{5.20}$$

$$[A][p'] = [b^*] \tag{5.21}$$

The difference in the two methods is the evaluation of the source term $[b]$. The pressure equation source term is evaluated using pseudo-velocities ($\hat{u}, \hat{v}$ and $\hat{w}$), whereas the pressure-correction source term is evaluated using the previously obtained velocities (u^*, v^* and w^*).

The velocities are updated using the velocity corrections (5.10), as in the SIMPLE algorithm. This procedure requires that a pressure-correction equation must also be solved to obtain the pressure corrections used in (5.10). The SIMPLER procedure solves the pressure equation first from guessed initial values of pressure and velocity.

Once a pressure field has been obtained the momentum equations are solved and the SIMPLE algorithm applied to obtain the pressure corrections needed for the velocity corrections.

5.1.2.1 Revised SIMPLER

In this revised version of the SIMPLER algorithm, a pressure equation is obtained in the same way as detailed in the preceding section. The pressure equation (5.20) can be transformed into a pressure correction equation as follows,

$$[A][p'] = [b'], \text{ where } [b'] = [\hat{b}] - [A][p^*] \tag{5.22}$$

where p^* is the preceding pressure solution. The pressure field is updated using (5.6) and instead of using pressure corrections to update the velocity components, the velocities are updated directly by the corrected pressure values. The control volume face velocity components are updated using equation (5.19). It is this mass conserving face velocity field that is fed directly into the discretised momentum equations. In order to advance the solution process, the nodal velocity components are also updated using equation (5.18).

5.2 Velocity and Pressure Checkerboarding

Co-located discretisation methods that solve and store pressure and velocity components at the same locations can suffer from checkerboard pressure predictions. These oscillating pressure fields are a result of linear interpolation techniques that express control volume pressure gradients in terms of alternate pressure values. This means that if pressure, located at consecutive grid points, oscillate as follows 1, 100, 1, 100 etc, this would be seen as a uniform pressure field, as alternate pressure values are the same. If a checkerboard pressure field can be supported by both the discretised form of the momentum and continuity equations spurious oscillating pressure predictions can emerge. This is illustrated for a one-dimensional co-located formulation with control volume face cross-sectional areas equal to one. Figure 5.1 shows a control volume P with west and east neighbours.

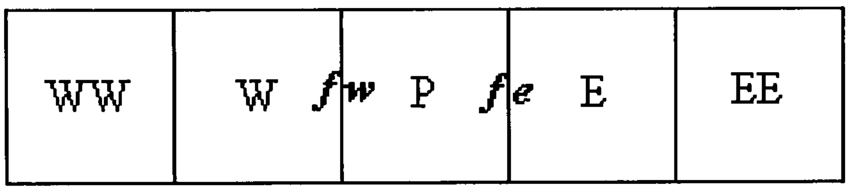


Figure 5.1: Grid used to illustrate velocity and pressure interpolation

The pressure gradient enters the momentum equation as a source term (5.1). Calculating the pressure gradient over control volume (P) using linear interpolation to obtain pressures at faces (fe) and (fw) gives:

$$\begin{aligned}\frac{\partial p}{\partial x} &= \frac{p_{fe} - p_{fw}}{\delta x} \\ &= \frac{\left(\frac{p_P + p_E}{2}\right) - \left(\frac{p_P + p_W}{2}\right)}{\delta x} \\ &= \frac{p_E - p_W}{2\delta x} \qquad (5.23)\end{aligned}$$

The discretisation does not involve the pressure values at point P. Only alternate control volume pressure values enter the equation. Thus, the u-momentum equation does not recognise a checkerboard pressure field and oscillating pressure predictions would be seen as a uniform pressure field. If this type of discretisation is maintained for the pressure term in the momentum equations the discretisation of the continuity equation must ensure that checkerboard pressure fields can not emerge. Discretising the continuity equation using linear interpolation to obtain the face u-velocity values from cell velocity values, as follows,

$$u_{fe} - u_{fw} = \frac{u_P + u_E}{2} - \frac{u_P + u_W}{2} \tag{5.24}$$

$$\begin{aligned} \text{where:} \quad u_P = & \quad \hat{u}_P + \frac{1}{a_P}(\frac{p_W - p_E}{2}) \\ u_E = & \quad \hat{u}_E + \frac{1}{a_E}(\frac{p_P - p_{EE}}{2}) \\ u_W = & \quad \hat{u}_W + \frac{1}{a_W}(\frac{p_{WW} - p_P}{2}) \end{aligned}$$

gives:

$$u_{fe} - u_{fw} = \frac{\hat{u}_E - \hat{u}_W}{2} + \frac{1}{a_E}(\frac{p_P - p_{EE}}{4}) - \frac{1}{a_W}(\frac{p_{WW} - p_P}{4}) \tag{5.25}$$

It can be seen that the same type of checkerboard pressure field supported by the discretised momentum equations is also supported by the discretised continuity equation, because $p_p = p_{EE} = p_{WW}$. In addition to the pressure checkerboarding, the linear interpolation of cell velocities results in only adjacent velocities appearing in the discretised continuity equation. Hence, a checkerboard velocity field would also satisfy the continuity equation. As a means of overcoming this problem the use of staggered grid arrangements was introduced by Harlow and Welsh [12] and was used in the development of the SIMPLE procedure. In the staggered grid arrangement the momentum equations are discretised over a different set of control volumes. The velocity components are solved directly on the pressure cell faces. No interpolation is required and the values of u_{fe} and u_{fw} are obtained directly from the solution of the u-momentum equation. Similarly, the pressure values are now located on the face of the momentum control volume and the pressure gradient term can be obtained from adjacent values. An oscillating pressure field would be recognised by the momentum equations.

In co-located solution techniques, to prevent checkerboarding in the final solution, the discretisation technique used in the momentum or continuity equation must provide a filter to remove oscillating results. Checkerboarding is overcome by devising interpolation procedures which express face velocities in terms of adjacent pressure values rather than alternate pressure values. The procedure, known as momentum interpolation, does not use pure linear interpolation to define the face velocity. An additional term is included which is dependent on adjacent pressure values. Several variations of this scheme have been proposed [7, 8, 27, 28], but the basic idea remains the same. It is these newly defined face velocity values that are used in the discretision of the continuity equation. Thus, at convergence, it is the face velocities that directly satisfy continuity. The cell-centred velocities only satisfy conservation indirectly in that they satisfy the momentum interpolation formula. Another method, which essentially uses the same key idea, is to define a new face velocity field which is driven by the pressure difference between adjacent cell values [93]. This new face velocity field can be considered as a type of staggered velocity field. Instead of deriving this velocity field directly, a pseudo velocity field (5.16), which contains the convection, diffusion, and source term, is obtained from the momentum equations, and a pressure gradient term added.

The basic idea behind these interpolation techniques is the replacement of the pressure gradient term, which is obtained using linear interpolation of cell velocities, with a new pressure gradient term which is calculated from adjacent pressure values. It is these face velocities, written in terms of adjacent pressure values, that are used to satisfy continuity. Thus, a checkerboard pressure field would be recognised by these face velocities. Although the momentum equations still contain a pressure gradient term that does not recognise a checkerboard pressure field, such a field would not satisfy the continuity equation.

5.3 Cell-Centred Procedure

The cell-centred procedure employed in this investigation uses the Rhie-Chow [8] interpolation scheme as a means of avoiding spurious pressure predictions and the SIMPLE solution algorithm. The discretisation and solution procedure is outlined as follows,

5.3.1 Rhie-Chow Interpolation Method

Equation (5.3) gives the discretised momentum equation for a control volume around node i. From the conservation principle of the control volume formulation, the velocity on a control volume face must also have an equation of the form:

$$a_f^{u_i} u_f^i + (\nabla_{x_i} p)_f = (\sum a_{nb} u_{nb})_f + b_f^{u_i} \tag{5.26}$$

The Rhie-Chow interpolation method approximates a solution for the right hand side of equation (5.26) using weighted linear interpolation of the corresponding terms in the nodal equations of the same form,

$$a_f^{u_i} u_f^i + (\nabla_{x_i} p)_f = \overline{(\sum a_{nb} u_{nb})_f + b_f^{u_i}} = \overline{a_f^{u_i} u_f^i + (\nabla_{x_i} p)_f} \tag{5.27}$$

where the overline indicates a weighted linear interpolation of the nodal values. The assumption that $a_f^{u_i} \approx \overline{a_f^{u_i}}$ is made, giving:

$$u_f^i = \overline{u_f^i} + \overline{d_f^i}(\overline{(\nabla_{x_i} p)_f} - (\nabla_{x_i} p)_f) \tag{5.28}$$

Taking α as the weighting factor, P the control volume node and A the adjacent control volume node.

$$\begin{aligned}
\overline{u^i_f} &= \alpha u^i_P + (1-\alpha)u^i_A \\
\overline{(\nabla_{x_i}p)_f} &= \alpha(\nabla_{x_i}p)_P + (1-\alpha)(\nabla_{x_i}p)_A \\
(\nabla_{x_i}p)_f &= A_f n^{x_i}_f(p_A - p_P) \\
a^{u_i}_f &= \alpha a^{u_i}_P + (1-\alpha)a^{u_i}_A \\
\overline{d^{u_i}_f} &= (a^{ui}_f)^{-1}
\end{aligned} \tag{5.29}$$

5.3.2 The Solution Procedure

The momentum equations are discretised as outlined in chapter 2, using Rhie-Chow interpolation to obtain the face velocity components. The control volume pressure gradient, which enters the momentum equations as a source term, is evaluated as a sum of surface integrals over each face bounding the control volume,

$$\int_{cv} \nabla_{x_i} p dV \simeq \sum_f A_f n^{x_i}_f p_f \tag{5.30}$$

where p_f is interpolated from adjacent values.

The continuity equation is discretised to form a pressure-correction equation as follows,

Using the SIMPLE approximation procedure equations (5.5) to (5.9), an approximation for the face velocity correction is obtained:

$$\begin{aligned}(u_i')_f &= -\frac{1}{(a_{u_i})_f}(\nabla_{x_i} p')_f \\ &= \frac{1}{(a_{u_i})_f} A_f n_f^{x_i}(p_P' - p_A') \end{aligned} \tag{5.31}$$

$(u_i^*)_f$ is obtained using Rhie-Chow interpolation. (5.7) and (5.31) are substituted into the discretised continuity equation (5.4) to give,

$$\sum_f \rho_f \frac{A_f^2 n_i^2}{a_i}(p_P' - p'_A)_f = \frac{\rho_P^0 V_P^0 - \rho_P V_P}{\Delta t} - \sum_f A_f \rho_f (\underline{u}^*.\underline{n})_f \tag{5.32}$$

where the subscript i in the first term indicates a summation over the three coordinate directions.

The above equation leads to a set of linear equations with weak diagonal dominance and can be written in the form:

$$a_P p_P' + \sum_{nb} a_{nb} p_{nb}' = b_P \tag{5.33}$$

Once the above equation has been solved and a pressure correction field obtained, the correct pressure field may be obtained using formula (5.7). The SIMPLE approximation tends to lead to an over-estimation of the pressure correction values, making the pressure-correction equation susceptible to divergence unless some under-relaxation is used during the iterative procedure. In the procedure used here a relaxation value of 0.6 is set as standard.

It is the element cell values that are updated using formula (5.7) rather than face values. The velocity correction values are obtained from pressure correction values using (5.9). Expanding this equation gives the velocity corrections in terms of pressure corrections in the element and all its neighbours:

$$u'_P = -\frac{1}{a_P}\sum_{nbrs} n_{x_i} A_f(\alpha_f p'_P + (1-\alpha_f)p'_{nb}) \tag{5.34}$$

5.4 Vertex-Based Procedure

The following section discusses the possible solution methods that could be employed in the vertex-based flow procedure. A detailed account of the discretisation and solution method adopted is given in section 5.4.2.

5.4.1 Discussion of Solution Method

The PHYSICA cell-centred technique can employ either the SIMPLEC or SIMPLE solution algorithm, using the Rhie-Chow interpolation scheme to avoid checkerboard pressure predictions. The vertex-based procedure could also proceed along these lines. However, there have been a number of reported problems associated with the Rhie-Chow scheme.

Croft [2] showed that difficulties can be encountered when there is mesh skewness. The Rhie-Chow interpolation method guarantees that the convective face fluxes satisfy continuity, but can produce non-physical cell centred values of velocity. This was illustrated in a simple test case of flow along a pipe, modelled with a single row of elements with a bend part way along the row. The analytical solution gives a velocity vector in the direction of the pipe both before and after the bend. PHYSICA fails to predict the change in flow direction after the pipe. The face convective fluxes satisfy continuity throughout the domain but at a cost to the element-based velocities. As more elements are placed across the flow the error reduces but is never completely eliminated. This problem was also reproduced in other CFD codes that employ Rhie-Chow interpolation. Majumdar [34] found that the Rhie-Chow scheme did not take into account the presence of under-relaxation factors in the discretised momentum equations and that the converged solution can be dependent on the under-relaxation factor employed. Choi [51] reported that in unsteady flow solutions the time step size also affected the converged velocity solution. Recently, Kawaguchi *et al* [52] found that the use of Rhie-Chow interpolation not only affects the accuracy of the converged solution but in certain circumstances may also cause checkerboard pressure predictions. Majumbar, Choi, and Kawaguchi have all proposed amendments to the original Rhie-Chow scheme, which take into account the effect of under-relaxation and time-step size. The problems reported appear to result from the Rhie-Chow approximation shown in equation (5.27). The proposed corrections involve expressing the terms in the discretised momentum equations explicitly, including the under-relaxation and unsteady term, for the cell face and cell-centred velocity. Equations are derived for the cell face velocity components with fewer terms omitted.

The method of Prakash and Baliga [93], which defines a new face velocity field which is dependant on adjacent pressure values, does not appear to suffer from the problems reported above. The momentum equations are discretised and solved to obtain a pseudo velocity field, as defined in (5.16). This pseudo velocity field comprises all terms included in the discretised momentum equations with the exception of the pressure gradient term. A new face velocity field is obtained from the pseudo velocities and face pressure gradient.

Employing the SIMPLE solution method requires some form of momentum interpolation to evaluate the pressure correction source term. Approximations are made during the solution process in the evaluation of the error in the velocity field (5.9). These approximations do not effect the final solution as the terms omitted would be zero at convergence. However, these approximations give an inconsistency between pressure and momentum corrections that can lead to oscillations in the continuity and velocity error. These oscillating errors can lead to longer simulation times. In the revised SIMPLER algorithm the source term does not contain any contribution from the pressure term. The source is evaluated from pseudo velocities using simple linear interpolation to obtain pseudo face values. The pressure is solved directly with no relaxation and the face velocity components are updated with no approximations being made.

The previous paragraphs have highlighted the problems associated with the Rhie-Chow interpolation scheme and the possible errors that can be introduced into the solution procedure when approximations are made. For this reason it was decided to employ the revised SIMPLER solution algorithm in the vertex-based flow procedure, defining a new face velocity field from pseudo velocities and face pressure gradients.

5.4.2 The Solution Procedure

To initiate the solution procedure the momentum equations are discretised using the vertex-based technique, given in chapter 2. Initially the face velocity is obtained employing the shape function interpolation functions,

$$u_f^i = \sum_{j=1}^{n} N_j u_j^i \tag{5.35}$$

where n is the number of nodes of the element that bounds the face, j is the element node, and N_j are the interpolation functions.

The control volume pressure gradient can be evaluated as (5.30) or as a sum of sub-volume integrals:

$$\int_{cv} \nabla_{x_i} p = \sum_{scv} V_{scv} \sum_{j=1}^{n} \frac{\partial N_j}{\partial x_i} p_j \tag{5.36}$$

Once an initial velocity field has been obtained, a pseudo velocity field ($\underline{\hat{u}}, \underline{\hat{v}}$ and $\underline{\hat{w}}$) is defined that does not depend directly on the pressure distribution, using (5.3), (5.16) and (5.17):

$$(\hat{u}^i)_{cv} = (u^i)_{cv} + (d^{u_i})_{cv}\left(\frac{\partial p}{\partial x_i}\right)_{cv} \tag{5.37}$$

Using conservation principles for the control volume face, a new velocity field ($\underline{\tilde{u}}, \underline{\tilde{v}}$ and $\underline{\tilde{w}}$) can now be defined that is dependent upon adjacent pressure values,

$$(\tilde{u}^i)_f = (\hat{u}^i)_f - (d^{u_i})_f\left(\frac{\partial p}{\partial x_i}\right)_f \tag{5.38}$$

were $(\hat{u}^i)_f$ and $(d^{u_i})_f$ are calculated from nodal values assuming linear variation within an element. The pressure gradient across a face is obtained using local element shape function derivatives,

$$\frac{\partial p}{\partial x_i} = \sum_{j=1}^{n} \frac{\partial N_j}{\partial x_i} p_j \tag{5.39}$$

where n is the number of nodes belonging to an element and N_j is the shape function associated with node j of the element (which bounds the face). The face pseudo velocity $(\hat{u}^i)_f$ and pressure gradient coefficient $(d^{u_i})_f$ are interpolated linearly from nodal values.

The continuity equation (5.2) is discretised using the $\underline{\tilde{u}}$ field as follows,

$$\frac{\rho^0_{cv}V^0_{cv} - \rho_{cv}V_{cv}}{\nabla t} - \sum_f \rho_f A_f(\underline{\tilde{u}}.\underline{n})_f = 0 \tag{5.40}$$

Substituting (5.38) in to (5.40) and rearranging gives:

$$\sum_{f}\sum_{j=1}^{n}\sum_{i=x,y,z}\rho_f A_f d_f^{u_i}\left(\frac{\partial N_j}{\partial x_i}.n_i\right)p_j=\sum_{scv}\frac{\rho_{scv}^0 V_{scv}^0-\rho_{scv}V_{scv}}{\Delta t}-\sum_{f}A_f\rho_f(\underline{\hat{u}}.\underline{n})_f \tag{5.41}$$

For Dirichlet boundary conditions, where velocity is specified, the known mass flux is transferred to the right-hand side of (5.41) and the corresponding pressure coefficient d^{u_i} is set to zero. If pressure is known at the boundary node, the control volume coefficient is set equal to one, all off-diagonal terms are set to zero and the right-hand side set equal to the known pressure. At outflows where pressure and velocity are not known, (5.41) is completed by estimating the mass flux out of the domain based on the most recently calculated values of the velocity field at the boundary.

This leads to a positive definite system of linear equations with a larger bandwidth than the cell-centred approach. For a hexahedral mesh there are a possible twenty-seven non-zero coefficients, compared to seven in the cell-centred case. The pressure equations can be transformed into a system of pressure-correction equations as in (5.22). No relaxation is applied to pressure, otherwise the resulting corrected $\underline{\tilde{u}}$ field will not satisfy mass conservation. The $\tilde{u}_i$ velocities are updated using (5.38) with no relaxation. It is this mass conserving face velocity field that is used in the discretised momentum equations. In order to advance the solution process the nodal velocities are also updated using (5.18), and a relaxation factor of 0.7 is applied as standard.

5.5 Hybrid Discretisation of Hydrodynamic Variables

This section investigates hybrid discretisation solutions for the hydrodynamic variables, pressure and velocity. It will be demonstrated later that the cell-centred discretisation method is fast and efficient, but does not handle skewed meshes with the ease of the vertex-based technique. The question is, is there a method that can combine the efficiency of the cell-centred technique whilst retaining the geometric flexibility of the vertex-based technique?

The pressure field could be obtained using the vertex-based discretisation procedure whilst using cell-centred discretisation for the momentum equations. Obtaining better resolution of the pressure field on distorted meshes should give improved resolution of the velocity field since pressure is the driving force for the components of velocity. The solution of the momentum equations is used as a vehicle for advancing the solution process. The velocity components are updated and defined in terms of local pressure gradients. Using the momentum equations to obtain a pseudo velocity field but defining a face velocity field using vertex-based techniques allows for better definition of the local pressure gradient (from 8 points on a hexahedral element) on a distorted mesh.

Conversely, the vertex-based technique, which has the ability to handle distorted meshes, could be used to discretise the momentum equations. Would the velocity field obtained from the discretised momentum equations be sufficient to couple with cell-centred techniques for the pressure field? The face velocity components would now be defined in terms of the pressure difference across an element face (from 2 points on a hexahedral mesh). Would this weaker definition of the pressure gradient term effect the final solution on distorted meshes?

In both scenarios mass is conserved over the scalar control volume not the momentum control volume. Unlike unstructured staggered grid procedures, the vertex-based and cell-centred control volume face integration points do not coincide, obtaining values at the location required involves interpolation. Estimating vertex-based values from cell-centred values and vice-versa could potentially introduce errors into the solution procedure.

As in co-located formulations the hybrid discretisation has the potential for decoupling to occur and checkerboard pressure fields to emerge. Velocity and pressure are only partially staggered. The velocity components are located at the corners of the pressure control volume and pressure is located at the corners of the momentum control volume. The pressure gradient term, calculated from checkerboarding corner values, would be a zero. The increase in pressure across one half of the control volume would cancel out with the decrease in pressure over the other half of the control volume. Conversely, using linear interpolation to obtain face velocity values would result in a checkerboard velocity field satisfying the continuity equation. In order to prevent spurious pressure predictions the hybrid discretisation of the continuity equation defines a face velocity field which is dependant on adjacent pressure values. Thus, solution of the continuity equation does not allow checkerboard pressure values to emerge.

A detailed account of the hybrid discretisation and solution procedure is given in the following sections.

5.5.1 Pressure - Vertex-Based, Velocity - Cell-Centred

The momentum equations are discretised using cell-centred techniques, no non-orthogonal correction terms are applied. The face velocity components are obtained using linear interpolation from element cell values. The element cell pressure gradient, which enters the momentum equation as a source term, is calculated from nodal values using local shape function derivatives:

$$\left(\frac{\partial p}{\partial x^i}\right)_e = \sum_{j=1}^{n} \frac{\partial N_j}{\partial x^i} p_j \tag{5.42}$$

The momentum equations are solved and components of velocity are obtained at the element centroid. Pseudo velocity components $\hat{u}_e, \hat{v}_e, \hat{w}_e$ and pressure gradient coefficients d_e^u, d_e^v, d_e^w are defined for each element, as in equations (5.16) and (5.17).

A new face velocity field $\tilde{\underline{u}}_f$, dependant on adjacent pressure values, is defined on the face of the vertex-based control volume. Shown in the vertex-based procedure (5.38), and given below for convenience.

$$(\tilde{u^i})_f = (\hat{u^i})_f - (d^{u_i})_f \left(\frac{\partial p}{\partial x_i}\right)_f \tag{5.43}$$

The above equation requires the sub-control volume face values to be defined for the pseudo velocity, pressure gradient coefficient and pressure gradient terms. Since pressure values are located at the mesh nodes, the face pressure gradients can easily be calculated using the derivatives of the interpolation functions, as shown in (5.39).

What is not so obvious is how to obtain vertex-based sub-control volume face values from the element based pseudo velocities and pressure gradient coefficients. To illustrate this, Figure 5.2, shows a two-dimensional mesh element containing faces of the vertex-based sub-control volumes. Assuming element centre values are known but values, ϕ_{scv}, are required at sub-control volume face integration points. If the mesh element shown contains a local maximum value, $\phi_{ele} = 4$, and all surrounding elements have values of $\phi_{nbr} = 3$. Assuming equal distances and linear variation, interpolating from element centre values would give $\phi_{scv} = 3.75$. However, if the mesh was distorted the sub-control volume face integration point would no longer lie on the line connecting adjacent element centre points. A non-conjunctionality, Figure 2.3, correction term would be required. If nodal values are obtained from element values, local linear shape functions can be employed to obtain vertex-based face values. On a structured mesh simple averaging of element values can easily approximate nodal values. However, there is the possibility of extrapolation errors occurring at the boundary. Unstructured meshes require the distance from the element centre to the mesh node to be taken into account as follows,

$$\phi_{node} = \frac{\sum_{i=1}^{ne}(\phi d^{-1})}{\sum_{i=1}^{ne} d^{-1}} \tag{5.44}$$

where d is the distance from the element centre to the mesh node and ne is the number of elements associated with the node.

Using weighted averaging to obtain nodal values and employing shape functions to obtain face values, introduces the potential for local minima and maxima values to be lost. Referring to Figure 5.2 and assuming $\phi_{ele} = 4$ and $\phi_{nbr} = 3$ the interpolated nodal values ϕ_{node} for the element would be $\phi_{node} = 3.25$ giving $\phi_{scv} = 3.25$ which is incorrect. An improved estimate can be obtained by including the element centre value when evaluating ϕ_{scv}. The element centroid forms a corner point of each face. If all other corner points are interpolated from nodal values. The vertex-based face value can then be evaluated from its corner values, which includes the element centroid, in this case giving $\phi_{scv} = 3.625$.

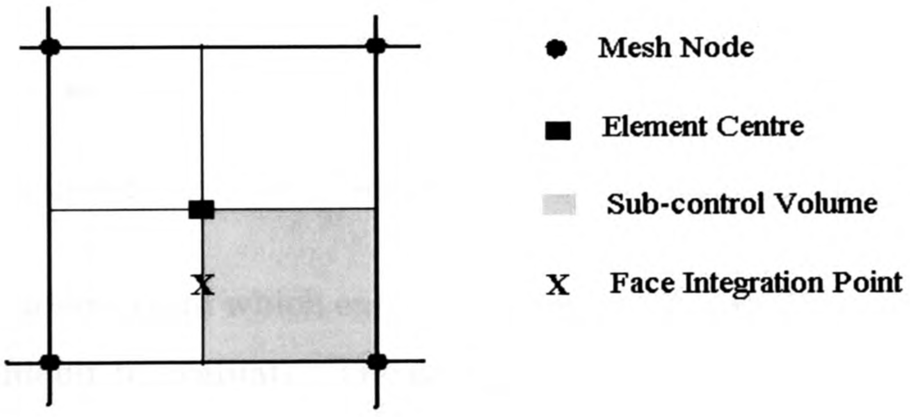


Figure 5.2: Element containing vertex-based control volume face

Once vertex-based face values have been obtained, the vertex-based form of the discretised continuity equation (5.41) can be transformed into a pressure-correction equation, following vertex-based procedures to obtain nodal pressure values. The mass conserving velocity field is the $\underline{\tilde{u}}$ field and all other transport equations should be discretised over the vertex-based control volume. The momentum equations are only used as a means of advancing the solution of $\underline{\tilde{u}}$. The element based velocity $\underline{u}$

is updated using (5.18) as,

$$(u_i)_e = (\hat{u}_i)_e - (d^{u_i})_e \left(\frac{\partial p}{\partial x_i}\right)_e \tag{5.45}$$

where the element pressure gradient is evaluated from nodal values using (5.42).

5.5.2 Pressure - Cell-Centred, Velocity - Vertex-Based

The momentum equations are discretised over the vertex-based control volume using the vertex-based discretisation technique. The face velocity components are simply interpolated from nodal values using local linear interpolation functions,

$$u_f^i = \sum_{j=1}^{n} N_j u_j^i \tag{5.46}$$

where n is the number of nodes associated with the element that contains the face.

The pressure gradient term which enters the momentum equations as a source term is a little more difficult to evaluate. The assumption is made that the pressure gradient across a sub-control volume is approximately equal to the pressure gradient over the element containing the sub-volume. Any errors arising from this assumption would tend to zero as the mesh is refined. Using Equation (5.30) to obtain the pressure gradient over an element leads to the element pressure gradient term being a function of alternate pressure values. Summing the element pressure gradients over the entire vertex-based control volume,

$$\left(\frac{\partial p}{\partial x_i}\right)_{cv} = \sum_{j=1}^{ne} \left(\frac{\partial p}{\partial x_i}\right)_j \tag{5.47}$$

where ne is the number of elements of which the control volume node is a vertex. This results in the vertex-based pressure gradient being a function of a series of adjacent pressure values. Unfortunately, due to cancelling out, a checkerboard pressure field would result in a zero pressure gradient across the vertex-based control volume. A checkerboard pressure field is still seen as a uniform pressure field by the momentum equations.

As with the vertex-based procedure an element face velocity $\underline{\tilde{u}}_e$, dependent upon adjacent pressure values, needs to be defined. Nodal pseudo velocities $\underline{\hat{u}}_n$ are defined as (5.16) and nodal pressure gradient coefficients as:

$$d^{u_i} = a_{u_i}^{-1} \tag{5.48}$$

Since each element face has a vertex located at its corner, element pseudo velocities and pressure gradient coefficients are simply averaged from corner values,

$$(\hat{u}^i_f)_e = \frac{\sum_{j=1}^{cnrs} \hat{u}^i_j}{cnrs} \tag{5.49}$$

and

$$(d^{u_i}_f)_e = \frac{\sum_{j=1}^{cnrs} d^{u_i}_j}{cnrs} \tag{5.50}$$

where $cnrs$ is the number of element face corners.

Giving an equation for the element face velocity $\underline{\tilde{u}}_e$ as,

$$(\tilde{u}^i_f)_e = (\hat{u}^i_f)_e - (d^{u_i}_f)_e(\nabla_{x_i}p)_f \tag{5.51}$$

where the pressure gradient across the element face is evaluated as:

$$(\nabla_{x_i} p)_f = A_f n_{x_i}(p_{adj} - p_{cv}) \tag{5.52}$$

Discretising the continuity equation (5.2) using the $\tilde{\underline{u}}$ field and substituting in (5.51) leads to,

$$\sum_f \rho_f \frac{A_f^2 n_i^2}{a_i}(p_{cv} - p_{adj})_f = \frac{\rho_P^0 V_P^0 - \rho_P V_P}{\Delta t} - \sum_f A_f \rho_f (\hat{\underline{u}}.\underline{n})_f \tag{5.53}$$

where the subscript i in the first term indicates a summation over the three coordinate directions.

Again the pressure equation can be transformed into a pressure correction equation as (5.22). No relaxation is applied to pressure. Mass is conserved over the element and the momentum equations are used as a means of advancing the solution, a standard relaxation factor of 0.7 is applied to velocities. The nodal velocities are updated using (5.18).

5.6 Test Case - Lid-Driven Cavity

The co-located cell-centred, co-located vertex-based and hybrid solution methods, outlined in the previous paragraphs, are applied to the solution of flow only. Solutions are obtained on a Cartesian and distorted meshes with the same number of elements. The results are compared with benchmark solutions and a measure of the error due to mesh distortion is made. The aim is to validate the solution methods and test their ability to handle mesh skewness.

This case considers flow induced in a square cavity by moving the high y boundary wall. All of the other boundary walls are stationary. The initial velocity field was set to zero in the whole domain. The velocity on the moving boundary was set to $1ms^{-1}$ in the x direction. The pressure was fixed to zero in the centre of the cavity. Material properties were set to give a Reynolds number of 100, density $\rho = 1kgm^{-3}$ and kinematic laminar viscosity $\nu_{lam} = 0.01m^2s^{-1}$. A steady state solution was sought. The results obtained are compared against the solutions of Ghia, Ghia and Shin [132], who produced a set of benchmark solutions using the Multigrid technique and 99 elements in both the x and y direction. The cell-centred and vertex-based co-located techniques and the two hybrid discretisation methods are validated with the benchmark solutions using a Cartesian mesh of 99 by 99 elements. Results are compared with solutions obtained employing a much coarser mesh, 35 elements in both the x and y direction, with different degrees of skewness. Figure 5.3 to Figure 5.5 show the meshes employed in the simulations. Mesh 1 being a standard Cartesian mesh, mesh 2 and 3 are distorted versions of mesh 1. The discretisation methods employed are:

1. Velocity and Pressure co-located at element centre (cc)
2. Velocity and Pressure co-located at mesh vertex (vb)
3. Velocity solved at element centre and Pressure at mesh vertex (ve_pn)
4. Velocity solved at mesh vertex and Pressure at element centre (vn_pe)

Velocity profiles are plotted for the u-velocity component, on a line parallel to the moving lid midway up the domain, and for the v-velocity component, perpendicular to the moving lid across the centre of the domain. For comparison purposes the velocity components are all plotted at the same geometric location, the mesh nodes, values being extrapolated from element values when velocity is solved at the element centre.

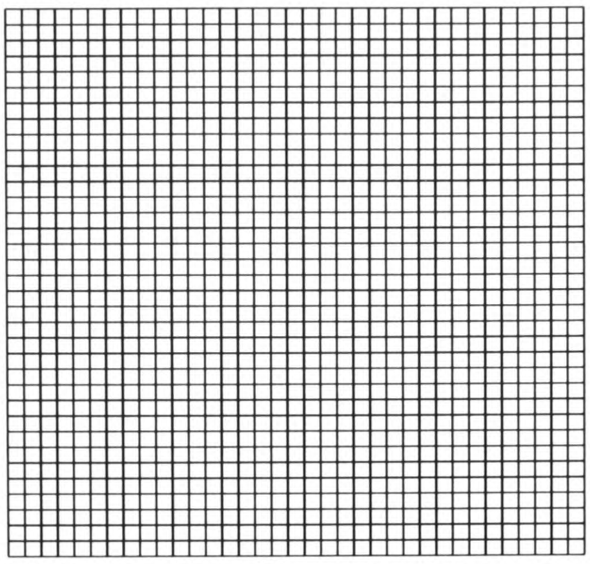


Figure 5.3: Mesh 1

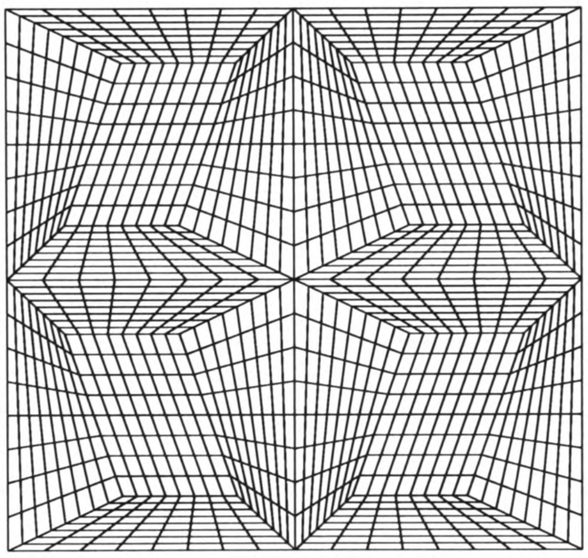


Figure 5.4: Mesh 2

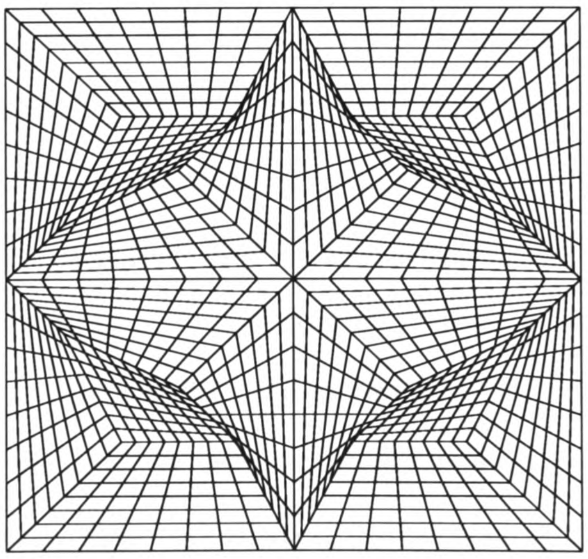
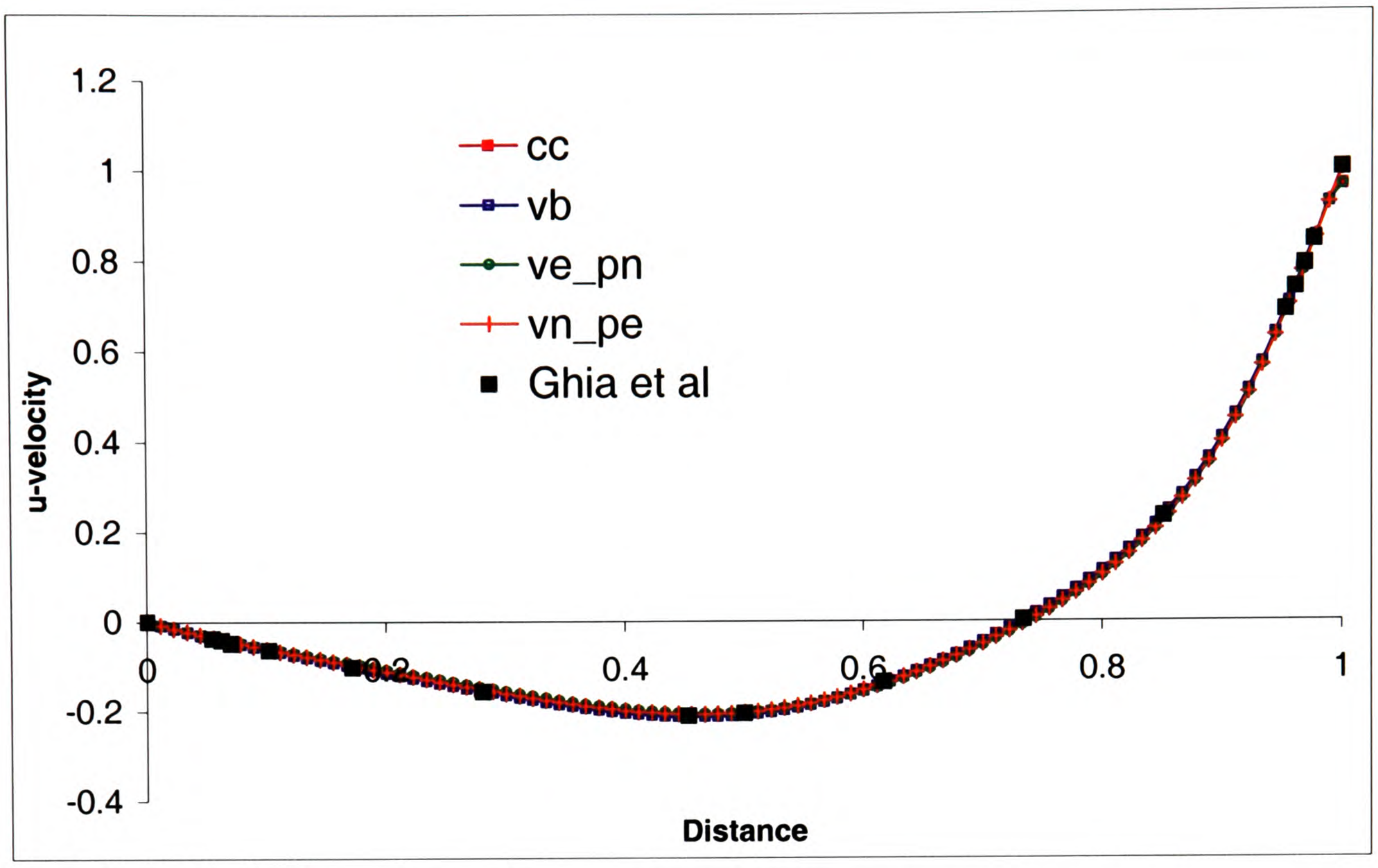


Figure 5.5: Mesh 3

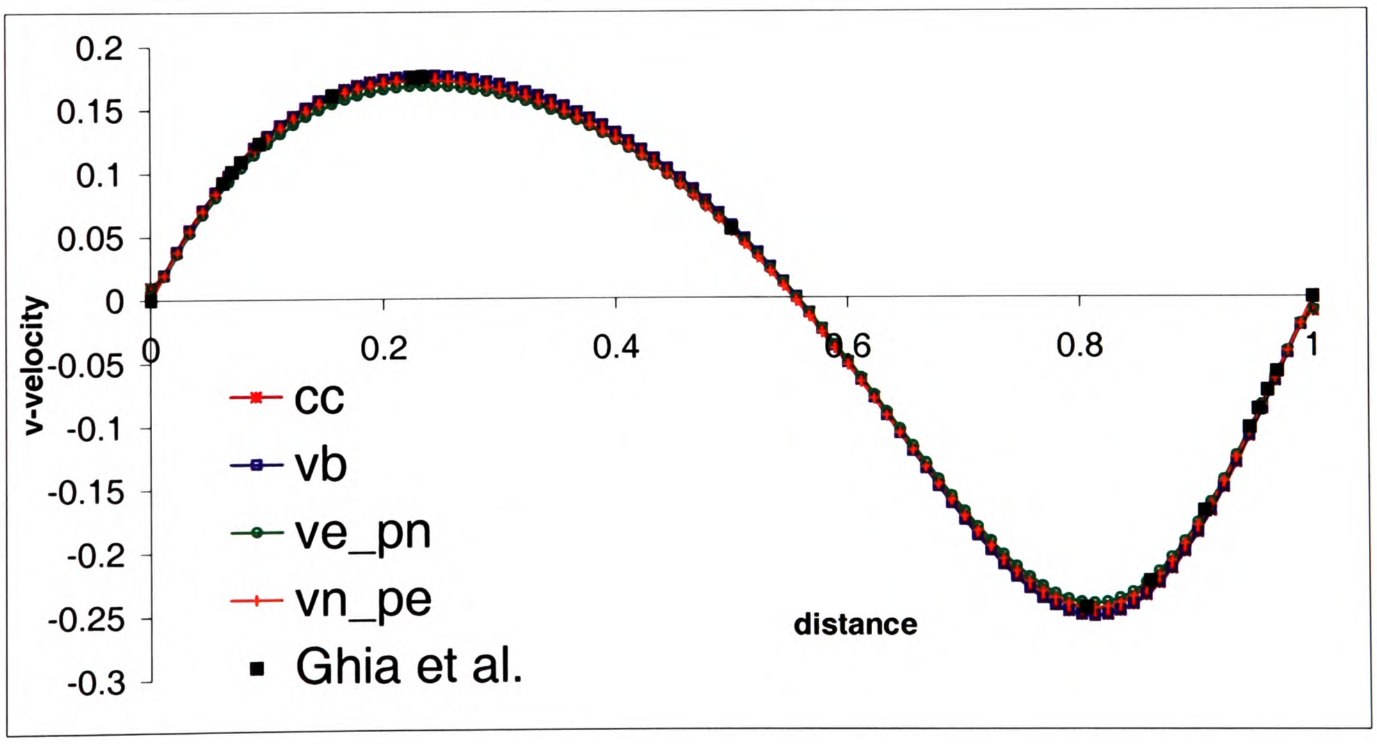
5.6.1 Results

Figure 5.6 shows the simulation results for a Cartesian 99 by 99 element mesh and the benchmark solutions of Ghia, Ghia and Shin. As can be seen from Figure 5.6 all the solution methods are in good agreement with the benchmark solutions.

Figures 5.7 to 5.10, pages 118 to 121 show the results obtained for simulations performed on meshes 1, 2, and 3. Included for comparison purposes are the results obtained on the much finer 99 by 99 element Cartesian mesh. Mesh 1 tests the ability of the solution method to obtain results on a fairly coarse mesh. Meshes 2 and 3 test the ability of the methods to handle distortion in a mesh. For both the distorted meshes 2 and 3, divergence was encountered if cell-centred non-orthogonal correction terms were included. For this reason, the (cc) and (ve_pn) methods discretise the momentum equations without the inclusion of these terms. The (vb) approach, Figure 5.8, handled all the meshes extremely well and good agreement was obtained with the solutions on the 99 by 99 element mesh. The (cc) results, Figure 5.7 failed to resolve the local minimum and maximum values of the v-velocity on mesh 2 and the solution failed on mesh 3 due to divergence. Convergence was achieved for the hybrid formulations on all meshes. The hybrid results seem dependent on the type of mesh employed, with no one method handling both distorted meshes well. Solving (vn_pe), Figure 5.10, gave reasonable results on mesh 2, the u-velocity profiles were in good agreement and the v-velocity profiles came close to achieving local maximum and minimum values. However, on mesh 3 the results degenerated and there were significant errors in both u- and v-velocity profiles. Conversely, using the (ve_pn) formulation, Figure 5.9, the distortion in mesh 3 was handled well and only slight errors were obtained in the solution. Mesh 2 solutions for the u-velocity were reasonable but the v-velocity solutions completely failed to resolve the local maximum and minimum values. A measure of the error due to mesh distortion is given in Table 5.1. The error was calculated using the vertex-based solutions as the base results and the error normalised using the value of the lid velocity.

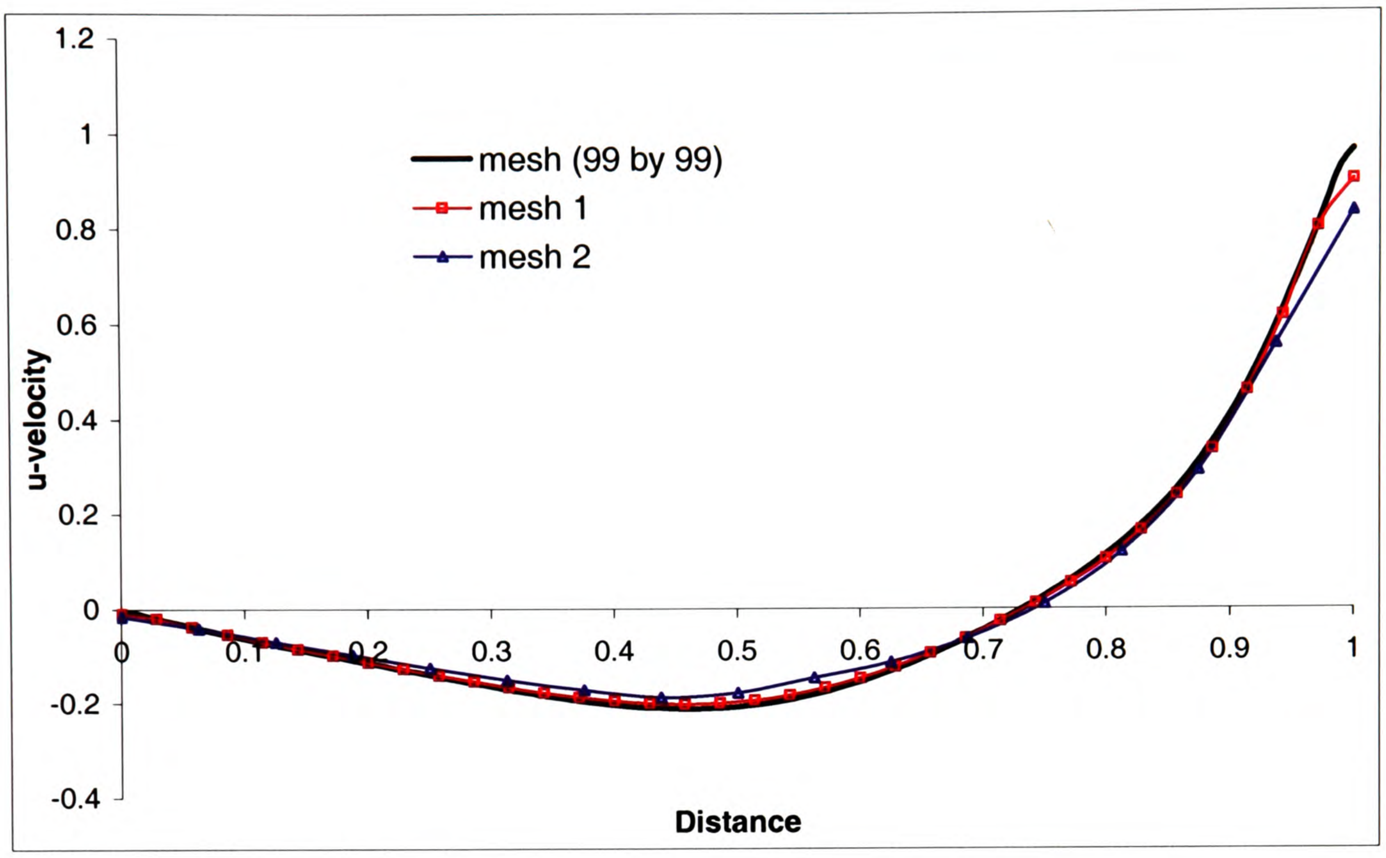


(a) u-velocity profiles

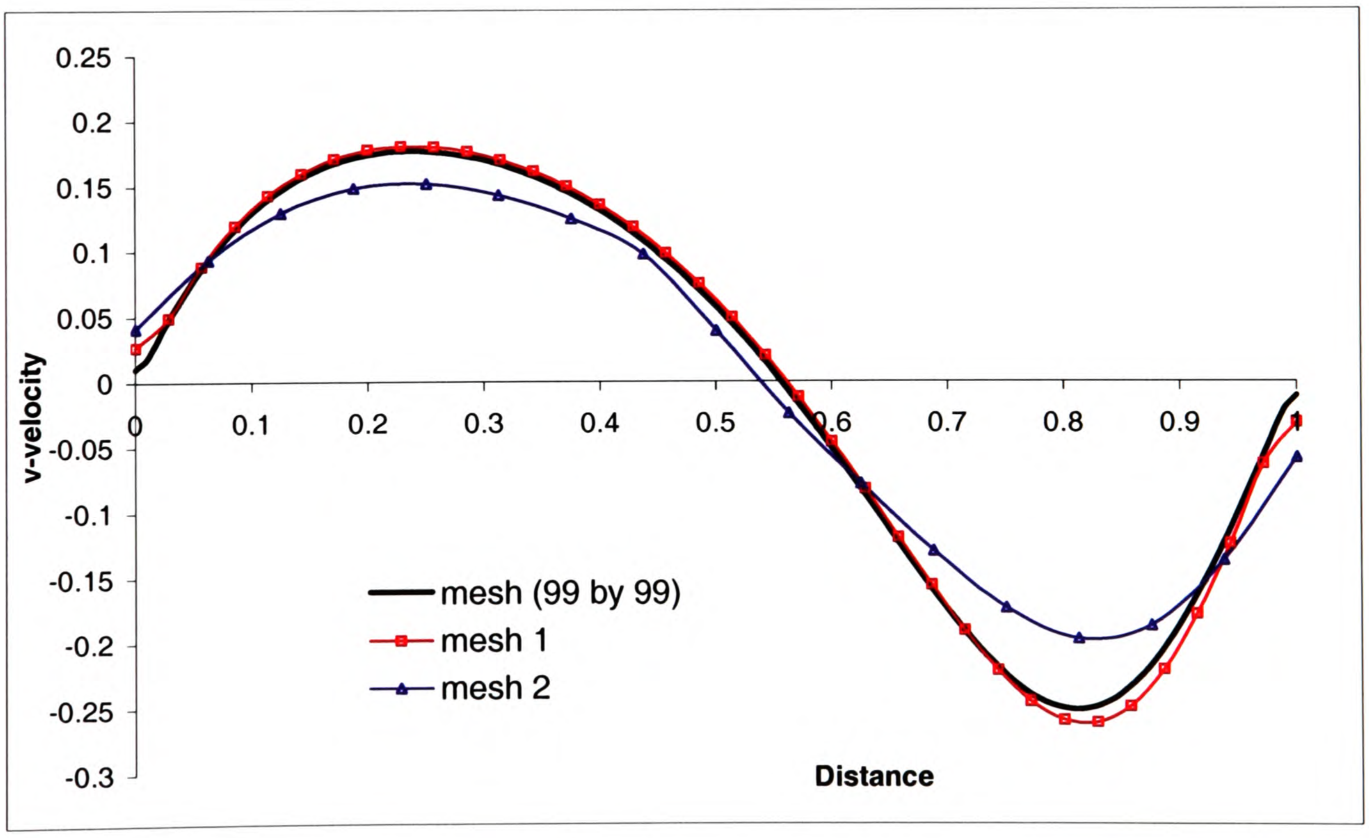


(b) v-velocity profiles

Figure 5.6: Cartesian mesh 99 by 99 elements a) u-velocity b) v-velocity

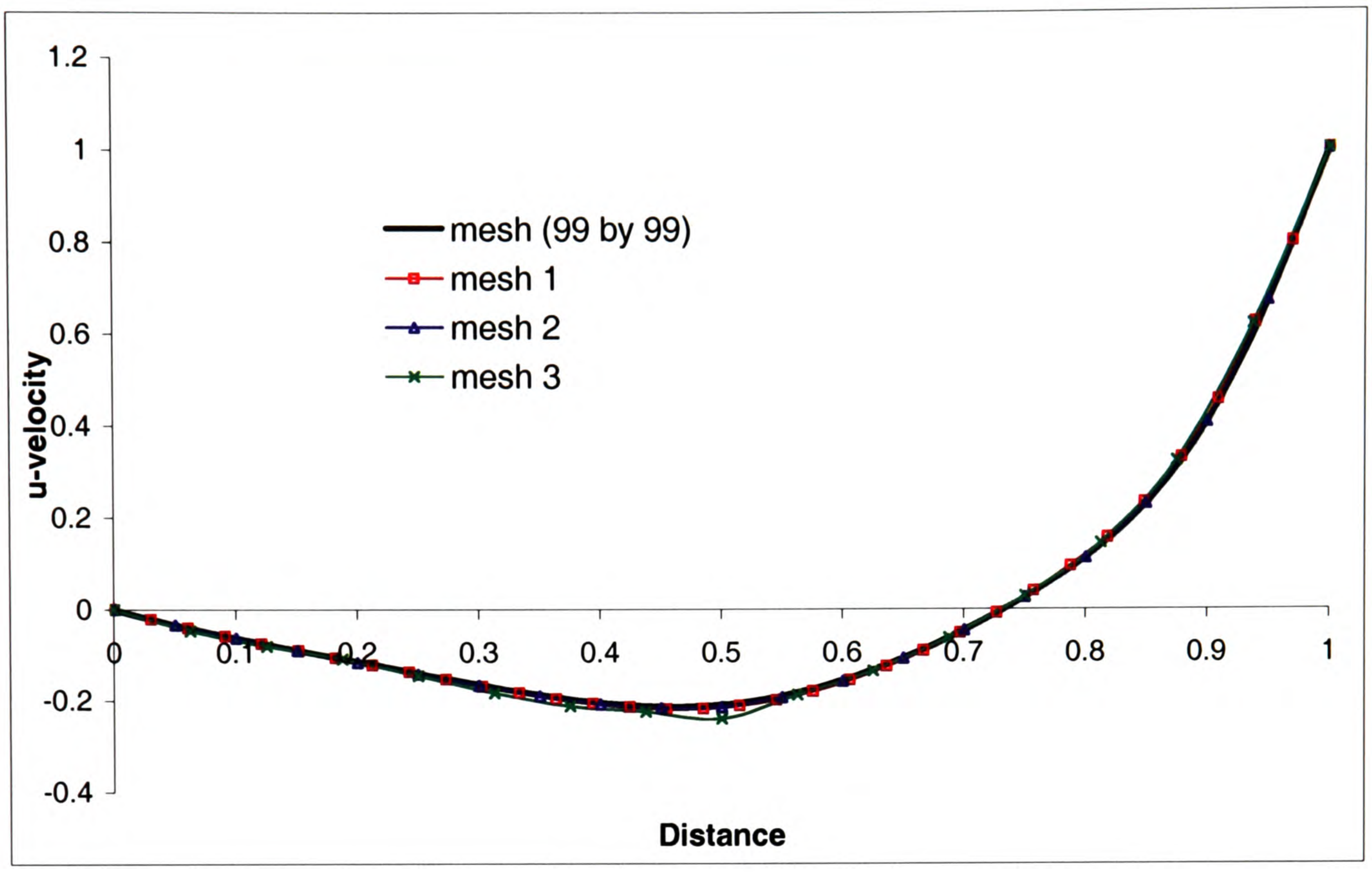


(a) u-velocity profiles

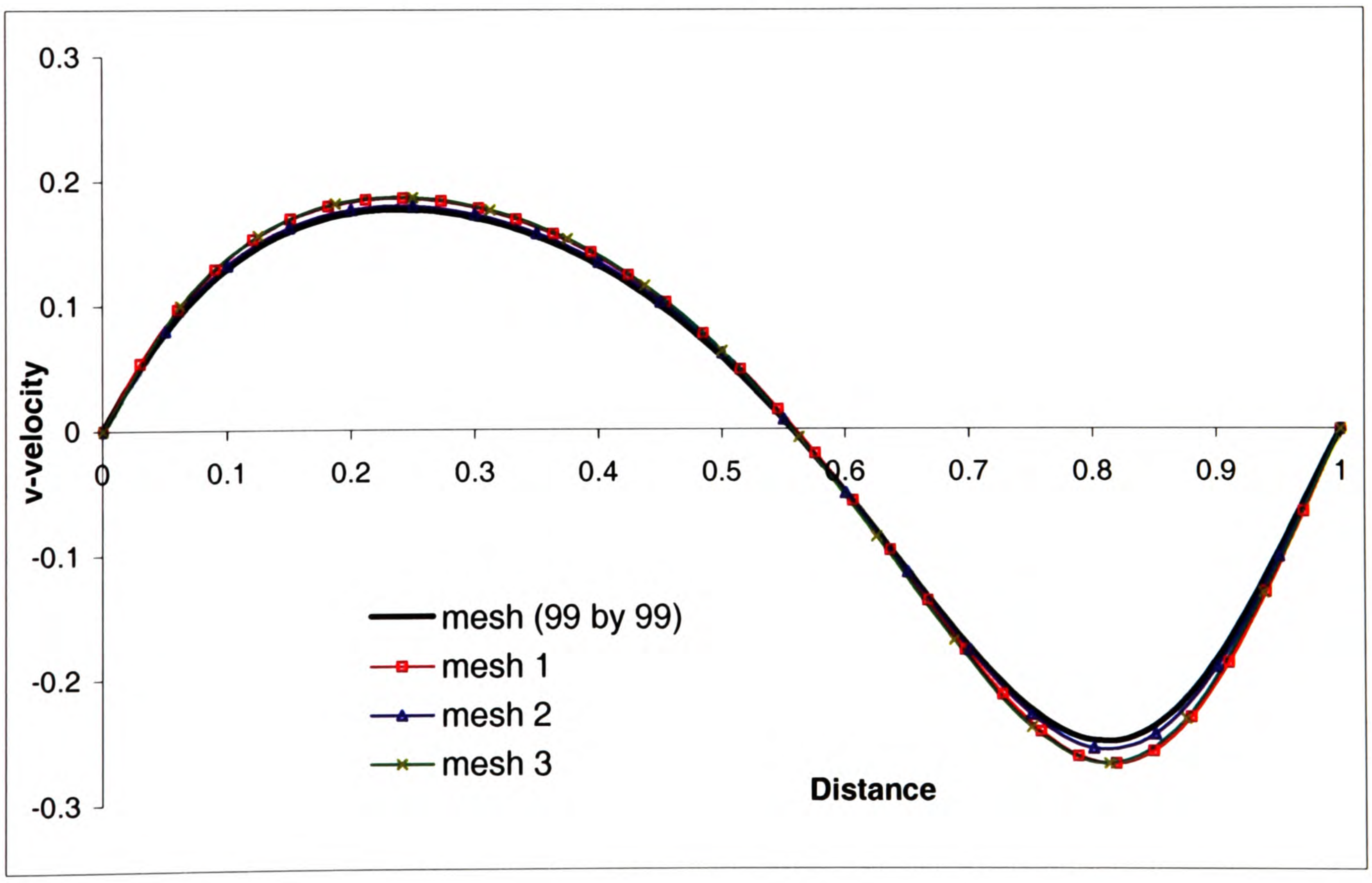


(b) v-velocity profiles

Figure 5.7: Co-located cell-centred method a) u-velocity b) v-velocity

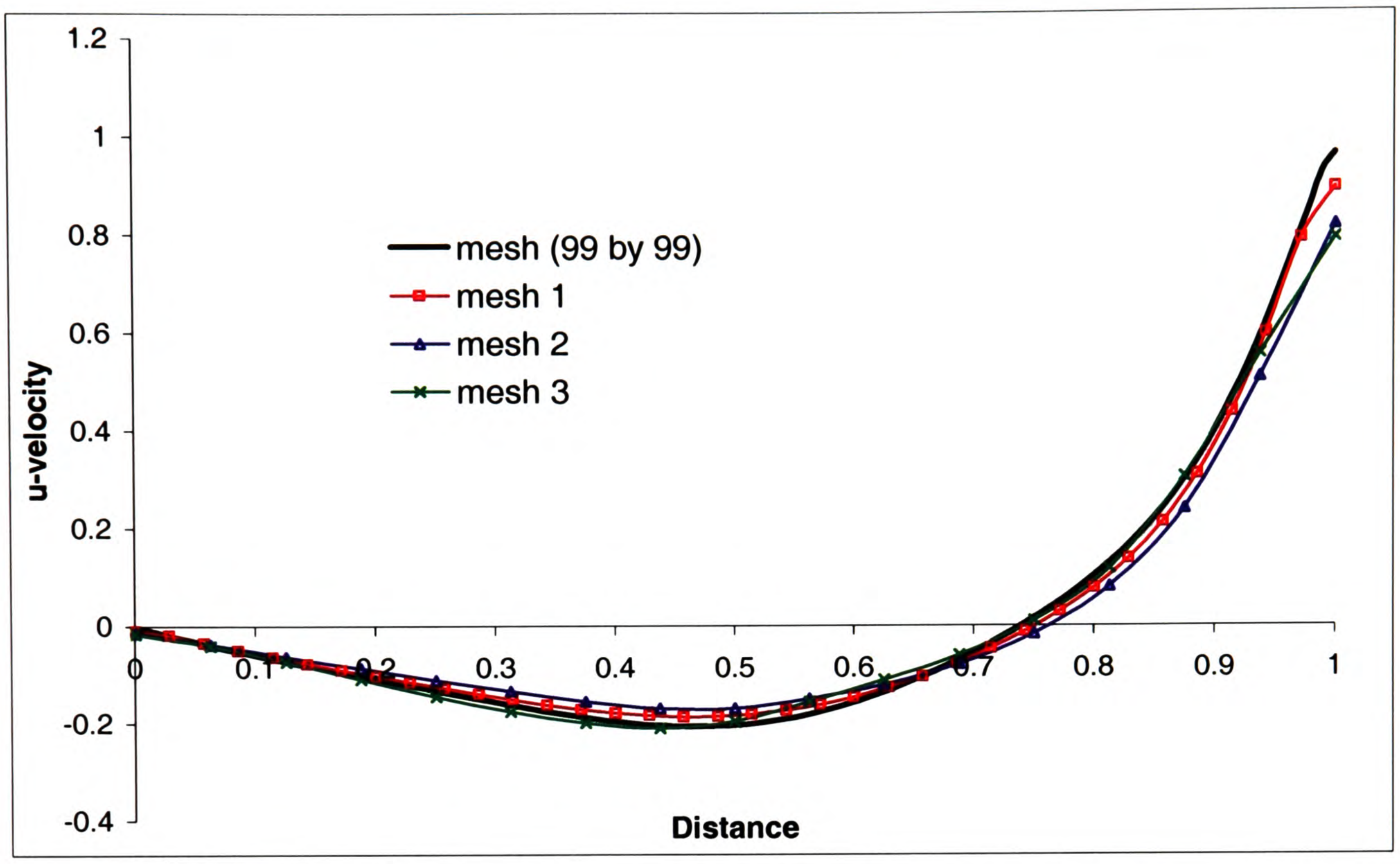


(a) u-velocity profiles

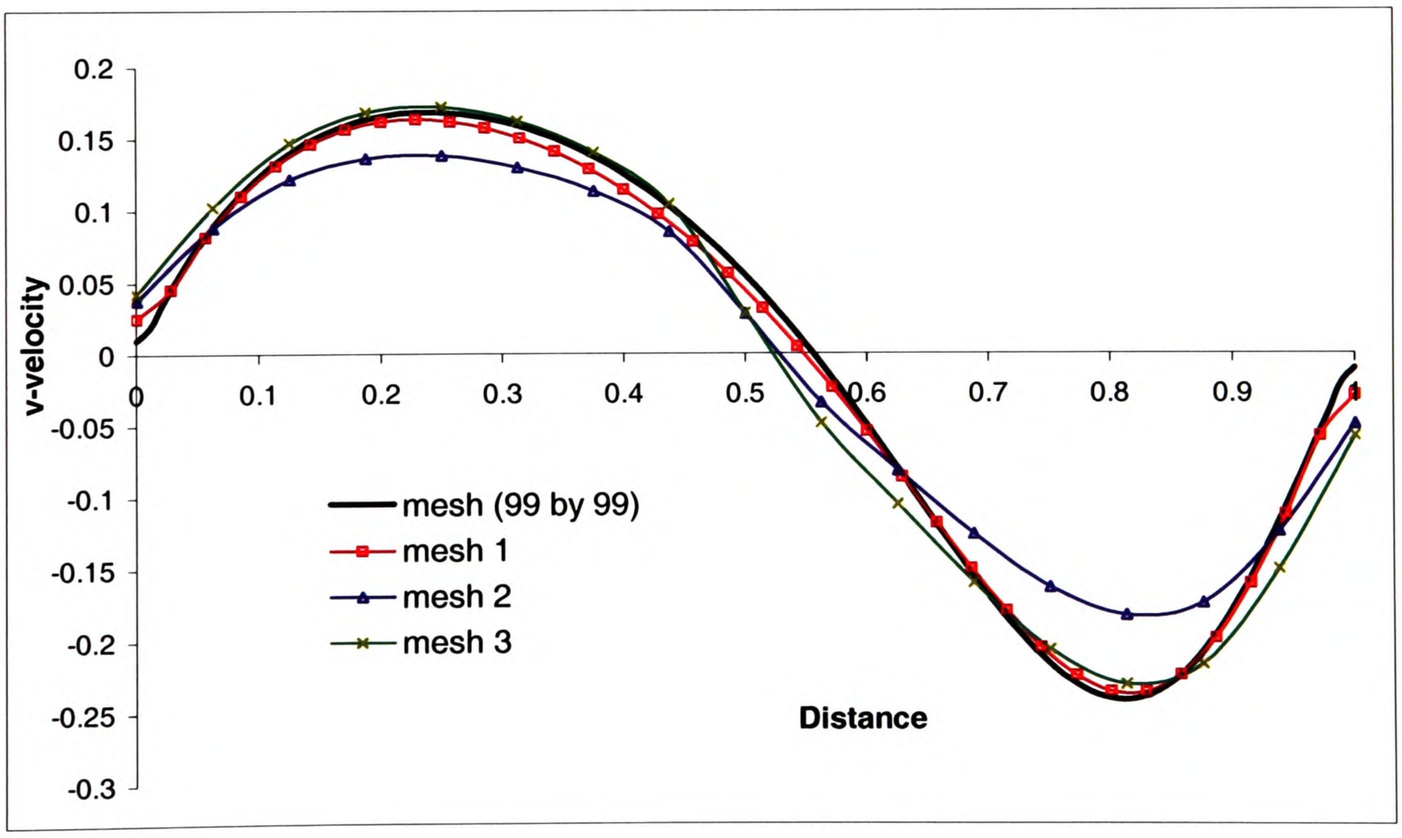


(b) v-velocity profiles

Figure 5.8: Co-located vertex-based method a) u-velocity b) v-velocity

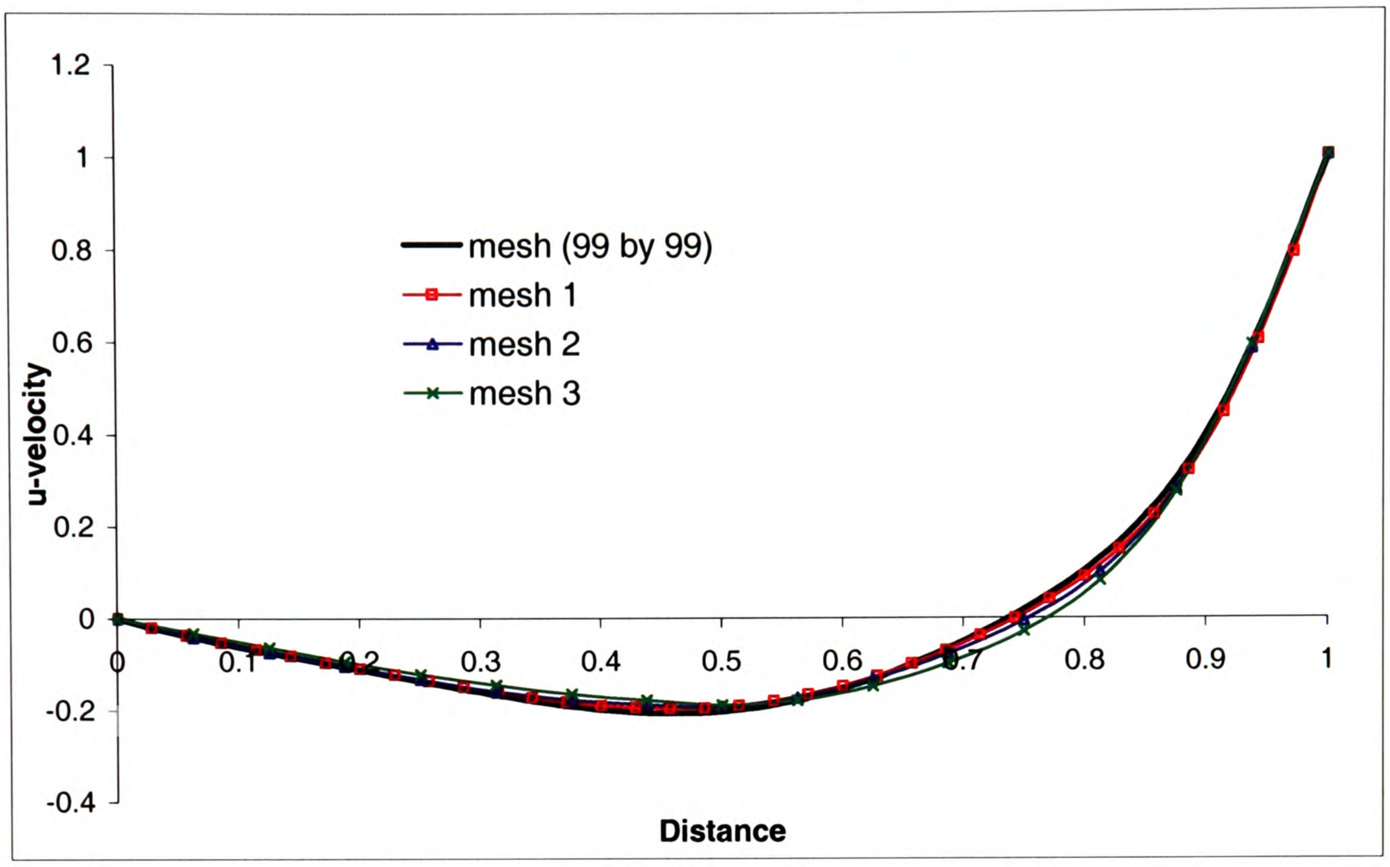


(a) u-velocity profiles

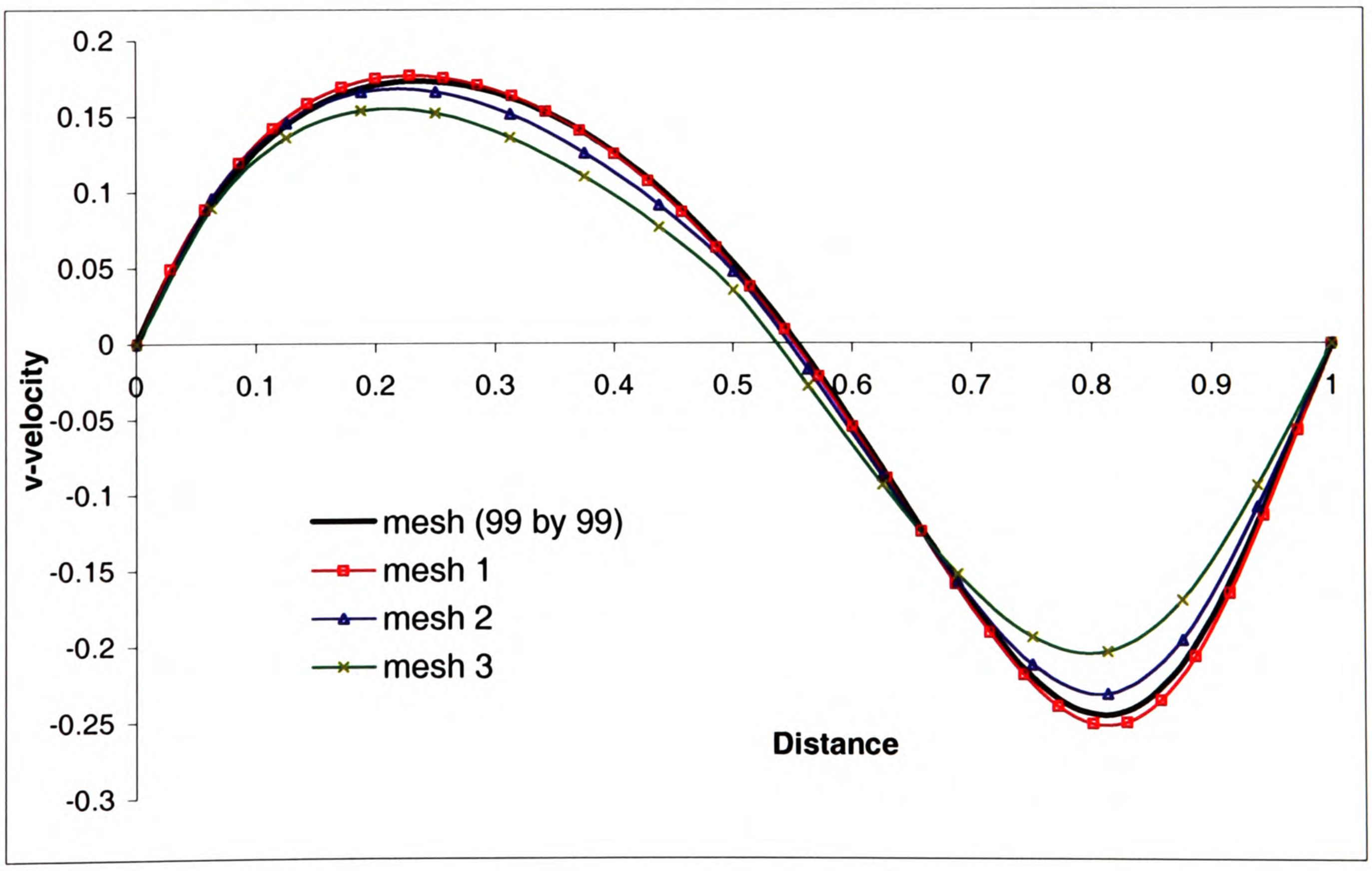


(b) v-velocity profiles

Figure 5.9: Velocity solved cell-centred, Pressure solved vertex-based, *ve_pn* method a) u-velocity b) v-velocity



(a) u-velocity profiles



(b) v-velocity profiles

Figure 5.10: Velocity solved vertex-based, Pressure solved cell-centred, *vn_pe* method a) u-velocity b) v-velocity

cc	u-velocity	v-velocity
mesh 2	2.96×10^{-2}	3.00×10^{-2}
mesh 3	-	-

ve_pn	u-velocity	v-velocity
mesh 2	4.63×10^{-2}	3.72×10^{-2}
mesh 3	3.02×10^{-2}	2.46×10^{-2}

vn_pe	u-velocity	v-velocity
mesh 2	1.57×10^{-2}	1.46×10^{-2}
mesh 3	3.05×10^{-2}	3.18×10^{-2}

Table 5.1: Normalised error due to mesh distortion

5.6.2 Convergence and Run Times

The solutions were considered converged when the L_2 norm of the change in the solution of all the flow variables fell to 10^{-4}. The flow residuals were monitored during the solution procedure and the Log of the L_2 norm is shown as a function of the number of iterations in Figures 5.11 to 5.14 for mesh 1. In the cc solution method, Figure 5.11, only slight oscillations occur in the pressure residuals and convergence was achieved in 112 iterations. Much larger oscillating pressure residuals are encountered when solving using the vb method, Figure 5.12. After approximately 90 iterations the pressure residual oscillations reduce and convergence is achieved in 129 iterations. In both methods only very slight oscillations occur in the velocity residuals. In the *ve_pn* solution method, Figure 5.13, only slight oscillations are encountered in the residuals but convergence is slow requiring 216 iterations. The *vn_pe* solution method, Figure 5.14, produces large oscillations in the pressure and u-velocity residuals but convergence is achieved in 113 iterations.

The moving lid cavity case is a two-dimensional problem but simulations were carried out on a mesh of hexahedral elements. In this case there are over twice as many mesh vertices than elements. Solving vertex-based, variables are resolved in two two-dimensional planes. In order to compare the solution methods, run times are given as an average per variable per solution point and are shown in Table 5.2 for all the meshes investigated. The simulations were performed on a AMD Athlon 1.39 GHz processor. The run times are case and mesh dependent, for the Cartesian mesh, the vb method took approximately 4.5 times the cc method. Of the hybrid methods solving *vn_pe* was the fastest taking approximately 2.5 times longer than cc. The *ve_pn* method was very slow to converge, the number of iteration required to achieve convergence was approximately 1.6-1.9 times the other methods. As a result the run time per variable per solution point was eight times greater than the cc technique. The cc run time remained constant for mesh 1 and 2. The vb and hybrid run times increased when solving on the distorted meshes 2 and 3.

	mesh 1	mesh 2	mesh 3
cc	5.44×10^{-4}	5.44×10^{-4}	-
vb	2.44×10^{-3}	4.88×10^{-3}	5.53×10^{-3}
ve_pn	4.36×10^{-3}	7.93×10^{-3}	7.33×10^{-3}
vn_pe	1.4×10^{-3}	2.65×10^{-3}	2.5×10^{-3}

Table 5.2: Run times per variable per solution point

The memory requirements for vertex-based solutions are far greater than cell-centred solutions. Storing the geometric requirements for a vertex-based control volume requires approximately 590 bytes compared to only 179 bytes for a cell-centred control volume. Extra memory is also required in the solution of a vertex-based variable. For the solution of the moving lid problem on meshes comprising of 8100 elements and 16562 vertices the average memory requirements per solution point was as follows: cc - 92 bytes, vb - 245 bytes, *ve_pn* - 381 bytes and *vn_pe* - 336 bytes.

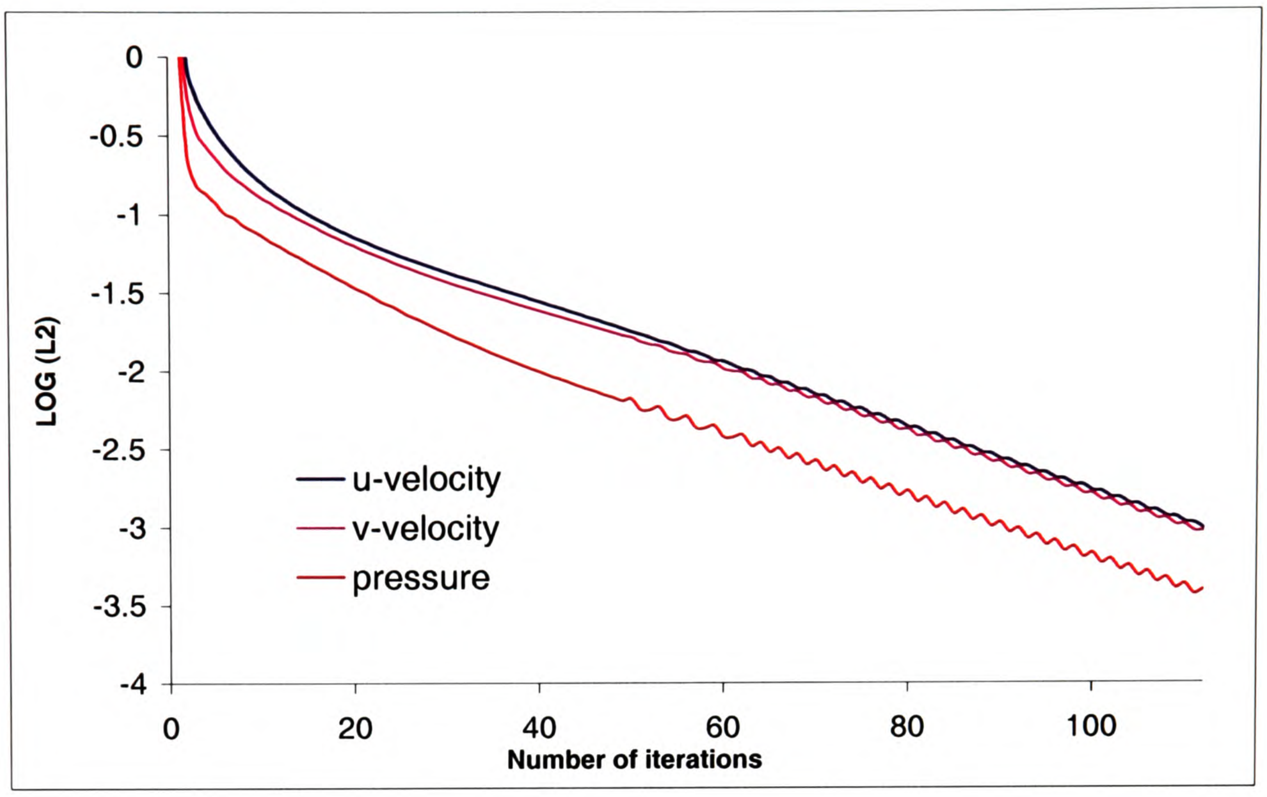


Figure 5.11: Co-located cell-centred (cc): Solution residual history .vs. iterations

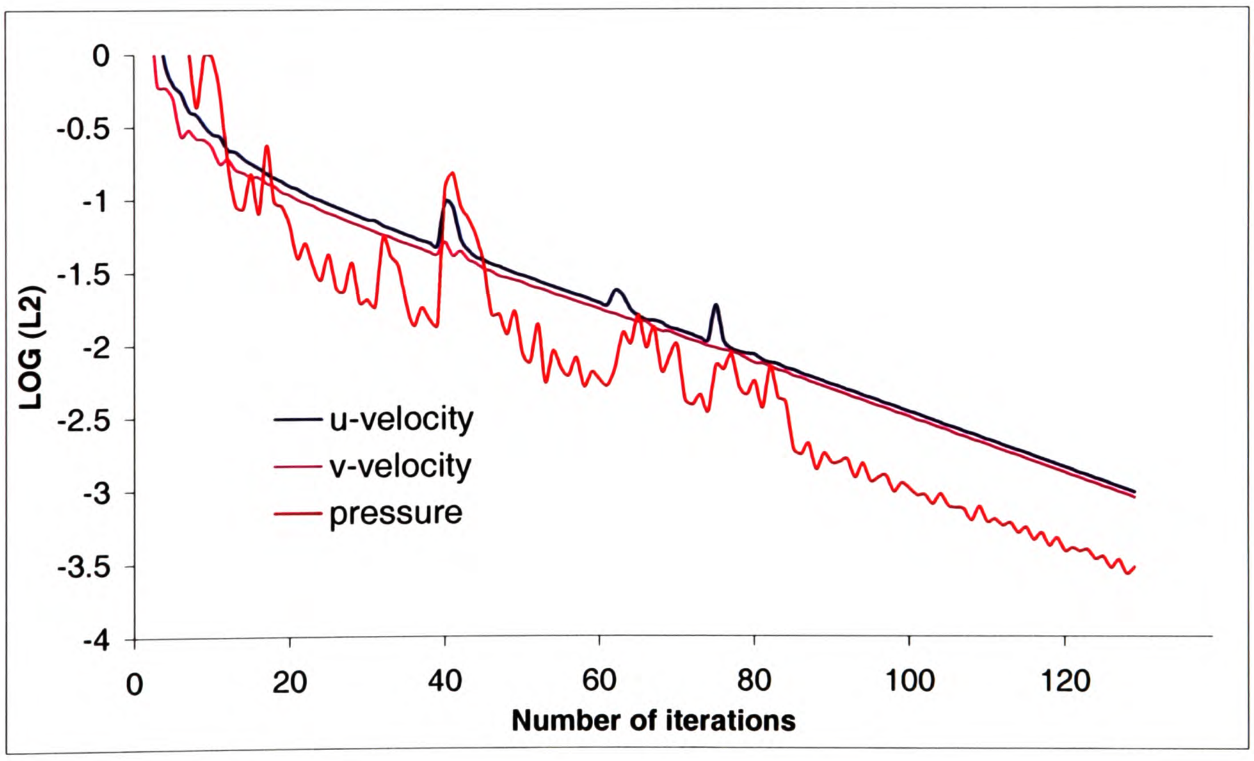


Figure 5.12: Co-located vertex-based (vb): Solution residual history .vs. iterations

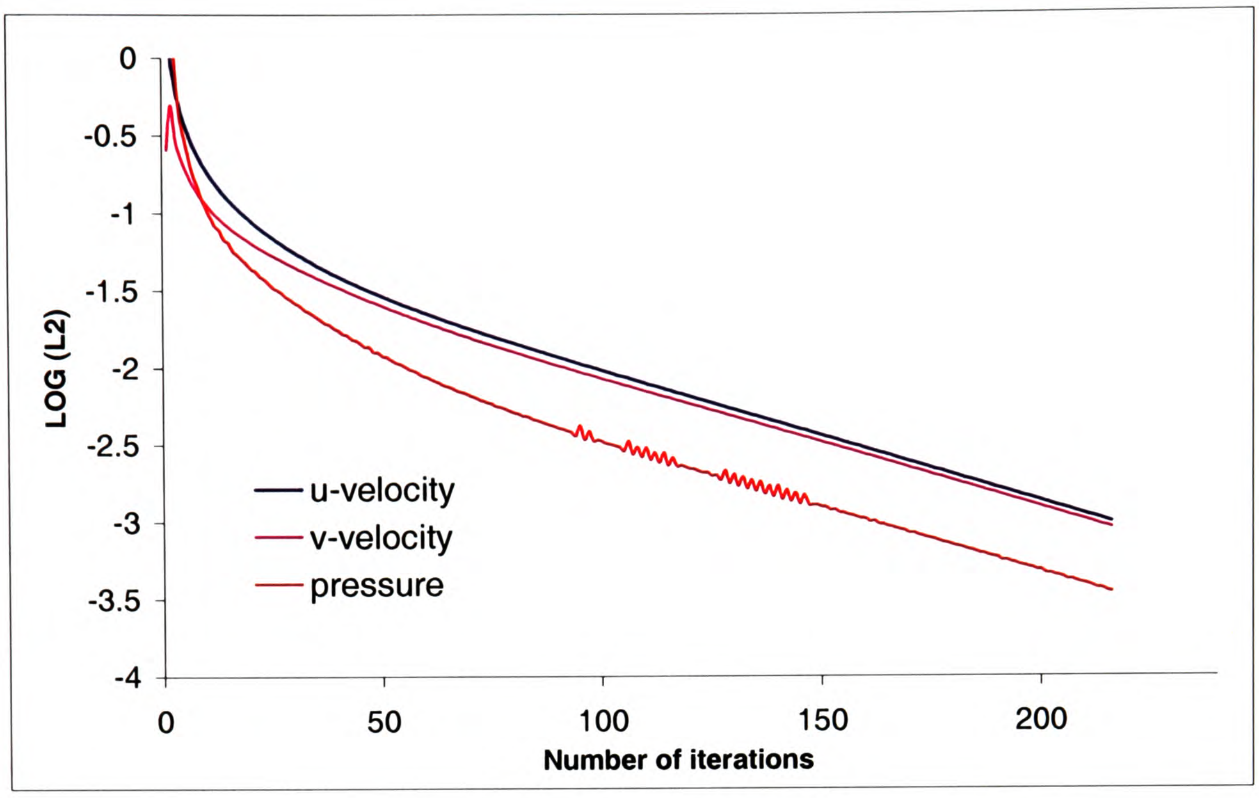


Figure 5.13:

Velocity cell-centred, pressure vertex-based (*ve_pn*): Solution residual history .vs. iterations

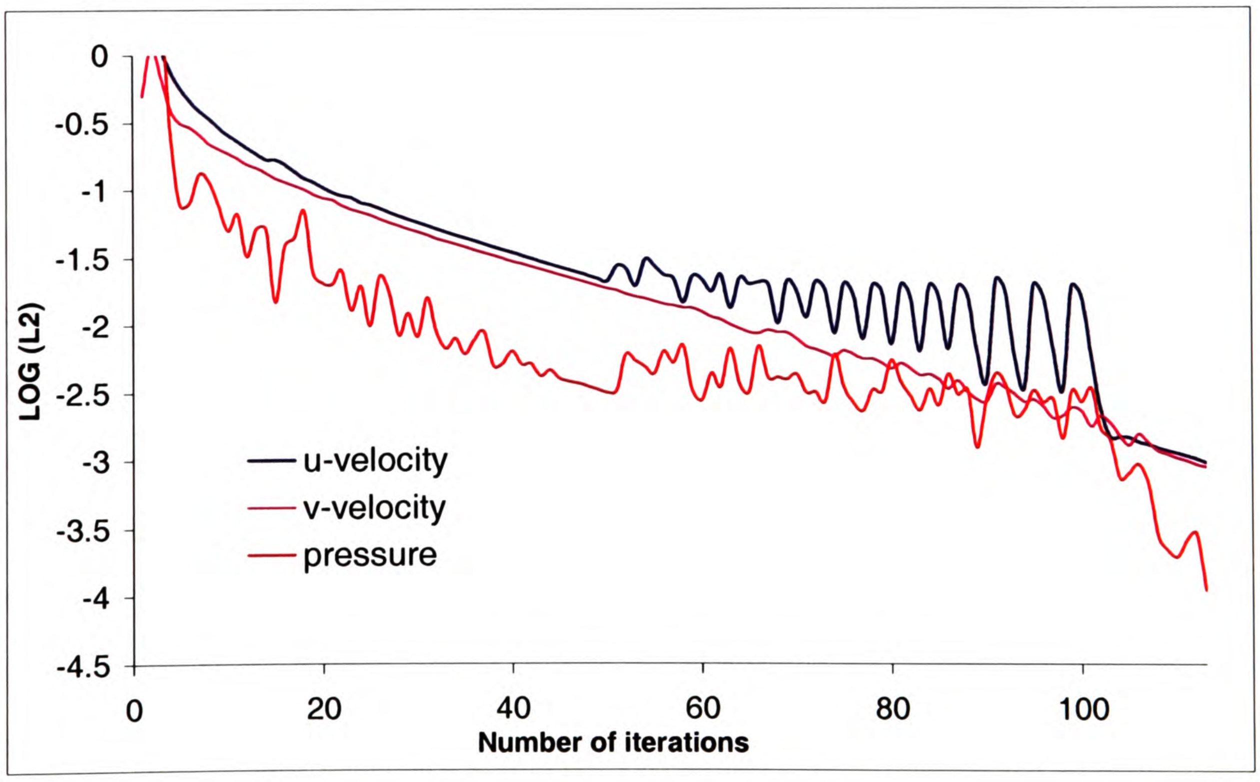


Figure 5.14:

Velocity vertex-based, pressure cell-centred (*vn_pe*): Solution residual history .vs. iterations

5.6.3 Discussion

The co-located vertex-based solution appears to handle mesh distortion with ease. No significant difference in the simulation results is obtained on a distorted mesh compared to those on a standard Cartesian mesh. The co-located cell-centred method struggles to produce solutions on distorted meshes. The inclusion of non-orthogonal correction terms, in an effort to improve solutions, can introduce more instabilities into the solution process. The cell-centred results can be significantly improved on meshes where the inclusion of correction terms produce solutions. However, on arbitrary distorted meshes divergence is often encountered. Solving cell-centred without accounting for non-orthogonality in the mesh produces erroneous or diverging results.

Solving the pressure vertex-based allows for pressure gradients to be more accurately represented on distorted meshes. The vertex-based method (5.39) calculates the face pressure gradient from eight vertex values for a hexahedral element whereas the cell-centred method (5.52) calculates face gradients from only two adjacent element values. Pseudo-velocities, not solutions, are being obtained directly from the momentum equations. The velocity components are updated from pseudo-velocities and vertex-based pressure gradients. Improving the resolution of the pressure field on a non-orthogonal mesh directly influences the velocity components. Non-orthogonal errors in the discretised momentum equations are included in the pseudo-velocities and must pass into the final solution. Using vertex-based techniques for pressure appears to allow the solution on distorted meshes which defeat cell-centred techniques. However, the solutions obtained include non-orthogonal errors. Solving the momentum equations vertex-based allows good resolution of pseudo-velocities to be obtained on distorted meshes but cell-centred face pressure gradients will include non-orthogonal errors.

5.7 Closure

The vertex-based discretisation technique has been extended to the solution of laminar flow problems. Methods used to ensure checkerboard pressure fields cannot emerge have been detailed. The iterative solution methods used in the vertex-based and cell-centred techniques have been outlined. Hybrid flow solution methods have been developed and discussed. Although hybrid methods enable solutions on distorted meshes where purely cell-centred solutions fail, non-orthogonal errors degrade the solution. The vertex-based method gave good results on all meshes employed.

Chapter 6

Combined Method: Vertex-Based-Cell-Centred

Within the finite volume multi-physics framework of PHYSICA there are a number of well established physics models that use cell-centred discretisation techniques for transported properties. If a flow field is obtained using vertex-based techniques could these vertex-based velocities be employed in the transport of cell-centred properties? The aim of this chapter is to show the capabilities of combining the vertex-based flow code with cell-centred discretisation techniques for other transported variables.

As illustrated in the solution of the lid-driven cavity, section 5.6, solving flow using vertex-based discretisation, where pressure and velocity components are co-located at the mesh vertices, allows for solutions on highly distorted non-orthogonal meshes. The extra computational cost per solution point, of solving flow vertex-based is case and mesh dependent. The moving lid problem required approximately 2.6 times more memory and took approximately 4.5 times longer to run than the cell-centred solutions. This extra computational time can be offset by its ability to produce good solutions on all types of meshes. The overall modelling time can be reduced by savings made in not having to generate highly orthogonal meshes. In complex flow simulations, if all transported quantities were solved using vertex-based techniques, escalating computational requirements would start to exceed time benefits in the mesh generation process. One solution to this problem of escalating memory and computational requirements, could be to solve the hydrodynamic variables using vertex-based procedures and all other transported quantities using cell-centred techniques. Having a good resolution of the flow field should enable the solution of other transported quantities on complex geometries.

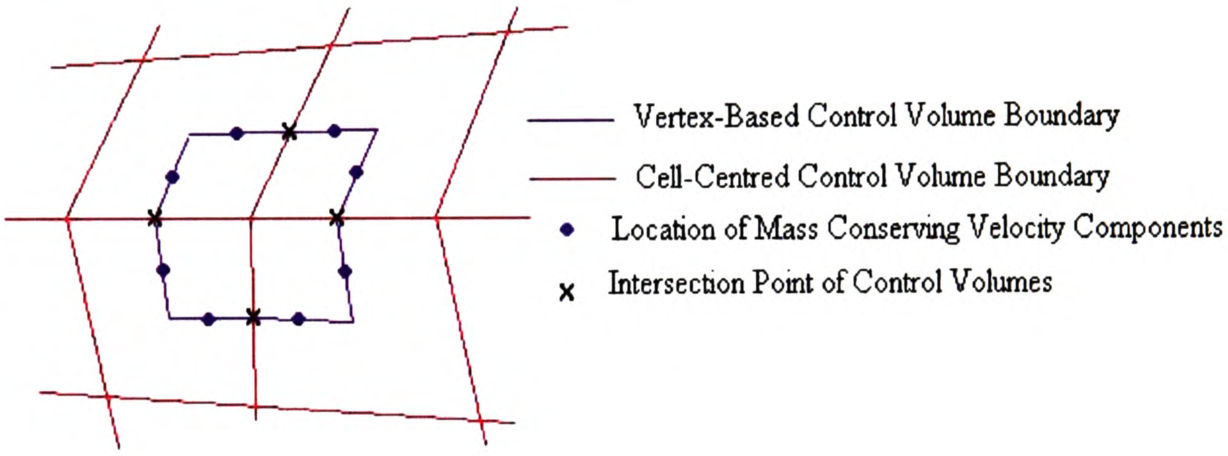

Vertex-Based Control Volume Boundary
Cell-Centred Control Volume Boundary
● Location of Mass Conserving Velocity Components
x Intersection Point of Control Volumes

Figure 6.1:

Location of mass conserving velocity components and intersection points of vertex-based and cell-centred control volume boundaries

Resolving the flow field vertex-based means that mass is conserved on the boundary of the vertex-based control volume. Since the element face centroid is a point on the boundary of the vertex-based control volume, indirectly mass is also conserved over the mesh element. The problem is how to obtain this mass conserving velocity component located at the element face centroid. Figure 6.1 shows the element cell-centred and vertex-based control volumes, the location of the mass conserving velocity components and the control volume face intersection points. All known mass conserving velocity values are located at the centre of a vertex-based face. Interpolating from these known values to obtain values at the intersection points introduces the potential for error. As mass conservation is enforced over the vertex-based control volume, any errors resulting from interpolating for the element face values would also decrease. In order to evaluate how well mass conservation is enforced over an element control volume, the continuity equation is also discretised over an element and residuals monitored.

In this chapter, the results are shown for a test case involving thermally driven flow. This case combines vertex-based flow with cell-centred heat transfer through the simulation of natural convection in a cavity. The results are compared to solutions obtained using purely vertex-based discretisation for all variables and validated with benchmark solutions. The simulations are run on a uniform Cartesian mesh and distorted versions of the mesh. The degradation of the solution, using purely cell-centred techniques, on a distorted mesh is illustrated. A measure of the error on the distorted meshes is given for all methods where solutions were achieved. In the combined vertex-based/cell-centred code, mass balance is enforced over the vertex-based control volume and extrapolated to an element control volume level. The extrapolated mass balance over an element control volume is investigated and compared with co-located cell-centred element mass conservation.

6.1 Thermally Driven Flow

De Vahl Davis and Jones [139] suggested that buoyancy-driven flow in a square cavity would be a suitable validation test case for CFD codes and published a set of benchmark solutions along with numerical solutions from thirty groups. The contributions used a variety of numerical techniques and mesh sizes, uniform and non-uniform. Many of the contributions suffered declining quality with increasing Rayleigh number, often despite the use of mesh refinement. A general conclusion of the study was:

> *'Although there are accurate contributions using both FEM and FDM,*

> *the former are by and large rather better. There is a lower tendency among the FEM entries towards degradation of performance with increasing Rayleigh number and a much lower number of contributions containing obvious major errors.'*

These results may depend on the type of mesh employed. A mesh with a denser distribution of mesh points in critical locations will probably lead to improved accuracy. These benchmark solutions give the maximum u- and v-velocity values along central planes for a number of Rayleigh numbers. The benchmark results are plotted along with the simulation results for comparison.

This problem involves buoyancy-driven flow in a square cavity. The fluid contained in the cavity is assumed incompressible and initially stationary. Thermal gradients across the solution domain result from opposing walls of differing temperatures. These thermal gradients lead to buoyancy forces which create flow. The buoyancy forces are calculated using the Boussinesq approximation. This approximation results in a source per unit volume of the form,

$$S_i = -\rho\beta g_i(T - T_{ref}) \tag{6.1}$$

where β is the thermal coefficient of volumetric expansion, g_i is the component of gravity in the i'th direction and T_{ref} is the reference temperature.

The source per unit volume S_i is calculated for each element and enters the i'the momentum equation as,

Cell-Centred:

$$(S_i)_{cv} = S_i V_{cv} \tag{6.2}$$

Vertex-Based:

$$(S_i)_{cv} = \Sigma_{scv} S_i V_{scv} \tag{6.3}$$

The simulations are run using a uniform Cartesian mesh, mesh 1 Figure 5.3 and two meshes with different degrees of skewness, mesh 2 Figure 5.4 and mesh 3 Figure 5.5 shown in Chapter 5, page 115. Meshes 2 and 3 are distorted versions of mesh 1. The solution domain consists of a 1m square and the boundary conditions are shown in Figure 6.2.

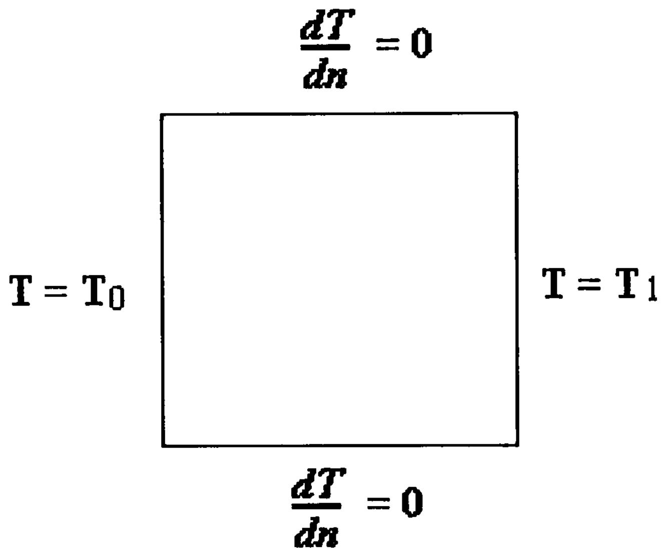


Figure 6.2: Boundary conditions for thermally driven flow

The material properties are those for air at $300K$, as shown below,

Specific Heat $1.007 \times 10^{03} m^2 s^{-2} K^{-1}$

Thermal Conductivity $2.63 \times 10^{-02} kgms^{-3} K^{-1}$

Density $1.161 kgm^{-3}$

Thermal Expansion $3.0 \times 10^{-03} K^{-1}$

Laminar Kinematic Viscosity $1.59 \times 10^{-05} m^2 s^{-1}$

Gravity $9.81 m s^{-2}$

with gravity acting in the negative y direction.

The Rayleigh number, Ra, is defined as,

$$Ra = \frac{\beta |\underline{g}| \Delta T L^3}{v k / (C_p \rho)} \tag{6.4}$$

where ΔT is the temperature difference between the two vertical walls.

The desired Rayleigh number is obtained by varying the temperatures on each of the vertical walls. The simulations were run for Rayleigh numbers of 10^3, 10^4, 10^5 and 10^6. Plots of the u-velocity along the central vertical plane and the v-velocity along the central horizontal plane for each Rayleigh number are shown along with the benchmark maximum values published by de Vahl Davis.

Figures 6.3 and 6.4, page 136, show u-velocity and v-velocity profiles obtained on a uniform mesh 1, consisting of 30 by 30 elements and 90 by 90 elements. Results are shown for purely cell-centred solutions (cc), purely vertex-based solutions (vb) and for the combined vertex-based/cell-centred solution ($vb - cc$) for Rayleigh numbers of $Ra = 10^3$, $Ra = 10^4$, $Ra = 10^5$ and $Ra = 10^6$.

The normalised distances and velocities are calculated from,

$$X = \frac{x}{L} \quad U = \frac{uL}{k/(C_p \rho)} \tag{6.5}$$

$$Y = \frac{x}{L} \quad V = \frac{vL}{k/(C_p \rho)}$$

$$\tag{6.6}$$

where L is the length of the sides of the cavity.

Table 6.1, page 135, shows a comparison of the simulation results, on a uniform 30 by 30 and a 90 by 90 element mesh, against the benchmark solution. The U_{max} and V_{max} values are the maximum value of the normalised velocity component along central planes. The Y_{max} and X_{max} value being the normalised position of this maximum value. The percentage error of the simulation results against benchmark solutions are shown in brackets for each solution method.

For the lower Rayleigh numbers 10^3 and 10^4 there are no significant differences between the solutions obtained on either mesh. All solution methods resolved the maximum benchmark values with less than 1% error. For a Rayleigh number of 10^5, mesh refinement reduces the percentage error for all the solution methods to less than 1.6%. On the 90 by 90 element mesh the (vb) results had less than 0.3% error with the u- and v- velocity benchmark solutions. The (cc) method had the largest percentage error in the u-velocity maximum values, 3.25% reducing to 1.59%. The (cc) method also contained the largest error in the v-velocity values, 3.93% for the 30 by 30 mesh but reduced to 0.4% with mesh refinement. For a Rayleigh number of 10^6 the (cc) method performed well on the fine mesh, resolving maximum values with less than 1% error. On the coarser 30 by 30 mesh the (cc) method completely fails to resolve the maximum v-velocity value with nearly 10% error. The (vb) method contained the largest error for the u-velocity value, 5.5% reducing to 3.4%, but resolved the maximum v-velocity with 1.5% error. The $(vb - cc)$ method performed extremely well on both meshes with a maximum error of 1.8% occurring on the 30 by 30 mesh. For all Rayleigh numbers the solutions could possibly be improved by positioning solution points at critical locations. On the coarser grid the error in the X_{max} and Y_{max} locations of the maximum values are significantly higher than those obtained on the finer mesh.

Results obtained on mesh 1, mesh 2 and mesh 3 are shown on pages 137 and 138 for vb and $vb - cc$ solutions. The simulations were run until the L_2 norm of the change

$Ra = 10^3$	U_{max}	Y_{max}	V_{max}	X_{max}
de Vahl Davis	3.649	0.187	3.696	0.178
vb (90 by 90)	3.629 (0.55%)	0.189 (1.0%)	3.679 (0.44%)	0.178 (0.12%)
vb (30 by 30)	*3.625 (0.66%)*	*0.199 (6.7%)*	*3.676 (0.52%)*	*0.173 (2.5%)*
cc (90 by 90)	3.637 (0.32%)	0.189 (1.0%)	3.713 (0.46%)	0.178 (0.12%)
cc (30 by 30)	*3.609 (1.08%)*	*0.199 (6.7%)*	*3.662 (0.92%)*	*0.173 (2.5%)*
vb-cc (90 by 90)	3.638 (0.29%)	0.189 (1.0%)	3.686 (0.27%)	0.178 (0.12%)
vb-cc (30 by 30)	*3.625 (0.65%)*	*0.199 (6.7%)*	*3.680 (0.44%)*	*0.173 (2.5%)*

$Ra = 10^4$	U_{max}	Y_{max}	V_{max}	X_{max}
de Vahl Davis	16.178	0.177	19.617	0.119
vb (90 by 90)	16.199 (0.13%)	0.178 (0.43%)	19.559 (0.29%)	0.122 (2.6%)
vb (30 by 30)	*16.156 (0.14%)*	*0.173 (2.0%)*	*19.529 (0.45%)*	*0.113 (5.4%)*
cc (90 by 90)	16.190 (0.08%)	0.178 (0.43%)	19.587 (0.15%)	0.122 (2.6%)
cc (30 by 30)	*16.091 (0.53%)*	*0.173 (2.0%)*	*19.437 (0.91%)*	*0.113 (5.4%)*
vb-cc (90 by 90)	16.194 (0.10%)	0.178 (0.43%)	19.570 (0.24%)	0.122 (2.6%)
vb-cc (30 by 30)	*16.221 (0.26%)*	*0.173 (2.0%)*	*19.561 (0.28%)*	*0.113 (5.4%)*

$Ra = 10^5$	U_{max}	Y_{max}	V_{max}	X_{max}
de Vahl Davis	34.730	0.145	68.590	0.066
vb (90 by 90)	34.831 (0.29%)	0.144 (0.38%)	68.622 (0.05%)	0.067 (0.97%)
vb (30 by 30)	*35.2 (1.3%)*	*0.143 (1.4%)*	*67.682 (1.3%)*	*0.057 (13.4%)*
cc (90 by 90)	35.284 (1.59%)	0.144 (0.38%)	68.315 (0.40%)	0.067 (0.97%)
cc (30 by 30)	*35.825 (3.1%)*	*0.143 (1.4%)*	*65.892 (3.93%)*	*0.057 (13.4%)*
vb-cc (90 by 90)	34.780 (0.14%)	0.144 (0.38%)	69.600 (1.47%)	0.067 (0.97%)
vb-cc (30 by 30)	*34.138 (1.7%)*	*0.143 (1.4%)*	*66.689 (2.7%)*	*0.057 (13.4%)*

$Ra = 10^6$	U_{max}	Y_{max}	V_{max}	X_{max}
de Vahl Davis	64.63	0.150	219.36	0.0379
vb (90 by 90)	62.410 (3.4%)	0.144 (3.71%)	222.78 (1.56%)	0.0333 (12.1%)
vb (30 by 30)	61.020 (5.6%)	0.143 (4.7%)	212.47 (3.1%)	0.0281 (24.6%)
cc (90 by 90)	64.559 (0.11%)	0.144 (3.7%)	217.38 (0.90%)	0.0333 (12.1%)
cc (30 by 30)	65.264 (0.98%)	0.143 (4.7%)	198.18 (9.6 %)	0.0281 (24.6%)
vb-cc (90 by 90)	63.867 (1.18%)	0.144 (3.71%)	219.85 (0.22%)	0.0333 (12.1%)
vb-cc (30 by 30)	63.442 (1.84 %)	0.143 (4.7%)	220.69 (0.61%)	0.0281 (24.6%)

Table 6.1: Comparison of simulation results against benchmark solutions

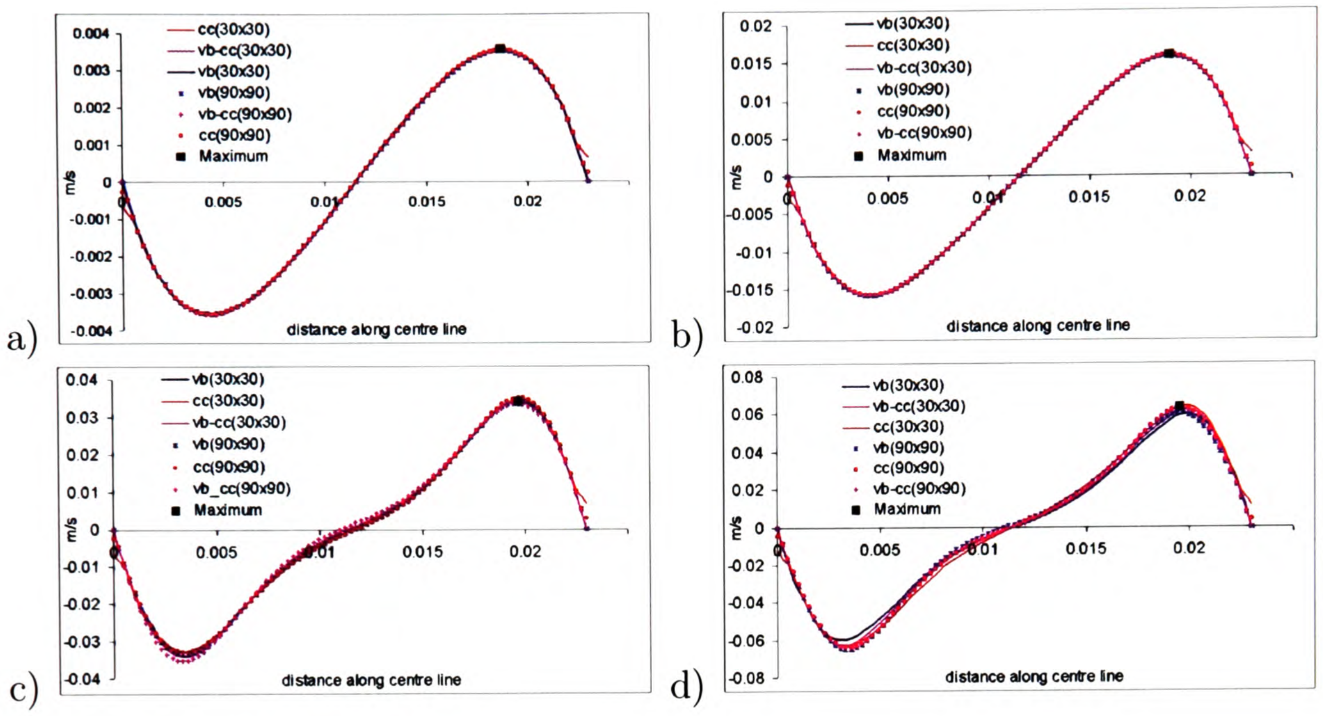


Figure 6.3: U-velocity profiles: a)Ra=10^3, b)Ra=10^4, c) Ra=10^5, d) Ra=10^6

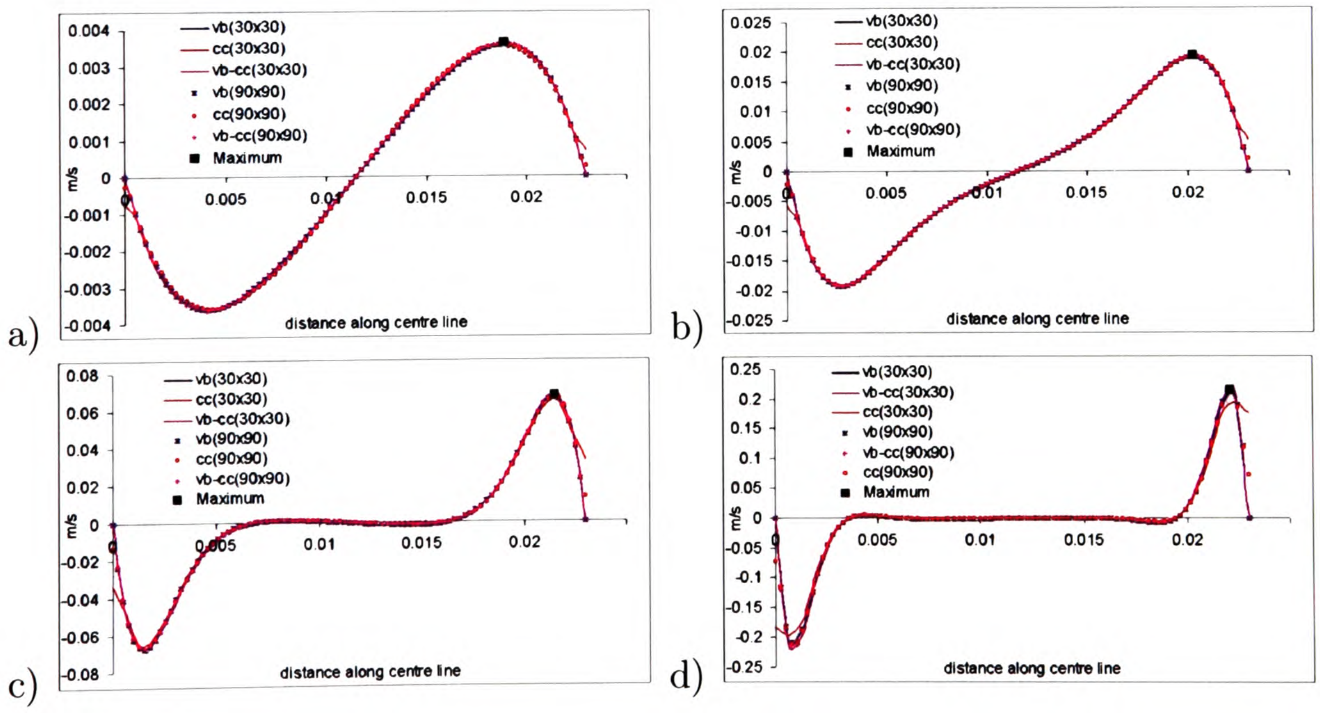


Figure 6.4: V-velocity profiles: a)Ra=10^3, b)Ra=10^4, c) Ra=10^5, d) Ra=10^6

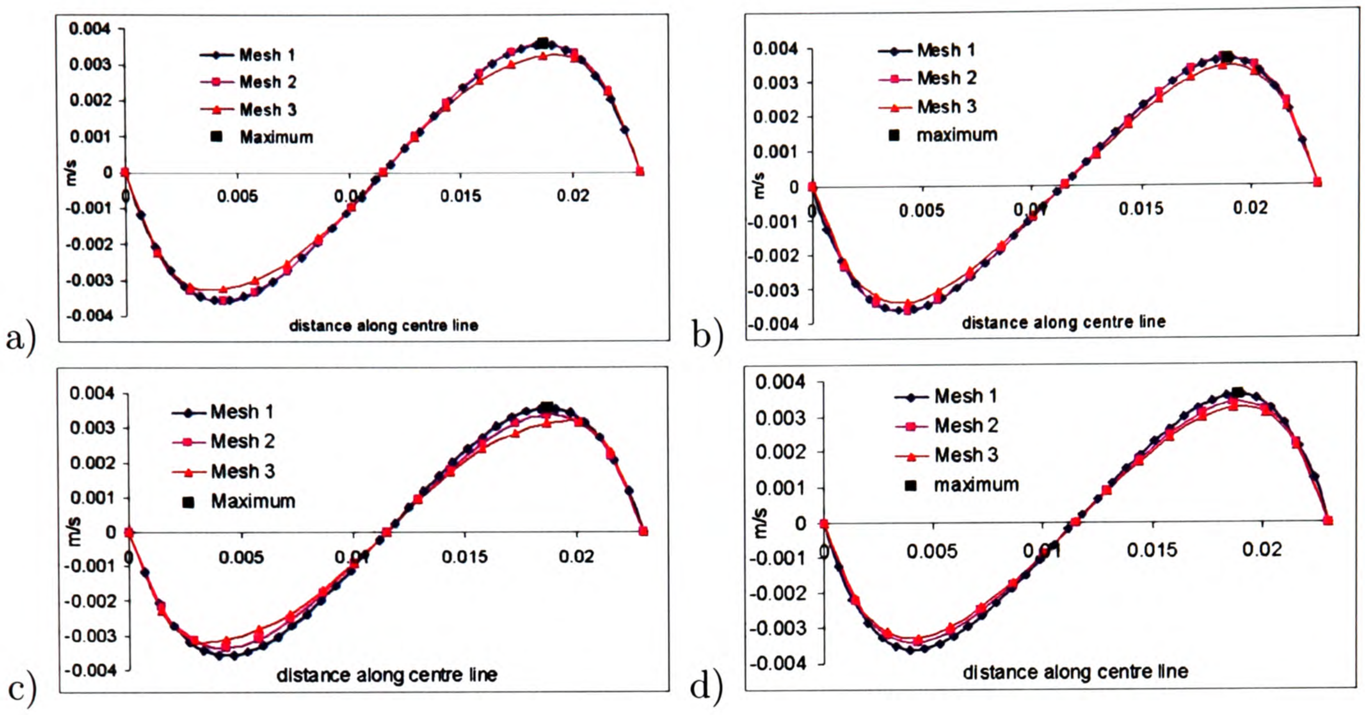


Figure 6.5: Ra=10^3 : a) u-velocity (vb) , b) v-velocity (vb), c) u-velocity (vb-cc), d) v-velocity (vb-cc)

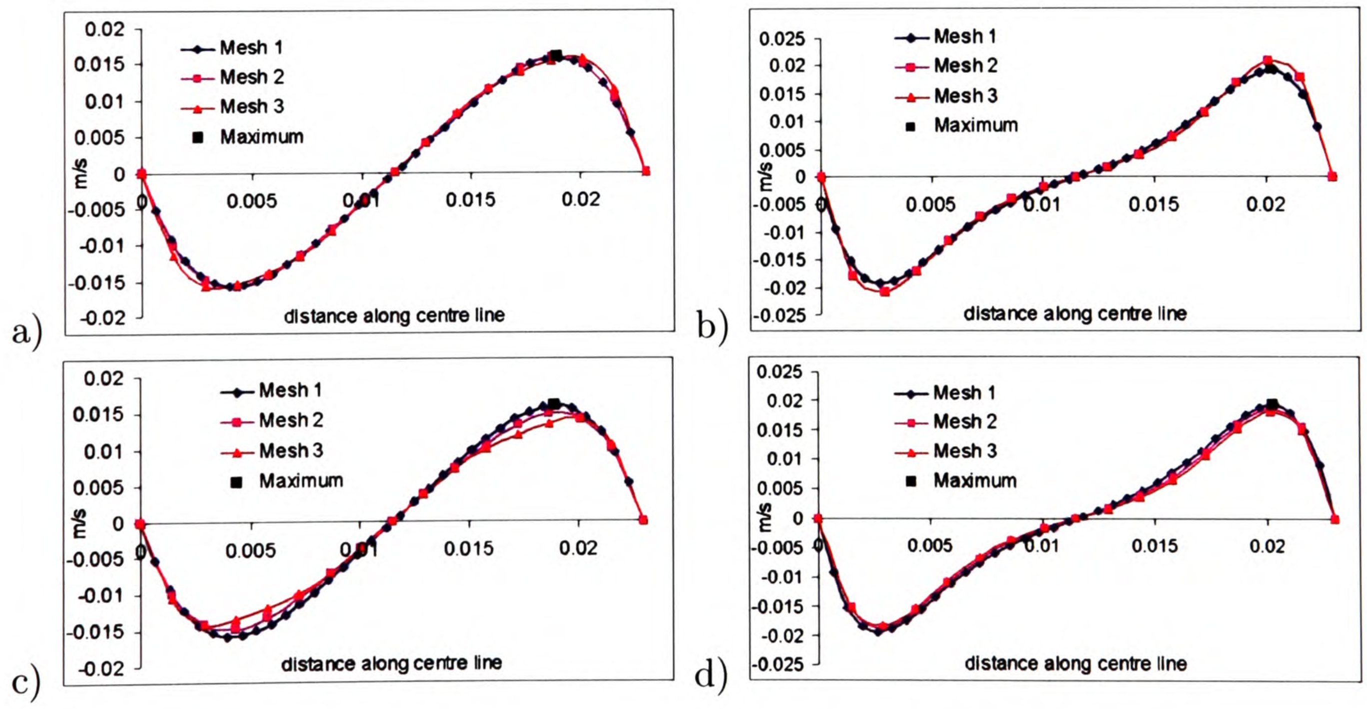


Figure 6.6: Ra=10^4: a) u-velocity (vb), b) v-velocity (vb), c) u-velocity (vb-cc), d) v-velocity (vb-cc)

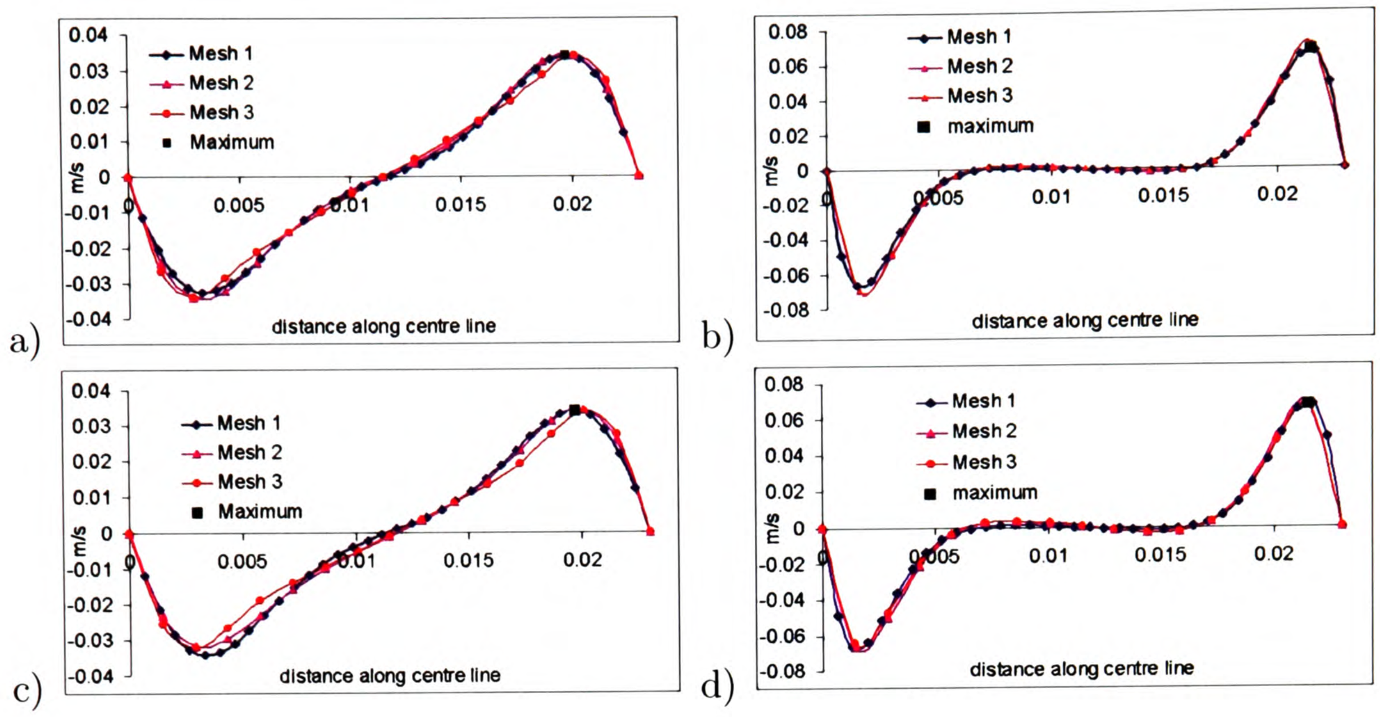


Figure 6.7: Ra=10^5 : a) u-velocity (vb) , b) v-velocity (vb), c) u-velocity (vb-cc), d) v-velocity (vb-cc)

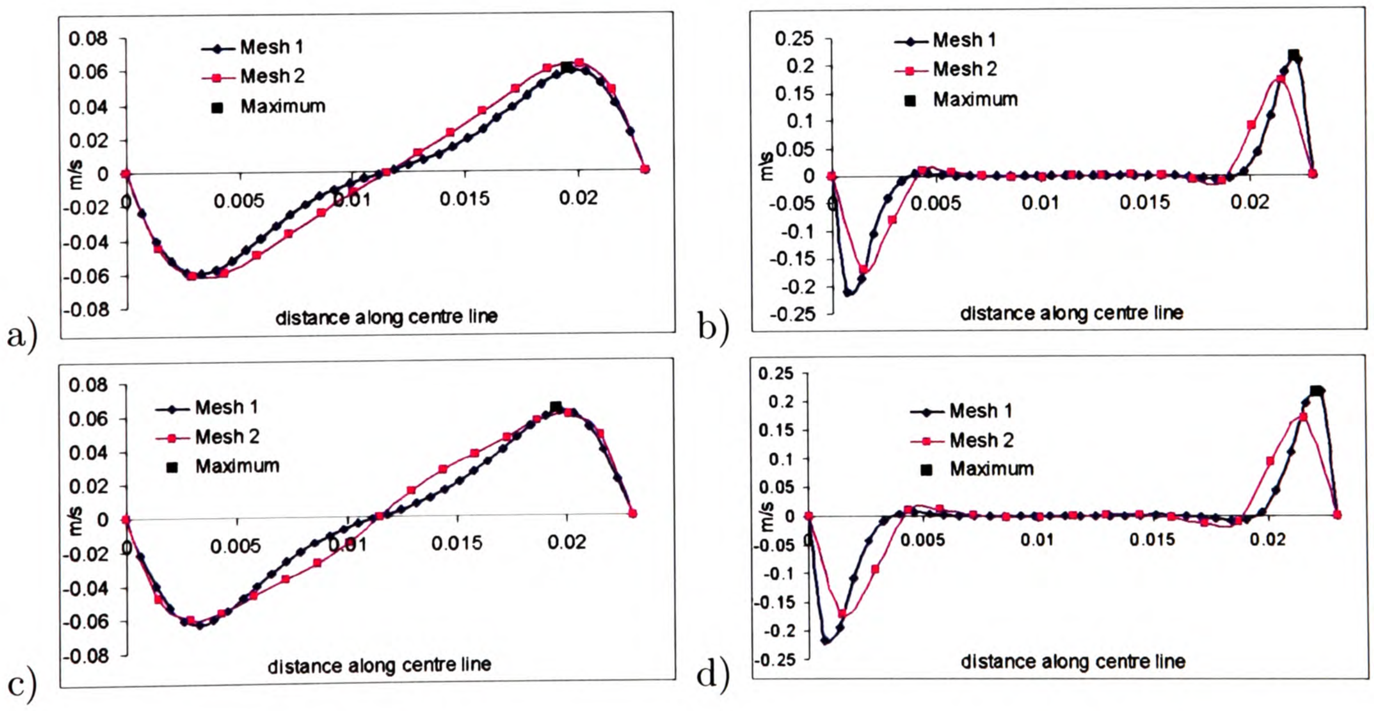


Figure 6.8: Ra=10^6: a) u-velocity (vb), b) v-velocity (vb), c) u-velocity (vb-cc), d) v-velocity (vb-cc)

in the solution of flow fell below 10^{-5}. Problems with convergence were encountered for the higher Rayleigh numbers $Ra = 10^5$ for the $vb - cc$ method and $Ra = 10^6$ for both methods, in these cases the convergence criteria was reduced to 10^{-4} A measure of the error due to mesh distortion was calculated for mesh 2 and mesh 3, using mesh 1 as the base result. The normalised error was calculated as,

$$error = \frac{\sum_n |residual|}{nU_{max}} \tag{6.7}$$

where U_{max} was taken from benchmark solutions and n is the number of solution points.

Simulation times on a AMD Athlon 1.39 Ghz processor, normalised error and any relaxation factors used (0.7 is the standard relaxation factor for velocity for vertex-based solutions) are given in Table 6.2 for purely vertex-based solutions and Table 6.3 for combined vertex-based/cell-centred solutions. For cases where the number of iterations required were similar, the $vb - cc$ simulation times were slightly faster. For the majority of the cases the vb method required less iterations to achieve convergence and consequently achieved lower simulation times. For all cases the maximum normalised error was in the order of 10^{-2} except for $Ra = 10^6$ where the $vb - cc$ solution obtained a maximum error of 0.108.

As discussed in Chapter 5, problems are encountered using purely cell-centred techniques on highly non-orthogonal meshes. On arbitrary distorted meshes divergence is often encountered when non-orthogonal correction terms are included in the discretisation process. On the meshes investigated, cc solutions could only be obtained without the inclusion of non-orthogonal corrections. This leads to significant errors in the results. No solutions could be obtained on mesh 3 for $Ra = 10^4$, $Ra = 10^5$ and $Ra = 10^6$. For illustration purposes, the cc results are shown for the u-velocity in Figure 6.9 and the v-velocity in Figure 6.10. The normalised error is given in Table 6.4.

Rayleigh Number 10^3	Mesh 1	Mesh 2	Mesh 3
Iterations	125	198	220
Time (secs)	24	43	50
Error (u-velocity)		4.74×10^{-03}	3.68×10^{-02}
Error (v-velocity)		4.71×10^{-03}	3.32×10^{-02}
Velocity Relaxation	0.7	0.7	0.7
Heat Relaxation			

Rayleigh Number 10^4	Mesh 1	Mesh 2	Mesh 3
Iterations	103	167	167
Time (secs)	20	38	38
Error (u-velocity)		7.82×10^{-03}	3.24×10^{-02}
Error (v-velocity)		8.39×10^{-03}	3.72×10^{-02}
Velocity Relaxation	0.7	0.7	0.7
Heat Relaxation			

Rayleigh Number 10^5	Mesh 1	Mesh 2	Mesh 3
Iterations	111	129	126
Time (secs)	22	29	28
Error (u-velocity)		1.94×10^{-02}	4.78×10^{-02}
Error (v-velocity)		1.58×10^{-02}	1.28×10^{-02}
Velocity Relaxation	0.7	0.7	0.6
Heat Relaxation			0.8

Rayleigh Number 10^6	Mesh 1	Mesh 2	Mesh 3
Iterations	383	362	Failed
Time (secs)	70	77	
Error (u-velocity)		9.28×10^{-02}	
Error (v-velocity)		3.94×10^{-02}	
Velocity Relaxation	0.6	0.6	
Heat Relaxation		0.8	

Table 6.2: Vertex-based solutions(vb)

Rayleigh Number 10^3	Mesh 1	Mesh 2	Mesh 3
Iterations	122	190	213
Time (secs)	18	32	35
Error (u-velocity)		3.22×10^{-02}	6.19×10^{-02}
Error (v-velocity)		3.25×10^{-02}	5.43×10^{-02}
Velocity Relaxation	0.7	0.7	0.7
Heat Relaxation			

Rayleigh Number 10^4	Mesh 1	Mesh 2	Mesh 3
Iterations	211	246	250
Time (secs)	35	52	51
Error (u-velocity)		3.42×10^{-02}	6.19×10^{-02}
Error (v-velocity)		2.62×10^{-02}	4.30×10^{-02}
Velocity Relaxation	0.7	0.7	0.7
Heat Relaxation			

Rayleigh Number 10^5	Mesh 1	Mesh 2	Mesh 3
Iterations	187	206	207
Time (secs)	31	42	42
Error (u-velocity)		4.04×10^{-02}	7.07×10^{-02}
Error (v-velocity)		2.11×10^{-02}	1.78×10^{-02}
Velocity Relaxation	0.6	0.6	0.6
Heat Relaxation			0.8

Rayleigh Number 10^6	Mesh 1	Mesh 2	Mesh 3
Iterations	301	492	Failed
Time (secs)	48	93	
Error (u-velocity)		1.08×10^{-01}	
Error (v-velocity)		6.02×10^{-02}	
Velocity Relaxation	0.6	0.6	
Heat Relaxation		0.8	

Table 6.3: Combined vertex-based/cell-centred solutions ($vb - cc$)

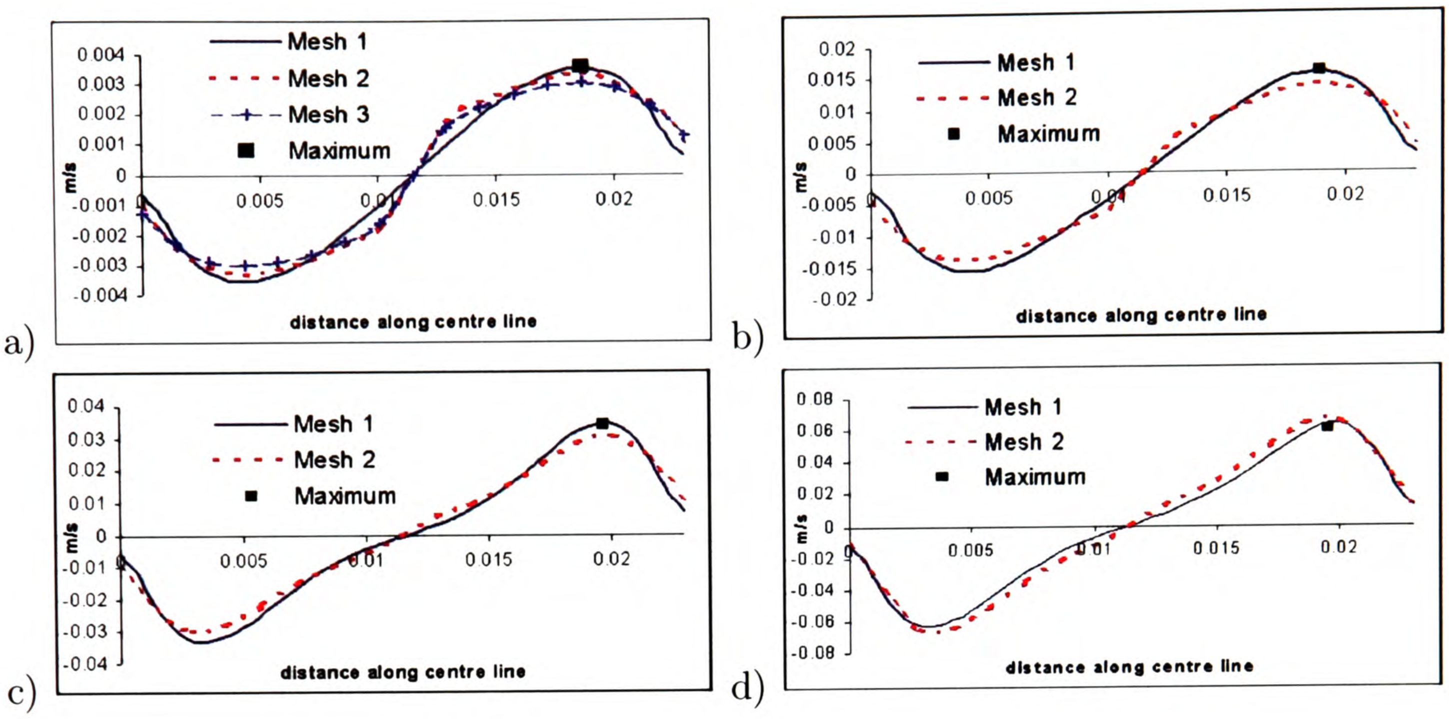


Figure 6.9: u-velocity (cc) : a) $Ra = 10^3$, b) $Ra = 10^4$, c) $Ra = 10^5$, d) $Ra = 10^6$

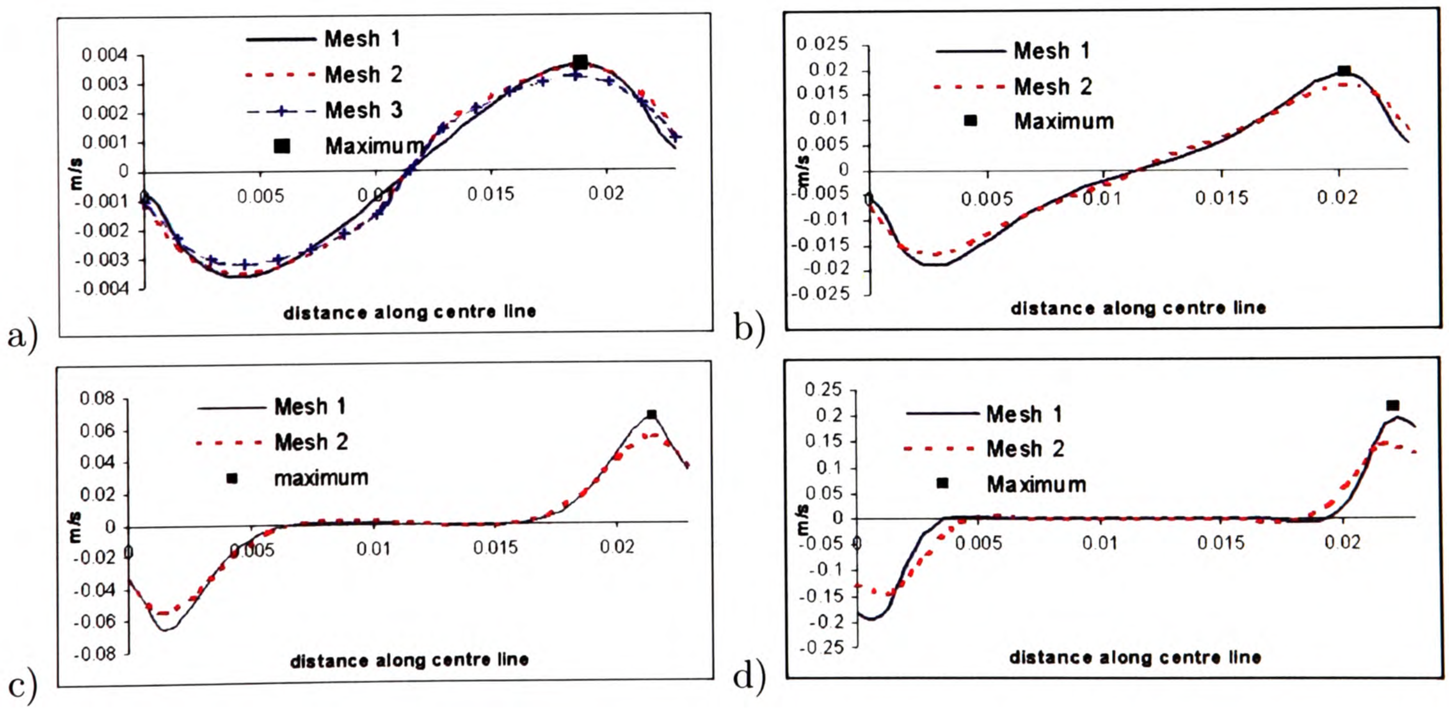


Figure 6.10: u-velocity (cc) : a) $Ra = 10^3$, b) $Ra = 10^4$, c) $Ra = 10^5$, d) $Ra = 10^6$

Element mass conservation was investigated for $vb-cc$ solutions, where mass conservation is enforced over the vertex-based control volume and element-based mass conserving velocity components are extrapolated from vertex-based values. Table 6.5 shows the L_2 norm of the mass residual for the $vb-cc$ vertex-based and element-based control volumes and for purely cell-centred solutions the element control volume. On the uniform mesh 1, the mass residual for the extrapolated element-based control volume is approximately 3 to 4 orders of magnitude higher than the vertex-based mass error. For the distorted meshes 1 and 2, the element-based mass error increases by approximately one to two orders of magnitude. The $vb-cc$ element-based mass error is of a similar magnitude to purely cell-centred element mass error and reduces as conservation is enforced over the vertex-based control volume.

Temperature contour plots obtained using mesh 3 are shown in Figures 6.12 to 6.17 for Rayleigh number of 10^3, 10^4 and 10^5. Solving temperature vertex-based gave smooth temperature plots on all mesh types. Using cell-centred techniques for temperature gave solutions that were slightly dependant on the skewness of the mesh, as illustrated in Figure 6.11 for a Rayleigh number of 10^3.

6.2 Closure

The vertex-based flow solution method has been coupled with cell-centred discretisation for a transported scalar. The method has been applied to a buoyancy-driven flow problem. Results have been compared with benchmark solutions for a number of Rayleigh numbers. Simulations were carried out on a uniform mesh and distorted versions of the same mesh. Good agreement with benchmark solutions were obtained on the uniform mesh and the solutions were only slightly degraded on the distorted mesh. At a Rayleigh number of 10^6 on the highly distorted mesh, divergence was encountered for both the $vb-cc$ and purely vertex-based method. A measure of the error was calculated and it was found that $vb-cc$ solutions compared

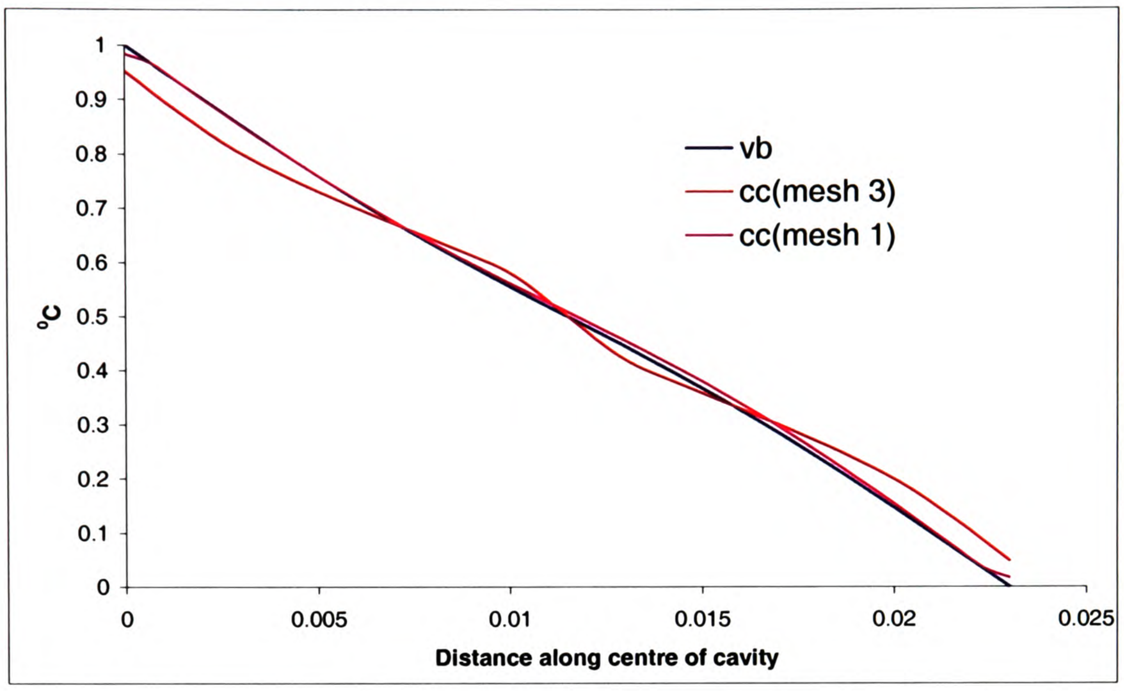


Figure 6.11: Ra=10^3, Temperature along line passing through centre of cavity

well with purely vertex-based results. Element mass conservation was investigated for the $vb-cc$ method and found to compare well with the level of mass conservation achieved when solving flow cell-centred.

Ra	u-velocity		v-velocity	
	Mesh 2	Mesh 3	Mesh 2	Mesh 3
10^3	1.02×10^{-1}	1.23×10^{-1}	8.74×10^{-2}	9.70×10^{-2}
10^4	9.66×10^{-2}	Failed	9.74×10^{-1}	Failed
10^5	9.74×10^{-1}	Failed	5.18×10^{-1}	Failed
10^6	1.34×10^{-1}	Failed	2.98×10^{-1}	Failed

Table 6.4: Purely *cc* solutions: Normalised error

	vb-cc	vb-cc	cc only
Control Volume	Vertex-Based	Element-Based	Element
Ra=10^3, Mesh 1	1.14×10^{-11}	9.44×10^{-08}	2.93×10^{-08}
Ra=10^3, Mesh 2	7.38×10^{-11}	2.05×10^{-07}	1.25×10^{-07}
Ra=10^3, Mesh 3	2.51×10^{-11}	8.89×10^{-05}	-
Ra=10^4, Mesh 1	3.01×10^{-11}	5.91×10^{-07}	9.85×10^{-09}
Ra=10^4, Mesh 2	5.24×10^{-11}	1.63×10^{-06}	2.97×10^{-08}
Ra=10^4, Mesh 3	6.51×10^{-11}	4.19×10^{-05}	-
Ra=10^5, Mesh 1	9.64×10^{-09}	4.95×10^{-06}	4.35×10^{-06}
Ra=10^5, Mesh 2	6.03×10^{-09}	2.36×10^{-05}	2.78×10^{-05}
Ra=10^5, Mesh 3	6.04×10^{-09}	2.37×10^{-05}	-
Ra=10^6, Mesh 1	2.52×10^{-08}	4.16×10^{-05}	3.51×10^{-05}
Ra=10^6, Mesh 2	2.40×10^{-08}	1.94×10^{-04}	4.89×10^{-05}

Table 6.5: Mass residual for $vb - cc$ and purely *cc* solutions

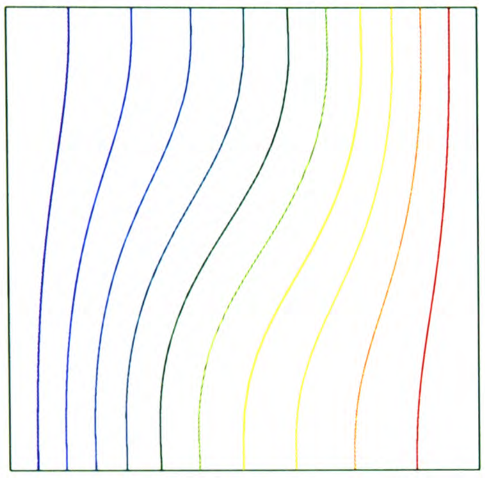


Figure 6.12: Ra=10^3, temperature (vb)

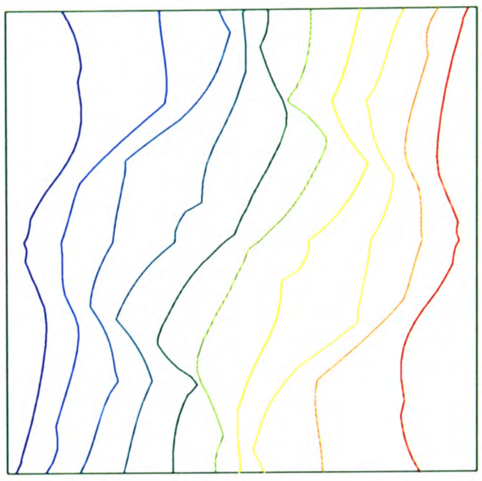


Figure 6.15: Ra=10^3, temperature (cc)

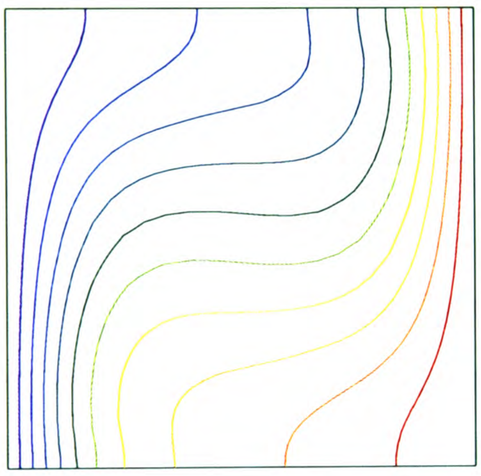


Figure 6.13: Ra=10^4, temperature (vb)

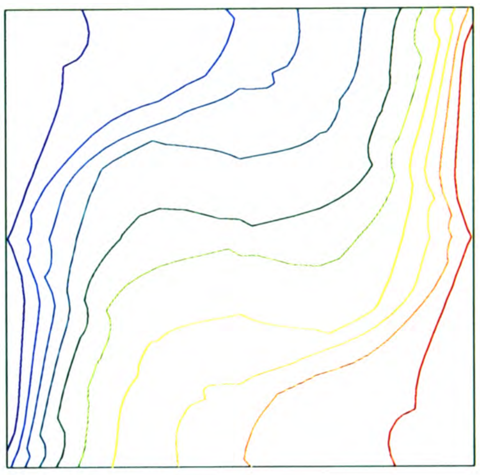


Figure 6.16: Ra=10^4, temperature (cc)

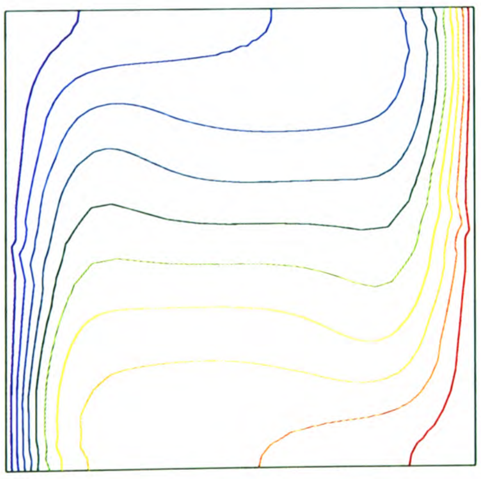


Figure 6.14: Ra=10^5, temperature (vb)

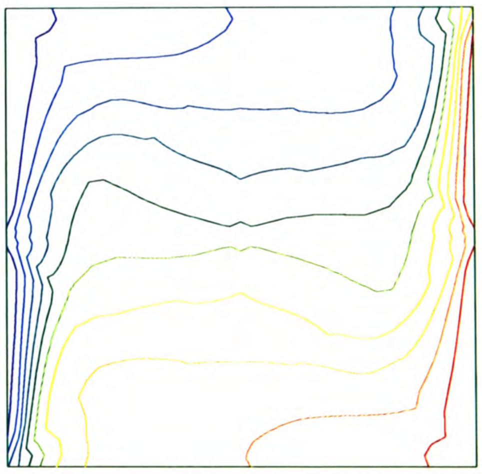


Figure 6.17: Ra=10^5, temperature (cc)

Chapter 7

Extending Vertex-Based Methods to Turbulent Flow

In the previous chapter vertex-based solution methods were applied to laminar flow. The test case presented in chapter 5 illustrated the ability of the co-located vertex-based discretisation method to resolve the velocity field on distorted meshes. This chapter investigates utilising the co-located vertex-based flow formulation in a combined vertex-based/cell-centred solution method to allow the simulation of turbulent flow problems. The $k - \epsilon$ turbulence model is employed, the transport equations for k and ϵ being discretised according to cell-centred techniques. The hydrodynamic variables, velocity and pressure, are solved using vertex-based techniques. Firstly, the combined vertex-based/cell-centred method is applied to flow over a backward facing step and results compared with solutions obtained solving all variables cell-centred. The second test case involves three-dimensional turbulent flow over an aircraft wing. The simulations are run on a uniform Cartesian mesh and distorted versions of the mesh. A comparison of the results, using combined vertex-based/cell-centred and purely cell-centred solutions is undertaken for uniform and distorted meshes.

Turbulence is an irregular motion, rotational, three-dimensional and highly dissipative in nature. It occurs in flows at high Reynolds numbers. Even in flows where the mean velocity and pressure vary in only one or two dimensions, turbulent fluctuations are always three-dimensional. These rotational flow structures, termed turbulent eddies, have a large range of length and time scales. Diffusivity is enhanced, maybe several orders of magnitude greater than for laminar flow, thus greatly increasing the transfer of mass, momentum and energy. The largest eddies interact with the mean flow extracting energy by a process called vortex stretching. Smaller eddies are stretched strongly by the larger eddies and only weakly by the mean flow. In this way energy cascades down from the largest to the smallest eddies where energy is dissipated by viscous forces. This implies that the larger-scale turbulent eddies are most responsible for the energy transfer and enhanced diffusivity.

Prediction of turbulence is one of the fundamental problems of computational fluid dynamics. Turbulence occurs at high Reynolds numbers and is fully three-dimensional and time-dependent. In contrast to laminar flow, turbulence develops as an instability which is irregular and intermittent. Because of the random and chaotic nature of the turbulence phenomena it is customary to work with the time-averaged form of the governing equations. The most popular method being Reynolds averaging which provides information about the overall mean flow properties. Time averaging of the governing equations gives rise to additional turbulent forces that approximate the turbulent random fluctuations. Turbulence models provide expressions to account for these additional terms in order to provide closure to the governing equations. They do not simulate the detail of the turbulent motion, only the effect of turbulence on the mean flow behaviour. Turbulence models are based on hypotheses about the turbulent process and require empirical input in the form of model constants and functions.

The governing equations for flow, given in the previous chapter, are the continuity equation,

$$\frac{\partial \rho}{\partial t} + div(\rho \underline{U}) = S_m \tag{7.1}$$

and the momentum conservation equation:

$$\frac{\partial(\rho U_i)}{\partial t} + div(\rho \underline{U} U_i) = \frac{\partial P}{\partial x_i} + div(\mu grad U_i) + S_i \tag{7.2}$$

In the above equations, the quantities U_i and P represent the instantaneous velocity component in the x_i direction and the instantaneous pressure. The main idea behind Reynolds time-averaging is to express U_i and P in terms of a time averaged component and a fluctuating component.

$$U_i = \bar{u}_i + v_i \qquad P = \bar{p} + \pi \tag{7.3}$$

Substituting (7.3) into equations (7.1) and (7.2) and dropping the overlines signifying the time averaged values, we obtain the governing equations for the mean flow quantities,

$$\frac{\partial \rho}{\partial t} + div(\rho \underline{u}) = S_m \tag{7.4}$$

and:

$$\frac{\partial(\rho u_i)}{\partial t} + div(\rho \underline{u} u_i) = \frac{dp}{dx_i} + div(\mu grad u_i) - \frac{\partial}{\partial x_j}(\rho \overline{v_i v_j}) + S_i \tag{7.5}$$

The non-linearity of equation (7.2) produces an extra turbulent term in equation (7.5), which represents the transport of momentum due to the turbulent motion. This term $-\rho\overline{v_i v_j}$ represents the transport of x_i momentum in the x_j direction and is known as the Reynolds stress. For a fully turbulent flow the stress acting on the fluid is much greater than the laminar diffusion term. Classical turbulence models provide expressions to account for the Reynolds stress term in order to provide closure to equations (7.4) and (7.5). Experimentally it is observed that the turbulent stresses are found to increase as the mean velocity gradients increase, and it was proposed by Boussinesq [140] in 1877 that the Reynolds stresses are proportional to the mean velocity gradients,

$$-\rho\overline{v_i v_j} = \mu_t(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \tag{7.6}$$

where μ_t is the turbulent or eddy viscosity. There is also a kinematic turbulent or eddy viscosity ν_t with dimensions m^2s^{-1}, where $\nu_t = \mu_t/\rho$.

Turbulent stresses vary from point to point in the flow and thus the turbulent viscosity μ_t is a local value dependent on turbulence. The turbulence problem can now be reduced to a problem of determining the local value of μ_t.

Given below are some of the methods used in turbulence modelling:

- Algebraic(zero-equation) models

  Prandtl [141] introduced the concept of the mixing-length model which prescribes an algebraic relation for the turbulent stresses by means of a formulae for μ_t as a function of its position. Van Driest [142] devised a viscous damping correction for the mixing-length model. Cebeci and Smith [143] refined the mixing-length concept for better use with attached boundary layers. Baldwin and Lomax [144] proposed an alternative model to eliminate some of the difficulty in defining a turbulence length scale from the shear-layer thickness.

- One-Equation Model

  Prandtl [145] introduced the first one-equation model in 1945. He proposed that the eddy viscosity depends on the turbulent kinetic energy (k), providing a differential equation to approximate the exact equation for k. The effects of flow history are taken into account thus improving turbulence predictions. Although less complex than the two-equation models, one-equation models have not proved popular.

- Two-Equation Model

  Two-Equation Models provide a more complete description of turbulence. The turbulence length scale changes for different flows, two-equation models represent a given turbulent flow by prescribing boundary and initial conditions. Kolmogorov [146] introduced the first complete turbulence model in 1942. In addition to the equation to model the turbulent kinetic energy (k), a second equation models the rate of dissipation of energy per unit volume and time (ω). Launder and Spalding [147] developed the most widely used two-equation model, the $k-\epsilon$ model. In this model ϵ models the dissipation rate of turbulent kinetic energy.

- Second-Order Closure Models

  Second-order closure models use seven equations, one for the turbulence length scale and six for the Reynolds stresses. Due to the complexity of this class of turbulence model they are not widely used in industry. There are two types of second-order closure models, Algebraic Stress Models (ASM) and Reynolds Stress Models (RSM). RSM solve the exact differential equation describing the Reynolds stress tensor. ASM model the Reynolds stresses by a series expansion of functionals. The model constants have been determined by many investigators, including Launder *et al*[148] and Gibson and Launder [149].

- Large Eddy Simulation

  The Navier-Stokes equations are solved for large scale eddies and the influence of very small scale eddies, the sub-grid scale, are approximated by a model. Sub-grid scale models include the Smagorinsky [150], $k-\epsilon$, and dynamic Smagorinsky [151]. This approach requires a very fine mesh and a very small time step.

- Direct Numerical Simulation

  The Navier-Stokes equations are discretised and solved numerically. This approach solves all turbulent eddy sizes, produces results of high accuracy but is computationally very expensive.

The $k - \epsilon$ two-equation turbulence model is one of the most popular and generally used in the simulation of turbulent flow problems. In this investigation the $k - \epsilon$ model is incorporated into the vertex-based flow code. The pressure and velocity components are solved using vertex-based techniques and the transport equations for the turbulent quantities k and ϵ are discretised using cell-centred techniques. A detailed description of the $k - \epsilon$ model is given in the following section.

7.1 $k - \epsilon$ Model

Exact transport equations for the turbulent kinetic energy (k) and the rate of viscous dissipation (ϵ) can be derived from the Navier-Stokes equations, however they contain many unknown and unmeasurable terms. The standard $k - \epsilon$ model is based upon certain assumptions and can be applied to a wide range of flows provided its fundamental assumptions are not grossly violated. The main assumption is one of local equilibrium where turbulent production and dissipation balance. This implies that the scales of turbulence are locally proportional to the scales of the mean flow.

Mixing length models assume, on dimensional grounds, that the kinematic turbulent viscosity ν_t can be expressed as the product of a turbulent velocity scale ϑ and a length scale ℓ,

$$\nu_t = C\vartheta\ell \tag{7.7}$$

where C is a dimensionless constant of proportionality and the dynamic turbulent viscosity is $\mu_t = C\rho\vartheta\ell$.

Using the same approach as in the mixing length models, a velocity scale ϑ and length scale ℓ is defined using k and ϵ as,

$$\vartheta = k^{1/2} \text{ and } \ell = \frac{k^{3/2}}{\epsilon} \tag{7.8}$$

and the eddy viscosity is defined as follows,

$$\mu_t = C_\mu \rho\vartheta\ell = \rho C_\mu \frac{k^2}{\epsilon} \tag{7.9}$$

where C_μ is a dimensionless constant.

The $k - \epsilon$ model uses the following transport equations for the solution of the k and ϵ.

The turbulent kinetic energy equation,

$$\frac{\partial(\rho k)}{\partial t} + div\left(\rho \underline{u} k\right) = div\left(\left[\mu_{lam} + \frac{\rho\nu_t}{\sigma_k}\right] grad(k)\right) + \rho\nu_t G - \rho\epsilon \tag{7.10}$$

and the dissipation rate equation:

$$\frac{\partial(\rho\epsilon)}{\partial t} + div\left(\rho \underline{u}\epsilon\right) = div\left(\left[\mu_{lam} + \frac{\rho\nu_t}{\sigma_\epsilon}\right] grad(\epsilon)\right) + C_{1\epsilon}\rho\nu_t G\frac{\epsilon}{k} - C_{2\epsilon}\rho\frac{\epsilon^2}{k} \tag{7.11}$$

The above equations contain five constants $C_\mu, \sigma_k, \sigma_\epsilon, C_{1\epsilon}$ and $C_{2\epsilon}$. The standard $k - \epsilon$ model employs values for the constants that are arrived at by comprehensive data fitting for a wide range of turbulent flows, [152],

$$C_\mu = 0.09; \;\sigma_k = 1.0; \;\sigma_\epsilon = 1.3; \;C_{1\epsilon} = 1.44; \;C_{2\epsilon} = 1.92 \tag{7.12}$$

In the implementation of the $k - \epsilon$ model the above equations are discretised using cell-centred techniques, as outlined in chapter 2. The momentum and continuity equations are discretised using either cell-centred or vertex-based techniques. The continuity equation remains unchanged whilst the laminar viscosity $\rho\nu_{lam}$ is replaced by $\rho\nu_{lam} + \rho\nu_t$ in the momentum equations.

In equations (7.10) and (7.11) the source terms contain the turbulent generation rate G, which is equal to:

$$\begin{aligned} G &= 2\left(\left[\frac{\partial u}{\partial x}\right]^2 + \left[\frac{\partial v}{\partial y}\right]^2 + \left[\frac{\partial w}{\partial z}\right]^2\right) \\ &+ \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right)^2 \end{aligned} \tag{7.13}$$

Since the turbulent kinetic energy k and the dissipation rate ϵ are being discretised over an element control volume, the turbulent generation rate G is element based. The calculation of G requires the values of the derivatives of the three Cartesian velocity components with respect to the three Cartesian directions. The techniques used to calculate the velocity gradients over an element are shown below

Vertex-Based - velocity gradients

When the velocity components are located at the mesh vertices the velocity gradient over the element volume V is given by:

$$\int_V \frac{\partial u_i}{\partial x_j} \approx V \frac{\partial u_i}{\partial x_j}\bigg|_{element} \tag{7.14}$$

Assuming linear variation over an element the vertex-based element shape function derivatives can be employed to estimate the velocity components with respect to the three Cartesian directions,

$$\frac{\partial u_i}{\partial x_j} = \sum_{k=1}^{n} \frac{\partial N_k}{\partial x_j}(u_i)_k \tag{7.15}$$

where n is the number of nodes associated with the element and the integration point is taken as the element centre, giving:

$$\int_V \frac{\partial u_i}{\partial x_j} \approx V \sum_{k=1}^{n} \frac{\partial N_k}{\partial x_j}(u_i)_k \tag{7.16}$$

Cell-Centred - velocity gradients

When the velocity components are solved at the element centre, the divergence theorem is employed, thus:

$$\int_V \frac{\partial u_i}{\partial x_j} = \sum_f \int_f u_i n_j dS \tag{7.17}$$

The above formula can be used to estimate the element derivative value as:

$$\frac{\partial u_i}{\partial x_j}\bigg| \approx \frac{1}{V} \sum_f A_f n_j (u_i)_f \tag{7.18}$$

This reduces the problem to one of estimating a representative value of u_i on each element face. Face values can then be extrapolated from element values.

The method used to estimate these values depends on whether the velocity components are being solved at the element centre (7.17) or element nodes (7.14). The vertex-based technique allows better definition of the velocity gradient over an element when the mesh is distorted. When velocity components are located at the element centre, element velocity gradients can contain errors due to mesh distortion. On distorted meshes, interpolating face values from neighbouring element values often results in error due to non-conjunctionality, Figure 2.3, of the adjacent elements.

Boundary Conditions

The $k-\epsilon$ model provides solutions for fully turbulent flow. Solutions are not required in the fluid layer next to a wall, as in this region viscous effects are more significant than turbulent ones. In this viscous sub-layer there is a linear relationship between the velocity at point p and the distance from the wall y:

$$u_p = \frac{\tau_w}{\mu} \tag{7.19}$$

In the log-law region just outside the viscous sub-layer, where,

$$30 < y^+ < 500 \tag{7.20}$$

measurements have shown that the generation of kinetic energy is balanced by its dissipation. Using these assumptions and the formula (7.7) for the eddy viscosity, the kinetic energy of turbulence is given as,

$$k = \frac{v_\tau^2}{\sqrt{C_\mu}} \tag{7.21}$$

and its dissipation as,

$$\epsilon = \frac{v_\tau^3}{Ky} \tag{7.22}$$

where K is Von Karman's constant, $K = 0.41$.

7.2 Backward Facing Step

The $k - \epsilon$ model is employed in the solution of turbulent flow over a backward facing step. The combined vertex-based/cell-centred ($vb - cc$) technique solves the flow variables using vertex-based procedures. Velocity components are extrapolated to the element faces thus allowing the turbulent quantities k and ϵ to be solved at the element-centre using cell-centred discretisation techniques. The results are compared with solutions obtained using purely cell-centred (cc) techniques, where all variables are solved at the element centre.

In this case fluid enters the domain through a channel of height H and flows over a step of height h into a channel of height $H + h$. The case simulated here employs an outflow channel that is three times the height of the step, the step height is set to 0.1 m. The geometry and boundary conditions are shown in Figure 7.1, where $H = 2h$.

The material properties are set to those of air and the inflow velocity U_{in} is varied to give the required Reynolds number. A no-slip boundary condition $u = v = 0.0$ is applied to the wall. The outflow boundary location is positioned 20 step heights downstream of the step, to limit its influence on the simulation. Pressure is set to zero at the outlet. The inflow boundary is positioned five step heights upstream of the step to allow the formation of a fully turbulent velocity profile before the step

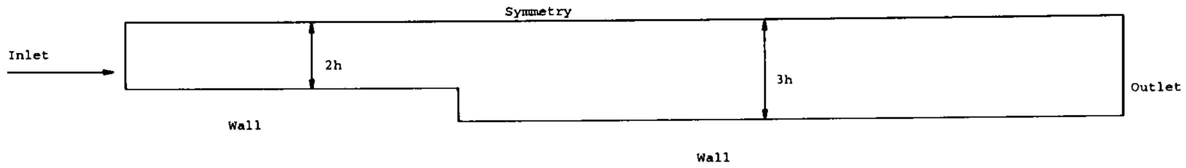


Figure 7.1: Backward facing step

is reached. The inflow turbulent quantities are estimated as, [2]:

$$\begin{aligned} K_{in} &= \frac{1}{4}0.018U_{in}^2 \\ \epsilon_{in} &= \frac{0.1643k_{in}^{1.5}}{0.09h} \end{aligned} \tag{7.23}$$

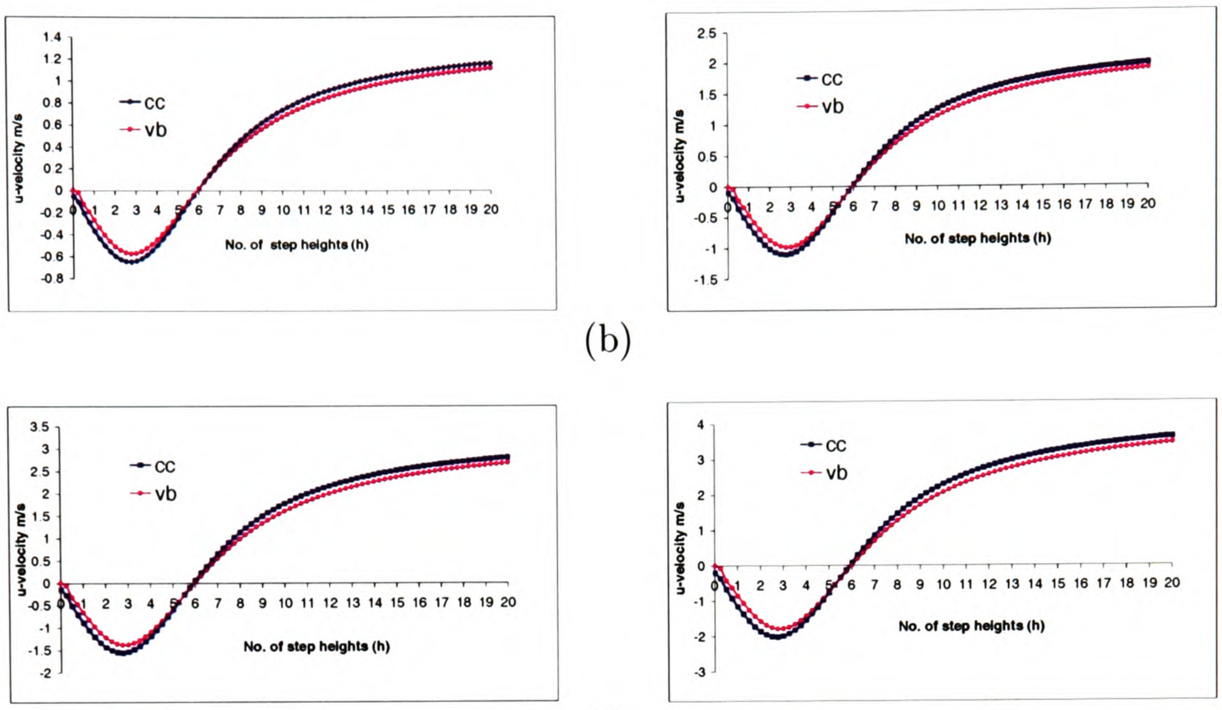
As the fluid flows over the step it detaches from the wall and reattaches on the bottom wall at a distance x_R from the step. It is this reattachment length x_R that is commonly used for validation purposes. The experimental data for this case [153, 154] indicates a reattachment length of approximately $(7.0 \pm 0.5)h$ with only slight variation for flows of differing Reynolds numbers. The standard $k - \epsilon$ model has been shown by many authors, including [155, 156], to under-predict the reattachment point, giving a value of x_R with in the range 5.8 to 6.1 step heights along the bottom wall.

The simulations were performed initially on a mesh consisting of 5200 elements. The u-velocity at the inlet was set to give Reynolds numbers of $30,000$, $50,000$, $70,000$ and $90,000$. Figure 7.2 shows the u-velocity values along the line of nodes adjacent to the wall. As can be seen from the plots, for all the Reynolds numbers investigated, the reattachment length, x_R, is at a distance of 6 step heights downstream. This value is well within the expected range for the standard $k - \epsilon$ model. There appears to be very little difference in the results of the $vb - cc$ and cc discretisation methods,

	cc	$vb - cc$
Elements	5200	5200
Vertices	10742	10742
Iterations	1492	899
No. of solution points	26000	42626
Time (sec), per solution point	0.0123	0.0304
Memory (bytes), per solution point	57.12	219.57

Table 7.1: Computational requirements - Backward facing step

Figure 7.4 shows the u-velocity contour plots, Figure 7.5 the v-velocity contour plots and Figure 7.6 the turbulent viscosity contour plots for a Reynolds number of 50000. Simulations were also performed on coarser meshes of 1400 and 3588 elements for a Reynolds number of 50,000. As can be seen from the plots of the wall u-velocity values, Figure 7.3, the $vb-cc$ method still gave good predictions of the reattachment point, $x_R = 5.66h$ on the 1400 element mesh and $x_R = 5.86h$ on the 3588 element mesh. The cc method under-predicted x_R on the coarser meshes, $x_R = 5.0h$ and $x_R = 5.24h$ for the 1400 and 3588 element meshes respectively. The computational times for both methods are given in Table 7.1. The simulations were carried out on a AMD Athlon 1.39 GHz processor. It should be noted that although this is a two-dimensional problem the simulations were performed on a hexahedral mesh and consequently the vertex-based flow solutions are resolved on two, two-dimensional planes. Therefore the run-times and memory requirements are given per solution point. The $vb-cc$ method required less iterations to achieve the convergence criteria of all residuals less than 1.0×10^{-4}. The $vb - cc$ simulation run time was four times greater than the cc times, but per solution point was only 2.5 times greater.



(a) (b)

(c) (d)

Figure 7.2: u-velocity values along wall for, (a) $Re = 30,000$ (b) $Re = 50,000$ (c) $Re = 70,000$ (d) $Re = 90,000$

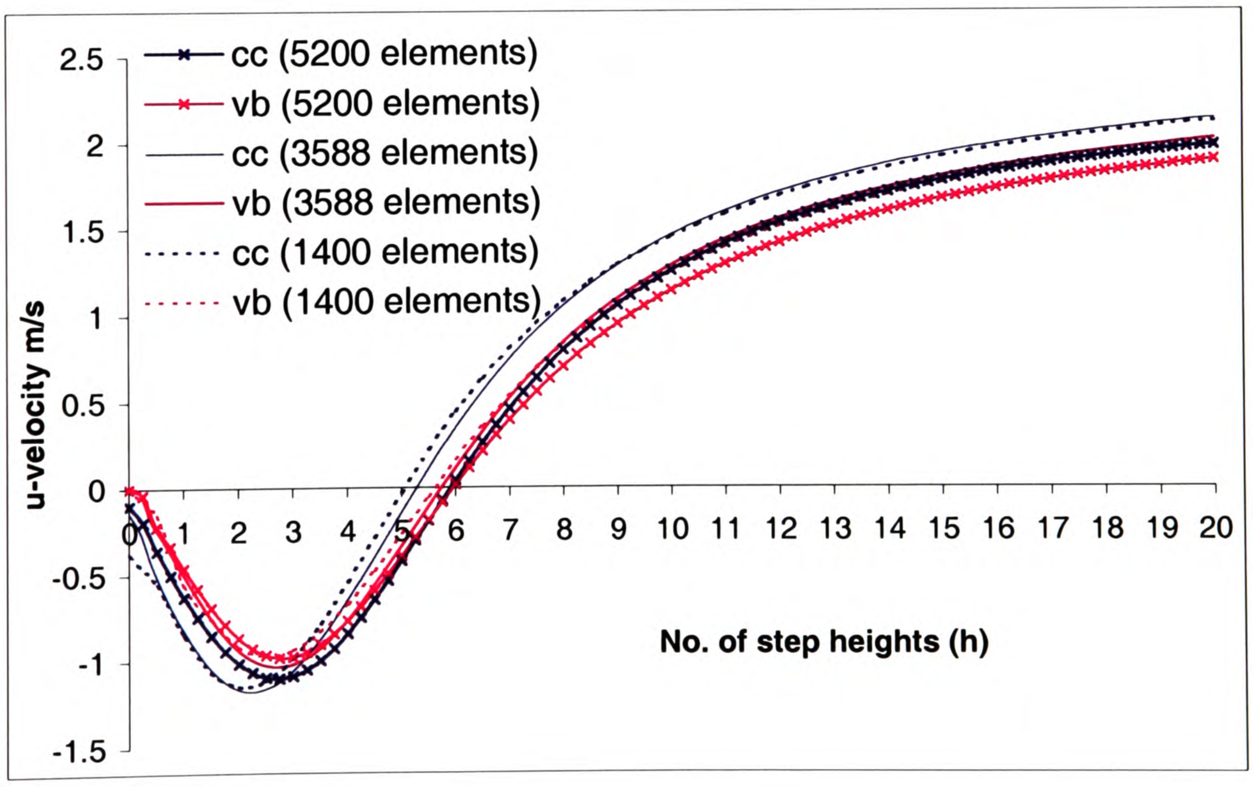


Figure 7.3: u-velocity values along wall for Reynolds number of 50,000

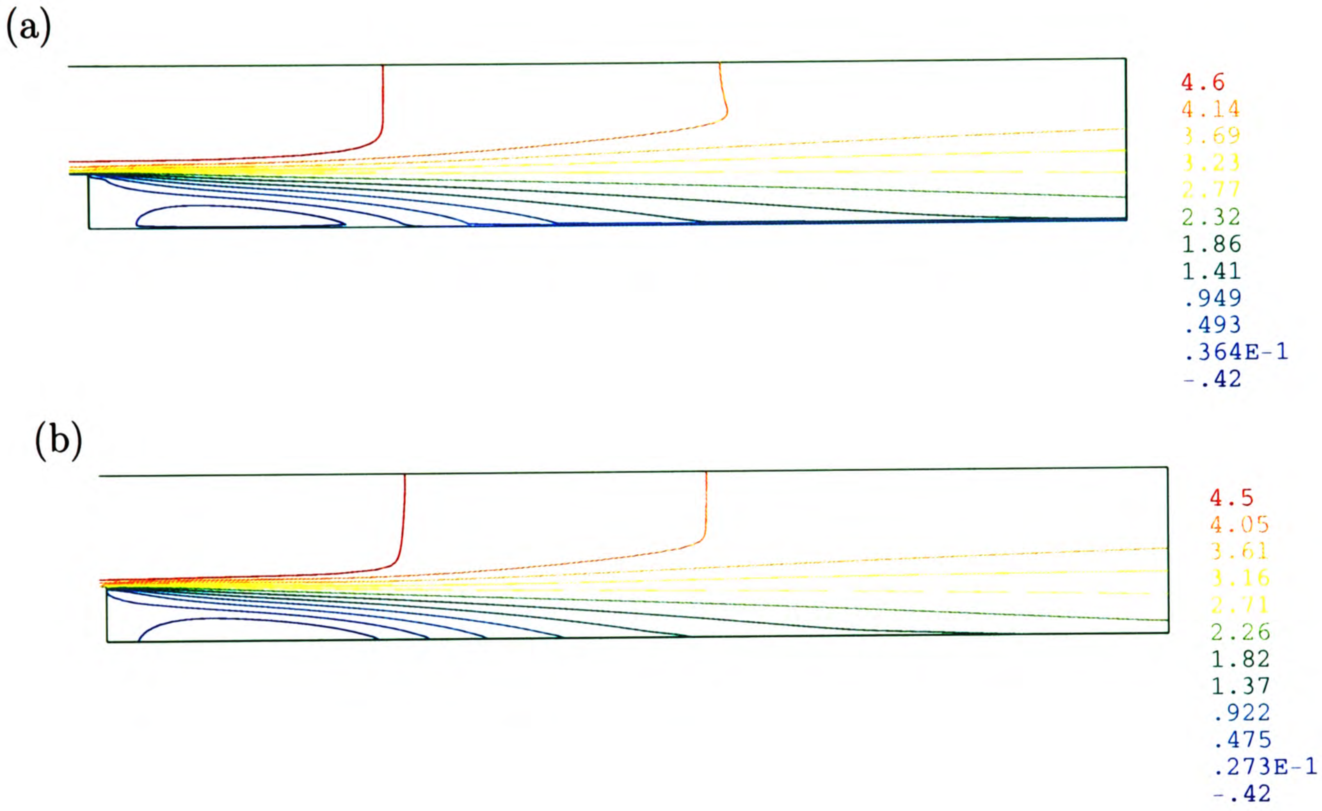


Figure 7.4: $Re = 50000$, u-velocity contours, (a) $vb - cc$, (b) cc

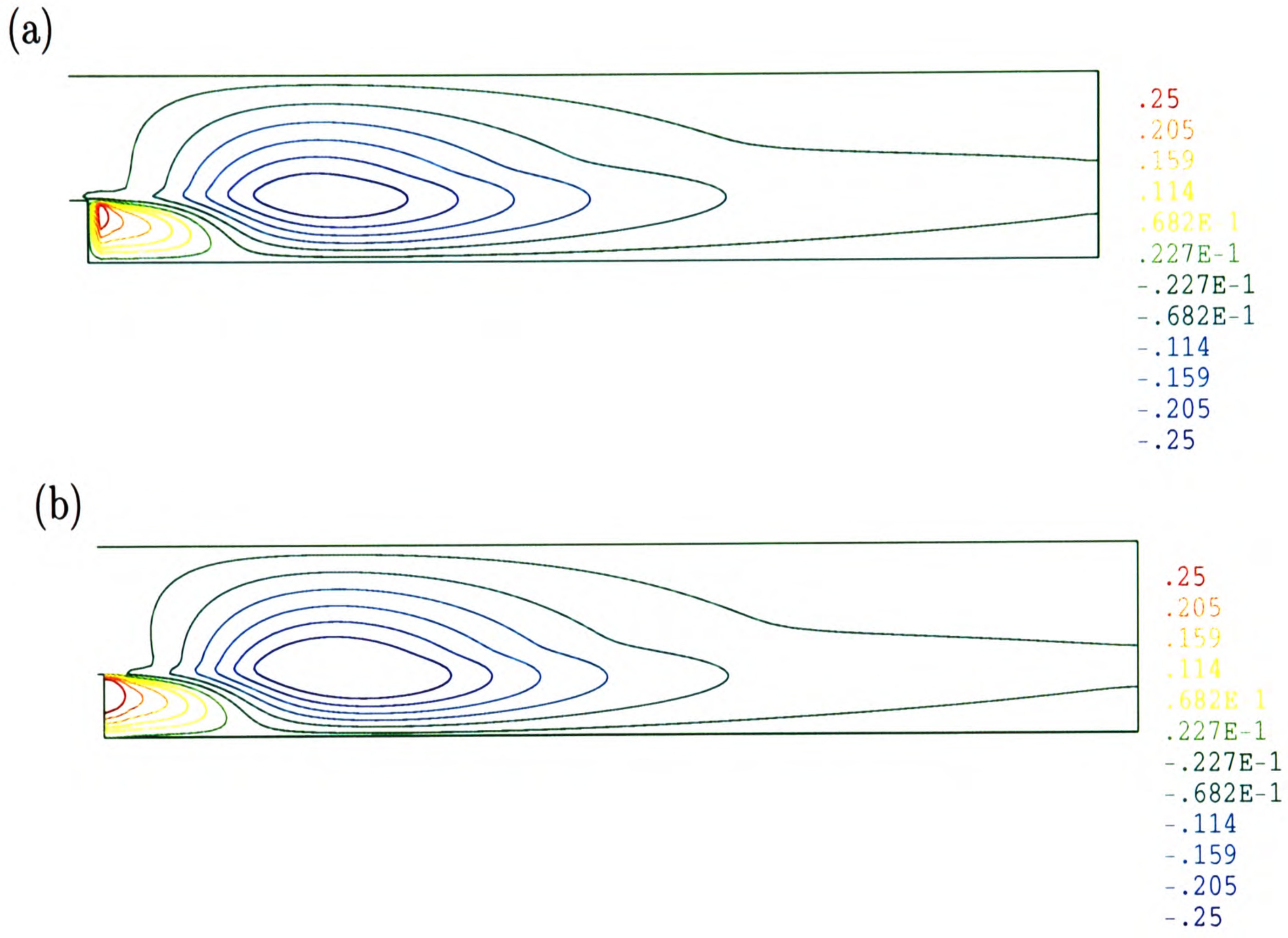


Figure 7.5: $Re = 50000$, v-velocity contours, (a) $vb - cc$, (b) cc

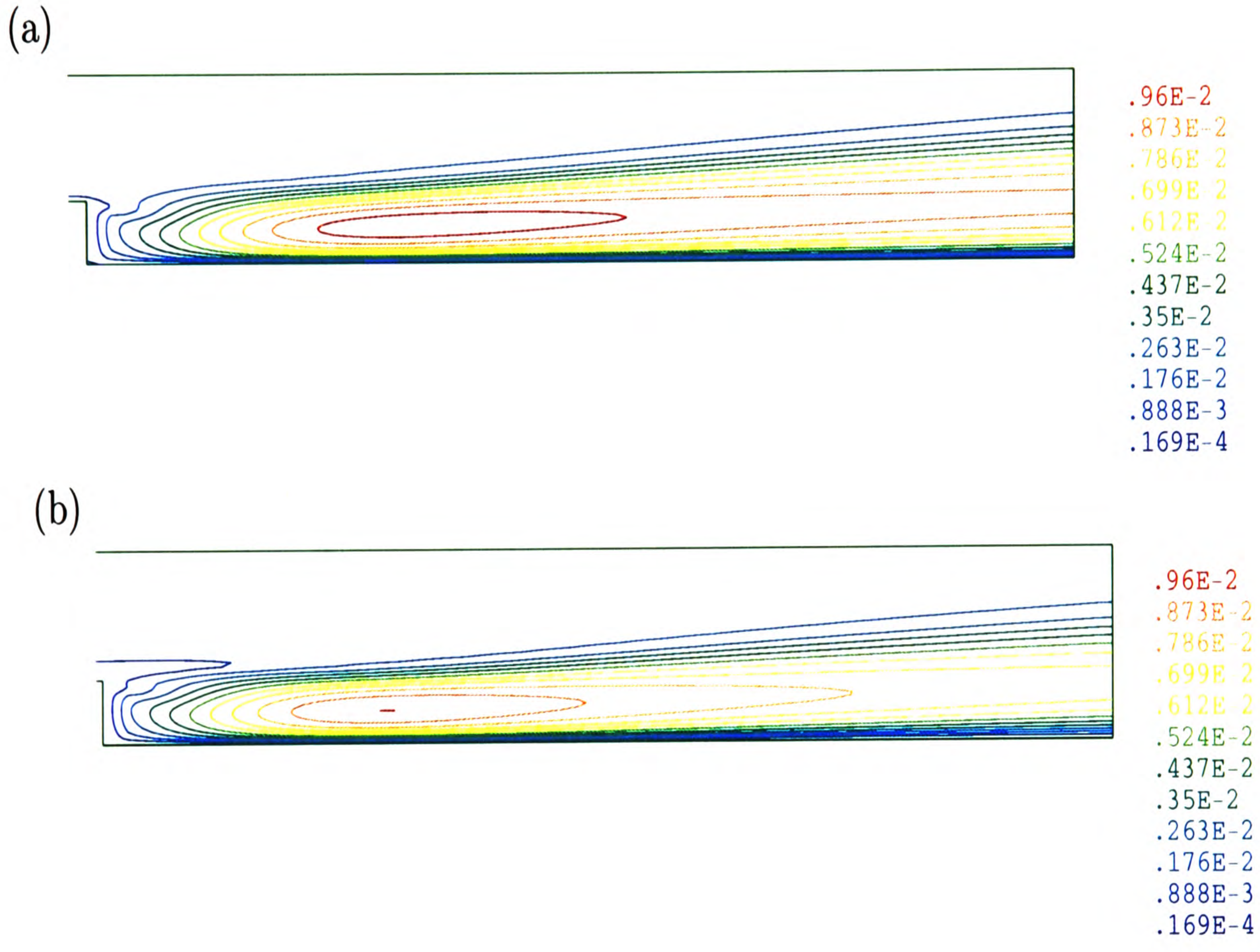


Figure 7.6: $Re = 50000$, turbulent viscosity contours, (a) $vb - cc$, (b) cc

7.3 Flow over an Aircraft Wing

This case involves flow over an aircraft wing. The geometry of the wing was taken from ONERA M6 specifications, [157]. The ONERA M6 wing is a classic CFD validation case for transonic flow. Many authors have investigated transonic flow over the ONERA M6 wing, at Mach numbers of 0.7 and above, i.e. supersonic flow, shock and turbulent boundary layer separation at various angles of attack. It is beyond the scope off this research to investigate the complexities of transonic flow, the simulations carried out employ low-speed Mach number of 0.3. A literature search for incompressible viscous flow over the ONERA wing at lower mach numbers was undertaken. No comparison data has been obtained and the simulations undertaken in this section will be used as a comparison of combined vertex-based/cell-centred techniques with purely cell-centred solutions.. The first case involves turbulent flow over a two-dimensional wing. Differences in the results obtained using $vb - cc$ techniques and purely cc solutions are investigated and discussed. Secondly, the methods are applied to three-dimensional flow over the wing employing uniform and distorted meshes. Lastly, the wing is rotated to a 10 degree angle of attack and solutions compared.

The ONERA M6 wing is a swept, semi-span wing with no twist. The airfoil shape is symmetric with respect to the chord, the co-ordinates of the ONERA cross-section are given in Appendix E. The co-ordinates have been linearly scaled near the trailing edge so that its thickness is zero. The geometric layout of the wing is given in Figure 7.7. The leading-edge sweep is 30 degrees and trailing edge sweep 15.8 degrees. The taper ratio is 0.562 giving a mean aerodynamic chord of 0.64607m. The initial flow conditions are u-velocity $96.1253ms^{-1}$, v- and w-velocity are zero. The properties of air are set to, density $1.3807kgm^{-3}$, laminar viscosity $1.62748 \times 10^{-5}m^2s^{-1}$ giving a Reynolds number of about 5 million. Velocity was fixed at the inflow boundary and zero pressure imposed at the outflow boundary. The speed of sound was taken as $320.418ms^{-1}$ giving a initial Mach number of 0.3. A wall boundary condition was applied to the velocity and turbulent variables at the wing surface. The solution convergence criteria required that the $L2$ norm of the change in the solution dropped by 5 orders-of-magnitude.

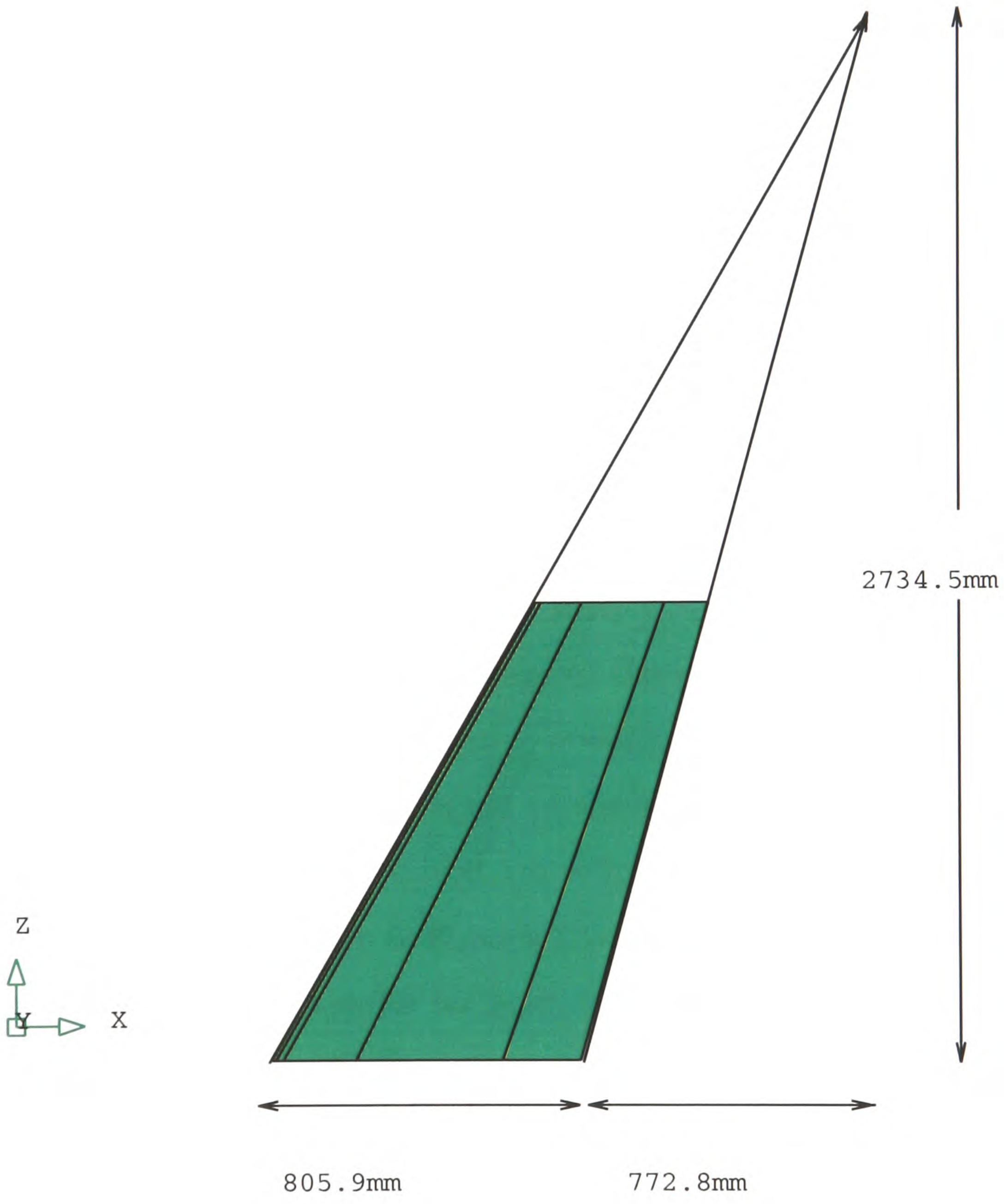


Figure 7.7: Geometry of swept back wing

7.3.1 Two-dimensional Flow

The two-dimensional simulation is performed on a uniform C-Mesh of approximately 4,500 elements and 25,000 elements. The cross-section dimensions of the airfoil are taken at, $z = 0$, of the ONERA M6 wing, Appendix E.

In the $vb - cc$ method the velocity component solution points are located on the surface of the wing. At these locations the values of velocity are set to zero. In the cc method the solution points are located at the element centre and a wall boundary condition is employed setting the velocity values on the wing surface to zero. Figures 7.9 and 7.10 show the u-velocity contours for cc and $vb-cc$ solutions respectively. On the coarser mesh the influence of zero velocity on the wing surface is much greater for $vb - cc$ solutions and the u-velocity values downstream of the trailing edge of the wing are lower than those obtained cell-centred. Refining the mesh produces more comparable results. In order to compare the results the u-velocity is plotted along a line from the wing surface in the positive y direction, Figure 7.13, and along the wake, on a line from the trailing edge of the wing in the x direction, Figure 7.15. As can be seen from the plots, the cc u-velocity results are higher than $vb - cc$ u-velocity results in the region adjacent to the wing surface and along the wake. The differences in the results obtained are reduced with mesh refinement. The v-velocity contour plots are shown in Figures 7.11 and 7.12 for cc and $vb - cc$ solutions respectively. The plots are similar for both methods, but on the coarser mesh the $vb - cc$ method fails to identify the local maximum and minimum v-velocity values in the region adjacent to the wing surface towards the trailing edge of the wing. This is shown in a plot of the v-velocity values on a line from the wing surface in the positive y direction, Figure 7.14.

The turbulent generation rate, which enters the turbulent variable transport equations as a source, is calculated from the velocity gradients, Equation 7.14. Figures 7.16 and 7.17 give the turbulent generation contours for $vb - cc$ and cc solutions respectively. As can be seen from the plots there is relatively small values of turbulent generation, when solving *cc* on the coarser mesh, in the region around the trailing edge of the wing. In contrast *vb* − *cc* solutions on the same mesh show large turbulent generation in this region. This is a consequence of much larger velocity gradients being obtained in this region for *vb* − *cc* solutions, as illustrated in Figure 7.13. In the viscous sub-layer, adjacent to the wing surface, laminar flow is assumed and the turbulent generation rate is set to zero. However, the vertex-based velocity solutions obtained just outside this region have a lower value than *cc* solutions and consequently larger gradients are encountered to achieve free stream velocity values. In the region downstream to the trailing edge of the wing, the vertex-based u-velocity values are influenced by the zero value located at the intersection of the upper and lower wing surface, Figure 7.8. In the cell-centred solution procedure the u-velocity at the intersecting face of the two adjacent downstream elements is interpolated from the adjacent element based values. Thus, the zero velocity on the surface of the wing has less influence on the downstream value. Figure 7.15 shows that even with mesh refinement the vertex-based u-velocity solutions along the line $y = 0$ are lower than cell-centred solutions. In the region downstream adjacent to the wing trailing edge larger velocity gradients are encountered when solving vertex-based and hence higher turbulent generation. On the finer mesh the maximum value obtained in this region was 8.94×10^6 *cc*, compared to 1.64×10^8 *vb* − *cc*.

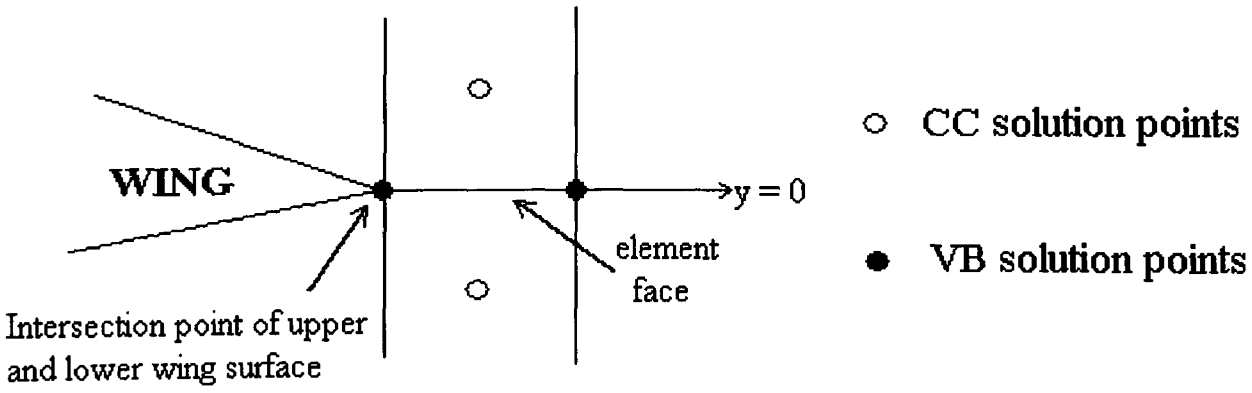


Figure 7.8: Vertex-based and cell-centred solution points at trailing edge of wing

Figures 7.18 and 7.19 show the turbulent viscosity contours for cc and $vb - cc$ solutions respectively. Due to the higher turbulent generation rate encountered when solving $vb - cc$, higher turbulent viscosity values are obtained near the trailing edge of the wing and along the wake at $y = 0$. The differences are more pronounced on the coarser mesh, with mesh refinement the $vb - cc$ turbulent viscosity values decrease giving a maximum value of 0.231 compared to 0.111 cc.

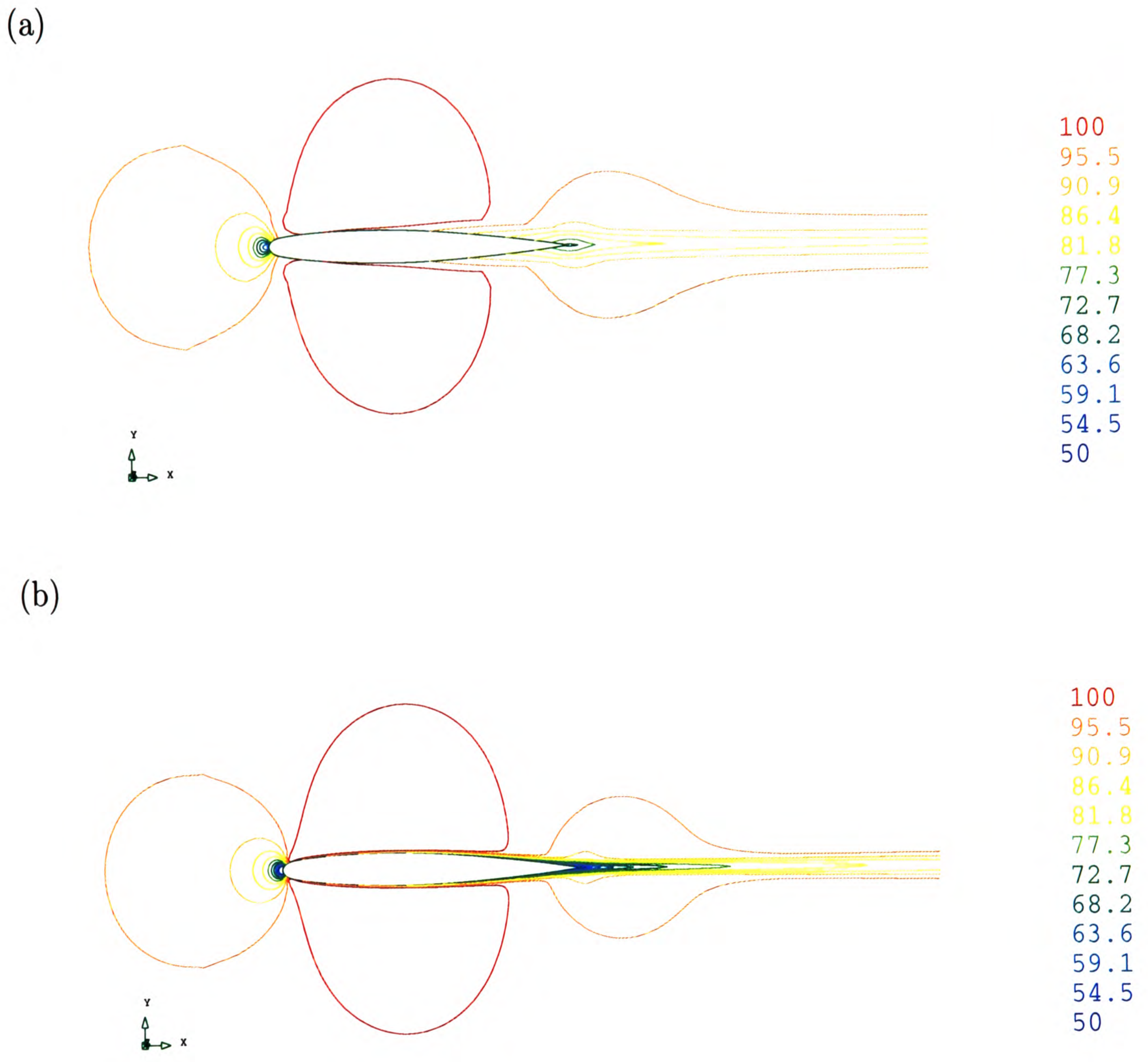


Figure 7.9: u-velocity contours (*cc*), (a) 4500 elements, (b) 25000 elements

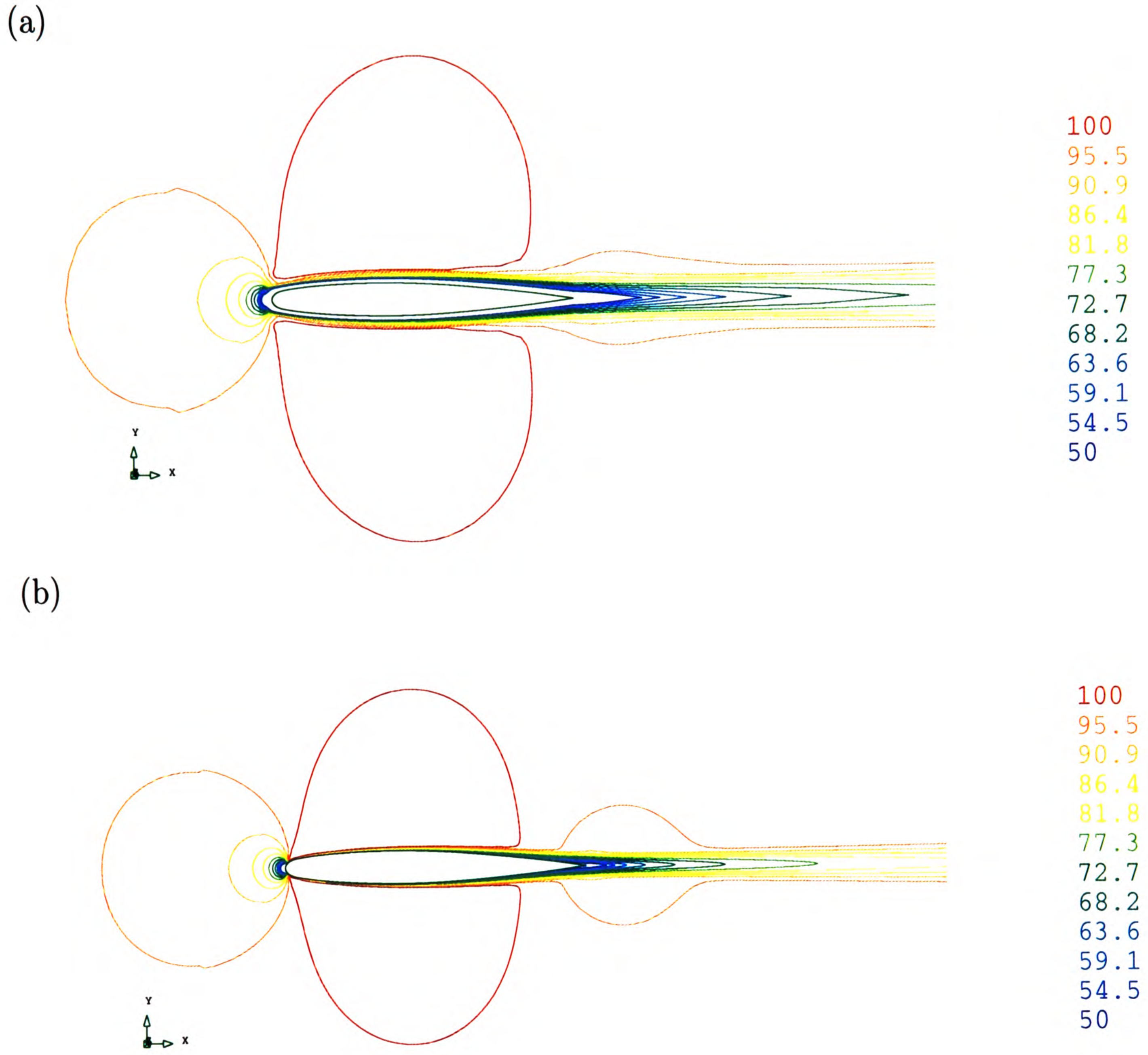


Figure 7.10: u-velocity contours $(vb - cc)$, (a) 4500 elements, (b) 25000 elements

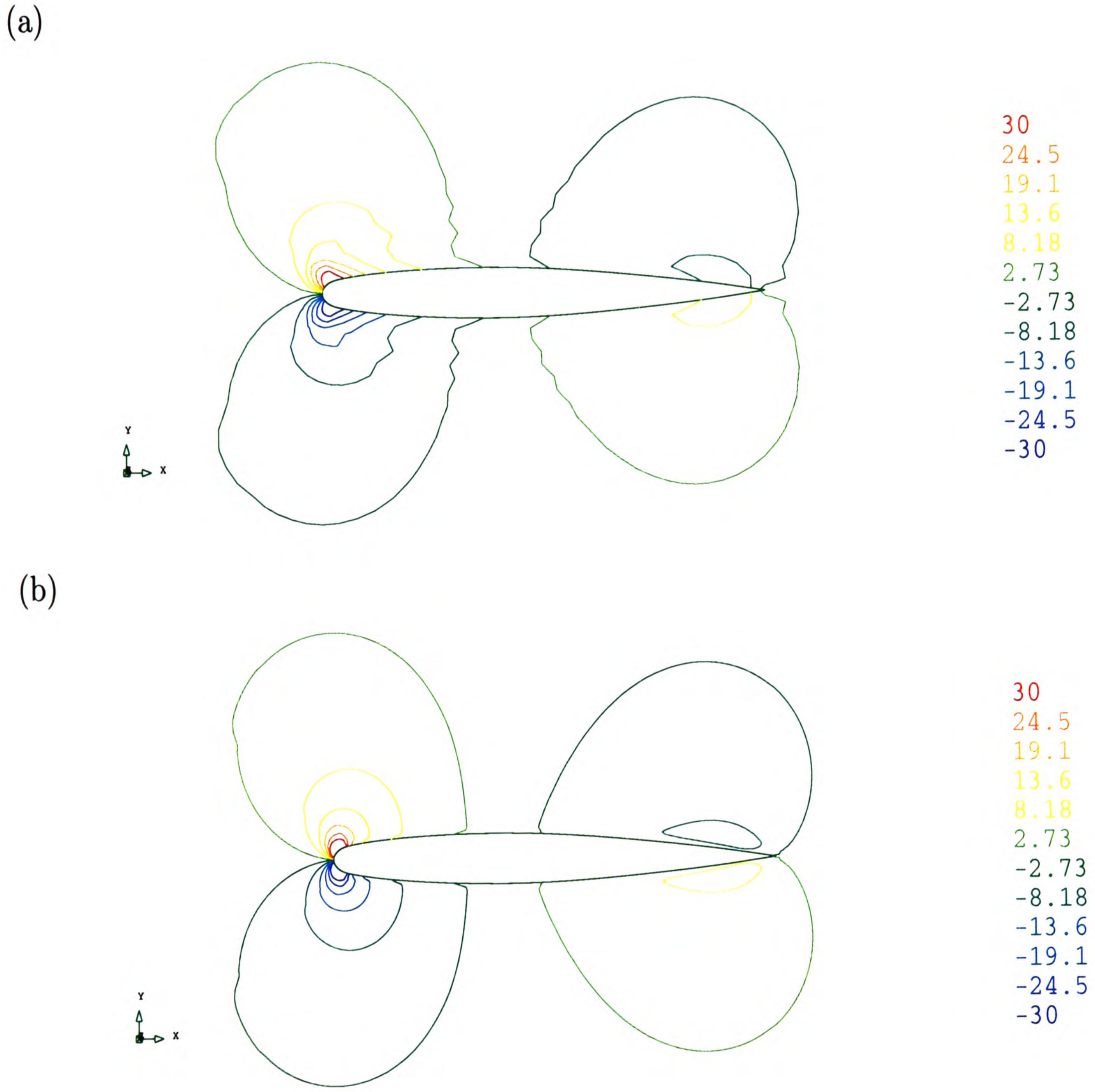


Figure 7.11: v-velocity contours (*cc*), (a) 4500 elements, (b) 25000 elements

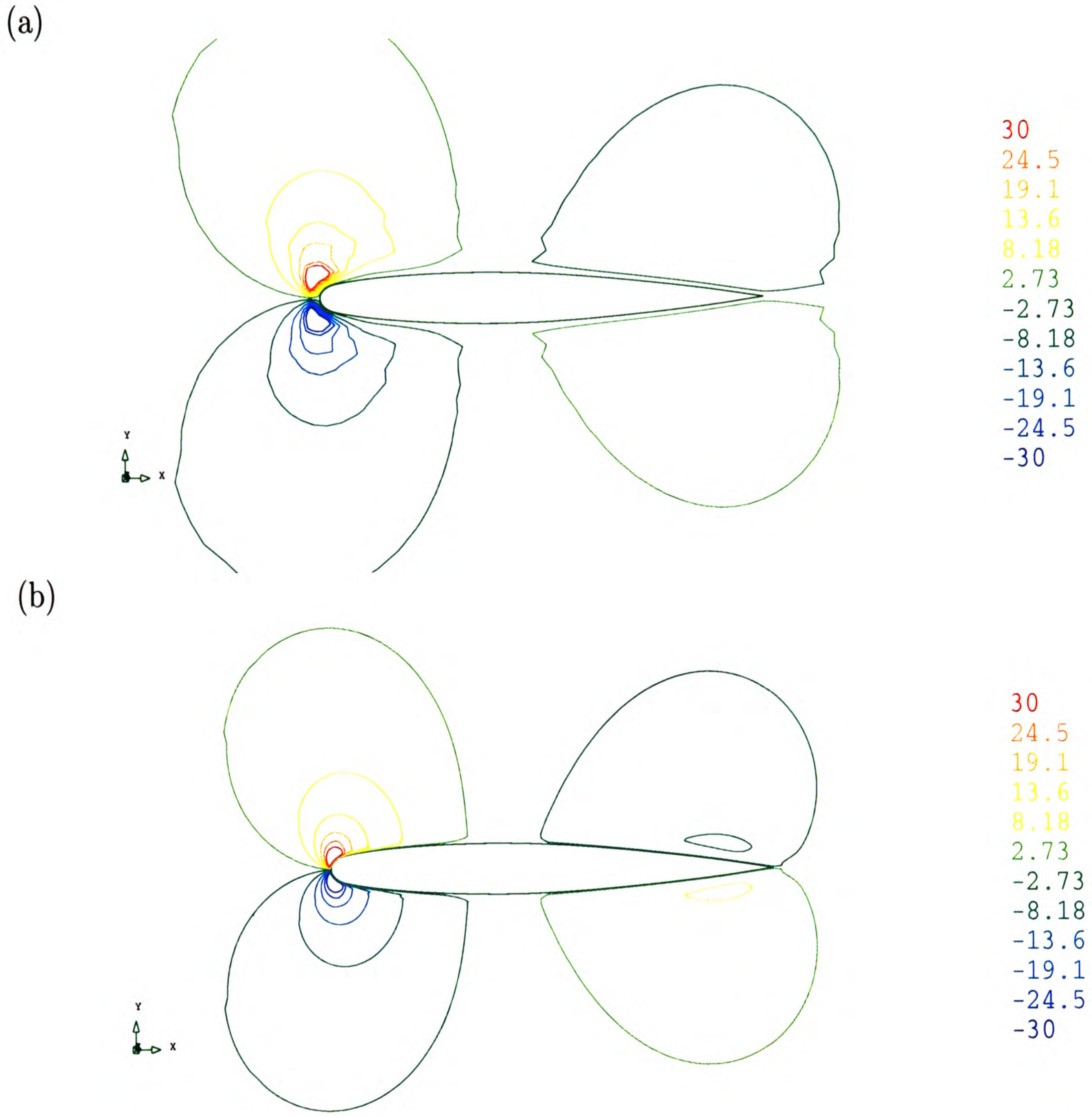


Figure 7.12: v-velocity contours ($vb - cc$), (a) 4500 elements, (b) 25000 elements

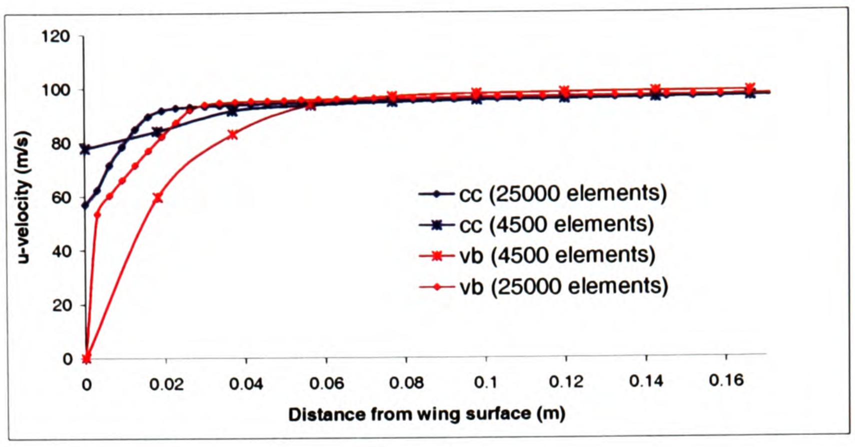


Figure 7.13: U-velocity along line from wing surface in the positive y direction

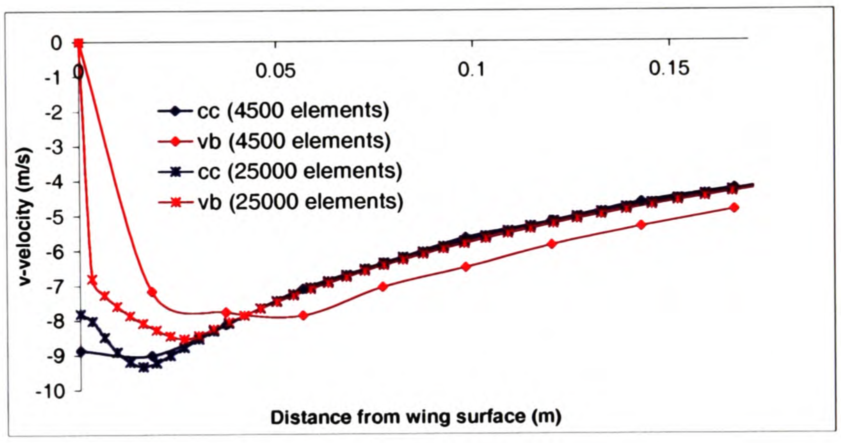


Figure 7.14: V-velocity along line from wing surface in the positive y direction

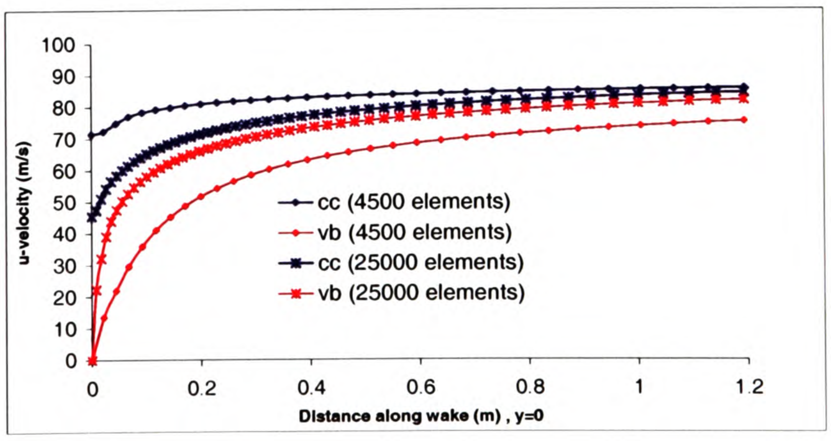


Figure 7.15: U-velocity along line from trailing edge of wing in the positive x direction

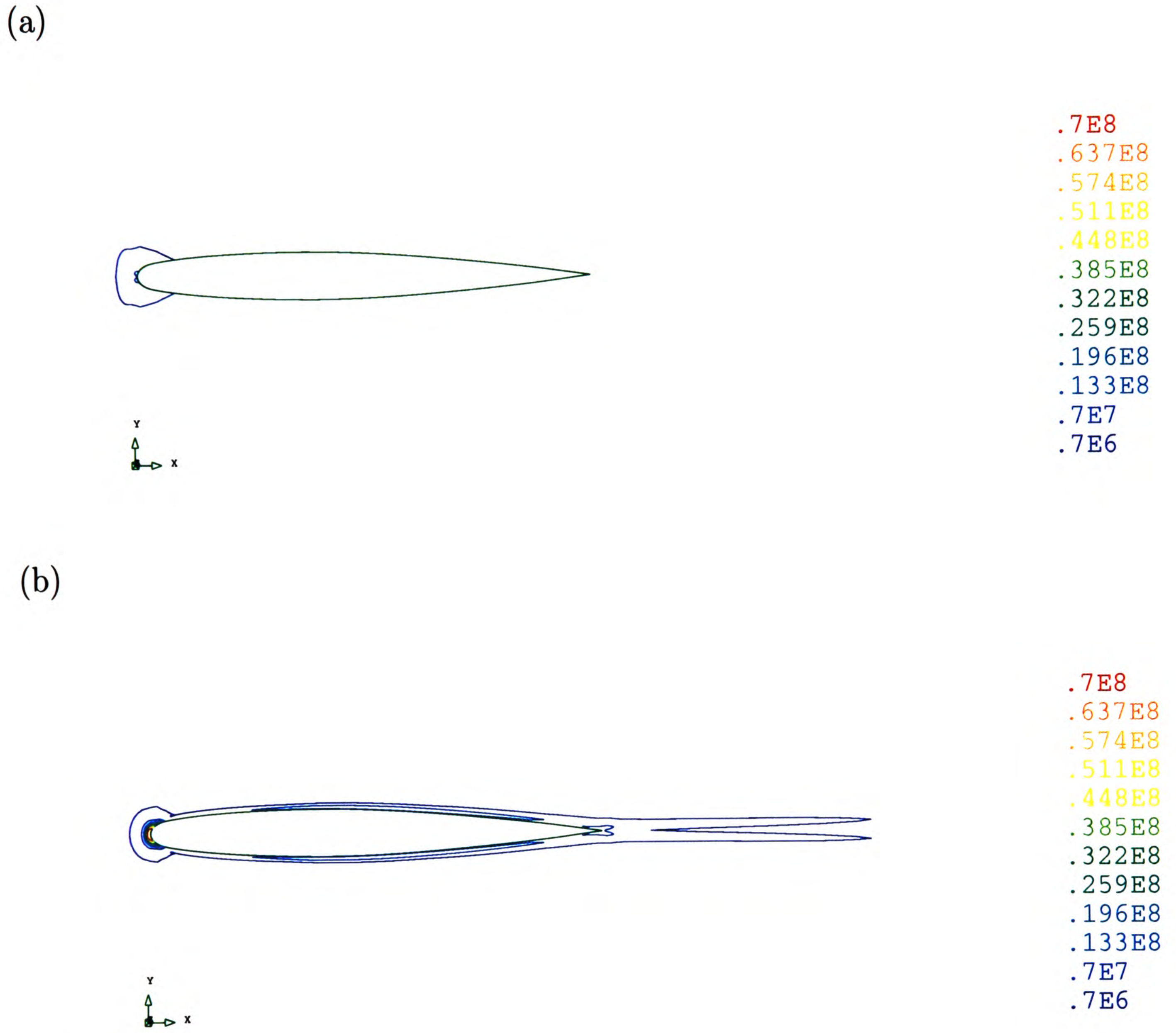


Figure 7.16: Turbulent generation contours (cc), (a) 4500 elements, (b) 25000 elements

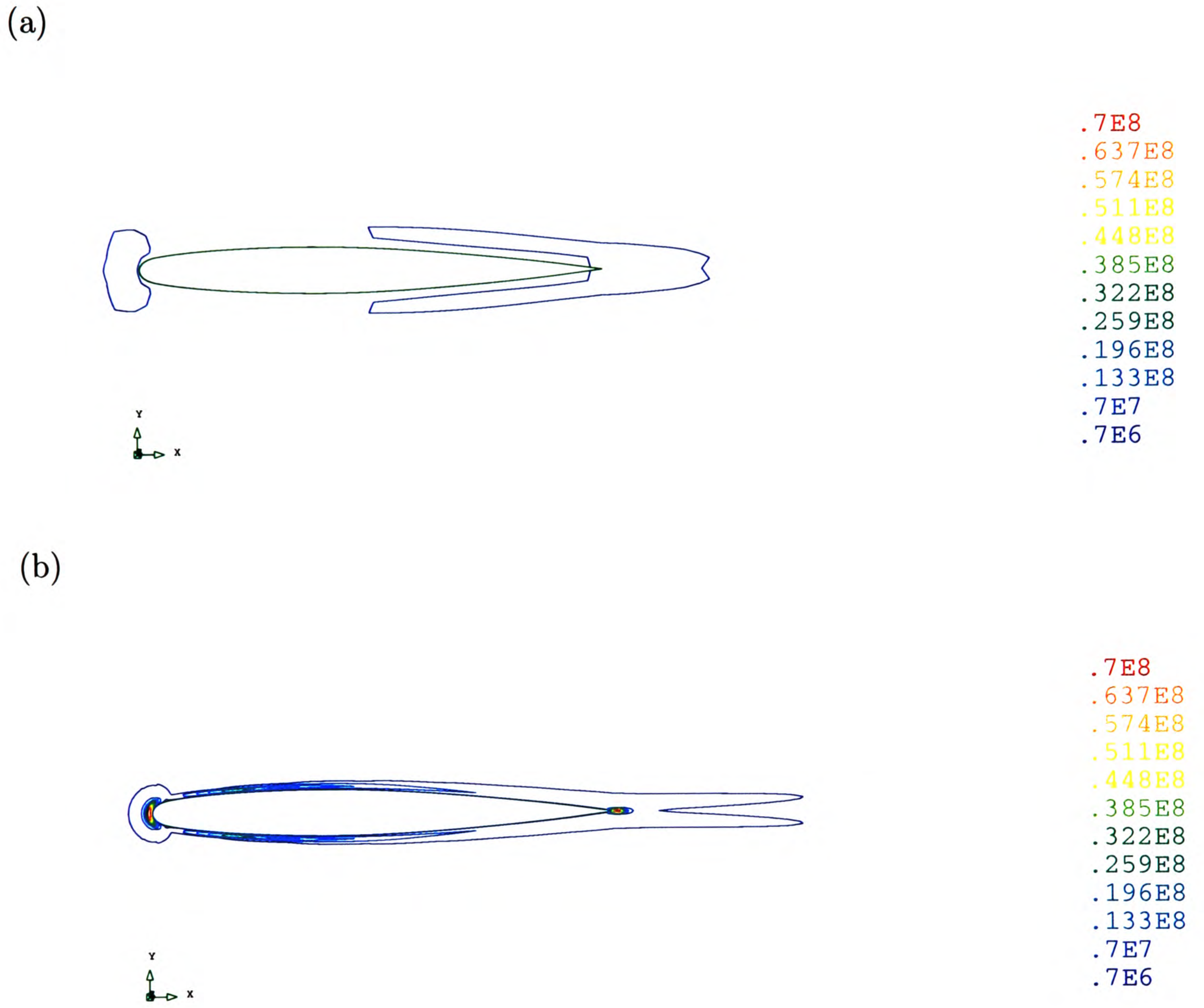


Figure 7.17: Turbulent generation contours ($vb - cc$), (a) 4500 elements, (b) 25000 elements

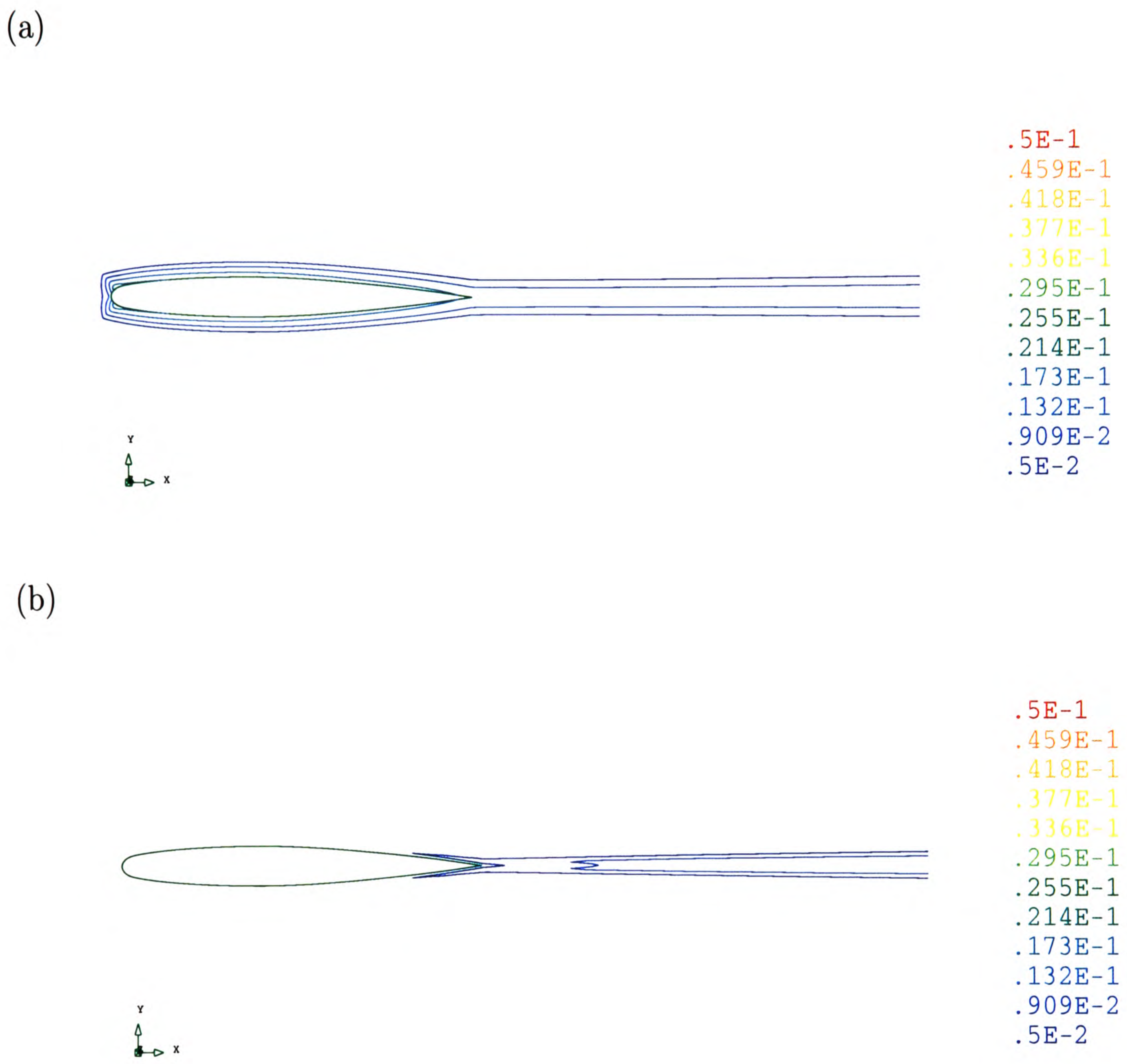


Figure 7.18: Turbulent viscosity contours (*cc*), (a) 4500 elements, (b) 25000 elements

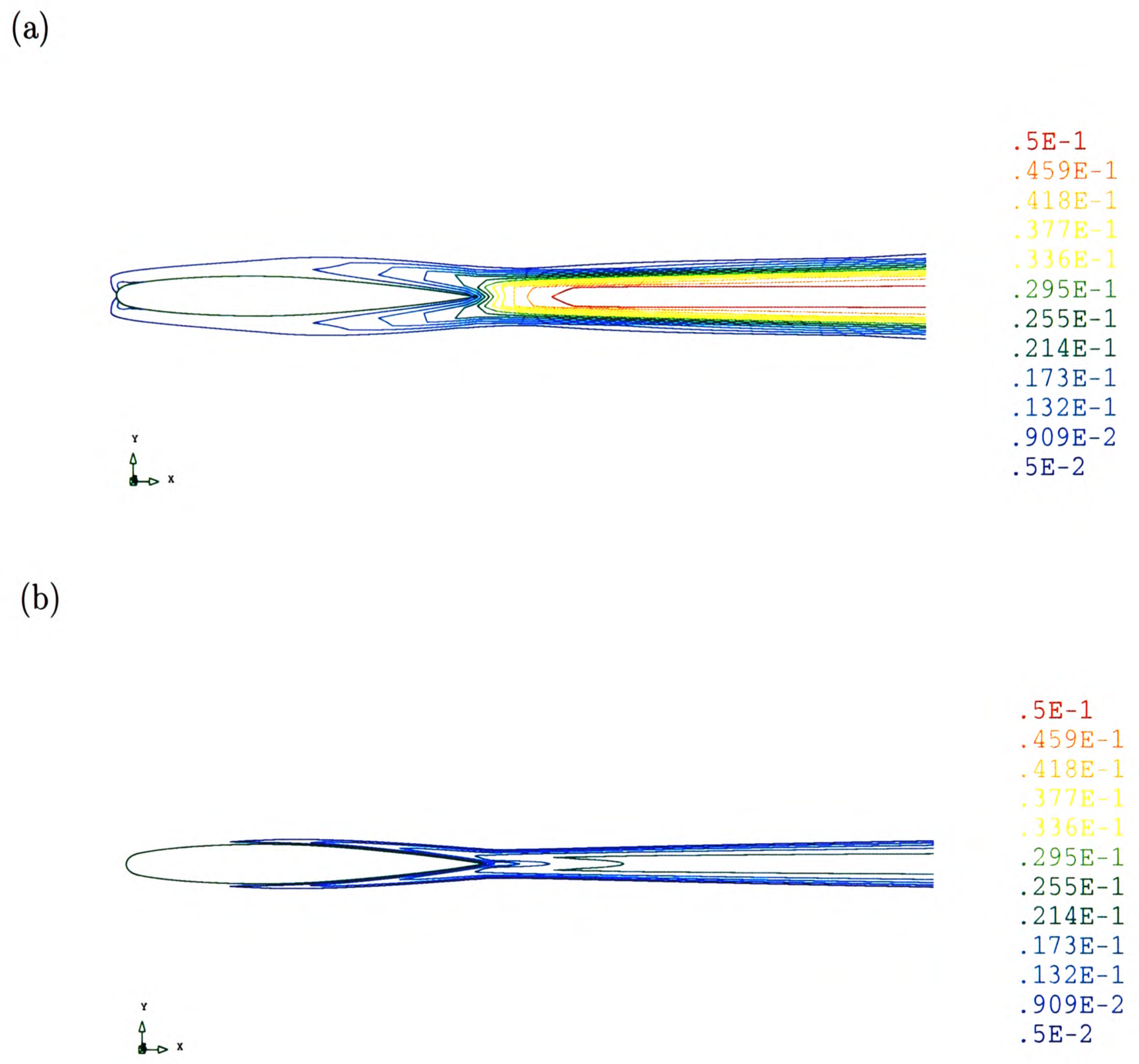


Figure 7.19: Turbulent viscosity contours $(vb - cc)$, (a) 4500 elements, (b) 25000 elements

7.3.2 Three-dimensional Flow - Uniform C-Mesh

A uniform C-Mesh, Figure 7.20, of approximately 100,000 elements is employed in the simulations. This is a relatively coarse mesh, complex flow simulations of flow over the ONERA M6 wing normally employ meshes of three to four hundred thousand elements. In order to compare the $vb-cc$ and cc solutions cross section contours are shown in three planes. 1) The Z-plane, at $z = 0.558$, which is approximately half the wing span. 2) The Y-plane, at $y = 0$, which is the symmetry plane. 3) The X-plane, $x = 0.403$ at $z = 0$, at a sweep angle of 30 degrees.

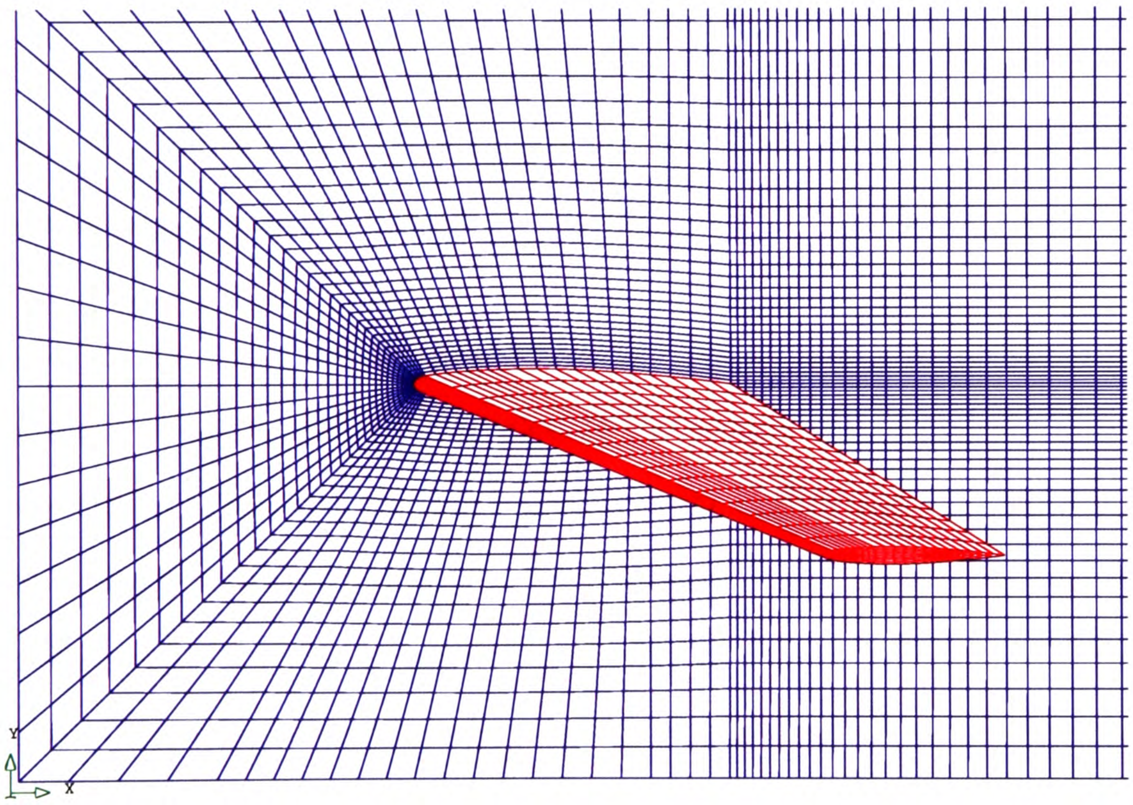


Figure 7.20: Three-dimensional C-Mesh, Mesh 1

Figures 7.21, 7.22 and 7.23 show the u-velocity plots in the Z-plane, Y-plane and X-plane respectively. The $vb - cc$ plots gave lower u-velocity values adjacent to the wing surface and downstream of the the wing trailing edge. The minimum u-velocity value for cc solutions was $49.6m/s$. The maximum values obtained were $105m/s$ for cc results and $107m/s$ for $vb - cc$ results. The v-velocity contour plots are shown in Figures 7.24 and 7.25 for the Z-plane and X-plane respectively. The v-velocity being zero in the Y-plane. The $vb - cc$ results gave slightly higher v-velocity values along the leading edge of the wing. The maximum obtained was $49.3m/s$ compared to $42.1m/s$ cell-centred. The cc solutions show a slight increase in the v-velocity in the region adjacent to the wing surface towards the trailing edge. This increase is not shown in the $vb - cc$ results. The w-velocity contours are shown in Figures 7.26, 7.27 and 7.28 for the Z-plane, Y-plane and X-plane respectively. For both methods similar results are obtained in the Z- and Y-plane with cc solutions giving slightly higher w-velocity values at the wing's leading edge. The maximum value obtained cc was $44.5m/s$ and $vb - cc$ $39m/s$. The cc solutions in the X-plane indicate some slight recirculation in the region adjacent to the tip of the wing. This is not shown in the $vb - cc$ results. The turbulent viscosity contours are shown in Figures 7.29, 7.30 and 7.31 for the Z-plane, Y-plane and X-plane respectively. The cc solutions show higher turbulent viscosity values in the region adjacent to the tip of the wing whereas the $vb - cc$ solutions show higher values downstream of the wing's trailing edge. The maximum value obtained cc was $.0394m^2s^{-1}$ and $vb - cc$ was $.0318m^2s^{-1}$.

As discussed in the previous section the vertex-based solutions produce higher velocity gradients in the region downstream of the wing's trailing edge. Consequently the turbulent generation rate at this location is far higher than that obtained when solving the velocity cell-centred, as shown in Figure 7.32. The higher turbulent generation in this region for $vb - cc$ solutions results in higher turbulent viscosity values downstream of the wing's trailing edge. In the cc solutions the turbulent generation rate is highest along the leading edge of the wing.

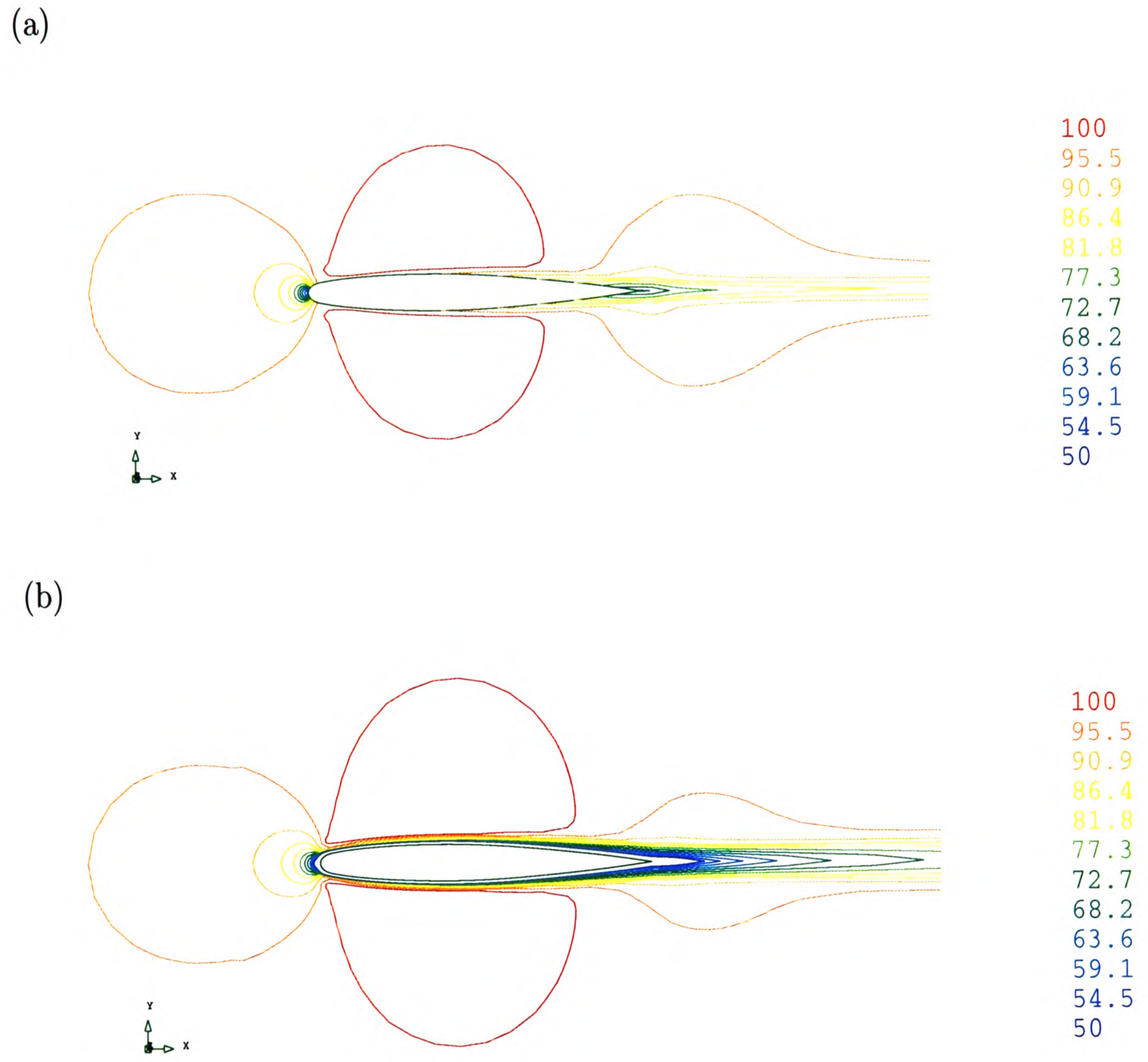


Figure 7.21: Z-plane: u-velocity contours, (a) cc, (b) $vb - cc$

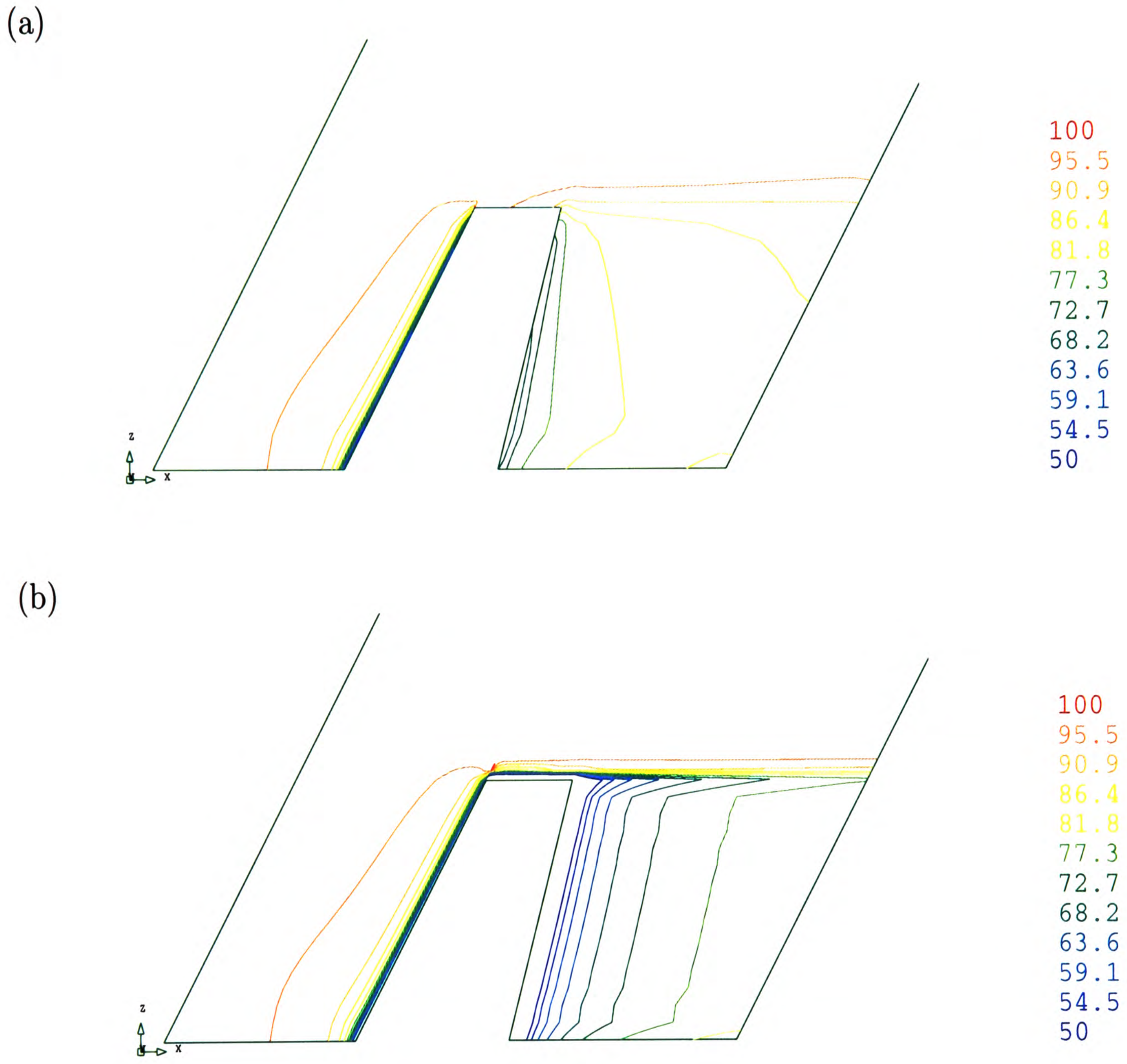


Figure 7.22: Y-plane: u-velocity contours, (a) *cc*, (b) $vb - cc$

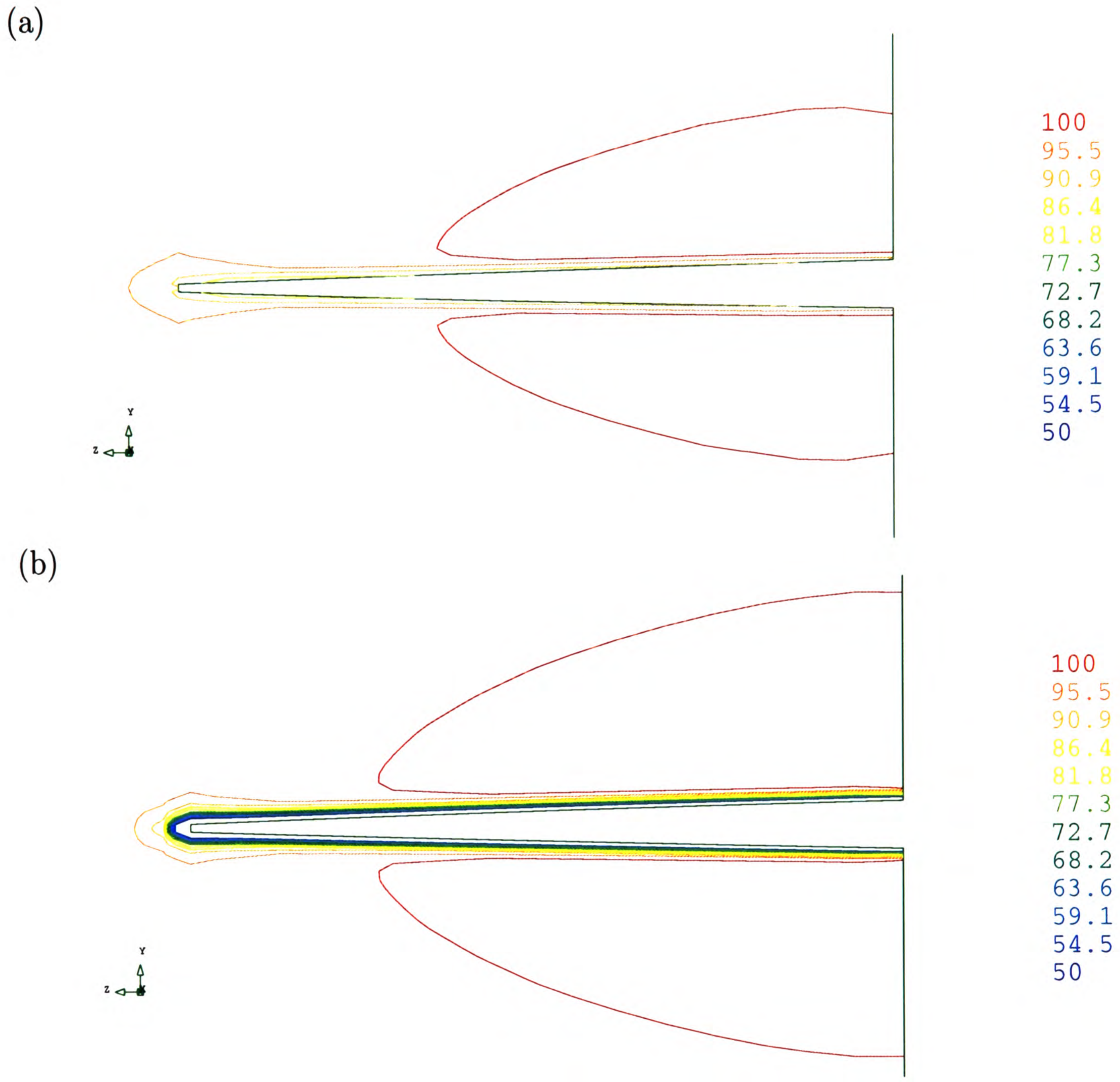


Figure 7.23: X-plane: u-velocity contours, (a) cc, (b) $vb - cc$

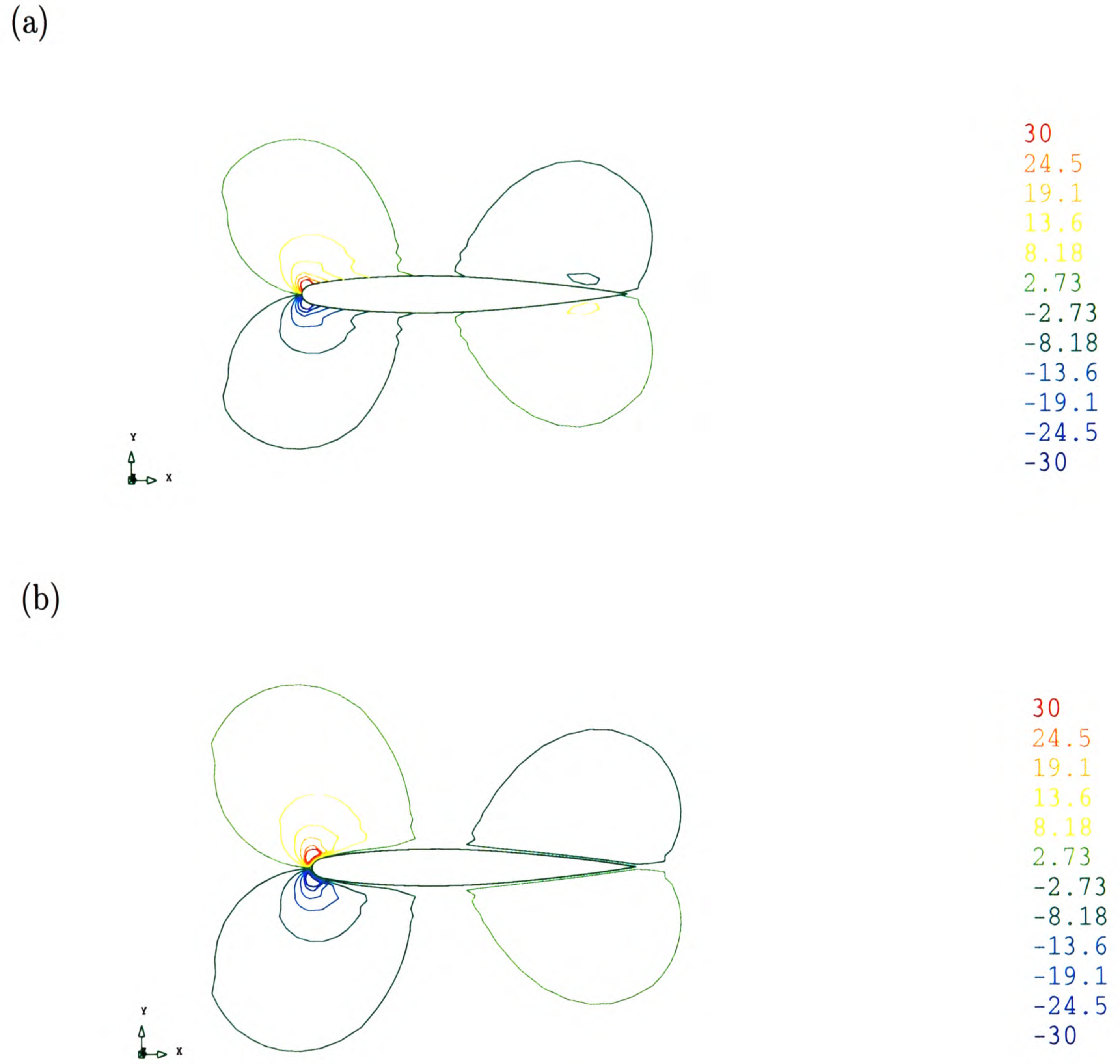


Figure 7.24: Z-plane: v-velocity contours, (a) *cc*, (b) $vb - cc$

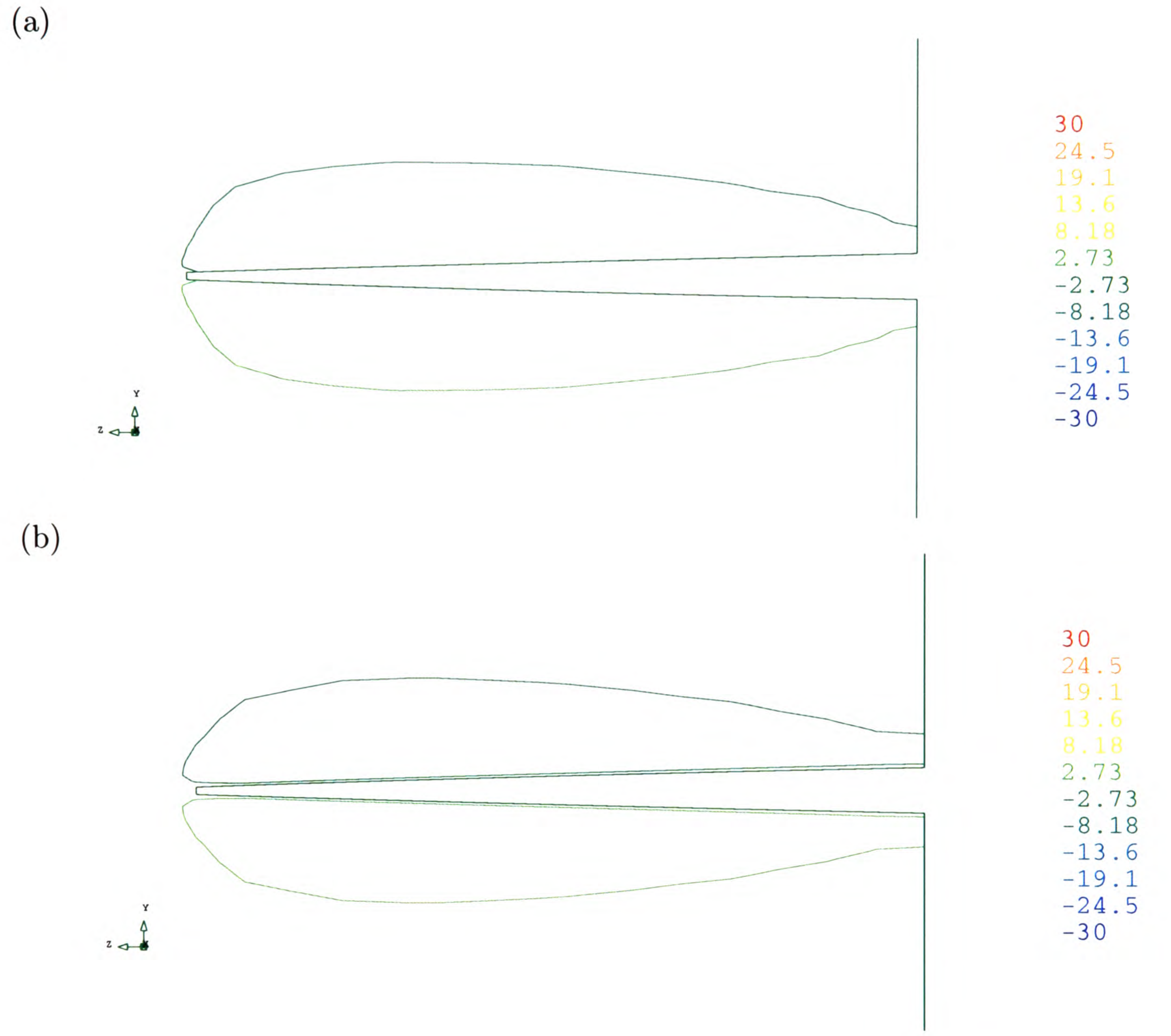


Figure 7.25: X-plane: v-velocity contours, (a) *cc*, (b) $vb - cc$

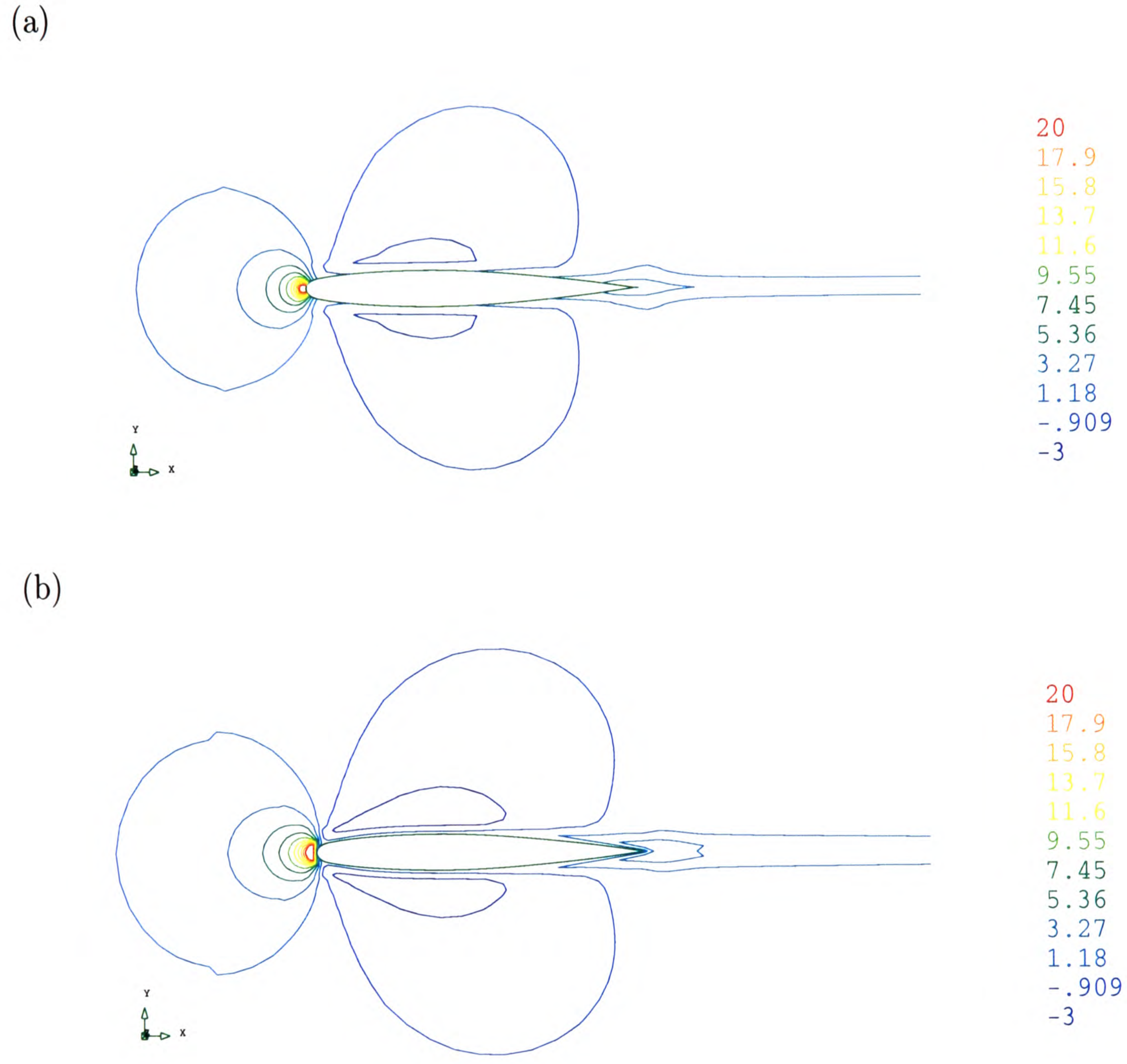


Figure 7.26: Z-plane: w-velocity contours, (a) *cc*, (b) $vb - cc$

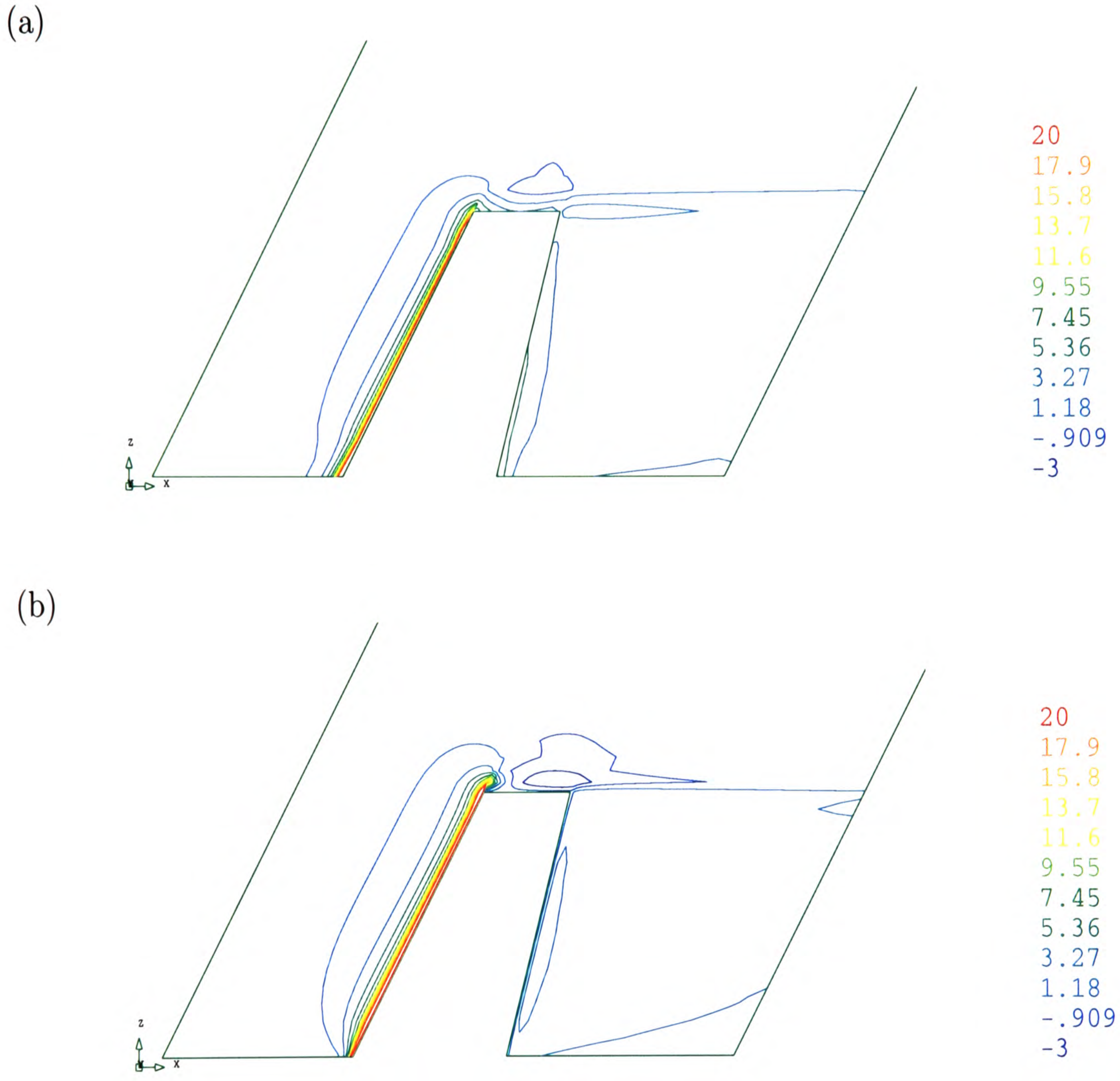


Figure 7.27: Y-plane: w-velocity contours, (a) cc, (b) $vb - cc$

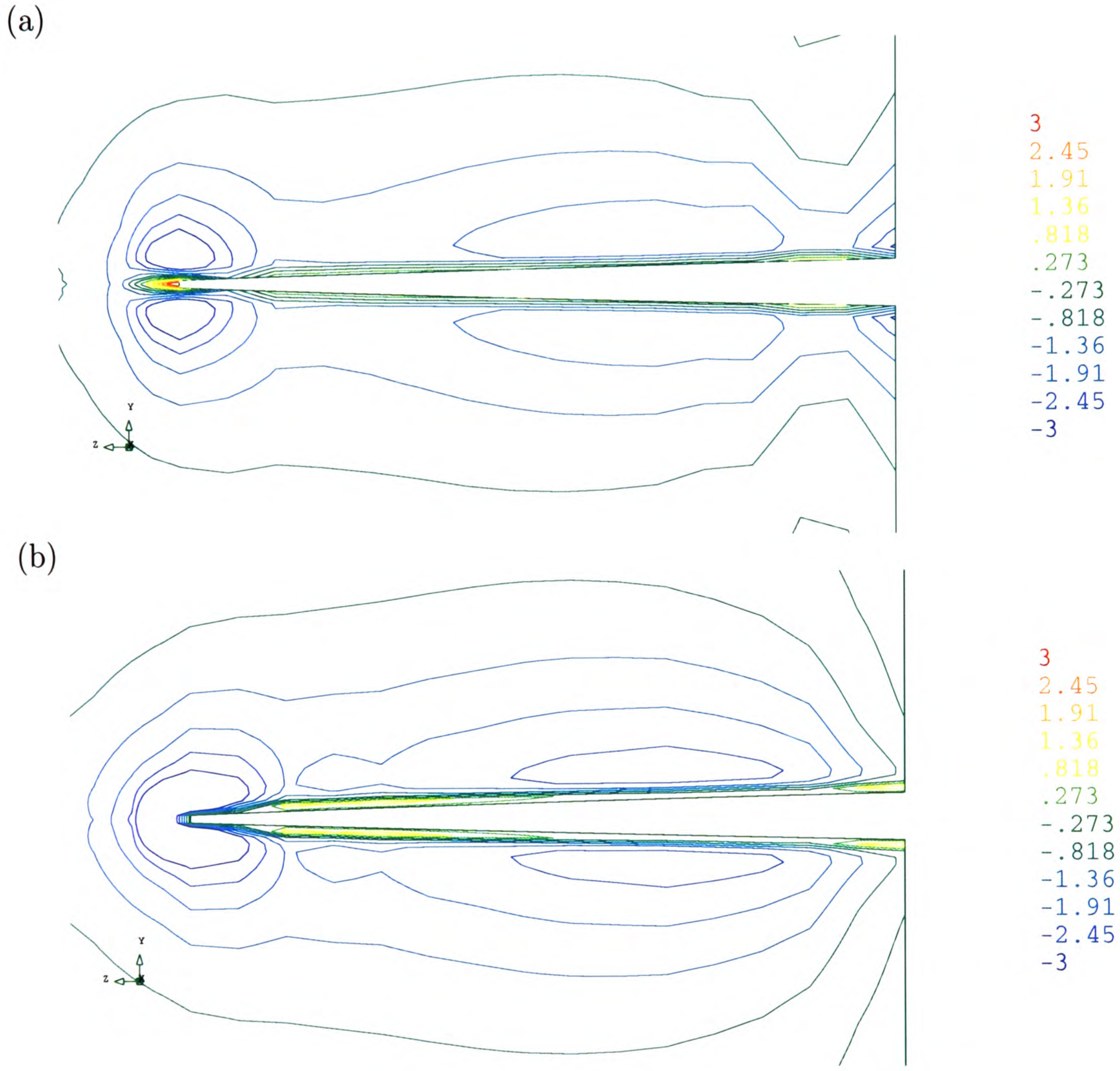


Figure 7.28: X-plane: w-velocity contours, (a) *cc*, (b) $vb - cc$

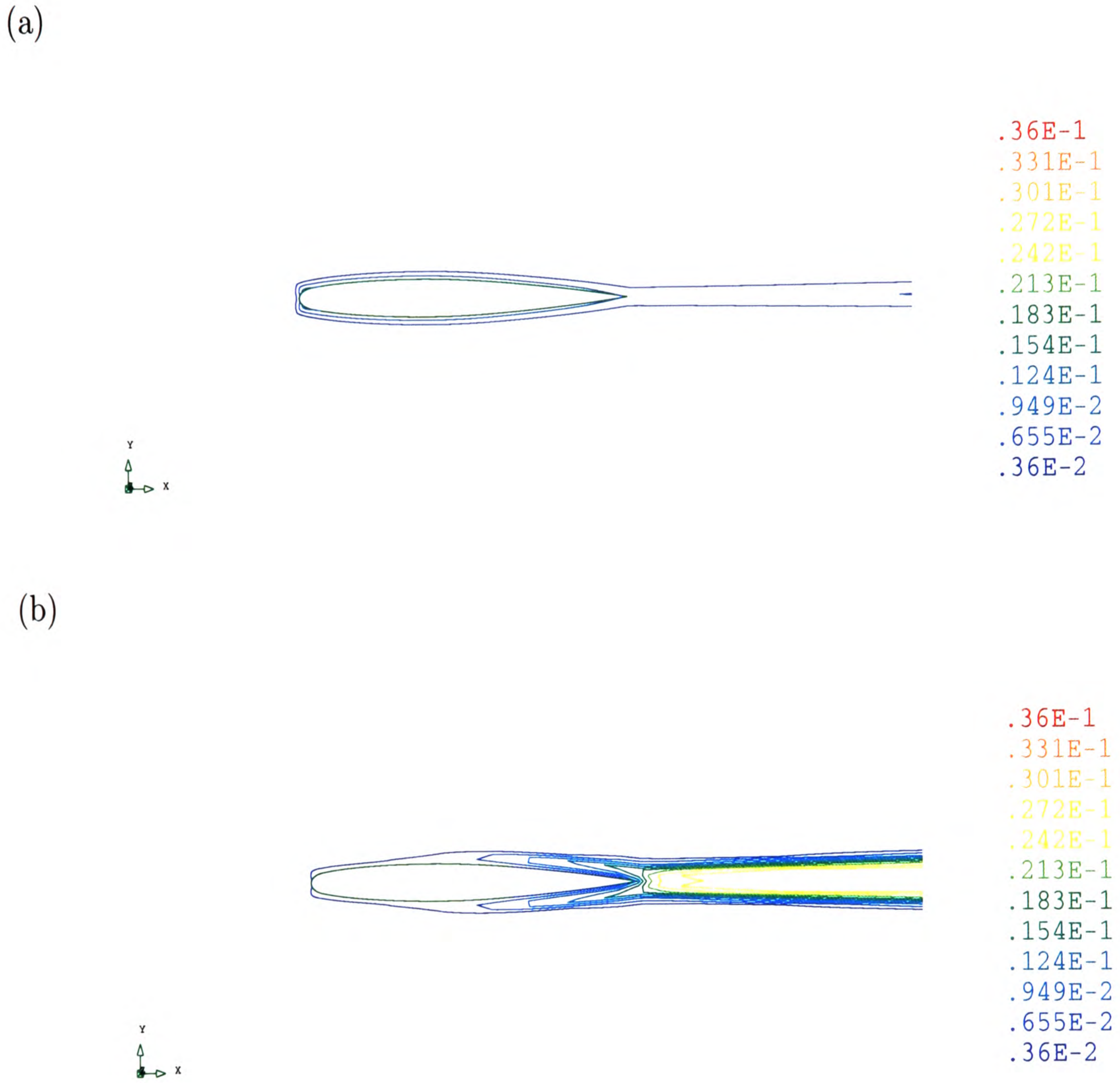


Figure 7.29: Z-plane: turbulent viscosity contours, (a) cc, (b) $vb - cc$

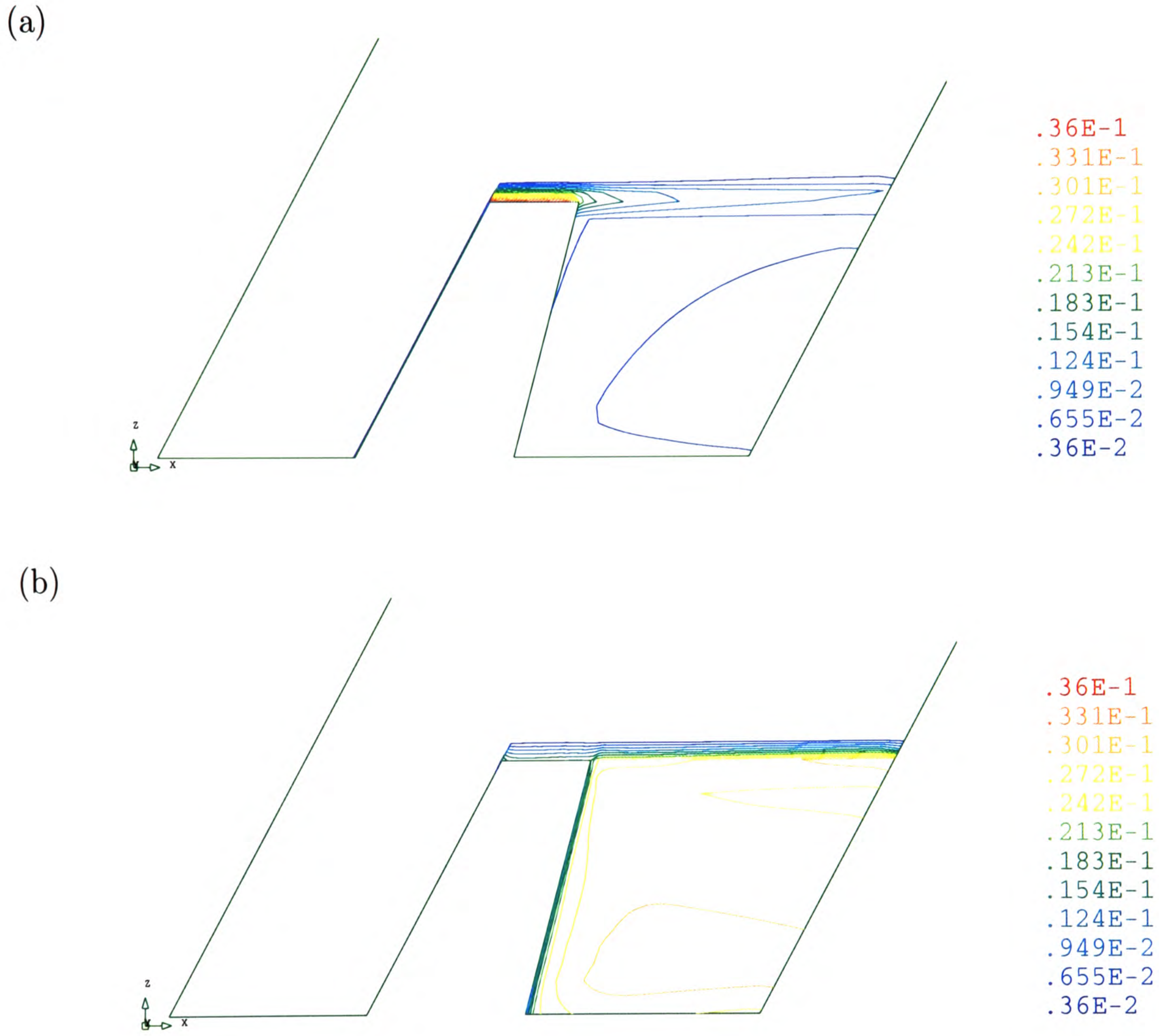


Figure 7.30: Y-plane: turbulent viscosity contours, (a) *cc*, (b) $vb - cc$

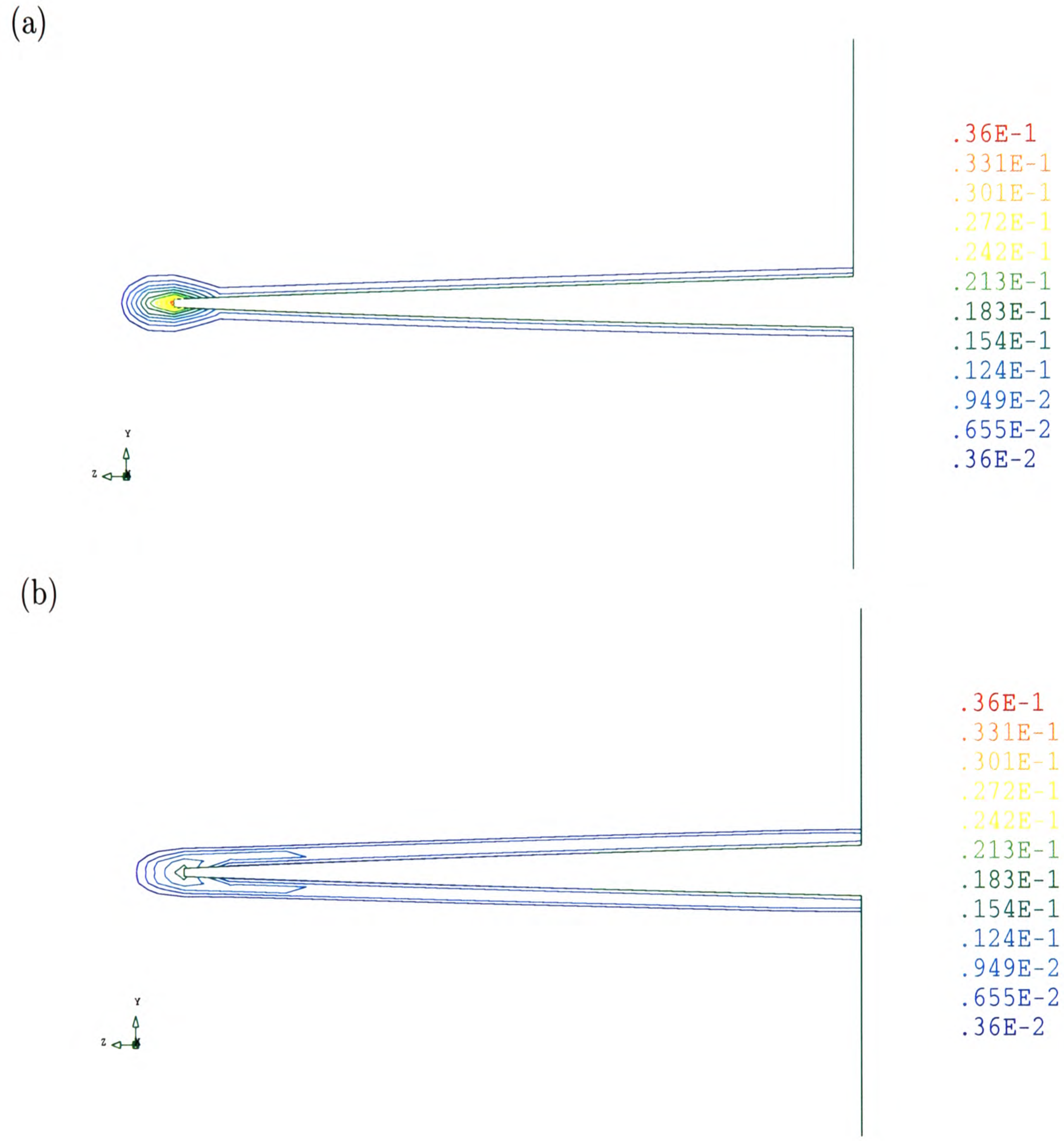


Figure 7.31: X-plane: turbulent viscosity contours, ((a) cc, (b) $vb - cc$

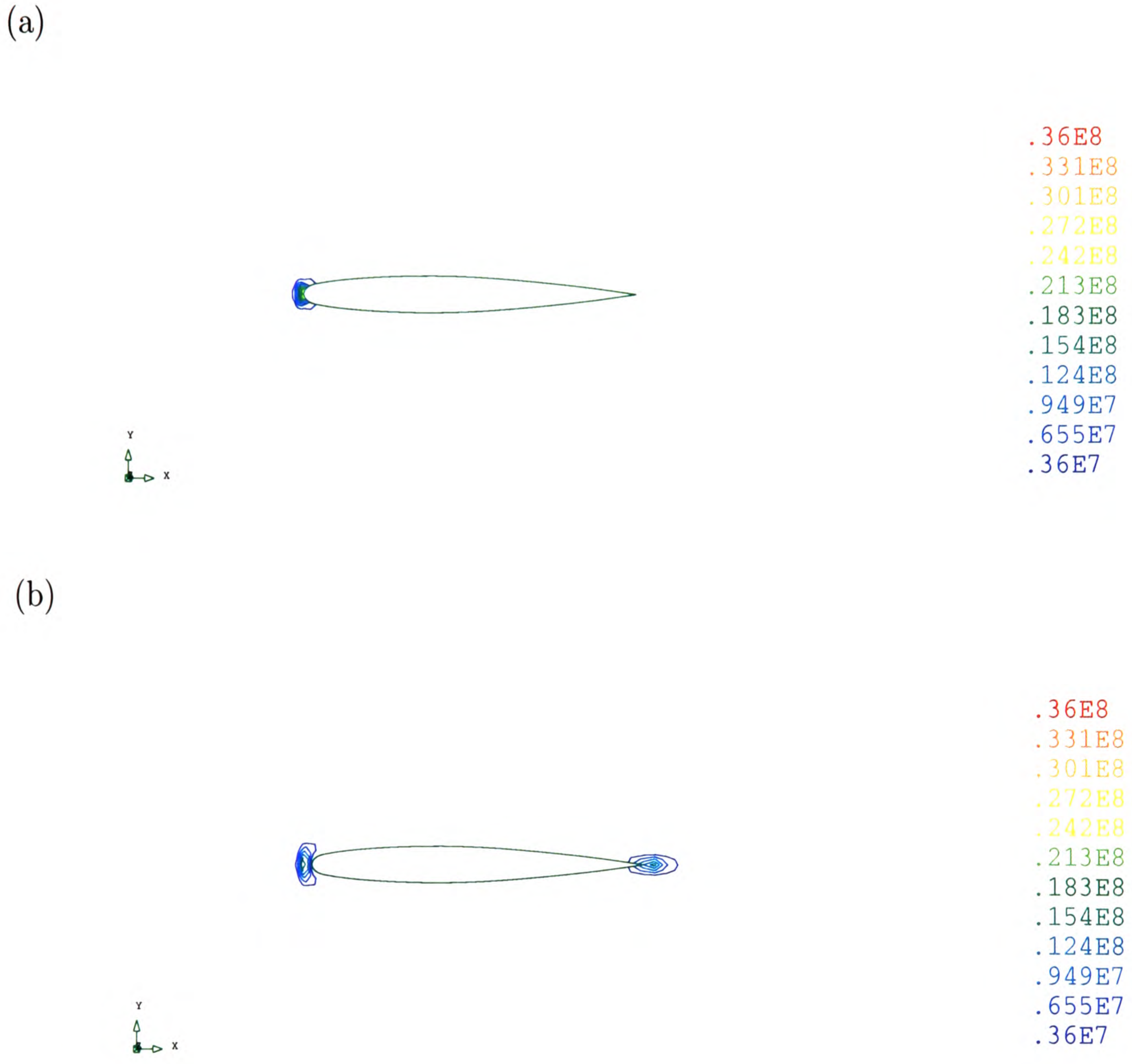


Figure 7.32: Z-plane: turbulent generation rate contours, (a) cc, (b) $vb - cc$

7.3.3 Distorted Three-dimensional Mesh

In order to compare how the solution methods performed on a distorted mesh, various degrees of skewness were introduced into mesh 1, Figure 7.20. The simulations as outlined in the previous section were performed employing two further meshes, mesh 2, Figure 7.33 and mesh 3, Figure 7.34. Due to the severity of the skewness in mesh 3, the cc method failed to produce solutions. Solutions were obtained on mesh 2 for the *cc* method and on both mesh 2 and 3 for the $vb - cc$ method.

Figures 7.35, 7.36 and 7.37 show the u-velocity plots obtained on the Z-plane, Y-plane and X-plane respectively. The $vb - cc$ solutions gave similar results for mesh 2 and mesh 3. The $vb - cc$ u-velocity values were lower adjacent to the wing surface and downstream of the wing's trailing edge than those obtained on mesh1. The maximum u-velocity value remained unchanged at $107m/s$. The *cc* solution did not achieve the minimum u-velocity value of $49.6m/s$ obtained on mesh1. The minimum value obtained on mesh 2 was $74.1m/s$. The v-velocity contours are shown in Figures 7.38 and 7.39 for the Z-plane and X-plane. The v-velocity in the Y-plane being zero. The v-velocity contour plots for $vb - cc$ solutions are similar to those obtained on mesh 1. The v-velocity value range decreased slightly from $[-49.3m/s$ - $49.3m/s]$ to $[-40.4m/s$ - $40.4m/s]$. The *cc* solutions gave a weaker v-velocity field than that of mesh1 with the value range decreasing from $[-42.1m/s$ - $42.1m/s]$ to $[-25.1m/s$ - $25.1m/s]$. The w-velocity contour plots are shown in Figures 7.40, 7.41 and 7.42 for the Z-plane, Y-plane and X-plane respectively. Again the $vb - cc$ w-velocity contour plots are comparable with those obtained on mesh 1 with the value range approximately remaining unchanged. The cc solutions gave a much weaker w-velocity field and failed to resolve the maximum values along the leading edge of the wing. The w-velocity value range reduced from $[-4.89m/s$ - $44.5m/s]$ to $[-4.83m/s$ - $19.7m/s]$. The turbulent viscosity contour plots, Figures 7.43, 7.44 and 7.45, show some smearing in values occurring for both solution methods. The $vb - cc$ solutions of mesh 2 and mesh 3 show much higher turbulent viscosity values in the region downstream from the trailing edge of the wing. This is the result of higher velocity gradients in the region downstream of the wing which gives higher turbulent generation rates.

These results illustrate the ability of the $vb - cc$ method to handle distorted meshes. Even on the highly distorted mesh, mesh 3, there is only slight deterioration in the flow accuracy. Obtaining the flow field vertex-based, on the highly distorted mesh, enabled the solution of the turbulent quantities using cell-centred techniques. The non-orthogonality issues in the mesh affects the resolution of the turbulent quantities but enables solutions where purely cell-centred methods would fail. The cc method failed to handle the highly distorted mesh, mesh 3, and the flow solution was degraded on mesh 2.

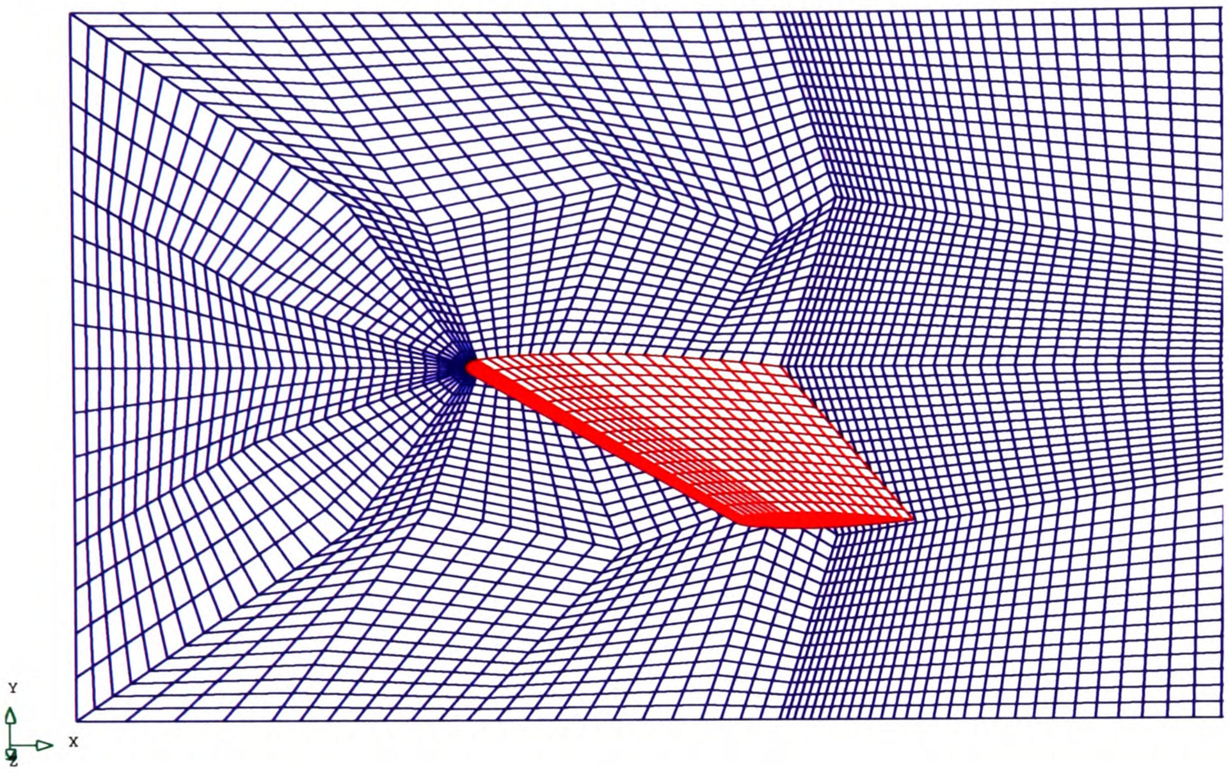


Figure 7.33: Distorted C-mesh, Mesh 2

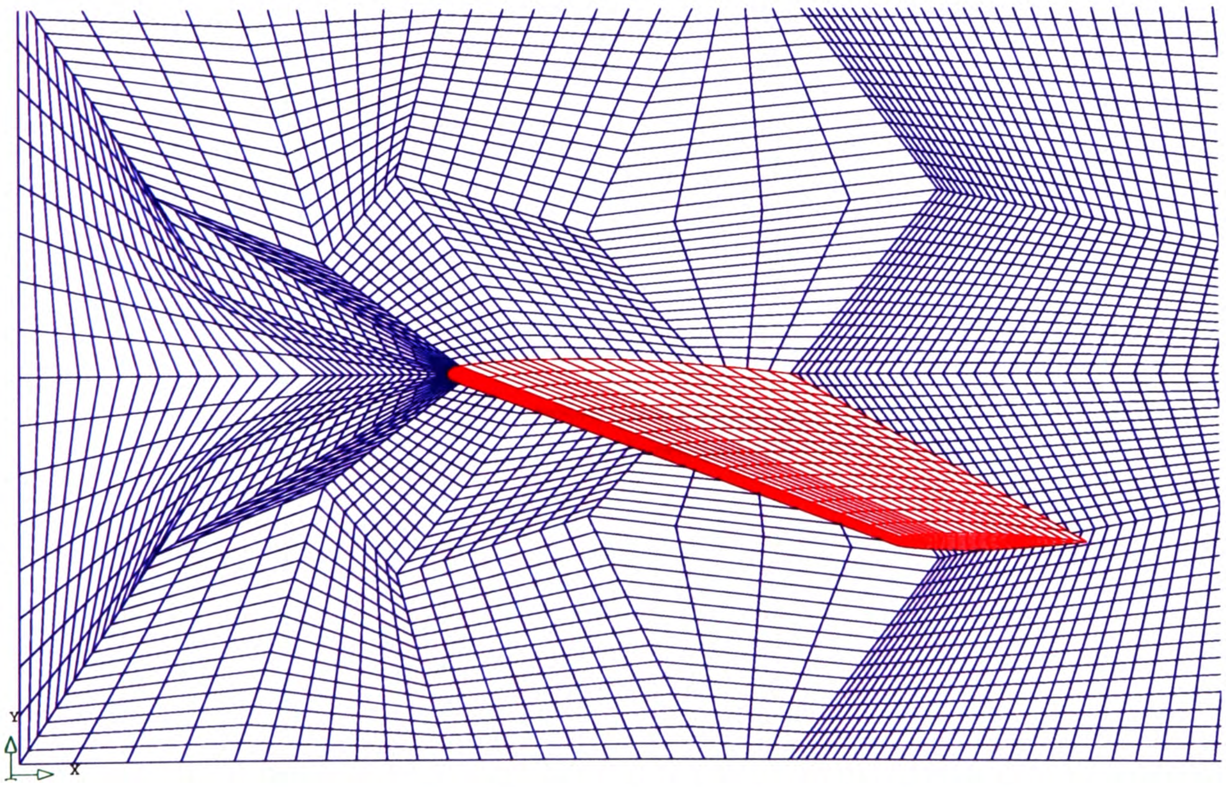


Figure 7.34: Distorted C-mesh, Mesh 3

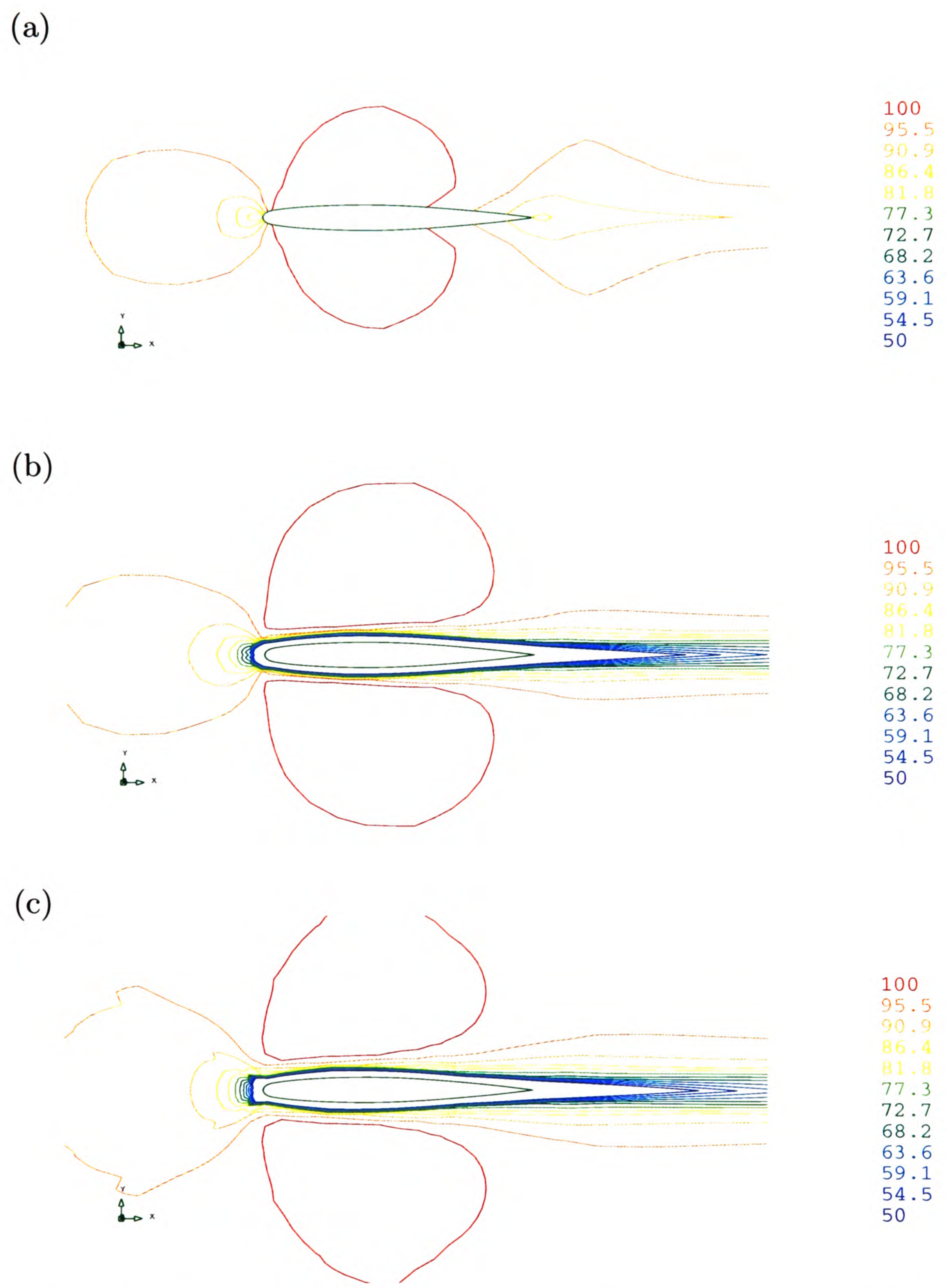


Figure 7.35: Z-plane: u-velocity contours, (a) *cc* mesh2, (b) $vb-cc$ mesh2, (c) $vb-cc$ mesh3

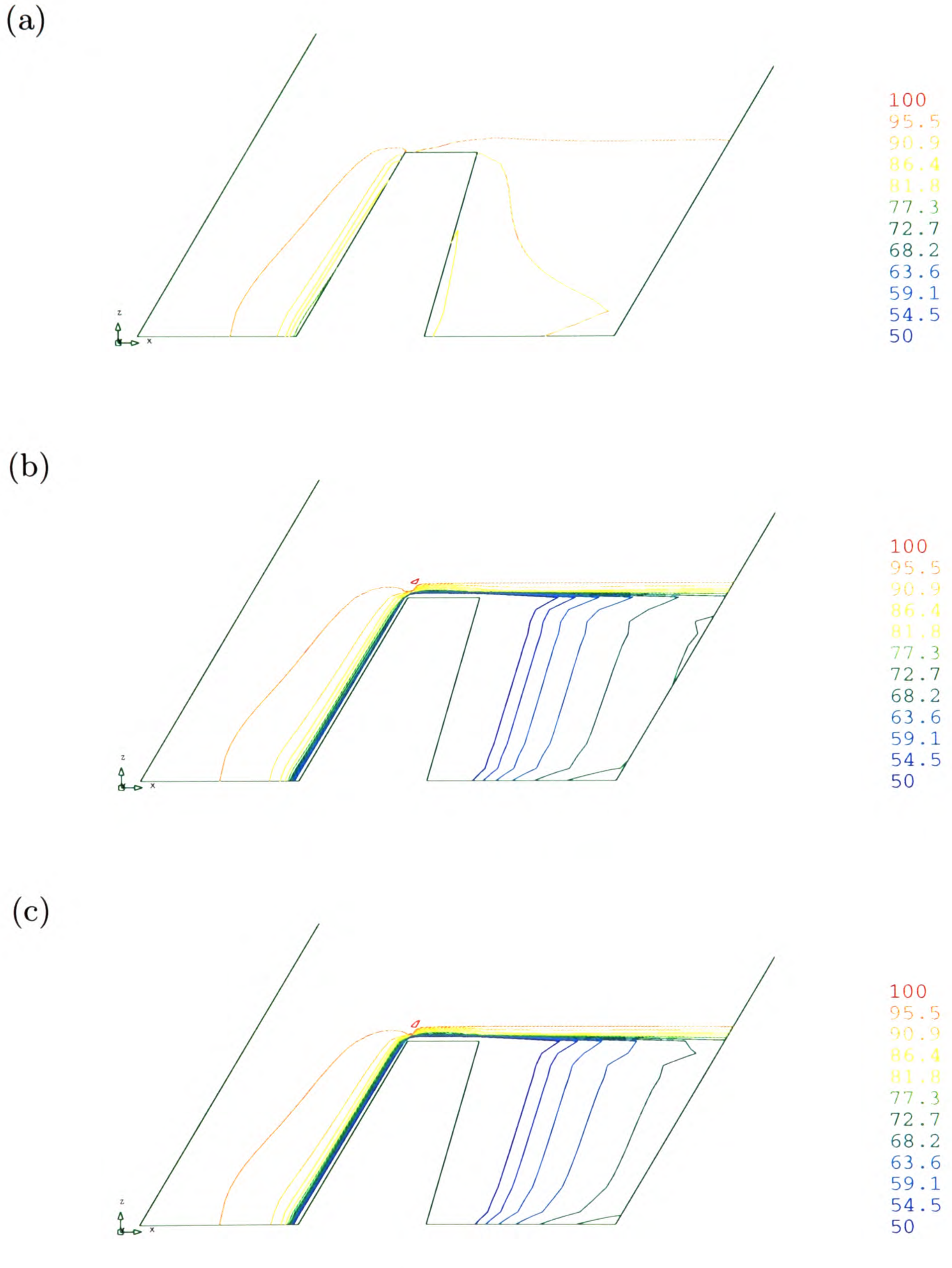


Figure 7.36: Y-plane: u-velocity contours, (a) *cc* mesh2, (b) *vb*−*cc* mesh2, (c) *vb*−*cc* mesh3

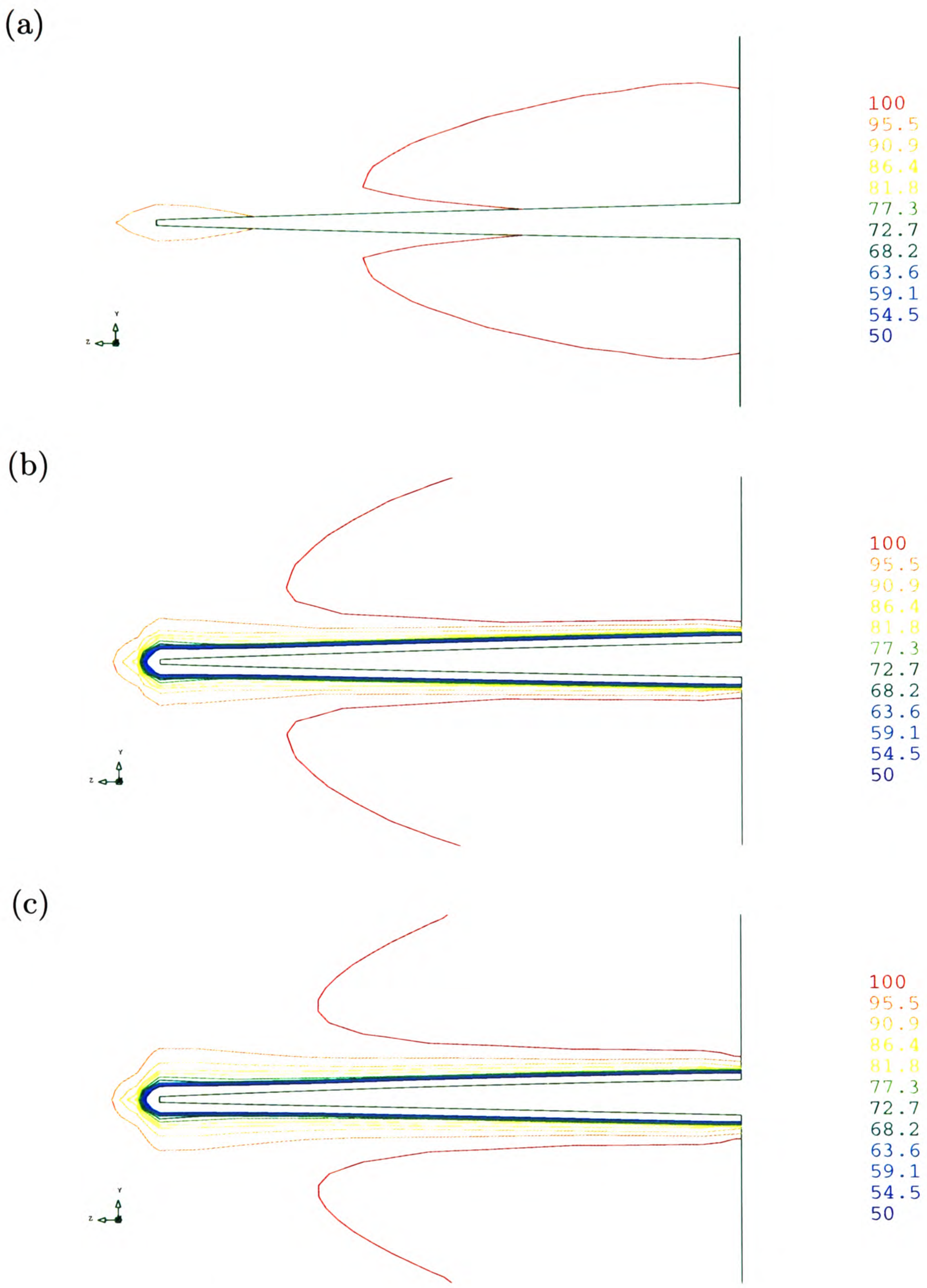


Figure 7.37: X-plane: u-velocity contours, (a) *cc* mesh2, (b) *vb*−*cc* mesh2, (c) *vb*−*cc* mesh3

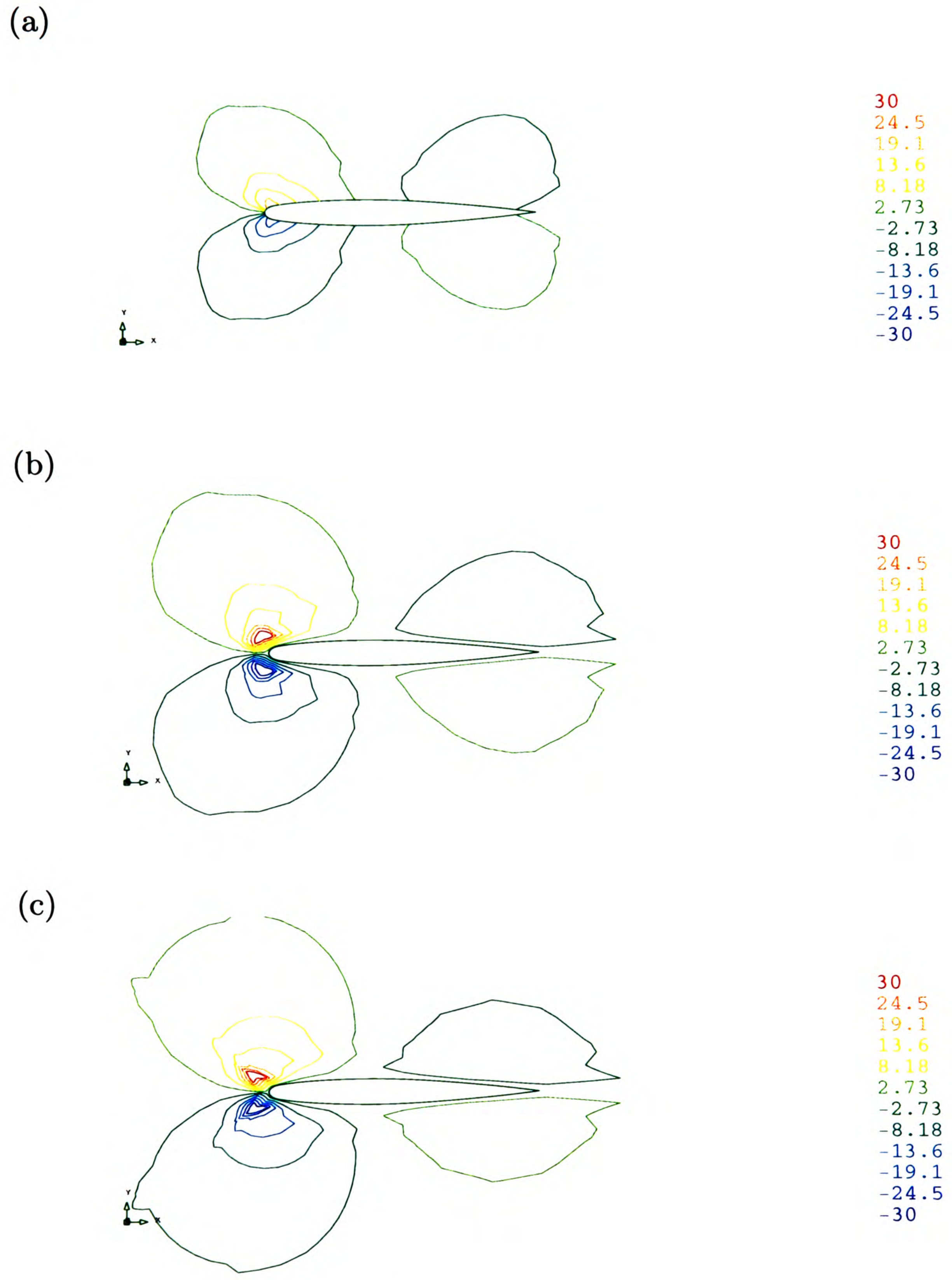


Figure 7.38: Z-plane: v-velocity contours, (a) *cc* mesh2, (b) $vb-cc$ mesh2, (c) $vb-cc$ mesh3

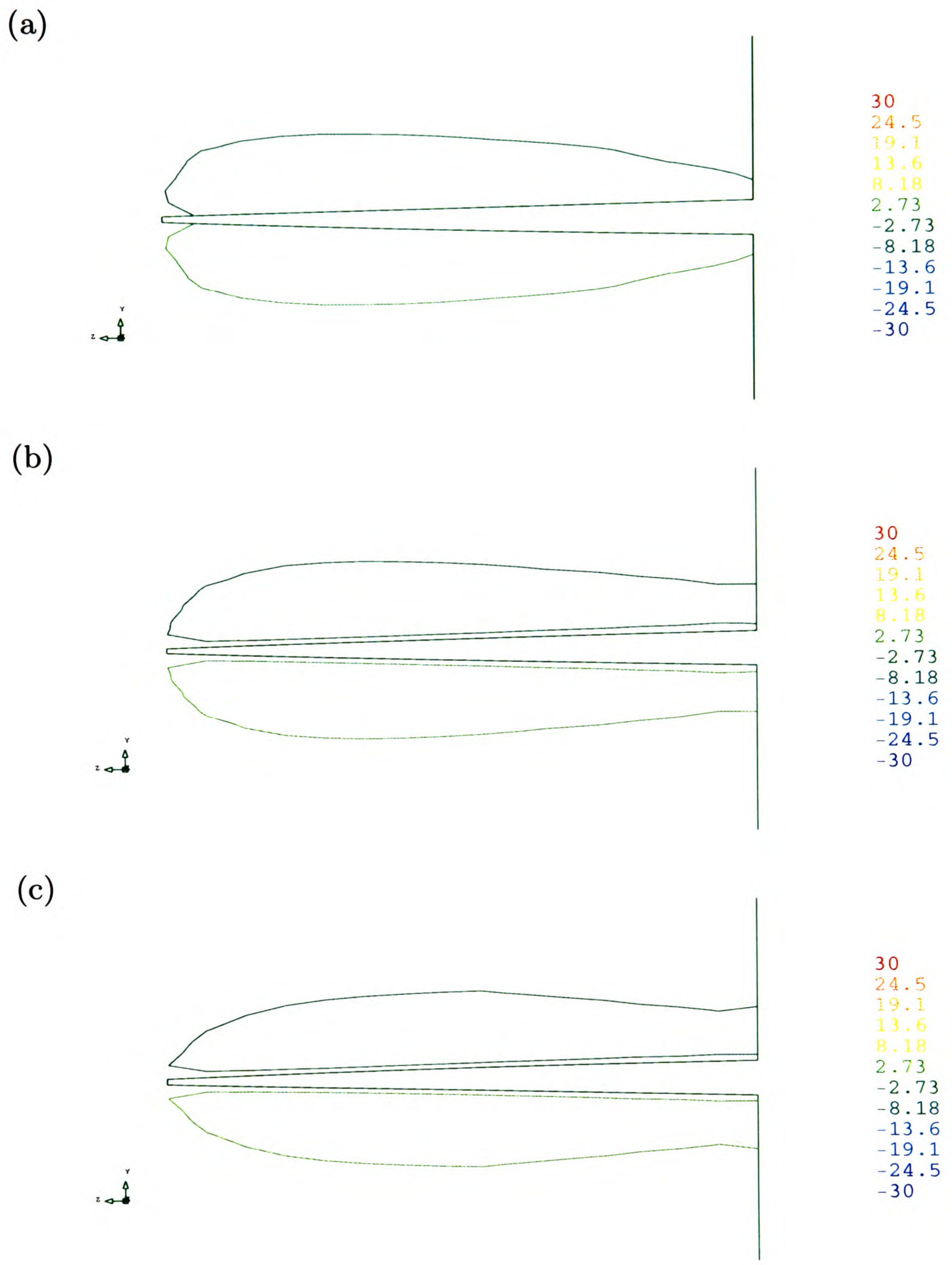


Figure 7.39: X-plane: v-velocity contours, (a) *cc* mesh2, (b) *vb−cc* mesh2, (c) *vb−cc* mesh3

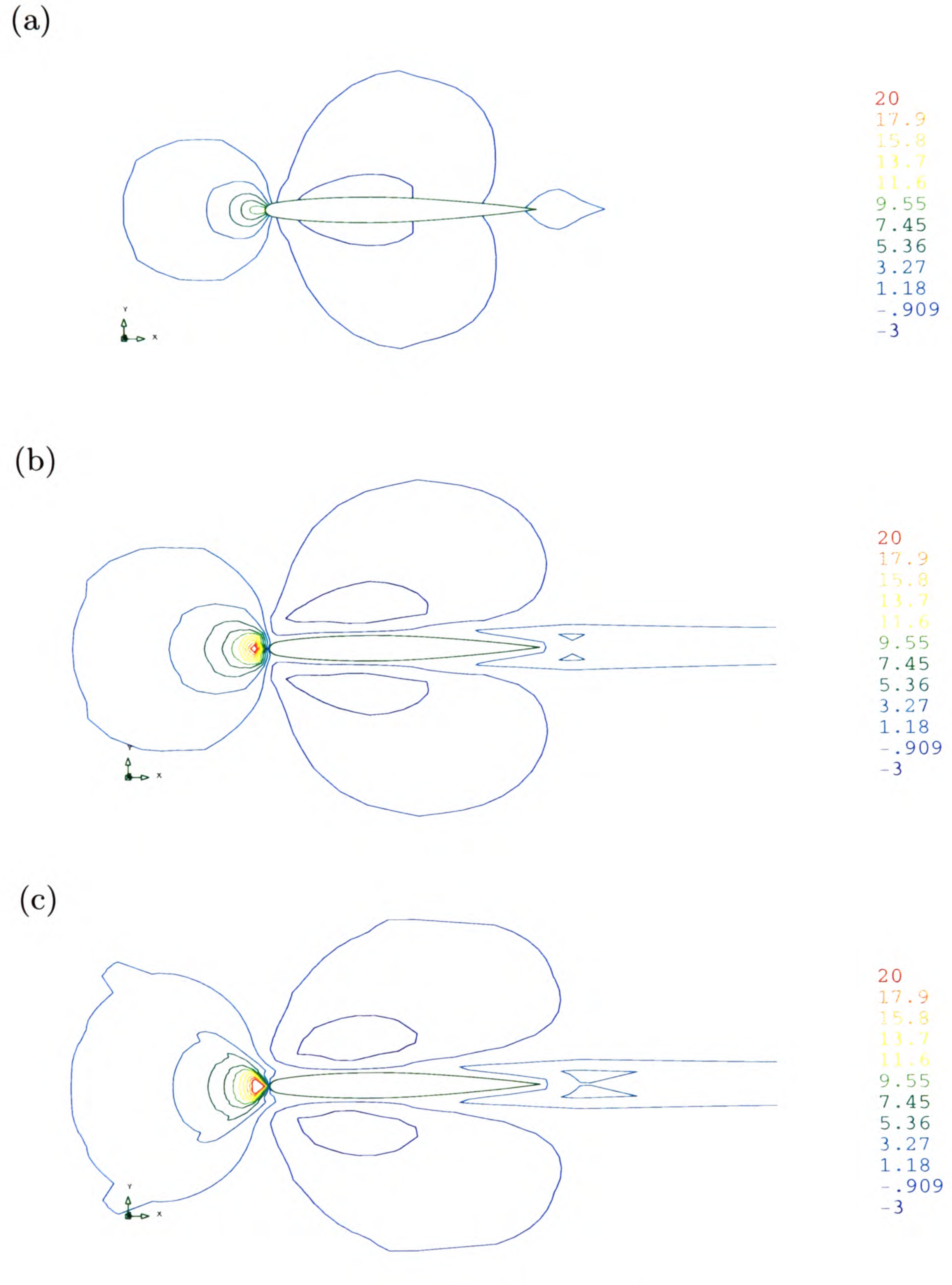


Figure 7.40: Z-plane: w-velocity contours, (a) *cc* mesh2, (b) $vb - cc$ mesh2, (c) $vb - cc$ mesh3

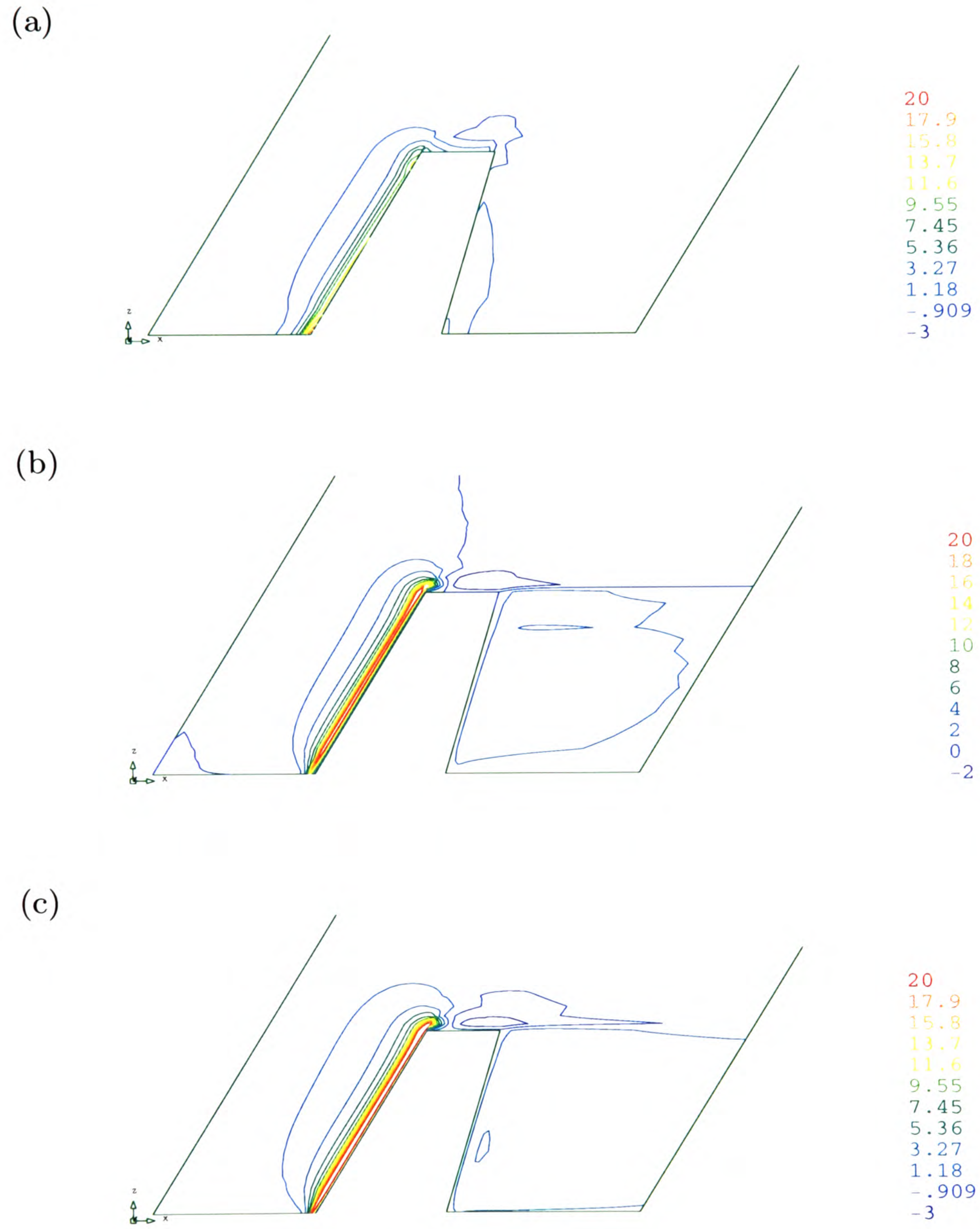


Figure 7.41: Y-plane: w-velocity contours, (a) *cc* mesh2, (b) $vb - cc$ mesh2, (c) $vb - cc$ mesh3

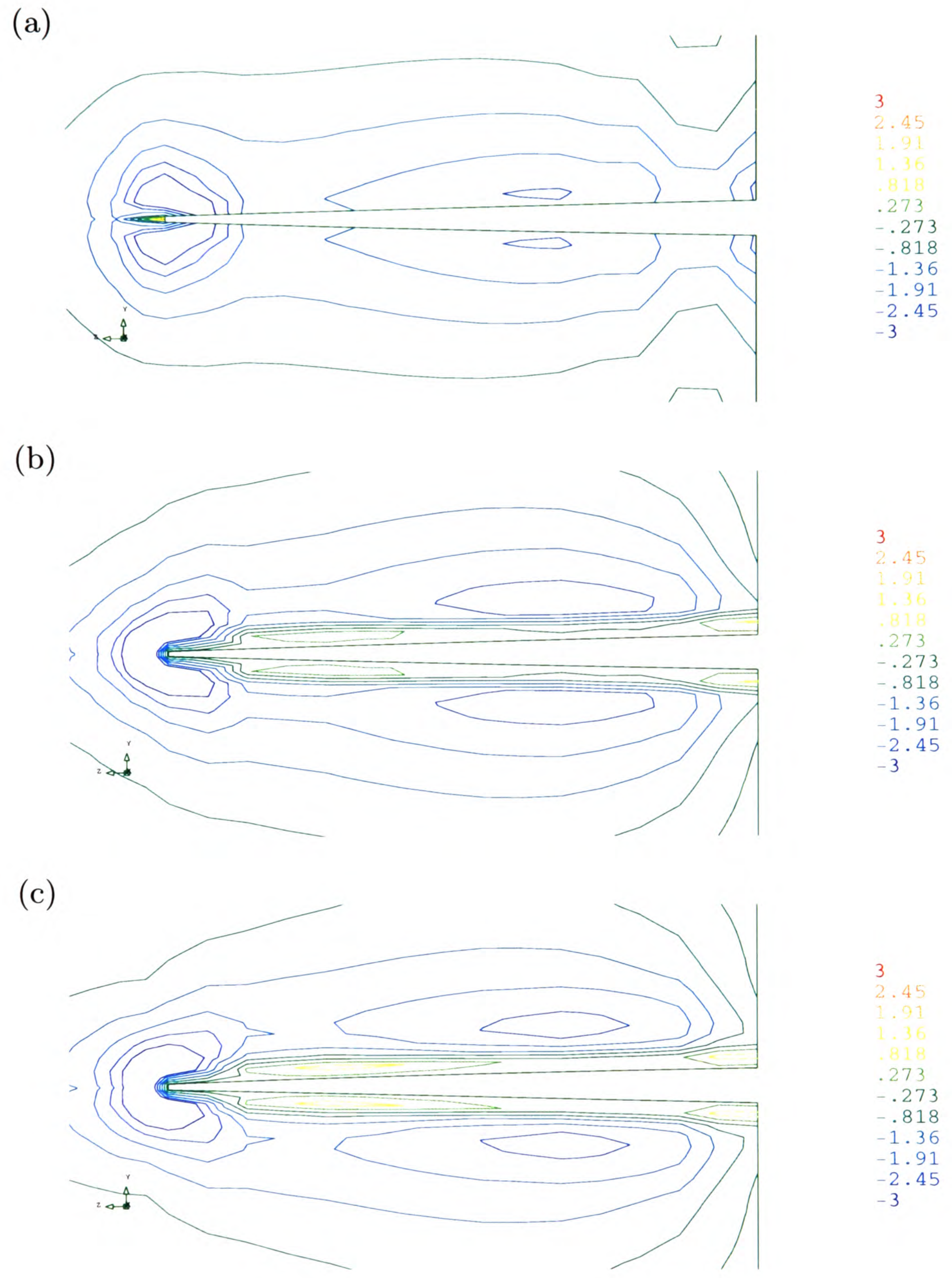


Figure 7.42: X-plane: w-velocity contours, (a) *cc* mesh2, (b) $vb - cc$ mesh2, (c) $vb - cc$ mesh3

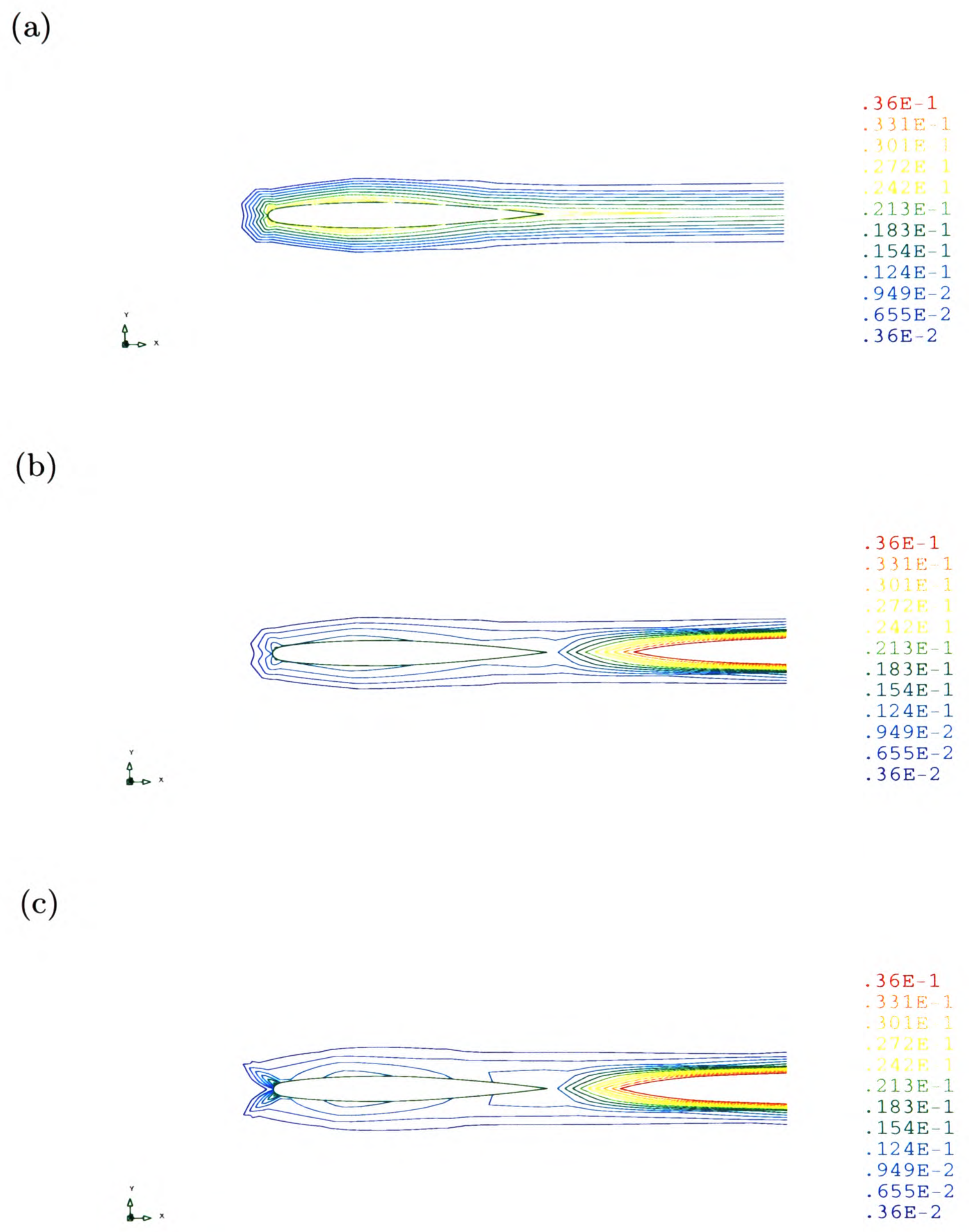


Figure 7.43: Z-plane: turbulent viscosity contours, (a) *cc* mesh2, (b) $vb - cc$ mesh2, (c) $vb - cc$ mesh3

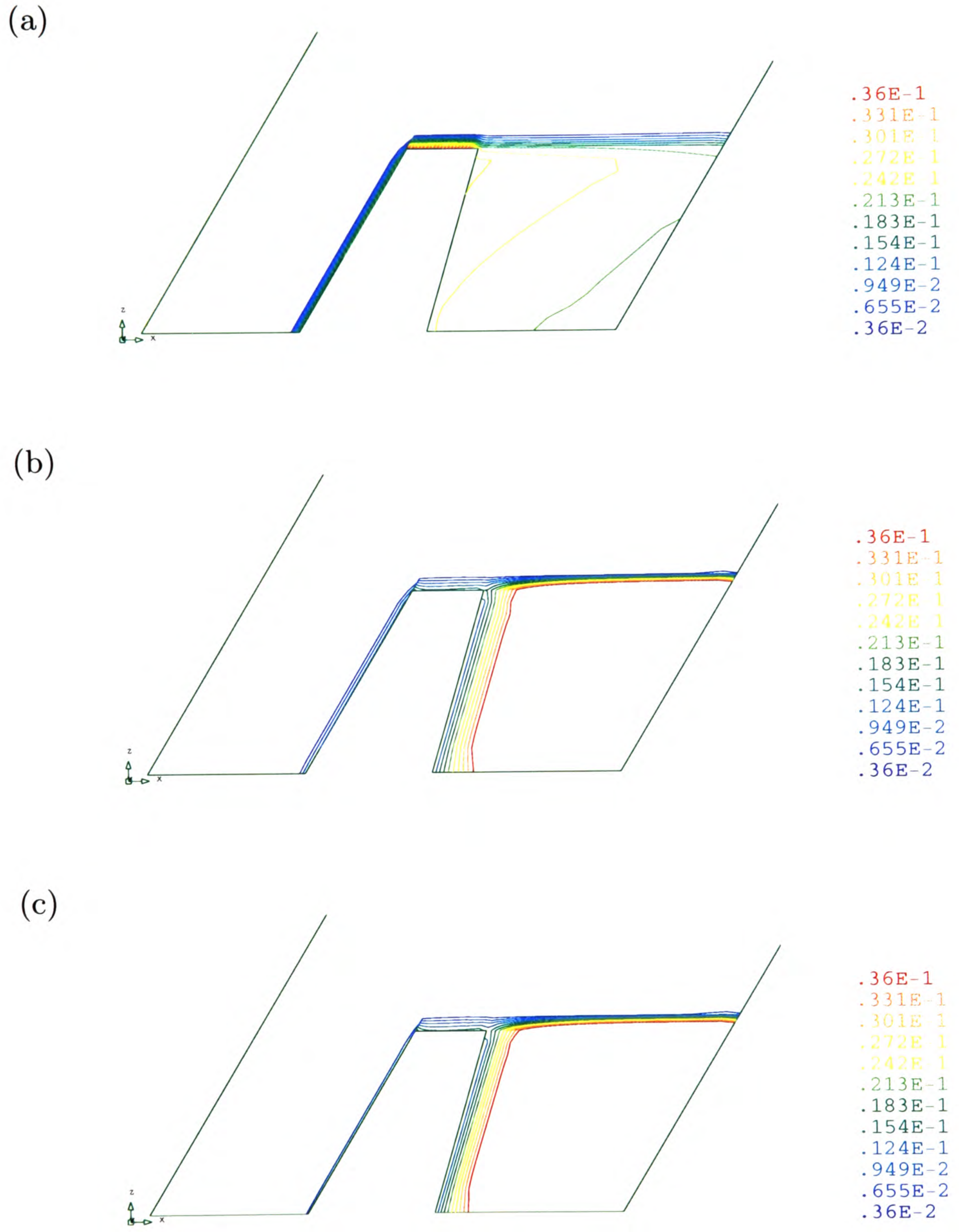


Figure 7.44: Y-plane: turbulent viscosity contours, (a) *cc* mesh2, (b) $vb - cc$ mesh2, (c) $vb - cc$ mesh3

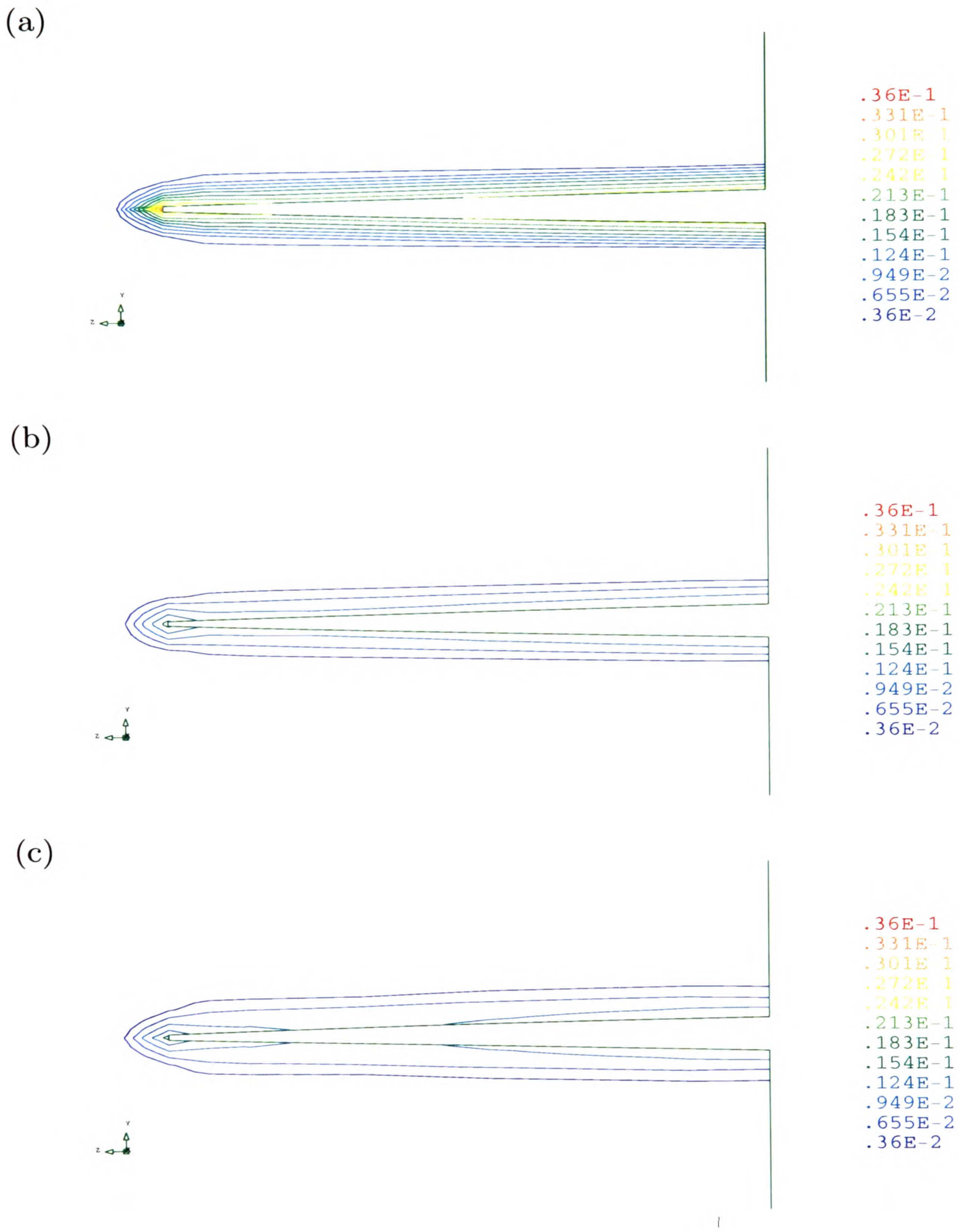


Figure 7.45: X-plane: turbulent viscosity contours, (a) *cc* mesh2, (b) $vb - cc$ mesh2, (c) $vb - cc$ mesh3

7.3.4 Wing at 10^0 angle of attack

Employing the uniform C-Mesh, Figure 7.20, the three-dimensional wing was rotated within the mesh by varying degrees. The initial and boundary conditions remained unchanged. Vortices should form on the upper surface of the wing and periodically be shed in the wake of the wing. The frequency that vortices are shed is often measured by the Strouhal number (S_t), a dimensionless parameter,

$$S_t = \frac{\omega l}{V} \tag{7.24}$$

where the characteristic time is the inverse of ω, the wake shedding frequency behind a separated airfoil. l is the mean aerodynamic chord and V is the free-stream velocity.

Vortices are typically shed at $S_t \approx 0.2$. For the flow speed being investigated vortex shedding should occur approximately every 0.03 seconds. In an attempt to identify transient fluctuations, simulations were performed employing a time step of 0.0015 seconds giving 20 time steps per vortex period. Various angles of attack up to an angle of 10 degrees were investigated, for both the $vb - cc$ and cc methods no transient perturbation of the solution was observed. This is probably due to the coarseness of the mesh. Both the $vb - cc$ and cc solution methods use upwind differencing of the convection term which is known to suffer from numerical diffusion. Numerical diffusion could be reduced by refining the mesh and employing a higher order differencing scheme.

The solutions of steady state time averaged flow past a wing at 10 degree angle of attack are shown in this section. The rotation of the wing within the mesh is an example of a practical application in which a mesh can become distorted. Figure 7.46 shows a cross section of the C-mesh after the wing has been rotated by 10 degrees.

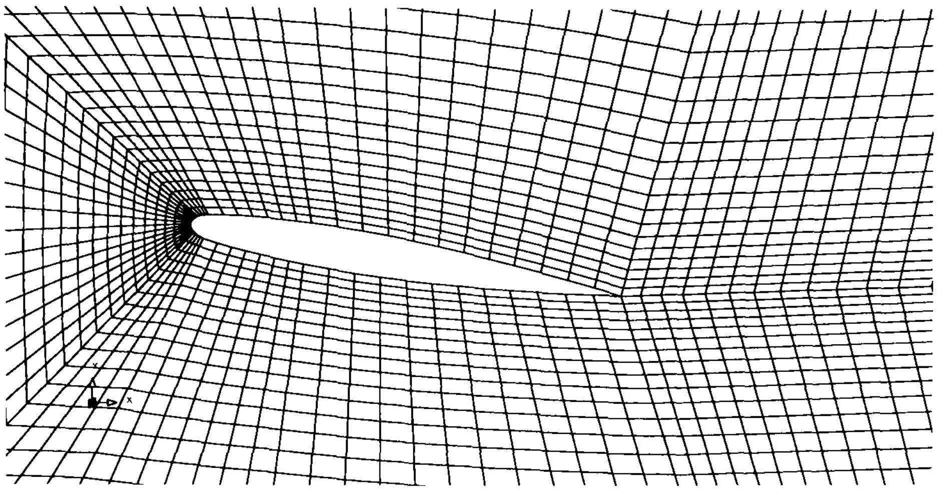


Figure 7.46: C-mesh, Wing rotated by 10 degrees

Velocity and turbulent viscosity contours are shown in Figures 7.47 to 7.59. Contour plots are shown at cross-sections Z-plane, Y-plane and X-plane defined in the previous section. The u-velocity contour plots are shown in Figures 7.47, 7.48 and 7.49. The $vb-cc$ results show lower u-velocity values downstream of the wing, markedly in the region downstream of the wing tip. The cc minimum u-velocity value of $14.7m/s$ was obtained downstream of the wing mid-section, whereas, the cc values adjacent to the wing tip were around $70m/s$ to $90m/s$. A maximum u-velocity of $126m/s$ was obtained for both methods. The v-velocity contour plots are shown in Figures 7.50, 7.51 and 7.52 for the Z-plane, Y-plane and X-plane respectively. The v-velocity plots are similar for both methods with the cc method giving a slightly larger v-velocity range of $[-26.5m/s$ - $96.5m/s]$ compared to $vb-cc$ values of $[-27.0m/s$ - $89.4m/s]$. The w-velocity contour plots are shown in Figures 7.53, 7.54 and 7.55 for the Z-plane, Y-plane and X-plane respectively. The cc plots show a stronger w-velocity field in the region adjacent to the upper wing surface and trailing edge of the wing. The cc w-velocity range was $[-20.8m/s$ - $55.5m/s]$ compared to $vb-cc$ values of $[-19.3m/s$ - $41.5m/s]$. The $vb-cc$ lower velocity values downstream of the wing tip and trailing edge lead to higher velocity gradients and hence higher turbulent generation rates over a larger region than cc solutions, see Figure 7.56. This in turn leads to higher $vb-cc$ turbulent viscosity adjacent to the upper wing surface and downstream of the wing tip. Figures 7.57, 7.58 and 7.59 show the turbulent viscosity contours for the Z-plane, Y-plane and X-plane respectively.

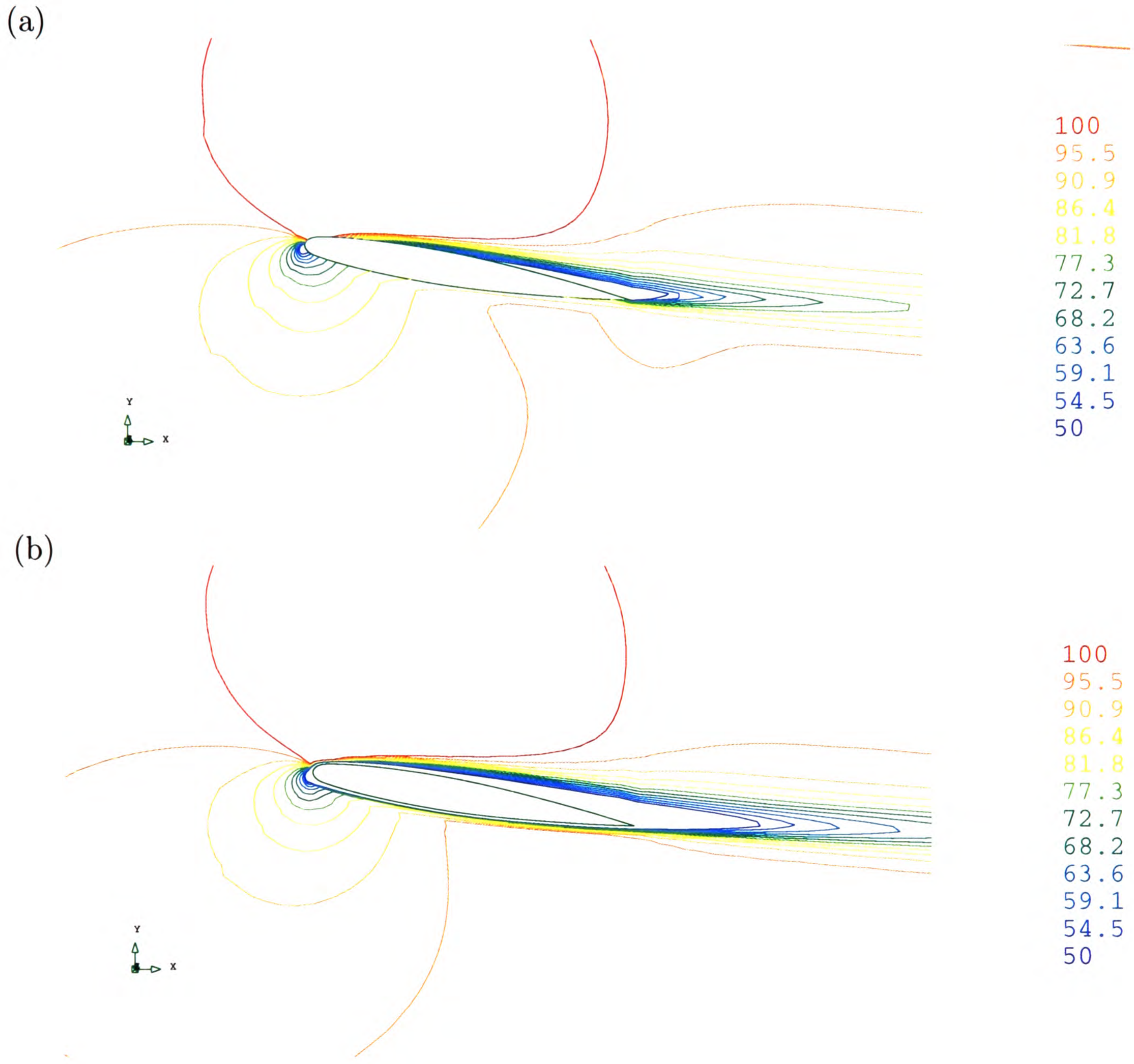


Figure 7.47: Wing at 10^0 angle of attack: Z-plane: u-velocity contours, (a) cc, (b) $vb - cc$

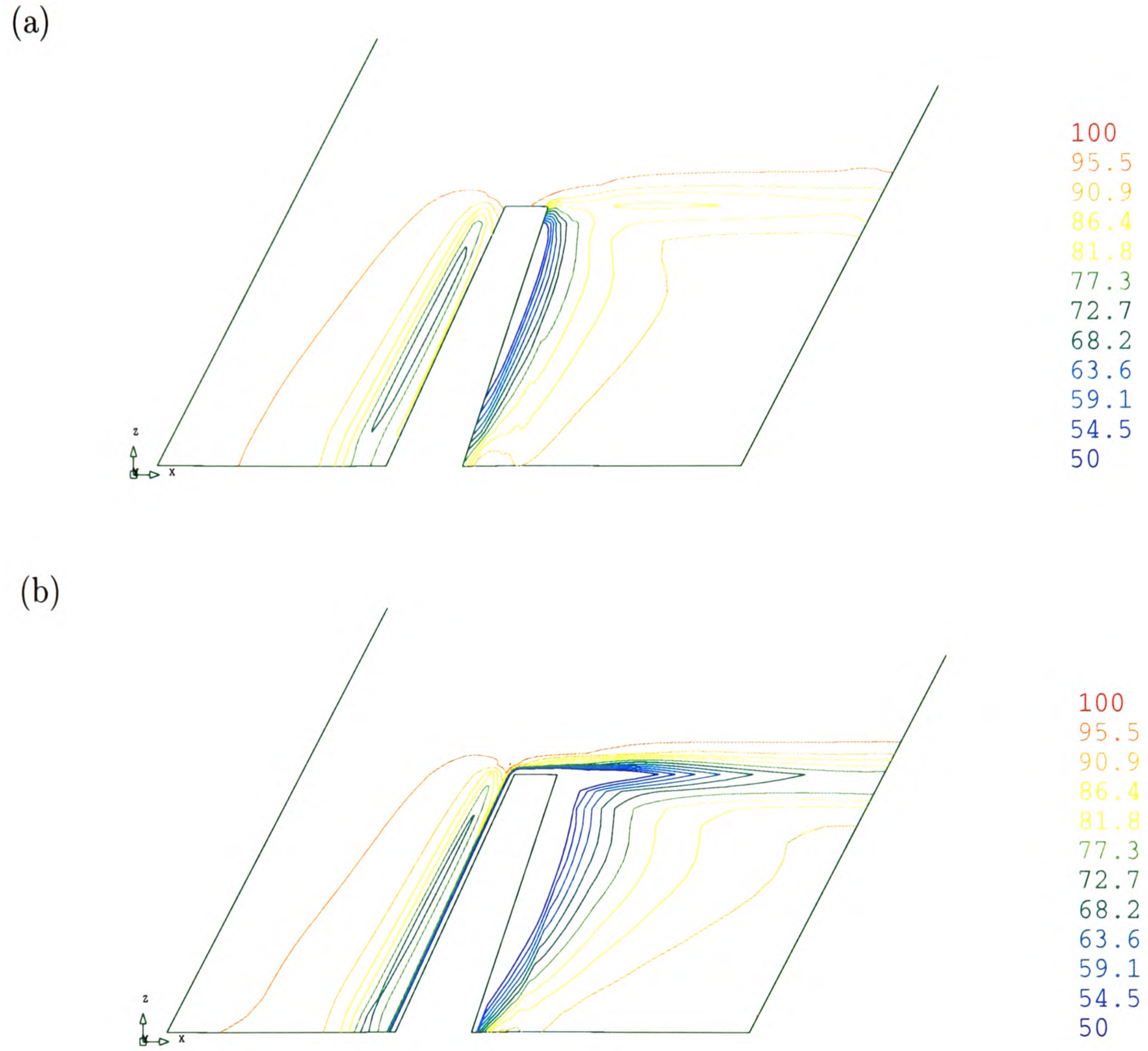


Figure 7.48: Wing at 10^0 angle of attack: Y-plane: u-velocity contours, ((a) *cc*, (b) $vb - cc$

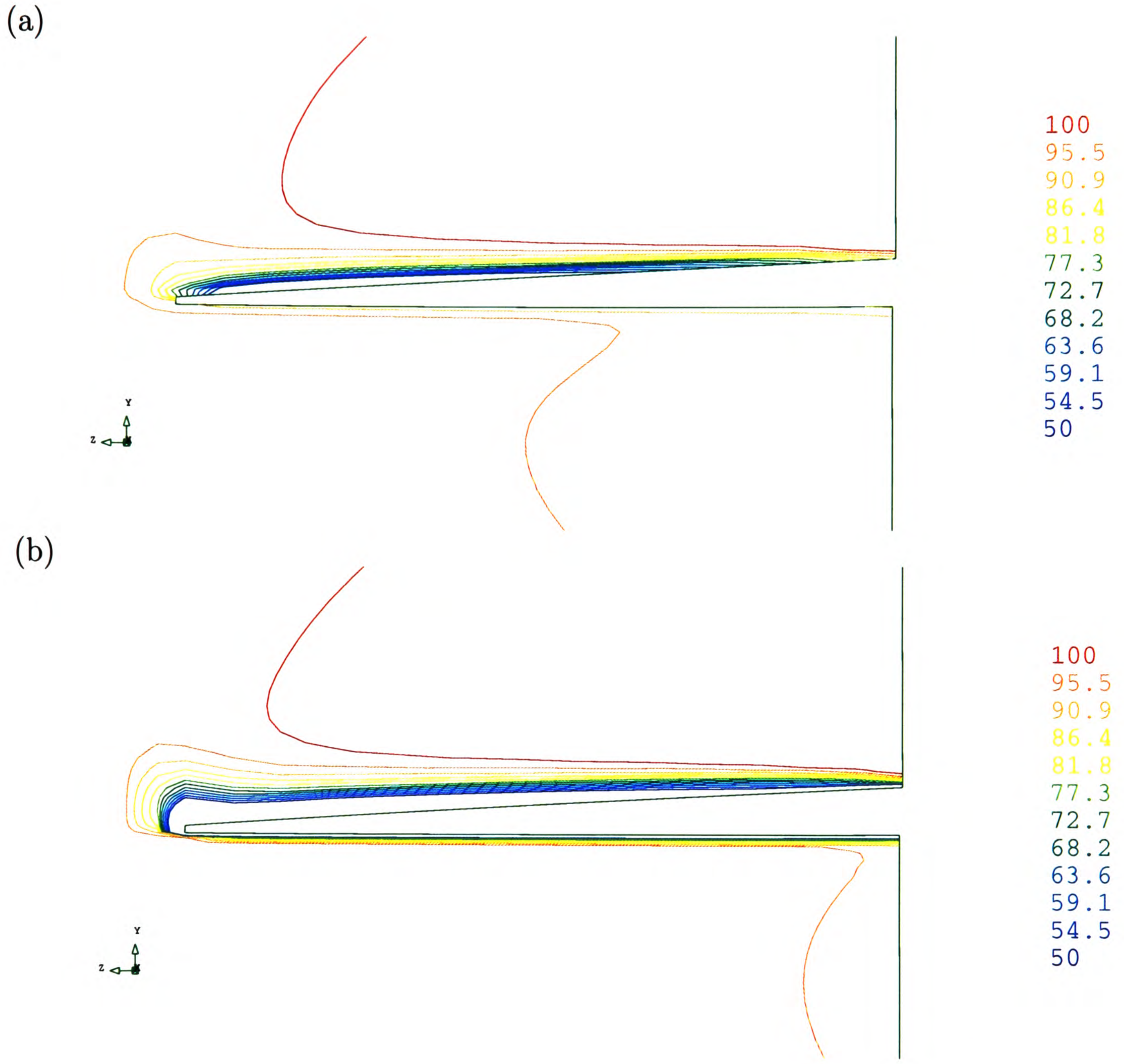


Figure 7.49: Wing at 10^0 angle of attack: X-plane: u-velocity contours, (a) *cc*, (b) $vb - cc$

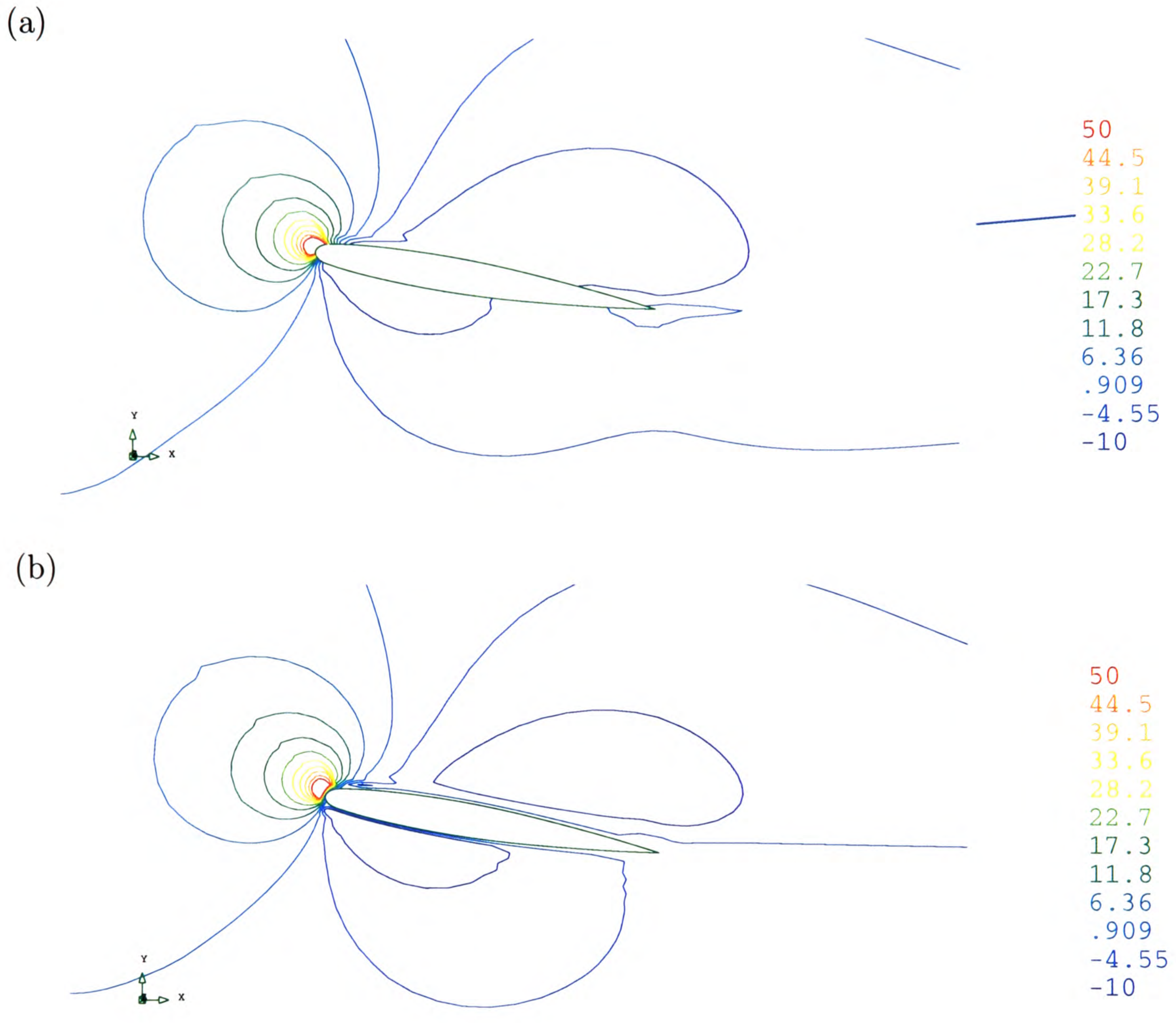


Figure 7.50: Wing at 10^0 angle of attack: Z-plane: v-velocity contours, (a) *cc*, (b) $vb - cc$

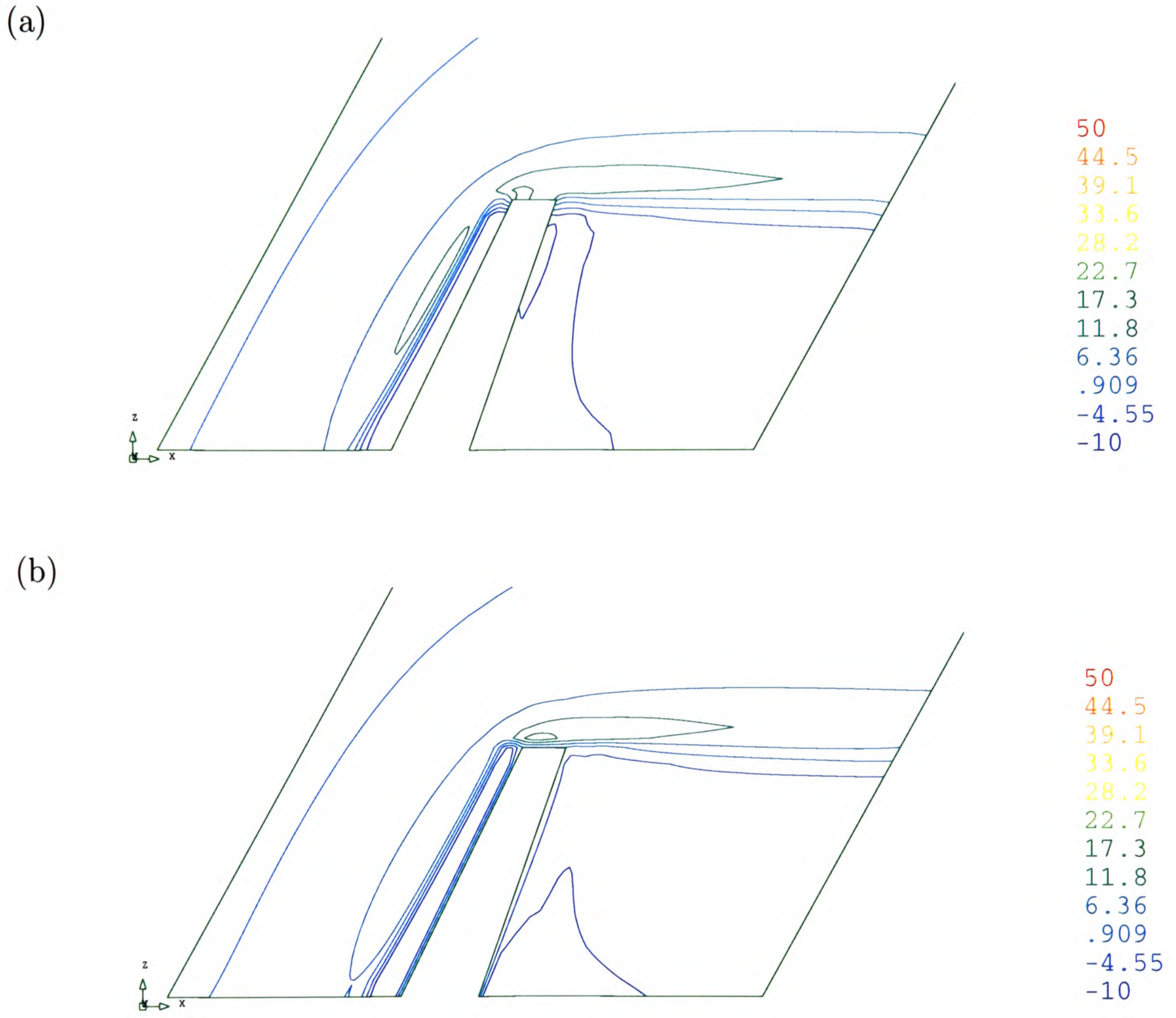


Figure 7.51: Wing at 10^0 angle of attack: Y-plane: v-velocity contours, (a) *cc*, (b) $vb - cc$

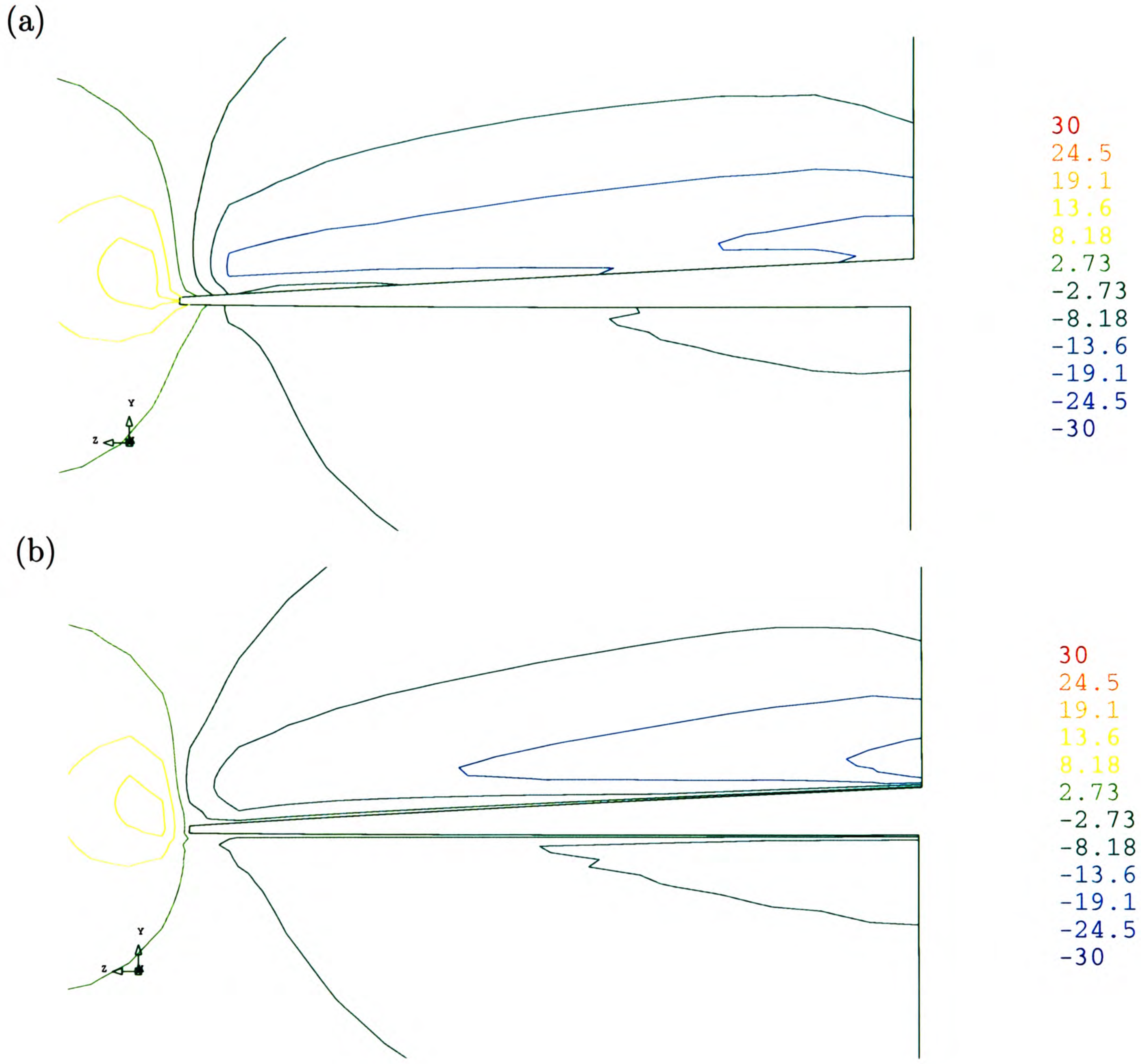


Figure 7.52: Wing at 10^0 angle of attack: X-plane: v-velocity contours, (a) *cc*, (b) $vb - cc$

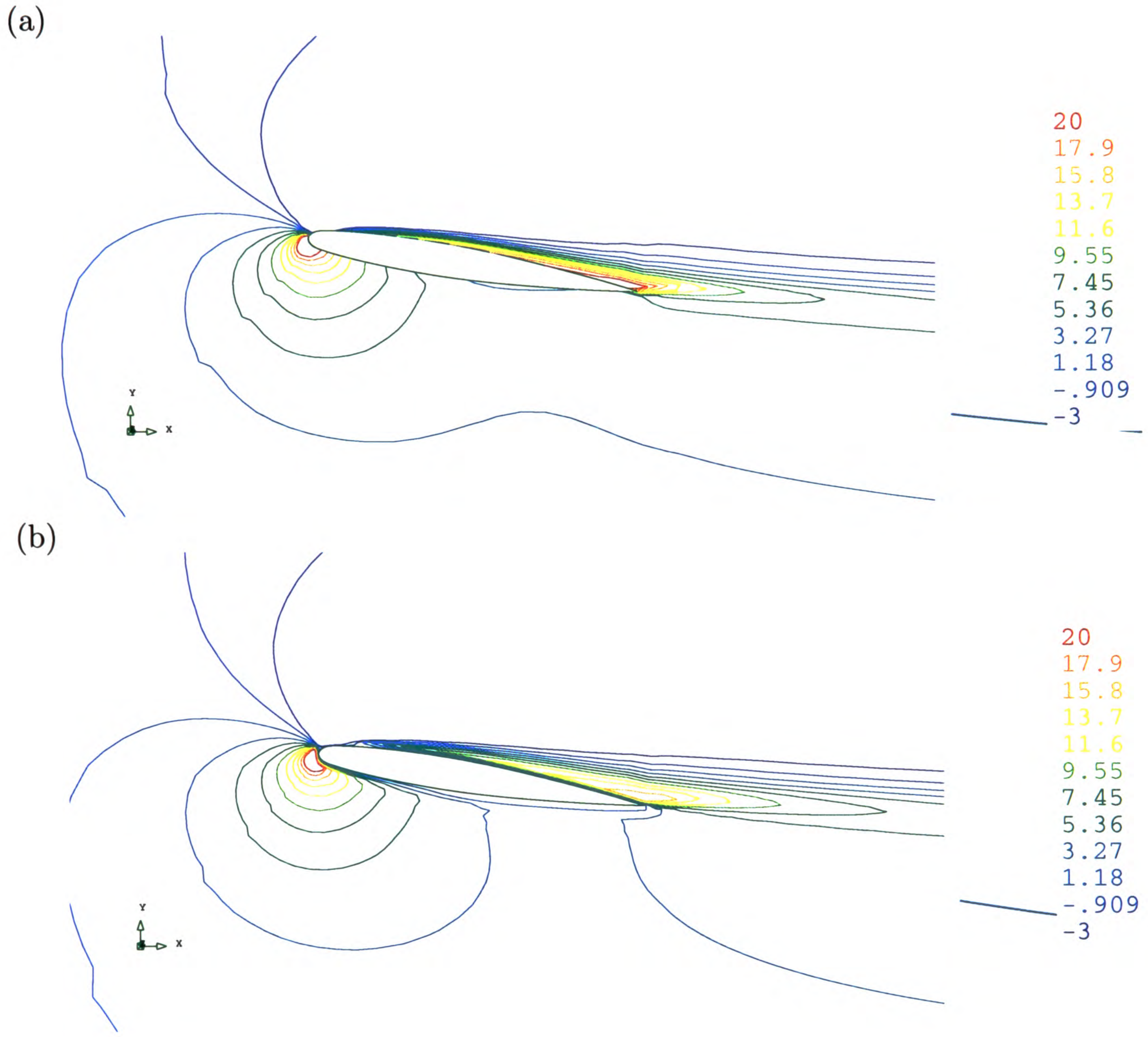


Figure 7.53: Wing at 10^0 angle of attack: Z-plane: w-velocity contours, (a) cc, (b) $vb - cc$

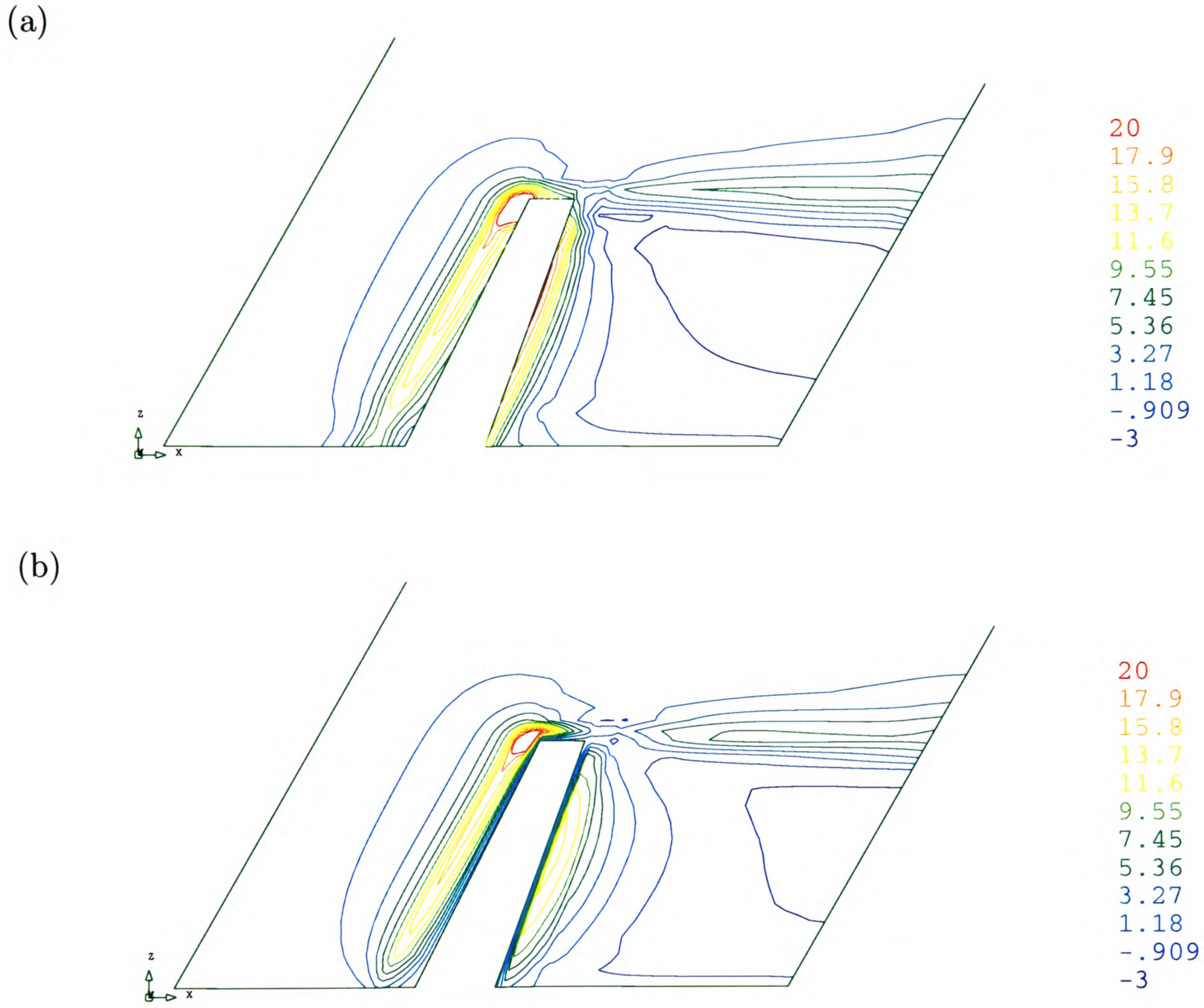


Figure 7.54: Wing at 10^0 angle of attack: Y-plane: w-velocity contours, (a) *cc*, (b) $vb - cc$

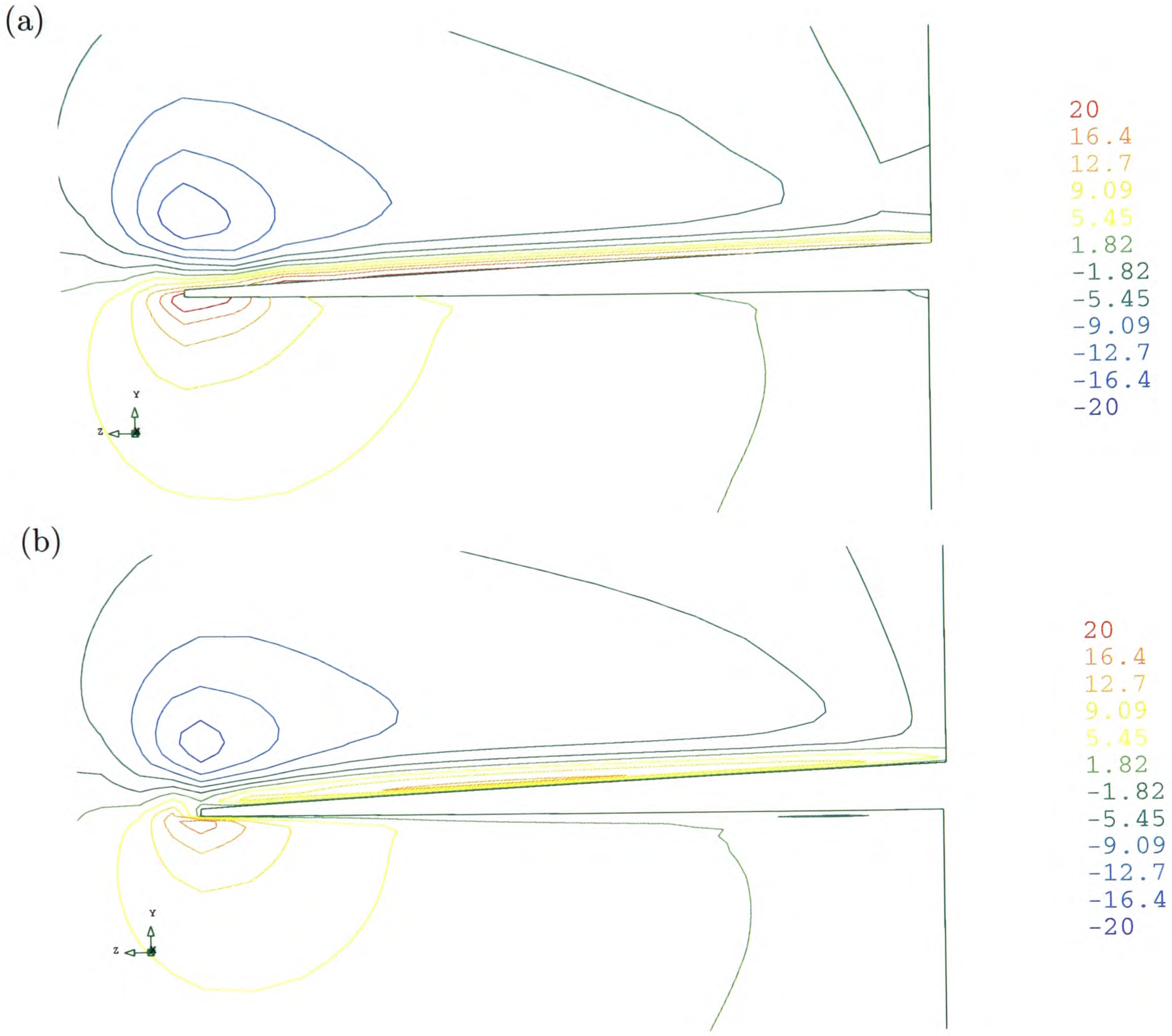


Figure 7.55: Wing at 10^0 angle of attack: X-plane: w-velocity contours, (a) cc, (b) $vb - cc$

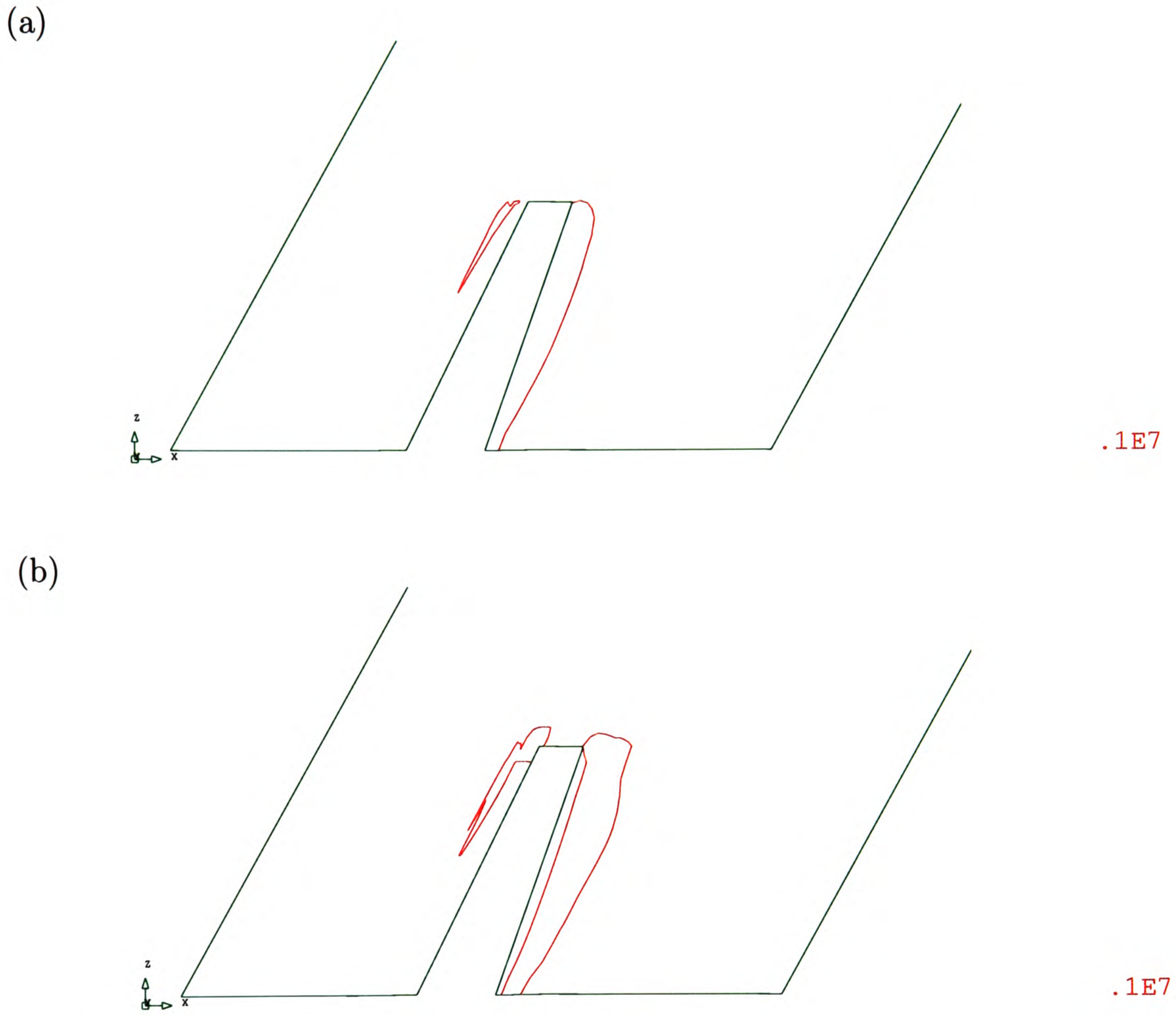


Figure 7.56: Wing at 10^0 angle of attack: Y-plane: turbulent generation rate contours, (a) *cc*, (b) $vb - cc$

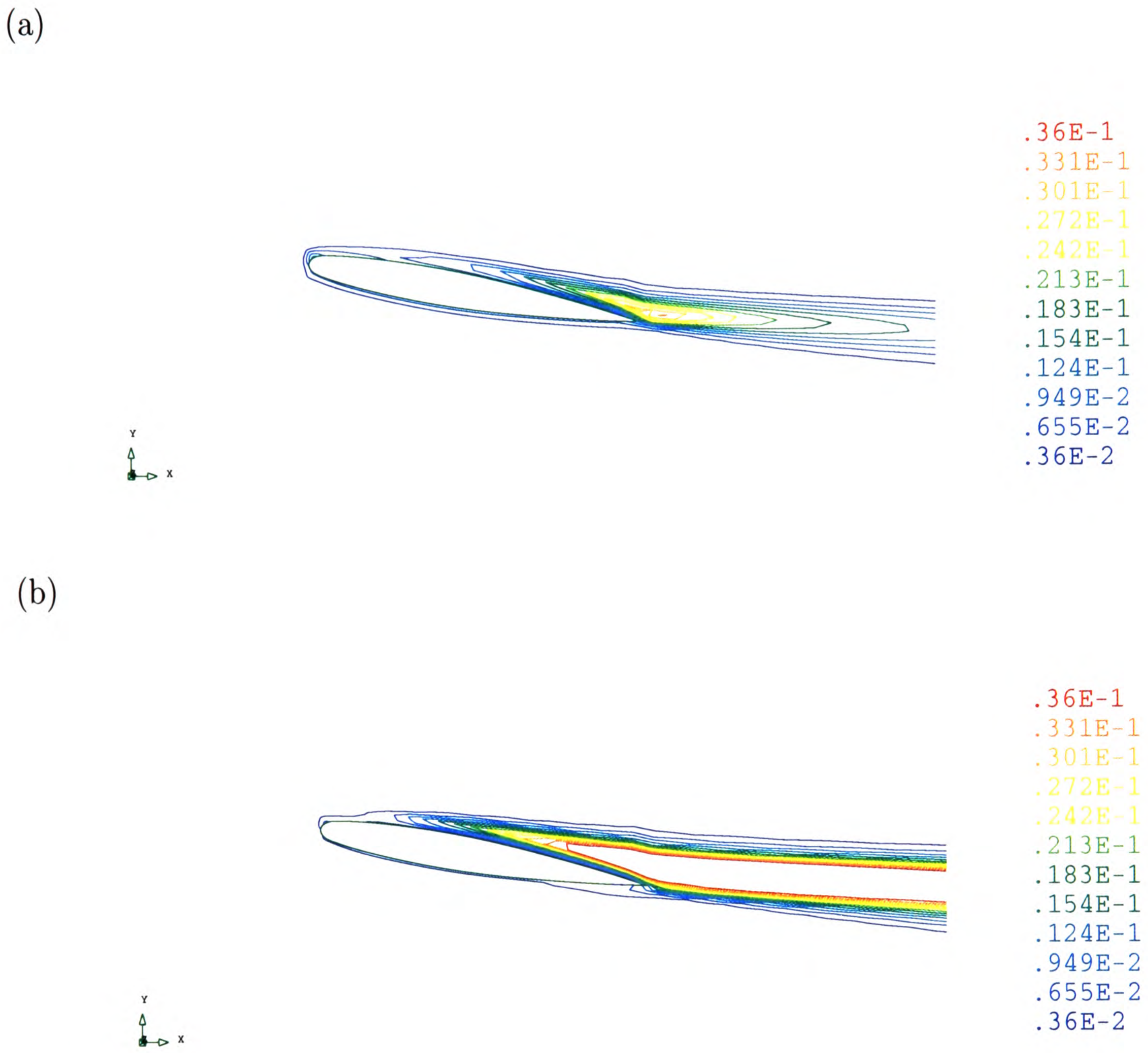


Figure 7.57: Wing at 10^0 angle of attack: Z-plane: turbulent viscosity contours, ((a) cc, (b) $vb - cc$

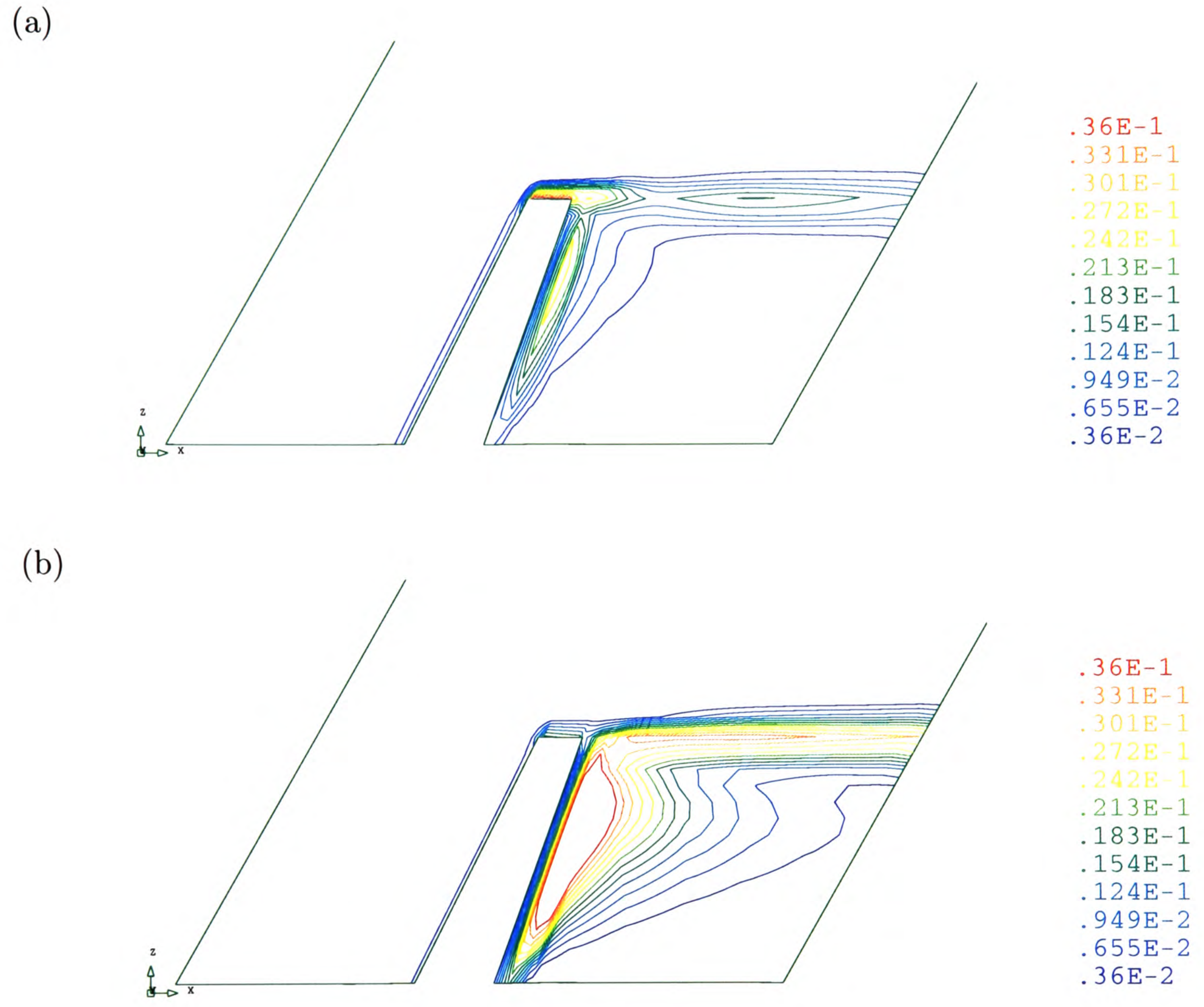


Figure 7.58: Wing at 10^0 angle of attack: Y-plane: turbulent viscosity contours, (a) cc, (b) $vb - cc$

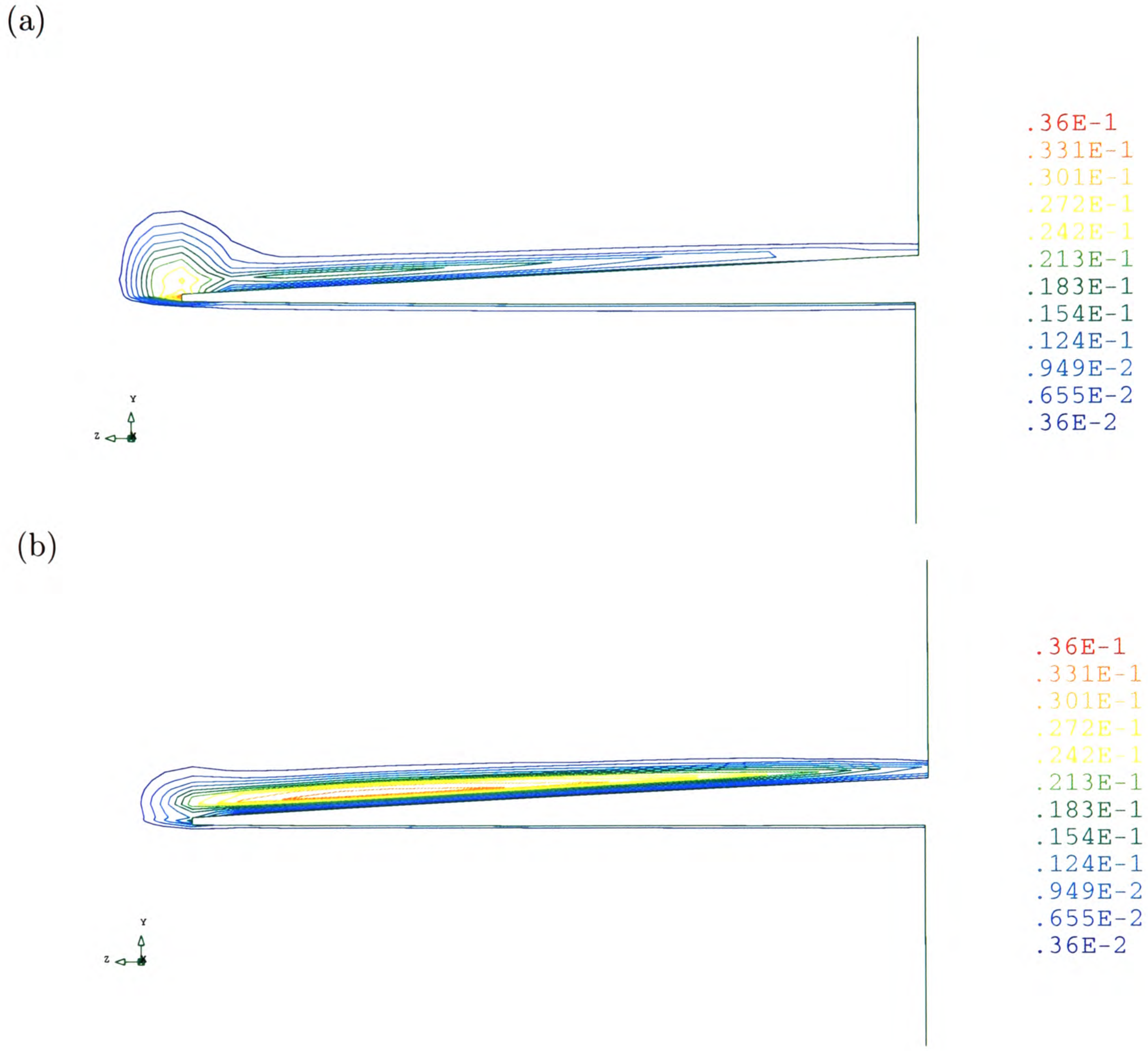


Figure 7.59: Wing at 10^0 angle of attack: X-plane: turbulent viscosity contours, (a) cc, (b) $vb - cc$

7.3.5 Run Time and Memory Requirements

The simulations were performed on a Pentium 4 CPU 2.53GHz. The mesh employed comprised of 101,412 elements and 108,314 nodes. To achieve the convergence criteria that the $L2$ norm of the change in the solution dropped by 5 orders-of-magnitude on the uniform C-mesh, the $vb-cc$ method required 254 iteration and the cc method required 370 iterations. The simulation time was approximately 66.5 minutes for the $vb-cc$ method and 26.27 minutes for the cc method. This gives the time required per iteration/per solution point to be 3.3×10^{-5} seconds and 7.0×10^{-6} seconds for $vb-cc$ and cc solutions respectively. On the distorted mesh the number of iterations required to achieve convergence increased to 302 for the $vb-cc$ method but remained approximately the same for cc solutions. The $vb-cc$ method required considerably more memory, 162.25 megabytes compared to 24.37 megabytes for cc solutions. This gives a memory requirement per solution point of 373 bytes and 42 bytes for $vb-cc$ and cc solutions respectively.

7.4 Closure

The vertex-based method has been extended to allow the solution of turbulent flow problems using the $k-\epsilon$ turbulence model. Vertex-based discretisation of the flow variables has been successfully coupled with cell-centred discretisation of the transported variables k and ϵ. The method has been validated on a case involving turbulent flow over a backward facing step for a number of Reynolds numbers. The results obtained $vb-cc$ were in good agreement with purely cell-centred solutions on the 5200 element mesh. The reattachment length was well within the expected range of $5.8h$ to $6.1h$ for the standard $k-\epsilon$ model on the fine mesh. On a coarser 1400 element mesh the $vb-cc$ method gave a reattachment length of $5.66h$ which is only slightly outside the expected range. Purely cell-centred solutions under predicted the value on the coarse mesh, giving a reattachment length of $5h$.

The $vb - cc$ method was extended to solve three-dimensional turbulent flow. Simulations were performed on flow over an aircraft wing, $vb - cc$ and cc solutions were compared. The differences in the simulation results have been investigated. In the $vb - cc$ method the zero velocity wall condition has a greater influence on neighbouring solution points leading to lower velocity values adjacent to and downstream of the wing. This in turn creates greater velocity gradients and hence a larger turbulent generation rate. Consequently, the turbulent viscosity values downstream are higher in the $vb - cc$ simulations than in the cc simulations. On the finer mesh the $vb - cc$ and cc velocity profile adjacent to the wing surface are in closer agreement, but the $vb - cc$ method still gives lower velocity values in the wake region.

Chapter 8

Conclusions and Suggestions for Future Work

During this research project a finite volume vertex-based discretisation technique has been investigated and applied to convection-diffusion problems and fluid flow. The vertex-based method has been applied to and compared with the cell-centred discretisation method on a variety of meshes and flow problems. Hybrid flow solution methods have been investigated in an attempt to capture the best aspects of both methods. The conclusions drawn from this investigation are summarised in the following paragraphs and suggestions for possible future work are given.

8.1 Concluding Remarks

The cell-centred discretisation technique has been shown to produce expeditious and reliable solutions on orthogonal meshes. For problems where the geometry can be represented by a good quality mesh the cell-centred method is computationally efficient producing a diagonally dominant solution matrix that can be solved speedily by nearly any iterative solver. However, fitting a highly orthogonal mesh to a complex geometry can be one of the most time consuming aspects of the modelling process. On distorted meshes the inclusion of non-orthogonal correction terms during the discretisation process to account for non-orthogonality in the mesh can be problematic. On arbitrary distorted meshes divergence is often encountered. Solving without accounting for mesh non-orthogonality introduces inaccuracies, with solutions having some dependence on the mesh skewness. When convergence is obtained on highly distorted meshes the solutions are often highly inaccurate. The non-orthogonal errors introduced into the solution method are multiplied when solving coupled variables. For flow solutions, errors in the velocity field will effect pressure solutions and vice versa. In this way errors accumulate, increasing as the solution procedure progresses. The use of the Rhie-Chow interpolation scheme to calculate face based velocity components can lead to face velocities that satisfy continuity but with non-physical element-based values. For multi-physics multi-mesh problems, extrapolating boundary values from element based values can introduce error into the solution procedure. For example, coupled stress problems can lead to mesh deformation during the solution procedure that introduces non-orthogonality issues.

The vertex-based discretisation technique has been shown to handle distorted meshes with ease. Solutions obtained on skewed meshes are comparable with solutions obtained on a uniform Cartesian mesh. The computational requirements of the vertex-based discretisation method are its main drawback. On a stationary mesh the discretisation procedure can be accelerated by the storage of the elemental shape functions and their derivatives at local integration points. However, on a moving mesh these quantities, which require the inversion of a Jacobian matrix associated with each element, need to be recalculated at each time-step. This, plus the additional topological and control volume specifications makes the vertex-based technique computationally very expensive. The vertex-based solution matrix is symmetric for sub-control volumes of equal size or asymmetric for unequal sub-control volumes, and positive-definite with a much larger bandwidth than the solution matrix of the cell-centred discretisation procedure. The increased communication between solution points leads to less solver iterations required to obtain solutions but increased solver computational time. Solving directly on the boundary wall, boundary solutions are more accurately represented and boundary information is more readily passed through the solution domain. The vertex-based computational time, per solved variable, is approximately 1.8 times that of the cell-centred method. The additional time requirements can be offset by time saved in mesh generation. But the solution of additional variables increasingly adds to the vertex-based simulation time.

In an attempt to reduce computational requirements, whilst retaining the ability of the vertex-based technique to handle mesh distortion, hybrid flow solutions were investigated. The partially staggered hybrid formulations and co-located techniques investigated in this research project, require specialist discretisation techniques to avoid checkerboard pressure predictions. Discretising either the momentum or continuity equation using vertex-based techniques allowed the solution on distorted meshes where purely cell-centred techniques fail. However, non-orthogonal errors introduced through cell-centred discretisation degraded the solutions and results appear to have some dependence on the distorted mesh. Discretising the momentum equations using vertex-based methods allows better solution of vertex velocity components, but face velocity components are dependent on cell-centred face pressure gradients. Errors in defining the pressure gradient across the face of a distorted control volume are introduced into the solution process. Solving pressure using vertex-based discretisation allows for face pressure gradients to be more accurately represented, but non-orthogonal errors are still included in the cell-centred discretisation of the momentum equations. Neither of the hybrid formulations gave consistently better results on all meshes investigated.

The solution of flow using vertex-based co-located hydrodynamic variables gives consistent results for all meshes investigated. The coupling of vertex-based flow with cell-centred thermal and turbulent quantities was investigated. The results obtained were encouraging with combined vertex-based/cell-centred techniques giving good solutions on all meshes investigated. Even on a relatively coarse mesh the vertex-based solutions identify local minimum and maximum values. The concern of mass conservation over the element control volume was addressed. As mass is conserved over the vertex-based control volume and extrapolated to element face values, mass conservation over the element cell is slightly degraded. However, the error in element mass conservation reduces as vertex-based mass conservation is more strictly enforced. The cell-centred discretisation of transported quantities still include non-orthogonal issues. Although transported quantities are slightly dependent on the amount of mesh skewness, non-orthogonal errors included in the cell-centred discretisation do not appear to significantly effect the final solution. Obtaining a good flow field on a distorted mesh aids the solution of other transported quantities using cell-centred techniques. This technique should enable reliable solutions on distorted meshes whilst reducing the computational requirements for multi-physics flow problems.

8.2 Future Work

The combined vertex-based/cell-centred discretisation procedure has shown promising results in the cases investigated in this research project. The vertex-based flow method could be coupled with additional PHYSICA modules that use cell-centred discretisation techniques for the transported variables. There are several other potential avenues for future investigation and suggestions are given in the following sections.

8.2.1 $k - \epsilon$ Model: Turbulent Generation Rate

Higher turbulent generation rates were encountered in $vb - cc$ solutions of turbulent flow over an aircraft wing than those found using cc solutions. This was more pronounced in the region downstream adjacent to the wings trailing edge. In the half elements adjacent to the wing surface the turbulent generation rate is set to zero since the fluid layer close to a wall is assumed to be laminar. In the elements downstream adjacent to the wings trailing edge, at the intersection of the upper and lower wing surface, the turbulent generation rate is calculated from the velocity gradients. Since these elements have a node that forms part of the wing surface, and at the wing surface the fluid is stationary, laminar flow may be assumed in the fluid layer close to the wall intersection point. In the $vb - cc$ method the turbulent generation rate in the sub-control volume downstream adjacent to the wall intersection point could be set to zero which would reduce the turbulent viscosity in the wake.

8.2.2 Vertex-Based Convection Scheme

In this research project the hybrid discretisation scheme has been employed. In highly convective flows the hybrid method reverts to upwind differencing. It is well known that upwind differencing introduces some degree of numerical diffusion. Numerical diffusion results in smearing of results and local minimum and maximum values can be lost. A number of higher order schemes have been successfully applied to cell-centred techniques. An alternative higher order directional-differencing scheme, along the lines of the MAW scheme, could be employed in the vertex-based procedure in an attempt to reduce the amount of numerical diffusion.

8.2.3 Vertex-Based Transient Term

Chapter 2 gave two possible discretisation methods for the transient term, Equation 2.19 and Equation 2.21. It was shown in the transient-diffusion test case, Chapter 3, that discretising the transient term using Equation 2.21 gave bounded converging improved solutions when diffusion dominated, i.e.

$$\frac{\rho c V_{scv} dx}{k} < \Delta t \tag{8.1}$$

During this research project the transient term has been discretised using Equation 2.19 to ensure bounded converging solutions for all Δt. A hybrid discretisation method employing Equation 2.21 could be investigated for the transient term. On a element by element basis, if the transient term dominated, the integration point could be taken at the mesh node, if diffusion dominated, the integration point could be taken at the sub-control centre. This method would ensure that the boundedness condition, $\sum a_{nb} > 0$ is not violated.

8.2.4 Mixed Element Types

Vertex-based discretisation was successfully applied to hexahedral, pentahedral and tetrahedral mesh elements for diffusion problems. The vertex-based flow solution has only been applied to hexahedral elements but could be extended to pentahedral, tetrahedral or any element for which a shape function exists. Problems could be encountered when solving other transported variables cell-centred due to non-orthogonal connections between element centres.

8.2.5 Multi-Mesh Solutions

Fitting a mesh to a complex geometry often results in non-orthogonal issues being confined to specific areas of the mesh. The mesh could be divided into orthogonal and non-orthogonal regions. This would allow vertex-based flow solutions on sections of the mesh of poor quality and cell-centred flow solutions in orthogonal regions. This technique would reduce memory and computational requirements.

Appendix A

Local Co-ordinate Systems

A.1 Normalised Triangular Element

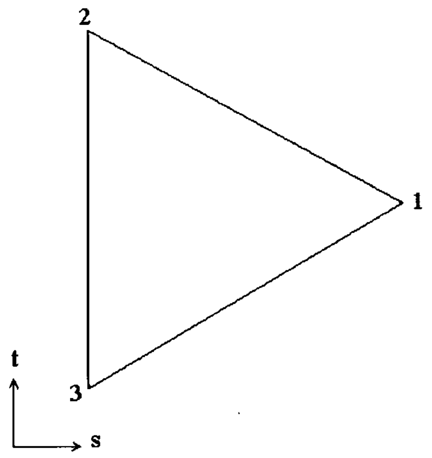


Figure A.1: Normalized triangle element

The local co-ordinates of the vertices in A.1 are as follows:

Vertex	s	t
1	1.0	0.0
2	-1/2	$\sqrt{3}/2$
3	-1/2	$-\sqrt{3}/2$

A.1.1 Integration Points

The integration points given in local co-ordinates, Table A.1, are located at the centre of the control volume faces shown in Figure A.2.

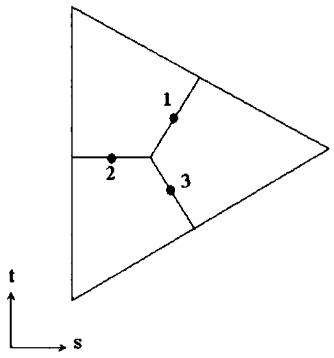


Figure A.2: Control volume faces of triangle element

Face	**s**	**t**
1	1/8	$\sqrt{3/8}$
2	-1/4	0
3	1/8	$-\sqrt{3/8}$

Table A.1: Triangle element face integration points

A.2 Normalised Tetrahedral Element

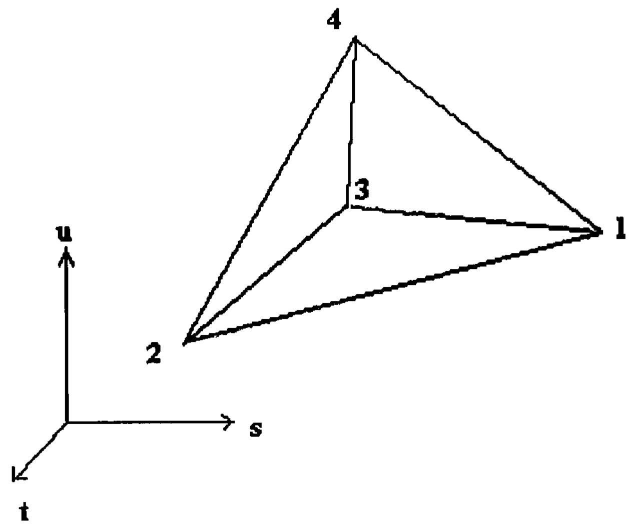


Figure A.3: Normalized tetrahedral element

The local co-ordinates of the vertices in A.3 are as follows:

Vertex	s	t	u
1	1.0	0.0	$-\sqrt{2}/4$
2	-1/2	$\sqrt{3}/2$	$-\sqrt{2}/4$
3	-1/2	$-\sqrt{3}/2$	$-\sqrt{2}/4$
4	0.0	0.0	$3\sqrt{2}/4$

A.2.1 Integration Points

The integration points given in local co-ordinates, Table A.2, are located at the centre of the control volume faces shown in Figure A.4.

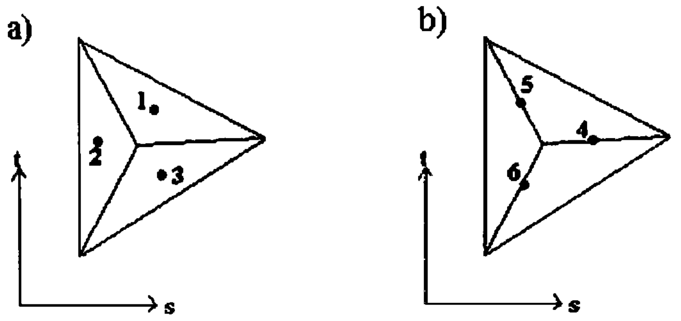


Figure A.4: Control volume faces of tetrahedral element
a) $u = -5\sqrt{2/48}$ plane, b) $u = 5\sqrt{2/48}$ plane

Face	s	t	u
1	1/8	$\sqrt{3/8}$	$-5\sqrt{2/48}$
2	-1/4	0	$-5\sqrt{2/48}$
3	1/8	$-\sqrt{3/8}$	$-5\sqrt{2/48}$
4	1/4	0	$5\sqrt{2/48}$
5	-1/8	$\sqrt{3/8}$	$5\sqrt{2/48}$
6	-1/8	-$\sqrt{3/8}$	$5\sqrt{2/48}$

Table A.2: Tetrahedral element face integration points

A.3 Normalised Pentahedral Element

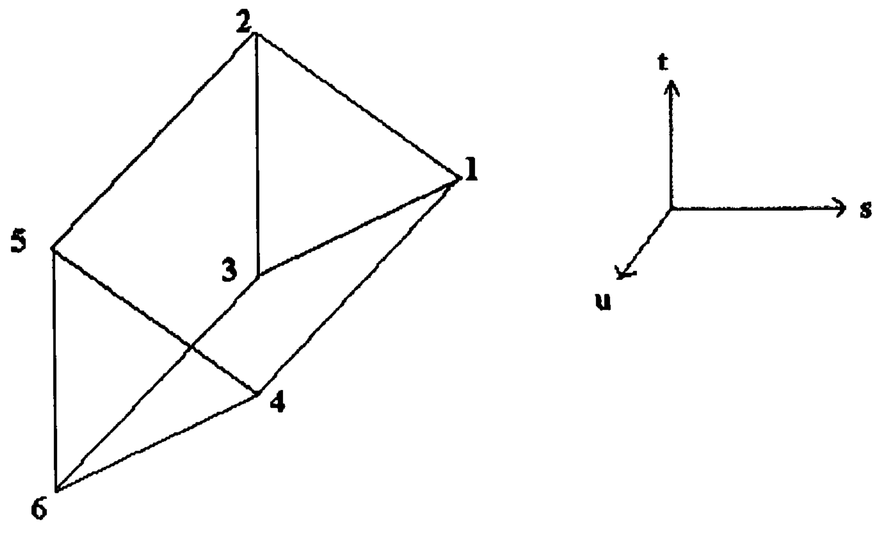


Figure A.5: Normalized pentahedral element

The local co-ordinates of the vertices in A.5 are as follows:

Vertex	**s**	**t**	**u**
1	1.0	0.0	-1.0
2	-1/2	$\sqrt{3}/2$	-1.0
3	-1/2	$-\sqrt{3}/2$	-1.0
4	1.0	0.0	1.0
5	-1/2	$\sqrt{3}/2$	1.0
6	-1/2	$-\sqrt{3}/2$	1.0

A.3.1 Integration Points

The integration points given in local co-ordinates, Table A.3, are located at the centre of the control volume faces shown in Figure A.6.

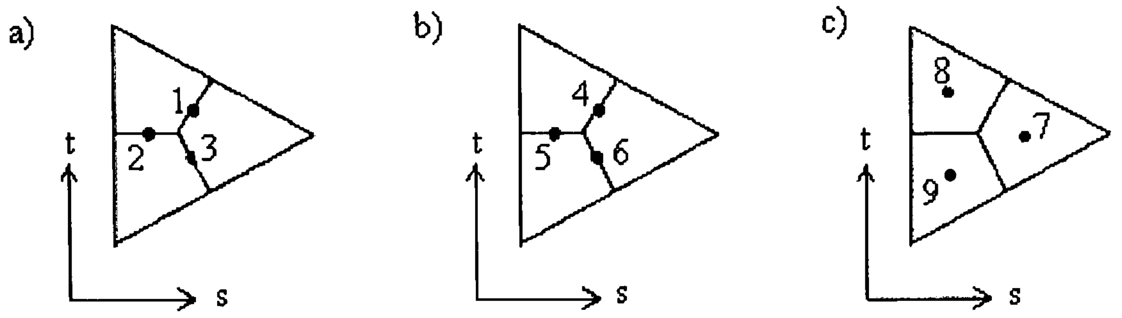


Figure A.6: Control volume faces of pentahedral element

a) $u = -1/2$ plane, b) $u = 1/2$ plane, c) $u = 0$ plane

Face	**s**	**t**	**u**
1	1/8	$\sqrt{3/8}$	-1/2
2	-1/4	0	-1/2
3	1/8	$-\sqrt{3/8}$	-1/2
4	1/8	$\sqrt{3/8}$	1/2
5	-1/4	0	1/2
6	1/8	$-\sqrt{3/8}$	1/2
7	3/8	0	0
8	-3/16	$3\sqrt{3/16}$	0
9	-3/16	$-3\sqrt{3/16}$	0

Table A.3: Pentahedral element face integration points

A.4 Normalised Hexahedral Element

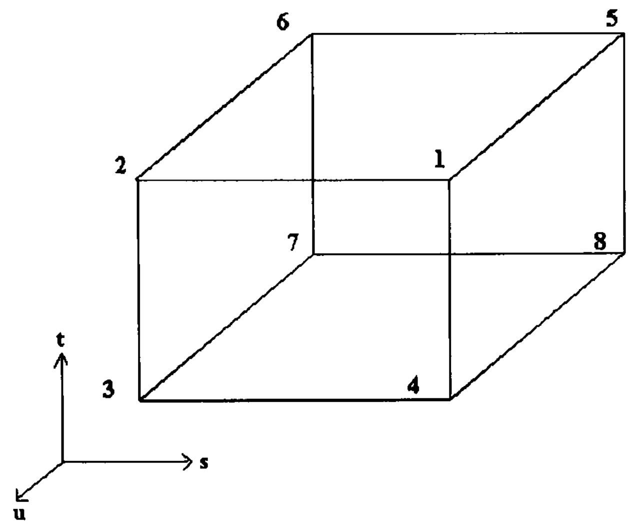


Figure A.7: Normalized hexahedral element

The local co-ordinates of the vertices in A.7 are as follows:

Vertex	s	t	u
1	1.0	1.0	1.0
2	-1.0	1.0	1.0
3	-1.0	-1.0	1.0
4	1.0	-1.0	1.0
5	1.0	1.0	-1.0
6	-1.0	1.0	-1.0
7	-1.0	-1.0	-1.0
8	1.0	-1.0	-1.0

A.4.1 Integration Points

The integration points given in local co-ordinates, Table A.4, are located at the centre of the control volume faces shown in Figure A.8.

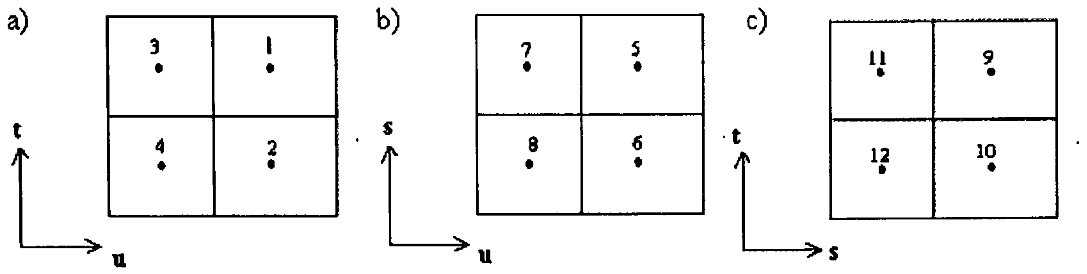


Figure A.8: Control volume faces of hexahedral element

a) $s = 0$ plane, b) $t = 0$ plane, c) $u = 0$ plane

Face	s	t	u	Face	s	t	u
1	0	1/2	1/2	7	1/2	0	-1/2
2	0	-1/2	1/2	8	-1/2	0	-1/2
3	0	1/2	-1/2	9	1/2	1/2	0
4	0	-1/2	-1/2	10	1/2	-1/2	0
5	1/2	0	1/2	11	-1/2	1/2	0
6	-1/2	0	1/2	12	-1/2	-1/2	0

Table A.4: Hexahedral element face integration points

Appendix B

Shape Functions

The shape functions and their local derivatives are defined in local co-ordinates s, t and u.

B.1 Triangular Element

Shape functions:-

$$N_1(s,t) = \tfrac{1}{3}(1+2s) \quad N_2(s,t) = \tfrac{1}{3}(1-s+\sqrt{3}t) \quad N_3(s,t) = \tfrac{1}{3}(1-s-\sqrt{3}t)$$

Local derivatives:-

$$\begin{array}{ll} \frac{\partial N_1}{\partial s} = 2/3 & \frac{\partial N_1}{\partial t} = 0 \\ \frac{\partial N_2}{\partial s} = -1/3 & \frac{\partial N_2}{\partial t} = 1/\sqrt{3} \\ \frac{\partial N_3}{\partial s} = -1/3 & \frac{\partial N_3}{\partial t} = -1/\sqrt{3} \end{array}$$

B.2 Tetrahedral Element

Shape functions:-

$$\begin{array}{ll}
N_1(s,t,u) = \frac{1}{4} + \frac{2}{3}s - \frac{1}{3\sqrt{2}}u & N_2(s,t,u) = \frac{1}{4} - \frac{1}{3}s + \frac{2\sqrt{3}}{6}t - \frac{1}{3\sqrt{2}}u \\
N_3(s,t,u) = \frac{1}{4} - \frac{1}{3}s - \frac{2\sqrt{3}}{6}t - \frac{1}{3\sqrt{2}}u & N_4(s,t,u) = \frac{1}{4} - \frac{1}{\sqrt{2}}u
\end{array}$$

Local derivatives:-

$$\begin{array}{lll}
\frac{\partial N_1}{\partial s} = \frac{2}{3} & \frac{\partial N_1}{\partial t} = 0 & \frac{\partial N_1}{\partial u} = \frac{-1}{3\sqrt{2}} \\
\frac{\partial N_2}{\partial s} = \frac{-1}{3} & \frac{\partial N_2}{\partial t} = \frac{2\sqrt{3}}{6} & \frac{\partial N_2}{\partial u} = \frac{-1}{3\sqrt{2}} \\
\frac{\partial N_3}{\partial s} = \frac{-1}{3} & \frac{\partial N_3}{\partial t} = \frac{-2\sqrt{3}}{6} & \frac{\partial N_3}{\partial u} = \frac{-1}{3\sqrt{2}} \\
\frac{\partial N_4}{\partial s} = 0 & \frac{\partial N_4}{\partial t} = 0 & \frac{\partial N_4}{\partial u} = \frac{1}{\sqrt{2}}
\end{array}$$

B.3 Pentahedral Element

Shape functions:-

$$\begin{array}{ll}
N_1(s,t,u) = \frac{1}{6}(1+2s)(1-u) & N_2(s,t,u) = \frac{1}{6}(1-s+\sqrt{3}t)(1-u) \\
N_3(s,t,u) = \frac{1}{6}(1-s-\sqrt{3}t)(1-u) & N_4(s,t,u) = \frac{1}{6}(1+2s)(1+u) \\
N_5(s,t,u) = \frac{1}{6}(1-s-\sqrt{3}t)(1+u) & N_6(s,t,u) = \frac{1}{6}(1-s-\sqrt{3}t)(1+u)
\end{array}$$

Local derivatives:-

$$\begin{array}{lll}
\frac{\partial N_1}{\partial s} = \frac{1}{3}(1-u) & \frac{\partial N_1}{\partial t} = 0 & \frac{\partial N_1}{\partial u} = \frac{-1}{6}(1+2s) \\
\frac{\partial N_2}{\partial s} = \frac{-1}{6}(1-u) & \frac{\partial N_2}{\partial t} = \frac{\sqrt{3}}{6}(1-u) & \frac{\partial N_2}{\partial u} = \frac{-1}{6}(1-s+\sqrt{3}t) \\
\frac{\partial N_3}{\partial s} = \frac{-1}{6}(1-u) & \frac{\partial N_3}{\partial t} = \frac{-\sqrt{3}}{6}(1-u) & \frac{\partial N_3}{\partial u} = \frac{-1}{6}(1-s-\sqrt{3}t) \\
\frac{\partial N_4}{\partial s} = \frac{1}{3}(1+u) & \frac{\partial N_4}{\partial t} = 0 & \frac{\partial N_4}{\partial u} = \frac{1}{6}(1+2s) \\
\frac{\partial N_5}{\partial s} = \frac{-1}{6}(1+u) & \frac{\partial N_5}{\partial t} = \frac{\sqrt{3}}{6}(1+u) & \frac{\partial N_5}{\partial u} = \frac{1}{6}(1-s+\sqrt{3}t) \\
\frac{\partial N_6}{\partial s} = \frac{-1}{6}(1+u) & \frac{\partial N_6}{\partial t} = \frac{-\sqrt{3}}{6}(1+u) & \frac{\partial N_6}{\partial u} = \frac{1}{6}(1-s-\sqrt{3}t)
\end{array}$$

B.4 Hexahedral Element

Shape functions:-

$$\begin{aligned}
N_1(s,t,u) &= \tfrac{1}{8}(1+s)(1+t)(1+u) & N_2(s,t,u) &= \tfrac{1}{8}(1-s)(1+t)(1+u)\\
N_3(s,t,u) &= \tfrac{1}{8}(1-s)(1-t)(1+u) & N_4(s,t,u) &= \tfrac{1}{8}(1+s)(1-t)(1+u)\\
N_5(s,t,u) &= \tfrac{1}{8}(1+s)(1+t)(1-u) & N_6(s,t,u) &= \tfrac{1}{8}(1-s)(1+t)(1-u)\\
N_7(s,t,u) &= \tfrac{1}{8}(1-s)(1-t)(1-u) & N_8(s,t,u) &= \tfrac{1}{8}(1+s)(1-t)(1-u)
\end{aligned}$$

Local derivatives:-

$$\begin{aligned}
\tfrac{\partial N_1}{\partial s} &= \tfrac{1}{8}(1+t)(1+u) & \tfrac{\partial N_1}{\partial t} &= \tfrac{1}{8}(1+s)(1+u) & \tfrac{\partial N_1}{\partial u} &= \tfrac{1}{8}(1+s)(1+t)\\
\tfrac{\partial N_2}{\partial s} &= \tfrac{-1}{8}(1+t)(1+u) & \tfrac{\partial N_2}{\partial t} &= \tfrac{1}{8}(1-s)(1+u) & \tfrac{\partial N_2}{\partial u} &= \tfrac{1}{8}(1-s)(1+t)\\
\tfrac{\partial N_3}{\partial s} &= \tfrac{-1}{8}(1-t)(1+u) & \tfrac{\partial N_3}{\partial t} &= \tfrac{-1}{8}(1-s)(1+u) & \tfrac{\partial N_3}{\partial u} &= \tfrac{1}{8}(1-s)(1-t)\\
\tfrac{\partial N_4}{\partial s} &= \tfrac{1}{8}(1-t)(1+u) & \tfrac{\partial N_4}{\partial t} &= \tfrac{-1}{8}(1+s)(1+u) & \tfrac{\partial N_4}{\partial u} &= \tfrac{1}{8}(1+s)(1-t)\\
\tfrac{\partial N_5}{\partial s} &= \tfrac{1}{8}(1+t)(1-u) & \tfrac{\partial N_5}{\partial t} &= \tfrac{1}{8}(1+s)(1-u) & \tfrac{\partial N_5}{\partial u} &= \tfrac{-1}{8}(1+s)(1+t)\\
\tfrac{\partial N_6}{\partial s} &= \tfrac{-1}{8}(1+t)(1-u) & \tfrac{\partial N_6}{\partial t} &= \tfrac{1}{8}(1-s)(1-u) & \tfrac{\partial N_6}{\partial u} &= \tfrac{-1}{8}(1-s)(1+t)\\
\tfrac{\partial N_7}{\partial s} &= \tfrac{-1}{8}(1-t)(1-u) & \tfrac{\partial N_7}{\partial t} &= \tfrac{-1}{8}(1-s)(1-u) & \tfrac{\partial N_7}{\partial u} &= \tfrac{-1}{8}(1-s)(1-t)\\
\tfrac{\partial N_8}{\partial s} &= \tfrac{1}{8}(1-t)(1-u) & \tfrac{\partial N_8}{\partial t} &= \tfrac{-1}{8}(1+s)(1-u) & \tfrac{\partial N_8}{\partial u} &= \tfrac{-1}{8}(1+s)(1-t)
\end{aligned}$$

Appendix C

Local-Global Transformations

Equations (C.1) define the co-ordinate transformations for x, y and z defined in the local co-ordinate system s, t and u,

$$\begin{aligned} x(s,t,u) &= \sum_{i=1}^{n} N_i(s,t,u)x_i \\ y(s,t,u) &= \sum_{i=1}^{n} N_i(s,t,u)y_i \\ z(s,t,u) &= \sum_{i=1}^{n} N_i(s,t,u)z_i \end{aligned} \tag{C.1}$$

where x_i, y_i and z_i are the global co-ordinates at node i and n is the number of nodes associated with the element under consideration.

The variation of a variable, ϕ, within an element is described by the same shape functions employed in Equations (C.1). Similar to the co-ordinate transformations ϕ defined in the local (s, t, u) co-ordinate system is described by,

$$\phi(s,t,u) = \sum_{i=1}^{n} N_i(s,t,u)\phi_i \tag{C.2}$$

where ϕ_i is the variable describe at node i.

The partial derivatives of ϕ with respect to the local co-ordinates can be described by the partial derivatives of the same shape functions,

$$\begin{aligned}
\frac{\partial\phi}{\partial s} &= \sum_{i=1}^{n} \frac{\partial N_i}{\partial s}(s,t,u)\phi_i \\
\frac{\partial\phi}{\partial t} &= \sum_{i=1}^{n} \frac{\partial N_i}{\partial t}(s,t,u)\phi_i \\
\frac{\partial\phi}{\partial u} &= \sum_{i=1}^{n} \frac{\partial N_i}{\partial u}(s,t,u)\phi_i
\end{aligned} \tag{C.3}$$

The local derivatives are mapped back to global derivatives by the following transformation,

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \\ \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \\ \frac{\partial N_i}{\partial u} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \\ \frac{\partial N_i}{\partial u} \end{bmatrix} \tag{C.4}$$

where J^{-1} is the inverse of the Jacobian matrix associated with the mesh element. The Jacobian can be defined in terms of the local shape functions defining the co-ordinate transformations:

$$J = \begin{bmatrix} \sum_{i=1}^{n} \frac{\partial N_i}{\partial s} x_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial s} y_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial s} z_i \\ \sum_{i=1}^{n} \frac{\partial N_i}{\partial t} x_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial t} y_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial t} z_i \\ \sum_{i=1}^{n} \frac{\partial N_i}{\partial u} x_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial u} y_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial u} z_i \end{bmatrix} \tag{C.5}$$

Appendix D

System of Equation Coefficients

The terms in the general transport equation (2.1) are discretised using vertex-based techniques to produce a system of linear equations of the form $[A][\phi] = [B]$, where $[\phi]$ is a vector of values of ϕ at the mesh vertices and $[A]$ is a matrix of coefficients. The resulting matrix of coefficients is symmetric and positive definite. An example of the type of coefficient matrix obtained, for a single row of 5 elements, is given in Figures D.1, D.2 and D.3 for the diffusion, transient and convection terms respectively. For all terms shape function approximations have been employed in the discretisation process giving coefficient values where $a > b > c > d$.

$$\left[\begin{array}{cccccccccccccccccccccccc}
a & b & -a & -b & & & & & & & & & b & c & -b & -c & & & & & & & & \\
b & a & -b & -a & & & & & & & & & c & b & -c & -b & & & & & & & & \\
-a & -b & 2a & 2b & -a & -b & & & & & & & -b & -c & 2b & 2c & -b & -c & & & & & & \\
-b & -a & 2b & 2a & -b & -a & & & & & & & -c & -b & 2c & 2b & -c & -b & & & & & & \\
& & -a & -b & 2a & 2b & -a & -b & & & & & & & -b & -c & 2b & 2c & -b & -c & & & & \\
& & -b & -a & 2b & 2a & -b & -a & & & & & & & -c & -b & 2c & 2b & -c & -b & & & & \\
& & & & -a & -b & 2a & 2b & -a & -b & & & & & & & -b & -c & 2b & 2c & -b & -c & & \\
& & & & -b & -a & 2b & 2a & -b & -a & & & & & & & -c & -b & 2c & 2b & -c & -b & & \\
& & & & & & -a & -b & 2a & 2b & -a & -b & & & & & & & -b & -c & 2b & 2c & -b & -c \\
& & & & & & -b & -a & 2b & 2a & -b & -a & & & & & & & -c & -b & 2c & 2b & -c & -b \\
& & & & & & & & -a & -b & a & b & & & & & & & & & -b & -c & b & c \\
& & & & & & & & -b & -a & b & a & & & & & & & & & -c & -b & c & b \\
b & c & -b & -c & & & & & & & & & a & b & -a & -b & & & & & & & & \\
c & b & -c & -b & & & & & & & & & b & a & -b & -a & & & & & & & & \\
-b & -c & 2b & 2c & -b & -c & & & & & & & -a & -b & 2a & 2b & -a & -b & & & & & & \\
-c & -b & 2c & 2b & -c & -b & & & & & & & -b & -a & 2b & 2a & -b & -a & & & & & & \\
& & -b & -c & 2b & 2c & -b & -c & & & & & & & -a & -b & 2a & 2b & -a & -b & & & & \\
& & -c & -b & 2c & 2b & -c & -b & & & & & & & -b & -a & 2b & 2a & -b & -a & & & & \\
& & & & -b & -c & 2b & 2c & -b & -c & & & & & & & -a & -b & 2a & 2b & -a & -b & & \\
& & & & -c & -b & 2c & 2b & -c & -b & & & & & & & -b & -a & 2b & 2a & -b & -a & & \\
& & & & & & -b & -c & 2b & 2c & -b & -c & & & & & & & -a & -b & 2a & 2b & -a & -b \\
& & & & & & -c & -b & 2c & 2b & -c & -b & & & & & & & -b & -a & 2b & 2a & -b & -a \\
& & & & & & & & -b & -c & b & c & & & & & & & & & -a & -b & a & b \\
& & & & & & & & -c & -b & c & b & & & & & & & & & -b & -a & b & a
\end{array}\right]$$

Figure D.1: Diffusion coefficient matrix

$$\left[\begin{array}{cccccccccccccccccccccccc}
a & b & b & c & & & & & & & & & b & c & c & d & & & & & & & & \\
b & a & c & b & & & & & & & & & c & b & c & b & & & & & & & & \\
b & c & 2a & 2b & b & c & & & & & & & c & d & 2b & 2c & c & d & & & & & & \\
c & b & 2b & 2a & c & b & & & & & & & d & c & 2c & 2b & d & c & & & & & & \\
& & b & c & 2a & 2b & b & c & & & & & & & c & d & 2b & 2c & c & d & & & & \\
& & c & b & 2b & 2a & c & b & & & & & & & d & c & 2c & 2b & d & c & & & & \\
& & & & b & c & 2a & 2b & b & c & & & & & & & c & d & 2b & 2c & c & d & & \\
& & & & c & b & 2b & 2a & c & b & & & & & & & d & c & 2c & 2b & d & c & & \\
& & & & & & b & c & 2a & 2b & b & c & & & & & & & c & d & 2b & 2c & c & d \\
& & & & & & c & b & 2b & 2a & c & b & & & & & & & d & c & 2c & 2b & d & c \\
& & & & & & & & b & c & a & b & & & & & & & & & c & d & b & c \\
& & & & & & & & c & b & b & a & & & & & & & & & d & c & c & b \\
b & c & c & d & & & & & & & & & a & b & b & c & & & & & & & & \\
c & b & c & b & & & & & & & & & b & a & c & b & & & & & & & & \\
c & d & 2b & 2c & c & d & & & & & & & b & c & 2a & 2b & b & c & & & & & & \\
d & c & 2c & 2b & d & c & & & & & & & c & b & 2b & 2a & c & b & & & & & & \\
& & c & d & 2b & 2c & c & d & & & & & & & b & c & 2a & 2b & b & c & & & & \\
& & d & c & 2c & 2b & d & c & & & & & & & c & b & 2b & 2a & c & b & & & & \\
& & & & c & d & 2b & 2c & c & d & & & & & & & b & c & 2a & 2b & b & c & & \\
& & & & d & c & 2c & 2b & d & c & & & & & & & c & b & 2b & 2a & c & b & & \\
& & & & & & c & d & 2b & 2c & c & d & & & & & & & b & c & 2a & 2b & b & c \\
& & & & & & d & c & 2c & 2b & d & c & & & & & & & c & b & 2b & 2a & c & b \\
& & & & & & & & c & d & b & c & & & & & & & & & b & c & a & b \\
& & & & & & & & d & c & c & b & & & & & & & & & c & b & b & a
\end{array}\right]$$

Figure D.2: Transient coefficient matrix

$$\begin{bmatrix}
a & b & a & b & & & & & & & & & b & c & b & c & & & & & & & & \\
b & a & b & a & & & & & & & & & c & b & c & b & & & & & & & & \\
-a & -b & & & a & b & & & & & & & -b & -c & & & b & c & & & & & & \\
-b & -a & & & b & c & & & & & & & -c & -b & & & c & b & & & & & & \\
 & & -a & -b & & & a & b & & & & & & & -b & -c & & & b & c & & & & \\
 & & -b & -a & & & b & c & & & & & & & -c & -b & & & c & b & & & & \\
 & & & & -a & -b & & & a & b & & & & & & & -b & -c & & & b & c & & \\
 & & & & -b & -a & & & b & c & & & & & & & -c & -b & & & c & b & & \\
 & & & & & & -a & -b & & & a & b & & & & & & & -b & -c & & & b & c \\
 & & & & & & -b & -a & & & b & c & & & & & & & -c & -b & & & c & b \\
 & & & & & & & & -a & -b & a & b & & & & & & & & & -b & -c & b & c \\
 & & & & & & & & -b & -a & b & a & & & & & & & & & -c & -b & c & b \\
b & c & b & c & & & & & & & & & a & b & a & b & & & & & & & & \\
c & b & c & b & & & & & & & & & b & a & b & a & & & & & & & & \\
-b & -c & & & b & c & & & & & & & -a & -b & & & a & b & & & & & & \\
-c & -b & & & c & b & & & & & & & -b & -a & & & b & c & & & & & & \\
 & & -b & -c & & & b & c & & & & & & & -a & -b & & & a & b & & & & \\
 & & -c & -b & & & c & b & & & & & & & -b & -a & & & b & c & & & & \\
 & & & & -b & -c & & & b & c & & & & & & & -a & -b & & & a & b & & \\
 & & & & -c & -b & & & c & b & & & & & & & -b & -a & & & b & c & & \\
 & & & & & & -b & -c & & & b & c & & & & & & & -a & -b & & & a & b \\
 & & & & & & -c & -b & & & c & b & & & & & & & -b & -a & & & b & c \\
 & & & & & & & & -b & -c & b & c & & & & & & & & & -a & -b & a & b \\
 & & & & & & & & -c & -b & c & b & & & & & & & & & -b & -a & b & a
\end{bmatrix}$$

Figure D.3: Convection coefficient matrix

Appendix E

ONERA M6 Wing Cross-section

The normalised co-ordinates of the ONERA wing cross-section are given below, where l is the wing chord length.

X/l	Y/l	X/l	Y/l	X/l	Y/l	X/l	Y/l
0.0000000	0.0000000	0.0241067	0.0242004	0.3761446	0.0489296	0.8174828	0.0219842
0.0000165	0.0006914	0.0297008	0.0258245	0.4018567	0.0488202	0.8407324	0.0195838
0.0000696	0.0014416	0.0364261	0.0273317	0.4274223	0.0484833	0.8638564	0.0170915
0.0001675	0.0022554	0.0444852	0.0287912	0.4528441	0.0479351	0.8868235	0.0145051
0.0003232	0.0031382	0.0541248	0.0303278	0.4781197	0.0471661	0.9061905	0.0121952
0.0005508	0.0040959	0.0656303	0.0320138	0.5032514	0.0461903	0.9225336	0.0101138
0.0008657	0.0051343	0.0793366	0.0338372	0.5282426	0.0450209	0.9363346	0.0083265
0.0012868	0.0062598	0.0956354	0.0357742	0.5530937	0.0436741	0.9479946	0.0068038
0.0018364	0.0074784	0.1149796	0.0377923	0.5778043	0.0421684	0.9578511	0.0055144
0.0025441	0.0087958	0.1378963	0.0398522	0.6023757	0.0405241	0.9661860	0.0044240
0.0034428	0.0102163	0.1649976	0.0419089	0.6268104	0.0387613	0.9732361	0.0035015
0.0045704	0.0117419	0.1919327	0.0436214	0.6511093	0.0368990	0.9792020	0.0027211
0.0059751	0.0133708	0.2187096	0.0450507	0.6752726	0.0349542	0.9842508	0.0020606
0.0077112	0.0150951	0.2453310	0.0462358	0.6993027	0.0329402	0.9885252	0.0015014
0.0098413	0.0168984	0.2717978	0.0471987	0.7231995	0.0308662	0.9921438	0.0010280
0.0124479	0.0187537	0.2981113	0.0479494	0.7469658	0.0287365	0.9952080	0.0006271
0.0156171	0.0206220	0.3242726	0.0484902	0.7705998	0.0265505	0.9978030	0.0002876
0.0194609	0.0224545	0.3502830	0.0488183	0.7941055	0.0243027	1.0000000	0.0000000

Bibliography

[1] N. Croft, K. A. Pericleous and M. Cross, 'PHYSICA: A multiphysics environment for complex flow processes'. In C. Taylor and P. Durbetaki, editors, *Numerical Methods in Laminar and Turbulent Flow '95*, 2, 1269-1280, 1995.

[2] T. N. Croft, 'Unstructured Mesh - Finite Volume Algorithms for Swirling, Turbulent, Reacting Flows'. Ph.D thesis, *The University of Greenwich*, 1998.

[3] P. W. McDonald, 'The computation of transonic flow through two-dimensional gas turbine cascades', *Technical Report 71-gt-89*, asme, 1971.

[4] S. V. Patankar and D. B. Spalding, 'A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows',*International Journal of Heat and Mass Transfer*, 15, 1787-1806, 1972.

[5] L.S. Caretto, R. M. Curr and D. B. Spalding, 'Two Numerical Methods for Three-Dimensional Boundary Layers', *Computer Methods in Applied Mechanics and Engineering*, 1, 39-57, 1972.

[6] S. V. Patankar, 'Numerical Heat Transfer and Fluid Flow', *Hemisphere Publishing Corporation*, 1980.

[7] C.Hsu, 'A Curvilinear-Coordinate method for Momentum, Heat and Mass Transfer in domains of Irregular Geometry', Ph.D thesis, *University of Minnesota*, 1981

[8] C. M. Rhie and W. L. Chow, 'Numerical Study of the Turbulent flow Past an Aerofoil with Trailing Edge Separation', *AIAA J.*, 21,11, 1525-1532, 1983.

[9] C. Hirsh, 'Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization', volume 1. *John Wiley and Sons*, 1988.

[10] A.J. Chorin, 'Numerical Solution of the Navier-Stokes Equations', *Math. Comput.*, 22, 745-762, 1968

[11] G.D.Raithby, 'Skew Upstream Differencing Schemes for Problems Involving Fluid Flow', *Comp. Meth. Appl. Mech. Eng.*, 9, 153-164, 1976

[12] F.H.Harlow and J.E.Welch, 'Numerical Calculation of Time-dependent Viscous Incompressible Flow of Fluid with Free Surface', *The Physics of Fluids*, 8, 2183-2189, 1965

[13] J.E.Welch, F.H.Harlow, J.P.Shannon and B.J.Daly, 'The MAC Method', *LASL Report No.LA-3425, Los Alamos scientific Laboratory*, Los Alamos, New Mexico, 1966

[14] S. Whitaker, 'Advances in Theory of Fluid Motion in Porous Media', *Industrial and Engineering Chemistry*, 61(12), 14-28, 1972

[15] W. J. Gordon and C. A. Hall, 'Construction of Curvilinear Co-ordinate Systems and applications to Mesh Gerneration', *International Journal for Numerical Methods in Engineering*, 7, 461-477, 1973

[16] J. F. Thompson, F. C. Thames and C. W. Mastin, 'Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Aritrary Two-Dimensional Bodies', *Journal of Computational Physics*, 15, 299-319, 1974

[17] J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, 'Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review', *Journal of Computational Physics*, 47,108, 1982

[18] P. Coelho and J. C. F. Peraira, 'Calculation Procedure for 3D Laminar Flows in Complex Geometries Using a Nonstaggered Nonorthogonal Grid System', *Applied Mathematical Modelling*, 17, 562-576, 1993

[19] W. Shyy and T. C. Vu, 'On the Adoption of Velocity Variable and Grid System for Fluid flow Computation in Curvilinear Coordinates', *Journal of Computational Physics*, 92, 82-105, 1991

[20] S. K. Choi, H. Y. Nam, Y. B. Lee and M. Cho, 'An Efficient Three-Dimensional Calculation Procedure for fIncompressible Flows in Complex Geometries', *Numerical Heat Transfer, Part B*, 23, 387-400, 1993

[21] I.Demirdzic and M. Peric, 'Finite Volume Method for Prediction of Fluid Flow in Arbitrary Shaped Domains with Moving Boundaries', *International Journal for Numerical Methods in Fluids*, 10, 771-790, 1990

[22] I. N. Poliakov and V. A. Semin, 'An Introduction into the Method for Implementing Multi-Block Grids and / or Grids with Refinement in PHOENICS', CHAM/TR401, *The PHOENICS Journal of Computational Fluid Dynamics and its Applications*,7(2), 34-57, 1994.

[23] D. Pan, C. H. Lu and J. C. Cheng, 'Incompressible Flow Solution on Unstructured Triangular Meshes', *Numerical Heat Transfer, Part B*, 26, 207-224, 1994

[24] P. Chow, 'A Control Volume Unstructured Procedure for Convection-Diffusion Solidification Processes', Ph.D Thesis, *The University of Greenwich*, 1993.

[25] M.Thomadakis and M. Leschziner, 'Numerical Simulation of Viscous Incompressible Flows Using a Pressure-Correction Method and Unstructured Grids', *in ECCOMAS '94 Conference*, 1994.

[26] P. Chow, M. Cross and K. Pericleous, 'A Natural Extension of the Conventional Finite Volume Method into Polygonal Unstructured Meshes for CFD Applications', *Applied Mathematical Modelling*, 20, 170-183, 1996.

[27] M. Peric, 'A Finite Volume Method for the Prediction of Three Dimensional Fluid flow in Complex Ducts', Ph.D thesis, *University of London*, 1985

[28] A.D.Burns and N.S.Wikkles, 'A Finite Difference Method for the Computation of fluid Flows in Complex Three Dimensional Geometries', *AERE R* 12342, Harwell Laboratory, Oxfordshire, U.K., 1987

[29] L.Davidson, 'A Pressure Correction Method for Unstructured Meshes with Arbitrary Control Volumes', *Int. J. Numer.Methods Fluids*, 22, 265-281, 1996

[30] M.Thomadakis and M.Leschziner, 'A Pressure-Correction Method for the Solution of Incompressible Viscous Flows on Unstructured Grids',*Int. J. Numer. Methods Fluids*, 22, 581-601, 1996

[31] S.R.Mathur and J.Y.Murthy, 'A Pressure-Based Method for Unstructured Meshes', *Numer. Heat Transfer, Part B*, 31, 195-215, 1997

[32] M. Reggio and R. Camarero, 'Numerical Solution Procedure for Viscous Incompressible Flows', *Num. Heat Transfer*, 10, 131-146, 1986.

[33] T.M.Shih and C.H.Tan, 'Effects of Grid Staggering', *Int. J. Numer. Meth. Fluids*, 9, 193-212, 1989

[34] S. Majumdar, 'Role of Underrelaxation of Momentum Interpolation for Flow with Nonstaggered Grids', *Numer. Heat . Trans.*, 13, 125-132, 1988

[35] J.Chen, 'The Numerical Solution of Complex Fluid Flow Phenomena', Ph.D thesis, *Dept. Mech. Eng., University of Leeds*, 1991 F.H

[36] Harlow and J.E. Welsh, 'Numerical calculations of time independent viscous incompressible flow of fluid with a free surface', *phys, Fluids*, 8 1965 p2182

[37] C. A. Hall, J. S. Peterson, T. A. Porsching and F. R. Sledge, 'The dual variable method for finite element discretisation of Navier Stokes Equation', *int. J. Numerical Methods Eng.*,21, 883, 1985

[38] R. A. Nicolaides The covolume approach to computing incompressible flow, Incompressible Computational Fluid Dynamics, M. D. Gunzburger and R. A. Nicolaides, eds, *Cambridge University Press*, 1993 p 234-295

[39] X.Zhang, D. Schmidt and J. B. Perot, 'Accuracy and Conservation Properties of a Three-Dimensional Unstructured Staggered Mesh Scheme for Fluid Dynamics', *Journal of Computational Physics*, 175, 764-791 2002.

[40] J. B. Perot, 'Conservation Properties of Unstructured Staggered Mesh Schemes', *Journal of Computational Physics*, 159 (1), 2000.

[41] J. B. Perot and X. Zhang, 'The Development of an Adaptive Moving Unstructured Staggered Mesh Method for Incompressible Flows'. *19th Biennial Numerical Analysis Conference*, Dundee, Scotland, June 2001.

[42] X. Zhang and J. B. Perot, 'Unsteady Flow Prediction of Turbulent Flow Around a Triangular Cylinder'. *ASME Fluid Engineering Summer Conference*, Boston, MA, June, 2000. FEDSM2000-11172.

[43] J. B. Perot and X. Zhang, 'Potential Turbulence Model Prediction of Flow Around a Triangular Cylinder Using an Unstructured Staggered Mesh Method', *Proceedings of the Eighth Annual Conference of the Computational Fluid Dynamics Society of Canada*, Montreal, June, 2000.

[44] J. M. Hyman and M. Shahkov, 'Mimic discretisation of Maxwell equations', *J. comput. Phys*, 151, 881, 1999

[45] J.M.Hyman and M.Shashkov, 'The orthogonal decomposition theorem for mimetic finite difference methods', *SIAM*, J.*Numer, Anal*, 36(3), 788, 1999

[46] T. J. Barth, 'Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes', *AIAA*, paper 91-0721, 1991

[47] D. J. Mavriplis and V. Venkatakrishnan, 'A three-dimensional Agglomeration Multigrid solver for the Reynolds Averaged Navier-Stokes equations on Unstructured Meshes', *NASA* CR 195073, April 1995

[48] A. Jameson, T. J. Baker and N. P. Weatherill, 'Calculation of Inviscid Transonic Flow over a Complete Aircraft', *AIAA*, Paper 86-0103, 1986.

[49] S. K. Chakrabartty, 'Vertex-based finite-volume solution of the two-dimensional Navier-Stokes equations'. *AIAA Journal*, Vol 28(10), pp 1829-1831, October 1990

[50] S. K. Chakrabartty, 'A finite volume nodal-point scheme for solving two-dimensional Navier-Stokes equation'. *Acta Mechanica*, Vol 84 (1-2), pp 139-153. 1990

[51] S. K. Choi, 'Note on the Use of Momentum Interpolation Method for Unsteady Flows', *Numerical Heat Transfer A*, vol. 36, pp. 545-550, 1999

[52] B. Y. Kawaguchi, Y. Kawaguchi, W. Tao and H. Wzoe, 'Checkerboard Pressure Predictions due to the Underrelaxation Factor and Time Step Size for a Nonstaggered Grid with Momentum Interpolation Method', *Numerical Heat Transfer, B*, vol. 41, pp.85-94, 2002

[53] B. A. Finlayson, 'The Method of Weighted Residuals and Variational Principles', *Academic Press*, New York, 1972

[54] J. T.Oden, 'A finite element analog of the Navier-Stokes equations', *Proc. Am. Soc. Civ. Eng.*, 96, EM4, 529-34, 1970.

[55] P. Hood and C. Taylor, 'Navier-Stokes equations using mixed interpolation', in Finite Element Methods for Flow Problems (eds J. T. O. C. Zienkiewicz, R. H. Gallagher and C. Taylor), pp. 121-132, *University of Alabama Press, Huntsville,* 1974.

[56] O. C. Zienkiewicz and R. L. Taylor. 'The finite element method: Volume's 1, 2 and 3: Solid and fluid mechanics, dynamics and non-linearity'. *McGraw-Hill*, Maidenhead, Berkshire, UK, 1991.

[57] A. J. Baker, 'Computation of Fluid Flow by the Finite Element Method', *McGraw-Hill*, New York, 1983

[58] T. J. Chung, 'Finite Element Analysis of Fluid Dynamics', *McGraw-Hill*, New York, 1978

[59] J.T. Oden and L.C. Wellford, 'Analysis of Flow of Viscous Fluid by the Finite Element Method', *AIAA J*, 10,12, 1590-1599, 1972

[60] C. Taylor and P. Hood, 'A Numerical Solution of the Navier-Stokes Equations Using the Finite Element Technique', *Computational Fluids*, 1, 73-100, 1973

[61] J. C. Heinrich, P. S. Huyakorn, O. C. Zienkiewicz and A. R. Mitchell,' An "Upwind" Finite Element Scheme for Two-Dimensional Convective Transport Equation', *Int. J. Numerical Methods Eng.*, 11, 131-143, 1977

[62] A. N. Brooks and T. J. R. Hughes, 'Streamline Upwind / Petrov-Galerkin formulation for Convection Dominated flows with Particular Emphasis on the Incompressible Navier-Stokes Equations', *Computational Methods Appl. Mech. Eng*, 32, 199-207, 1982

[63] G. F. Carey and T. Plover, 'Variable Upwinding and Adaptive Mesh Requirement in Convection-Diffusion', *Int. J. Numerical Methods Eng.*, 19, 341-354, 1983

[64] R. T. Cheng, 'Numerical Solution of the Navier-Stokes Equations by the Finite Element Method', *Phys. Fluids*, 15,12, 2093-2105, 1972

[65] G. E. Schneider, G. D. Raithby, and M. M. Yovanovich, 'Finite-Element Solution Procedures for Solving the Incompressible Navier-Stokes Equations Using Equal Order Variable Interpolation', *Num. Heat. Transfer*, 1, 433-451, 1978

[66] P. Hood and C. Taylor, 'Navier-Stokes equations Using Mixed Interpolation', Finite Element Methods in Flow Problems', *University of Alabama*, Huntsville Press., 121-132, 1974

[67] P. S. Huyakorn, C. Taylor, R. L. Lee, and P. M. Gresho, 'A Comparison of Various Mixed-Interpolation Finte Elements in the Velocity-Pressure Formulation of the Navier-Stokes Equations, *Comput. Fluids*, 5, 25-35, 1978

[68] R. L. Sani, P. M. Gresho, R. L. Lee, and D. F. Griffiths, 'The Cause and Cure (?) of the Spurious Pressures Generated by Certain FEM Solutions of the Incompressible Navier-Stokes Equations, Part 1', *Int .J. Numer. Meth. Fluids*, 1,17-30,1981

[69] T. J. R. Hughes, W. K. Liu and A. Brooks, 'Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation', *J. Comput. Phys*, 30, 1-60, 1979

[70] M. Bercovier and M. S. Engelman, ' A Finite Element for the Numerical Solution of Viscous Incompressible Flows', *J. Comp. Phys.*, 30, 181-201, 1979

[71] J. C. Heinrich and R. S. Marshall, 'Viscous Incompressible Flow by a Penalty Function Finite Element Method', *Comput. Fluids*, 1, 73-83, 1981

[72] M. Fortin and A. Fortin, 'A generalisation of Aszawa's Algorithm for the Solution of the Navier-Stokes Equations', *Commum. Appl. Numer. Methods*, 1, 205-210, 1985

[73] M. S. Engelman, R. L. Sani, P. M. Gresho and M. Bercovier, 'Consistent vs. Reduced Integration Penalty Methods for Incompressible Media Using Several and New Elements', *Int. J. Numer. Meth. Fluids*, 2, 25- , 1982

[74] S.W. Kim and R. A. Decker, 'Velocity-Pressure Integrated versus Penalty Finite Element Methods for High Reynolds Number Flows', *Int. J. Numer. Meth. Fluids*, 9, 43-57, 1989

[75] A. M. Winslow, 'Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Mesh', *J.Comp.Phys.*, 2, 149-172, 1967

[76] D. Williamson, 'Numerical Integration of Fluid Flow over Triangular Grids', *Monthly Weather Rev.*, 97, 885-895, 1969

[77] M. Feistauer, J. Felcman and M. Lukáčová-Medvid'ová, 'Combined Finite Element-Finite Volume Solution of Compressible flow', *Journal of Computational and Applied Mathematics*, 63, 179-199, 1995

[78] V. Dolej, 'Combined finite volume-finite element methods fo compressible flows on unstructured meshes', Ph.D. Thesis, *University Karlova*, Prague, 1998

[79] S. Choudhury and R.A.Nicolaides, 'Discretization of Incompressible Vorticity-Velocity Equations on Triangular Meshes', *Int. J. Numer. Meth. Fluids*, 11, 823-833, 1990

[80] D. Mavriplis, 'Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation', *J.Comp. Phys.*, 90, 271-291, 1990

[81] A. Jameson and D. Mavriplis, 'Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh', *AIAA Paper*, 85-0435, Jan. 1985

[82] A. Lahrmann, 'An Element Formulation for the Classical Finite Difference and Finite Volume Method Applied to Arbitrarily Shaped Domains', *Int. J. Numer. Meth. Fluids*, 11, 893-913, 1992

[83] C. R. Swaminathan and V. R. Voller, 'Streamline Upwind Scheme for Control-Volume Finite Elements, Part I. Formulation', *Numer. Heat Transfer, Part B.*, 22, 95-107, 1992

[84] M. Feistauer, 'Discontinuous Galerkin Method: Compromise between FV and FE Schemes', *3rd International symposium on Finite Volume Methods*, Porquerolles, France, June 24-28, 2002

[85] E. Onate, M. Cervera and O. C. Zienkiewicz, 'A finite volume format for structural mechanics', *Int. Journal for Num. Methods in Engg.*, 37, 181-201, 1994.

[86] Y. D. Fryer, C. Bailey, M. Cross and C. H. Lai, 'A control volume procedure for solving the elastic stress-strain equations on an unstructured mesh', *Appl. Math. Modelling*, 15, 639-645, 1991.

[87] Y. D. Fryer, 'A Control Volume Unstructured Grid Approach to the Solution of the Elastic Stress-Strain Equations', Ph.D thesis, *The University of Greenwich*, 1993.

[88] C. Bailey and M. Cross, 'A finite volume procedure to solve elastic solid mechanics problems in three dimensions on an unstructured mesh', *Int. Journal for Num. Methods in Engg.*, 38, 1757-1776, 1995.

[89] G. A. Taylor, 'A Vertex Based Discretisation Scheme Applied to Material Nonlinearity within a Multi-Physics Finite Volume Framework', PhD thesis, *The University of Greenwich*, 1996.

[90] J. B. Campos Silva and L. F. M. Moura, 'A Control-Volume Finite-Element Method (CVFEM) for Unsteady, Incompressible Viscous Fluid Flows', *Numerical Heat Transfer, B*, 40, 61-82, 2001

[91] B. R. Baliga and S.V. Patankar, 'A Control Volume Finite-Element Method for Two-Dimensional Fluid flow and Heat Transfer', *Numerical Heat Transfer*, 6, 245-261, 1983.

[92] B. R. Baliga, T. T. Pham and S. V. Patankar, 'Solution of Some Two-Dimensional Incompresible Fluid Flow and Heat Transfer Problems', Using a Control Volume Finite-Element Method', *Numerical Heat Transfer*, 6, 245-261, 1983

[93] C. Prakash and S. V. Patankar, 'A Control Volume-Based Finite-Element Method for solving the Navier-Stokes equations using Equal-order velocity-pressure interpolation', *Numerical Heat Transfer*, 8, 259-280, 1985

[94] B. R. Baliga, 'A Control-Volume Based Finite Element Method for Convective Heat and Mass Transfer', Ph.D. thesis, *Dept. of Mech. Eng., University of Minnesota*, Minneapolis, 1978

[95] S. Ramadhyani, 'Solution of the Equations of Convective Heat, Mass and Momentum Transfer by the Finite Element Method Using Quadrilateral Elements', Ph.D thesis, *University of Minnesota*, Minneapolis, 1979

[96] C. Prakash, 'A Finite Element Method for Predicting Flow through Ducts with Arbitrary Cross-Sections', Ph.D. thesis, *University of Minnesota*, Minneapolis, 1981

[97] C. Prakash, 'An Improved Control Volume Finite-Elemetn Method for Heat and Mass Transfer, and for Fluid Flow Using Equal-Order Velocity-Pressure Interpolation', *Numer. Heat. Trans.*, 9, 253-276, 1986

[98] M. J. Raw, 'A New Control-Volume-Based Finte Element Procedure for the Numerical Solution of the Fluid Flow and Scalar Transport Equations', Ph.D. thesis, *University of Waterloo*, Waterloo, Ont., 1985

[99] N. A. Hookey, 'A CVFEM for Two-Dimensional Viscous Compressible Fluid Flow', Ph.D. thesis. *Dept. of Mech. Eng., McGill University*, Montreal, Que., 1989

[100] B. R. Baliga and S. V. Patankar, 'A New Finite Element Formulation for Convection Diffusion Problems', *Numerical Heat Transfer*, 3, 393-409, 1980

[101]] H. J. Saabas and B. R. Baliga, 'Co-located equal-order control-volume finite-element method for multidimensional, incompressible, fluid flow - Part1: Formulation', *Numerical Heat Transfer*, 26, 381-407, 1994

[102] H. J. Saabas and B. R. Baliga, 'Co-located equal-order control-volume finite-element method for multidimensional, incompressible, fluid flow - Part11: Verification', *Numerical Heat Transfer*, 26, 409-424, 1994

[103] G. E. Schneider and M. J. Raw, 'A Skewed Positive Influence Coefficient Upwinding Procedure for Control Volume Based Finite Element Convection Diffusion Computation', *Numerical Heat Transfer*, 9, 1-26, 1986

[104] G. E. Schneider and M. J. Raw, 'A Control Volume Finite - Element Method for Heat Transfer and Fluid Flow Using Colocated Variables', *Numerical Heat Transfer*, 11, 363-390, 1987

[105] M. J. Raw, G. E. Schneider, and V. Hassani, 'A Nine-Noded Quadratic Control Volume Based Finite Element for Heat Conduction', *J.Spacecraft*, 22,523-529,1985

[106] J. Banaszek, 'Comparison of Control Volume and Galerkin Finite Element Methods for Diffusion Type Problems', *Numer. Heat Transfer*, 16, 59-78, 1989.

[107] I. Wintruff, C. Gunther, and A. G. Class, 'An Interface-Tracking Control-Volume Finite-Element Method for Melting and Solidification Problems - Part 1 and 2', *Numerical Heat Transfer, B*, vol. 39, pp. 101-125, 2001.

[108] M. Reyes, J. Rincon and J. Damia, 'Simulation of Turbulent Flow in Irregular Geometries Using a Control-Volume Finite-Element Method', *Numerical Heat Transfer*, B, vol. 39, pp. 79-89, 2001

[109] J. C. Slattery, 'Momentum, Energy and Mass Transfer in Continua', *Robert E. Kreiger Publishing Company*, Huntingdon, New York, Second Edition, 1981

[110] M. R. Hestenes, 'Iterative Methods for Solving Linear Equations', *Journal of Optimization Theory and Applications*, 11(4), 323-334, 1973

[111] R. Fletcher, 'Conjugate Gradient Methods for Indefinite Systems', *Lecture Notes in Mathematics*, 506, 73-89, 1976

[112] J. B. Scarborough, 'Numerical Mathematical Analysis', *Johns Hopkins University Press*, 4th edn, Baltimore, 1958

[113] H. K. Versteeg and W. Malalasekera, 'Computational Fluid Dynamics: The Finite Volume Method', *Longman Scientific and Technical Group*, 1995

[114] R. Courant, E. Isaacon and M. Rees, 'On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Difference', *Communications in Pure and Applied Mathematics*, 5, 243-255, 1952

[115] G. D. Raithby, 'A Critical Evaluation of Upwind Differencing Formulation for Differential Expressions Involving both First and Second Derivatives', *International Journal for Numerical Methods in Engineering*, 4, 551-559, 1972

[116] D. B. Spalding. 'A Novel Finite -difference Formulation for Differential Expressions Involving Both First and Second Derivatives', *Int. J. Numer. Methods Eng*, 4, 551, 1972.

[117] P. H. Gaskell and A. K. C. Lau, 'Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm', *Int.J.Num.Meth.Fluids*, 8, 617, 1988.

[118] J. E. Fromm, 'A method for reducing dispersion in convective difference schemes', *J.Comp.Phys.*, 3, 176, 1968.

[119] B. Koren, 'A robust upwind discretisation method for advection, diffusion and source terms', *In: Numerical Methods for Advection-Diffusion Problems*, Ed. C.B.Vreugdenhil and B.Koren, Vieweg, Braunschweig, 117, 1993.

[120] B. P. Leonard, 'A stable and accurate convective modelling procedure based on quadratic upstream interpolation', *Comp. Methods Appl. Mech. Eng.*, 19, 59, 1979.

[121] B. Van Leer, 'Towards the ultimate conservative difference scheme II', *J.Comp.Phys.*, 14, 361, 1974.

[122] B. Van Leer, 'Towards the ultimate conservative difference scheme V', *J.Comp.Phys.*, 32, 101, 1979.

[123] R. K. Agrawal, 'A third-order accurate upwind scheme for Navier- Stokes at high Reynolds numbers', *AIAA* 81-0112, 1981.

[124] F. S. Lien and M. A. Leschziner, 'Upstream monotonic interpolation for scalar transport with application to complex turbulent flows', *Int. J .Num. Meth. Fluids,* 19, 527, 1994.

[125] J. Zhu and W. Rodi, 'A low dispersion and bounded convection scheme', *Comp. Meth. Appl. Mech. and Engng.*, 98, 345, 1991.

[126] T. Hayase, J. A. C. Humphrey and R. Greif, 'A consistently Formulated QUICK Scheme for Fast and Stable Convergence Using Finite-volume Iterative Calculation Procedures', *J. Comput. Phys.*, 98, 108-118, 1992

[127] N.P.Waterson and H.Deconinck, 'A unified approach to the design and application of bounded higher-order convection schemes', *Numerical Methods in Laminar and Turbulent flow*, IX(1), 203-214, 1995.

[128] M. K. Patel, N. C. Markatos and M. Cross, ' Technical Note - Method of Reducing False-Difusion Errors in Convection-Diffusion Problems', *Applied Mathematical Modelling*, 9, 302-306, 1985

[129] C. Carey, T. J. Scanlon and S. M. Fraser, 'SUCCA - An Alternative Scheme to Reduce the Effects of Multidimensional False Diffusion', *Applied Mathematical Modelling*, 17(5), 263-270, 1993

[130] M. K. Patel, M. Cross and N. C. Markatos, 'An assessment of Flow Oriented Schemes for reducing 'False Diffusion', *Int. J. Num. Meth. Eng.*, 26, 2279-2304, 1988

[131] N. A. Hookey and B. R. Baliga, 'Evaluation and Enhancements of Some Control-Volume Finite-Element Methods', *Numerical Heat Transfer*, 14, 255-293, 1988

[132] U. Ghia, K. N. Ghia and C. T. Shin, 'High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method', *J. Computational Physics*, vol. 48, pp. 387-411, 1982

[133] M. N. Özisk, 'Heat Transfer - A Basic Approach', *McGraw-Hill*, New York, 1985

[134] A. D. Brent, V. R. Voller and K. J. Reid, 'Enthaply-Porosity Technique for Modelling Convection-Diffusion Phase Change: Application to melting of a Pure Metal', *Numerical Heat Transfer*, 13, 297-318, 1988

[135] V. R. Voller and C. R. Swaminathan, 'General Source-Based Method for Solidification Phase Change', *Numerical Heat Transfer, Part B*, 19, 175-189, 1991

[136] M. Cross, S. Johnson and P. Chow, 'Mapping Enthalpy-Based Solidification Algorithms onto Vector and Parallel Architectures', *Applied Mathematical Modelling*, 13, 702-709, 1989

[137] V. R. Voller and C. Prakash, 'A Fixed Grid Numerical Modelling Metholodology for Convection-Diffusion Mushy Region Phase-Change Problems', *International Journal of Heat and Mass Transfer*, 30(8), 1987

[138] J. P. Van Doormal and G. D. Raithby, 'Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows', *Numer. Heat Transfer*, 7, 147-163, 1984

[139] G. de Vahl Davis and I. P. Jones, 'Natural Convection in a Square Cavity: A Comparison Exercise', *International Journal for Numerical Methods in Fluids*, 3, 227-248, 1983

[140] J. Boussinesq, 'Theorie de L'Ecoulement Toubillant', *Mem. Acad. Sci.*, 23(46), 1877

[141] L. Prandtl, 'Uber die ausgebildete Turbulenz', *ZAMM*, 5, 136-139, 1925

[142] E. R. Van Driest, 'On Turbulent Flow Near a Wall', *Journal of the Aeronautical Sciences*, 23, 1007, 1956

[143] T. Cebeci and A. M. O. Smith, 'Analysis of Turbulent Boundary Layers', *Appl. Math. and Mech.*, XV, 1974

[144] B. S. Baldwin and H. Lomax, 'Thin-Layer Appoximation and Algebraic Model for Separated Turbulent flows', *AIAA paper*, 78-257, 1978

[145] L. Prandtl, 'Uber ein neues formelsystem fur die ausgebildete Turbulenz', *Nacr. Akad. Wiss. Gottingen, Math-Phys.*, K1, 6-19, 1945

[146] A. N. Kolmogorov, 'Equations of Turbulent Motion of an Incompressible Fluid', *Izvestia Academy of Sciences, USSR; Physics*, 6, 1 & 2, 56-58, 1942

[147] B. E. Launder and D. B. Spalding, 'Mathematical Models of Turbulence', *Academic Press, London*, 1972

[148] B. E. Launder, G. J. Reece and W. Rodi, 'Progress in the Development of a Reynolds-Stress Turbulence Closure', *Journal of Fluid Mechanics*, 68, 3, 537-566, 1975

[149] M. M. Gibson and B. E. Launder, 'Ground Effects on Pressure Fluctuations in the Atmospheric Boundary Layer', *Journal of Fluid Mechanics*, 86, 491-511, 1978

[150] M. Ciafalo, 'Large-eddy Simulation: Models and Applications', *Adv. Heat Transfer*, 25, 322-408, 1994

[151] S. Ghosal, T. Lund, P. Moin and K. Akselvoll, ' A Dynamic Localization Model for Large-eddy Simulation of Turbulent Flows', *Journal of Fluid Mechanics*, 286, 229-255, 1995

[152] B. E. Launder and D. B. Spalding, 'The Numerical Computation of Turbulent Flows: A Review', *Computers and Fluids*, 15(2), 269-289, 1974

[153] J. Kim, S. J. Kline and J. P. Johnston, 'Investigation of a Reattachment Turbulent Shear Layer: Flow over a Backward-Facing Step', *Transactions of the ASME, Journal of Fluids Engineering*, 102, 302-308, 1980

[154] J. K. Eaton and J. P. Johnston, 'A Review of Research on Subsonic Turbulent Flow Reattachment', *AIAA*, Paper AIAA-80-1438, 1980

[155] M. Nallasamy and C. P. Chen, 'Studies on Effects of Boundary Conditions in Confined Turbulent Flow Predictions', *Technical Report CR-3929*, NASA, 1985

[156] S. A. Syed and L. M. Chiappetta, 'Finite Difference Methods for Reducing Numerical Diffusion in TEACH-Type Calculations', *AIAA*, Paper AIAA-85-0056, 1985

[157] V. Schmitt and F. Charpin, 'Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers', *Experimental Data Base for Computer Program Assessment.* Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138, May 1979