Unstructured Mesh - Finite Volume Algorithms for Swirling, Turbulent, Reacting Flows

Thomas Nicholas Croft

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 Philosopy

This research programme was carried out in collaboration with Rio Tinto Zinc (Australia)

June 3, 1998

Abstract

The work presented in this thesis develops techniques, employing the Finite Volume discretisation method, which allow the numerical simulation of three dimensional heat transfer and fluid flow problems using unstructured meshes. The method solves and stores all variables at the element centres which lowers storage requirements and generally shortens run times compared with the Control Volume-Finite Element approach.

Correction terms are formulated which address two of the main forms of errors caused by mesh skewness.

To allow a generic handling of any unstructured mesh the Cartesian components of velocity are solved under all circumstances. This leads to the requirement to adjust the discretisation of the momentum equations when there is significant flow curvature. The changes are presented in this study both when the position of the flow axis is known prior to the simulation and when its position is known only as a result of the simulation, this being the case when there is more than one source of swirling flow.

These original features contribute to a Computational Fluid Dynamics code which is capable of solving swirling, turbulent fluid flow and reactive, radiative heat transfer on highly complex geometries. Specifically the techniques are applied to the simulation of processes occurring in the direct smelting of iron.

The use of the Finite Volume method makes it relatively easy to employ many techniques and physical models developed for structured codes. The evaluation of the face convective fluxes is effected through the Rhie - Chow interpolation method. The SIMPLE algorithm is used in the pressure - velocity coupling. In the simulation of swirling flows it is shown that both the standard and ReNormalisation Group k-ϵ models fail to accurately predict turbulent effects. An anisotropic hybrid (k-ϵ and mixing length) model is developed which produces excellent numerical results for the flows of interest. The Simple Chemical Reaction Scheme is used to evaluate the transport of the various chemical species. Radiation effects are simulated through the use of the radiosity model. A series of simulation results are presented which show the capabilities of the methods in test cases ranging from simple heat transfer problems through to the simulation of two swirling jets in a three dimensional unstructured mesh.

Acknowledgements

I would like to express my gratitude to my supervisors Professor Mark Cross and Professor Koulis Pericleous for their advice and guidance throughout the study.

Without the drive and financial support provided by Rio Tinto Zinc this research would not have covered as wide a range of topics within the relatively short time scale. I would particularly like to thank Mark Davis for his many contributions to this body of work and for his efforts in trying to improve my report writing skills.

The starting point for this research was the two dimensional, unstructured mesh code UIFS, the CFD part of which was written by Peter Chow. Both the code and the advice of Peter made the preliminary work on creating a three dimensional code much easier than it might have been.

I would also like to thank the numerous colleagues who have contributed to various aspects of this thesis and to my knowledge in general. In particular I must acknowledge John Ewer and Chris Bennett for making it possible to use significant amounts of research software*, the use of which has seriously influenced my academic progress.

Finally I would like to thank my parents for their support through my many years in education.

*It should be noted that this does not refer to any CFD type software

Contents

Appendices

List of Figures

Nomenclature

Roman Letters

a	=	Proportion of radiation absorbed per unit length	97
a_P	=	System matrix diagonal coefficient for element P	20
a_{nb}	=	Coefficient multiplying neighbour nb	20
A_f	=	Area of a face	17
$\underline{\underline{A}}$	=	System Matrix	13
$\underline{b}$	=	Source vector	13
c_p	=	Specific Heat	91
C_R	=	Eddy break–up constant	96
d_{AP}	=	Distance between centres of elements A and P	17
D_f	=	Diffusion Conductance through face f	20
e	=	Proportion of radiation scattered per unit length	97
E	=	Black body emissive power of the fluid	97
E	=	Wall roughness parameter	106
f	=	Liquid fraction	98
f	=	Mixture fraction	95
f_{st}	=	Stoichiometric mixture fraction	93
F_f	=	Strength of convection through face f	20

G	=	Turbulent generation rate	104
h	=	Enthalpy	91
H	=	Heat of reaction	94
$\underline{J}$	=	Total flux of variable ϕ	23
k	=	Thermal Conductivity	91
k	=	Turbulent kinetic energy	96
l	=	Turbulent length scale	108
L	=	Latent heat of solidification	98
m_x	=	Mass fraction of species x	94
$\underline{n}$	=	Unit outward normal	17
p	=	Pressure	28
P	=	Peclet number	22
$\underline{\hat{r}}$	=	Unit vector in radial flow direction	55
$\underline{r}_{ax}$	=	Point on flow axis	55
$r_{1/2}$	=	Jet half width	112
Re	=	Reynolds number	106
R	=	Gas Constant	94
R	=	Radiosity	97
s	=	Stoichiometric ratio	93
s	=	Wall skin friction factor	106
S_C	=	Constant coefficient of linearised source term	16
S_P	=	Linear coefficient of linearised source term	16
t	=	Time	13
T	=	Temperature	92
T_L	=	Liquidus temperature	98
T_S	=	Solidus temperature	98

$\underline{u}$	=	Velocity Vector	13
u_θ	=	Tangential component of velocity	51
u_i	=	i'th component of the Cartesian velocity vector	48
u_r	=	Radial component of velocity	51
u_z	=	Axial component of velocity	51
U_τ	=	Resultant friction velocity	105
$\underline{v}^{cy}$	=	Cylindrical polar velocity vector	51
V	=	Element Volume	14
W	=	Molecular weight of fluid mixture	94
W_x	=	Molecular weight of species x	94
$\underline{\hat{z}}$	=	Unit vector in axial flow direction	55

Greek Letters

α	=	Interpolation Factor	18
Δt	=	Size of time step	15
ϵ	=	Dissipation Rate of Turbulence	96
Γ	=	Diffusion Coefficient	13
κ	=	Karman constant	106
λ	=	Mixing length	112
μ_{eff}	=	Effective dynamic viscosity	48
ν_t	=	Turbulent kinematic viscosity	103
ν_0	=	Background kinematic viscosity	112
ν_m	=	Mixing length model kinematic viscosity	112
ϕ	=	General Scalar Variable	13
$\underline{\phi}$	=	Solution Vector	13
ρ	=	Density	13

σ	=	Stefan-Boltzmann constant	97
σ_t	=	Turbulent Prandtl number	103
τ	=	Wall shear stress	106
θ	=	Cylindrical polar angle	51
$\underline{\hat{\theta}}$	=	Unit vector in tangential flow direction	55

Subscripts and Superscripts

'	=	Correction form of a variable	47
*	=	Current estimate of the solution	47
0	=	Previous time step values	14
A	=	Adjacent element	17
f	=	Face	17
nb	=	Neighbouring Elements	20
P	=	Element currently being discretised	14

CHAPTER ONE

Introduction

§1.1 Objectives and Requirements

The aim of this thesis is to develop Computational Fluid Dynamics (CFD) algorithms which allow the simulation of the swirling, turbulent fluid flow and radiative, reactive heat transfer processes prevalent in a wide range of industrial applications. Amongst the class of processes to which these algorithms could be applied are glass furnace burners(1–3), coal fired power generation(4), automobile carburetor design(5), steel making(6) and power station boilers(7). The immediate area of application of the algorithms is the reaction vessel in the HIsmelt® Process(8–10) which involves the direct smelting of iron ore. The intention is to develop a tool which can be used, with a level of confidence, to quickly determine the relative merits of a variety of model configurations. In this context the methods used must be able to handle any complexity of the physical solution domain and all boundary conditions without significant loss of accuracy. One of the major influences on the flow patterns within the reaction vessel is the presence of one or more mass sources which may have a significant rotational velocity component. A consequence of this is that any algorithm developed as part of this work must be capable of simulating multiple swirling jets in complex geometries. Turbulent effects will be highly significant in the vessel, hence there is requirement for a turbulence model which can simulate the non-isotropic nature of the turbulent eddies in high swirl regions. Additionally, algorithms are required to resolve the chemical reactions, radiation effects and calculate the effects of the metal droplets. Given the problem complexity, and the potentially large number of elements required in the three dimensional simulations of the physical domain, it is necessary to develop an efficient solution procedure, both in terms of simulation time and storage requirements, to minimise the modelling costs.

Over the last 25 years a wide range of physical models have been developed for Finite Volume codes. Many aspects of the modelling of fluid flow and heat transfer in the reaction vessel can be handled by making relatively small changes to pre-existing algorithms. One of the aims of this study is to develop a framework which allows easy conversion of algorithms written for traditional Finite Volume codes, whether fully orthogonal, Body Fitted Coordinate (BFC) or multiblock.

One of the main considerations in the choice of methodology is the need to model multiple swirling jets. In the simulation of swirling flow, knowledge of the position of the axis of fluid rotation is necessary to accurately interpolate the velocity vector. The modelling of swirling jets commonly involves the use of either an axi-symmetric or a cylindrical mesh, the axis of which is aligned with the flow axis, and the solution of the radial, tangential and axial components of velocity. In the case of multiple jets the generation of a mesh which aligns each jet axis with a mesh feature, for example the axis of a polar mesh, is a difficult task even for a code using multiblock, BFC, cylindrical polar meshes, Additionally the interaction of the jets means that the position of the axes of flow rotation are not known prior to the simulation. One method would be to use grid adaption techniques to guarantee that mesh and flow axes coincide throughout the simulation. A better method, in terms of stability and computational effort, would be to use a static mesh and an interpolation scheme which locates the position of the flow axis. Having located the axis a suitable interpolation technique, for the velocity vector, could be applied

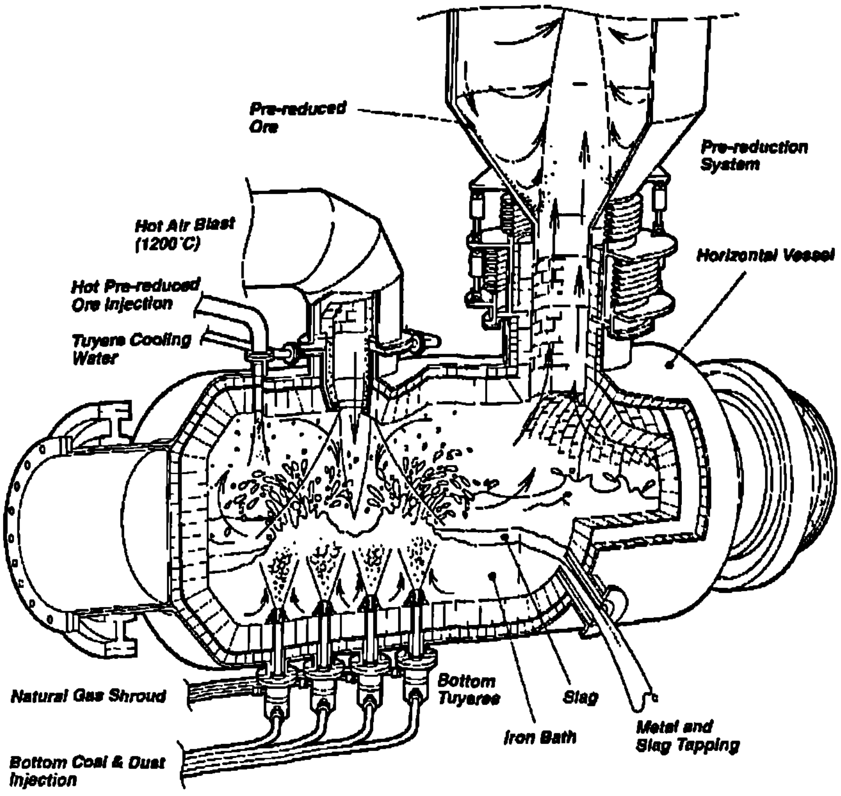

Figure 1.2.1: Conceptual HIsmelt® Smelt Reduction Vessel$^{(9)}$

§1.2 The HIsmelt® Process

The immediate area of application of the algorithms developed as part of this study is the HIsmelt® direct smelting Process. The process concerns the smelting of iron ore to produce reduced iron units which can immediately be used in the steel making process. The Smelt Reduction Vessel (SRV), shown conceptually in Figure 1.2.1, is the centre of the HIsmelt® process. The reduction process involves the cracking of volatiles from coal, metal then dissolves the carbon leading to the ore being smelted and the formation of slag. Because of the overall endothermic nature of these processes there is a need to supply considerable quantities of heat to the bath. This is provided by the combustion processes taking place in the region above the bath zone.

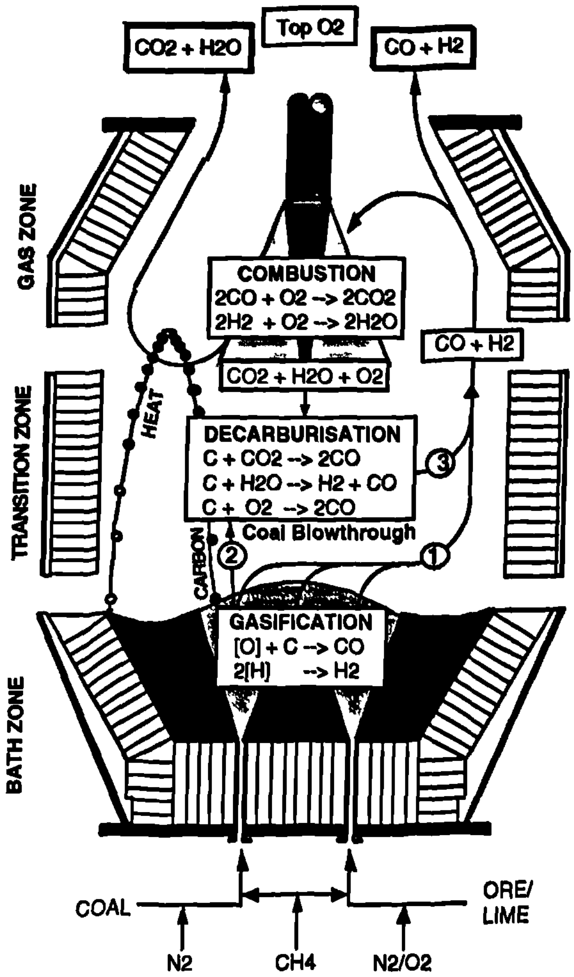

Figure 1.2.2: Post Combustion Stoichiometry[9]

The SRV is divided into three distinct regions, the Bath Zone, the Transition Zone and the Gas Zone to facilitate the modelling of the smelting process. Figure 1.2.2 shows the position of these regions as well as the major reactions occurring in each of them. This study will be concerned only with the Transition Zone and the Gas Zone (details of the Bath Zone modelling can found in[11,12]). In the modelling process there are no differences in the handling of the two zones, together termed the top space, the location of the Transition Zone being one of the results of any simulation rather than a model parameter. The distinguishing feature of the Transition Zone is the presence of droplets which have been ejected from the bath as a result of bottom injection processes. The presence of the droplets acts as a restriction to the flow of the continuum gas phase. In addition, momentum is transferred between the discrete and continuum phases through friction/drag effects.

The combustion processes taking place in the continuum phase mainly involve the oxidation of carbon monoxide to produce carbon dioxide. The oxygen is provided through the main mass source(s) of the gas phase whilst the carbon monoxide is a product of the reactions within the bath. There are also decarburisation processes taking place in the droplets which absorb oxygen and carbon dioxide from the continuum phase and release carbon monoxide. At the temperatures prevalent in the top space radiation becomes an important mechanism for the transport of energy. The presence of droplets can significantly affect the radiating properties of the areas in which they are present. All these aspects must be considered in constructing a model which hopes to accurately predict fluid flow and heat transfer in the top space.

§1.3 Review of Relevant Work

Over the last three decades the use of CFD techniques in the modelling of many physico-chemical applications has grown from being able to model only single phase, low Reynolds number flows to its current level of use in a wide range of applications. This level of growth has been enhanced by the advances in computer technology which have vastly reduced the computational times for all simulations as well as increasing the size of problems which can be solved. With the growing use of parallel computers in the solution of CFD problems it is envisaged that this growth will continue for some time yet.

One of the most commonly used approaches to the discretisation of the transport equations is the Finite Volume (FV) methodology. This approach has the advantage that whatever the transport mechanism, convection, diffusion or other, the conservation of a solved quantity is satisfied through simple approximations to the various terms in the transport equation. This simplicity, combined with the comparatively low storage requirements, have made this approach the favoured technique employed by many CFD modellers.

The earliest applications(13,14) of the Finite Volume approach used fully orthogonal Cartesian meshes. This approach leads to the simplest equations for the various terms contributing to the discretised equations but does suffer in its ability to handle physically realistic solution domains. The use of full and/or partial blockages(15) of elements, in an attempt to sculpt the Cartesian mesh to the physical curved shape, allows a generally more realistic answer to be obtained but there are always doubts about results in partially blocked elements.

The use of Curvilinear Grids(16–21), also termed Body Fitted Coordinates (BFC), allows the physical domain to be accurately fitted for a large number of cases. The mapping of these grids onto their topologically equivalent Cartesian mesh, with the associated mapping of the transport equations, extends the class of problems to which the FV technique can be applied. A similar methodology, in which the transformation to a computational domain is implicit in the discretisation techniques, has been used by Demirdžić and Perić(22) to solve problems with moving boundaries. The problems with this type of approach are that the use of BFC meshes increases the storage requirements and adds considerably to the complexity of the equations being solved. The approximations made to calculate the various terms become significantly more difficult to calculate. This commonly leads to further approximations being made and as a consequence errors become significant if the physical grid differs substantially from the computational Cartesian mesh.

The use of multi-block techniques(23), in which the solution domain consists of a number of BFC type blocks, further extends the abilities of these methods to being able to solve over a complex geometry, without moving from solving on a Cartesian mesh. This method can equal the unstructured approach in its ability to fit the outer surfaces of a solution domain but internal to the domain it lacks the flexibility of being able to use any element type.

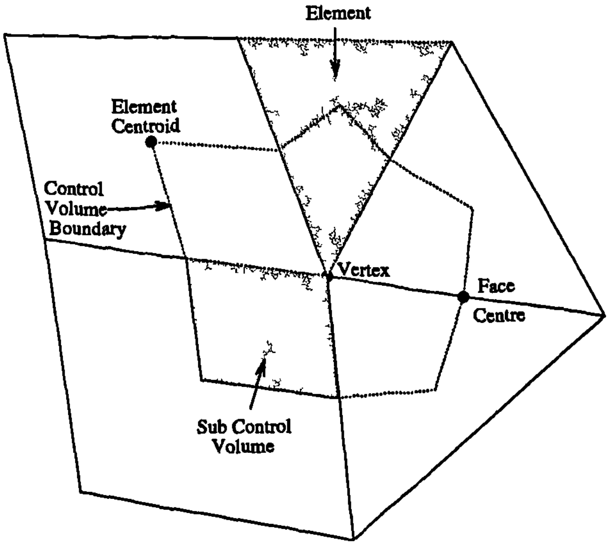

Figure 1.3.1: Control Volume for CV-FE method

The Finite Volume technique has been extended by a number of authors into forms which allow the discretisation of the conservation equations over unstructured meshes. Thomadakis[24] used a staggered mesh approach, solving the velocity components at the element centres and the pressures at the element vertices, to simulate low Reynolds number flow problems. Pan *et al.*[25] used an unstructured mesh consisting of triangular elements to solve a variety of laminar flow problems. Chow[26] employed the FV method, using polygonal element types, triangles through to octagons, to solve simple flow and heat transfer problems. Most of the work on Finite Volume unstructured mesh techniques, including the papers mentioned in this section, have so far been concerned with two dimensional meshes.

An alternative approach is the Control Volume based - Finite Element Method (CV-FE) first used by Winslow[27] to solve magnetostatic problems. In this approach the solution domain is divided into elements as in finite element calculations. A control volume is then formed about each of the vertices, Figure 1.3.1, and the conservation equations are discretised over these control volumes. Shape functions are used to describe the variation of a variable within an element allowing the contributions from the integration points in every sub control volume to be evaluated. The method was extended to the solution of hydrodynamic and heat transfer problems by Baliga and Patankar[28] and many others[29–31]. It has been applied to the solution of problems in the fields of aerodynamics(32), solidification(33) and stress-strain analysis(34). The CV-FE technique does allow the use of a fully unstructured mesh and hence has the ability to accurately model the physical solution domain and allows the use of any element type for which shape functions exist. The draw back in the method is the extra computational effort required to calculate the contributions to the discretised equations. In addition, the discretised linear system of matrices can have significantly more non zero coefficients than when using the FV method. As a consequence, more computational operations are performed in the iterative matrix solver.

In a number of application areas hybrids of the Finite Volume approach have been employed. In the simulation of aerodynamic flows block structured meshes with local unstructured refinement using pentahedral and tetrahedral elements have been employed by Shaw *et al.*(35) and Childs *et al.*(36). Flows involving shock waves were modelled by Feistauer *et al.*(37) using Finite Volume discretisation for the inviscid first order terms and Finite Element approximations for the viscous dissipative terms. Bailey *et al.*(38) used the Finite Volume method to model heat transfer, solidification and residual flow and used the CV-FE method for the solid mechanics to perform simulations of metal casting. In addition, other discretisation techniques exist for use on unstructured meshes, for example the fluctuation-splitting method(39,40) used in the solution of the Euler equation, but the majority of research into the use of unstructured meshes, in the discretisation of the Navier-Stokes equation, involves the use of varieties of either the Finite Volume or CV-FE methods.

The aim of the work presented here is to extend the Finite Volume methodology to the solution of transport equations on three dimensional unstructured meshes. This should lead to an approach which has the advantages of the lower storage requirements and run times, compared to the CV-FE approach, whilst providing the flexibility regarding the types of meshes which can be employed in the spatial discretisation of the solution domain. The methods presented here are an extension of the two dimensional of the work of Chow(26,41). The first stage of the extension to three dimensions formed part of a Masters Thesis(42) which contains the preliminary design for a partial differential equation solver. Although the implementation at that stage employed the CV-FE method, many of the utility routines and data structures, which arose out of the design, form the framework on which CWNN$^{(43)}$, the CFD code in which the algorithms discussed in this thesis have been implemented, was built.

§1.4 Outline of Contents

The methods used in the solution of a generic scalar variable are described in Chapter 2. At the start of this chapter the methods used in the discretisation of each term in a general transport equation are presented. At this stage it is assumed that the mesh over which the conservation equation is being discretised is non skew. The requirement to use a differencing scheme in the evaluation of the convective flux is described in the following section. At this stage a number of standard first order schemes are discussed. When using collocated variables the values of pressure and/or velocity in adjacent elements can become decoupled. The Rhie - Chow$^{(44)}$ interpolation method, used to avoid the decoupling, is described in the next section. The techniques described previously in the chapter have assumed that there is no mesh skewness. The final section in the chapter describes two forms of mesh skewness, namely non-orthogonality and non-conjunctionality, and presents terms which address the errors caused by their presence.

Chapter 3 covers the algorithms specific to the solution of the hydrodynamic variables. Initially the methods used on a Cartesian mesh are discussed. The coupling of the pressure with the velocity components through the use of the SIMPLE algorithm is described next. The following two sections describe the methods used when modelling swirling flows. The first of these sections covers the case when the flow axis can be aligned with the y=0, x=0 line. When using either a cylindrical or an axi-symmetric mesh it is conventional to solve for the cylindrical polar components of velocity. To simplify the extension to the case when the positions of the axes are not known prior to the simulation, the Cartesian components of velocity are solved in CWNN in all circumstances. This section details the effect this choice has on the calculation of the face velocity values and on the contribution from the convection term. The next section in this chapter describes the interpolation method used when the position of the flow axes is not known prior to the simulation. An investigation into the advantages of the use of higher order differencing schemes in the modelling of swirling flows was undertaken. As part of this study a number of three point schemes were introduced into CWNN. Details of the schemes are presented in this Chapter along with test cases which

i. indicate the benefit of their use in solving a general problem.

ii. show their performance in solving swirling flows.

The final section of this chapter describes an investigation into the efficiency of the several linear matrix solvers.

The physical models used in the solution of turbulent, reacting flows are described in Chapter 4. The first section in this chapter describes the computation of the temperature field through the solution of the enthalpy equation. The algorithms used to model the effects of combustion processes, radiation effects and solidification on the temperature field are detailed in this section. The second section in this chapter describes the modelling of turbulence. The standard k-ϵ and ReNormalisation Group k-ϵ (RNG) models are described. These models, because of their isotropic assumptions, fail to accurately predict turbulence levels, and hence the flow field, in regions with significant flow curvature. The final part of this section introduces an anisotropic turbulence hybrid (mixing length, k-ϵ) model, which considerably improves the solution for a class of swirling flow simulations. As the tracking of massless particles forms part of the general interpolation scheme discussed in Chapter 3 some form of particle tracking is required. The implementation of particle tracking algorithms, both for massless and weighted particles, within the framework of unstructured mesh code, is detailed in the final section of this chapter.

1.4. Outline of Contents

The methods described in the Chapters 2-4 of this thesis have been implemented in a code called CWNN. Chapter 5 presents a set of standard validation test cases for the available models. The first set of test cases involves one dimensional heat transfer. The test case shows the errors caused by the use of an unstructured mesh with a high level of skewness. The correction terms are then applied to show the advantages of their use. The second test case involves the simulation of a moving lid cavity. Results from simulations using a structured mesh, unstructured with no correction terms and unstructured with corrections are compared with the standard benchmark solutions. The third case shows a combined heat transfer and fluid flow problem with the simulation of buoyancy driven flow in a cavity. Comparisons are made with benchmark solutions at a variety of Rayleigh numbers. The final case in this chapter involves turbulent flow over a backward facing step.

Chapter 6 describes the simulations performed in the building of a model of the top space in the HIsmelt process. The initial work involved the simulation of swirling flow, both laminar and turbulent. The preliminary work on the laminar flow highlights the need for the adjustments to the discretisation methods when simulating swirling flow. The simulation of turbulent flow in an annulus shows the deficiencies of the varieties of the k-ϵ model. In the next set of test cases the implementation of both the diffusion controlled and kinetically influenced forms of the Simple Chemical Reaction Scheme(45) are compared with numerical results from an alternative source. The final set of test cases show the various stages of modelling of the HIsmelt process starting with an axi-symmetric simulation of fluid flow, through a full three dimensional simulation of two jets with chemical reactions to an axi-symmetric two-phase simulation using the full top space model.

In Chapter 7 the conclusions of this study are stated. The chapter also includes details of a number of areas where additional work could lead to further advances to the methods presented in this thesis.

CHAPTER TWO

Discretisation Methods for a Transported Scalar

The aim of this chapter is to introduce the Finite Volume techniques which are employed in the discretisation of the general conservation equation on an unstructured mesh. The methods described in this chapter can be applied to any equation describing the transport of a scalar quantity, for example temperature, concentration, turbulent kinetic energy. Under certain assumptions these methods can also be applied to the momentum conservation equations. For the purposes of this chapter it will be assumed that the momentum equations have been solved and the current estimates to the velocity and pressure fields have been calculated. A discussion of the special aspects of the momentum equation will be presented in the following chapter.

Initially it will be assumed that the mesh over which the conservation equation is being discretised is fully non skew. A non skew mesh is one in which for every pair of adjacent elements

- the line connecting the centroids of the adjacent elements is parallel to the normal of the face between the elements, termed orthogonality.
- the line connecting the centroids of the adjacent elements intersects the face between the elements at the face centroid, termed conjunctionality by the author.

Each term in the conservation equation is discussed in turn. The use of differencing schemes in the handling of the joint diffusion - advection contribution is reviewed. Special attention is paid to the convection term where the Rhie and Chow Interpolation method[44] is used to calculate face velocities. The difficulties encountered with the use of this algorithm within an unstructured mesh framework are discussed. The terms which are necessary when there is mesh skewness are defined at the end of this chapter. These terms address the problems of two types of skewness, namely non-orthogonality and non-conjunctionality.

§2.1 Discretisation on a Non Skew Mesh

The aim, of the various methods discussed within this chapter, is that given a conservation equation, for a scalar quantity ϕ, of the form

$$\underset{\text{Transient}}{\frac{\partial(\rho\phi)}{\partial t}} \quad + \quad \underset{\text{Convection}}{\mathrm{div}(\rho\underline{u}\phi)} \quad = \quad \underset{\text{Diffusion}}{\mathrm{div}(\Gamma_\phi \mathrm{grad}(\phi))} \quad + \quad \underset{\text{Source}}{S_\phi} \qquad (2.1.1)$$

suitable approximations can be made to each term in (2.1.1) which allows it to be expressed as a linear matrix equation of the form

$$\underline{\underline{A}}\,\underline{\phi} = \underline{b} \qquad (2.1.2)$$

where $\underline{\phi}$ is a vector of the values of ϕ at a finite number of selected points, such that the solution of (2.1.2) gives as close an approximation as possible to the solution of (2.1.1).

The techniques used to discretise the various conservation equations are based on the cell centred Finite - Volume formulation. In this method the domain over which the equation is to be solved is divided into a set of non-overlapping polyhedral control volumes. At the centre of each of these control volumes a single node is positioned. This set of nodes form the selected points at which the value of ϕ is sought. The Finite Volume method involves integrating the conservation equation, (2.1.1), over each control volume, as well as over time, and a solution is sought which makes each of these integrals equal to zero. Assumptions are made as to the variation, between nodes, of the quantities involved in the equation. For each control volume this leads to a linear equation involving the unknown values of the scalar quantity at the node in the control volume and the nodes in the neighbouring control volumes. Considering all such linear equations together produces a matrix equation of the form (2.1.2).

In the following sections the discretisation methods applied to each of the terms in the conservation equation are described in turn.

§2.1.1 Transient Term

Integrating the transient term over time and a control volume leads to the need to approximate

$$\int_{t-\Delta t}^{t} \int_{V} \frac{\partial(\rho\phi)}{\partial t}\, dV\, dt \tag{2.1.3}$$

If the mesh is stationary, V does not change with time, then the order of the integrals can be reversed. Using the superscript 0 to indicate values at the previous time step equation (2.1.3) becomes

$$\int_{V} (\rho\,\phi - \rho^0\,\phi^0)\, dV \tag{2.1.4}$$

Assuming that values at the centre of the control volume, signified by the subscript P, are representative of the average value in the control volume this equation is approximated by

$$V_P\,\left(\rho_P\,\phi_P - \rho_P^0\,\phi_P^0\right) \tag{2.1.5}$$

This particular study will use stationary meshes in all simulations but the above discretisation can be easily generalised to allow the possibility of mesh movement or adaption. In the case of a moving mesh it can be shown(46) that as long as the velocity used in the convection term is the relative velocity

$$\underline{v} - \underline{v}_m$$

where $\underline{v}$ is the velocity of the continuum and $\underline{v}_m$ is the mesh velocity then the integrated form of the transient term becomes

$$\int_{t-\Delta t}^{t} \frac{d}{dt} \int_V \rho\phi \, dV \, dt \qquad (2.1.6)$$

Using the same approximations as for a static mesh this is equal to

$$\int_V \rho\phi \, dV \Big|_{t-\Delta t}^{t} \qquad (2.1.7)$$

Again assuming the value of a variable at the centre of a control volume is representative equation (2.1.7) can be discretised to produce

$$V_P \, \rho_P \, \phi_P - V_P^0 \, \rho_P^0 \, \phi_P^0 \qquad (2.1.8)$$

For every other term in the conservation equation fully implicit assumptions are used. This means that the time integration of the remaining terms simply leads to a multiplying factor of Δt. The whole discretised equation is divided by this multiplying factor in order that the choice of time step only affects the contribution from the transient term. Thus the final form of the discretised transient term is

$$\frac{V_P \, \rho_P \, \phi_P - V_P^0 \, \rho_P^0 \, \phi_P^0}{\Delta t} \qquad (2.1.9)$$

In the following sections, describing the discretisation of the remaining terms, the integration over time will be ignored.

§2.1.2 Source Term

When the source term is discretised it is initially expressed in a linearised form(47)

$$S_\phi = S_C - S_P\,\phi \tag{2.1.10}$$

where S_C and S_P can be functions of any stored value including ϕ. To maintain diagonal dominance of the resulting equation the value of S_P must not be negative. The equations used to evaluate the values of S_C and S_P can significantly affect both the rate of convergence and stability of the solution procedure. For speed as large a portion of the source as possible should be placed in the linear, $S_P\,\phi$, part. This speed will be compromised if the size of S_C is also increased. Stability will be affected if changes in any aspect of the solution results in large changes in the values of S_C and S_P. For example, a temperature source of the form $T_d^2 - T^2$, where T_d is a stored value, could be linearised into the forms

$$\begin{aligned} S_C &= T_d^2 - T_l^2 & S_P &= 0.0 \\ S_C &= (T_d + T_l)\,T_d & S_P &= (T_d + T_l) \\ S_C &= T_d^2 & S_P &= T_l \end{aligned} \tag{2.1.11}$$

where the subscript l signifies a previous iteration value. The methods at the top of the list potentially lead to a very stable, but slow convergence rate. The later methods would be expected to be quicker but less stable.

When the linearised source term is integrated over the control volume all terms are evaluated at the centre of the element to give a contribution

$$V_P\,(S_C - S_P\,\phi_P) \tag{2.1.12}$$

§2.1.3 Diffusion Term

The first step in the discretisation of the diffusion term is to use the divergence theorem

$$\int_V \text{div}(\underline{F})\, dV = \int_S \underline{F}.\underline{n}\, dS \tag{2.1.13}$$

to convert the volume integral into the surface integral

$$\int_V \text{div}(\Gamma_\phi \text{grad}(\phi))\, dV = \int_S \Gamma_\phi \text{grad}(\phi).\underline{n}\, dS \tag{2.1.14}$$

where $\underline{n}$ is the unit outward normal to the surface. This equation can be simplified by recognising that $\text{grad}(\phi).\underline{n}$ is equal to the gradient of ϕ in the direction of the normal. The surface of any polyhedral control volume consists of a set of faces. Consequently equation (2.1.14) can be written as a sum of surface integrals over each face bounding the control volume.

$$\sum_f \int_f \Gamma_\phi \frac{\partial \phi}{\partial n} dS \tag{2.1.15}$$

For a fully orthogonal mesh the line connecting the nodes, in the control volumes either side of face f, is parallel to the normal to face. Let A be the element, adjacent to element P, which also has face f as one of its sides 2.1.1.e at the Then the normal gradient is approximately equal to

$$\frac{\partial \phi}{\partial n} \simeq \frac{\phi_A - \phi_P}{d_{AP}} \tag{2.1.16}$$

where d_{AP} is the distance between the centroids of elements A and P. Hence the discretised form of equation (2.1.15) is

$$\sum_f (\Gamma_\phi)_f A_f \left(\frac{\phi_A - \phi_P}{d_{AP}}\right) \tag{2.1.17}$$

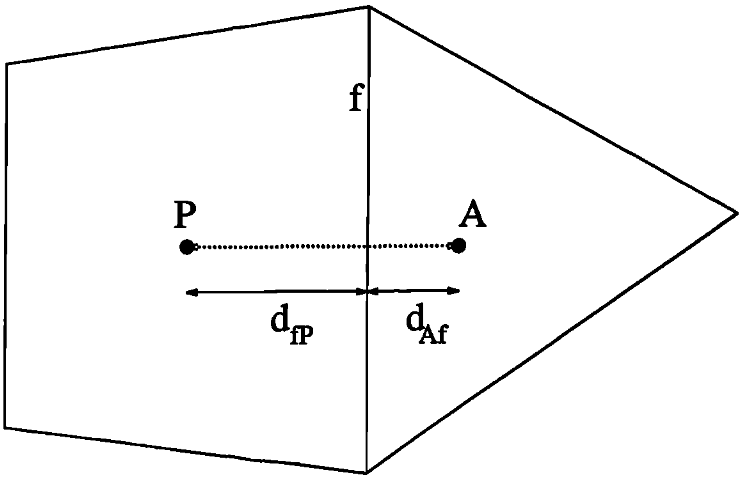

Figure 2.1.1: Adjacent Control Volumes

where A_f is the area of face f. The only remaining decision is how to calculate Γ_ϕ on the face. Its value could be calculated by means of an arithmetic mean i.e.

$$\begin{aligned}(\Gamma_\phi)_f &= \alpha_f(\Gamma_\phi)_P + (1-\alpha_f)(\Gamma_\phi)_A \\ \alpha_f &= \frac{d_{Af}}{d_{Af}+d_{fP}}\end{aligned} \tag{2.1.18}$$

This method suffers from the drawback that if $(\Gamma_\phi)_A$ is equal to zero it is to be expected that there would be no flux of ϕ at the interface of the control volumes containing P and A whilst equation (2.1.18) approximates a value for Γ_ϕ between the nodes as $\alpha_f(\Gamma_\phi)_P$ which will not normally be the expected 0. Similarly if $(\Gamma_\phi)_A$ is relatively much less than $(\Gamma_\phi)_P$ there would be relatively little resistance to the flux of ϕ between P and the interface compared to that between A and the face. In this case it would be expected that $(\Gamma_\phi)_f$ would depend on $(\Gamma_\phi)_A$ and inversely on α_f whereas equation (2.1.18) would lead to $(\Gamma_\phi)_f = \alpha_f(\Gamma_\phi)_P$. A better model for the variation of Γ_ϕ between the nodes is to use the harmonic mean

$$(\Gamma_\phi)_f = \frac{(\Gamma_\phi)_A(\Gamma_\phi)_P}{\alpha_f(\Gamma_\phi)_P + (1-\alpha_f)(\Gamma_\phi)_A} \tag{2.1.19}$$

This formula gives $(\Gamma_\phi)_f = 0.0$ if either $(\Gamma_\phi)_A$ or $(\Gamma_\phi)_P$ is zero and for $(\Gamma_\phi)_P >> (\Gamma_\phi)_A$

$$(\Gamma_\phi)_f \approx (\Gamma_\phi)_A/\alpha_f$$

as required.

§2.1.4 Convection Term

The following discussion will use the symbol $\underline{u}$ to represent the relative velocity of the fluid. For the cases of interest in this study, which all use a static mesh, the relative velocity is equal to the fluid velocity. When using a moving mesh this relative velocity is equal to the fluid velocity minus the mesh velocity. Use of the relative velocity means that the discretisation of the convection term is consistent with the methods used to discretise the transient term in section 2.1.1.

As with the diffusion term the divergence theorem is used to transform the volume integral of the convection term into a surface integral

$$\int_V \mathrm{div}(\rho\,\underline{u}\,\phi)\,dV \;=\; \int_S \rho\,(\underline{u}\,.\,\underline{n})\,\phi\,dS \tag{2.1.20}$$

The surface integral is again split into a set of integrals over each of the faces bounding the control volume and each of the values in the integrand are estimated on the face to give

$$\sum_f \rho_f\,(\underline{u}\,.\,\underline{n})_f\,A_f\phi_f \tag{2.1.21}$$

In this equation the value of ρ_f is given the value in the upwind element. Thus

$$\begin{aligned} \rho_f &= \rho_P \qquad && \text{if } (\underline{u}\,.\,\underline{n})_f > 0.0 \\ &\text{and} && \\ \rho_f &= \rho_A \qquad && \text{if } (\underline{u}\,.\,\underline{n})_f < 0.0 \end{aligned} \tag{2.1.22}$$

The Rhie - Chow interpolation method, used to evaluate the normal component of velocity at the face, $(\underline{u}\,.\,\underline{n})_f$, will be described later in this chapter. To complete the discretisation of the convection term some approximation has to be made regarding the calculation of the face value of ϕ. One possible method is to use arithmetic averaging

$$\phi_f = \alpha_f \, \phi_P + (1-\alpha_f)\, \phi_A \qquad (2.1.23)$$

The problem with this choice and a selection of alternatives will be discussed in the following section on differencing schemes. Assuming that arithmetic averaging is being used the discretised form of the convection term becomes

$$\sum_f \rho_f \, (\underline{u} . \underline{n})_f \, A_f \, [\alpha_f \, \phi_P + (1-\alpha_f)\, \phi_A] \qquad (2.1.24)$$

§2.2 Differencing Schemes

Having obtained expressions for the discretised form of each of the terms in the conservation equation in the previous section, the discretised form of the full equation is obtained by simply adding together all these contributions. If only the convection and diffusion terms are considered, and using arithmetic averaging in the evaluation of the face value of ϕ in the convection term, the discretised equation becomes

$$\sum_f A_f \left[\rho_f \, (\underline{u} . \underline{n})_f \, \{\alpha_f \, \phi_P + (1-\alpha_f)\, \phi_A\} + (\Gamma_\phi)_f \left(\frac{\phi_P - \phi_A}{d_{AP}} \right) \right] = 0 \qquad (2.2.1)$$

The quantities F_f and D_f are now introduced where

$$F_f = A_f \, \rho_f (\underline{u} . \underline{n})_f \qquad D_f = A_f \, (\Gamma_\phi)_f / d_{AP} \qquad (2.2.2)$$

F_f is the strength of the convection of ϕ and D_f is the diffusion conductance. Now (2.2.1) can be rewritten as

$$a_P \phi_P = \sum_{nb} a_{nb} \, \phi_{nb} \qquad (2.2.3)$$

where the summation is over all neighbouring elements. The equations for the coefficients in equation (2.2.3) are

$$\begin{aligned} a_{nb} &= D_f - (1-\alpha_f)F_f \\ a_P &= \sum_f (D_f + \alpha_f F_f) \\ &= \sum_{nb} a_{nb} + \sum_f F_f \end{aligned} \qquad (2.2.4)$$

The discretisation techniques can be applied to the steady state continuity equation

$$\mathrm{div}(\rho \underline{u}) = 0 \qquad (2.2.5)$$

which is a special case of the general conservation equation (2.1.1), with ϕ equal to 1.0 and Γ_ϕ equal to 0.0. Substitution of these values of Γ_ϕ and ϕ in equation (2.2.1) gives

$$\sum_f F_f = 0 \qquad (2.2.6)$$

Putting this equality into equation (2.2.4) gives

$$a_P = \sum_{nb} a_{nb}$$

Difficulties arise in obtaining the solution of this equation as the coefficients are not guaranteed to satisfy the condition that they remain positive. This can lead to the solution of the discretised equation becoming unbounded, $\phi_P > \phi_{nb}$ for all neighbours, which is physically unrealistic in the absence of a source (P. 83 Patankar$^{(47)}$ for this and other objections).

In order to avoid the problems observed in the previous formulation a better assumption is sought for the variation of ϕ between adjacent nodes. In the upwind scheme, first suggested by Courant, Isaacson and Rees$^{(48)}$, there is no alteration to the handling of the diffusion term but the convection term uses the upwind nodal value of ϕ for the estimated interface value. Thus

$$\begin{aligned} \phi_f &= \phi_P && \text{if } F_f > 0.0 \\ &&& \text{and} \\ \phi_f &= \phi_A && \text{if } F_f < 0.0 \end{aligned} \tag{2.2.7}$$

In this case the coefficients in (2.2.3) are

$$\begin{aligned} a_{nb} &= D_f + \max(-F_f, 0.0) \\ a_P &= \sum_f [D_f + \max(F_f, 0.0)] \\ &= \sum_{nb} a_{nb} + \sum_f F_f \end{aligned} \tag{2.2.8}$$

The upwind approach guarantees that all the coefficients remain positive and consequently leads to a bounded solution. Unfortunately the scheme over predicts diffusive effects leading to numerical smearing$^{(49)}$.

If the exact solution of the one dimensional convection - diffusion problem

$$\frac{d}{dx}(\rho u \phi) = \frac{d}{dx}\left(\Gamma_\phi \frac{d\phi}{dx}\right) \tag{2.2.9}$$

is sought over a region ranging from 0 to L with the boundary conditions

$$\begin{aligned} \text{At} \quad x &= 0 \qquad \phi = \phi_0 \\ \text{At} \quad x &= L \qquad \phi = \phi_L \end{aligned} \tag{2.2.10}$$

the solution is

$$\frac{\phi - \phi_0}{\phi_L - \phi_0} = \frac{\exp(Px/L) - 1}{\exp(P) - 1} \tag{2.2.11}$$

where P is the Peclet number

$$P \equiv \frac{\rho u L}{\Gamma_\phi} \tag{2.2.12}$$

which is the ratio of the strength of the convection to the strength of the diffusion. It can be seen that for approximately zero values of the Peclet number the problem reverts to one of pure diffusion. In this case the solution given by (2.2.11) means that the variation of ϕ between element centroids is nearly linear. When the flow is in the positive direction, u and hence P being positive, the values of ϕ in the domain are influenced more by ϕ_0, which is the upstream value of ϕ. Similarly for negative flows the value of ϕ is more dependent on its value at L which again is the upstream value. As P increases in either the positive or a negative manner the value of ϕ within the domain becomes more strongly influenced by the relevant upstream value of ϕ. If |P| increases then dϕ/dx tends to 0 at the centre of the region indicating that diffusion is nearly absent. The upwind scheme always uses a linear relationship between ϕ and x and hence at large |P| values it overestimates the influence of the diffusion term[47].

Let $\underline{J}$ represent the total flux which is equal to the sum of the diffusion flux and the convective flux. Then

$$\underline{J} = \rho\underline{u}\phi - \Gamma_\phi \text{grad}(\phi) \tag{2.2.13}$$

and so the convection - diffusion equation can be written as

$$\text{div}(\underline{J}) = 0 \tag{2.2.14}$$

which after integration over a control volume transforms into the discretised equation

$$\sum_f A_f(\underline{J}.\underline{n})_f = 0 \tag{2.2.15}$$

In the exponential scheme the exact solution of the equation as shown in equation (2.2.11) is used as a profile between points P and A. The suffices 0 and L are replaced by P and A and d_{AP} replaces L. Under this assumption the expression for $(\underline{J}.\underline{n})_f$ is

$$(\underline{J}.\underline{n})_f = F_f\left(\phi_P + \frac{\phi_P - \phi_A}{\exp(P_f) - 1}\right) \tag{2.2.16}$$

where

$$P_f = \frac{F_f}{D_f} = \frac{\rho_f(\underline{u}.\underline{n})_f}{(\Gamma_\phi)_f} \tag{2.2.17}$$

Using equation (2.2.16) the coefficients in the discretised equation become

$$\begin{aligned} a_{nb} &= \frac{F_f}{\exp(P_f) - 1} \\ a_P &= \sum_{nb} a_{nb} + \sum_f F_f \end{aligned} \tag{2.2.18}$$

This scheme produces the exact solution for the steady state one dimensional problem given any value of the Peclet number. The problems with this differencing scheme are that exponentials are expensive in terms of computational time and for greater than one dimensions the scheme is not exact and so the computation time cannot be justified.

If the equations (2.2.18) are considered it can be seen that as the Peclet number increases to positive infinity then

$$a_{nb}/D_f \longrightarrow 0.0$$

and as the Peclet number goes to negative infinity

$$a_{nb}/D_f \longrightarrow -P_f$$

and when the Peclet number is close to zero

$$a_{nb}/D_f = 1.0 - 0.5P_f$$

The hybrid scheme, developed by Spalding(50), uses three straight line sections to approximate these properties. For $P_f <$ -2 it uses

$$a_{nb}/D_f = -P_f$$

for $|P_f| < 2$

$$a_{nb}/D_f = 1.0 - 0.5P_f$$

and for $P_f > 2$

$$a_{nb}/D_f = 0.0$$

This means that

$$\begin{aligned} a_{nb} &= \max\left(-F_f, D_f - \frac{F_f}{2}, 0\right) \\ a_P &= \sum_{nb} a_{nb} + \sum_f F_f \end{aligned} \qquad (2.2.19)$$

It should be noted that when the Peclet number is in the range -2 to 2 then the hybrid scheme reduces to the central difference scheme and outside this range it uses a modification of the upwind scheme where the diffusion has been set to zero. In this way the shortcomings of the upwind scheme are not shared by the hybrid scheme.

The hybrid scheme provides a good approximation to the exact solution but there is a rather large difference around $|P_f| = 2$ due to the fact that the diffusion is set to zero at these values. The power law scheme, described by Patankar[51], provides a better approximation by means of a slightly more complex algorithm which is not that expensive to compute. The power law uses

$$\begin{aligned} a_{nb}/D_f &= -P_f && \text{for } P_f < \text{-10} \\ a_{nb}/D_f &= (1+0.1P_f)^5 - P_f && \text{for -10} < P_f < 0 \\ a_{nb}/D_f &= (1-0.1P_f)^5 && \text{for } 0 < P_f < 10 \\ a_{nb}/D_f &= 0.0 && \text{for } P_f > 10 \end{aligned}$$

It can be seen that for $|P_f| > 10$ the power law and the hybrid schemes are identical. The power law scheme gives an extremely close approximation to the exponential scheme and because of its relatively low computational expense it offers an attractive

Scheme	Formulae for $A(\|P\|)$
Central Differencing	1 - 0.5 $\|P\|$
Upwind	1
Hybrid	max (0, 1-0.5 $\|P\|$)
Power Law	max(0, (1 - 0.1 $\|P\|$)^5)
Exponential	$\|P\|$ / exp($\|P\|$) - 1

Table 2.1: Definition of Differencing Schemes

alternative.

The discretised convection - diffusion equation can now be written in a form independent of the differencing scheme being used

$$\sum_f \left[D_f\, A(|P_f|) + \max(-F_f,\, 0.0)\right](\phi_P - \phi_A) + F_f\,\phi_P \;=\; 0 \qquad (2.2.20)$$

where the formulae used for the function $A(|P|)$ for the various first order differencing schemes are given in Table 2.1. A considerable amount of effort has been expended by many researchers, including$^{(52-54)}$, to build and improve on these schemes. Section 3.5 looks into the merits of using higher order schemes, which use values of ϕ in elements other than the immediate neighbours of the face, to estimate the value of ϕ on the face.

The ideas contained in this section can now be combined to discretise the full conservation equation for a scalar, equation (2.1.1). Combining the discretised forms of each term in the equation gives

$$\frac{(\rho_P\phi_P V_P - \rho_P^0\phi_P^0 V_P^0)}{\Delta t} + \sum_f \left[\rho_f\,(\underline{u}.\underline{n})_f\,\phi_f\,A_f \;-\; (\Gamma_\phi)_f\left(\frac{\phi_A - \phi_P}{d_{AP}}\right)A_f\right] \;=\; (S_C - S_P\,\phi_P)\,V_P \qquad (2.2.21)$$

Using the same arguments which led to equation (2.2.20) the preceding equation can be written as

$$\frac{(\rho_P \phi_P V_P - \rho_P^0 \phi_P^0 V_P^0)}{\Delta t} + \sum_f [\{D_f\, A(|P_f|) + \max(-F_f,\, 0.0)\}\, (\phi_P - \phi_A) + F_f\, \phi_P] = (S_C - S_P\, \phi_P)\, V_P \qquad (2.2.22)$$

The continuity equation

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho\, \underline{u}) = 0 \qquad (2.2.23)$$

can similarly be discretised to give

$$\frac{\rho_P V_P - \rho_P^0 V_P^0}{\Delta t} + \sum_f F_f = 0 \qquad (2.2.24)$$

Multiplying equation (2.2.24) by ϕ_P and substituting into equation (2.2.22) gives an equation of the form

$$a_P\, \phi_P = \sum_{nb} a_{nb}\, \phi_{nb} + b_P \qquad (2.2.25)$$

where the summation is over all elements which share a face with element P. The coefficients in equation (2.2.25) are calculated from the formulae

$$\begin{aligned} a_{nb} &= D_f\, A(|P_f|) + \max(-F_f,\, 0.0) \\ b_P &= S_C V_P + \frac{V_P^0\, \rho_P^0}{\Delta t}\, \phi_P^0 \\ a_P &= \sum_{nb} a_{nb} + \frac{V_P^0\, \rho_P^0}{\Delta t} + S_P\, V_P \end{aligned} \qquad (2.2.26)$$

Equation (2.2.25) is the final form of the discretised equation which can now be solved, after decisions are made as to how to handle faces coinciding with the boundaries of the solution domain, to give the values of ϕ at all the element centroids.

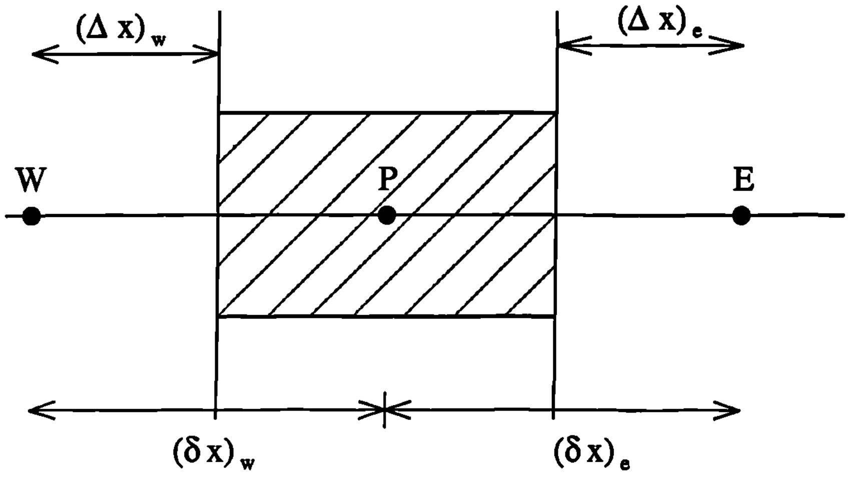

Figure 2.3.1: One Dimensional Control Volume

§2.3 Rhie - Chow Interpolation Method

If the one dimensional x momentum equation is considered

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u u)}{\partial x} = \frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right) - \frac{\partial p}{\partial x} \tag{2.3.1}$$

where u is the x component of velocity, then the pressure derivative term must be discretised over the control volume. If the control volume considered contains the node P with neighbouring nodes W and E as in Figure 2.3.1 then the contribution to the discretised equation from this term is p_e - p_w. Assuming that the faces e and w are midway between the nodes either side of them, then using linear interpolation for the face values of pressure

$$p_w - p_e = \frac{p_W + p_P}{2} - \frac{p_P + p_E}{2} = \frac{p_W + p_E}{2} \tag{2.3.2}$$

This indicates that the pressure differential term in the momentum equation leads to a discretised form which is a relationship between alternate nodal pressure values rather than adjacent ones. As a consequence the pressure differential is calculated over a coarser mesh than the one used for all other quantities. It also means that an oscillatory pressure field, in which, for example, the pressure at consecutive grid points follows the sequence 1,100,1,100,1 etc, would be treated as a uniform pressure field as alternate grid points have the same value. In higher dimensions the problem leads to the checkerboard effect where in each of the dimensions alternate grid points have the same pressure value but not adjacent nodes. This means that in three dimensions the pressure field could take 8 different values but still the discretised pressure differential term would give a zero pressure differential. Given any pressure field this checkerboard field can be added to it without changing the pressure gradient values. Thus if p is the correct pressure field and u the correct velocity field which together satisfy the discretised equations, then so does $p + \lambda p_c$ and u where p_c is the checkerboard pressure field. Any process has only one correct pressure field at any time. This means that any pressure obtained as a result of solving the momentum and continuity equations may predict the correct velocity whilst having an error in the pressure field equal to λp_c.

Consider now the steady one dimensional continuity equation for an incompressible flow, which is

$$\frac{du}{dx} = 0 \tag{2.3.3}$$

Integrating this over the control volume of Figure 2.3.1, and assuming that the faces lie midway between their nodes, the following equation is obtained

$$u_e - u_w = 0 \tag{2.3.4}$$

or

$$\frac{u_P + u_E}{2} - \frac{u_W + u_P}{2} = 0 \tag{2.3.5}$$

or

$$u_E - u_W = 0 \tag{2.3.6}$$

The problems of an equation which tests alternate nodal values, rather than adjacent ones, is again obtained. Hence, similar to the pressure, a checkerboard velocity field would satisfy the continuity equation.

The standard mechanism for handling the above problem is the use of a staggered mesh. The staggered grid for the velocity components was first suggested by Harlow and Welch[55] and forms the basis of the SIMPLE procedure[14] and of the SIVA procedure[13]. In the staggered grid the u velocity is calculated at the faces of the control volume which have their normal in the x coordinate direction. Similarly the v and w velocities are calculated at the faces of the control volume with normals in the y and z directions respectively. The calculation of the velocities at the control volume faces means that in equation (2.3.6) the values of u_e and u_w are obtained directly from the solution of the u component of velocity. This means that the discretised continuity equation now consists of a relationship between adjacent u values and so the checkerboard velocity pattern is no longer possible. Similarly the staggered grid leads to the pressure differential between adjacent nodes becoming the driving force of the velocity component solved for between these nodes.

This staggering of the velocity components is of huge benefit on structured meshes but the method does not extend easily to unstructured meshes. In the approach presented here all quantities are solved and stored at the element centroid. The face values of the velocity components have to be calculated from these element based values. This leads to the need to employ an alternative interpolation method, to those described earlier in this section, which does not suffer from the checkerboard effect. The Rhie - Chow interpolation method[44] offers one approach which satisfies these requirements.

The u momentum equation

$$\frac{\partial(\rho u)}{\partial t} + \mathrm{div}(\rho \underline{u} u) = -\frac{\partial p}{\partial x} + \mathrm{div}(\mu\,\mathrm{grad}(u)) + S \qquad (2.3.7)$$

can be discretised, using the techniques described earlier in this chapter, over the control volume about a node P to produce an equation which can be written in the form

$$a_P u_P + (\nabla_x p)_P \;=\; (\textstyle\sum a_{nb} u_{nb})_P + S_P \qquad (2.3.8)$$

where $\nabla_x p$ is the discretised contribution from the pressure gradient term. Similarly for the adjacent node A

$$a_A u_A + (\nabla_x p)_A = (\sum a_{nb} u_{nb})_A + S_A \tag{2.3.9}$$

From the conservation principle of the control volume formulation the u velocity at a point on the face between the nodes must also have a discretised momentum equation of the form

$$a_f u_f + (\nabla_x p)_f = (\sum a_{nb} u_{nb})_f + S_f \tag{2.3.10}$$

The Rhie - Chow interpolation method uses the equations (2.3.8) and (2.3.9) to approximate a solution of (2.3.10). It is assumed that the right hand side of (2.3.10) may be approximated by using a weighted linear interpolation of the corresponding terms in (2.3.8) and (2.3.9). Thus

$$a_f u_f + (\nabla_x p)_f = \overline{(\sum a_{nb} u_{nb})_f + S_f} = \overline{a_f u_f} + \overline{(\nabla_x p)_f} \tag{2.3.11}$$

where the overline in the above equation indicates a weighted linear interpolation of the variable. Assuming that $a_f \approx \overline{a_f}$ then

$$u_f = \overline{u_f} + \overline{d_f}(\overline{\nabla_x p_f} - \nabla_x p_f) \tag{2.3.12}$$

where if α is the weighting factor used

$$\begin{aligned} \overline{u_f} &= \alpha u_P + (1-\alpha) u_A \\ \overline{\nabla_x p_f} &= \alpha \nabla_x p_P + (1-\alpha)\nabla_x p_A \\ \nabla_x p_f &= A_f n_x (p_A - p_P) \\ a_f &= \alpha a_P + (1-\alpha) a_A \\ \overline{d_f} &= a_f^{-1} \end{aligned} \tag{2.3.13}$$

All that is now required to complete the Rhie - Chow interpolation method is to select the weighting factor α. The obvious choice is that α should be equal to the distance of the node in element A to the face divided by the distance from the same

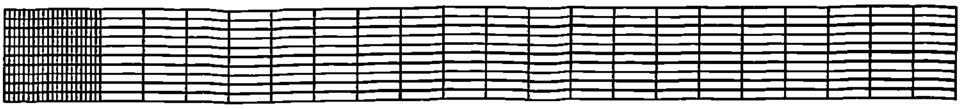

Figure 2.3.2: Mesh Used for Rhie - Chow Weighting Test Case

node to the node in element P. The problem with this choice was highlighted when solving two dimensional flow in a duct. A very fine mesh was used near the inflow boundary in an attempt to accurately model the formation of the flow profile. In the region beyond this fine mesh a much coarser mesh was used, see Figure 2.3.2. In this region the velocity on the centre line of the duct does not change and the pressure decreases at a constant rate. Using weighted linear interpolation leads to the centre line values of the velocity component along the duct shown in Figure 2.3.3. As a comparison a plot from Flow3d[(56)], Version 2.3.2, is also shown on the graph. Flow3d, like CWNN, uses Rhie and Chow interpolation and collocated variables but, unlike CWNN, it uses weighted averaging in the calculation of the convection fluxes. It can be seen that immediately upon changing to the coarser mesh the value of the velocity exhibits a physically unrealistic decrease. Examining the terms of the Rhie - Chow interpolation method for this case, assuming the decrease takes place after the profile has been fully formed, should give $u_A = u_P = u_f$. This means that the correction terms should be equal to zero. As the pressure gradient is constant, C, at this point

$$\begin{aligned}(\nabla_x p)_P &= V_P C \\ (\nabla_x p)_A &= V_A C\end{aligned} \tag{2.3.14}$$

where V_P is the volume of element P. Hence

$$\overline{(\nabla_x p)_f} = C \left\{ \alpha V_P + (1-\alpha) V_A \right\} \tag{2.3.15}$$

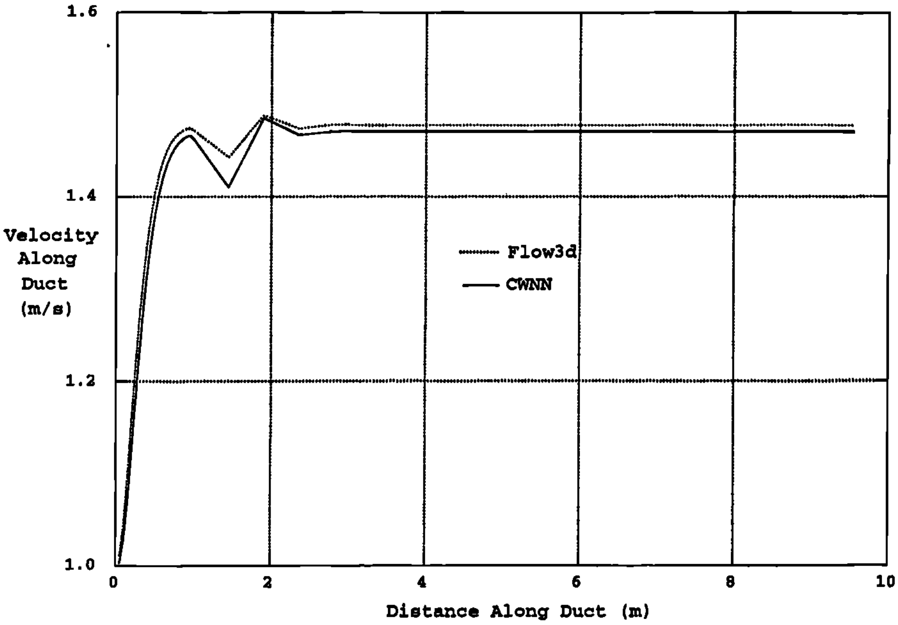

Figure 2.3.3: Rhie - Chow Weighting Test Case, Velocity Along Duct

The second correction term gives

$$\begin{aligned}(\nabla_x p)_f &= (p_A - p_P)\,A_f \\ &= (p_P + d_{AP}\,C - p_P)\,A_f\end{aligned} \qquad (2.3.16)$$

where d_{AP} is the distance between the nodes in elements A and P. As these nodes lie at the centroids of the elements, for the mesh used in this simulation

$$d_{AP} = \frac{0.5\,V_P + 0.5\,V_A}{A_f} \qquad (2.3.17)$$

and so

$$(\nabla_x p)_f = C\left(\frac{V_P}{2} + \frac{V_A}{2}\right) \qquad (2.3.18)$$

Unless the value of α is equal to 0.5 there will be a non zero contribution from the correction term, which is equal to equation (2.3.15) minus equation (2.3.18). This shows that the weighting factor used in the pressure gradient correction term should be 0.5 rather than distance weighted interpolation. For consistency the value of the

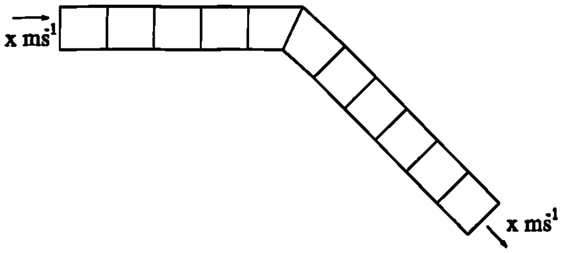

Figure 2.3.4: Rhie - Chow Test Case

weighting factor is set to 0.5 in all terms. This choice needs to examined further to ascertain the validity of the use of the factor 0.5 in each term. A search of relevant literature found only one reference to the choice of weighting factors in Rhie - Chow interpolation. In the CFX4[57], Version 4.1, manual it is stated that the code uses weighted interpolation on the velocity and 0.5 factors in the pressure gradient terms.

§2.3.1 Unresolved Problems with Rhie - Chow Interpolation

The current implementation of the Rhie - Chow interpolation scheme in CWNN works well for Cartesian or near Cartesian meshes. Difficulties can be exhibited when there is mesh skewness. A simple test case, which shows the problem with Rhie - Chow interpolation, is a single row of elements with a bend part of the way along the row, Figure 2.3.4, the cross sectional area of the row being the same before and after the bend. Symmetry conditions are applied to all the boundaries except the two ends of the pipe. Flow enters the pipe through the left hand boundary and leaves through the right hand boundary. The analytic solution to this problem is a velocity vector in the direction of the pipe both before and after the change of direction. The numerical results for the velocity field obtained from a simulation of this case are shown in Figure 2.3.5. The expected flow along the pipe, after the change of direction, has not been predicted. If the convection fluxes are examined the flux across each face internal to the domain is the same. This means that continuity is satisfied throughout the solution domain but at a cost to the element based velocities. Putting more elements across the flow reduces the errors but never eliminates it completely. This shows that Rhie - Chow interpolation guarantees that

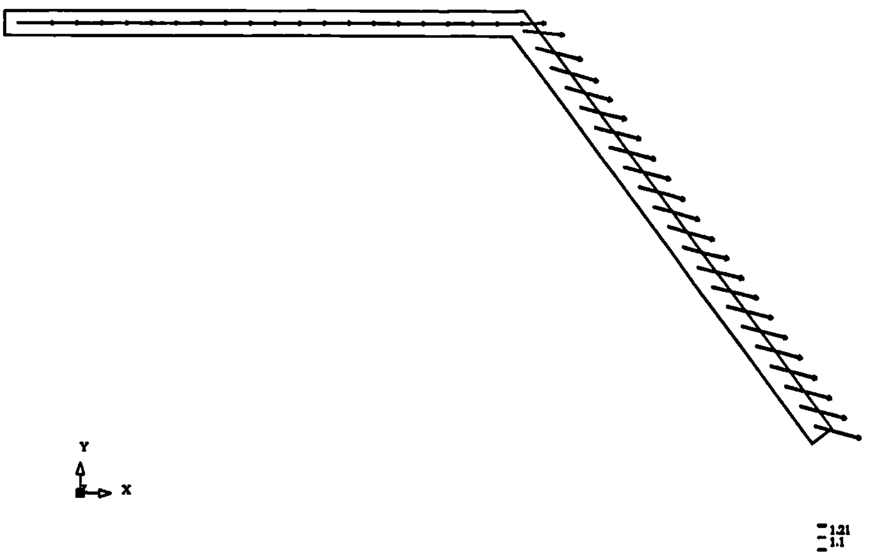

Figure 2.3.5: Line of Elements, Velocity Vectors

the face convective fluxes satisfy continuity even if the cell centre values of velocity do not. This problem has also been reproduced in the Flow3d[56] code, Version 2.3.2, which also uses the Rhie - Chow interpolation method.

The second point regarding the implementation of Rhie - Chow interpolation concerns the evaluation of the pressure gradient for the imaginary control volume about the face. Figure 2.3.6 shows two triangular control volumes, A and P, separated by face f. If it is assumed that the pressure gradient is constant, with value a, and in

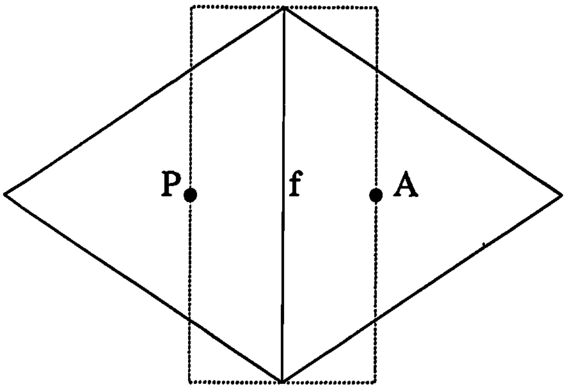

Figure 2.3.6: Control Volume About Face

the direction of the normal to the face between A and P then

$$(\nabla_n p)_P \;=\; (\nabla_n p)_A \;=\; \frac{1}{2} h\, b\, a \tag{2.3.19}$$

assuming both elements have base area b and height h. For the control volume about the face the estimate of the value $(\nabla_n p)_f$ is given by

$$b\,(p_A - p_P) \;=\; b\left(\left(p_P + \frac{2}{3} h\, a\right) - p_P\right) \;=\; \frac{2}{3} h\, b\, a \tag{2.3.20}$$

This means that the combined correction terms from the average of the gradients in the adjacent elements minus the estimate for the gradient for the imaginary face element gives a non-zero value. An example of a flow satisfying the assumptions made in this analysis is flow along a duct after a profile has been formed. In this case the velocity in each element would be the same, if n is in the direction of the flow, and the velocity on the face should be given by linear interpolation of the velocity component without the correction suggested above.

The above are just two areas of concern regarding Rhie - Chow Interpolation method. A number of other points can be raised including which terms to use, or drop, in the interpolation of the conservation equations. When these problems were first encountered an extensive literature search failed to provide any suggestions for corrective methods which could be used in the Rhie - Chow interpolation method.

§2.4 The Cross Diffusion Terms

When using an unstructured mesh it is possible to accurately fit a mesh to any solution region without the use of cell blockages. In addition, features can be placed in the mesh to aid the solution procedure, for example a polar region can be positioned near an inflow where there is a rotational component to the fluid velocity. Fitting meshes to complex boundaries or around regions where specific mesh characteristics have been positioned can frequently lead to mesh skewness. There are several types of mesh skewness ranging from non-orthogonality, the definition of which will be given in the following sections, through elements which are not convex and ultimately leading to elements which have bounding faces which intersect each other. The process of generating a mesh should not allow elements which have intersecting faces or are not convex. A good tool will also reduce the level of other types of skewness but for highly complex geometries there will nearly always be some degree of skewness in the generated mesh. To accurately model the transfer of conserved quantities through a region where there is mesh skewness requires corrections to be made to the usual discretisation of the transport equations. To simplify these corrections the skewness of a mesh is split into a number of contributing components. Correction terms are presented in the remainder of this section for two aspects of skewness. These two components of skewness are termed non-orthogonality, which affects only the diffusion terms, and non-conjunctionality.

§2.4.1 Non-Orthogonality

One of the main potential problems with an unstructured mesh is in the handling of the diffusion term. Integration of the diffusion term over a control volume leads to the necessity to estimate the derivative of ϕ with respect to the face normal. In the structured case the line connecting the nodes on either side of a face lies along the face normal. This allows the derivative to be estimated by use of the formula

$$\frac{\partial \phi}{\partial n} = \frac{\phi_A - \phi_P}{d_{AP}} \tag{2.4.1}$$

In the non-orthogonal case the line connecting the two adjacent nodes is no longer parallel to the face normal vector, i.e. the angle θ in figure 2.4.1 is no longer 90 degrees, and so the above formula needs to be amended. If v represents the vector along the line connecting the adjacent nodes then

$$\frac{\partial \phi}{\partial v} = \frac{\phi_A - \phi_P}{d_{AP}} \tag{2.4.2}$$

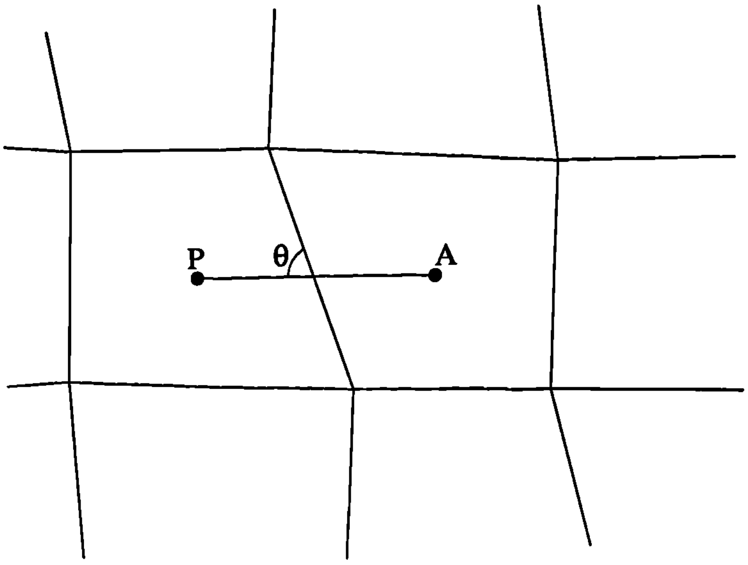

Figure 2.4.1: Non-Orthogonal Control Volume

The normal vector can be written as a component in the $\underline{v}$ direction plus a tangential component of the form

$$\underline{n} = (\underline{v}.\underline{n})\underline{v} + \underline{\beta} \tag{2.4.3}$$

and so it is possible to express the differential with respect to the normal in terms of the differential with respect to the vector connecting the nodes

$$\frac{\partial \phi}{\partial n} = (\underline{v}.\underline{n})\frac{\partial \phi}{\partial v} + \beta\frac{\partial \phi}{\partial t} \tag{2.4.4}$$

where t is the unit vector in the direction $\underline{\beta}$. Now it is necessary to express the differential with respect to t in terms of relatively easily calculated quantities. Obviously $\underline{\beta}$ can be written as a sum of its Cartesian components and so equation (2.4.4) can be expressed in the form

$$\frac{\partial \phi}{\partial n} = (\underline{v}.\underline{n})\frac{\partial \phi}{\partial v} + \beta_x\frac{\partial \phi}{\partial x} + \beta_y\frac{\partial \phi}{\partial y} + \beta_z\frac{\partial \phi}{\partial z} \tag{2.4.5}$$

The problem now reduces to the calculation of the Cartesian derivatives of ϕ on the control volume faces. In order to calculate the Cartesian derivatives of ϕ at a node the derivative is integrated over the control volume about the node in the following ways,

$$\int_V \frac{\partial \phi}{\partial x} dV = V \left\{ \frac{\partial \phi}{\partial x} \right\}_P \tag{2.4.6}$$

and making use of the divergence theorem

$$\begin{aligned} \int_V \frac{\partial \phi}{\partial x} dV &= \int_S \phi n_x dS \\ &= \sum_f A_f \phi_f n_x \end{aligned} \tag{2.4.7}$$

Together these give an equation for the nodal value of the derivative in terms of a sum of the face values of ϕ. Linear interpolation between nodal values of the derivatives allows the face values to be calculated.

The non-orthogonality terms affect the discretised equation by changing the strength of diffusion to

$$D_f = A_f \frac{(\Gamma_\phi)_f}{d_{AP}} (\underline{v}.\underline{n})_f \tag{2.4.8}$$

and there is an extra source equal to

$$A_f (\Gamma_\phi)_f \left(\beta_x \frac{\partial \phi}{\partial x} + \beta_y \frac{\partial \phi}{\partial y} + \beta_z \frac{\partial \phi}{\partial z} \right) \tag{2.4.9}$$

The benefit of the use of the orthogonality corrections is demonstrated in the results chapter, Chapter 5, of this thesis. The use of a non-orthogonal mesh is shown to introduce errors into the simulation results. The inclusion of the non-orthogonality corrections terms produces a significant improvement in the accuracy of the answer. This improvement in the numerical prediction is shown both for a heat transfer problem, section 5.1.2, and for a fluid flow problem, section 5.2.2.

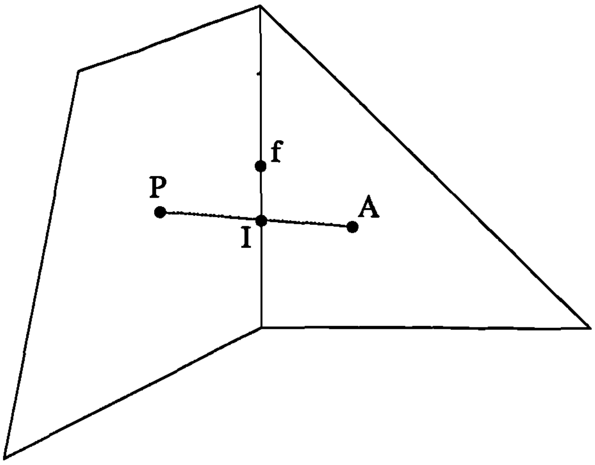

Figure 2.4.2: Non Conjunctional Control Volume

§2.4.2 Non-Conjunctionality

In the evaluation of most of the terms contributing to the discretised equation it is necessary to estimate face values of a number of quantities. This estimated value should be an average value for the whole of the face. At most stages of the discretisation procedure it is assumed that variables change linearly between two points. If this is assumed for the face then the average value of the quantity will be found at the face centroid. Usually the estimation of the face value is achieved through interpolation, in most cases using distance weighting, along the line connecting the nodes in the elements either side of the face. In many meshes the line connecting the centres of the elements either side of the face does not pass through the centre of the face. The term "non-conjunctional" is used here to represent this form of mesh skewness. Figure 2.4.2 shows a case where the face intersection point, I, of the line connecting adjacent nodes, does not coincide with the face centre, f.

The non-conjunctionality correction terms start from the interpolated value at the intersection point and use extrapolation based on the gradients of the quantity to obtain an estimate at the face centre

$$\phi_f \;=\; \phi_I + \underline{d}_{If} \,.\, \mathrm{grad}\phi \qquad (2.4.10)$$

where $\underline{d}_{If}$ is a vector from the intersection point to the face centre. Problems could be encountered in the evaluation of the gradients of ϕ, equations (2.4.6) and (2.4.7), as the calculation of the gradients require the knowledge of face values i.e.

$$\frac{\partial \phi}{\partial x} = \frac{1}{V}\sum_f A_f n_x \left(\phi_I + \sum_j \left[\frac{\partial \phi}{\partial x_j} \right]_f (d_{If})_j \right) \qquad (2.4.11)$$

where $[\partial\phi/\partial x_j]_f$ is calculated using interpolation of its value in the elements either side of face f. This dependence could be resolved by transforming equation (2.4.11) and the similar equations for the y and z gradients into a 3 by 3 matrix equation for the unknown element gradients. The solution of this equation would require the inversion of the matrix which would be computationally expensive for a very large mesh. An alternative method is used, in which the gradients are stored and previous iteration values are used on the right hand side of equation (2.4.11). This correction is not applied to any of the gradient terms as the most commonly used assumption is that quantities vary linearly between points so gradients of gradients would be zero leading to no correction.

The errors caused by this form of mesh skewness are shown in an example of the heat bar test cases contained in Chapter 5. The example also shows the benefit of applying the above corrections.

CHAPTER THREE

Solution Procedure for Fluid Flow

In the previous chapter the Finite Volume methods used in the discretisation of the differential equation describing the transport of a conserved variable, ϕ, on an unstructured mesh were described. These methods assume that the flow field is known whereas in the general case the flow field must be calculated by solving a number of such conservation equations. As the momentum transport equations are of the same form as the differential equation in section 2.1, with $\phi = u, v$ or w and $\Gamma = \mu$, it would appear to be possible to use the methods described previously to obtain the flow field. An iterative solution method would handle the non-linearity of the convection term but the problem arises due to the pressure gradient source term. If a pressure field had been obtained then there would be little difficulty in solving the momentum equations. The problem is that there is no conservation equation in which the pressure replaces ϕ in the general equation.

This chapter examines the techniques used in obtaining both the velocity and pressure fields. Initially the methods employed in the calculation of the pressure field are examined. The next three sections of this chapter examine the methods used to calculate the velocity field. The first of these sections explains the techniques used when there is little or no flow curvature. The following two sections examine the problems associated with flow curvature, firstly when the axis about which the flow is rotating can be aligned with a mesh axis and secondly when the position of the flow axis is calculated during the simulation.

In Chapter 2 the commonly used first order differencing schemes were discussed. These schemes are often subject to inaccuracies, associated with numerical diffusion, which cause problems in the resolution of sharp gradients. In addition it has been claimed[58] that the use of some higher schemes can be advantageous in the simulation of swirling flows on unstructured meshes. A number of three point schemes are investigated to ascertain their merits in the solution of cases involving both conventional and swirling flows.

Once the conservation equation has been discretised the resulting set of linear equations has to solved. The final section, in this chapter, discusses a selection of iterative matrix solvers which can be used to solve the set of equations. The speed, both in terms of iterations and CPU time, of the solvers is compared in the solution of a linear heat transfer problem. The relevance of this test case to the simulation of a coupled problem is discussed at the end of the chapter.

§3.1 Momentum - Pressure Coupling

One possible approach to the evaluation of the velocity field is the stream function / vorticity method in which the pressure is eliminated from the momentum equations by cross differentiation. In two dimensions it is possible to define a stream function which reduces the set of two momentum and the continuity equation to two equations. This method works well in a number of cases but some vorticity boundary conditions are difficult to specify. In addition, it is not easy to obtain the pressure field from the solved for variables. The major disadvantage of this method is in three dimensional problems for which a stream function can not be defined and the set of three momentum and the continuity equation transform into six vorticity equations. This makes the method computationally expensive and so an alternate method is sought which in some manner uses the continuity equation to obtain the pressure field.

If the correct pressure field is used in the momentum equations then the calculated velocities will satisfy the continuity equation. This indirect definition of the pressure field by the continuity equation would aid a direct solution method but it is of little use for the iterative methods used here.

The approach taken is that of the SIMPLE algorithm[14]

§3.1.1 The SIMPLE Algorithm

The continuity equation

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \underline{u}) = 0 \tag{3.1.1}$$

discretises in the form

$$\frac{\rho_P V_P - \rho_P^0 V_P^0}{\Delta t} + \sum_f A_f (\rho \underline{u} . \underline{n})_f = 0 \tag{3.1.2}$$

where the superscript 0 signifies the previous time step value, Δt is the time step and the summation is over all faces of the control volume signified by the subscript P. Rhie - Chow interpolation gives the equation for a face velocity component as

$$u_f = \overline{u_f} + \frac{1}{a_p}\left(\overline{\nabla_x p_f} - \nabla_x p_f\right) \tag{3.1.3}$$

where the overline indicates linear interpolation of the relevant quantity between the element centre values in the elements either side of face f. In equation (3.1.3) the $\nabla_x p_f$ term is approximated by $A_f n_x (p_A - p_P)$ where the subscripts P and A represent the current and adjacent elements respectively. At any solution stage, given estimated element centre values for pressure, p^*, and u velocity, u^*, the face value of the u component of velocity is given by

$$u_f^* = \overline{u_f^*} + \frac{1}{a_p}\left(\overline{\nabla_x p_f^*} - \nabla_x p_f^*\right) \tag{3.1.4}$$

The aim is to improve the guessed pressure p^* in order that the starred velocity components get progressively closer to satisfying the continuity equation. If p is the correct pressure then

$$p = p^* + p' \tag{3.1.5}$$

where p' is the pressure correction. Similarly the starred velocities are related to their correct values by

$$u = u^* + u' \qquad v = v^* + v' \qquad w = w^* + w' \tag{3.1.6}$$

If equation (3.1.4) is subtracted from equation (3.1.3) then

$$u'_f = \overline{u'_f} + \frac{1}{a_p}\left(\overline{\nabla_x p'_f} - \nabla_x p'_f\right) \tag{3.1.7}$$

is obtained. The use of all terms in equation (3.1.7) would produce an equation in which the pressure correction in an element is directly dependent on corrections in both neighbouring elements and elements adjacent to these neighbours. This would lead to the need to solve a linear matrix with a much larger number of non zero elements per row than any of the matrices constructed in the solution of the other solved variables. In order to simplify the resulting pressure correction equation, the first two terms in equation (3.1.7) are dropped. At convergence the terms which have been ignored would be zero so this assumption does not affect the final answer, only the way in which the final answer is approached. Hence

$$\begin{aligned} u'_f &= -\frac{1}{a_p}\left(\nabla_x p'_f\right) \\ &= \frac{1}{a_p} A_f n_x (p'_P - p'_A) \end{aligned} \tag{3.1.8}$$

Finally substituting equation (3.1.6), using equation (3.1.8) for the correction term, into the discretised continuity equation (3.1.2) gives

$$\sum_f \rho_f A_f^2 \frac{n_i^2}{a_i} (p'_P - p'_A)_f = \frac{\rho_P^0 V_P^0 - \rho_P V_P}{\Delta t} - \sum_f A_f \rho_f (\underline{u}^* . \underline{n})_f \tag{3.1.9}$$

where the subscript, i, in the first term indicates a summation over the three coordinate directions. The above equation can be written in the form

$$a_P p'_P + \sum_{nb} a_{nb} p'_{nb} = b_P \tag{3.1.10}$$

where the summation in equation (3.1.10) is over all elements sharing a face with element P and

$$\begin{aligned} a_{nb} &= -\rho_f A_f^2 \sum_{i=1}^{3} \frac{n_i^2}{a_i} \\ a_P &= -\sum_{nb} a_{nb} \\ b_P &= \frac{\rho_P^0 V_P^0 - \rho_P V_P}{\Delta t} - \sum_f A_f \rho_f (\underline{u}^*.\underline{n})_f \end{aligned} \tag{3.1.11}$$

These equations for the coefficients lead to a set of linear equations with weak diagonal dominance, the diagonal being at least as large as the sum of absolute values the off diagonal elements. Boundary conditions, or a fixed reference pressure point, will guarantee diagonal dominance on some rows. Consequently[47], the matrix equation can be solved using the same iterative techniques, discussed in section 3.6, which are employed to obtain the solution for any solved scalar variable.

Having solved equation (3.1.10) to obtain the pressure corrections the pressure field can be updated using equation (3.1.5). The approximations used in the method frequently lead to an over-estimation of the value of the corrections so they are usually relaxed, $p = p^* + \gamma p'$ where γ is usually set to 0.6 but may be much less at the early stages of high Reynolds number flows.

Equation (3.1.8) gives an equation to calculate the face velocity corrections. The problem is that the velocity correction need to be applied to the element centre values rather than these face values. If the discretised form of the momentum equation is considered

$$a_P u_P = \sum_{nb} a_{nb} u_{nb} + b - \nabla_x p_P \tag{3.1.12}$$

then at any stage given a guessed pressure field, p^*, an estimated velocity field, v^*, can be calculated

$$a_P u_P^* = \sum_{nb} a_{nb} u_{nb}^* + b - \nabla_x p_P^* \tag{3.1.13}$$

subtracting equation (3.1.13) from (3.1.12) and using equation (3.1.5) and (3.1.6) an equation is obtained expressing the velocity correction in the element as a function of neighbouring velocity corrections and the integrated gradient of the pressure correction in the element

$$a_P u'_P = \sum_{nb} a_{nb} u'_{nb} - \nabla_x p'_P \tag{3.1.14}$$

The summation term is dropped at this stage as its retention would require the solution of another set of linear equations to obtain the velocity corrections. This leads to the formula

$$\begin{aligned} u'_P &= -\frac{1}{a_P} \nabla_x p'_P \\ &= -\frac{1}{a_P} \sum_f A_f n_x p'_f \end{aligned} \tag{3.1.15}$$

for the velocity correction. Expanding the right hand side gives the following equation for the velocity correction in terms of pressure corrections in the element and all its neighbours.

$$u'_P = -\frac{1}{a_P} \sum_{nb} n_x A_f \left(\alpha_f p'_P + (1 - \alpha_f) p'_{nb} \right) \tag{3.1.16}$$

If the pressure correction is relaxed below 0.6 it was found advantageous to relax the velocity correction to retain a 1:0.6 relationship between velocity and pressure relaxation factors.

§3.2 Cartesian Interpolation

The conservation equation for the Cartesian velocity component u_i can be written in the form

$$\frac{\partial(\rho u_i)}{\partial t} + \operatorname{div}(\rho \underline{u} u_i) = -\frac{\partial p}{\partial x_i} + \operatorname{div}(\mu_{eff}\,\operatorname{grad} u_i) + S_{u_i} \tag{3.2.1}$$

The areas affected by the choice of the interpolation method, used in the calculation of the velocity vector on a face, are

1 the interpolation of the velocity term in the Rhie - Chow interpolation method.

2 the formulation of the convection term in (3.2.1).

Cartesian interpolation can be used when there is little or no curvature in the flow. When using Cartesian interpolation the first term in the Rhie - Chow interpolation method uses linear interpolation of the respective Cartesian component. Hence, for the velocity component in the y direction, v, the method uses

$$\overline{v_f} = \alpha_f v_P + (1-\alpha_f) v_A \tag{3.2.2}$$

to calculate the face value of v.

The same holds for the convection term

$$A_f \rho_f (\underline{u}.\underline{n})_f v_f \tag{3.2.3}$$

In the above equation the value of v_f is calculated using the equation

$$v_f = \alpha_f v_P + (1-\alpha_f) v_A \tag{3.2.4}$$

This is exactly the same form of interpolation as used for a general scalar variable allowing identical methods to be used in the solution procedure. Thus when using

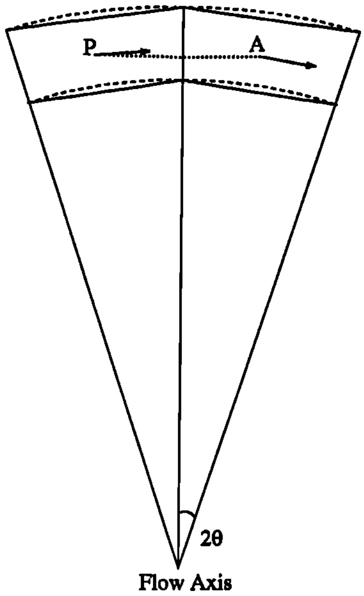

Figure 3.3.1: Use of Cartesian Interpolation with a Rotating Flow Field

Cartesian interpolation the three components of velocity can be treated as three independent scalar variables in the solution of the fluid flow.

§3.3 Cylindrical Interpolation

If one considers a forced vortex flow field, the analytic solution of which is given by[59],

$$\begin{aligned} u_r &= 0 \\ u_\theta &= \Omega\, r \end{aligned} \tag{3.3.1}$$

the use of Cartesian interpolation to estimate the u velocity value on the face between elements A and P (shown in Figure 3.3.1) gives

$$\begin{aligned} u_f &= \alpha_f\, u_P + (1-\alpha_f)\, u_A \\ &= 0.5\, \cos(\theta)\, (u_\theta)_P + 0.5\, \cos(\theta)\, (u_\theta)_A \\ &= \Omega\, r\, \cos(\theta) \end{aligned} \tag{3.3.2}$$

In the above analysis it is assumed that the mesh used is as close an approximation to a cylindrical polar mesh as possible whilst using straight sided, polygonal, elements.

Such a mesh would guarantee that the centroids of both the elements and the face centroid are equal distances from the flow axis. For such a mesh it would be expected that

$$u_f = \Omega r \tag{3.3.3}$$

which may be considerably different from equation (3.3.2) if θ is large. Because the intersection point of the line connecting the centroids of elements A and P with the face does not go through the face centroid the "non-conjunctionality" terms presented in the previous chapter could be used. In this case the distance between the intersection point and the face centroid is

$$r - r\cos(\theta)$$

and the u velocity gradient between the intersection point and the face centroid, this being the radial direction, is

$$\Omega$$

Hence the use of "non-conjunctionality" terms would give an estimate for the face velocity of

$$\begin{aligned} u_f &= \alpha_f u_P + (1 - \alpha_f)\, u_A + \text{Non-Conjunctional Terms} \\ &= \Omega \cos(\theta)\, r + (r - r\cos(\theta))\, \Omega \qquad (3.3.4) \\ &= \Omega r \end{aligned}$$

which is equal to the analytic value. The difficulty with the use of the "non-conjunctionality" terms is that they are always calculated from previous iteration values and placed in the source term. The use of any quantity which is a function of previous iteration values usually slows convergence. A preferred alternative would be to develop a method which adjusts the system matrix to account for curvature effects.

If the flow rotates about an axis, the position of which is known prior to the simulation, then interpolation of the cylindrical polar components of velocity can be used. In a code employing a structured or Body Fitted Coordinate (BFC) cylindrical mesh it is usual to solve for the cylindrical components of velocity. In an unstructured mesh code, when the mesh may bear little or no resemblance to a cylindrical mesh, the advantages of solving for the cylindrical components are far outweighed by the disadvantages. Even block structured codes, such as PHOENICS(60), find the handling of cylindrical coordinate systems problematic. Large numbers of extra terms are required to correctly handle BFC cylindrical meshes. These become even more complex when handling rotating flow using a general unstructured mesh. As a consequence, the Cartesian velocity components are solved for, whilst using interpolation of the cylindrical components of velocity to estimate face velocity values. This use of cylindrical interpolation leads to a necessity to adjust the Rhie - Chow interpolation method and the convection term.

When using cylindrical interpolation the face velocity vector is calculated using linear interpolation of the cylindrical components of velocity. If $\underline{v}^{cy}$ is the velocity vector consisting of the cylindrical velocity components, $(u_r,\, u_\theta,\, u_z)^T$ then

$$(\underline{v}^{cy})_f \;=\; \alpha_f\,(\underline{v}^{cy})_P \;+\; (1-\alpha_f)\,(\underline{v}^{cy})_A \tag{3.3.5}$$

Because it is the Cartesian components of velocity which are solved, not the cylindrical components, the face values of the Cartesian components need to be calculated to evaluate velocity gradients and the convection flux of the momentum components. To calculate the interpolated face Cartesian velocity components the first step is to calculate the cylindrical components in the elements either side of the face using the equations

$$\begin{aligned} u_r &= u\cos\theta + v\sin\theta \\ u_\theta &= -u\sin\theta + v\cos\theta \\ u_z &= w \end{aligned} \tag{3.3.6}$$

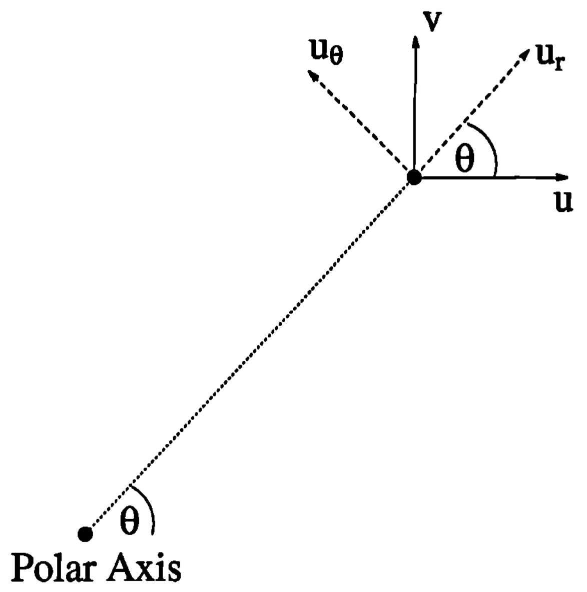

Figure 3.3.2: Conversion of Cartesian Velocity Components to Cylindrical

where θ is the cylindrical polar angle, see Figure 3.3.2. Interpolation using equation (3.3.5) then gives the values of the cylindrical components on the face. The face values of the Cartesian velocity components can then be calculated using

$$\begin{aligned} u &= u_r \cos\theta - u_\theta \sin\theta \\ v &= u_r \sin\theta + u_\theta \cos\theta \\ w &= u_z \end{aligned} \tag{3.3.7}$$

In the discretisation of the convection term it is necessary to express the face value of the conserved quantity as a function of the values of the quantity in the elements either side of the face. Considering just the u component of velocity equation (3.3.7) gives

$$u_f = (u_r)_f \cos\theta_f - (u_\theta)_f \sin\theta_f \tag{3.3.8}$$

In order to express the face values of the cylindrical components of velocity on the right of equation (3.3.8) in terms of the element Cartesian components, in elements P and A, it is first necessary to substitute the face cylindrical components using equation (3.3.5) which gives

$$\begin{aligned} u_f \; = \; & [\alpha_f (u_r)_P \; + \; (1-\alpha_f)(u_r)_A] \cos\theta_f \; - \\ & \qquad [\alpha_f (u_\theta)_P \; + \; (1-\alpha_f)(u_\theta)_A] \sin\theta_f \end{aligned} \tag{3.3.9}$$

If the cylindrical velocity components are now expressed in terms of the Cartesian equivalents the above equation becomes

$$\begin{aligned} u_f \; = \; & \cos\,\theta_f \,[\,\alpha_f \,\{u_P \cos\,\theta_P \; + \; v_P \sin\,\theta_P\} + \\ & \qquad (1-\alpha_f)\,\{u_A \cos\,\theta_A \; + \; v_A \sin\,\theta_A\}\,] \\ & -\sin\,\theta_f \,[\,\alpha_f \,\{-u_P \sin\,\theta_P \; + \; v_P \cos\,\theta_P\} + \\ & \qquad (1-\alpha_f)\,\{-u_A \sin\,\theta_A \; + \; v_A \cos\,\theta_A\,] \end{aligned} \tag{3.3.10}$$

Rearranging equation (3.3.10) by collecting terms multiplying the Cartesian components of velocity in the two elements gives

$$\begin{aligned} u_f \; = \; & \alpha_f \, u_P \,[\cos\,\theta_f \, \cos\,\theta_P + \sin\,\theta_f \, \sin\,\theta_P\,] + \\ & (1-\alpha_f)\, u_A \,[\cos\,\theta_f \, \cos\,\theta_A + \sin\,\theta_f \, \sin\,\theta_A\,] + \\ & \alpha_f \, v_P \,[\cos\,\theta_f \, \sin\,\theta_P - \sin\,\theta_f \, \cos\,\theta_P\,] + \\ & (1-\alpha_f)\, v_A \,[\cos\,\theta_f \, \sin\,\theta_A - \sin\,\theta_f \, \cos\,\theta_A\,] \end{aligned} \tag{3.3.11}$$

Using simple trigonometric equalities for the sin and cos of differences of angles allows equation (3.3.11) to be simplified to give

$$\begin{aligned} u_f \; = \; & \alpha_f \, u_P \, \cos(\theta_P - \theta_f) + (1-\alpha_f)\, u_A \, \cos(\theta_A - \theta_f) + \\ & \alpha_f \, v_P \, \sin(\theta_P - \theta_f) + (1-\alpha_f)\, v_A \, \sin(\theta_A - \theta_f) \end{aligned} \tag{3.3.12}$$

In order to simplify the above equation, and in the end aid the discretisation of the convection term, it is assumed that the face, f, lies midway between the two adjacent element centroids. This introduces an error into the calculation procedure but it has the advantage that only one angle has to be calculated. In addition, it aids in the formation of a contribution which maintains diagonal dominance of the resulting system matrix. These assumptions allow the angle θ_D, which is equal to half the cylindrical polar angle between elements A and P, to be used as a replacement for the two angle differences appearing in equation (3.3.12)

$$\theta_D = \frac{1}{2}(\theta_P - \theta_A) = \theta_P - \theta_f = \theta_f - \theta_A \qquad (3.3.13)$$

Using θ_D in equation (3.3.12) gives

$$u_f = \cos\theta_D\,[\alpha_f\,u_P + (1-\alpha_f)\,u_A] + \sin\theta_D\,[\alpha_f\,v_P - (1-\alpha_f)\,v_A] \qquad (3.3.14)$$

Comparison of equation (3.3.14) with the Cartesian form of this equation, (3.2.2), shows that the first of the above terms is the Cartesian terms multiplied by cos θ_D. In addition there is a second term which is a function of the v velocity in the elements either side of the face. In the discretised equation for Cartesian interpolation the contribution from the combined diffusion and convection term is

$$(u_P - u_A)\,[D_f\,A(|P_f|) + \max(-F_f, 0.0)] \qquad (3.3.15)$$

where $D_f = \mu_f A_f / d_f$, $F_f = \rho_f A_f (\underline{u}.\underline{n})_f$ and the Peclet number $P_f = F_f / D_f$. This changes in the cylindrical interpolation case to

$$\begin{aligned}(u_P - u_A)\,[D_F\,A(|P_f|) + \cos\theta_D\,\max(-F_f, 0.0)] + \\ (v_P - v_A)\,\sin\theta_D\,\max(-F_f, 0.0)\end{aligned} \qquad (3.3.16)$$

The v component of velocity requires a similar approach to obtain an equation for its face value in the form of equation (3.3.14). The equation for v takes the form

$$v_f = \cos\theta_D\,[\alpha_f\,v_P + (1-\alpha_f)\,v_A] + \sin\theta_D\,[\alpha_f\,u_P - (1-\alpha_f)\,u_A] \qquad (3.3.17)$$

which leads to a contribution to the system matrix from a single face of the form

$$\begin{aligned}(v_P - v_A)\,[D_f\,A(|P_f|) + \cos\theta_D\,\max(-F_f, 0.0)] + \\ (u_P - u_A)\,\sin\theta_D\,\max(-F_f, 0.0)\end{aligned} \qquad (3.3.18)$$

The face value of the w component of velocity is given by Cartesian type interpolation

$$w_f \;=\; \alpha_f\, w_P \;+\; (1-\alpha_f)\, w_A \tag{3.3.19}$$

and so the contribution to the w system matrix from a face is the same as in the Cartesian case

$$(w_P - w_A)\, [D_f\, A(|P_f|) + \max(-F_f, 0.0)] \tag{3.3.20}$$

§3.4 General Interpolation

The most general case is when there exists significant flow curvature but the position of the axis of this curvature is unknown prior to the simulation. In order to accurately estimate the Cartesian velocity components on a face both the position of the axis influencing the flow at the face and the axial direction are required. The ideal solution, to determining the location and direction of the axis, is a method which allows an element to calculate both values from the velocity in the element and in the neighbouring elements.

In an attempt to determine the position and direction of the axis through considering the velocity vectors in the element and its neighbours it was first assumed that the flow is purely rotational. This allows the radial component of velocity to be considered as negligible and so the velocity in element P is given by

$$\underline{v}_P \;=\; u_\theta\, \underline{\hat{\theta}}_P + u_z\, \underline{\hat{z}}_P \tag{3.4.1}$$

where $\underline{\hat{\theta}}_P$ and $\underline{\hat{z}}_P$ are unit vectors in the tangential and axial directions for element P. Obviously, if $\underline{r}_{ax}$ is the point on the flow axis on the same flow plane as the centroid of element P, $\underline{r}_P$, then

$$\underline{r}_{ax} - \underline{r}_P \;=\; \lambda\, \underline{\hat{r}}_P \tag{3.4.2}$$

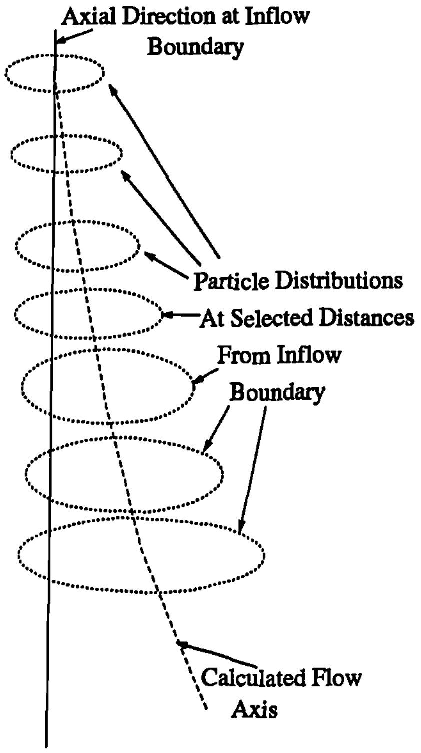
Axial Direction at Inflow Boundary
Particle Distributions At Selected Distances From Inflow Boundary
Calculated Flow Axis

Figure 3.4.1: "Shotgun" Approach to Determining Flow Axis

for some value of the constant λ. Each element has a pair of equations of the form (3.4.1) and (3.4.2). The problem is that to determine the flow direction for any element extra assumptions have to made usually relating to the directions of the unit cylindrical vectors in the neighbouring elements. When there are several axes influencing the flow field, for example several swirling jets, this can lead to serious errors in the estimations of the unit vectors and the position of the axis.

An alternative approach considered a short particle track emanating from the element centroid. As in the previous method it is assumed that the radial component of velocity is negligible and hence equation (3.4.1) is valid. For a short particle track it can be assumed that all points on the track lie on a circle with $\underline{r}_{ax}$ at its centre. Hence, taking any three points on the track allows a point on the flow axis to be calculated. Equation (3.4.2) can then be used to determine the the unit radial vector for element P. The problem now is how to determine the tangential and axial directions. All the information available has been used to estimate the radial direction and there is no information left which allows the velocity vector to be split into its components.

The currently used approach recognises that in many problems involving flow curvature there is usually at least one boundary condition which is the source of the flow curvature. For the problems of interest to this study this boundary condition is also a source of mass which enters the solution domain with a known axial and tangential component of velocity. The method uses a 'shotgun' approach in which a number of massless particles are started at equal distances and at the same radial distance from the inlet flow axis. The particles are tracked until they have travelled a specified distance in the axial direction, the initial axial direction being that at the mass inflow boundary. The positions of the particles are now averaged to give a new point on the flow axis. This step continues until at least one of the particles stops travelling in the chosen axial direction at which time the calculation of locations on the flow axis stops. This method is used to calculate the location of each of the 'jet' axes in turn. For each element, the closest of the points forming the jet axes is calculated. The flow in the element is assumed to be influenced by the jet on which this closest point lies. In order to calculate the radial, axial and tangential directions for the element the second closest point, on the same jet as the closest, to the element centroid is calculated. Let $\underline{r}_{ax}^{j}$ be the j'th point on the flow axis where $\underline{r}_{ax}^{1}$ is a point on the mass inflow boundary. Then if the two closest points are $\underline{r}_{ax}^{j}$ and $\underline{r}_{ax}^{(j+1)}$ a vector in the axial direction is given by

$$\underline{z}_P = \underline{r}_{ax}^{(j+1)} - \underline{r}_{ax}^{j} \tag{3.4.3}$$

The radial direction is calculated by first determining the closest point on the flow axis using

$$\underline{r}_c = \underline{r}_{ax}^{j} + \left[\left(\underline{r}_P - \underline{r}_{ax}^{j}\right) \cdot \hat{\underline{z}}_P\right] \hat{\underline{z}}_P \tag{3.4.4}$$

The radial direction for the element lies along the vector from this closest point to the position of the element centroid

$$\hat{\underline{r}}_P = \frac{\underline{r}_P - \underline{r}_c}{|\underline{r}_P - \underline{r}_c|} \tag{3.4.5}$$

The tangential vector for the element is calculated by taking the cross product of the axial and radial vectors

$$\underline{\hat{\theta}}_P = \underline{\hat{z}}_P \, \mathrm{x} \, \underline{\hat{r}}_P \tag{3.4.6}$$

Having calculated the direction vectors for each element it is now possible to calculate the Cartesian velocity components at the faces. If the elements either side of a face are influenced by different jets then Cartesian interpolation is used as it is to expected that the influences will cancel each other and lead to little flow curvature on the face. Otherwise the axial, tangential and radial components of velocity in the elements either side of the face are calculated using

$$\begin{aligned} u_r &= \underline{v} . \underline{\hat{r}} \\ u_\theta &= \underline{v} . \underline{\hat{\theta}} \\ u_z &= \underline{v} . \underline{\hat{z}} \end{aligned} \tag{3.4.7}$$

where $\underline{v}$ is a vector consisting of the Cartesian velocity components. Linear interpolation can then be used to calculate the r, θ, z components of velocity on the face. The direction vectors for the face are calculated using the equations

$$\begin{aligned} \underline{\hat{z}}_f &= \frac{\underline{\hat{z}}_A + \underline{\hat{z}}_P}{|\underline{\hat{z}}_A + \underline{\hat{z}}_P|} \\ \underline{R}_f &= \underline{\hat{r}}_A + \underline{\hat{r}}_P \\ \underline{\hat{r}}_f &= \frac{\underline{R}_f - (\underline{R}_f . \underline{\hat{z}}_f)\, \underline{\hat{z}}_f}{\left|\underline{R}_f - (\underline{R}_f . \underline{\hat{z}}_F)\, \underline{\hat{z}}_f\right|} \\ \underline{\hat{\theta}}_f &= \underline{\hat{z}}_f \, \mathrm{x} \, \underline{\hat{r}}_f \end{aligned} \tag{3.4.8}$$

The i'th Cartesian component of velocity is now calculated using the formula

$$(u_i)_f = (u_r)_f\, (\hat{r}_i)_f + (u_\theta)_f\, (\hat{\theta}_i)_f + (u_z)_f\, (\hat{z}_i)_f \tag{3.4.9}$$

where $(\hat{z}_i)_f$ signifies the i'th component of $(\underline{\hat{z}})_f$ when expressed as a Cartesian vector.

In the calculation of the convective contribution to the u_x momentum equation start from equation (3.4.9) replacing i with x. Substituting the velocity components on the right of this equation with the values in the elements either side of the face gives

$$\begin{aligned}(u_x)_f = {} & [\alpha_f\,(u_r)_P + (1-\alpha_f)(u_r)_A]\,(\hat{r}_x)_f \\ & + [\alpha_f\,(u_\theta)_P + (1-\alpha_f)(u_\theta)_A]\,(\hat{\theta}_x)_f \\ & + [\alpha_f\,(u_z)_P + (1-\alpha_f)(u_z)_A]\,(\hat{z}_x)_f\end{aligned} \tag{3.4.10}$$

Now expressing the velocities in terms of the Cartesian equivalents the equation becomes

$$\begin{aligned}(u_x)_f = {} & (\hat{r}_x)_f\,\alpha_f\,\{[(\hat{r}_x)_P(u_x)_P + (\hat{r}_y)_P(u_y)_P + (\hat{r}_z)_P(u_z)_P] \\ & + (1-\alpha_f)[(\hat{r}_x)_A(u_x)_A + (\hat{r}_y)_A(u_y)_A + (\hat{r}_z)_A(u_z)_A]\} \\ & + (\hat{\theta}_x)_f\,\alpha_f\,\{[(\hat{\theta}_x)_P(u_x)_P + (\hat{\theta}_y)_P(u_y)_P + (\hat{\theta}_z)_P(u_z)_P] \\ & + (1-\alpha_f)[(\hat{\theta}_x)_A(u_x)_A + (\hat{\theta}_y)_A(u_y)_A + (\hat{\theta}_z)_A(u_z)_A]\} \\ & + (\hat{z}_x)_f\,\alpha_f\,\{[(\hat{z}_x)_P(u_x)_P + (\hat{z}_y)_P(u_y)_P + (\hat{z}_z)_P(u_z)_P] \\ & + (1-\alpha_f)[(\hat{z}_x)_A(u_x)_A + (\hat{z}_y)_A(u_y)_A + (\hat{z}_z)_A(u_z)_A]\}\end{aligned} \tag{3.4.11}$$

Rearranging this equation gives

$$\begin{aligned}(u_x)_f = {} & \alpha_f\,(u_x)_P\,[(\hat{r}_x)_f\,(\hat{r}_x)_P + (\hat{\theta}_x)_f\,(\hat{\theta}_x)_P + (\hat{z}_x)_f\,(\hat{z}_x)_P] + \\ & (1-\alpha_f)\,(u_x)_A\,[(\hat{r}_x)_f\,(\hat{r}_x)_A + (\hat{\theta}_x)_f\,(\hat{\theta}_x)_A + (\hat{z}_x)_f\,(\hat{z}_x)_A] + \\ & \alpha_f\,(u_y)_P\,[(\hat{r}_x)_f\,(\hat{r}_y)_P + (\hat{\theta}_x)_f\,(\hat{\theta}_y)_P + (\hat{z}_x)_f\,(\hat{z}_y)_P] + \\ & (1-\alpha_f)\,(u_y)_A\,[(\hat{r}_x)_f\,(\hat{r}_y)_A + (\hat{\theta}_x)_f\,(\hat{\theta}_y)_A + (\hat{z}_x)_f\,(\hat{z}_y)_A] + \\ & \alpha_f\,(u_z)_P\,[(\hat{r}_x)_f\,(\hat{r}_z)_P + (\hat{\theta}_x)_f\,(\hat{\theta}_z)_P + (\hat{z}_x)_f\,(\hat{z}_z)_P] + \\ & (1-\alpha_f)\,(u_z)_A\,[(\hat{r}_x)_f\,(\hat{r}_z)_A + (\hat{\theta}_x)_f\,(\hat{\theta}_z)_A + (\hat{z}_x)_f\,(\hat{z}_z)_A]\end{aligned} \tag{3.4.12}$$

The term

$$(\hat{r}_x)_f\,(\hat{r}_x)_P + (\hat{\theta}_x)_f\,(\hat{\theta}_x)_P + (\hat{z}_x)_f\,(\hat{z}_x)_P \tag{3.4.13}$$

is the dot product of two unit vectors and its value will be the cosine of the angle between the vectors. Thus equation (3.4.13) can be assumed to be equal to $\cos(\theta_P - \theta_f)$ and as in the cylindrical interpolation scheme the assumptions of equation (3.3.13) are used to simplify equation (3.4.12) with

$$\cos(2\theta_{D_x}) = (\hat{r}_x)_A(\hat{r}_x)_P + (\hat{\theta}_x)_A(\hat{\theta}_x)_P + (\hat{z}_x)_A(\hat{z}_x)_P \tag{3.4.14}$$

Simple trigonometric identities give the value of $\cos(\theta_{D_x})$. The remaining terms in equation (3.4.12) of the form

$$(\hat{r}_x)_f(\hat{r}_y)_P + (\hat{\theta}_x)_f(\hat{\theta}_y)_P + (\hat{z}_x)_f(\hat{z}_y)_P \tag{3.4.15}$$

produce a value similar to $\sin(\theta_P - \theta_f)$. Again using the approximations of equation (3.3.13)

$$\sin(2\theta_{D_{xy}}) = (\hat{r}_x)_A(\hat{r}_y)_P + (\hat{\theta}_x)_A(\hat{\theta}_y)_P + (\hat{z}_x)_A(\hat{z}_y)_P \tag{3.4.16}$$

and again trigonometric identities can be used to give the value of $\sin(\theta_{D_{xy}})$. The contribution to the u-momentum equation from this face now takes the form

$$\begin{aligned}(u_P - u_A)\,[D_f\,A(|P_f|) + \cos(\theta_{D_x})\,\max(-F_f, 0.0)] &+ \\ H_u\,\max(-F_f, 0.0)&\end{aligned} \tag{3.4.17}$$

where H_u represents the 4 curvature terms from equation (3.4.12). This equation differs from the cylindrical equivalent equation, (3.3.16), only in the way $\cos(\theta_{D_x})$ is calculated and the formulation of the curvature terms. The v and w momentum equations also lead to equations of the same form as equation (3.4.17)

This method allows interpolation of the curvature components of velocity for a case when the flow axis can not be aligned with the line x=0, y=0. It also allows the solution of problems involving several flow axes spread throughout the solution domain. The main drawback of the method is that it requires the curvature to be a result of mass inflow boundary conditions. It is desirable to construct a general interpolation scheme based on quantities local to each element rather than the global approach used in the current method. Having formulated a method which calculates the unit vectors in the radial, tangential and axial directions for each element the methods presented in this section, from equation (3.4.7) onwards, will still be applicable.

§3.5 Higher Order Differencing Schemes

In section 2.2 the reasons for the use of differencing schemes were discussed. In addition, a number of first order schemes were introduced, which in many cases satisfy the requirements of a differencing scheme. Difficulties arise when these first order schemes are required to handle problems involving very sharp gradients in a solved quantity. In this circumstance the schemes over estimate the diffusive influence and cause an under estimation of any gradients.

The use of higher order differencing schemes can significantly improve the resolution of sharp gradients(61). For first order schemes the contribution to the discretised equation from a face is only a function of the solved variable's values in the elements either side of the face. For higher order schemes a face's contribution is a function of a larger number of elements. All the schemes discussed in this section require three element values, those in the upwind, downwind and "upwind upwind" elements.

The ultimate reason for the investigation into higher order schemes concerned their application to swirling flows. Shore *et al*(58) ran a series of tests involving the simulation of swirling flow in a pipe using a BFC block structured mesh. Their work indicated problems associated with the use of the hybrid scheme for such a case.

§3.5.1 Choosing the "Upwind Upwind" Element

When using a three point differencing scheme the upwind, downwind and "upwind upwind" elements must be chosen for each face. The upwind element, element u in Figure 3.5.1, is the element on the side of the face where $\underline{u} . \underline{n}$ is non-negative, $\underline{n}$ being the unit outward normal to the face of the element. The downwind element,

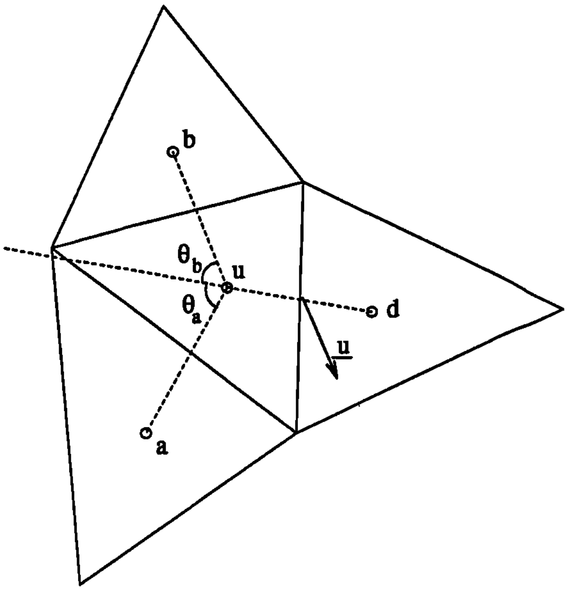

Figure 3.5.1: Determination of "Upwind Upwind" Element

element d in Figure 3.5.1, is the element on the other side of the face to element u. For a Cartesian mesh the third element is chosen by taking a mesh line containing both elements either side of the face. The third element is the element on this line adjacent to the upwind element which is not the downwind element[62]. In general for an unstructured mesh there is no entity equivalent to a mesh line. In this case the third element is chosen by taking the adjacent element to the upwind element, in Figure 3.5.1 this could be either element a or b, which gives the maximum value of

$$\frac{(\underline{r}_i - \underline{r}_u)}{|(\underline{r}_i - \underline{r}_u)|} \cdot (\underline{r}_u - \underline{r}_d) = \theta_i \tag{3.5.1}$$

where $\underline{r}_i$ is the position of the centroid of element i. It can be seen that the calculation of the "upwind upwind" element, using equation (3.5.1), does not use the face velocity apart from in the evaluation of the upwind element. On a Cartesian mesh this method for selecting the third point would choose the same element as the mesh line technique.

§3.5.2 QUICK

The QUICK scheme[61] uses the formulation

$$\phi_f = 0.75\,\phi_u + 0.375\,\phi_d - 0.125\,\phi_{uu} \tag{3.5.2}$$

to calculate the face value of a variable ϕ. When using this expression in the convective term, steps have to taken to guarantee diagonal dominance of the system matrix. One possible linearisation, used by Flow3d[56] Version 2.3.2, gives a contribution to the diagonal of the system matrix of the form

$$D_f + 0.625 \max(-F_f,\, 0.0) \tag{3.5.3}$$

The negative of this value is placed in the off diagonal position corresponding to the adjacent element. There is also a source term of the form

$$\begin{aligned} -0.125 \max(F_f,\, 0.0)\,(\phi_P - \phi_{uu}) - 0.125 \max(-F_f,\, 0.0)\,(\phi_{uu} - \phi_A) \\ - 0.375 \max(F_f,\, 0.0)\,(\phi_A - \phi_P) \end{aligned} \tag{3.5.4}$$

§3.5.3 QUICKER

The QUICKER algorithm[63] is an extended and revised form of the QUICK algorithm. The algorithm accounts for the need to obtain positive contributions to the system matrix and diagonal dominance in its formulation rather than leaving this to the implementation. The contribution to the system matrix is

$$(D_f + 0.75\,|F_f|)\,(\phi_P - \phi_A) \tag{3.5.5}$$

and to the source vector

$$0.125\,F_f\,\phi_{uu} - 1.125 \max(F_f,\, 0.0)\,\phi_A + 1.125 \max(-F_f,\, 0.0)\,\phi_P \tag{3.5.6}$$

§3.5.4 SMART

The QUICK and QUICKER schemes do not guarantee that the estimated face value of a quantity is bounded by the local element values, termed interpolative boundedness. The SMART scheme[64] was formulated to satisfy the Convective Boundedness Criterion (CBC) which states that for a scheme to possess computed boundedness it must have both interpolative boundedness and convective stability (this is basically equivalent to upwind bias). The scheme adds

$$[D_f + (0.625 + \alpha)\ \max(-F_f,\ 0.0)]\,(\phi_P - \phi_A) \tag{3.5.7}$$

into the system matrix and there is a contribution to the source vector of the form

$$\begin{aligned} &-(0.125 + \alpha)\ \max(F_f,\ 0.0)\,(\phi_P - \phi_{uu}) \\ &\quad -(0.125 + \alpha)\ \max(-F_f,\ 0.0)\,(\phi_{uu} - \phi_A) \\ &\quad -(0.375 - \alpha)\ \max(F_f,\ 0.0)\,(\phi_A - \phi_P) \end{aligned} \tag{3.5.8}$$

The scheme uses

$$\phi_P^* = \frac{\phi_u - \phi_{uu}}{\phi_d - \phi_{uu}} \tag{3.5.9}$$

to determine which of a set of equations to use to calculate the value of α

$$\begin{aligned} \alpha &= (0.25\,\phi_P^* - 0.375)/(2.0\,\phi_P^* - 1.0) & 0.0 \geq \phi_P^* &\geq 1.0 \\ \alpha &= (2.25\,\phi_P^* - 0.375)/(2.0\,\phi_P^* - 1.0) & 0.0 < \phi_P^* &< (1/6) \\ \alpha &= 0.0 & (1/6) < \phi_P^* &\leq (5/6) \\ \alpha &= (0.625 - 0.75\,\phi_P^*)/(2.0\,\phi_P^* - 1.0) & (5/6) < \phi_P^* &< 1.0 \end{aligned} \tag{3.5.10}$$

§3.5.5 OSPRE

The final scheme tested was the OSPRE scheme[(54)]. The scheme uses the values

$$\begin{aligned} d_1 &= \phi_d - \phi_u \\ d_2 &= \phi_u - \phi_{uu} \\ d_3 &= d_1^2 + d_1\, d_2 + d_2^2 \\ d_4 &= d_1^2 + 0.25\,(d_1\, d_2 + d_2^2) \end{aligned} \tag{3.5.11}$$

to calculate the contributions to the discretised equations. The value

$$D_f + \min(1.0,\ d_3/d_4)\,|F_f| + \max(0.0,\ d_3/d_4 - 1.0)|F_f| \tag{3.5.12}$$

is added onto the diagonal of the system matrix and

$$-D_f - \min(1.0,\ d_3/d_4)\,|F_f| \tag{3.5.13}$$

is put in the position associated with the adjacent element. The value added into the source vector is

$$\begin{aligned} -\max(F_f,\ 0.0)\,\phi_A + \max(-F_f,\ 0.0)\,\phi_P& \\ + \max(0.0,\ d_3/d_4 - 1.0)|F_f|\,\phi_A& \end{aligned} \tag{3.5.14}$$

§3.5.6 Test Cases

To test the merits of using higher order differencing schemes the moving lid cavity was used as preliminary test case. The solution domain consisted of a unit cube spatially discretised using a Cartesian mesh employing 33 elements in the x and y directions and a single element in the third direction. Using this mesh the numerical results from the simulation of this test case, see Chapter 5, in CWNN shows limited deviation from the benchmark solutions when the Reynolds number is either 100 or 400. In this set of tests the Reynolds number is set to 1,000 to make the differences between

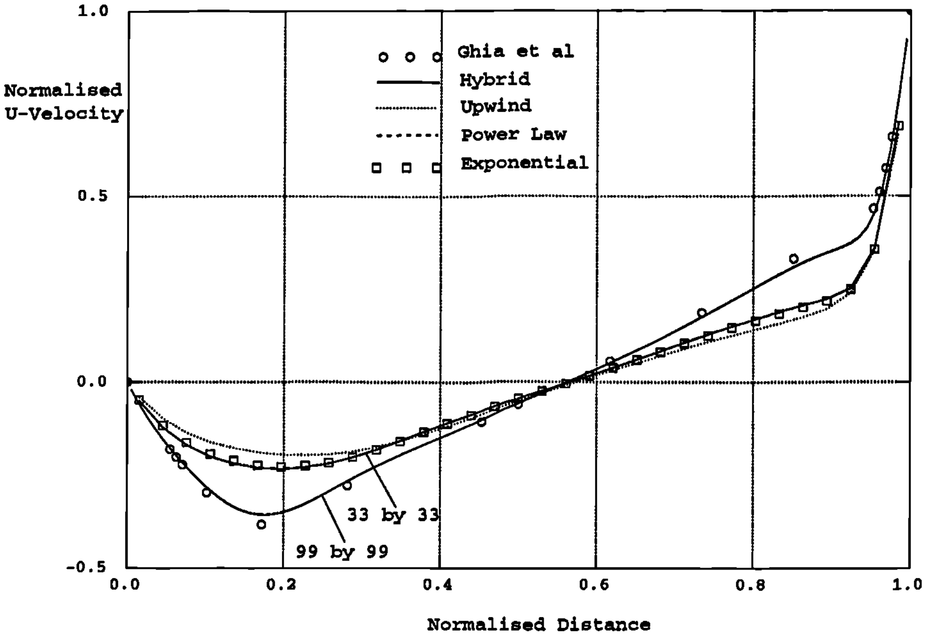

Figure 3.5.2: Comparison of First Order Schemes, U-Velocity Moving Lid, Re=1000

the schemes and the benchmark solution clearer. Figures 3.5.2 and 3.5.3 show comparisons of the first order schemes, hybrid, upwind, power law and exponential, against the benchmark solutions of Ghia[65]. On the 33 by 33 mesh none of the results are particularly close to the Ghia results. Of the first order schemes the upwind scheme performs the worst with very little difference between the three other schemes. This confirms that the hybrid scheme is generally the best option because its accuracy compares well with the accuracy of the computationally more expensive schemes. The result from a simulation using a 99 by 99 mesh and the hybrid scheme is also shown in Figures 3.5.2 and 3.5.3. It can be seen that refining the mesh does produce answers which compare well with those of Ghia. The problem is that with 9 times as many elements in the solution domain the simulations take at least 9 times as long. Figures 3.5.4 and 3.5.5 show the results, obtained using the 33 by 33 mesh, for the higher order schemes discussed in this section. The hybrid scheme is also included on the plots to allow comparisons with the first order schemes. All the higher order schemes produce considerably better solutions, compared to the hybrid scheme, but this accuracy is limited by the fairly coarse mesh being used. Of the higher order schemes there seems little difference between the QUICK and QUICKER schemes

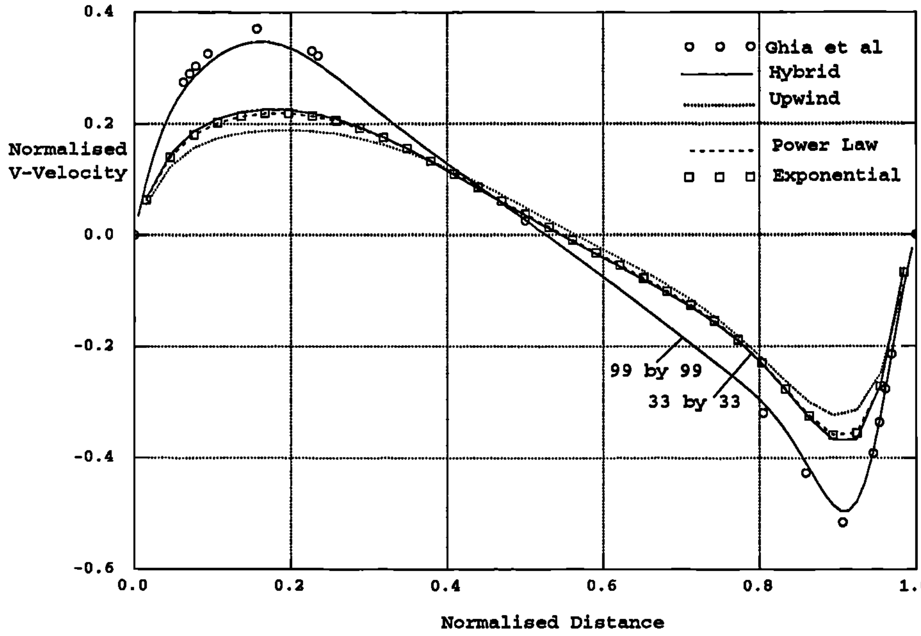

Figure 3.5.3: Comparison of First Order Schemes, V-Velocity Moving Lid, Re=1000

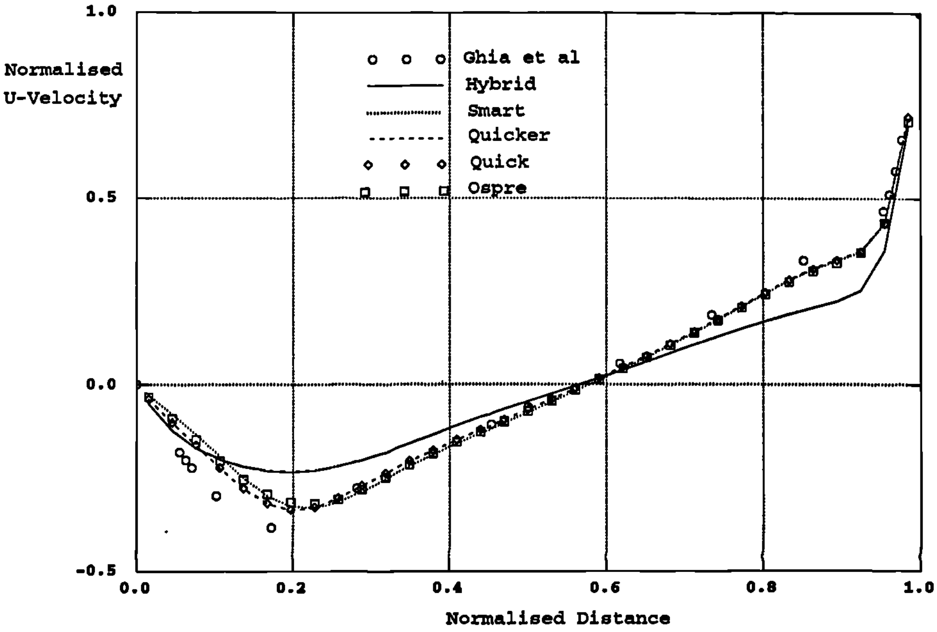

Figure 3.5.4: Comparison of Higher Order Schemes, U-Velocity Moving Lid, Re=1000

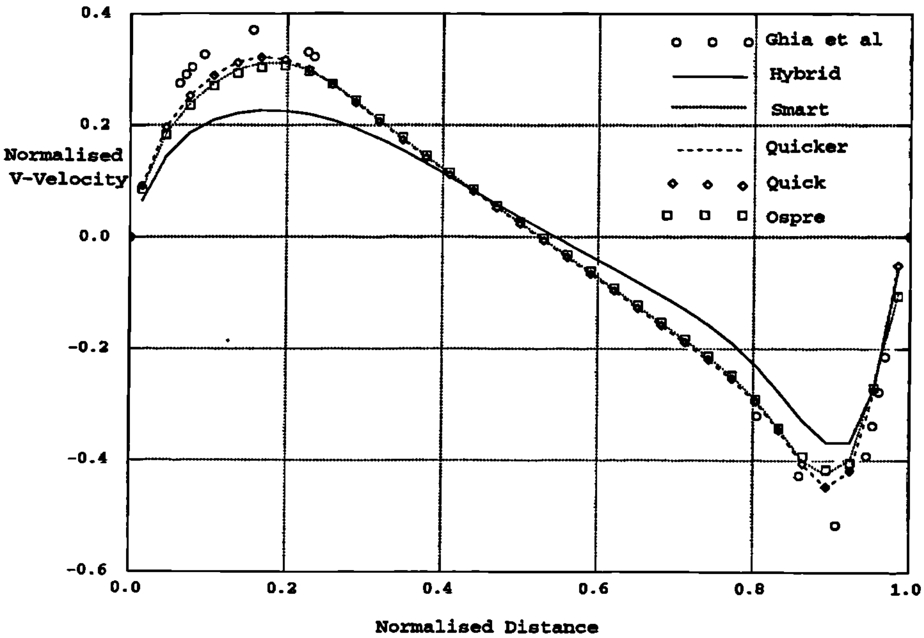

Figure 3.5.5: Comparison of Higher Order Schemes, V-Velocity Moving Lid, Re=1000

which perform slightly better than the other two schemes. Similarly there is little difference between the SMART and OSPRE schemes both of which produce poorer answers, compared to QUICK, around the maxima and minima of the graphs.

The comparative simulation times for the hybrid and QUICK schemes were 12 seconds and 15 seconds. The extra simulation time for QUICK is caused by the need to calculate the third element in the differencing scheme and the extra number of operations required to set up the system matrices. To get results using the hybrid scheme which match those of the higher order schemes requires a mesh which has around 60 by 60 elements. On such a mesh the simulation time is 40 seconds hence for similar accuracy the higher schemes require a much shorter run time for this problem.

The second case investigated here concerns the simulation of a forced vortex. The mesh used for the simulation is shown in Figure 3.5.6. Figure 3.5.7 shows graphs of the resultant velocity on radial lines when using the hybrid, upwind, QUICK and SMART schemes. The plots show that the first order schemes fail to predict the

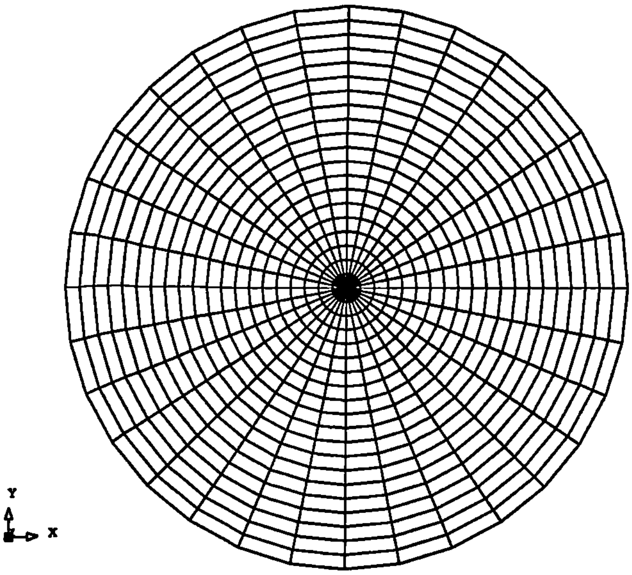

Figure 3.5.6: Mesh Used For Forced Vortex Test Case

expected straight lines whilst the higher order schemes very nearly match the analytic answers. One of the main reasons why the higher order schemes perform better on this case is that they do not reduce the diffusion contribution from a face. In the hybrid scheme the contribution from diffusion is reduced by a factor dependent on the face Peclet number. In pure forced vortex flow the cylindrical polar conservation equations being solved(59) are

$$\begin{aligned}
&r-\textit{component} &\quad \rho\frac{v_\theta^2}{r} &= \frac{\partial p}{\partial r} \\
&\theta-\textit{component} &\quad 0 &= \mu\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}(r\,v_\theta)\right) \\
&z-\textit{component} &\quad 0 &= -\frac{\partial p}{\partial z}-\rho\,g
\end{aligned} \qquad (3.5.15)$$

the analytic solution to which, assuming v_θ is bounded at $r = 0.0$ and has value 5 at $r = 1.0$, is given by

$$v_\theta = 5\,r \qquad (3.5.16)$$

From the above equations it can be seen that the diffusion term should be identically

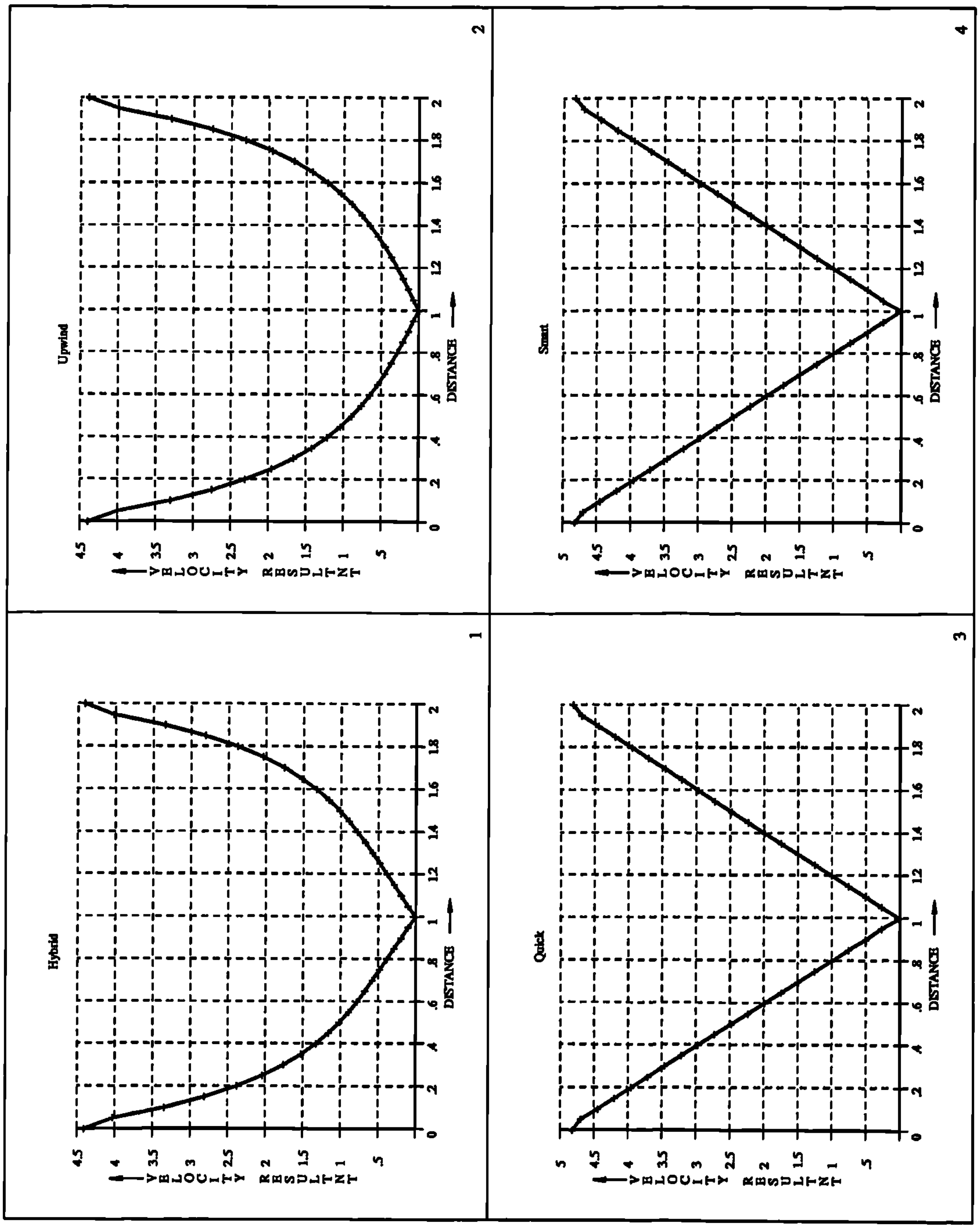

Figure 3.5.7: Forced Vortex Results

zero everywhere. Consequently, the sum of the contributions to the diffusion term, over all faces of an element, should be zero. This is the case if no corrective adjustments are made to the calculated diffusion contribution from a face, this being true for QUICK and SMART. In the hybrid scheme the diffusion contribution from a face is reduce by a function of the face Peclet number. Since each face can have a different value for the Peclet number this will lead to a non zero contribution from the sum of diffusion contributions. The finite contribution from the diffusion term introduces errors into the answers. In the paper by Shore[58] the higher order scheme which produced answers comparable to those of the hybrid scheme, the CONDIF scheme, also reduces the diffusion contribution from a face as a function of the Peclet number.

§3.6 Solvers

The discretisation of a conservation equation produces a set of linear equations which can be written as a matrix of the form

$$\underline{\underline{A}}\,\underline{\phi} = \underline{b} \tag{3.6.1}$$

A direct solution method could be used to obtain the exact solution of this set of equations. One disadvantage of such a method is the need to store a n by n matrix where n is the number of locations at which the solution is being sought - it is not unusual for n to be in the order of 10^5 so the memory requirements are of the order of $4*10^{10}$ bytes. Even by current standards of computer hardware this is an extremely large amount of memory. Secondly the time required to produce a solution using a direct solution method is large. The direct method may be justified if a very accurate solution of a small number of equations was required. In CFD this is not generally true because approximations have usually been made in the calculation of $\underline{\underline{A}}$ and $\underline{b}$. Commonly both $\underline{\underline{A}}$ and $\underline{b}$ depend, either directly or indirectly, in all of the solved variables. Hence a number of equations of the form (3.6.1), where the matrix $\underline{\underline{A}}$ is different on each occasion, have to be solved to obtain the solution(s) to the simulation. The aim, at each stage, is to improve the current estimate of the solved variables consequently moving $\underline{\underline{A}}$ and $\underline{b}$ towards their forms which give the converged simulation result. Given these constraints a much better methodology is the use of iterative linear matrix solvers. In general these methods use the current approximation to the solution $\underline{\phi}^i$ to obtain an improved answer $\underline{\phi}^{i+1}$ where

$$\underline{\phi}^{i+1} = \mathrm{f}(\underline{\phi}^i, \underline{\underline{A}}, \underline{b}) \tag{3.6.2}$$

for some function f such that the technique satisfies

$$|\underline{\phi}_i - \underline{\phi}^*| \rightarrow 0 \qquad \text{as } i \rightarrow \infty \tag{3.6.3}$$

where $\underline{\phi}^*$ is the exact solution. One of the main advantages of the iterative approach is that at each stage of the solution procedure the amount of effort spent in solving the matrix equations can be controlled to aid the whole solution procedure. In the next few sections a number of these techniques are briefly outlined. The algorithms for each of the solvers, and the ILU preconditioner, are presented in the sections of Appendix B.

§3.6.1 Jacobi Over Relaxation

In the Jacobi iterative method the i+1 iteration value of ϕ in the j'th element, represented by the symbol ϕ_j^{i+1}, is obtained using the equation

$$\phi_j^{i+1} = \frac{1}{A_{jj}} \left(b_j - \sum_{k \neq j} A_{jk} \phi_k^i \right) \tag{3.6.4}$$

This method is converted into the Jacobi Over Relaxation, JOR, method by use of a relaxation factor α. In the JOR method the i+1 iteration value of ϕ_j consists of α times the value produced by the Jacobi iterative value plus (1-α) times the previous iteration value of ϕ_j.

$$\phi_j^{i+1} = (1-\alpha)\phi_j^i + \frac{\alpha}{A_{jj}} \left(b_j - \sum_{k \neq j} A_{jk} \phi_k^i \right) \tag{3.6.5}$$

The method uses the terminology "Over Relaxation" because usually the value of α is chosen to be ≥ 1 and is used to advance the solution more quickly than would be the case in the standard Jacobi iterative method.

§3.6.2 Successive Over Relaxation

The basis for the Successive Over Relaxation, SOR, method is the Gauss- Seidel iterative scheme. The equation for the Gauss-Seidel method

$$\phi_j^{i+1} = \frac{1}{A_{jj}}\left(b_j - \sum_{k\neq j} A_{jk}\phi_k^m\right) \tag{3.6.6}$$

is the same as in the Jacobi method except the last symbol on the right hand of the equation has the superscript m rather than i. In the Gauss-Seidel method this superscript takes the following values

$$\begin{aligned} m &= i && \text{if } k > j \\ m &= i+1 && \text{if } k < j \end{aligned}$$

This algorithm is extended into the over relaxation form in the same manner as for the JOR method.

$$\phi_j^{i+1} = (1-\alpha)\phi_j^i + \frac{\alpha}{A_{jj}}\left(b_j - \sum_{k\neq j} A_{jk}\phi_k^m\right) \tag{3.6.7}$$

§3.6.3 Residual Jacobi and Successive Over Relaxation

One of the major problems in both the SOR and JOR methods is that numerical errors can affect both what is recognised as a converged solution and the number of iterations necessary to obtain the solution. One method to reduce the effect of such inaccuracies is to use a residual method. In such a method the equation solved at each iteration is

$$\underline{\underline{A}}\,\underline{y}^{i+1} = \underline{r}^{i} \tag{3.6.8}$$

where

$$\underline{r}^{i} = \underline{b} - \underline{\underline{A}}\,\underline{\phi}^{i} \tag{3.6.9}$$

At each stage of the procedure one or more iterations of (3.6.8) are carried out, using the selected method, to obtain $\underline{y}^{i+1}$ The solution $\underline{\phi}^{i}$ and residual $\underline{r}^{i}$ are then updated using the equations

$$\begin{aligned} \underline{\phi}^{i+1} &= \underline{\phi}^{i} + \underline{y}^{i+1} \\ \underline{r}^{i+1} &= \underline{r}^{i} - \underline{\underline{A}}\,\underline{y}^{i+1} \end{aligned} \tag{3.6.10}$$

§3.6.4 Conjugate Gradient Method

The Conjugate Gradient Method[66–68] can be used to obtain the solution of a symmetric, positive-definite set of linear equations. Let $f(\underline{\phi})$ be the quadratic form

$$f(\underline{\phi}) = \frac{1}{2}\underline{\phi}^{T}\,\underline{\underline{A}}\,\underline{\phi} - \underline{b}^{T}\,\underline{\phi} + c \tag{3.6.11}$$

It can be shown that the gradient of f is equal to

$$f'(\underline{\phi}) = \frac{1}{2}\underline{\underline{A}}^{T}\underline{\phi} + \frac{1}{2}\underline{\underline{A}}\,\underline{\phi} - \underline{b} \tag{3.6.12}$$

which if $\underline{\underline{A}}$ is symmetric reduces to

$$f'(\underline{\phi}) = \underline{\underline{A}}\,\underline{\phi} - \underline{b} \tag{3.6.13}$$

Hence the solution of (3.6.1) corresponds to a critical point of the quadratic form f. It can also be shown that if $\underline{\underline{A}}$ is symmetric

$$f(\underline{x}) = f(\underline{\phi}) + \frac{1}{2}(\underline{x} - \underline{\phi})^T \underline{\underline{A}}(\underline{x} - \underline{\phi}) \qquad (3.6.14)$$

If $\underline{\underline{A}}$ is also positive-definite the term on the right of (3.6.14) is positive if $\underline{x} \neq \underline{\phi}$. Consequently, the solution of (3.6.1) is a global minimum of f.

To find the minimum of equation (3.6.1) the method of steepest descent could be used. In this method the search for the solution is begun by taking an initial guess, $\underline{\phi}^0$. The method updates the current estimate to the solution by moving in the direction in which f decreases most rapidly. This is in the opposite direction to the gradient of f, which by equation (3.6.13) is given by $\underline{b} - \underline{\underline{A}}\,\underline{\phi}$. If the residual, $\underline{r}^i$, is defined as

$$\underline{r}^i = \underline{b} - \underline{\underline{A}}\,\underline{\phi}^i \qquad (3.6.15)$$

then the update to the solution is in the direction of the residual. The updated solution is given by

$$\underline{\phi}^{i+1} = \underline{\phi}^i + \alpha^i \underline{r}^i \qquad (3.6.16)$$

with the only remaining unknown being the value of α^i. The approach taken by the method of steepest descent is to chose α^i so that it minimises $f(\underline{\phi}^{i+1})$. This is achieved when the gradient, with respect to α^i of $f(\underline{\phi}^{i+1})$ is equal to zero. Thus

$$\frac{d}{d\alpha^i} f(\underline{\phi}^{i+1}) = f'(\underline{\phi}^{i+1})^T \frac{d}{d\alpha^i}\underline{\phi}^{i+1} = f'(\underline{\phi}^{i+1})^T \underline{r}^i \qquad (3.6.17)$$

Thus as $-f'(\underline{\phi}^{i+1})$ is equal to $\underline{r}^{i+1}$ the value of α^i should be chosen so that successive residuals are orthogonal. The updated residual is not known until the value of α is known but the following set of equations

$$\begin{aligned}(\underline{r}^{i+1})^T\underline{r}^i &= 0\\ (\underline{b}-\underline{\underline{A}}\,\underline{\phi}^{i+1})^T\underline{r}^i &= 0\\ (\underline{b}-\underline{\underline{A}}(\underline{\phi}^i+\alpha^i\underline{r}^i))^T\underline{r}^i &= 0\\ (\underline{b}-\underline{\underline{A}}\,\underline{\phi}^i)^T\underline{r}^i-\alpha^i(\underline{\underline{A}}\,\underline{r}^i)^T\underline{r}^i &= 0\\ (\underline{r}^i)^T\underline{r}^i-\alpha^i(\underline{\underline{A}}\,\underline{r}^i)^T\underline{r}^i &= 0\\ \alpha^i &= \frac{(\underline{r}^i)^T\underline{r}^i}{(\underline{r}^i)^T\underline{\underline{A}}\,\underline{r}^i}\end{aligned} \tag{3.6.18}$$

show how the value of α^i can be determined from the current residual.

In the steepest descent method the search vectors are equal to the residuals. Consequently, the method suffers from the possibility that several iterations can use search vectors that are in the same direction. Obviously it is preferable that any search direction is only visited once and the correct step size, in the direction, is calculated at the first attempt. A possible answer to this requirement is to select a set of orthogonal search directions, $\underline{d}^i$, and take only one step in each direction. If

$$\underline{\phi}^{i+1} = \underline{\phi}^i+\alpha^i\underline{d}^i \tag{3.6.19}$$

then the value of α^i can be determined because the error vector, $\underline{e}^{i+1} = \underline{\phi}-\underline{\phi}^{i+1}$, must be orthogonal to $\underline{d}^i$. This gives

$$\begin{aligned}(\underline{d}^i)^T\underline{e}^{i+1} &= 0\\ (\underline{d}^i)^T(\underline{e}^i+\alpha^i\underline{d}^i) &= 0\\ \alpha^i &= -\frac{(\underline{d}^i)^T\underline{e}^i}{(\underline{d}^i)^T\underline{d}^i}\end{aligned} \tag{3.6.20}$$

The problem with this method is that to calculate α^i it is necessary to know the current error. To calculate the error it is necessary to know the solution. Hence there is a "Catch 22" situation in which it is necessary to know the solution before one can calculate the solution.

A better approach is to make the search directions A-orthogonal. The vectors $\underline{d}^j$ and $\underline{d}^i$ are A-orthogonal if

$$(\underline{d}^j)^T \underline{\underline{A}}\,\underline{d}^i = 0 \qquad (3.6.21)$$

The use of A-orthogonal search directions means that the error vector, $\underline{e}^{i+1}$, has to be A-orthogonal to $\underline{d}^i$ to enforce the requirement of a single step in each direction. From equation (3.6.20) this leads to the value of α^i being given by

$$\begin{aligned} \alpha^i &= \frac{(\underline{d}^i)^T \underline{\underline{A}}\,\underline{e}^i}{(\underline{d}^i)^T \underline{\underline{A}}\,\underline{d}^i} \\ &= \frac{(\underline{d}^i)^T \underline{r}^i}{(\underline{d}^i)^T \underline{\underline{A}}\,\underline{d}^i} \end{aligned} \qquad (3.6.22)$$

The only requirement now is to create a set of A-orthogonal search directions. This can be achieved by using the conjugate Gram-Schmidt process. Select any set of linearly independent vectors, $\underline{u}^i$. The A-orthogonal set of vectors are constructed by setting $\underline{d}^0 = \underline{u}^0$ and

$$\underline{d}^i = \underline{u}^i + \sum_{k=0}^{i-1} \beta_{ik}\,\underline{d}^k \qquad (3.6.23)$$

To evaluate the value of β_{ij}, where $i > j$, the transpose of equation (3.6.23) is multiplied by $\underline{\underline{A}}\,\underline{d}^j$ and the A-orthogonality of the search vectors is used to give

$$\begin{aligned} (\underline{d}^i)^T \underline{\underline{A}}\,\underline{d}^j &= (\underline{u}^i)^T \underline{\underline{A}}\,\underline{d}^j + \sum_{k=0}^{i-1} \beta_{ik}\,(\underline{d}^k)^T \underline{\underline{A}}\,\underline{d}^j \\ 0 &= (\underline{u}^i)^T \underline{\underline{A}}\,\underline{d}^j + \beta_{ij}\,(\underline{d}^j)^T \underline{\underline{A}}\,\underline{d}^j \\ \beta_{ij} &= -\frac{(\underline{u}^i)^T \underline{\underline{A}}\,\underline{d}^j}{(\underline{d}^j)^T \underline{\underline{A}}\,\underline{d}^j} \end{aligned} \qquad (3.6.24)$$

The disadvantages of this method are that:

1. in order to calculate the current search direction all previous search directions must be stored.

2. the number of operations required to construct the full set of search directions is of the same order as the number of operations required to perform Gaussian elimination.

At this point consider the residual vectors. At any stage of the process

$$\underline{r}^i = -\sum_{k=i}^{n-1} \delta_k \underline{\underline{A}}\,\underline{d}^k \qquad (3.6.25)$$

From this equation it can be seen that

$$(\underline{d}^j)^T \underline{r}^i = 0 \qquad (3.6.26)$$

for $i > j$. Similarly taking the inner product of equation (3.6.23) with $\underline{r}^j$ gives

$$(\underline{u}^j)^T \underline{r}^i = 0 \qquad (3.6.27)$$

for $i > j$. If $j = i$ this inner product gives

$$(\underline{d}^i)^T \underline{r}^i = (\underline{u}^i)^T \underline{r}^i \qquad (3.6.28)$$

The Conjugate Gradient Method creates the search directions by conjugation of the residuals, $\underline{u}^j = \underline{r}^j$. From equation (3.6.27) this leads to

$$(\underline{r}^j)^T \underline{r}^i = 0 \qquad i \neq j \qquad (3.6.29)$$

It can be shown that

$$\underline{r}^{j+1} = \underline{r}^j - \alpha^j \underline{\underline{A}}\,\underline{d}^j \qquad (3.6.30)$$

Taking the inner product of this equation with $\underline{r}^i$ gives

$$
\begin{aligned}
(\underline{r}^i)^T\underline{r}^{j+1} &= (\underline{r}^i)^T\underline{r}^j - \alpha^j(\underline{r}^i)^T\underline{\underline{A}}\,\underline{d}^j \\
\alpha^j(\underline{r}^i)^T\underline{\underline{A}}\,\underline{d}^j &= (\underline{r}^i)^T\underline{r}^j - (\underline{r}^i)^T\underline{r}^{j+1} \\
\alpha^j(\underline{r}^i)^T\underline{\underline{A}}\,\underline{d}^j &= \begin{cases} \frac{1}{\alpha^i}(\underline{r}^i)^T\underline{r}^i & i = j \\ -\frac{1}{\alpha^{i-1}}(\underline{r}^i)^T\underline{r}^i & i = j+1 \\ 0 & \text{otherwise} \end{cases} \\
\beta_{ij} &= \begin{cases} \frac{1}{\alpha^{i-1}}\frac{(\underline{r}^i)^T\underline{r}^i}{(\underline{d}^{i-1})^T\underline{\underline{A}}\,\underline{d}^{i-1}} & i = j+1 \\ 0 & i > j+1 \end{cases}
\end{aligned}
\tag{3.6.31}
$$

As a consequence of the choice of the residuals as the linearly independent vectors used to construct the search directions, most of the β_{ij} have now become zero. To simplify the equation the convention $\beta_{i(i-1)} = \beta^i$ is used. Now using equation (3.6.30)

$$
\begin{aligned}
\beta^i &= \frac{(\underline{r}^i)^T\underline{r}^i}{(\underline{d}^{i-1})^T(\underline{r}^{i-1} - \underline{r}^i)} \\
&= \frac{(\underline{r}^i)^T\underline{r}^i}{(\underline{d}^{i-1})^T\underline{r}^{i-1}}
\end{aligned}
\tag{3.6.32}
$$

Applying equation (3.6.28) this becomes

$$
\beta^i = \frac{(\underline{r}^i)^T\underline{r}^i}{(\underline{r}^{i-1})^T\underline{r}^{i-1}} \tag{3.6.33}
$$

It is now possible to construct the complete algorithm for the conjugate gradient method which is given in Appendix B.5.

Certain of the matrices produced in the field of CFD satisfy or nearly satisfy the requirement of producing a symmetric, positive-definite matrix, for example heat transfer by conduction only, but once fluid flow is being solved, and the SIMPLE algorithm is being employed to couple the velocity and pressure fields, the matrix is no longer symmetric.

§3.6.5 Bi-Conjugate Gradient Method

The bi-conjugate gradient method (Bi-CG)(69, 70) extends the theories of the conjugate methods to the solution of non-singular, unsymmetric square matrices. It is possible to create a symmetric, positive definite matrix by pre-multiplication of equation (3.6.1) by the transpose of $\underline{\underline{A}}$ and then use the conjugate gradient method to solve

$$\left[(\underline{\underline{A}})^T\underline{\underline{A}}\right]\underline{\phi} = (\underline{\underline{A}})^T\underline{b} \tag{3.6.34}$$

The difficulty with this method is the need to multiply two matrices. This requires $O(n^3)$ numerical operations. In addition, if $\underline{\underline{A}}$ is highly sparse a significant amount of extra memory may be required to store the matrix $(\underline{\underline{A}})^T\underline{\underline{A}}$. Finally the condition number of $(\underline{\underline{A}})^T\underline{\underline{A}}$ is equal to the square of the condition number of $\underline{\underline{A}}$ which leads to high numbers of iterations and loss of accuracy in the iterative solver.

The BiCG method constructs four sequences of vectors the residuals, $\underline{r}^k$, the search directions, $\underline{u}^k$, and their shadow equivalent forms, $\underline{\tilde{r}}^k$ and $\underline{\tilde{u}}^k$. The series of vectors are constructed to maintain the biorthogonality of the residuals

$$(\underline{\tilde{r}}^j)^T\underline{r}^i = (\underline{\tilde{r}}^i)^T\underline{r}^j = 0 \qquad \text{for} \quad j < i \tag{3.6.35}$$

and the biconjugacy of the search directions

$$(\underline{\tilde{u}}^i)^T\underline{\underline{A}}\,\underline{u}^j = (\underline{u}^i)^T\underline{\underline{A}}^T\,\underline{\tilde{u}}^j = 0 \qquad \text{for} \quad j < i \tag{3.6.36}$$

In addition, the sequences of residuals and search directions are formed so that the shadow form of one is orthogonal to the ordinary form of the other

$$(\underline{\tilde{r}}^i)^T\underline{u}^j = (\underline{r}^i)^T\underline{\tilde{u}}^j = 0 \qquad \text{for} \quad j < i \tag{3.6.37}$$

An initial estimate to the solution of equation (3.6.1), $\underline{\phi}^1$, is made and the starting residuals and search directions are calculated from

$$\underline{u}^1 = \underline{\tilde{u}}^1 = \underline{r}^1 = \underline{\tilde{r}}^1 = \underline{b} - \underline{\underline{A}}\,\underline{\phi}^1 \tag{3.6.38}$$

From this point onwards the required properties of the sequences of vectors are maintained through the equations

$$\begin{aligned}
\alpha^k &= \frac{(\underline{\tilde{r}}^k)^T \underline{r}^k}{(\underline{\tilde{u}}^k)^T \underline{\underline{A}}\,\underline{u}^k} \\
\underline{\phi}^{k+1} &= \underline{\phi}^k + \alpha^k \underline{u}^k \\
\underline{r}^{k+1} &= \underline{r}^k - \alpha^k \underline{\underline{A}}\,\underline{u}^k \\
\underline{\tilde{r}}^{k+1} &= \underline{\tilde{r}}^k - \alpha^k \underline{\underline{A}}^T \underline{\tilde{u}}^k \\
\beta^k &= \frac{(\underline{\tilde{r}}^{k+1})^T \underline{r}^{k+1}}{(\underline{\tilde{r}}^k)^T \underline{r}^k} \\
\underline{u}^{k+1} &= \underline{u}^k + \beta^k \underline{u}^k \\
\underline{\tilde{u}}^{k+1} &= \underline{\tilde{u}}^k + \beta^k \underline{\tilde{u}}^k
\end{aligned} \tag{3.6.39}$$

The Bi-CG method seems to offer the ideal method for iteratively obtaining the solution of a linear set of equations but it has a number of disadvantages. The algorithm requires two matrix multiplications per iteration which is twice the number required by the standard Conjugate Gradient Method. The convergence behaviour of the algorithm is often quite irregular which in finite precision arithmetic may slow down the rate of convergence. In addition, the method can break down although this can be avoided by the use of 'look ahead strategies'(71, 72).

§3.6.6 Bi-Conjugate Gradient Stabilised(l)

The Conjugate Gradient Squared, CGS, method of Sonneveld(73) proposed that rather than calculating the shadow residuals the computational work could be expended in extra reductions in the Bi-CG residuals. In the Bi-CG method the residual at the i'th step is given by

$$\underline{r}^i = P_i(\underline{\underline{A}})\,\underline{r}^0 \tag{3.6.40}$$

for some polynomial P_i. Similarly the shadow residuals satisfy

$$\underline{\tilde{r}}^i = P_i(\underline{\underline{A}})\,\underline{\tilde{r}}^0 \tag{3.6.41}$$

The value of ρ_i in the Bi-CG algorithm is given by

$$\rho_i = (\underline{\tilde{r}}^i)^T \underline{r}^i = (P_i(\underline{\underline{A}})\,\underline{\tilde{r}}^0)^T P_i(\underline{\underline{A}})\,\underline{r}^0 = (\underline{\tilde{r}}^i)^T P_i^2(\underline{\underline{A}})\underline{r}^0 \tag{3.6.42}$$

The argument behind the CGS method is that if the contraction operator, P_i, reduces the residual, $\underline{r}^0$, to $\underline{r}^i$ then it may be advantageous to apply the operator twice. Although the computational effort required by the CGS method is of the same order as a Bi-CG, the algorithm can lead to convergence rates twice as fast as the Bi-CG. Difficulties arise because there is no reason why the reduction operator should reduce the residual $P_i(\underline{\underline{A}})\,\underline{r}^0$. In many cases the convergence is very irregular and problems can be encountered if the initial residual is small, i.e. the starting guess is close to the solution.

The Bi-Conjugate Gradient Stabilised Method(74) attempted to overcome the irregular convergence of the CGS method by using a series of first degree Minimum Residual polynomial updates to the residual of the form

$$\underline{r}^k = (1 - \omega_k \underline{\underline{A}})\,\underline{r}^{k-1} \tag{3.6.43}$$

where the value of ω_k is chosen to produce the minimum residual. In many cases the Bi-CGStab method converges very smoothly and often performs better than the Bi-CG or CGS with very little extra computational effort. The algorithm fails when ω_k becomes close to zero which may occur if $\underline{\underline{A}}$ has imaginary roots in which the imaginary part is large compared to the real part. The BiCGStab2 algorithm(75) uses a second degree Minimum Residual polynomial. The method is an improvement on the Bi-CGStab method but stagnation and breakdown may occur at stages where the intermediate first degree Minimum Residual polynomials are being calculated.

These methods can be extended to produce the Bi-CGStab(l)(76) in which a l'th degree Minimum Residual polynomial is used to update the residual. At each stage of the algorithm l steps of the Bi-CG algorithm are performed to give l residual vectors. Gram-Schmidt orthogonalisation is performed to produce an orthogonal basis, $\underline{q}^i$, for the residuals using

$$\begin{aligned} \underline{q}^1 &= \underline{r}^1 \\ \underline{q}^i &= \underline{r}^i + \sum_{j=1}^{i-1} \underline{q}^j \tau_{ij} \end{aligned} \qquad (3.6.44)$$

where

$$\tau_{ij} = \frac{(\underline{q}^j)^T \underline{r}^i}{(\underline{q}^j)^T \underline{q}^j} \qquad (3.6.45)$$

The minimum residual solution is begun by setting

$$\Upsilon_1^j = \frac{(\underline{r}^0)^{\mathrm{T}} \underline{q}^i}{(\underline{q}^j)^T \underline{q}^j} \qquad j = 1 \; to \; l \qquad (3.6.46)$$

Then

$$\begin{aligned} \Upsilon_0^l &= \Upsilon_1^l \\ \Upsilon_0^j &= \Upsilon_1^j - \sum_{i=j+1}^{l} \tau_{ji} \Upsilon_0^i \\ \Upsilon_2^j &= \Upsilon_0^{j+1} + \sum_{i=j+1}^{l-1} \tau_{ji} \Upsilon_0^{i+1} \end{aligned} \qquad (3.6.47)$$

The updates to the solutions are given by

$$\begin{aligned}
\underline{u}^0 &= \underline{u}^0 - \sum_{j=1}^{l} \Upsilon_0^j \underline{u}^j \\
\underline{\phi}^0 &= \underline{\phi}^0 + \sum_{j=1}^{l-1} \Upsilon_0^j \underline{q}^j + \Upsilon_2^l \underline{r}^0 \\
\underline{r}^0 &= \underline{r}^0 - \sum_{j=1}^{l} \Upsilon_1^j \underline{q}^j
\end{aligned} \tag{3.6.48}$$

§3.6.7 GMRES

The GMRES(k) algorithm[77] searches for the best possible solution $\underline{r}^k$, over the Krylov subspace $< \underline{r}^0, \underline{\underline{A}}\,\underline{r}^0, \underline{\underline{A}}^2\,\underline{r}^0, \ldots, \underline{\underline{A}}^{k-1}\,\underline{r}^0 >$ by solving the minimisation problem

$$\begin{aligned}
\underline{x}^k &= \underline{x}^0 + \underline{r}^k \\
\|\underline{r}^k\| &= \min_{y} \|\underline{r}^0 + \underline{\underline{A}}\,\underline{y}\| \\
\underline{r}^0 &= \underline{b} - \underline{\underline{A}}\,\underline{x}^0 \\
\underline{r}^k &= \underline{b} - \underline{\underline{A}}\,\underline{x}^k
\end{aligned} \tag{3.6.49}$$

The methods used to calculate the orthogonal base of the Krylov space using Gram-Schmidt orthogonalisation and the calculation of the minimum residual are exactly the same as for the BiCGSTAB(l) solver. The obvious disadvantage of this method is that as k increases the storage requirements rise linearly and the number of operations rises quadratically. In order to reduce these requirements the GMRES(k,m)[77] algorithm was proposed. This algorithm amounts to a restarted GMRES where on the iteration when k is equal to m the search vectors are discarded and the process begins again.

§3.6.8 Pre-Conditioning Methods

In many cases the use of a suitable pre-conditioner can lead to a solution being obtained in significantly fewer iterations. The main difficulties concerned with the use of a pre-conditioner are the need for extra storage and the extra computational overhead. The simplest pre-conditioner is the Jacobi pre-conditioner which divides each of the set of linear equations by the diagonal element. This method can be advantageous if there is considerable differences is the magnitude of the diagonal terms but has limited use for many of the matrices obtained in the field of CFD.

A more useful set of methods are those based on the Incomplete Lower Upper (ILU) factorisation of the matrix being solved, $\underline{\underline{A}}$. In the ILU(0) decomposition the only non-zero entries in both the lower and upper matrices correspond to non-zero entries in the $\underline{\underline{A}}$ matrix. It should be noted that the non-zero entries in any row of $\underline{\underline{A}}$ correspond to the neighbours of the element represented by the row. In all other respects the decomposition is performed as in a complete LU decomposition. For ILU(1) non-zero entries will be generated in a row in positions corresponding to neighbours of the neighbours of the element. In general in the ILU(n) decomposition non-zero elements can be generated for any element which is a (n+1)'th neighbour of the current element. The matrices used in the CFD field are highly sparse. If m is the largest number of faces bounding any element, used in the meshing of the solution domain, then the largest number of non-zero entries on any row of the discretised equation is (m+1). The use of anything beyond ILU(0) factorisation will produce storage requirements larger than that required by the system matrix, in extreme cases the storage can be significantly larger. As a consequence this examination, into the benefits of pre-conditioning methods, has been restricted to the consideration of ILU(0) decomposition.

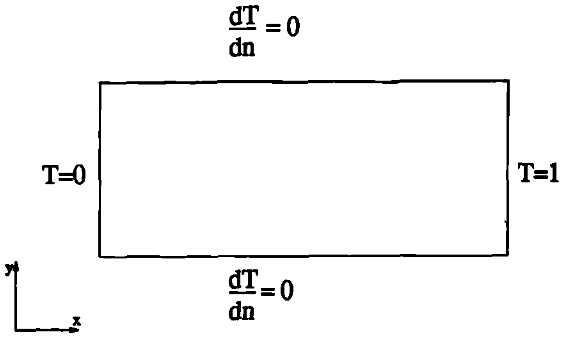

Figure 3.6.1: Boundary Conditions for Solver Test Case

§3.6.9 Test Cases

The main test case used in the evaluation of the solvers presented in the previous sections, involved heat transfer along a bar. The boundary conditions used, shown in Figure 3.6.1, guarantee that the problem can be solved in a single outer iteration of the code. The 2 dimensional test case employed a mesh with 10 elements in the y direction and 100 elements in the x direction making a total of 1000 elements. With the initial temperature field set to zero the various solvers took the following times and iterations

Solver	Iterations	Time (sec)
JOR	1488	0.273
SOR	793	0.167
Residual JOR	1488	0.874
Residual SOR	793	0.440
CGM	83	0.028
Bi-CG	83	0.045
BiCGSTAB(2)	46	0.051
GMRES(k,4)	324	0.157
GMRES(k,10)	150	0.094

The above results were obtained without the use of any pre-conditioner with the exception of the JOR, SOR and their residual equivalents where the system matrix is normalised, all rows are divided by their diagonals, prior to solution. Under these conditions the conjugate gradient type solvers require the fewest iterations and the least time. If the Jacobi pre-conditioner is used the table for the same simulation now becomes

Solver	Iterations	Time (sec)
CGM	81	0.032
BiCG	81	0.051
GMRES(k,4)	144	0.091
GMRES(k,10)	140	0.098

The use of the Jacobi pre-conditioner has slightly reduced the number of iterations required in all cases except for GMRES(k,4) which needed an unexpectedly high number of iterations when used without a pre-conditioner. For this solver the number of iterations has halved and it is now comparable in run time with the GMRES(k,10) with the advantage that it requires less storage.

If the ILU(0) pre-conditioner is used then the iterations and run times become

Solver	Iterations	Time (sec)
CGM	27	0.028
BiCG	23	0.043
GMRES(k,4)	28	0.039
GMRES(k,10)	30	0.042

In all cases there has been a significant reduction in the number of iterations required. The GMRES now becomes competitive with the various CG methods both in terms of iterations needed to obtain convergence and run time. Their disadvantage of requiring extra storage is now the only drawback to the use of the GMRES. On larger problems a similar behaviour is observed with the conjugate gradient solvers having a fractional advantage over the GMRES as the problem size increases.

The figures presented here indicate the relative merits of the various solvers when used to obtain a very nearly exact solution of a linear system of equations. The problem in CFD is that, in nearly all simulations, at any stage considerable assumptions have been made in the evaluation of the coefficients in the system matrix and source vector. These assumptions usually mean that a very accurate solution of the linear system is of limited benefit in obtaining the solution of the problem.

A test case more representative of those faced in CFD simulations is the moving lid cavity problem. For this case the coefficients in all linear equations depend directly on the convection fluxes which are calculated from the solved velocities. For this type of problem the velocity field is mainly calculated as a result of the corrections resulting from the use of the SIMPLE algorithm. This means that accurate solution of the discretised momentum equation is unnecessary, in fact instabilities frequently arise if more than a low number of momentum solver iterations are used. For this test case, 2 solver iterations are used in the solution of the velocity components, 5 solver iterations causing divergence early in the simulation for all solvers. The reason for this sensitivity is not fully clear. Consequently a fuller investigation, into gaining an understanding of strategies to use in the running of a unstructured code, may be advantageous. Twenty solver iterations were used for the pressure correction equation, this being an optimal value for reducing the overall time taken for the simulation. A solution was obtained to an accuracy of 10^{-5}. Under these conditions the number of outer iterations, in every outer iteration a system matrix is set up for each of the solved variables, was as follows

Solver	Iterations
SOR	238
BiCG	245
GMRES(k,4)	245

Given the extra storage and additional operations of these methods the SOR is the ideal choice for this problem.

The behaviour highlighted in these two test cases is representative of the majority of CFD simulations. The GMRES method or the variations of the Conjugate Gradient method do offer the advantages that they progress the solution much quicker than the point by point methods. The problems arise because of the approximations made in the calculation of the system matrix. As a consequence the solution of the set of linear equations at any stage of the simulation is unlikely to be equal to the final solution. In many cases the iterative solution procedure, although solving the matrix equation accurately, can move the current estimate of the solution away from the final solution. This causes instabilities and often divergence. For the highly coupled problems of interest in this study it has been found that only the use of Jacobi Over Relaxation (JOR) on the momentum components and Successive Over Relaxation (SOR) on the pressure correction equation consistently allows solution. This can mean that in an outer iteration the values of velocity and pressure may change by relatively small quantities. This constrains the changes needed in the scalar variables and leads to a smoother, if somewhat slow, path to convergence. On the scalar variables, particularly ones that are very loosely coupled, it is possible to use the whole field solution (GMRES and the varieties of Conjugate Gradient) methods. Despite this the usual choice is to use the SOR method on all such scalar variables.

CHAPTER FOUR

Models of Physical Processes

In the previous chapter the methods used in the solution of laminar flow problems were presented. This chapter presents the remaining algorithms which allow a simulation involving swirling, turbulent, reacting flow to be performed. The first section of this chapter deals with the transport of the enthalpy, in particular the handling of the diffusion term in the enthalpy conservation equation. When combustion processes are taking place in the flow, energy will be released, or absorbed, as the reactions take place. The modelling of the combustion processes is performed within the framework of the Simple Chemical Reaction Scheme(45). The transport of energy by means of radiation effects is modelled through the use of a composite radiosity model(78). For completeness, although not directly relevant to this work but part of the CWNN framework, the algorithms used in the simulation of solidification, with the associated release of enthalpy, are discussed at the end of this section.

The second section in this chapter concerns the modelling of turbulent effects. The most commonly used turbulence model is the standard k-ϵ model. Details of the conservation equations are presented along with the algorithms used to evaluate turbulence near walls and the associated effects on other variables. In addition, the methods used in the linearisation of the source terms in the turbulent kinetic energy and dissipation rate equations are discussed. The isotropic nature of the standard k-ϵ model means that it does not model swirling flows particularly well. As an alternative the ReNormalisation Group k-ϵ model(79, 80) was investigated. Claims made regarding its ability to model swirling flows and its very close resemblance to the standard k-ϵ model made the RNG method an attractive alternative. Finally, in this section, an anisotropic (mixing length, k-ϵ) hybrid model is presented. This model directly addresses the non-isotropic nature of swirling flows by using different length scales in the three cylindrical polar coordinate directions.

The chapter concludes with the algorithms which contribute to the tracking of particles in an unstructured mesh code. In the general interpolation scheme there is a need for the tracking of massless particles which are used to determine the position of the flow axis. In this section the algorithms are generalised to allow the tracking of particles with mass. Algorithms to handle areas such as collisions with boundaries and particle instabilities, potentially leading to the fragmentation of the droplet, are presented.

§4.1 The Enthalpy Equation

The enthalpy conservation equation is of the form

$$\frac{\partial(\rho h)}{\partial t} + \mathrm{div}(\rho \underline{u} h) \;=\; \mathrm{div}\left\{k\,\mathrm{grad}\left(\frac{h}{c_p}\right)\right\} + S_h \qquad (4.1.1)$$

where k is the thermal conductivity, c_p the specific heat and S_h represents any source of enthalpy. The discretisation of this equation follows the methods discussed in previous chapters with the only area of special interest being the handling of the specific heat within the diffusion term.

In the momentum equation the diffusion term discretises to the form

$$\kappa(v_A - v_P) \qquad (4.1.2)$$

for some κ. This equation has the same coefficient multiplying both v_A and v_P and so will lead to the same value being added to the system matrix, although with different signs, on the diagonal and in the position corresponding to node A. This ensures that diagonal dominance is maintained. In the case of the enthalpy equation the diffusion term discretises to

$$\frac{k_f A_f}{d_{AP}} \left\{ \left(\frac{h}{c_p}\right)_A - \left(\frac{h}{c_p}\right)_P \right\} \underline{n}.\underline{v}_{AP} \tag{4.1.3}$$

for any particular face where k_f is the estimated value of the conductivity on the face, A_f the face area, d_{AP} is the distance between the nodes A and P, $\underline{n}$ is the unit outward normal to the face and $\underline{v}_{AP}$ is a unit vector from P to A. It will frequently be the case that the specific heat will have different values at the two nodes and so steps must be taken to ensure that there is diagonal dominance in any resulting system matrix. Obviously equation (4.1.3) should be transformed to

$$\frac{k_f A_f}{d_{AP}}[\kappa(h_A - h_P) + \beta] \, \underline{n}.\underline{v}_{AP} \tag{4.1.4}$$

with κ and β being chosen such that if the specific heat has the same values at nodes A and P then β is equal to zero. The forms used are

$$\kappa = \frac{\alpha(c_p)_A + (1-\alpha)(c_p)_P}{(c_p)_A(c_p)_P} \tag{4.1.5}$$

$$\beta = \left[\alpha h_A^l + (1-\alpha)h_P^l\right] \left\{ \frac{(c_p)_P - (c_p)_A}{(c_p)_P(c_p)_A} \right\} \tag{4.1.6}$$

where $\alpha = d_{Af} \ / \ d_{AP}$ with d_{Af} being the distance of A from the face along the vector $\underline{v}_{AP}$ and the superscript l indicates the previous iteration value of the variable.

Having solved the discretised form of equation (4.1.1) to obtain the values of the enthalpy at the element centroids it is possible to calculate the temperature field. It is assumed that

$$h = c_p T \tag{4.1.7}$$

This equation allows the temperature to be obtained from the enthalpy if the value of the specific heat is known. In the case of the specific heat being a function of temperature an iterative approach is required to solve equation (4.1.7).

§4.1.1 Combustion

In most real combustion processes many intermediatory chemical species may be created prior to reaching the final oxidation products. To model this process accurately requires the calculation of the mass fractions of all these species at each stage of combustion. This would require the solution of a large number of coupled scalar equations. In general the large amount of computational time required to resolve these equations can not be justified when the main interest is the outcome of the combustion rather than the steps taken to reach the final stage. What is required is a simplified approach which allows the modelling of the combustion process through the solution of a small number of equations. One such methodology is the Simple Chemical Reaction System (SCRS)(45).

§4.1.1.1 Simple Chemical Reaction Scheme

In the SCRS the many routes of the combustion process, the intermediate chemical species and the combining ratios are replaced by a single formula

$$1 \text{ kg fuel} + s \text{ kg oxidant} \longrightarrow (1+s) \text{ kg product} \qquad (4.1.8)$$

where s is a constant termed the stoichiometric ratio. The above equation allows for any, or all, of the constituents in the reaction to consist of mixtures e.g. the oxidant could be air as long as the product is also a mixture consisting of the unreacting species in air such as N_2. If equation (4.1.8) is divided through by (1+s) then the equation

$$f_{st} \text{ kg fuel} + (1\text{-}f_{st}) \text{ kg oxidant} \longrightarrow 1 \text{ kg product} \qquad (4.1.9)$$

is obtained where f_{st} is the stoichiometric mixture fraction for the reaction and is equal to $1/(1+s)$.

The aim of the SCRS is to calculate the mass fractions of fuel (m_{fu}), oxidant (m_{ox}) and product (m_{pr}) throughout the solution domain. These values may then be used to calculate the fluid properties depending on the assumptions being employed. When using the SCRS it is common to use the ideal gas law to calculate the density

$$\rho = \frac{pW}{RT} \tag{4.1.10}$$

where p is the pressure, T the temperature, R the gas constant and W the mixture molecular weight calculated using the equation

$$\frac{1}{W} = \frac{m_{fu}}{W_{fu}} + \frac{m_{ox}}{W_{ox}} + \frac{m_{pr}}{W_{pr}} \tag{4.1.11}$$

where W_x is the molecular weight of species x. It is also possible to calculate a mixture specific heat

$$c_p = m_{fu}\, c_{fu} + m_{ox}\, c_{ox} + m_{pr}\, c_{pr} \tag{4.1.12}$$

where c_x is the specific heat of species x although frequently it is assumed that each species has the same value for c_p.

When using the SCRS the mixture enthalpy, h, is treated as the conserved variable and this is related to the temperature by the equation

$$h = c_pT + m_{fu}H \tag{4.1.13}$$

where H is the heat of reaction. As a consequence of this equation, when solving for enthalpy, any source of mass fraction of fuel has to be mirrored by a corresponding enthalpy source.

There are two main variations of the SCRS which are described in the following sections.

§4.1.1.2 Diffusion Controlled Model

In the diffusion controlled variant of the SCRS it is assumed that fuel and oxidant can not coexist in any element. If, through any transport mechanism, both fuel and oxidant are present in an element they react instantaneously until one of the species is completely destroyed. When using the model a single extra conservation equation is solved for a quantity termed the mixture fraction, f,

$$\frac{\partial(\rho f)}{\partial t} + \mathrm{div}(\rho \underline{u} f) = \mathrm{grad}\left[\Gamma\, \mathrm{div} f\right] \tag{4.1.14}$$

where Γ is the diffusion coefficient for the mixture fraction. Its value is conventionally taken to be the same as the value of the diffusion coefficient in the enthalpy equation, i.e. the thermal conductivity divided by the specific heat. The mass fractions of the species are calculated from the mixture fraction using the formulae

$$\begin{array}{lll} f < f_{st} & \left\{ \begin{array}{ll} m_{fu} & = 0 \\ m_{ox} & = \frac{f_{st}-f}{f_{st}} \end{array} \right. & \\ f \geq f_{st} & \left\{ \begin{array}{ll} m_{fu} & = \frac{f-f_{st}}{1-f_{st}} \\ m_{ox} & = 0 \end{array} \right. & \\ \text{all } f & \quad m_{pr} \quad = 1 - m_{ox} - m_{fu} & \end{array} \tag{4.1.15}$$

§4.1.1.3 Kinetically Influenced Model

The kinetically influenced method assumes that both oxidant and fuel can coexist in an element for a limited time. The model solves the same conservation equation for mixture fraction, as in the diffusion controlled model, but also determines the mass fraction of fuel by the solution of its own transport equation

$$\frac{\partial(\rho m_{fu})}{\partial t} + \text{div}(\rho \underline{u} m_{fu}) = \text{grad}\,[\Gamma\, \text{div}(m_{fu})] + R_{fu} \tag{4.1.16}$$

where the source term R_{fu} represents the reaction rate of fuel and the diffusion coefficient, Γ, takes the same value as the mixture fraction diffusion coefficient. There are a number of models available for this source term one of which is the Eddy Break-up model(81,82). The Eddy Break-up model assumes that the reaction rate is controlled by the turbulent eddies which cause the species to mix. This is modelled by making the reaction rate a function of the levels of the turbulent variables k and ϵ. In this case

$$R_{fu} = -C_R m_{fu} \rho \frac{\epsilon}{k} \tag{4.1.17}$$

where C_R is termed the eddy break-up constant and is frequently assigned the value of unity. In order to prevent negative mass fractions of fuel and aid numerical stability this source term is linearised(83) into the form

$$S_{m_{fu}} = R^l_{fu} V - \frac{R^l_{fu} m^l_{fu} V}{m^l_{fu} - m_{fu,b}} + \frac{R^l_{fu} m_{fu} V}{m^l_{fu} - m_{fu,b}} \tag{4.1.18}$$

where V is the volume of the element, the superscript l signifies the previous iteration value of the relevant quantity and $m_{fu,b}$ is the mass fraction of burnt fuel which is calculated using the formula

$$m_{fu,b} = \max\left[0, \frac{f - f_{st}}{1 - f_{st}}\right] \tag{4.1.19}$$

The mass fractions of oxidant and product are calculated from the formulae

$$\begin{aligned} m_{ox} &= s\left[m_{fu} - \frac{f - f_{st}}{1 - f_{st}}\right] \\ m_{pr} &= 1 - m_{ox} - m_{fu} \end{aligned} \tag{4.1.20}$$

§4.1.2 Radiation

In many processes radiation plays a significant role in the transport of energy. The transfer of energy due to radiation in an absorbing, scattering and emitting medium is formulated in the Radiative Transfer Equation(84) (RTE). The nature of this equation, and the potential spectral variation of the radiative properties, means that it is usually very difficult to solve. As a consequence, simplifying assumptions are usually made which has led to many different radiation models(85). Many of these models, for example the flux radiation model(86–88), solve for the radiation fluxes along selected directions. The use of a suitable mesh aligned with these selected directions makes this approach highly attractive. Unfortunately this type of approach is not particularly suitable when using unstructured meshes.

One method, for modelling the radiative heat transfer on unstructured meshes, is to solve an equation for the composite radiosity(78). The radiosity is an average of the incoming and outgoing radiation fluxes integrated over all directions of the solid angle. The equation for the radiosity, R, takes the form

$$\text{div}\left[\frac{4}{3(a+s)}\text{grad}R\right] + a*(E-R) \;=\; 0 \qquad (4.1.21)$$

where a is the absorption coefficient (proportion of radiation absorbed per unit length), s is the scattering coefficient (proportion of radiation scattered per unit length) and E is the black body emissive power of the fluid calculated using the formula

$$E \;=\; \sigma T^4 \qquad (4.1.22)$$

where T is the temperature of the fluid and σ is the Stefan- Boltzmann constant. Equation (4.1.21) is a simplified form of the general equation with both the convection and transient terms absent. This should allow its solution using the techniques employed for any of the other scalar variables.

Transfer of heat through radiation leads to a source in the enthalpy equation equal to the negative of the source in the radiosity equation (4.1.21)

$$S_h = a * (R - E) \tag{4.1.23}$$

§4.1.3 Solidification

When the effects of solidification are being modelled there is an extra source into the enthalpy conservation equation equal to

$$-L\frac{\partial(\rho f)}{\partial t} - L\,\mathrm{div}(\rho \underline{u} f) \tag{4.1.24}$$

where L is the latent heat of solidification and f is the liquid fraction[89]. The first of the above sources models the release of energy as the material solidifies and the second models the convective movement of energy held by the liquid form of the material. The value of the liquid fraction is usually calculated from the formulae

$$f = \begin{cases} 1 & \text{if } T > T_L \\ (T - T_S)/(T_L - T_S) & \text{if } T_S \le T \le T_L \\ 0 & \text{if } T < T_S \end{cases} \tag{4.1.25}$$

where T_L is the liquidus temperature, the temperature above which the material is fully liquid, and T_S is the solidus temperature, the temperature below which the material is completely solid. The liquid fraction allows the phase change to be modelled as it moves through the solution domain. This allows the numerical solution to the problem to be obtained without the use of a mesh which moves with the solidification front. For each time step the method used, starting with old time step values for all variables, is

1. Obtain values for the material properties given the current temperature.
2. Solve the discretised enthalpy equation.
3. Calculate the temperatures.
4. Correct the liquid fraction.

These steps continue until convergence has been obtained and the solution procedure moves on to the next time step. The first of the above steps will depend on the equations being used whilst step 2 was covered in section 4.1. The temperature value is obtained from the enthalpy by use of the relevant relationship which is usually

$$h = c_p T$$

Calculating the liquid fraction directly from equation (4.1.25) can lead to large oscillations in the solution. This is especially true if the solidus and liquidus temperatures are equal. In this case a very small decrease in temperature can cause an element to change from liquid to solid. This releases a large amount of energy causing the temperature to rise and the element to return to liquid. This reabsorbs all the energy released and the liquid - solid - liquid cycle begins again. To reduce the occurrences of this type of oscillation a correction to the liquid fraction, Δ, is calculated. A number of equations can be used to calculate this correction(89–91). One of the more general methods is that suggested by Swaminathan(92) for which the correction is given by

$$\Delta = \frac{h_P - c_p F^{-1}(f)}{\dfrac{V\delta H_P}{a_P} + \dfrac{\partial F}{\partial T} c_p} \tag{4.1.26}$$

In this formula h_P is the nodal enthalpy value, $F(T)$ is the function used to calculate the liquid fraction given the temperature, V is the volume of the control volume and a_P is the diagonal element of the enthalpy system matrix associated with node P and

$$\delta H_P = \frac{\rho L}{\delta t} \tag{4.1.27}$$

Hence given the formula (4.1.25) for the liquid fraction then equation (4.1.26) can be written as

$$\Delta = \frac{h_P - c_p\{(T_L - T_S)f + T_S\}}{\frac{V\rho L}{a_P\delta t} + c_p(T_L - T_s)} \tag{4.1.28}$$

A useful discussion of correction methods available can be found in Voller and Swaninathan(93) which covers a number of methods along with potential shortcomings of each method.

§4.2 Turbulence

In the work described in this and the previous chapters the conservation equations solved by means of discretisation were the continuity equation

$$\frac{\partial\rho}{\partial t} + \mathrm{div}(\rho\underline{U}) = 0.0 \tag{4.2.1}$$

the momentum conservation equations

$$\frac{\partial(\rho U_i)}{\partial t} + \mathrm{div}(\rho\underline{U}U_i) = \frac{\partial P}{\partial x_i} + \mathrm{div}(\mu\mathrm{grad}U_i) + S_i \tag{4.2.2}$$

and the enthalpy equation

$$\frac{\partial(\rho h)}{\partial t} + \mathrm{div}(\rho\underline{U}h) = \mathrm{div}\left[k\,\mathrm{grad}\left(\frac{h}{c_p}\right)\right] + S_h \tag{4.2.3}$$

the equations being expressed in tensor notation where applicable. In these conservation equations the quantities U_i, P and h represent respectively the instantaneous velocity component in the x_i direction, the instantaneous pressure and the instantaneous enthalpy. It is assumed that these values consist of a time averaged component and a fluctuating term

$$U_i = \overline{u_i} + v_i \quad P = \overline{p} + \pi \quad h = \overline{h} + \eta \tag{4.2.4}$$

Substitution of the equalities of (4.2.4) into the conservation equations (4.2.1) - (4.2.3) and dropping the overlines signifying the time averaged values, produces for the continuity equation

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho\, \underline{u}) \;=\; 0.0 \tag{4.2.5}$$

the momentum conservation equations become

$$\frac{\partial(\rho u_i)}{\partial t} + \text{div}(\rho \underline{u} u_i) \;=\; \frac{dp}{dx_i} + \text{div}(\mu \text{grad} u_i) - \frac{\partial}{\partial x_j}(\rho \overline{v_i v_j}) + S_i \tag{4.2.6}$$

and the enthalpy equation transforms into

$$\frac{\partial(\rho h)}{\partial t} + \text{div}(\rho \underline{u} h) \;=\; \text{div}\left[k \text{ grad}\left(\frac{h}{c_p}\right)\right] - \frac{\partial}{\partial x_i}(\rho \overline{v_i \eta}) + S_h \tag{4.2.7}$$

The above equations are the governing equations for the mean flow quantities. No approximations have been made in their derivation. The non-linearity of the equations (4.2.2) and (4.2.3) has produced terms in equations (4.2.6) and (4.2.7) which are the correlations between fluctuating velocities, $\rho\overline{v_i v_j}$, and between velocity and enthalpy $\rho\overline{v_i \eta}$. The correlations represent the transport of momentum and heat due to turbulent motion. The term -$\rho\overline{v_i v_j}$ is the transport of x_i momentum in the x_j direction and it acts as a stress on the fluid so it is termed the turbulent or Reynolds stress; -$\rho\overline{v_i \eta}$ is the transport of enthalpy in the x_i direction and is termed the turbulent heat flux. For a fully turbulent flow the turbulent stresses and fluxes are much greater than the laminar diffusion terms. The equations (4.2.5) - (4.2.7) can be solved for the mean values of the velocity, pressure and enthalpy once the correlation terms have been determined. Transport equations for these correlations exist but they contain turbulence correlations of higher orders. This means that closure of these equations can not be obtained, as conservation equations for any order of correlation contain terms involving higher order correlations. Thus an alternative approach is required to evaluate the correlations.

Turbulence occurs at high Reynolds numbers and is the cause of an eddying motion, with a large range of eddy sizes, in the fluid. The movement of a fluid is turbulent if it is rotational, diffusive, intermittent, highly disordered and dissipative. The largest eddies, which are associated with the low frequency fluctuations, are determined by the boundary conditions of the flow and their size is of the same order as that of the flow domain. The smaller eddies are determined by the viscous effects. The transport of momentum and heat is mainly effected by the large scale turbulent motion and this is the major contributor to the correlation terms. As a consequence it is the large scale motion which needs to be simulated in the turbulence model in order to determine the correlation terms and hence the velocities and the enthalpy fields.

The larger scale eddies interact with the flow extracting kinetic energy from the average motion and setting up the large scale turbulent motion. The eddies act as vortex elements which stretch each other. In this way the energy is transmitted to smaller scale eddies. The process continues until the viscous forces dissipate the energy. The rate at which energy, from the mean flow, is transformed into turbulent motion is determined from the large scale motion. The energy transformed is the only energy available to the smaller scale motion and hence the only energy to be dissipated. Thus the rate of dissipation of energy is also determined by the large scale motion despite dissipation being effected by viscous actions and taking part in the small eddies. The larger the Reynolds number, i.e. the smaller the viscous effects, the smaller the dissipative eddies are relative to the large scale motion.

Boussinesq$^{(94)}$ introduced the eddy viscosity concept which assumes that the turbulent stresses are proportional to the mean velocity gradients in a similar manner to viscous stresses in laminar flows. This is expressed as

$$-\overline{v_i v_j} = \nu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \tag{4.2.8}$$

where ν_t is defined as the turbulent or eddy viscosity. The turbulent viscosity is not a fluid property but is a value dependent on the turbulence and so may vary from place to place within the fluid. This does not overcome the difficulties of solving a turbulent problem but it does reduce the problem to one of determining the value of ν_t. The term in (4.2.8) which includes the Kronecker delta is necessary because the sum of the normal stresses must be equal to twice the kinetic energy. Without this term the normal stresses are given by

$$\overline{v_1^2} = -2\nu_t\frac{\partial u_1}{\partial x_1}, \quad \overline{v_2^2} = -2\nu_t\frac{\partial u_2}{\partial x_2}, \quad \overline{v_3^2} = -2\nu_t\frac{\partial u_3}{\partial x_3} \tag{4.2.9}$$

the sum of which is zero by the continuity equation. The formula for the kinetic energy of the fluctuating motion is given by

$$k = \frac{1}{2}\left(\overline{v_1^2}+\overline{v_2^2}+\overline{v_3^2}\right) \tag{4.2.10}$$

The Kronecker delta term assures that the sum of the normal stresses is equal to the required $2k$. The normal stresses act in a direction perpendicular to faces in the same manner as pressure forces and as k is a scalar the second part of (4.2.8) constitutes a pressure. In the elimination of the correlation term in the momentum equation the second part is absorbed into the pressure gradient term. Thus the appearance of k in equation (4.2.8) does not force its calculation in order to solve the momentum equation, only the determination of the turbulent viscosity is required.

Analogue to the turbulent momentum transport the turbulent heat transport is assumed to be proportional to the gradient of the enthalpy

$$-\overline{v_i\eta} = \Gamma\frac{\partial h}{\partial x_i} \tag{4.2.11}$$

where Γ is the turbulent diffusivity of heat. In the same way as the eddy viscosity is not a fluid property Γ depends not on the fluid but on the state of turbulence. The Reynolds analogy between heat and momentum transport implies that the turbulent diffusivity is related to ν_t by

$$\Gamma = \frac{\nu_t}{\sigma_t} \tag{4.2.12}$$

where σ_t is the turbulent Prandtl number for enthalpy. It has been shown experimentally that the Prandtl number does not vary to any great extent across a flow or between flows. The value of σ_t is a function of buoyancy(95) but in what follows the value of 1.0 will be given to it.

A number of turbulent models exist from the zero equation models, such as those based on the assumption of constant eddy viscosity or those involving deriving equations for the mixing length, all the way through to the higher order models which employ more than two equations. The lower order models offer the advantages of being computationally cheap whilst giving adequate answers for a number of problems. The greater the number of equations used the more the potential that the scheme will produce an adequate solution to a general problem. One of the most widely used methods is the k-ϵ model.

§4.2.1 Standard k-ϵ Turbulence Model

At high Reynolds numbers the rate of dissipation of kinetic energy ϵ is equal to the viscosity multiplied by the fluctuating vorticity. An exact transport equation for the fluctuating vorticity, and thus the dissipation rate, can be derived from the Navier Stokes equation. The k-ϵ model consists of the turbulent kinetic energy equation

$$\frac{\partial k}{\partial t} + \mathrm{div}(\rho \underline{u} k) = \mathrm{div}\left(\left[\mu_{lam} + \frac{\rho\nu_t}{\sigma_k}\right] \mathrm{grad} k\right) + \rho\nu_t G - \rho\epsilon \tag{4.2.13}$$

and the dissipation rate equation

$$\begin{aligned} \frac{\partial \epsilon}{\partial t} + \mathrm{div}(\rho \underline{u} \epsilon) &= \mathrm{div}\left(\left[\mu_{lam} + \frac{\rho\nu_t}{\sigma_\epsilon}\right] \mathrm{grad} \epsilon\right) \\ &+ C_{1\epsilon}\rho\nu_t G\frac{\epsilon}{k} - C_{2\epsilon}\rho\frac{\epsilon^2}{k} \end{aligned} \tag{4.2.14}$$

where G represents the turbulent generation rate which is equal to

$$\begin{aligned} G = 2\left(\left[\frac{\partial u}{\partial x}\right]^2 + \left[\frac{\partial v}{\partial y}\right]^2 + \left[\frac{\partial w}{\partial z}\right]^2\right) + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 \\ + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right)^2 \end{aligned} \qquad (4.2.15)$$

In the implementation of this model the Kolmogorov - Prandtl expression for the turbulent viscosity is used

$$\nu_t = C_\mu \frac{k^2}{\epsilon} \qquad (4.2.16)$$

together with equation (4.2.11) for the eddy diffusivity. In the above C_μ, σ_k, σ_ϵ, $C_{1\epsilon}$ and $C_{2\epsilon}$ are all taken to be constants and are given respectively the values(96) 0.09, 1.0, 1.3, 1.44 and 1.92.

The methods used in the discretisation of the above equations are the same as those defined in previous chapters with the term on the right of each of (4.2.13) and (4.2.14) being treated as source terms. The continuity equation remains unchanged whilst the only differences in the momentum and enthalpy equations is the change in the diffusion coefficient. The laminar viscosity $\rho\nu_{lam}$ is replaced by $\rho\nu_{lam} + \rho\nu_t$ and the conductivity k in the enthalpy equation is replaced by $k + (\rho\nu_t/\sigma_t)$.

Handling of the boundary conditions concerned with turbulent flows is the same as for laminar flows except in the case of walls. The k-ϵ model provides accurate solutions only for fully turbulent flows. In the part of the flow near to walls there exist regions in which the local Reynolds number of turbulence, $k^2/\nu\epsilon$, is so small that the viscous effects become more significant than the turbulent ones. In this viscous sublayer very steep gradients occur so for accurate modelling many grid points would be required in this region. Fortunately it is not necessary to discretise the k and ϵ conservation equations over this region as there exist suitable laws which relate the wall conditions to values of the dependent variables just outside the viscous sublayer.

In the region just outside the sublayer[97], such that

$$30 < \frac{yU_\tau}{\nu_{lam}} < 100 \tag{4.2.17}$$

where y is the perpendicular distance to the wall and U_τ is the resultant friction velocity, the generation of turbulent kinetic energy is balanced by its dissipation. In this region

$$\frac{\tau}{\rho k} = \frac{U_\tau^2}{k} = \sqrt{C_\mu} \tag{4.2.18}$$

where τ is the wall shear stress. The wall skin friction factor, s, which is defined by

$$s = \frac{\tau}{\rho V^2} \tag{4.2.19}$$

in which V is the fluid speed, is determined from the formula

$$\sqrt{s} = \frac{\kappa}{\ln(1.01 + E\,Re\sqrt{s})} \tag{4.2.20}$$

where κ, the Kaman constant, is equal to 0.435, E, the wall roughness parameter, usually being set to 9.0, which represents smooth walls, and Re, the local Reynolds number, which is defined by the formula

$$Re = \frac{Vy}{\nu_{lam}} \tag{4.2.21}$$

For a node positioned just outside the viscous sublayer the effective viscosity is set to the maximum of the laminar dynamic viscosity $\rho\nu_{lam}$ and sVy. From equation (4.2.18) the kinetic energy of turbulence at this point is given by

$$k = \frac{\tau}{\rho\sqrt{C_\mu}} = \frac{sV^2}{\sqrt{C_\mu}} \tag{4.2.22}$$

As previously stated in this region the production of turbulent kinetic energy is equal to its dissipation. This sets

$$\epsilon = U_\tau^2 \frac{\partial V}{\partial y} \tag{4.2.23}$$

The value of the gradient term is calculated by substituting (4.2.19) into equation (4.2.20) to remove s. Differentiation of the resulting equation, with respect to y, gives a formula for $\partial V/\partial y$

$$\begin{aligned} \frac{\partial V}{\partial y} &= \frac{1}{\kappa}\sqrt{\frac{\tau}{\rho}}\frac{1}{y + [\frac{E}{\nu_{lam}}\sqrt{\frac{\tau}{\rho}}}] \\ &\approx \frac{1}{\kappa}\sqrt{\frac{\tau}{\rho}}\frac{1}{y} \end{aligned} \tag{4.2.24}$$

Substitution of this into equation (4.2.23) and using (4.2.18) gives

$$\epsilon = \frac{0.1643}{\kappa}\frac{k^{1.5}}{y} \tag{4.2.25}$$

The source terms in the conservation equations (4.2.13) and (4.2.14) include the quantity G which is the rate of generation of turbulent kinetic energy the formula for which is given in equation (4.2.15). The calculation of G requires the values of the derivatives of the three Cartesian velocity components with respect to the three Cartesian directions. The simplest method for the estimation of any of these is to use the following formulae

$$\int_V \frac{\partial u_i}{\partial x_j} \approx V \left.\frac{\partial u_i}{\partial x_j}\right|_P \tag{4.2.26}$$

but using the divergence theorem

$$\int_V \frac{\partial u_i}{\partial x_j} = \sum_F \int_F u_i n_j dS \tag{4.2.27}$$

By making the usual approximations to the right hand side of (4.2.27) and equating to (4.2.26) the following estimate for the derivative's value at P is obtained.

$$\left.\frac{\partial u_i}{\partial x_j}\right|_P \approx \frac{1}{V}\sum_F A_F n_j (u_i)_F \qquad (4.2.28)$$

This formula only requires the estimation of a representative value of u_i on each face of the control volume and so can use the methods discussed previously, for example those included in the section on the cross product diffusion terms. The only care that is needed is in the handling of walls. At a wall there *is no generation of turbulence* so for near wall elements care is required in the calculation of the differentials.

The source terms, particularly those in the dissipation rate equation, are non linear. Three methods of linearisation are available within the code in line with those used by PHOENICS(60). Assuming that the linearised form of the sources can be written in the form S_C - $S_L\phi$ then the following gives the formulae for the coefficients for the various methods.

METHOD 1

k source terms :- $S_C = \nu_t \rho G$

$S_L = C_D \nu_t \rho / (C_\mu l^2)$

ϵ source terms :- $S_C = C_D \rho \nu_t C_{1\epsilon} G \nu_t / (C_\mu l^2)$

$S_L = C_{2\epsilon} C_D \rho \nu_t / (C_\mu l^2)$

METHOD 2

k source terms :- $S_C = (C_{2\epsilon} - 1.0) C_\mu C_D k^2 \rho / \nu_t + 1.5 \rho \nu_t G$

$S_L = 0.5 \nu_t \rho G / k + C_{2\epsilon} C_\mu C_D k \rho / n u_t$

ϵ source terms :- $S_C = (C_\mu C_D \rho k / \nu_t)(C_{1\epsilon} \nu_t G + (C_{2\epsilon} - 1.0)\epsilon)$

$S_L = (2.0 C_{2\epsilon} - 1.0) C_\mu C_D \rho k / \nu_t$

METHOD 3

k source terms :- $S_C = \nu_t \rho G$

$S_L = C_D C_\mu k\rho/\nu_t$

ϵ source terms :- $S_C = C_\mu C_D \rho k C_{1\epsilon} G$

$S_L = C_{2\epsilon} C_\mu C_D \rho k/\nu_t$

The earlier the method in the above list the quicker convergence will be obtained whilst the later methods offer greater numerical stability.

§4.2.2 ReNormalisation Group k - ϵ Model

ReNormalisation Group (RNG)(79,80) methods provide a general set of rules which allow physical problems to be expressed in terms of equations governing the large-scale, long-time behaviour of the system. This "coarse-graining" embodied in the RNG approach has been applied to areas as diverse as high-energy particle physics, critical phenomena and areas of fluid dynamics such as turbulence and combustion. The RNG method can be applied to any scale invariant phenomenon which has no externally imposed characteristic length and time scale.

One of the most important features of high Reynolds number turbulent flows is their large range of both space and time scales. The size of eddies in high Reynolds number, turbulent flows range from the energy - containing eddies of size L, the integral scale, down to the dissipation-scale eddies of size $L/R^{3/4}$, where R is a microscale Reynolds number. To accurately solve the three dimensional Navier-Stokes equation for such a turbulent flow would require storage of order $O(R^{9/4})$. As the time scale of turbulent eddies is of order $O(R^{3/4})$ the amount of computational work is order $O(R^3)$. If R is large both the number of operations and the storage requirements are prohibitively large.

High Reynolds number, turbulent flow can be characterised by three ranges of spatial scales

1. The energy spectrum is strongly anisotropic as well as being non universal for wavenumbers k=O(π/L). In this range the integral scale is affected by both the geometry of the flow and the physiochemical processes taking place.

2. At smaller scales, the inertial range of small scale eddies, the velocity fluctuation spectrum, $E(k)$, is approximately given by the Kolmogorov law

$$E(k) = C_k \bar{\epsilon}^{2/3} k^{-5/3} \tag{4.2.29}$$

   where C_k is the Kolmogorov constant and $\bar{\epsilon}$ is the energy dissipation per unit volume.

3. For wavenumbers in the range $k > O(K_d)$ the eddies have very low energy due to viscous dissipation. In this range the energy spectrum decreases exponentially with k.

Once the inertial range of eddies have been expressed in a quantitively accurate manner, the RNG methods(98, 99) can be applied to obtain coarse-grained equations for the turbulence related variables.

The range of scales of the effective excitation in turbulence lie between the low, energy containing wavenumber $k_0 = 2\pi/L$ and the high wavenumber viscous cutoff Λ. The RNG method removes a narrow band of modes near Λ by expressing them in terms of lower modes in the range $k_0 < k < \Lambda \exp^{-l}$. Having removed this band of modes the equations of motions for the remaining modes is a modified system of Navier-Stokes equations in which the eddy viscosity, force and non-linear coupling have all been affected. In turn, further modes are iteratively removed from the dynamics. Thus the RNG method produces a form of the Navier-Stokes equation which allows the computation on coarser grids.

The use of the RNG k-ϵ models requires the use of an equation for the effective viscosity, ν_{eff}, of the form

$$\nu_{eff} = \nu_{lam} + \nu_t = \nu_{lam}\left[1 + \sqrt{\frac{C_\mu}{\nu_{lam}}\frac{k}{\sqrt{\epsilon}}}\right]^2 \qquad (4.2.30)$$

In the above equation C_μ has the value 0.0845. In addition, the values of many of the constants in the conservation equations for turbulent kinetic energy and dissipation rate also take different values from those used in the standard k-ϵ model. The table below highlights the differences in the constant's values for the two models

	Standard k-ϵ	RNG k-ϵ
$C_{1\epsilon}$	1.44	1.42
$C_{2\epsilon}$	1.92	1.68
σ_k	1.0	0.7194
σ_ϵ	1.314	0.7194
C_D	0.1643	0.1567

In addition to the above changes in the values of the constants there is an extra source term in the dissipation rate equation which now becomes

$$\begin{aligned}\frac{\partial\epsilon}{\partial t} + \mathrm{div}(\rho\underline{u}\epsilon) &= \mathrm{div}\left(\left[\mu_{lam} + \frac{\rho\nu_t}{\sigma_\epsilon}\right]\mathrm{grad}\epsilon\right) \\ &+C_{1\epsilon}\rho\nu_t G\frac{\epsilon}{k} - C_{2\epsilon}\rho\frac{\epsilon^2}{k} - R\end{aligned} \qquad (4.2.31)$$

where the extra term, R, is given by

$$R = 2\nu_{lam} G_{ij}\overline{\frac{\partial u_l}{\partial x_i}\frac{\partial u_l}{\partial x_j}} \qquad (4.2.32)$$

In the RNG k-ϵ model the value of R is calculated using the formula

$$R = \frac{C_\mu\eta^3(1-\eta/\eta_0)}{1+\beta\eta^3}\frac{\epsilon^2}{k} \qquad (4.2.33)$$

where $\eta = \sqrt{G}k/\epsilon$, $G = 2G_{ij}G_{ij}$, and the constants η_0 and β take the values 4.8 and 0.012 respectively.

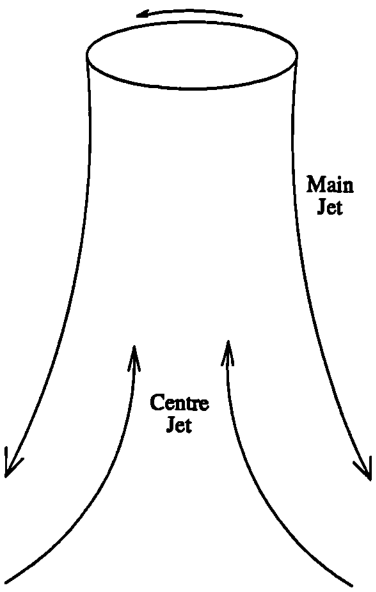

Figure 4.2.1: Main and Centre Jets

§4.2.3 Anisotropic Model

One of the main assumptions of the standard k-ϵ model is that the turbulence of the flow can be characterised by a length scale. This length scale is a measure of the size of the turbulent eddies and is assumed to be representative of all directions. This assumption is invalid for swirling flows in which the length scale is direction dependent. A non-isotropic$^{(100)}$ (mixing length, k-ϵ) hybrid model is suggested in an attempt to overcome some of the shortcomings of the standard k-ϵ model.

The model employs the standard k-ϵ model to calculate the length scale to be used for both the radial and axial components of velocity. In the tangential direction the model uses a hybrid of the k-ϵ model and a mixing length model. The mixing length model calculates the turbulent viscosity using the formula

$$\nu_m = \lambda^2 \sqrt{\left(\frac{\partial u_\theta}{\partial r}\right)^2 + \left(\frac{\partial u_z}{\partial r}\right)^2} + \nu_0 \qquad (4.2.34)$$

where the value of the mixing length, λ, and background turbulence, ν_0, are given by the formulae

$$\begin{aligned} \lambda &= c * 2.5 r_{1/2} \\ \nu_0 &= 100 * \nu_{lam} \end{aligned} \qquad (4.2.35)$$

For highly swirling flows there may be a region, near the flow axis, where the axial velocity component is in the opposite direction to the Main Jet. The flow in this region is termed the Centre Jet, see Figure 4.2.1. In this central recirculation region the value of c in equation (4.2.35) takes the value 0.18 whilst in the Main Jet it has value 0.09. These values have been chosen to be in agreement with values for the normalised turbulent length scale for an annular jet, for the Main Jet, and a round jet(101), for the Centre Jet. The symbol $r_{1/2}$ represents the jet half width. In the Centre Jet $r_{1/2}$ is the radial distance from the axis to the position at which the absolute value of the axial velocity drops to half its value on the axis. In the Main Jet $r_{1/2}$ is half the distance between two points, on either side of the maximum of the absolute value of the axial velocity, at which the absolute axial velocity drops to half its maximum value. If no Centre Jet exists then $r_{1/2}$ for the Main Jet uses the same formula unless the absolute axial velocity does not drop below half its maximum value on the axis side of the maximum. In this case the value for $r_{1/2}$ is the radial distance from the position of the maximum velocity to the position of the outer half maximum. The k-ϵ model is used for the calculation of the radial length scale outside the outer half width. Also once the maximum axial velocity has fallen below a prescribed limiting value it is assumed that swirling effects are not dominant and the k-ϵ model is used at all radii.

The only other change that is necessary, when using this anisotropic model, is to the equation for the turbulent generation rate, G. The standard k-ϵ model equation for G, (4.2.15), is used as the basis for the anisotropic model's calculation of G. The change to this formulation is that the radial gradient of the tangential velocity component is replaced by

$$\sqrt{\frac{\nu_m}{\nu_t}} \frac{\partial u_\theta}{\partial r} \qquad (4.2.36)$$

where ν_t is the value of the turbulent viscosity calculated using the k-ϵ model.

If CWNN is used to solve a problem over an axi-symmetric mesh the tangential flow direction corresponds to the y direction of the mesh. This means that ν_m is used as the turbulent viscosity in the v velocity conservation equation whilst ν_t is used in the u and w equations. For a cylindrical mesh the w velocity component uses ν_t but the turbulent viscosity for both the u and v velocity components is a function of both ν_t and ν_m. For an element with centre at position (r,θ,z), in the cylindrical coordinate system, the turbulent viscosities for the u and v velocity components are given by the equations

$$\begin{aligned} \nu_u &= \sqrt{(\nu_m \cos\theta)^2 + (\nu_t \sin\theta)^2} \\ \nu_v &= \sqrt{(\nu_m \sin\theta)^2 + (\nu_t \cos\theta)^2} \end{aligned} \qquad (4.2.37)$$

When the general interpolation scheme is being used the equation

$$\nu_i = \nu_t + (\nu_m - \nu_t)\, l_i^2 \qquad (4.2.38)$$

is used to calculate the turbulent viscosity in the i direction, In the above l_i is the i'th component of a unit vector in the flow's tangential direction.

§4.3 Particle Tracking

In the section 3.4, which details the methods used in the general interpolation scheme, use was made of algorithms which track massless particles. The current section will cover the more general algorithms included in CWNN which perform the tracking of particles of any mass, massless particle simply being a special case.

In addition, heat transfer, chemical reactions and fluid flow processes in the top space of the HIsmelt® reaction vessel are strongly influenced by the presence of metal droplets(9). These metal droplets are emitted form the metal bath into the top space in large enough numbers, in places above 5% of the volume, that they form a significant obstacle to the flow of the gas phase. Additionally, because the droplets are cooler than the gas they draw energy from the continuum phase. Chemical reactions in, and especially on the surfaces of, the droplets consume O_2 and CO_2 and release CO to the gas phase. Consequently, to accurately resolve the flow and heat transfer in the top space of the reaction vessel it is necessary to predict the presence, both in terms of volume fraction and surface area, of droplets. These required quantities are calculated after tracking a large number of particles. Further details of the use of tracking in the HIsmelt® processes are given alongside the results in Chapter 6.

At the start of the tracking procedure it is assumed that the position, velocity and properties of the particle are known. The estimation of properties of the continuum phase at the particle location is easiest if shape functions are used. These require the knowledge of the property values at the element vertices rather than at the element centroid where they are stored. To speed up the tracking algorithm the necessary vertex values, velocity, density, viscosity and perhaps temperature, are calculated and stored. The value of a property, ϕ, at vertex v is estimated using

$$\phi_v = \frac{\sum_e \frac{\phi_e}{d_{ev}}}{\sum_e \frac{1}{d_{ev}}} \tag{4.3.1}$$

where the summation is over all elements, e, which have v as one of their vertices, ϕ_e is the value of ϕ at the centroid of element e and d_{ev} is the distance between vertex v and the centroid of element e. For the velocity components any known fixed value boundary conditions are used to overwrite the estimated value at a boundary vertex.

The first step of tracking a particle is to determine which element the particle starts in. To begin with a rough estimate is made by finding the element centroid closest to the particle's starting position. To determine if the particle is actually in this element the value d_{max} is sought where

$$d_{max} = \max[\hat{\underline{n}}_f \,.\, (\underline{r}_f - \underline{r}_d)] \tag{4.3.2}$$

In the above equation the maximum is taken over all faces, f, of the element, $\hat{\underline{n}}_f$ is the unit outward normal to the face, $\underline{r}_f$ is any point on the face and $\underline{r}_d$ is the position of the particle. If the value of d_{max} is negative then the particle is in the element, otherwise the step is repeated for the element the other side of the face which gave the final value of d_{max}. This continues until an element is found which gives a negative value of d_{max}, such an element will exist if the particle starts in the solution domain.

The next step is to calculate the fluid properties at the particle location, which as previously stated, involve the use of shape functions. If the particle is located at position (s, t, u) in local coordinates then the property value at that position is calculated using

$$\phi(s,t,u) = \sum_j N_j(s,t,u)\,\phi_j \tag{4.3.3}$$

where the shape functions, $N_j(s, t, u)$, for a variety of commonly used element types are given in Appendix A. The only problem that remains is the calculation of the local coordinate position of the particle, (s, t, u), which corresponds to the actual position of the particle, (x, y, z). It is possible to exactly find the values of the local coordinates but this requires considerable computational effort so an alternative approach is taken. The closest vertex, of the element containing the particle, to the particle's location is taken as the base point for the estimation of the values of the local coordinates. The element is rotated into a position such that the closest vertex is associated with shape function $N_1(s, t, u)$. For a brick type it is assumed that the three directions s, t and u coincide with the 3 edges leaving the closest vertex. The choice of which edge corresponds to the s direction is not important. Once the choice

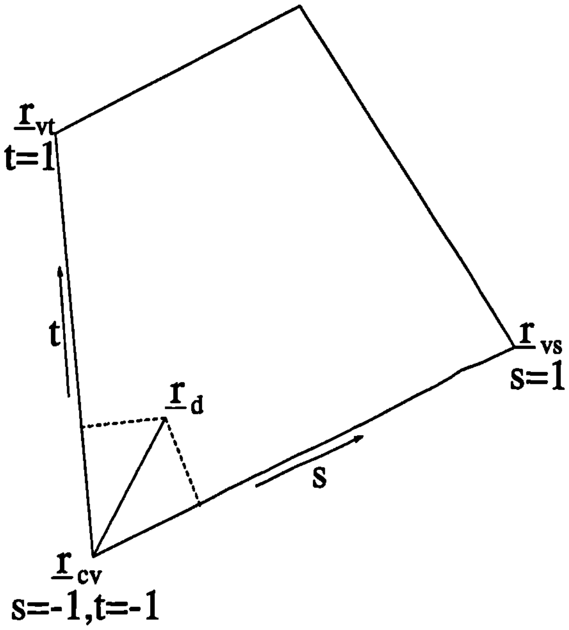

Figure 4.3.1: Estimation of Local Coordinate Values

has been made the value of s is calculated using

$$\begin{aligned} \alpha_s &= \frac{(\underline{r}_d - \underline{r}_{cv}) \cdot (\underline{r}_{vs} - \underline{r}_{cv})}{|\underline{r}_d - \underline{r}_{cv}|\,|\underline{r}_{vs} - \underline{r}_{cv}|} \\ s &= \max[\,-1.0,\ \min(1.0 - 2.0\alpha_s, 1.0)\,] \end{aligned} \tag{4.3.4}$$

where $\underline{r}_{cv}$ is the position of the closest vertex and $\underline{r}_{vs}$ is the vertex at the other end of the edge representing the s direction. The equations for t and u are exactly the same as equation (4.3.4) except $\underline{r}_{vs}$ is replaced with the position of the vertex at the end of the relevant edge. Equations of this type exist for the other element types for which the shape functions are given in Appendix A.

At this stage of the tracking algorithm any calculations required to ascertain if the particle is subject to fragmentation should be performed. This topic is discussed in section 4.3.1.

Having checked for break up, the next stage of the algorithm calculates the velocity of the particle. The drag coefficient for the particle is given by

$$C_{drag} = \left(\frac{24.0}{Re_d}\right)(1.0 + 0.15\,Re_d^{0.687}) + \frac{0.42}{1.0 + 42500.0\,Re_d^{-1.16}} \tag{4.3.5}$$

where the droplet Reynolds number, Re_d is calculated using the equation

$$Re_d = \frac{D_d \, \Delta V_{cd}}{\nu_{lam}} \tag{4.3.6}$$

The friction coefficient, F_c, is given by

$$F_c = \frac{0.75 \, C_{drag} \, \rho_c \, \Delta V_{cd}}{\rho_d \, D_d} \tag{4.3.7}$$

and the updated particle velocity is given by

$$\underline{V}_d^{new} = \left(\frac{1}{1.0 + F_c \, \delta t}\right) (\underline{V}_d^{old} + [F_c \underline{V}_c + \underline{g}] \, \delta t_d) \tag{4.3.8}$$

where $\underline{g}$ is the gravity vector and δt_d is the particle time step. The size of the time step is chosen so that a selected number of steps, n_e, are taken in each element. The formula used for the time step is

$$\delta t_d = \frac{2.0 \, d_e}{|V_d|^2 \, n_e} \tag{4.3.9}$$

where d_e is the distance from the element centroid to the closest point on any of the element's faces.

Once the new velocity of the particle has been calculated the new particle position can be evaluated. This is achieved using

$$\underline{r}_d^{new} = \underline{r}_d^{old} + \delta t_d \underline{V}_{move} \tag{4.3.10}$$

where

$$\underline{V}_{move} = 0.5(\underline{V}_d^{old} + \underline{V}_d^{new}) \tag{4.3.11}$$

The element containing the new particle position can now be calculated by calculating d_{max} as in equation (4.3.2) except that only faces for which

$$\underline{V}_{move} \cdot \underline{\hat{n}}_f > 0 \tag{4.3.12}$$

are considered. If the particle has not left the element it was in at the start of the step then all sources due to the particle can be added into that element. Otherwise the face through which the particle left the element is found by finding the maximum value, over all faces of the element, of δt_1 where

$$\delta t_1 = \frac{(\underline{r}_d - \underline{r}_f)\cdot\hat{\underline{n}}_f}{\underline{V}_{move}\cdot\hat{\underline{n}}_f} \tag{4.3.13}$$

and both numerator and denominator in equation (4.3.13) must be positive. The time spent in this starting element is δt_1. The search now moves into the element the other side of face. If the particle does not finish its movement in this element the time the particle spent in this element is calculated. A value for the time at which the particle left this element, δt_2, is calculated in the same way as for the original element using equation (4.3.13). The time spent in this element is then given by $\delta t_2 - \delta t_1$. This continues until the element containing the particle is discovered with $\delta t_d - \delta t_{i-1}$ being the time the particle spent in this element if it is the i'th element visited on this particle movement step.

The above method, for estimating the time a particle spends in an element, assumes that the face through which the particle leaves an element is not part of a boundary. If the face belongs to a boundary then a decision has to be made about whether the particle will bounce back into the solution domain or be lost, as a consequence of which tracking of the droplet will cease. For many boundary types, such as inlets and outlets, the particle will be lost from the domain but at a wall bounce may occur. If e, the restitution coefficient for the wall, has a non-positive value, the particle is assumed to stick to the wall and tracking ceases. Otherwise the particle is positioned just inside the wall, the time step of the particle motion is adjusted to the time at which the particle hit the wall and the velocity of the particle is set to

$$\underline{V}_d = \underline{V}_d - (1.0 + e)(\underline{V}_d\cdot\hat{\underline{n}}_f)\hat{\underline{n}}_f \tag{4.3.14}$$

The tracking algorithm now returns to the stage at which the fluid properties, at the particle location, are estimated. The tracking continues until either the particle leaves the domain or the time the particle has been tracked reaches a time limit.

§4.3.1 Droplet Breakup

In the case of liquid droplets, once the fluid properties at the particle location have been estimated checks can be made to determine if the particle is subject to break up. The simplest model uses a critical Weber number, We_c, to determine a stable particle diameter, D_s

$$D_s = \frac{\sigma_d We_c}{\rho_c (\underline{V}_c - \underline{V}_d)^2} \tag{4.3.15}$$

where ρ_c is the density of the continuum phase at the particle location, $\underline{V}_c$ and $\underline{V}_d$ are the velocities of the continuum and particle respectively and σ_d is the surface tension of the particle. If the current diameter of the particle, D_d is greater than the stable diameter the particle break up occurs and the new diameter is calculated using

$$D_d = K\, D_d \tag{4.3.16}$$

where the constant K is chosen to fit experimental data.

In reality the processes of break up are much more complex than those presented above. Kolev(102) presents an excellent review of the various causes of particle fragmentation covering a number of possible situations. One of the main points raised in his paper is that when a particle enters a region in which it is unstable it does not immediately suffer break up. The instabilities build in the particle for some time prior to fragmentation taking place. Putting together several equations taken from(102) the following algorithm introduces the time delay factor into the break up model. Initially the time counter since a break up, δt_s, is set to zero. The first step of the algorithm calculates the particle Reynolds number, Re_d

$$Re_d = \frac{D_d \Delta V_{cd}}{\nu_c} \tag{4.3.17}$$

where

$$\Delta V_{cd} = |\underline{V}_c - \underline{V}_d| \tag{4.3.18}$$

If Re_d is less than 200.0 the particle is in a region where it is stable. Break up can not occur and the value of δt_s is reset to zero. Otherwise the critical time is calculated using the equations

$$\begin{aligned} We_{cd} &= \frac{\rho_c \Delta V_{cd}^2 D_d}{\sigma_d} \\ \delta t_{crit} &= \sqrt{\frac{\rho_d}{\rho_c} \frac{2.3 \, We_{cd}^{0.25} \, D_d}{|\underline{V}_d|}} \end{aligned} \tag{4.3.19}$$

If the time since the last break up is greater than the critical time then fragmentation may occur. The value of the stable Weber number, We_{dx} is calculated using

$$\begin{aligned} We_{dx} &= 5.28 & Re_d &> 2000.0 \\ We_{dx} &= 36.0 \left(\frac{24.0}{Re_d} + \frac{20.1807}{Re_d^{0.615}} - \frac{16.0}{Re_d^{0.75}} \right) & 2000.0 > Re_d &> 200.0 \end{aligned} \tag{4.3.20}$$

The stable particle diameter is given by

$$D_{dx} = \frac{We_{dx} \sigma_d}{\rho_c \Delta V_{cd}} \tag{4.3.21}$$

If the current particle diameter is greater than this stable value fragmentation occurs. The number of extra particles created because of the break up is given by

$$n_s = \mathrm{INT} \left[\left(\frac{D_d}{D_{dx}} \right)^3 \right] \tag{4.3.22}$$

and the new particle diameter is

$$D_d = \sqrt[3]{\frac{D_d^3}{n_s + 1}} \tag{4.3.23}$$

CHAPTER FIVE

Validation Test Cases

The aim of this chapter is to present a set of test cases which show the capabilities of CWNN for simple heat transfer and turbulent fluid flow problems. The first of these cases involves one dimensional heat transfer. This case is used to test the mesh skewness correction terms and show that for a case with a linear analytic solution the correction terms produce an exact answer. Laminar fluid flow, through the use of the moving lid cavity, is tested in the next case. Again the correction terms are used to show they correct errors caused by mesh skewness in a non-linear problem. Combined fluid flow and heat transfer, along with buoyancy effects, are tested through the simulation of natural convection in a cavity. The standard k-ϵ turbulence model is validated by the simulation of the backward facing step. The results from the standard k-ϵ model are compared against those obtained using the ReNormalisation Group k-ϵ model.

§5.1 Heat Transfer along a Bar

Probably the simplest problem involves one dimensional, steady state heat transfer along a bar. The equation being solved is

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) = 0.0 \tag{5.1.1}$$

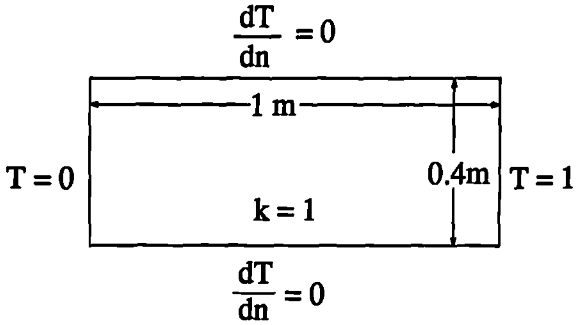

Figure 5.1.1: Set Up for Heat Transfer along a Bar

which, assuming that the thermal conductivity is constant, gives the solution

$$T = C_1 x + C_2 \tag{5.1.2}$$

For the test case presented here the problem solved and the outer dimensions of the mesh used are indicated in Figure 5.1.1 For this set of boundary conditions the solution is

$$T = x \tag{5.1.3}$$

§5.1.1 Cartesian Mesh

The first mesh employed in the solution of the problem used a Cartesian mesh with 20 elements in the 1m along the bar and 8 elements in the 0.4 m across the bar. Figure 5.1.2 shows a comparison of the analytic answer and the numerical results. In this case there is exact agreement between the two sets of data.

§5.1.2 Triangular Mesh, No Corrections

A triangular mesh was created from the mesh used for the Cartesian simulation by dividing each element into 2 triangles. The division of each element was achieved with a line from the top left corner to the lower right corner of the element. This

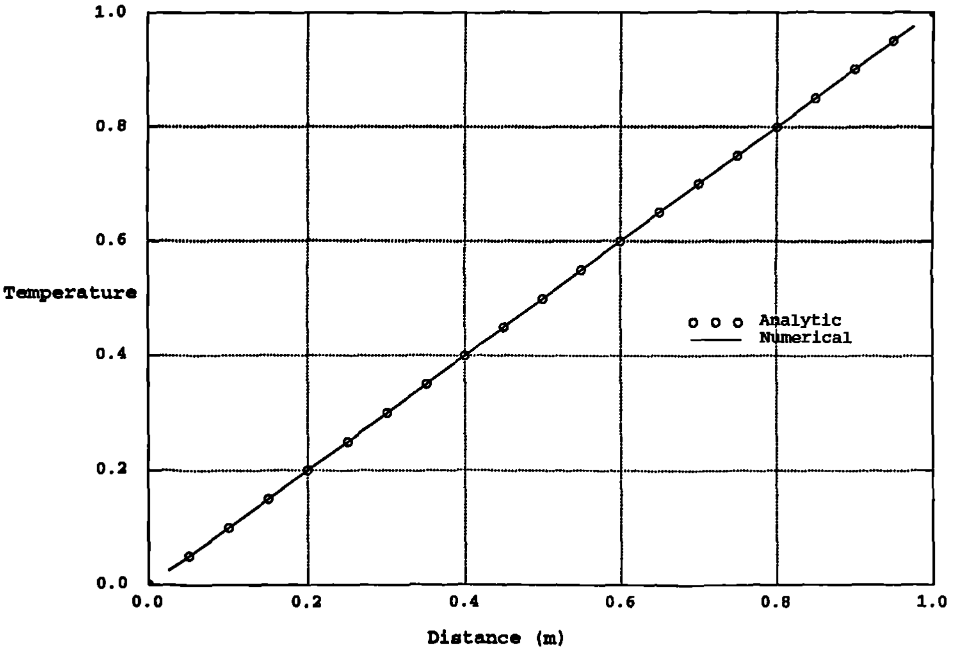

Figure 5.1.2: Heated Bar, Cartesian Mesh Results

creates a mesh which is fully conjunctional, except at the boundaries, but some of the connections are non-orthogonal. Figure 5.1.3 shows that the connection between triangles which make up the original squares is orthogonal, triangles P and B, but between triangles which where originally part of different squares, triangles P and A, the connection is non-orthogonal. A simulation using this mesh, without the use of terms to address the non-orthogonality of the mesh, leads to the results of Figure 5.1.4. The simulation results of this case, for which the analytic answer is only a function of x distance, show some dependence on the y direction. This is because of the error in the discretisation of the diffusion term caused by the absence of non-orthogonality correction terms. The results show a higher than expected thermal

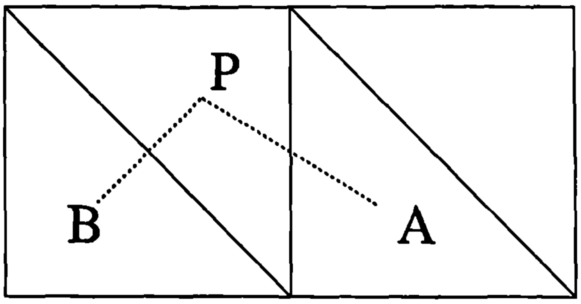

Figure 5.1.3: Triangular Mesh

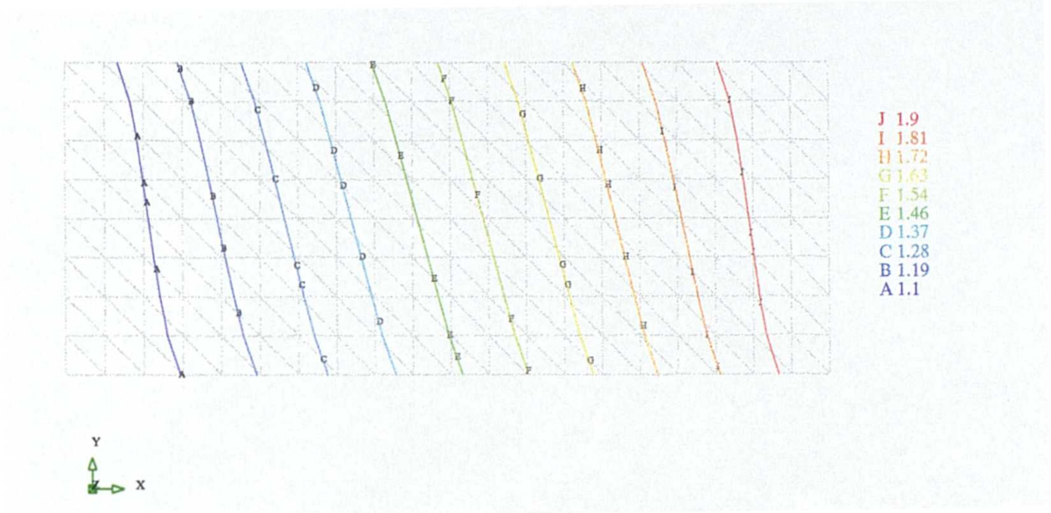

Figure 5.1.4: Heated Bar, Triangular Mesh Results, No Non-Orthogonality Terms

gradients in the upper left and lower right corners of the solution domain.

§5.1.3 Triangular Mesh, Non-Orthogonality Corrections

The inclusion of the non-orthogonality correction terms, defined in section 2.4, leads to the solution displayed in Figure 5.1.5. With the exception of near the boundaries, where the extrapolation of element values to nodal values has given errors, these results exhibit no dependence on the y distance. A plot of the temperature values from this simulation, along any horizontal line of nodes internal to the solution domain, will lead to a plot identical to Figure 5.1.2. This shows that the non-orthogonality terms do offer first order corrections to the problems caused by this form of mesh skewness.

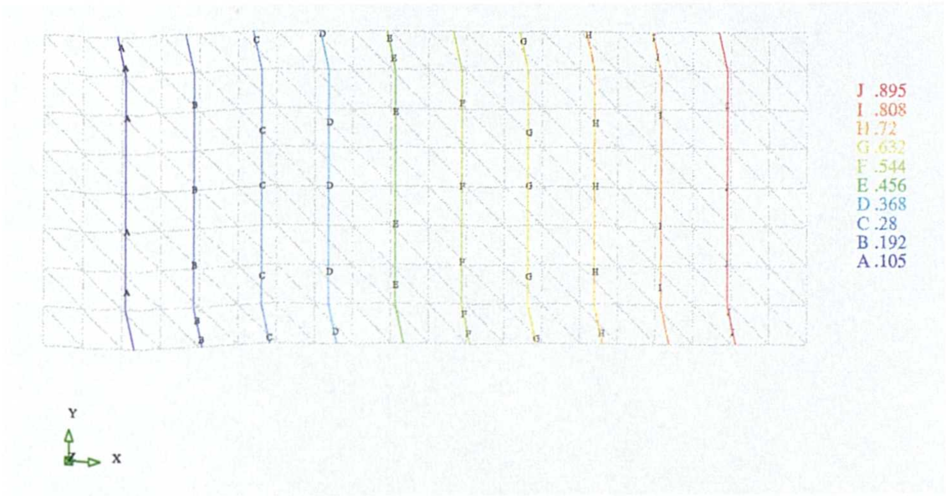

Figure 5.1.5: Heated Bar, Triangular Mesh Results, Non-Orthogonality Terms

§5.1.4 Triangular Mesh, Conjunctionality Corrections

A triangular mesh, similar to that used in the examples showing the benefits the orthogonality corrections, was constructed from a 10 by 1 Cartesian mesh. The outer dimensions of the mesh were 10 metres by 0.1 metres. The resulting mesh requires orthogonality corrections for all the internal connections. These corrections require the calculation of the gradients of any solved variable. The calculation of these gradients require an estimate of the value of the variable on each face, equations (2.4.6) and (2.4.7). The internal faces are fully conjunctional but the connections to the lower and upper boundaries are not, assuming that the estimate of the face value is made at the closest point to the element centroid. A pair of simulations were performed in which the boundary conditions were those shown in Figure 5.1.1. The results of these simulations, one with and one without conjunctionality corrections, are shown in Figure 5.1.6 When the correction terms are not used the simulation result progresses in a step like fashion. The values in triangles which were originially part of the same square are very nearly the same. It can be seen that the result from

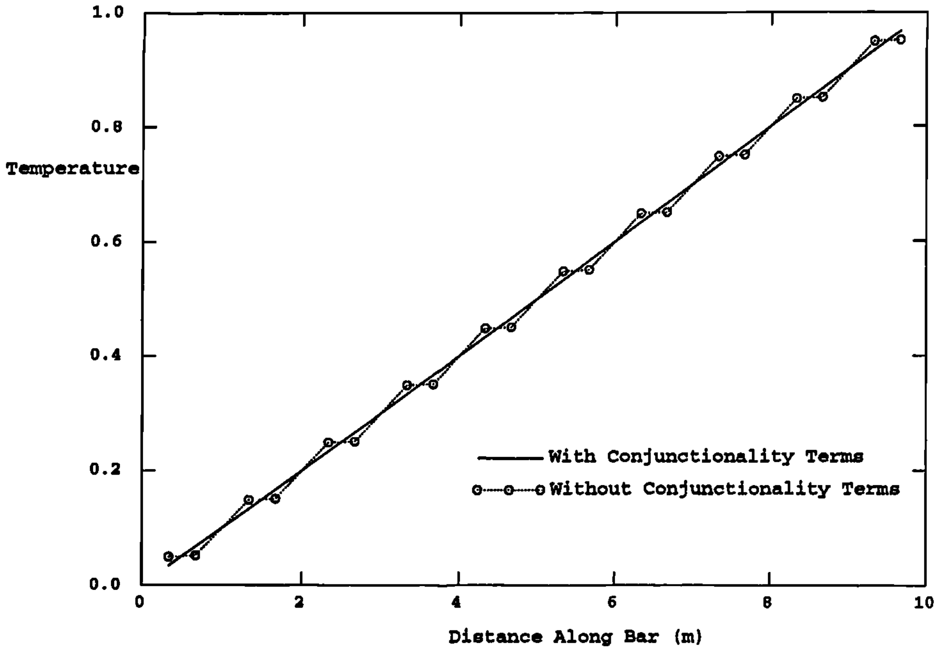

Figure 5.1.6: Heated Bar, Conjunctionality Corrections

the simulation using the correction terms matches the analytic, straight line, answer. Comparing the numerical results of the simulation for the three cases

- No correction Terms.
- Orthogonality Corrections Only.
- Both Orthogonality and Conjunctionality Correction.

the following table of results is obtained

Analytic Answer	No Corrections	Orthogonality	Both Corrections
3.33333E-02	3.98482E-02	4.87165E-02	3.32591E-02
6.66667E-02	5.99716E-02	5.10642E-02	6.65580E-02
1.33333E-01	1.39867E-01	1.48741E-01	1.33242E-01
1.66667E-01	1.59991E-01	1.51090E-01	1.66581E-01
2.33333E-01	2.39886E-01	2.48766E-01	2.33265E-01
2.66667E-01	2.60010E-01	2.51115E-01	2.66603E-01

The above table shows that the use of the orthogonality correction terms on their own causes a deterioration in the accuracy of the simulation results compared to those obtained with no corrections. To obtain a near analytic result both the conjunctionality and the orthogonality corrections have to be used.

§5.1.5 Multiple Materials

The final simulation, using only heat transfer, tests the ability to handle a mesh containing a number of materials. The problems consists of three 1m wide regions. The first and last regions have a thermal conductivity of 1 kg m sec^{-3} K^{-1} and the conductivity of the middle region is 0.1 kg m sec^{-3} K^{-1}. The boundary conditions place a fixed temperature of 0 K on the left side of the domain, a fixed temperature flux of 1 on the right side and adiabatic conditions on all other boundary faces.

Analytically solving this case, starting from equation (5.1.1), gives after one integration

$$k\frac{dT}{dx} = C \tag{5.1.4}$$

for some constant C, On the right side of the domain $k = 1$ and $dT/dn = 1$ which as $n = x$ at this point gives

$$k\frac{dT}{dx} = 1 \tag{5.1.5}$$

and using the values for k in the various regions

$$\frac{dT}{dx} = \begin{cases} 1.0 & 0 \leq x \leq 1 \\ 10.0 & 1 \leq x \leq 2 \\ 1.0 & 2 \leq x \leq 3 \end{cases} \tag{5.1.6}$$

solving this set of equations, using $T = 0$ at $x = 0$ and matching the temperatures

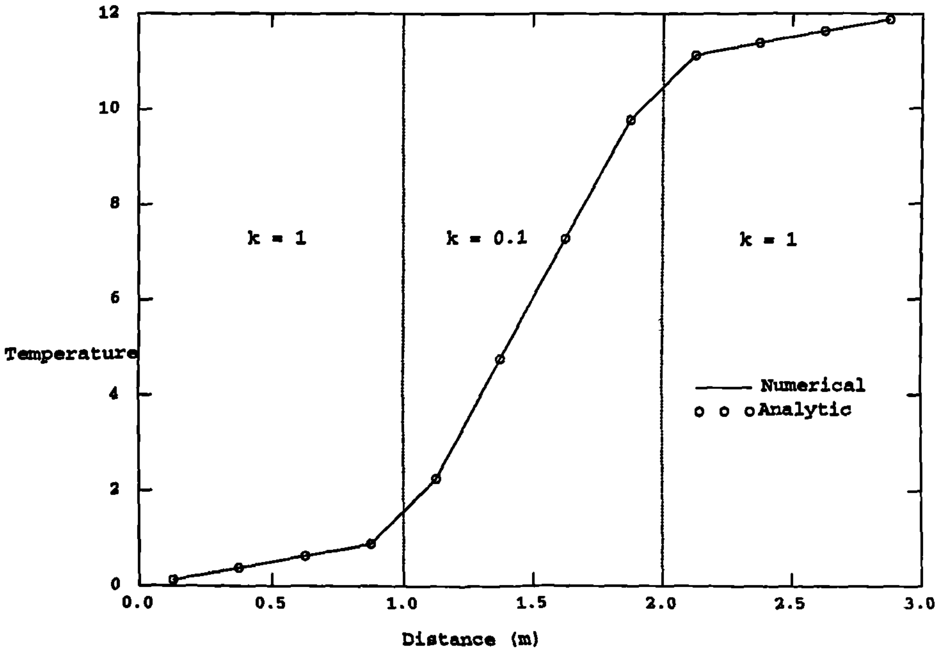

Figure 5.1.7: Results for Heat Transfer with Multiple Materials

at the material discontinuities, gives

$$T = \begin{cases} x & 0 \leq x \leq 1 \\ 10x - 9 & 1 \leq x \leq 2 \\ x + 11 & 2 \leq x \leq 3 \end{cases} \tag{5.1.7}$$

The results from the simulation of this case, using only 4 elements across each 1m material region, are shown with the analytic answer in Figure 5.1.7. The simulation results match the analytic answer exactly even on the very coarse mesh.

§5.2 Moving Lid Cavity

One of the standard test cases for the hydrodynamics section of a CFD code involves the simulation of a two or three dimensional box with a moving lid. This case involves flow in a domain bounded by walls all of which are stationary with the exception of one which has a non zero velocity component parallel to the domain. For this set of test cases the results are compared against those obtained by Ghia and Ghia[65] who produced a set of benchmark solutions, for a variety of Reynolds numbers, using the Multigrid technique. The simulations performed here employ a mesh which is much coarser than that used by Ghia and Ghia. For the Reynolds number equal to 100 this coarse mesh is sufficient to produce answers which compare favourably with the benchmark solutions.

§5.2.1 Two dimensions, Cartesian Mesh, $Re = 100$

For the first test case in this section a 1m square domain was used with 33 elements in both the x and y directions. The velocity placed on the high y boundary was 1 ms^{-1} in the x direction. To ensure a Reynolds number of 100 the density, ρ, was set to 1 kg m^{-3} and the kinematic laminar viscosity, ν_{lam}, was given the value 0.01 $\mathrm{m}^2\ \mathrm{sec}^{-1}$. In all the graphs of results presented in this section, normalised values are used. The normalised velocities parallel, u/U_{lid}, and perpendicular, v/U_{lid}, to the moving lid use the value of the lid velocity, U_{lid} to achieve the normalisation. Because the solution domain is a unit square there is no need to normalise distances across the domain. Figure 5.2.1 shows a graph of the velocity component parallel to the moving lid on a line in the y direction midway across the domain. For this Reynolds number there is excellent agreement between the two sources of results. Figure 5.2.2 compares the results for normalised velocity component perpendicular to the moving lid, v/U_{lid}, along a x line midway up the solution domain. Again there is excellent agreement between the two sets of results.

§5.2.2 Two dimensions, Triangular Mesh, $Re = 100$

The same case, as described in the previous section, was run using a mesh where each of the squares in the original mesh were divided into two triangles. The division of the squares into triangles was performed in the same way as for the mesh used in the

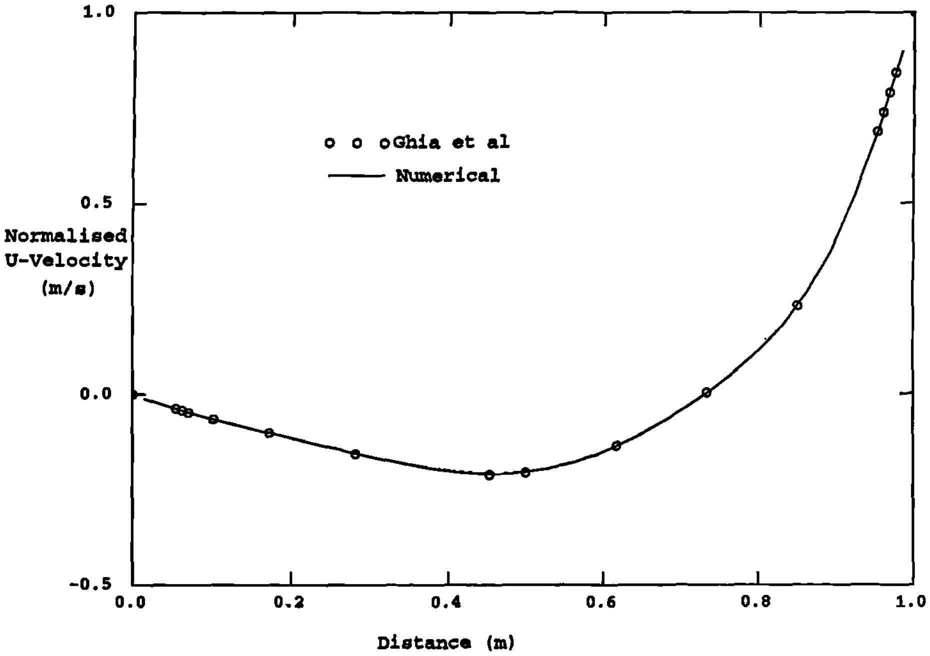

Figure 5.2.1: Moving Lid Cavity, U-Velocity, Cartesian Mesh, Re=100

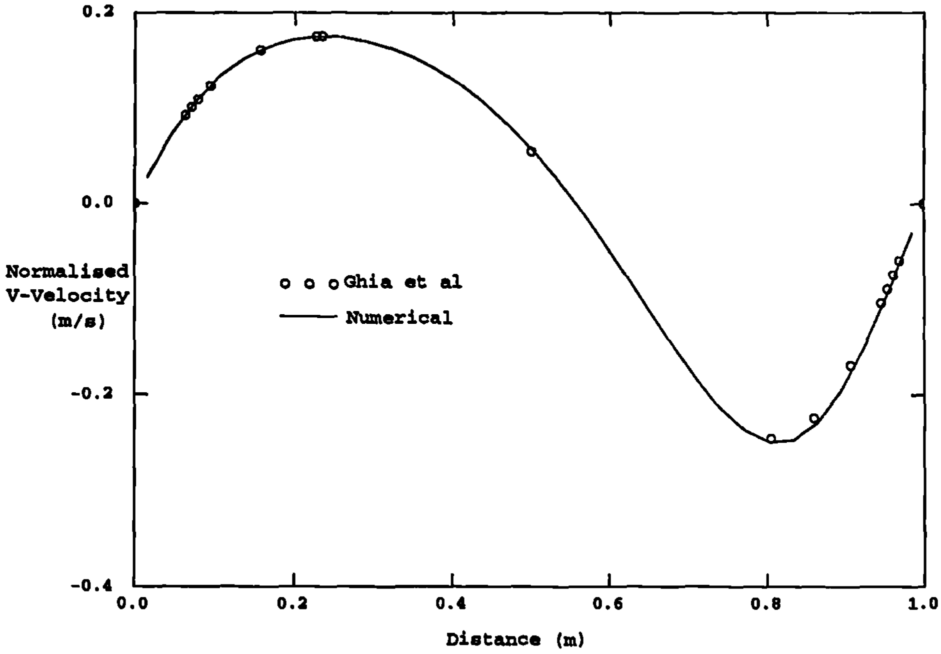

Figure 5.2.2: Moving Lid Cavity, V-Velocity, Cartesian Mesh, Re=100

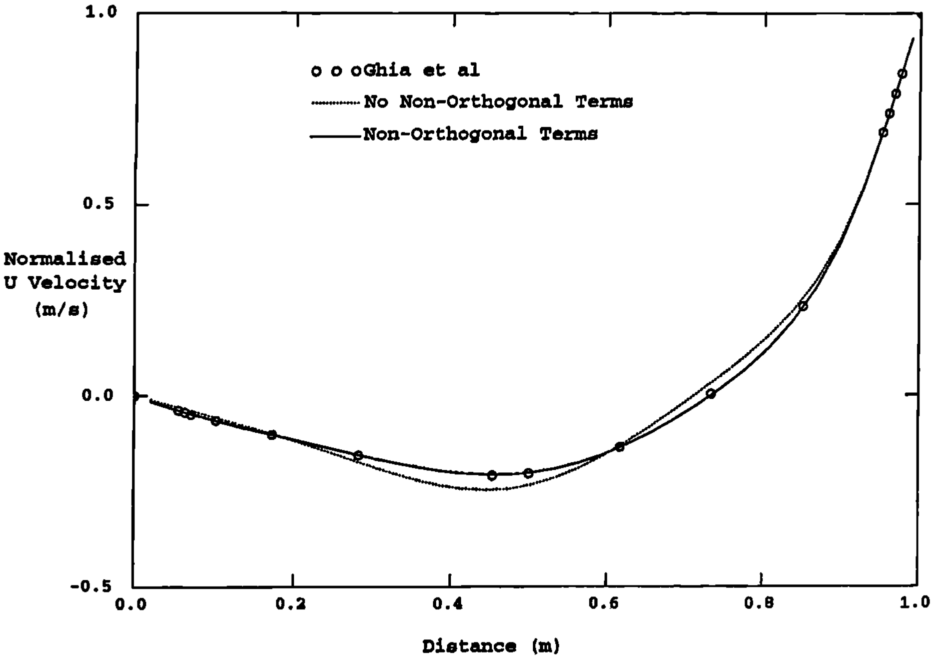

Figure 5.2.3: Moving Lid Cavity, U-Velocity, Triangle Mesh

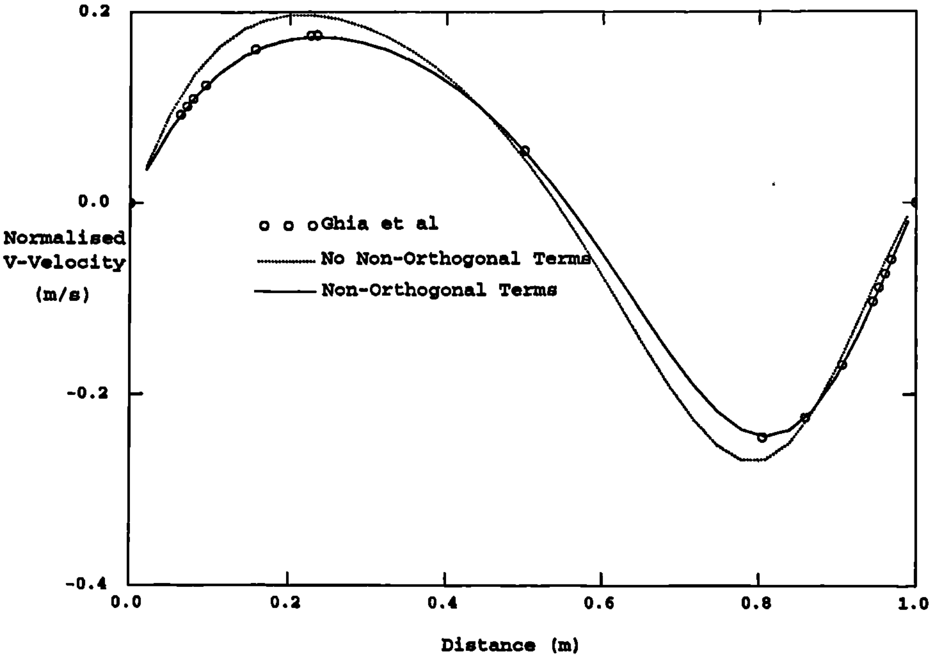

Figure 5.2.4: Moving Lid Cavity, V-Velocity, Triangle Mesh

heat transfer along a bar test case, section 5.1.2. The case was first simulated without the use of the mesh skewness correction terms. The results, Figures 5.2.3 and 5.2.4, show that when the corrections for the non-orthogonality of the mesh are not used there is a notable difference in the results of Ghia and those predicted by CWNN. In a subsequent simulation the correction terms were employed. It can be seen from Figures 5.2.3 and 5.2.4 that the numerical results now match the benchmark solutions as well as the results obtained using a Cartesian mesh.

§5.2.3 Two dimensions, Cartesian Mesh, Re = 400

A case, using the same Cartesian mesh as in section 5.2.1, was run with the moving wall velocity set to 4 ms^{-1}, giving a Reynolds number of 400. For this case the graphs of normalised u and v velocities are shown in Figures 5.2.5 and 5.2.6. It can be seen that for this case the numerical results produced by CWNN do not exactly match those predicted by Ghia *et al.* There are two reasons for the lack of accuracy in the simulation of this case. The first is the relatively coarse mesh used in the simulation. In the simulations performed by Ghia *et al.* the mesh consisted of 129 elements in both the x and y directions (Figures 5.2.5 and 5.2.6 also show the results obtained using a 99 by 99 Cartesian mesh which match the Ghia results extremely well). Secondly the diffusive nature of the first order differencing schemes causes an underestimation of the maximum values. The effects of higher order schemes on this case were examined in section 3.5.

§5.3 Natural Convection Cavity

This problem, involving buoyancy driven flow in a cavity, was first suggested as a suitable validation test case for CFD codes by Jones[103]. The boundary condition for the problem involved two vertical walls which differ in temperature leading to a thermal gradient across the solution domain. This thermal gradient leads to the

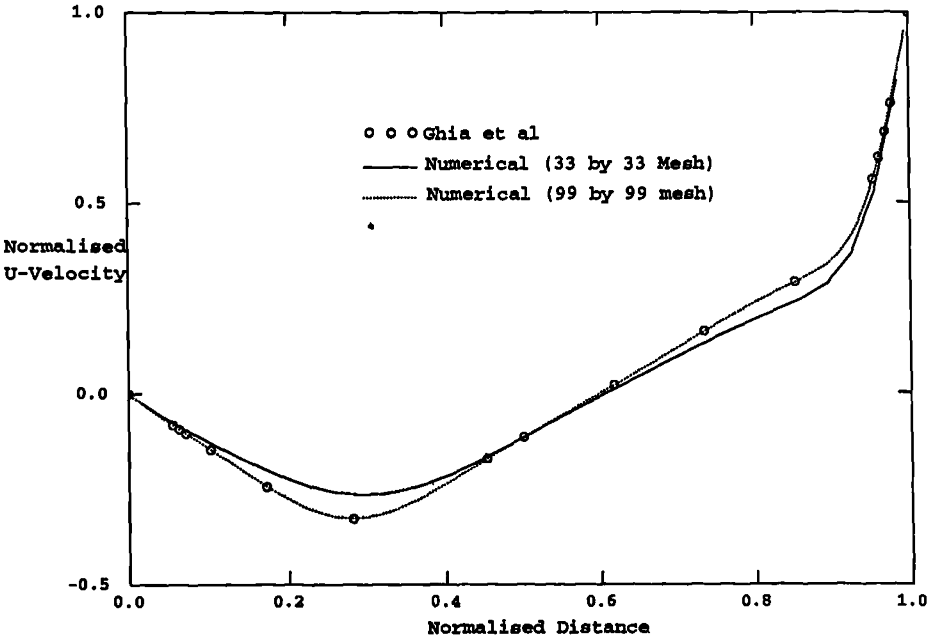

Figure 5.2.5: Moving Lid Cavity, U-Velocity, Cartesian Mesh, Re=400

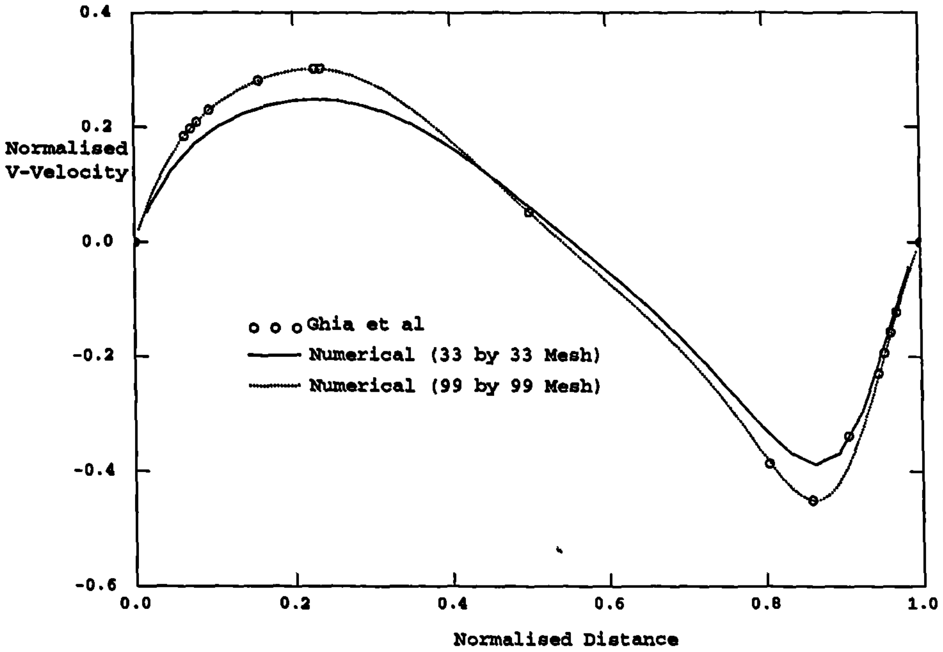

Figure 5.2.6: Moving Lid Cavity, V-Velocity, Cartesian Mesh, Re=400

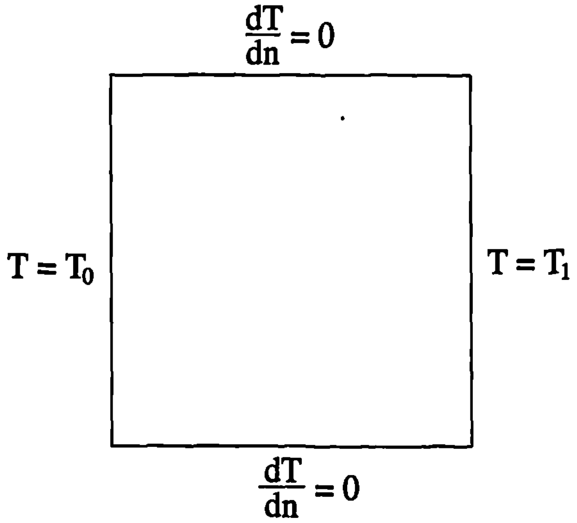

Figure 5.3.1: Boundary Conditions For Natural Convection Problem

buoyancy forces varying between the walls giving rise to flow. Adiabatic conditions are assumed on the remain boundaries, Figure 5.3.1.

It is assumed that the fluid is incompressible with the buoyancy forces being calculated using the Boussinesq approximation. Using this approximation the buoyancy source per unit volume into the i'th momentum equation takes the form

$$S_i = -\rho_{ref}\,\beta\, g_i\,(T - T_{ref}) \tag{5.3.1}$$

where ρ_{ref} is the reference density, which is equal to the constant density of the fluid, 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.

A set of benchmark solutions for the natural convection problem were published by de Vahl Davis and Jones(104) along with numerical results from thirty groups. These benchmark results are compared against the numerical results obtained from a set of simulations of the cases in CWNN. The normalised distances and velocities used in the results are calculated from

$$X = \frac{x}{L} \qquad U = \frac{u\,L}{k/(c_p\,\rho)}$$
$$Y = \frac{y}{L} \qquad V = \frac{v\,L}{k/(c_p\,\rho)}$$

Where L is the length of the sides of the cavity. The Rayleigh number, Ra, is defined as

$$Ra = \frac{\beta\,|g|\,\Delta t\,L^3}{\nu\,k/(c_p\,\rho)} \tag{5.3.2}$$

where Δt is the temperature difference between the two vertical walls, $|T_0 - T_1|$.

For the simulations presented here in this section the solution domain consisted of a 1m square employing a uniform Cartesian mesh with 33 elements in each direction. The properties for each run were approximately those for air at 300 K

Specific Heat	1.007E+03 $m^2\ sec^{-2}\ K^{-1}$
Thermal Conductivity	2.630E-02 kg m $sec^{-3}\ K^{-1}$
Density	1.161E+00 kg m^{-3}
Thermal Expansion	3.000E-03 K^{-1}
Laminar Kinematic Viscosity	1.590E-05 $m^2\ sec^{-1}$
Gravity	1.000E+01 m sec^{-2}

with gravity acting in the negative y direction. The temperatures on each of the vertical walls were varied to give the desired Rayleigh number with $T_1 > 0$ and T_0 = -T_1.

Plots of the velocity vectors and contour plots of temperature, u velocity and v velocity are shown in Figures 5.3.2 to 5.3.5 for Rayleigh numbers 10^3, 10^4, 10^5 and 10^6.

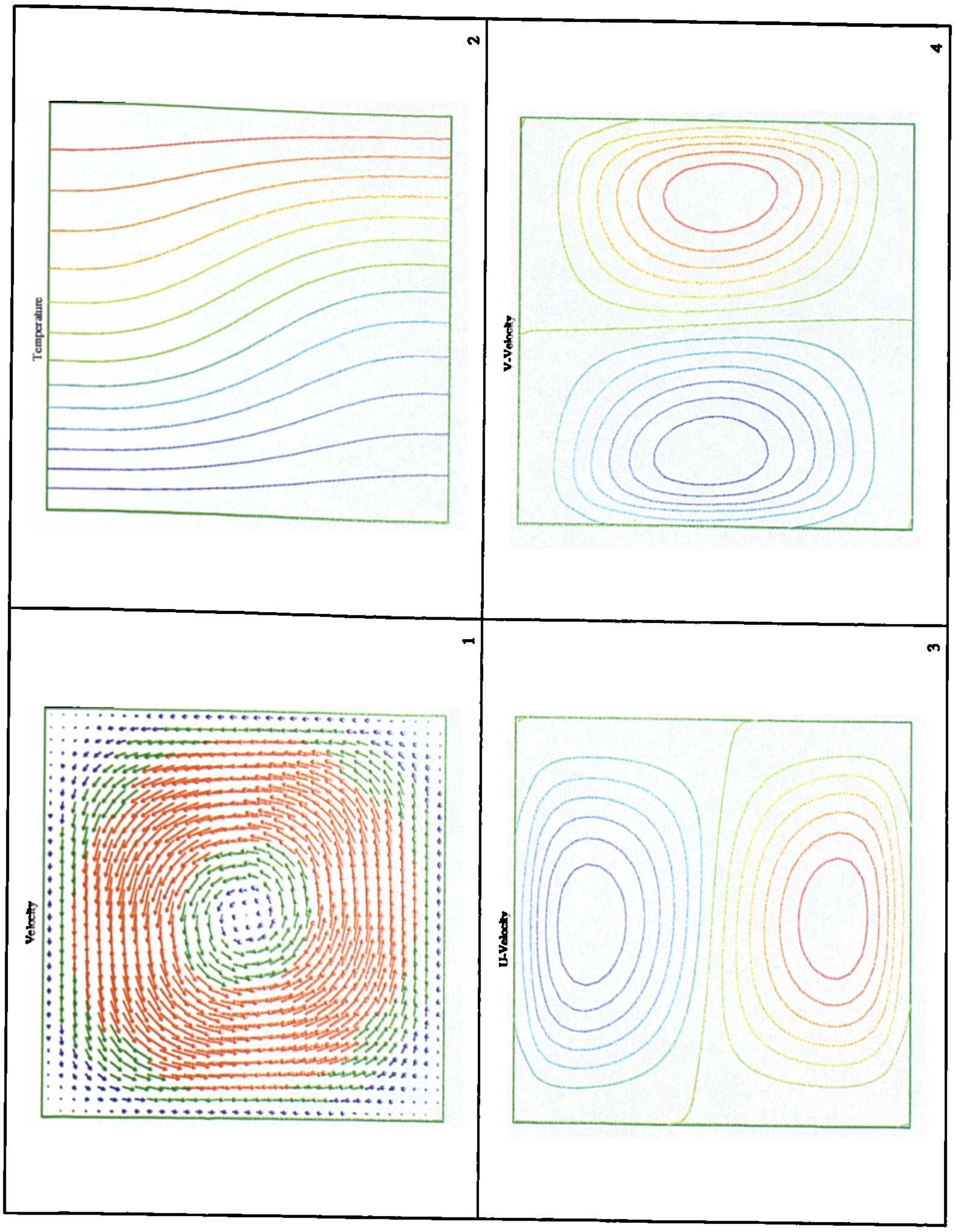

Figure 5.3.2: Natural Convection, Uniform Cartesian Mesh, Ra=10^3

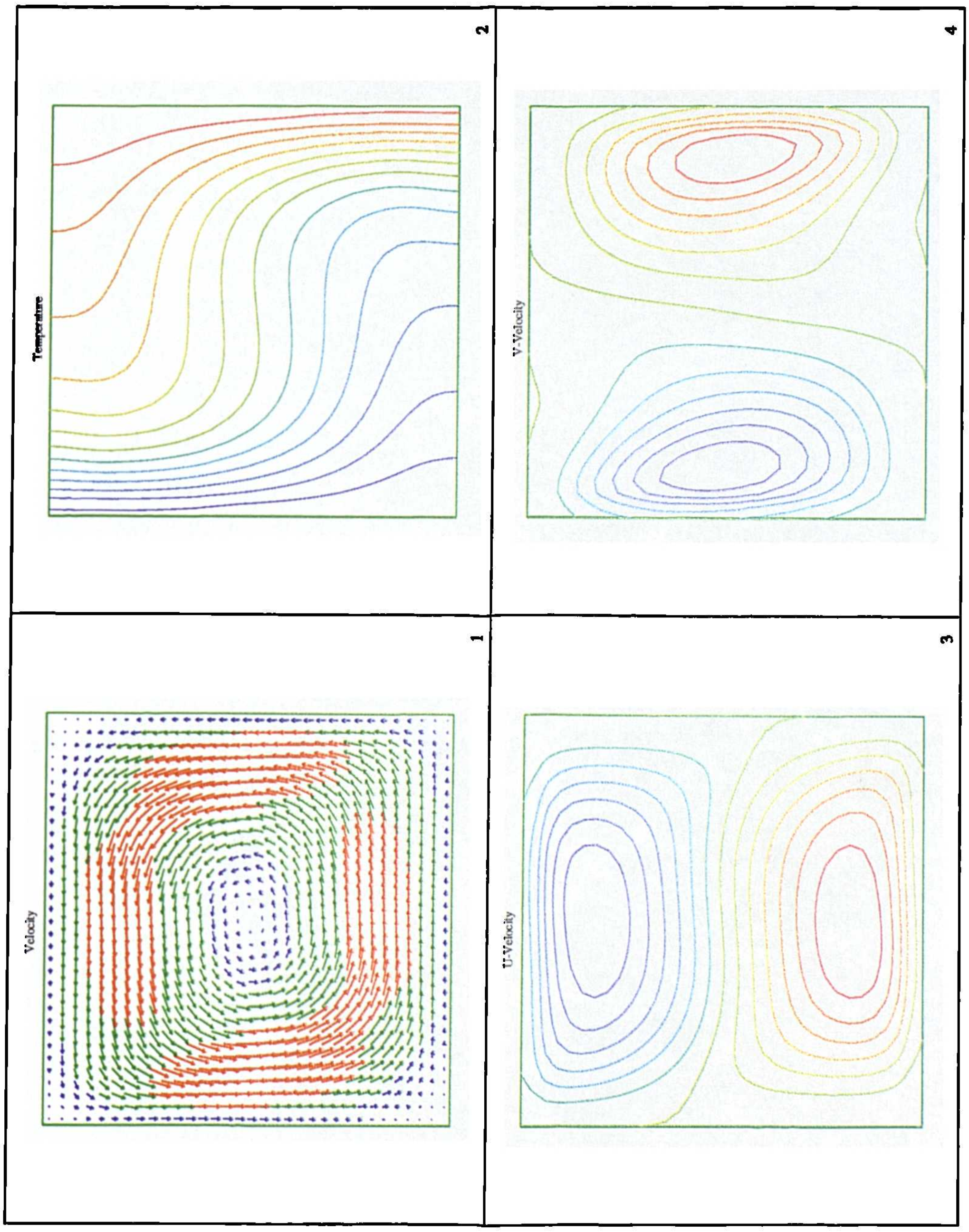

Figure 5.3.3: Natural Convection, Uniform Cartesian Mesh, Ra=10^4

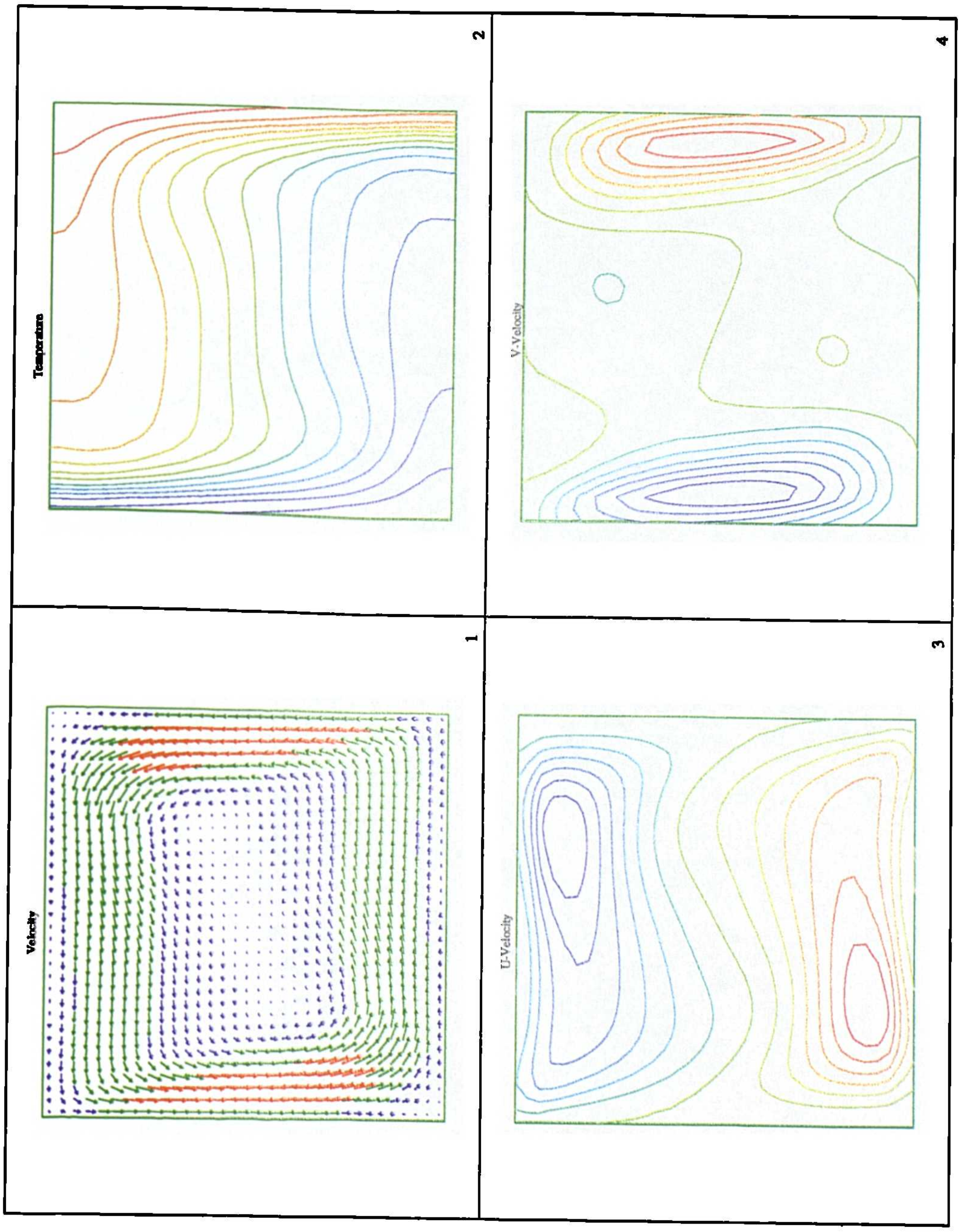

Figure 5.3.4: Natural Convection, Uniform Cartesian Mesh, Ra=10^5

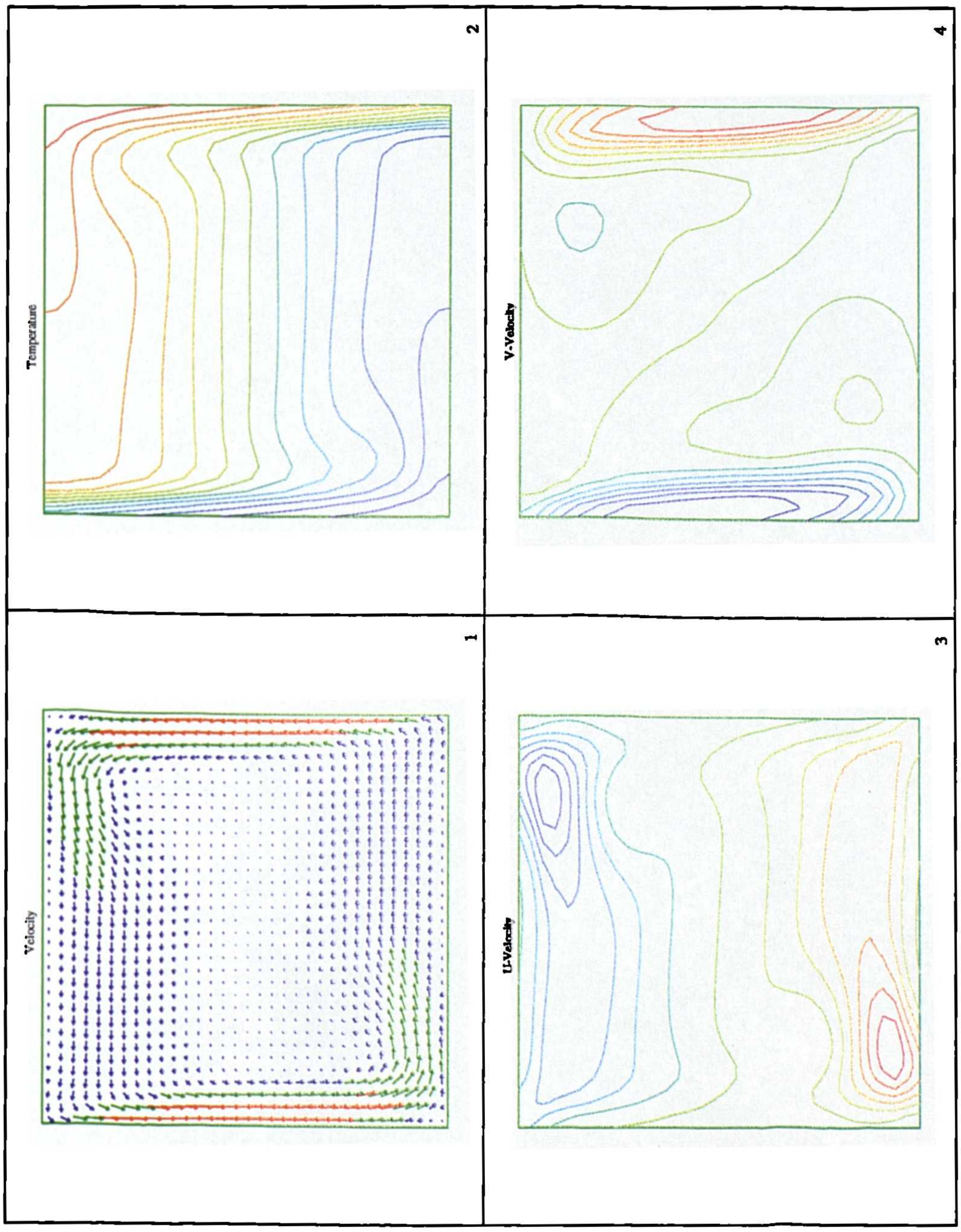

Figure 5.3.5: Natural Convection, Uniform Cartesian Mesh, Ra=10^6

Each simulation started with a zero velocity field and the domain temperature equal to the average of the hot and cold wall temperatures. The simulations were run until the L_2 norm of the change in the enthalpy value between iterations fell below 10^{-11} times the Rayleigh number. The following table shows a comparison of the outer iterations required for each of the simulations. All the simulations used linear relaxation of 0.6 for the pressure correction equation, this being the standard(47) when using the SIMPLE algorithm, with any additional relaxation factors being noted in the table below.

Rayleigh Number	Number of Iterations	Momentum False Time Step
10^3	568	None
10^4	468	None
10^5	2008	None
10^6	7974	1.0

The following table shows a comparison of the simulation results against the benchmark solution published by de Vahl Davis, which are enclosed in brackets. The U_{max} value is the maximum value of the normalised u velocity on the central vertical plane with Y_{max} being the normalised y position of this maximum. Similarly V_{max} and X_{max} are the maximum v velocity and its position on the central horizontal plane.

	10^3	10^4	10^5	10^6
U_{max}	3.644 (3.649)	16.123 (16.178)	34.82 (34.73)	65.485 (64.63)
Y_{max}	0.197 (0.187)	0.167 (0.177)	0.136 (0.145)	0.136 (0.15)
V_{max}	3.692 (3.696)	19.529 (19.617)	67.295 (68.59)	209.07 (219.36)
X_{max}	0.167 (0.178)	0.106 (0.119)	0.076 (0.066)	0.045 (0.0379)

The agreement with the benchmark solutions is reasonable throughout the range of Rayleigh numbers although there is some deterioration in accuracy as the Rayleigh

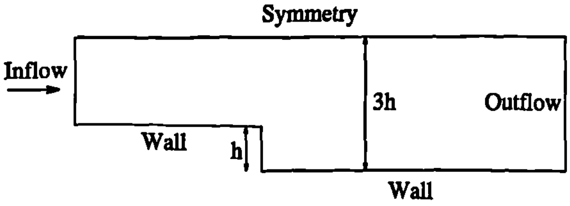

Figure 5.4.1: Backward Facing Step, Solution Domain

number increases. This deterioration is consistent with the findings of most contributors to the study of this problem[(104)]. In this study the maximum/minimum locations were element centre locations, none of the interpolation techniques suggested by de Vahl Davis were employed, so the accuracy of the predictions was constrained by the 33 by 33 mesh used in the simulations. If a mesh with the same number of elements, 81 by 81, as was used to obtain the benchmark solutions is employed, the figures for the 10^6 Rayleigh number test case are

	Uniform Mesh	de Vahl Davis
U_{max}	65.02	64.63
Y_{max}	0.154	0.15
V_{max}	217.47	219.36
X_{max}	0.031	0.0379

These results show considerable improvements on those obtained on the 33 by 33 mesh but to resolve the boundary layers a non-uniform mesh is necessary[(104)].

§5.4 Backward Facing Step

§5.4.1 Standard k - ϵ Model

Turbulent flow over a backward facing step is a standard test case used to compare and validate turbulence models. The case presented here[(105)], Figure 5.4.1, uses a domain in which the channel height is three times the height of the step, h. The inflow boundary is positioned five step heights upstream of the step to limit the effect of the assumptions made at the inflow boundary on the solution in the region after the step. At this boundary the turbulent quantities are estimated using the formulae

$$\begin{aligned} k_{IN} &= \frac{1}{4} 0.018\, U_{IN}^2 \\ \epsilon_{IN} &= \frac{0.1643\, k_{IN}^{1.5}}{0.09\, h} \end{aligned} \tag{5.4.1}$$

The inflow velocity, U_{IN}, is chosen to give a Reynolds number of 88,000. The outflow boundary is placed 20 step heights downstream of the step, again to to limit its influence on the simulation. The reattachment length, x_R, is a commonly used parameter to determine the ability of a turbulence model to correctly simulate the backward facing step. This parameter is the distance from the step to the position on the wall, at the bottom of the channel, at which the velocity along the channel becomes positive. The experimental data for this case(106, 107) indicate that the reattachment length, x_R, is equal to 7.0±0.5 times the step height. It has been shown by many authors, for example(108, 109), that the reattachment point predicted by the standard k-ϵ model lies in the range 5.8 - 6.1. Figure 5.4.2 shows a plot of the velocity component along the channel in the line of elements immediately above the channel wall. Each marker is separated by half the step height which gives a reattachment length of 6.0 which is well within the expected range of values for the standard k-ϵ model. The mesh used to obtain these results is fairly coarse using 40 elements along the channel between the step and the outflow boundary and 25 across the channel. For a much finer mesh consisting of just under 10,000 elements the prediction of the reattachment length was 5.93 step heights. This value is not significantly different from the value obtained using the 1,000 element mesh.

§5.4.2 RNG k-ϵ Model

The backward facing step problem, as described in the previous section was simulated using the RNG variation of the k-ϵ model. For this case the graph of the velocity component along the channel, in the line of elements immediately above

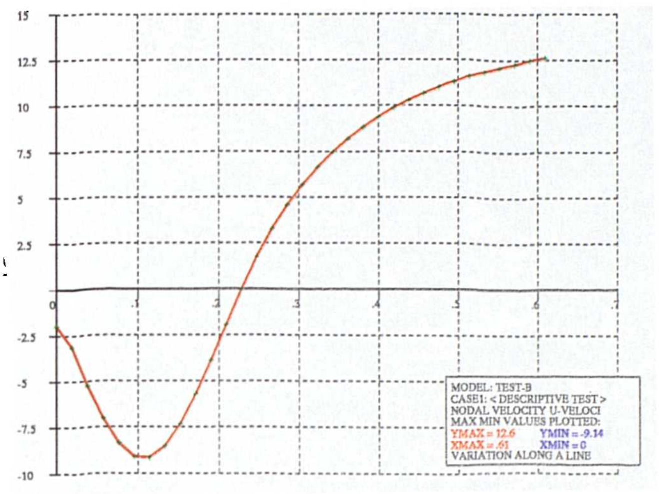

Figure 5.4.2: Backward Facing Step, k-ϵ model, Reattachment Length

the lower wall, is plotted in Figure 5.4.3. For the RNG model the prediction of the reattachment length is 6.14. This is slightly closer to the experimental results than the standard k-ϵ model but is still a 12.3% error, compared with 15.3%. This trend is consistent with all other turbulent comparison cases run by the author. In nearly all cases the RNG k-ϵ model produces a slightly better result than the standard k-ϵ model, but the errors in the RNG model are still of the same order as the errors in the results from the standard k-ϵ model. The claims that the RNG k-ϵ model offers a significant improvement on the standard k-ϵ model(110) have not been noted in the CWNN implementation of the model.

§5.5 Run Time Behaviour

Treating the mesh as unstructured means that extra computation effort is required at all stages of the procedure, for example $\underline{u} \cdot \underline{n}$ is equal to $\pm u$ in PHOENICS whereas in CWNN 3 multiplications and 2 additions are required in the calculation. These extra calculations lead to a run time which is approximately 4 to 4.5 times that of

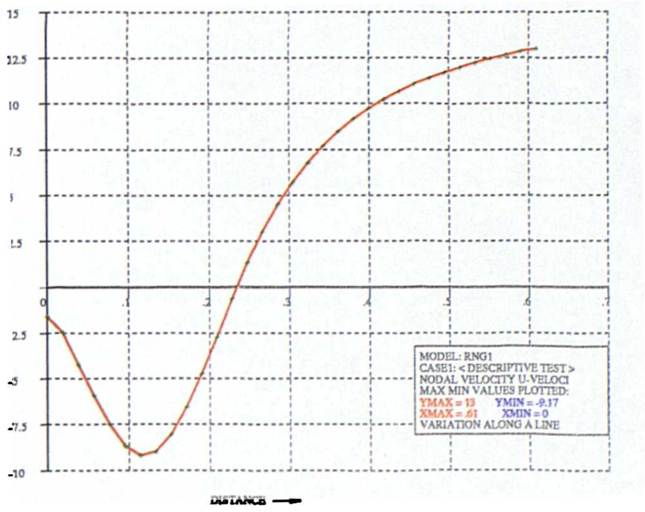

Figure 5.4.3: Backward Facing Step, RNG model, Reattachment Length

PHOENICS in fully Cartesian mode.

The number of iterations required to obtain convergence in CWNN is usually higher than the number required by PHOENICS although this is very dependent on the case being simulated and the level of convergence required. For example, in the case of parabolic flow along a duct PHOENICS requires around 50 iterations whilst CWNN requires 130 to converge to 1.0E-05. This difference can reduce significantly for elliptic problems. A simulation of the moving lid cavity, Reynolds number 100, requires 60 iterations in CWNN and 65 in PHOENICS.

CHAPTER

SIX

Simulations Relevant to the HIsmelt® Process

The models and algorithms described in the previous chapters, and incorporated into the CWNN code, were adapted and extended to the modelling of a real industrial problem. Collaboration with Rio Tinto Zinc in modelling iron production processes requires the development of models in CWNN which allow the simulation of the fluid flow and heat transfer processes occurring in the top space of the Smelt Reduction Vessel (SRV)(9) in the HIsmelt® Process.

In the HIsmelt® Process, described in Chapter 1, reactions in the top space generate energy which is fed back into the liquid metal bath to aid the reduction process. Single, or multiple, hot air jets enter the SRV carrying the oxygen required by these reactions. It was found that swirl has a significant influence on the efficiency of the process(10) and for this reason the correct characterisation for swirl becomes an important parameter in the model. Bottom injection processes in the metal bath cause a "fountain" of metal droplets to be ejected into the top space. These droplets influence the fluid flow, through friction effects, and the heat transfer, through radiative transfer of energy and chemical reactions on the droplet surface(111), of the continuum phase.

This Chapter presents some of the simulations performed in the development of a model of the top space of the HIsmelt® process. The simulation of swirling flows is tested in the first test case which involves an annular solution domain. Both laminar and turbulent flows, with a comparison of turbulence models, are simulated with the numerical results compared against both experimental data and numerical results from simulations performed using PHOENICS.

The combustion processes occurring in the top space release large amounts of energy which it is hoped will be transferred to the metal bath to provide the energy required to smelt the iron. Consequently, the reactions occurring in the top space must be modelled. This is effected through the use of the Simple Chemical Reaction Scheme(45) (SCRS). The second validation test case compares numerical results from both CWNN and PHOENICS for a case employing the SCRS.

The last section in this chapter shows a set of results from simulations relating to the HIsmelt® process. The first of these cases involves swirling, turbulent flow in an axi-symmetric simulation. For this simulation experimental data was available which allowed the various turbulence models to be compared. The second test involves the simulation of a three dimensional domain with two separate sources of swirl and chemical reactions taking place. In this case the general interpolation scheme is needed to evaluate the position of the jets and to calculate the convection fluxes. In the third case the complete top space model is used with an axi-symmetric mesh. This case includes the modelling of a droplet phase through Lagrangian tracking and the use of the radiosity and SCRS models.

§6.1 Swirling Flow in an Annulus

A series of papers by Morsi and Clayton(112–114) investigated, both experimentally and numerically, turbulent swirling flow along an annular pipe. In the experiments an annulus of inner radius 56mm was used with two settings for the outer radius which led to the ratio of inner to outer radii taking the values 0.51 and 0.61. The case considered here deals with the value of the ratio being equal to 0.51. A series of experiments were performed over a range of Reynolds numbers from 9,000 to 29,000. The swirl at the inlet being controlled by a set of guide vanes, the adjustment of which allowed the swirl number to be controlled. In all cases the tangential velocity component at the inlet had an approximate free vortex profile.

§6.1.1 Laminar Annular Flow

The initial simulations of the annulus were performed assuming laminar flow. For these simulations an inflow Reynolds number of 1,800 was used. The tangential component of velocity at the inflow boundary satisfied the equation

$$u_\theta = \frac{u_z r_o}{r} \tag{6.1.1}$$

where r_o is the outer radius of the annulus. This variation in the tangential velocity is approximately equivalent to a vane angle, ψ, of 45° in the Morsi experiments. Initially the case was run using an axi-symmetric mesh employing Cartesian interpolation of the velocity components. A comparison of the tangential component of velocity from the numerical simulations in CWNN and PHOENICS and the predictions presented by Morsi is shown in Figure 6.1.1. The results of CWNN seriously over predict the decay of the tangential component of velocity. This is almost completely due to the handling of the contributions from the third dimension of the mesh, the tangential direction. If the cylindrical interpolation scheme, introduced in section 3.3, is used instead of interpolation of the Cartesian velocity components then the prediction of the decay of the tangential velocity component is that shown in Figure 6.1.2 This shows a considerable improvement in the prediction of the decay of the tangential component along the annulus. The results in this case closely match those produced by PHOENICS although there is significant differences with those predicted by Morsi. Figures 6.1.3 and 6.1.4 show comparisons of the tangential and axial profiles predicted by CWNN and PHOENICS at distances 0.15m, 0.45m 0.95m and 1.95m along the annulus. Again there is good agreement between the two codes. Comparison of the profiles for the tangential component of velocity with the numerical simulations performed by Morsi show excellent agreement in terms of shape but the Morsi simulations do not predict as quick a decay of swirl.

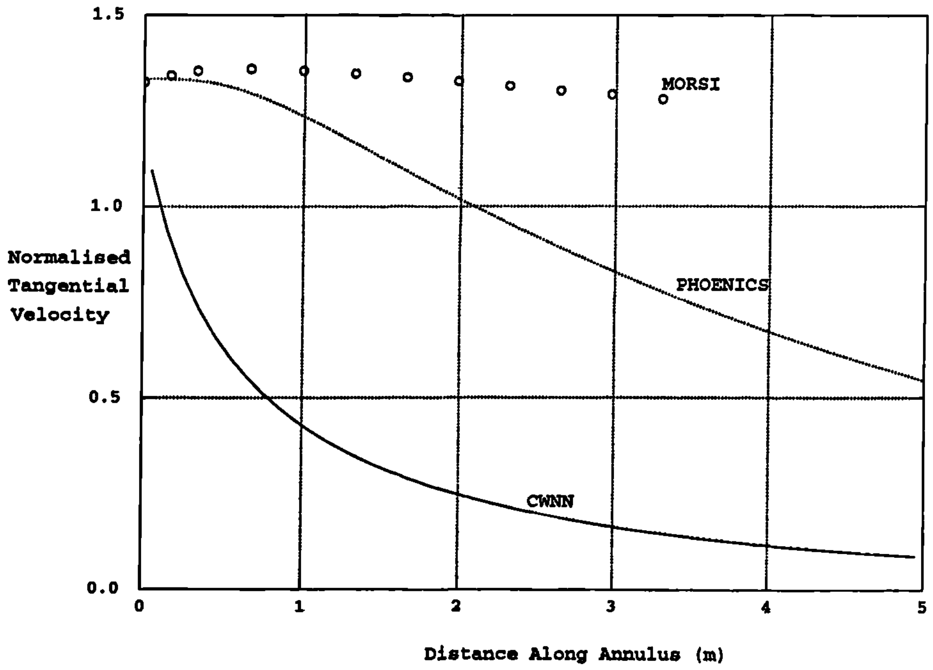

Figure 6.1.1: Laminar Flow in an Annulus, Cartesian Interpolation

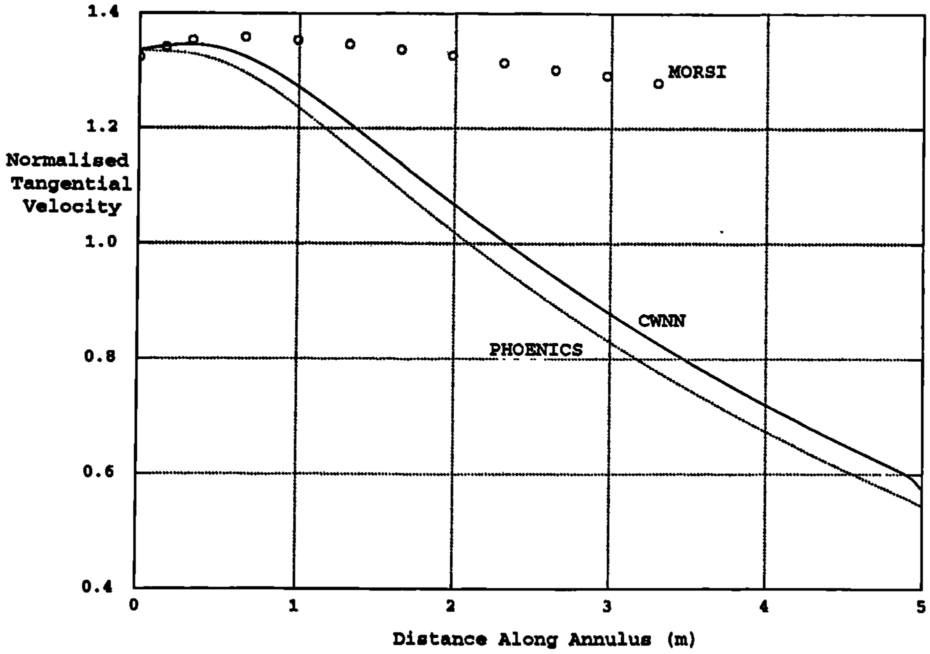

Figure 6.1.2: Laminar Flow in an Annulus, Cylindrical Interpolation

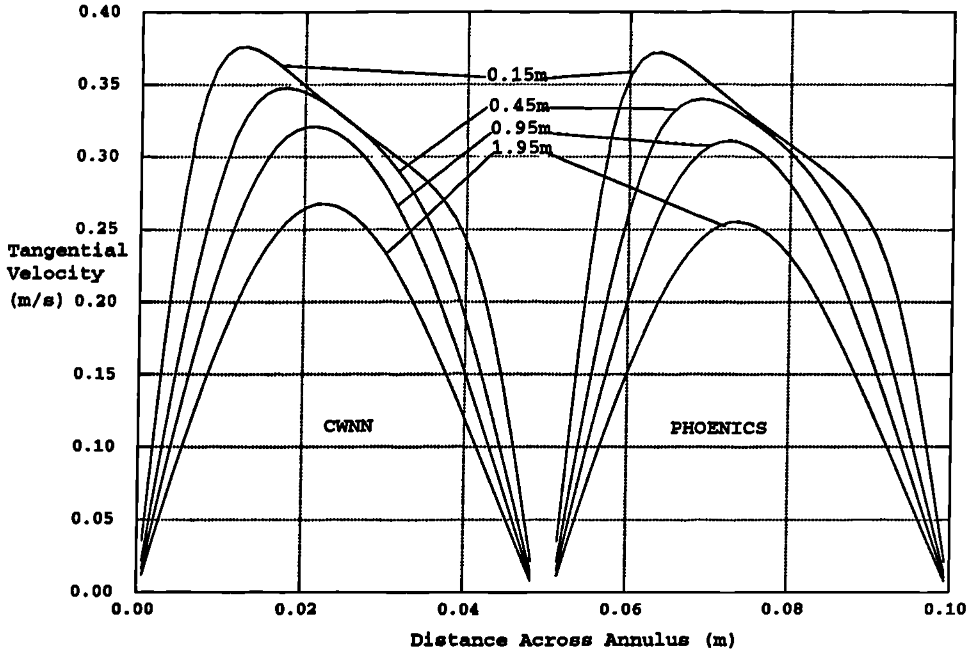

Figure 6.1.3: Laminar Flow in an Annulus, Tangential Velocity

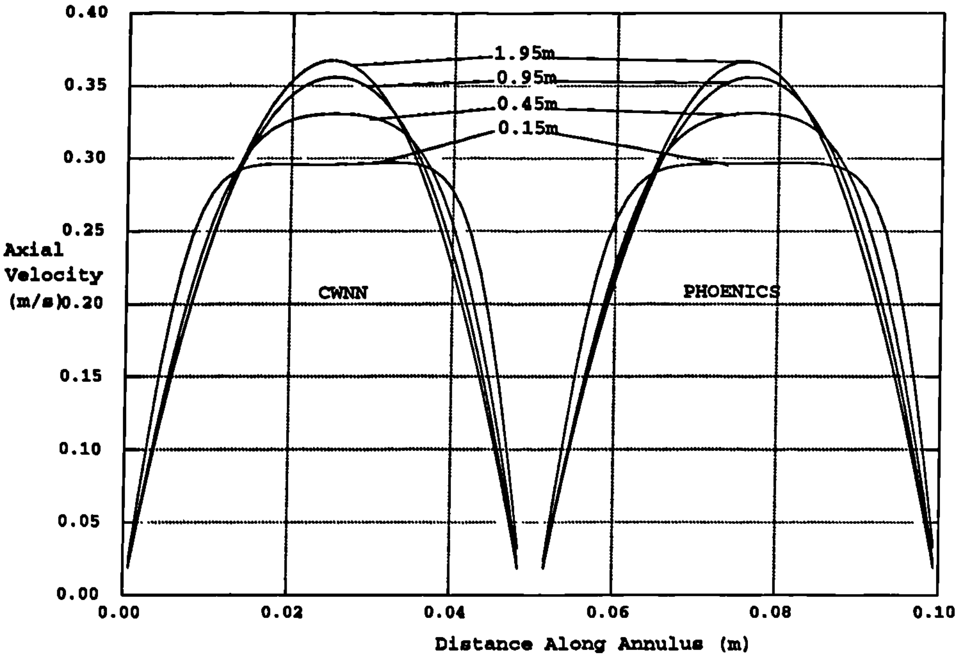

Figure 6.1.4: Laminar Flow in an Annulus, Axial Velocity

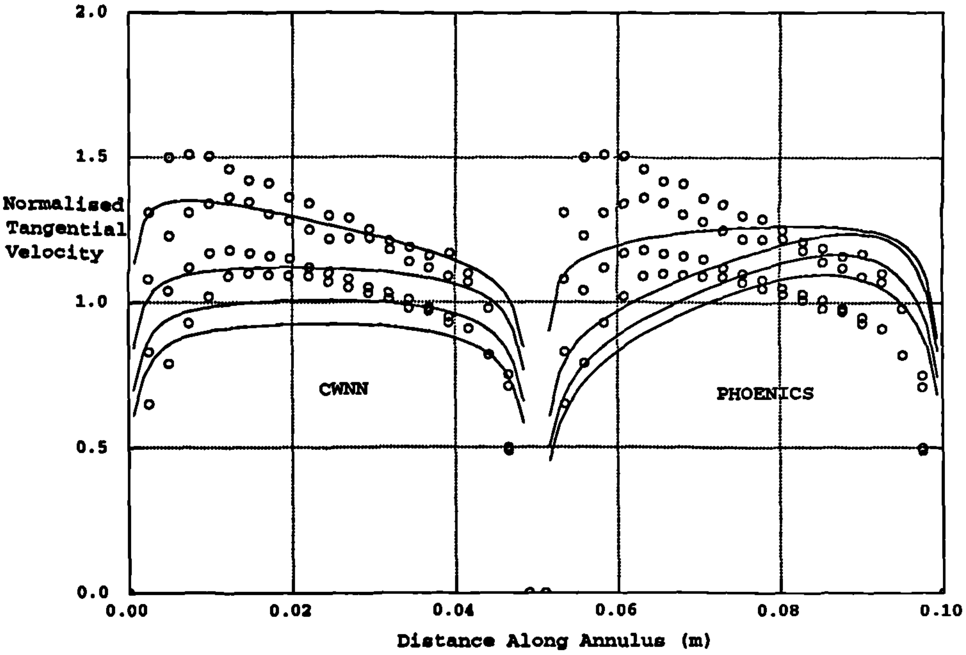

Figure 6.1.5: Turbulent Flow in an Annulus, k-ϵ Model Comparison

§6.1.2 Turbulent Annular Flow Test

For the simulation of turbulent flow along an annulus the inflow Reynolds number was set to 28,900. The tangential component of velocity at the inflow boundary was calculated from equation (6.1.1). The turbulent quantities at the inflow boundary were evaluated using the equations

$$\begin{aligned} k_{IN} &= 0.035\,U_{IN}^2 \\ \epsilon_{IN} &= \frac{0.1643\,k_{IN}^{1.5}}{0.09\,h} \end{aligned} \qquad (6.1.2)$$

Initially the case was simulated in CWNN and PHOENICS with the numerical results compared against the experimental data obtained by Morsi. Figure 6.1.5 shows a comparison of the predicted tangential velocities from both codes against the Morsi data at distances 0.17, 0.5, 0.83 and 1.16 metres along the annulus. It can be seen that the k-ϵ model does not predict the experimental profiles particularly well. In CWNN the decay of the swirl component of velocity is over predicted and the profile flattens much quicker than in the experimental data. There obviously is

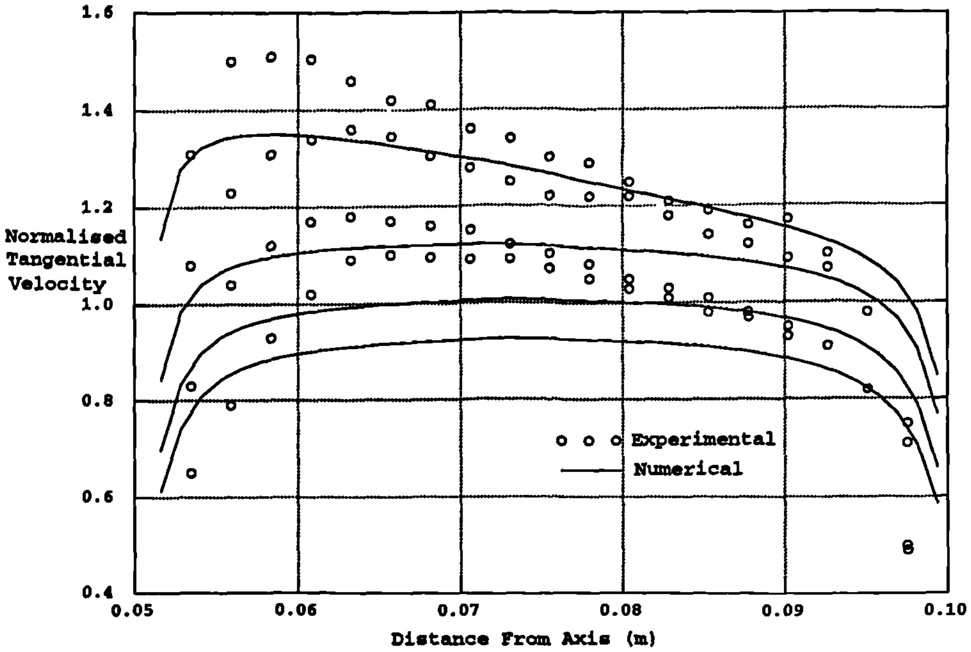

Figure 6.1.6: Turbulent Flow in an Annulus, Standard k-ϵ Model

a problem with the version of PHOENICS used to simulate this test case as the profiles are almost a mirror image of what they should be. This is not due to errant specification of the inflow boundary condition as both codes match in the first row of elements immediately after the inflow boundary. The above case was next used to compare the abilities of the standard k-ϵ model, two versions of the RNG k-ϵ model and the k-L mixing length model to predict swirling flows. The only difference between the versions of the RNG k-ϵ model was in the calculation of the turbulent viscosity. In the first version the equation described in the section on the RNG model, equation (4.2.30), was used for the effective viscosity whereas in the second version the equation employed in the standard k-ϵ model, equation (4.2.16), was used. Figures 6.1.6 to 6.1.9 show plots of the normalised tangential velocity, for each of the turbulence models, plotted with the experimental data. There is little difference in the results obtained from the standard k-ϵ model and either of the two versions of the RNG k-ϵ model. The second version of the RNG k-ϵ model produces the best results with the standard k-ϵ model giving the worst but this variation is small compared to their deviations from the experimental data. This is consistent with the results from the backward facing step and other test cases not presented

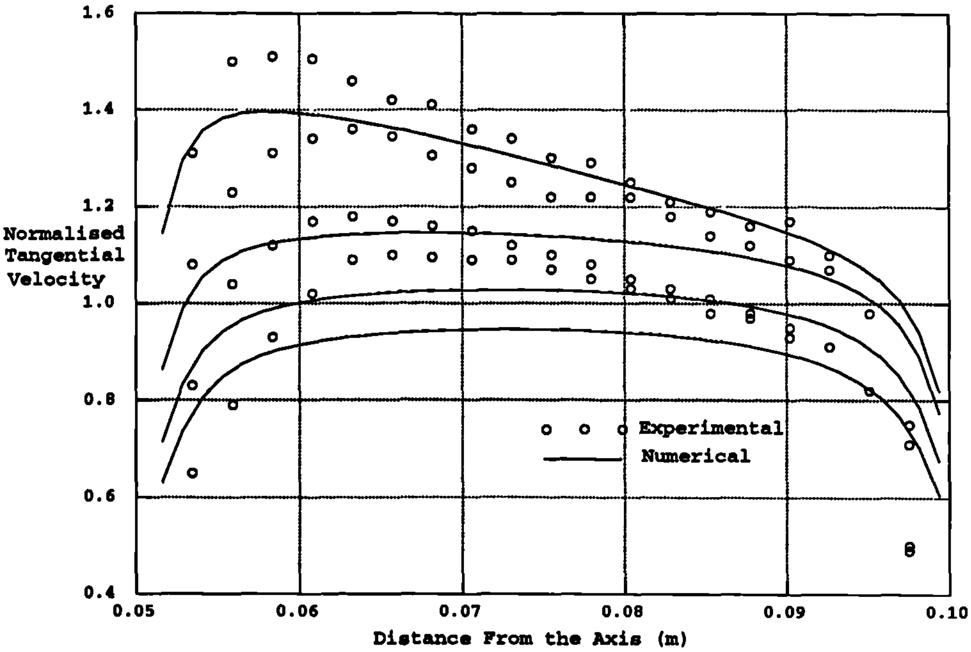

Figure 6.1.7: Turbulent Flow in an Annulus, RNG k-ϵ Model, Version 1

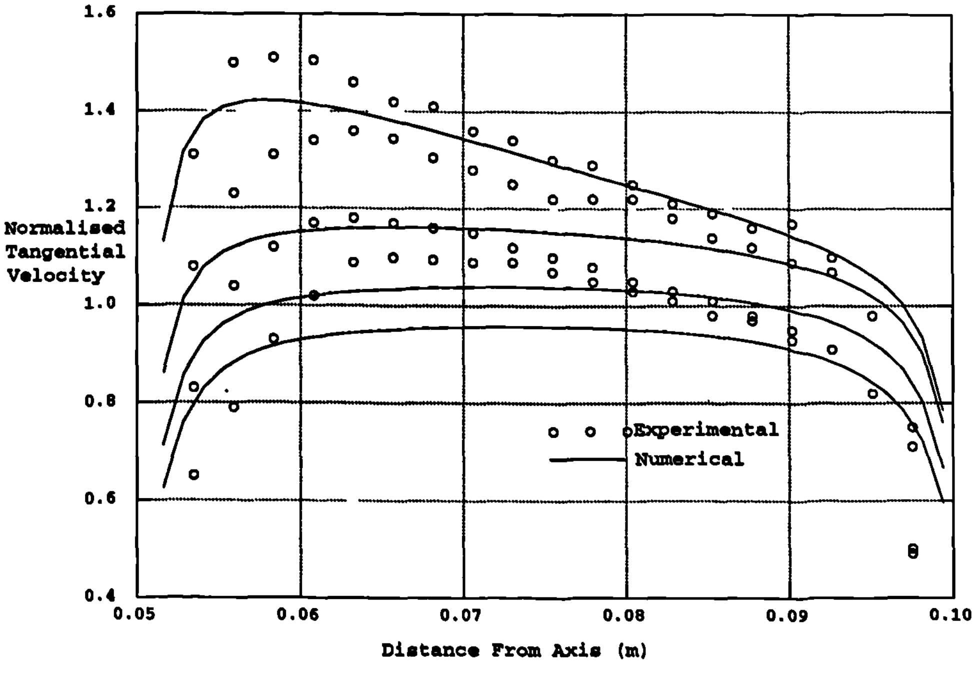

Figure 6.1.8: Turbulent Flow in an Annulus, RNG k-ϵ Model, Version 2

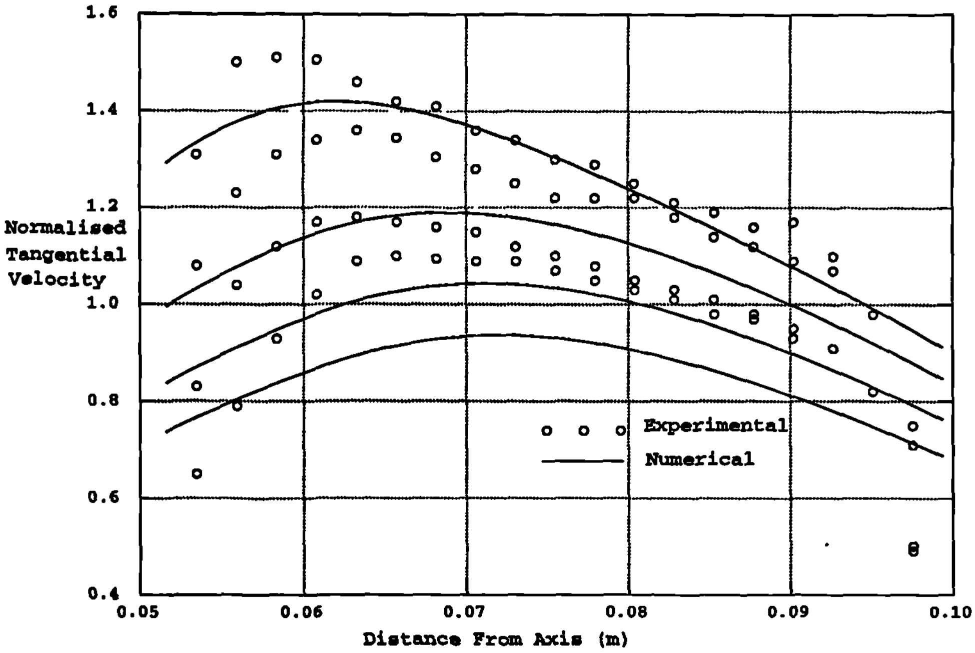

Figure 6.1.9: Turbulent Flow in an Annulus, k - L Mixing Length Model

here in which the RNG k-ϵ model produces answers which are very similar to those obtained from the use of the standard k-ϵ model. The k-L mixing length model does lead to an answer which predicts the movement of the maximal tangential velocity from the inner wall to the centre of the annulus better than the other models. Where the implementation of the model fails is in its prediction of the flow near the walls.

§6.2 Combustion Test Cases

Gaseous chemical reaction was simulated in the code using the Simple Chemical Reaction Scheme[45] (SCRS) model. In order to test the implementation of the

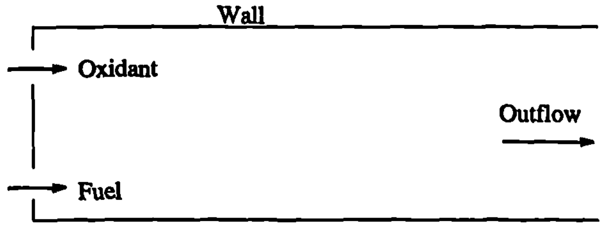

Figure 6.2.1: Boundary Conditions For Combustion Test Case

SCRS model a test problem involving the mixing of two jets was simulated in both CWNN and PHOENICS and the results were compared. The test case is shown diagrammatically in Figure 6.2.1.

§6.2.1 Diffusion Controlled Model

For the test case of the diffusion controlled model both the oxidant and fuel jets entered the solution domain at a speed of 0.1ms^{-1} and a temperature of 300K. The density was calculated using equations (4.1.10) and (4.1.11) with the molecular weights of the species being

Fuel	14
Oxidant	14
Product	28

All other properties were assumed to be constant with the heat of reaction equal to 10^6 and the stoichiometric value of f being 0.5. The mesh consisted of 40 elements in the 2m across the duct and 50 elements in the 10m along the duct. Figure 6.2.2 shows a comparison of the predicted temperatures on the horizontal centre line of the domain. PHOENICS predicts a slightly lower temperature at the 1m station but as this difference is about 3° in a rise of over 440° this is hardly a significant difference. The comparison of mass fractions of product at 0.3m, 1.9m and 5.9m, Figure 6.2.3, shows excellent agreement between the two codes.

§6.2.2 Kinetically Influenced Model

The effect of turbulence on the combustion process was modelled using the Eddy Break Up model(81, 82). To test this model the problem simulated in the previous subsection was used with the inflow velocity raised to 10 ms^{-1}. The turbulent quan-

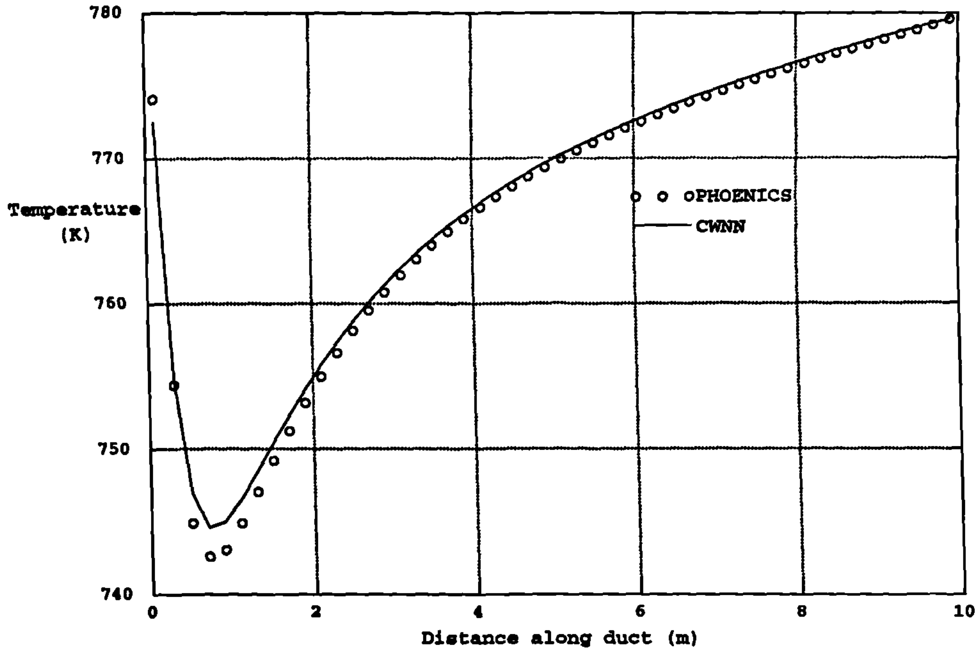

Figure 6.2.2: Diffusion Controlled Combustion, Centre Line Temperatures

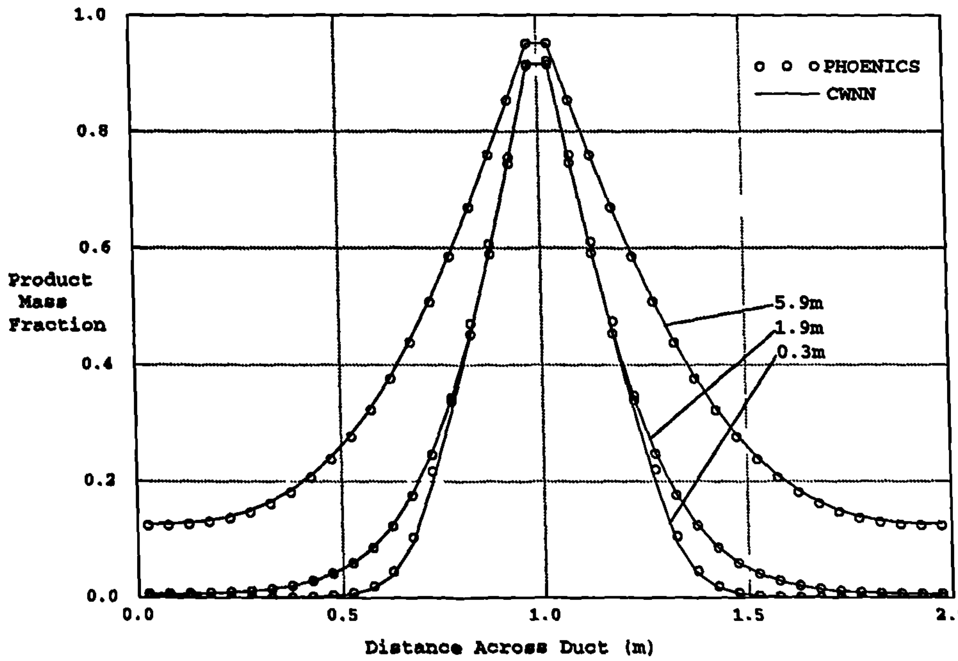

Figure 6.2.3: Diffusion Controlled Combustion, Mass Fractions of Product

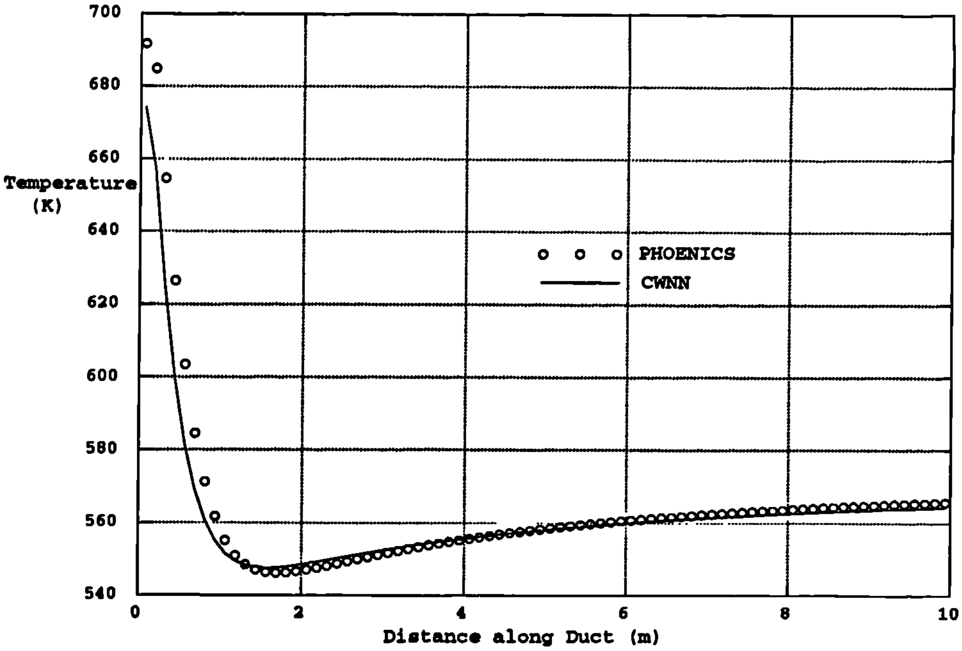

Figure 6.2.4: Kinetically Influenced Combustion, Centre Line Temperatures

tities at the inflow boundary were given the values of 0.1, for kinetic energy, and 1.0 for dissipation rate. All other boundary conditions, fluid properties and quantities relating to the reactions took the same values as in the Diffusion Controlled test case. The mesh outer dimensions were again 10m by 2m but in this case 40 elements were used across the duct and 80 along it. Figure 6.2.4 shows a comparison of the centre line temperatures predicted by PHOENICS and CWNN. Again the codes show excellent agreement. Figure 6.2.5 shows comparisons of the mass fraction of product on profiles across the duct at distances 0.56m, 1.81m and 3.68m along the duct. At the first of these cross sections there is a slight difference in the peak values, caused by the different discretisation techniques employed and the fact that the velocities are calculated at different locations. At all other distances from the inlets the profiles match extremely well.

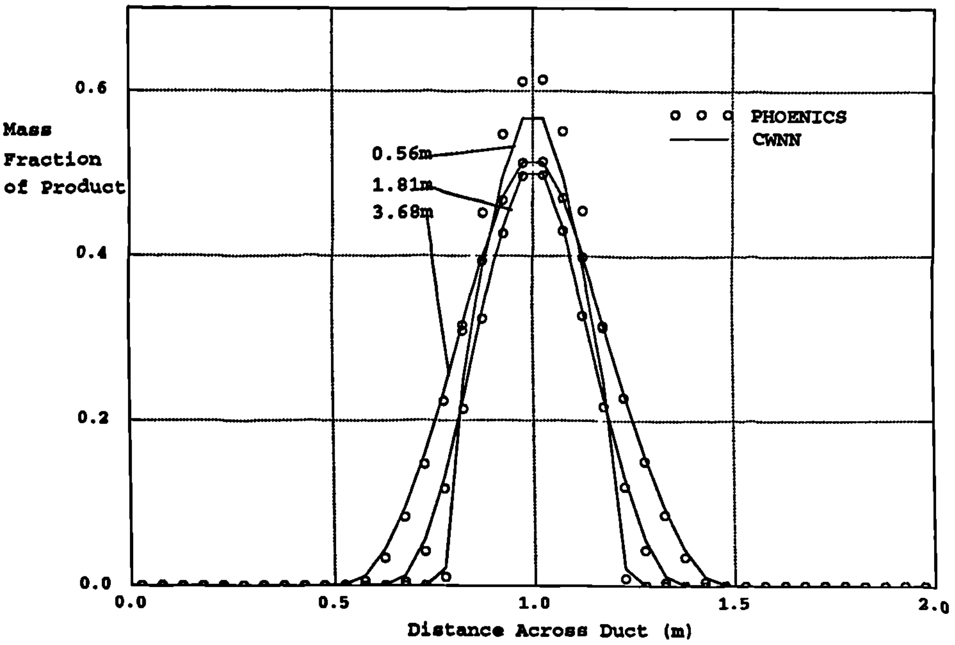

Figure 6.2.5: Kinetically Influenced Combustion, Mass Fractions of Product

§6.3 HIsmelt Cases

A series of simulations, relating to the HIsmelt direct smelting process(9,10), were performed to evaluate the ability of CWNN to simulate strongly swirling, turbulent, reacting flows. The three cases presented here indicate the steps taken in building up the current model. The first of these cases validates the anisotropic turbulence model, introduced in section 4.2.3, against experimental results. The second uses a three dimensional geometry with two swirling jets. For this case the general interpolation scheme is required to correctly interpolate the velocity vectors. Additionally, equations are solved to model heat transfer with combustion taking place. The final test case uses the full top space model in an axi-symmetric test case. In this case metal droplets are tracked in order to determine porosities which influence the flow of the continuum phase. The combustion model is required to simulate an inert species, the mass fraction of which varies independently of the other species. Radiation effects due to the presence of the droplets are also built into the model.

§6.3.1 Case 1: Turbulent, Axi-Symmetric, Single Tuyere

The mesh used in the first Rio Tinto Zinc test case is shown in Figure 6.3.1. The mesh consists of 50 elements in the axial direction and 76 elements in the radial direction in the main body of the domain. Extra blocks are added at the outlet, top left of the mesh, and inlet regions in attempt to reduce the influence of any assumptions made at these boundaries. Figure 6.3.2 shows a close up of the mesh in the region around the inlet. There is a degree of non-orthogonality in this region but the correction terms, presented in section 2.4, mean that there is no difficulty in solving on this mesh. The simulation was performed as incompressible, isothermal and turbulent with an inlet flowrate of 1.4Nm^{3}s^{-1} and a swirl number of around 0.6, where the swirl number is calculated using the formula

$$\frac{1}{r_{max}} \frac{\int \rho u_\theta u_z r \, dS}{\int \rho u_z^2 \, dS} \tag{6.3.1}$$

where the integration is over the area of the inlet, u_z and u_θ are respectively the velocity components in the axial and tangential directions and r_{max} is the outer radius of the inlet.

Figure 6.3.3 shows a velocity vector plot for the region around the inlet. The most significant aspect of the flow field is the reverse flow in the region around the polar axis. The advantage of such a flow pattern in the HIsmelt® process is that it leads to a main jet which has a large surface area on which chemical reactions can occur. These exothermic reactions cause a release of energy which raises the temperature of the main jet. This temperature is transferred to the metal droplets and bath as the main jet passes through the transition zone. Consequently, the chemical reactions provide a proportion of the energy required to smelt the iron in the bath. The reverse flow, also known as the centre jet, acts as a transport mechanism for the fuel species, bringing it into contact with the oxygen bearing main jet. The second and third test cases will model the influence of the thermal and chemical aspects of the HIsmelt® process. The aim of this test case is to indicate the relative merits of the turbulence models.

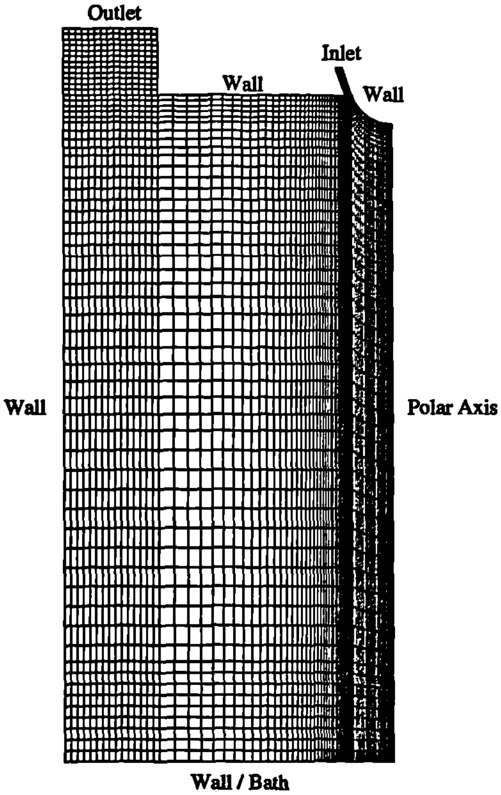

Figure 6.3.1: Full Mesh for Rio Tinto Zinc Case1

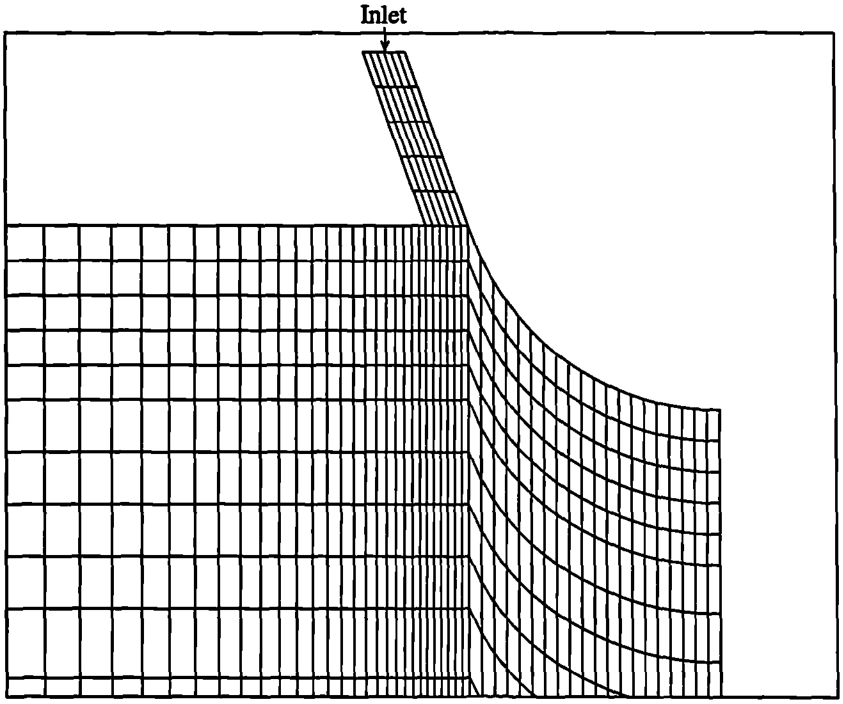

Figure 6.3.2: Mesh Near Inlet, Rio Tinto Zinc Case1

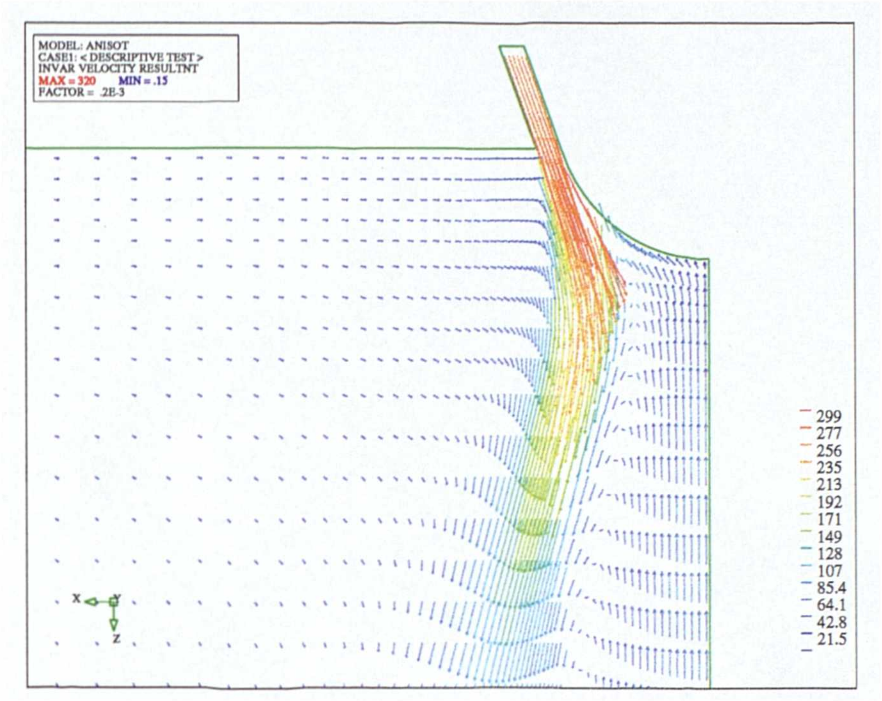

Figure 6.3.3: Flow Field for Rio Tinto Zinc Case1

The simulation was performed using the standard k-ϵ model, the RNG k-ϵ model and the anisotropic model. As has been observed in previous test cases there was very little difference between the flow fields predicted by the RNG and standard k-ϵ models. Axial and radial velocity profiles are presented in Figure 6.3.4 for both the standard k-ϵ model and the anisotropic model. In addition, experimental data(115) for the case is also included on the plots. Near the inlet there is little difference between axial velocities predicted by the two models. At the larger distances from the inlet, from 200mm onwards, the anisotropic model predicts the position of the maximum axial velocity much better than the k-ϵ model although both models under predict the amount of reverse flow near the axis. For the swirl component of velocity the anisotropic model matches the experimental data extremely well, much better than the k-ϵ model. There is a discrepancy between the numerical and experimental results for the swirl velocity at 300mm. On a polar axis there must be zero swirl velocity, predicted by both numerical simulations, but the experiment gives a value of -10m^{-1}. The reason for the experiment predicting non zero swirl on the axis is not clear but the possibility of the flow axis moving from the axis of symmetry must be considered.

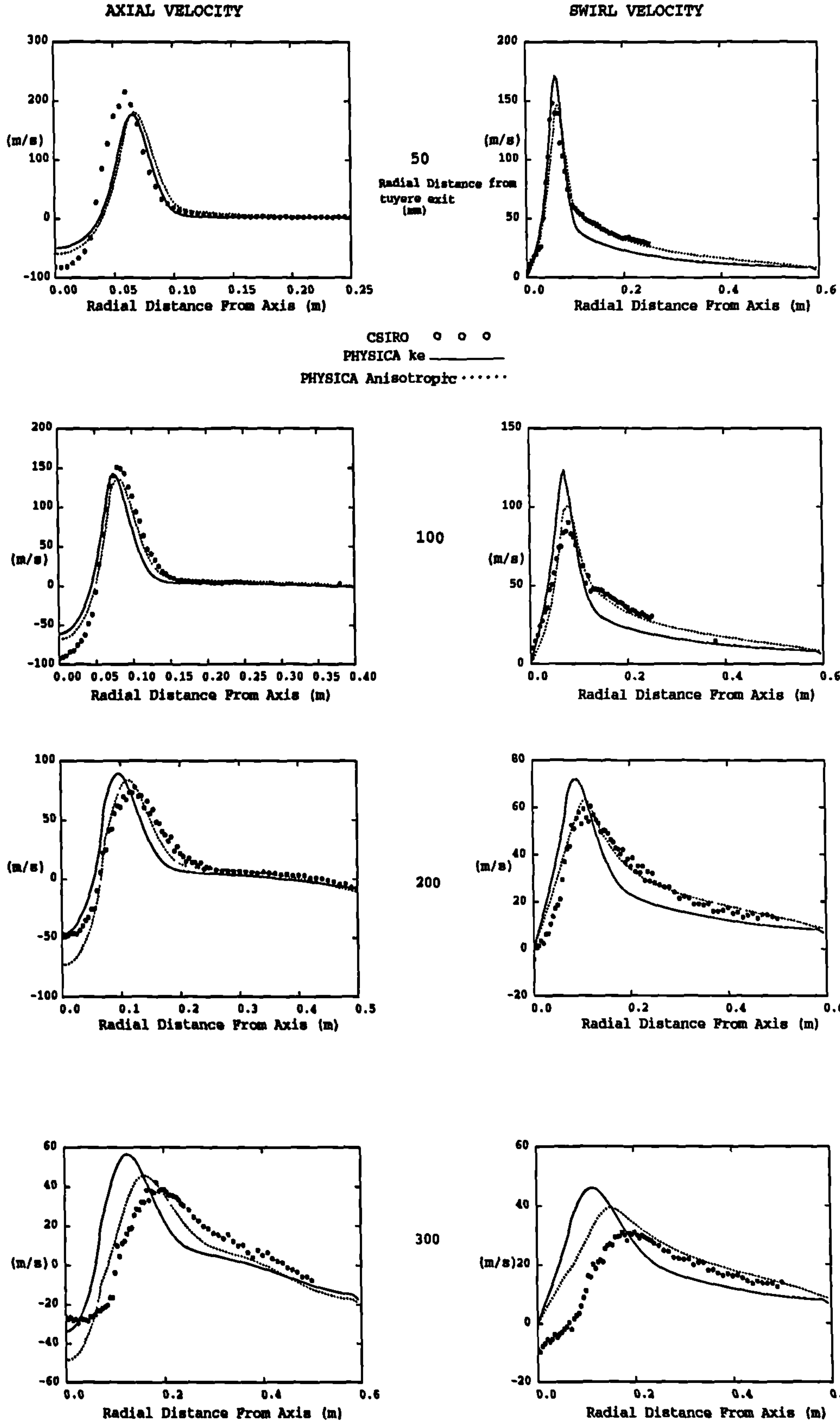

Figure 6.3.4: Axial and Tangential Velocity Profiles, Rio Tinto Zinc Case 1

§6.3.2 Case 2: Three Dimensional, Twin Jet, Smelt Reduction Vessel

The second test case models the fluid flow and heat transfer, including chemical reactions, in a domain representing a possible design for the HIsmelt® Smelt Reduction Vessel (SRV), Figure 1.2.1. The function of the SRV is to smelt iron in the metal bath. The reactions taking place in the bath are endothermic, consequently there is a need to provide large quantities of heat to the bath. This energy is produced by chemical reactions taking place in the top space, the region of the SRV above the metal bath.

This test case consists of the simulation of a pair of jets entering a barrel shaped solution domain, Figure 6.3.5. The lance jet enters the domain vertically whilst the tuyere jet is aimed towards the first at an angle of 45° to the vertical, see Figure 6.3.6. The flow leaves the domain through the outlet located at the right end of the domain as seen in Figure 6.3.6. The simulation was performed as reacting and turbulent with the Simple Chemical Reaction Scheme used to model the combustion and the anisotropic model used for the turbulence. The jets entered the solution domain at 256 ms^{-1} with a swirl number of 0.6. The jets were assumed to consist of air 23% O_2, at a temperature of 1,500 K. The fuel, CO, entered the domain from the base of the mesh at 1 ms^{-1} and a temperature of 1,800 K.

The mesh used in the simulation was constructed in Patran(116) using a mixture of wedge and hexahedral element types to produce a mesh consisting of just under 90,000 elements. The main concern in the generation of a mesh for a geometry of the type required for this simulation is maintaining a high degree of orthogonality. The correction terms presented in this thesis will improve an answer but their value should remain small compared to the terms contributing to the system matrix. Figure 6.3.6 shows the symmetry plane of the mesh used in the simulation. It is beneficial to align the mesh with any flow characteristics, which leads to the two cone sections leading from the inlets. These cones must not intersect to avoid *highly non-orthogonal* connections, consequently one cone has to bend. The orthogonality of a connection s measured by the dot product of the unit normal of the face between two elements, $\hat{\underline{n}}$, with a unit vector in the direction from one element centroid to the other, $\hat{\underline{v}}$

$$\dot{n} = \hat{\underline{n}} . \hat{\underline{v}} \tag{6.3.2}$$

If this value is 1.0 the connection is orthogonal with the value falling as orthogonality deteriorates. The mesh generated in Patran had a worst orthogonality of 0.015. This means that the normal and line connecting the centroids are almost at right angles to each other. In an attempt to improve the orthogonality of the mesh a simple point "shuffle" utility was written. This utility iteratively moved the mesh by visiting each corner of every face in turn. The utility attempts to improve the mesh so that no $\dot{n}$ value is less than a "cut off" value $\dot{n}_c$. At each stage a point is moved to a new position $\underline{r}^{i+1}$ using to the formula

$$\underline{r}^{i+1} = \underline{r}^i + \gamma^i \hat{\underline{v}}^i \lambda \tag{6.3.3}$$

where λ is a relaxation factor, $\hat{\underline{v}}^i$ is the i^{th} iteration value of the unit vector connecting the element centroids either side of the face and

$$\begin{aligned} \gamma^i &= \omega \xi \\ \omega &= \frac{\hat{\underline{v}}^i . \underline{\eta}^i}{|\underline{\eta}^i|} \end{aligned} \tag{6.3.4}$$

In the above equation $\underline{\eta}^i$ is the i^{th} iteration value of the vector from the point to the face centre and ξ is calculated using the formulae

$$\xi = \begin{cases} \frac{\dot{n}-\dot{n}_c}{\dot{n}_c}\alpha_f & \text{if } \omega < 0 \text{ and } \dot{n} > 0 \\ \frac{\dot{n}_c-\dot{n}}{\dot{n}_c}(1-\alpha_f) & \text{if } \omega > 0 \text{ and } \dot{n} > 0 \\ \alpha_f \omega & \text{if } \omega < 0 \text{ and } \dot{n} < 0 \\ (1-\alpha_f)\omega & \text{if } \omega > 0 \text{ and } \dot{n} < 0 \end{cases} \tag{6.3.5}$$

where α_f is the linear interpolation weighting value for the face. The algorithm worked best if the cut off $\dot{n}$ value was not significantly higher than the worst orthogonality of the mesh. Boundary points were not moved in any way which imposes some limits on the quality of the final mesh. The algorithm improved the mesh considerably but in spite of all efforts some small portions of the mesh were only slightly improved. Figure 6.3.7 shows a contour plot of the orthogonality values on a plane just above the base.

Figures 6.3.8 to 6.3.11 show contour plots of the vertical velocity on the symmetry plane and on 3 horizontal planes at selected heights up the barrel. It can be seen that the interaction of the swirl in the jets leads to them drifting away from their axes, with the left jet moving slightly to the top of the plane and the right jet moving to the bottom. The returning gases from the left end of the domain to the outlet at the right of the domain can be seen to cause both jets to be blown to the right of the domain in Figure 6.3.12 which shows a contour plot of the rotational component of velocity. Figure 6.3.13 shows a plot of temperature and Figures 6.3.14 and 6.3.15 show the mass fractions of oxidant and fuel respectively. It can be seen that all the combustion takes place either in the jets or in the region between the jets leading to a high gas temperature in this region. Part of this energy is extracted by the bath to be available for the reactions which smelt the iron. The rest leaves the domain through the outlet region. For this simulation the outlet temperature is around 2,200 K around 400 K higher than the bath temperature. This is energy which is potentially lost to the process. Such a high loss through the outlet would be unacceptable in the real process but there are other transport mechanism which are needed to complete the model of the top space. These influences are introduced in the following test case.

§6.3.3 Case 3: Simulation Employing the Top Space Model

The third of the Rio Tinto Zinc test cases involves an axi-symmetric simulation employing the complete top space model, as shown in Figure 1.2.2. The anisotropic turbulence model is used in the resolution of turbulent effects. The combustion model is based on the SCRS diffusion controlled model but, because the mass fraction of

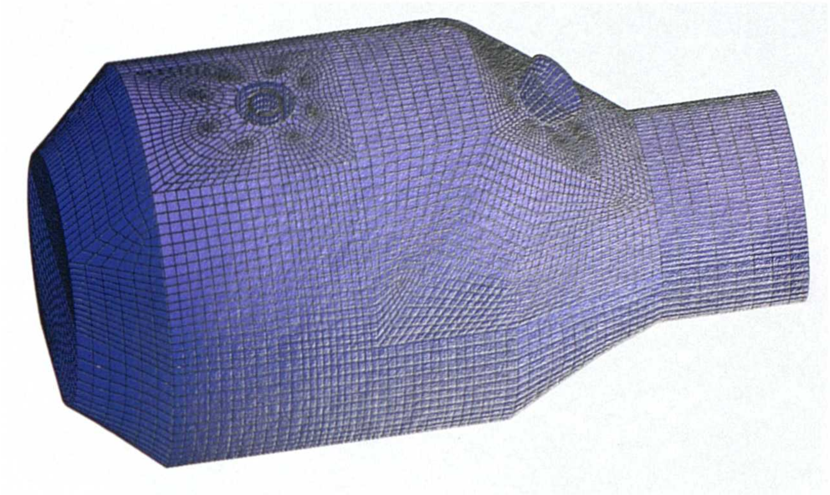

Figure 6.3.5: Surface of Mesh, Rio Tinto Zinc Case 2

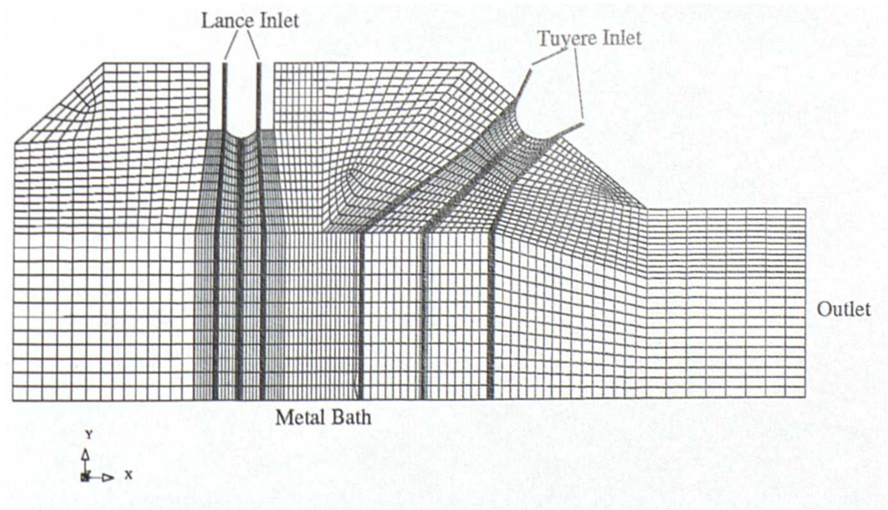

Figure 6.3.6: Symmetry Plane of Mesh, Rio Tinto Zinc Case 2

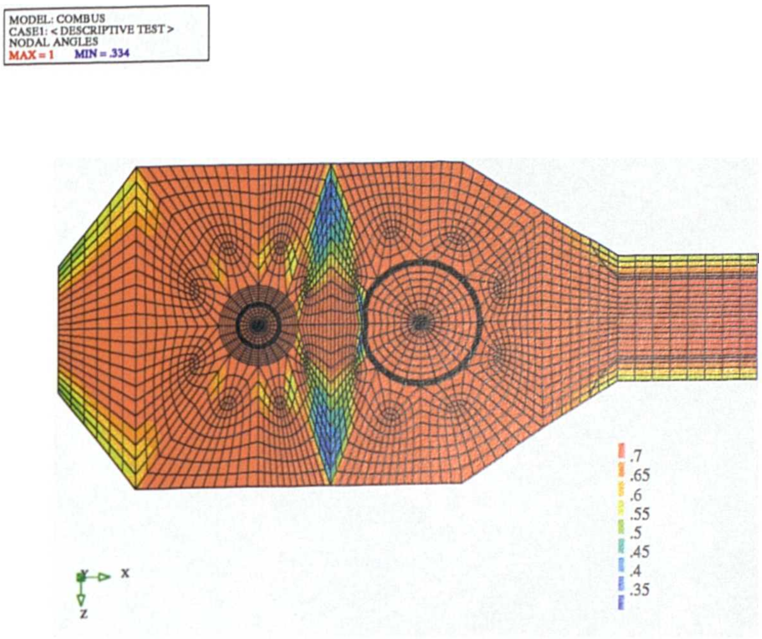

Figure 6.3.7: Mesh Quality, Mesh Base, Rio Tinto Zinc Case 2

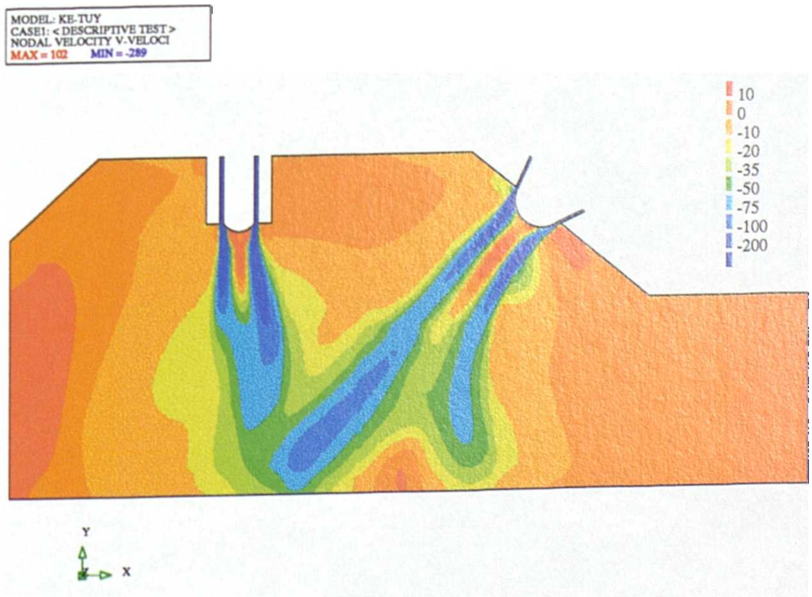

Figure 6.3.8: Vertical Velocity on Symmetry Plane, Rio Tinto Zinc Case 2

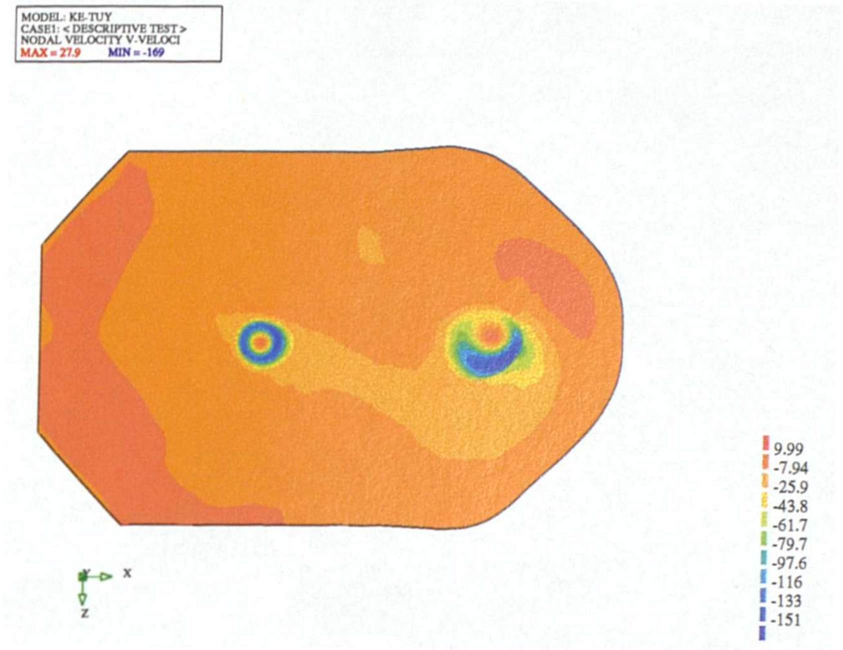

Figure 6.3.9: Vertical Velocity near Inlets, Rio Tinto Zinc Case 2

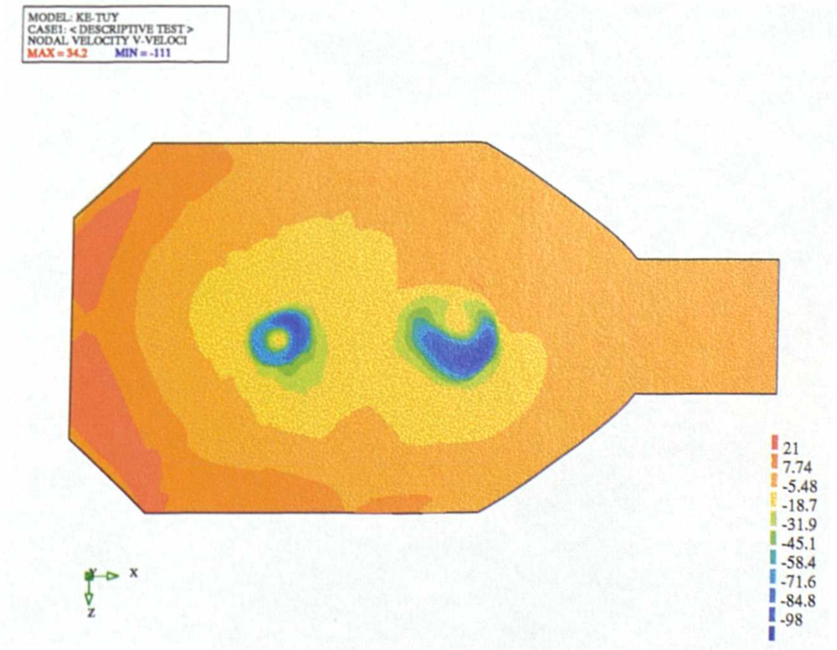

Figure 6.3.10: Vertical Velocity, Horizontal mid Plane, Rio Tinto Zinc Case 2

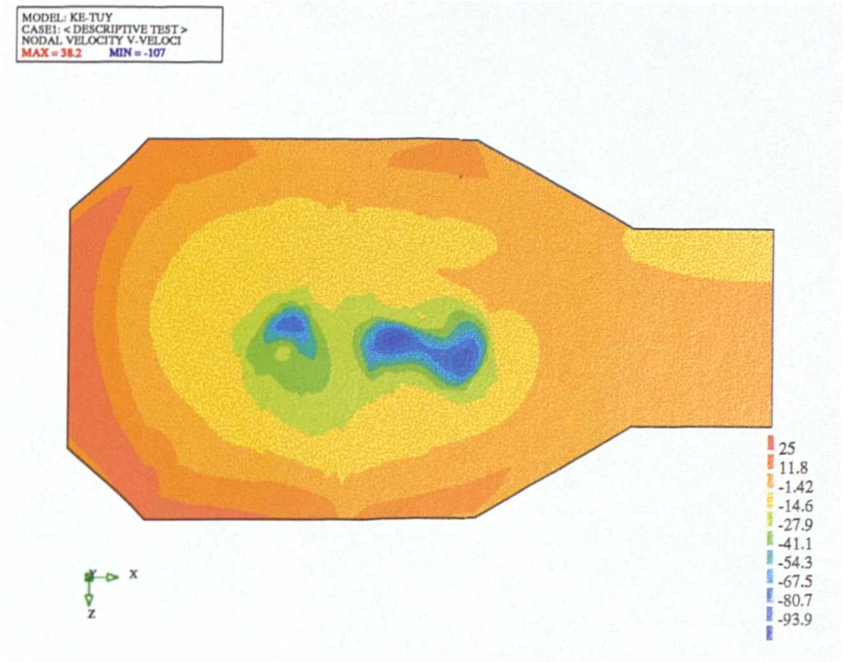

Figure 6.3.11: Vertical Velocity, Base Plane, Rio Tinto Zinc Case 2

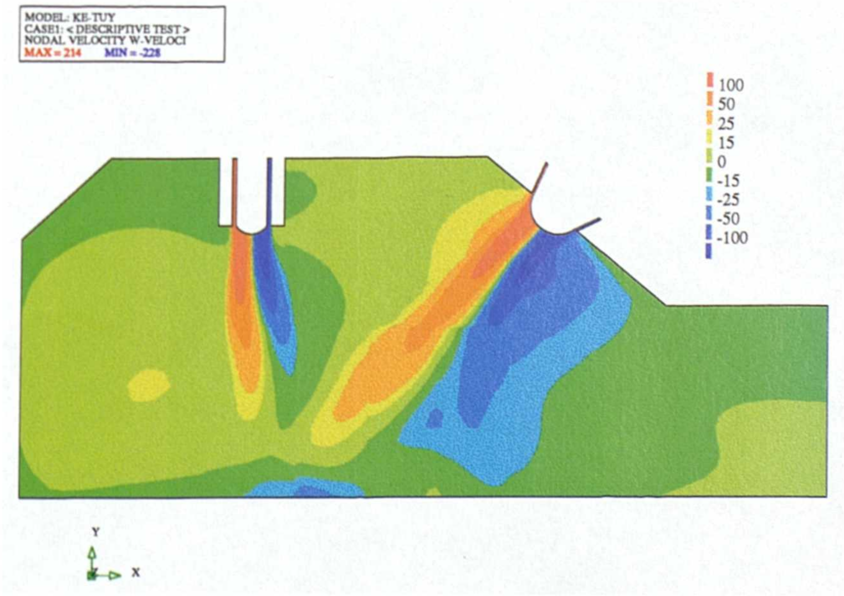

Figure 6.3.12: Swirl Velocity, Symmetry Plane, Rio Tinto Zinc Case 2

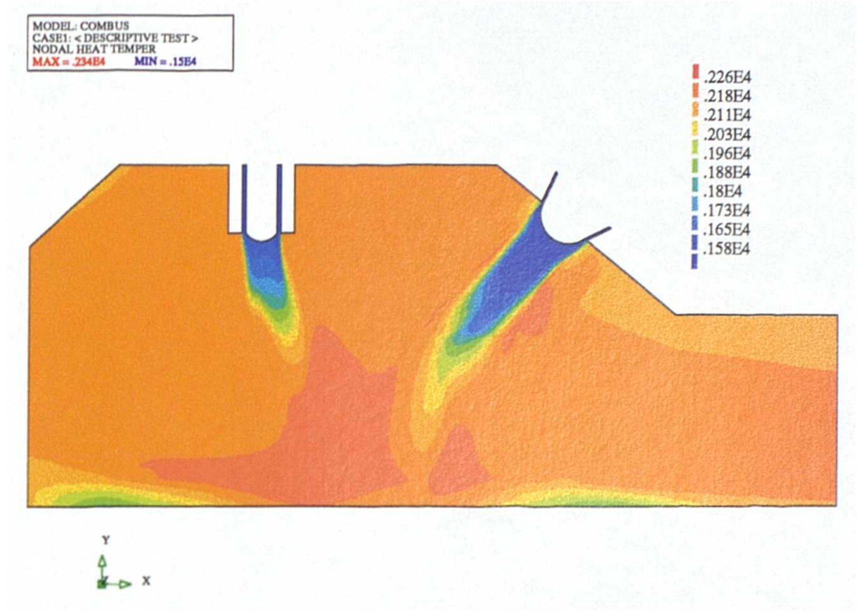

Figure 6.3.13: Temperature Contours, Symmetry Plane, Rio Tinto Zinc Case 2

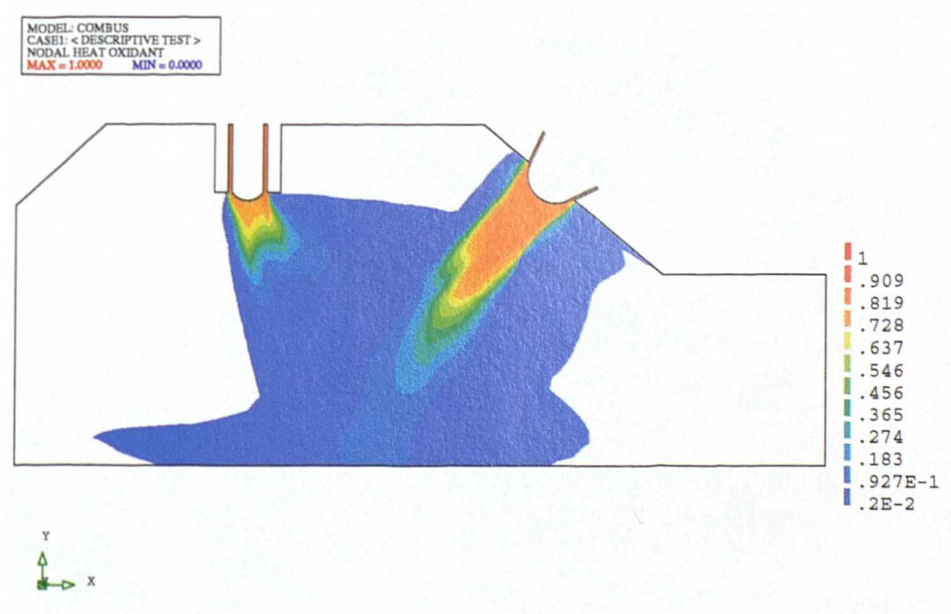

Figure 6.3.14: Mass Fraction of Oxidant, Rio Tinto Zinc Case 2

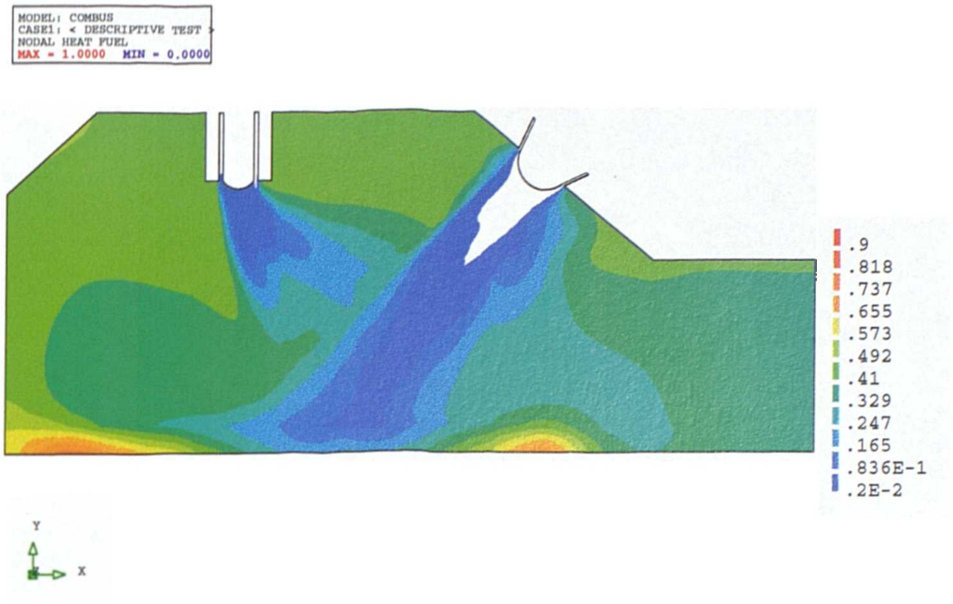

Figure 6.3.15: Mass Fraction of Fuel, Rio Tinto Zinc Case 2

the inert species, N_2, is not linearly related to the mass fraction of any other species, an extra conservation equation, for the mass fraction of the inert species, needs to be solved. This extra equation has no source term and is identical in form to the equation solved for the mixture fraction. The two solved variables in the modified SCRS are f_1 and f_2 where

$$\begin{aligned}
m_{N_2} &= f_2 \\
f &= f_1/(1.0 - f_2) \\
m_{ox} &= \begin{cases} 0 & f > f_{st} \\ (1 - f_2)(f - f_{st})/(1 - f_{st}) & f < f_{st} \end{cases} \\
m_{fuel} &= \begin{cases} (1 - f_2)(1 - f/f_{st}) & f > f_{st} \\ 0 & f < f_{st} \end{cases} \\
m_{pr} &= 1.0 - f_2 - m_{ox} - m_{fuel}
\end{aligned} \tag{6.3.6}$$

where f is the mixture fraction and f_{st} is the stoichiometric value of the mixture fraction. Figure 6.3.16 shows the above relationships diagrammatically for a chosen value of f_2.

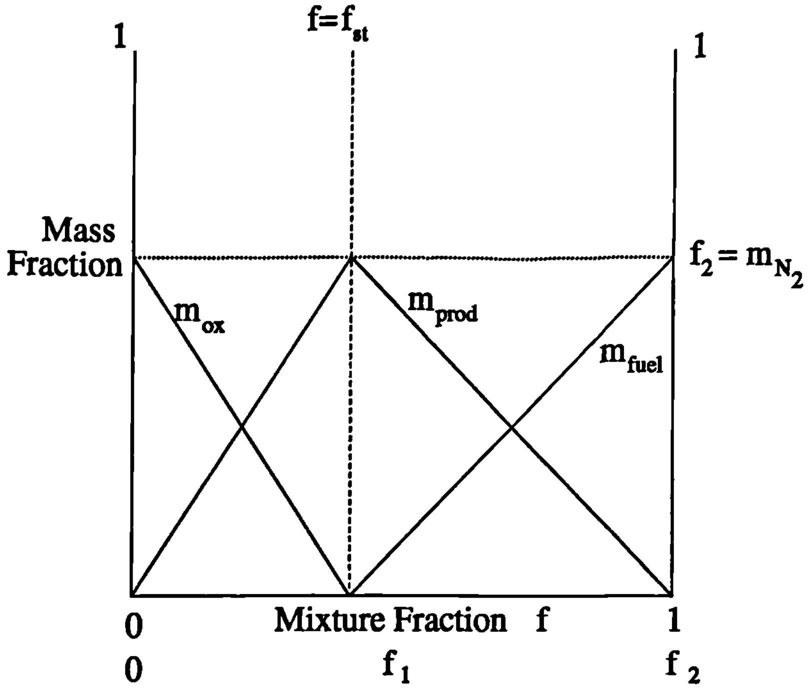

Figure 6.3.16: Modified SCRS Algorithm, Graph of Mass Fractions against f_1

Metal droplets are emitted from the metal bath into the top space creating the transition zone between the metal bath and the Gas Zone, Figure 1.2.2. The mass of metal expelled from the bath is divided into packets with one droplet representing each packet. Each droplet is given a random diameter and initial speed using a normal distribution with the mean and spread of values taken from tin bath models(117) and water models(118). Random numbers also determine the initial direction of the droplet's motion. Lagrangian tracking is then used to determine the path travelled by the droplet. To reduce the influences of the starting values attributed to the droplets it is beneficial to track in the order of 10,000 droplets. The surface area of the droplets is the most important parameter governing the heat, chemical species and momentum exchange between the gas and the droplets. It was found(10) that unless break up of the droplets by jet motion was properly accounted for, the correct response of the process to an increase in the jet velocity could not be numerically reproduced. A simple break up model is used in which the stable droplet diameter is given by

$$\frac{\sigma_d W_{crit}}{\rho_c \left|\underline{u}_c - \underline{u}_d\right|} \tag{6.3.7}$$

where σ_d is the surface tension of the droplet, W_{crit} is the critical Weber number and the subscripts c and d indicate values for the continuum and droplet phases respectively. If this stable diameter is less than the current droplet diameter, break up occurs. If the droplet breaks up, its size is reduced by a preset factor. The size of the droplet in any element it visits and its residence time allow the calculation of a volume fraction of droplets for each element as well as the total surface area of droplets in the element. The surface area is used to calculate the absorption and scattering coefficients used in the radiosity equation. Also the presence of droplets in an element provides a friction source into the momentum equations and a heat source in the enthalpy equation.

The mesh used in the simulation, consisting of 2665 elements, is shown in Figure 6.3.17. The boundary conditions and model parameters for the simulations presented here were as follows.

The tuyere boundary conditions were:

Flowrate	25,000 Nm^3/hr
Axial Velocity	526.0 ms^{-1}
Swirl Number	0.25
Volume Fraction of O_2	0.205
Volume Fraction of N_2	0.775
Volume Fraction of CO_2	0.02
Air Temperature	1150.0 C

the bath boundary conditions were:

Flowrate of N_2	1,600 Nm^3/hr
Flowrate of CO	6,525 Nm^3/hr
Bath Temperature	1500.0 C

the particle constants were:

Droplet Mass Flowrate	3325.0 kg/s
Average Droplet Size	0.02 m
Average Droplet Velocity	3.5 ms^{-1}
Velocity Spread	3.5 ms^{-1}
Droplet Surface Tension	1.20 Nm^{-1}
Critical Weber Number	13.0
Size Fraction after Breakup	0.06147

In the above tables the unit Nm^3/hr is the volumetric flowrate at atmospheric pressure and 0.0° C. A 200 iteration simulation of this case took 420 seconds running on a DEC Alpha. The simulation was run with droplet tracks being calculated every 6 outer iterations over the first 30 iterations of a 200 iteration simulation. Convergence was assumed when the error in the energy and species balances fell below 0.5% of their inflow boundary values. This required in the region of 3000 iterations. The results from this simulation is show in Figures 6.3.18 to 6.3.23. The single contour line on each of these figures corresponds to the volume fraction of droplets being equal to 0.001. This is effectively the boundary of the transition zone. The reason for the lack of droplets in the region near the axis can be seen from Figure 6.3.18. The high velocities in this area immediately force the metal droplets down and back towards the bath. The temperature contours of Figure 6.3.19 show the effects of the combustion processes taking place on the outer edges of the jet. The mass fractions of the oxidant, O_2, fuel, CO, and product CO_2 can be seen in Figures 6.3.20, 6.3.21 and 6.3.22 respectively. All the oxygen is consumed in the jet prior to it reaching the surface of the bath. Consequently, the gas phase is at its hottest on entry into the transition zone. As the gas passes through the transition zone, energy is transferred to the metal droplets. The rate of transfer is related to the temperature difference between the droplet and the gas phase. For this simulation the gas temperature falls from a maximum value of 2300 C to 1558 C at the outlet. This compares with a temperature of the droplets of 1500 C. This means that in this case the droplets ex-

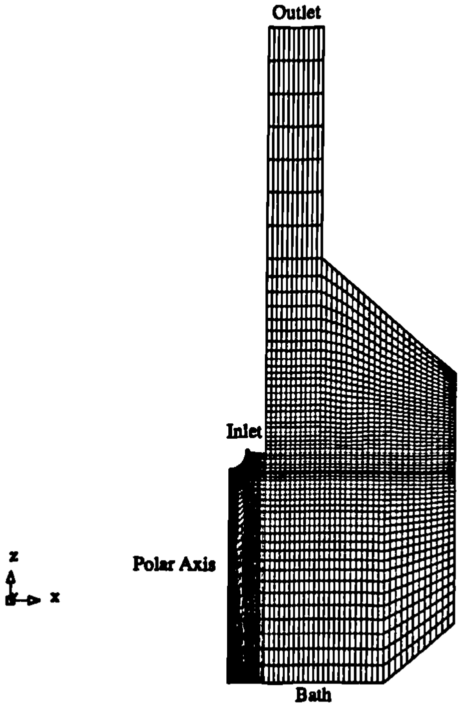

Figure 6.3.17: Mesh used in the Simulations of Rio Tinto Zinc Case 3

tract a significant proportion of the available energy. Figure 6.3.23 shows a contour plot of the volume fraction of droplets. The hole 'punched' into the droplets by the jet is very clear in this plot.

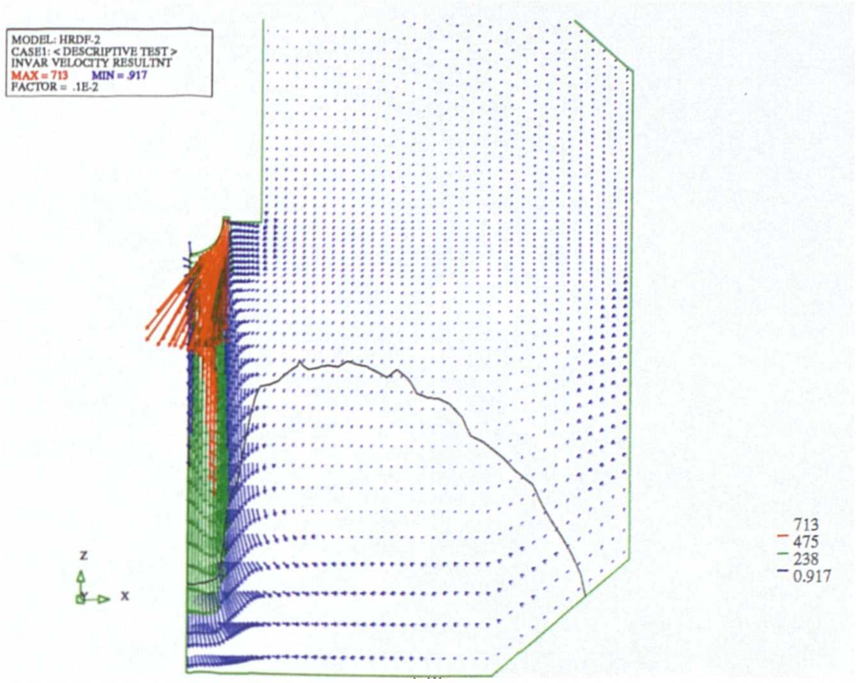

Figure 6.3.18: Velocity Vectors, Rio Tinto Zinc Case 3

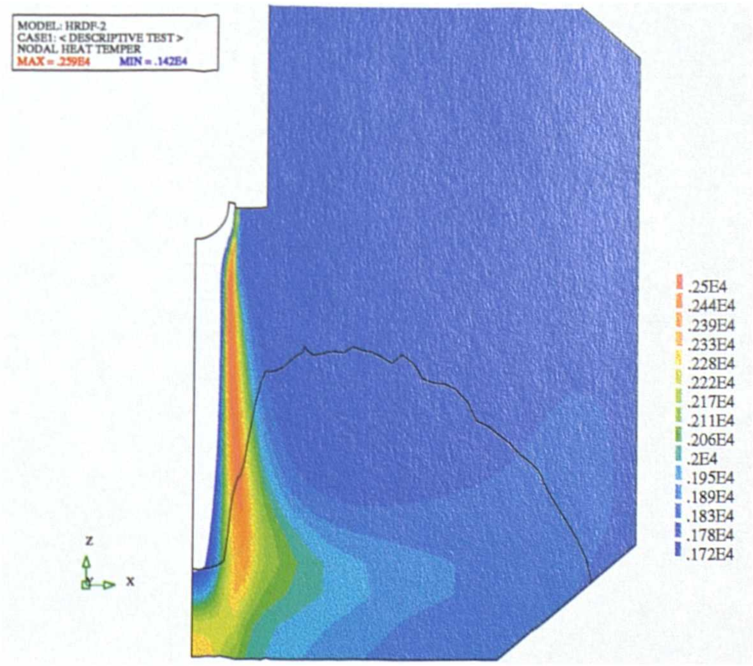

Figure 6.3.19: Temperature Contours, Rio Tinto Zinc Case 3

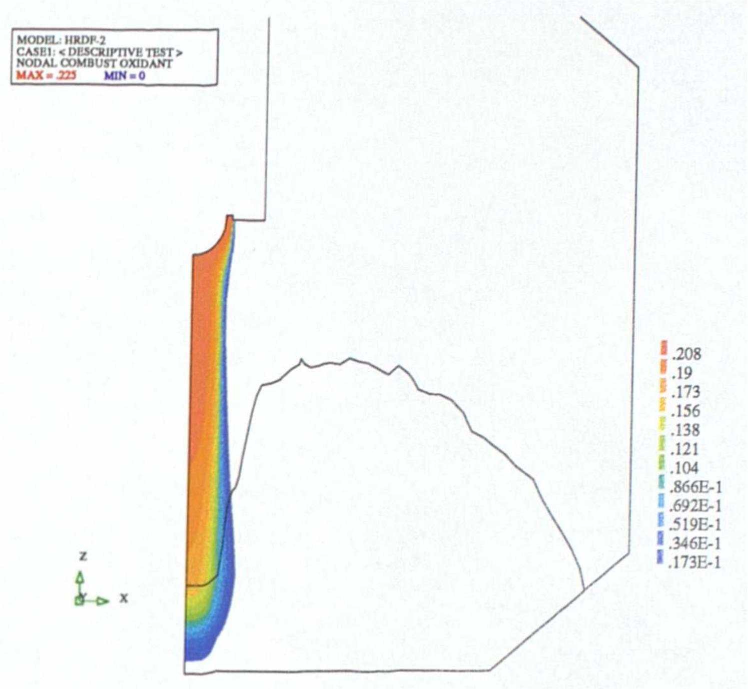

Figure 6.3.20: Mass Fraction of Oxidant, Rio Tinto Zinc Case 3

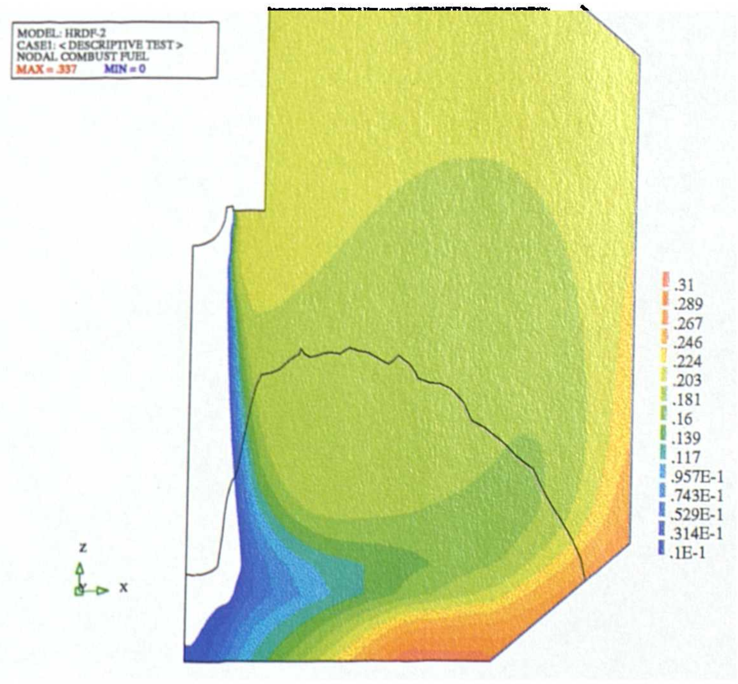

Figure 6.3.21: Mass Fraction of Fuel, Rio Tinto Zinc Case 3

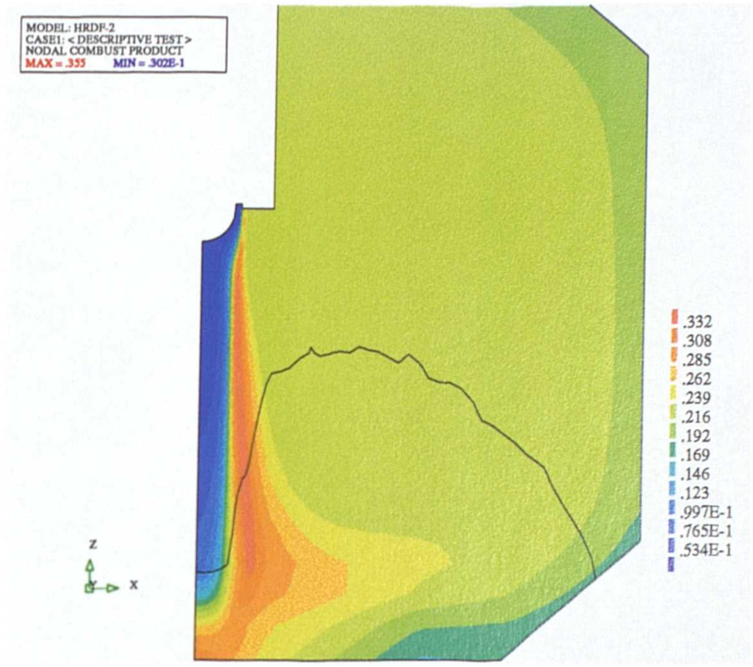

Figure 6.3.22: Mass Fraction of Product, Rio Tinto Zinc Case 3

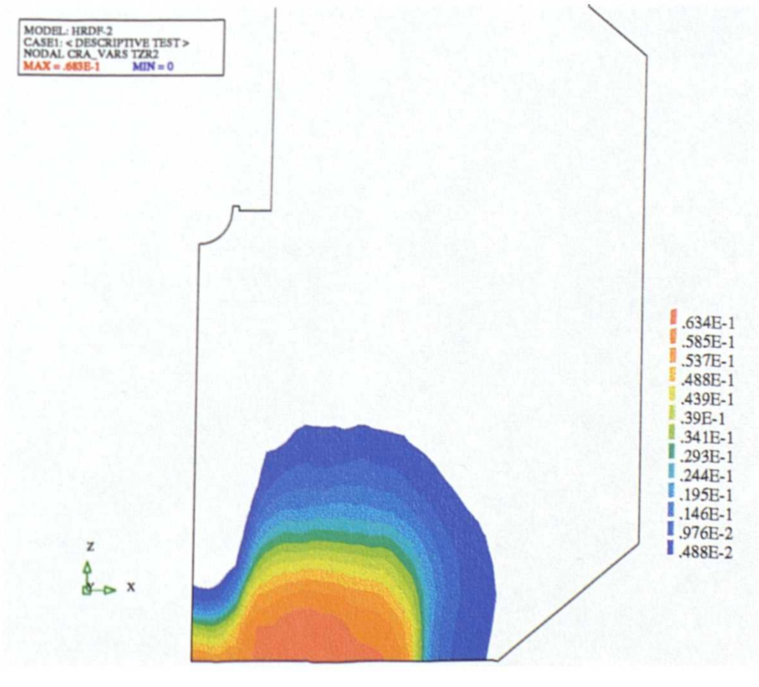

Figure 6.3.23: Droplet Volume Fraction, Rio Tinto Zinc Case 3

CHAPTER SEVEN

Conclusions

§7.1 Concluding Remarks

The work presented in this thesis describes the

- mathematical basis
- algorithms
- hysical models

that provide a set of tools to solve swirling, turbulent, reacting flows on unstructured meshes within the Finite Volume framework All quantities are solved and/or stored at the centroid of the elements of the mesh. To alleviate the possibility of checkerboard effects in the pressure and velocity fields the Rhie - Chow Interpolation Method is used to evaluate face values of the velocity vector. The pressure field is calculated by means of the solution of a pressure correction equation based on the SIMPLE algorithm. The mesh, over which the solution is obtained, can consist of a mixture of any type of polyhedral elements. This is only limited by the use of the tracking algorithms, which need shape functions for each element type used, and the availability of tools both to generate the desired element type and then to visualise the results. The results from simulations on Cartesian meshes compare well with those obtained from both PHOENICS, a staggered mesh, block structured code, and Flow3d (now known as CFX), a structured, collocated mesh code.

When using unstructured meshes care has to be taken to handle the skewness of the mesh. If no correction terms are included in the discretisation of the conservation equations there may be a preferential direction for the transport of the *solved quantity*. The contributing factors to mesh skewness are divided into two independent effects in order to simplify the corrections. These two forms of skewness are termed non-orthogonality and non-conjunctionality. Correction terms for both these forms of mesh skewness are introduced in this study. Test cases are presented which show that use of these terms will lead to a correct answer being obtained on meshes which would otherwise lead to an erroneous result. These corrections are validated for both a scalar variable and for use within the hydrodynamics section of the code.

For most problems it is possible to treat the three components of velocity as three independent scalar variables. This allows the same methods to be used in their solution as for any other variable. The only difference being in the use of the SIMPLE algorithm to obtain corrections to the current estimate of the velocity field. When solving flows with a high degree of curvature in the velocity field it has been shown that the use of interpolation of the Cartesian components of velocity leads to an over prediction of the decay of the rotational component of velocity. One option would be to solve the cylindrical polar components of velocity. This option is sensible if a cylindrical polar mesh is used but in such a case there is no advantage in using an unstructured mesh code. In general the solution of the polar components of velocity on an unstructured mesh would lead to some highly complex correction terms. The option used is to solve the Cartesian components of velocity in all cases and to use interpolation techniques based on flow characteristics. When the axis of the flow rotation can be aligned with the line y=0, x=0 then cylindrical polar interpolation is used. The adjustments to both the estimation of the velocity vector on a face and to the convection term are shown to produce very similar answers to a code solving for the cylindrical components of velocity. For the case when the position of the axis of the flow curvature is unknown prior to the simulation the general interpolation scheme is introduced.

One of the major difficulties in the modelling of swirling jets is the prediction of the turbulent quantities. The problem with the standard k-ϵ model is that it assumes isotropic turbulent eddies, but in swirling flows the eddies can be considerably stretched in the tangential direction. Test cases are presented, which collaborate the findings of previous work(119,120), which indicate there are considerable differences between the predictions of the k-ϵ model and experimental data. One of the newer advances in the field of turbulence modelling is the use of the ReNormalisation Group (RNG) Methods. Claims have been made that applying these methods to the standard k-ϵ model produces a model which is able to accurately model swirling flows. As part of this study the RNG k-ϵ model has been compared with the standard k-ϵ model in a number of swirling flows. In all cases the RNG k-ϵ model has given a better answer than the standard k-ϵ model but the these improvements are negligible compared with the difference between the k-ϵ model and experimental data. In order to accurately solve highly turbulent swirling flows a hybrid, mixing length and k-ϵ, model has been developed. This model uses the standard k-ϵ model for all components of velocity except for the tangential component which uses a modified mixing length model. The predictions from this model are shown to be a significantly better fit to experimental data than those of the k-ϵ model.

§7.2 Future Work

For many problems the algorithms described in this study produce suitably accurate numerical results. Despite this a number of areas remain where further work is required to improve the methods described in this thesis. Several of these topics are discussed in the remainder of this section.

§7.2.1 Calculation of Convection Fluxes

In section 2.3.1 two problems with the Rhie - Chow interpolation method(44) where discussed. The first of these problems showed that although the calculated face based convective fluxes satisfy continuity throughout the domain, the element based values may not. A test case which highlighted the problem was simulated in both CWNN and Flow3d. The second problem was shown to exist when triangular elements are used through the analysis of a simple problem. An investigation into the various questions relating to Rhie - Chow interpolation, and whether an alternative interpolation scheme should be used, would benefit anyone using the collocated mesh approach.

§7.2.2 Pressure Correction Algorithm

One of the areas which could significantly reduce the simulation time is in the formulation of the pressure correction equation. The current method frequently leads to oscillations in the residuals for both the continuity error and the errors in the momentum field. It is common to see the continuity error reducing whilst the momentum error is increasing. After a number of iterations the trend will be reversed with the continuity error rising and the momentum error falling. This indicates that there is an inconsistency between the calculated pressure and momentum corrections. Currently the only non zero, off diagonal coefficients in the pressure correction equation correspond to immediate neighbours of the element. There may be benefit in changing this to include neighbours of these neighbours. In the SIMPLE algorithm, section 3.1.1, the error in the velocity field is given by

$$u'_f = \overline{u'_f} + \frac{1}{a_p}\left(\overline{\nabla_x p'_f} - \nabla_x p'_f\right) \qquad (7.2.1)$$

In the present formulation of the pressure correction equation the first two terms on the right of this equation are dropped. An alternative is only to drop the first term.

The $\overline{\nabla_x p'_f}$ term, which is the average of the gradients of the pressure correction in the two elements either side of face f, will lead to coefficient corresponding to neighbours of neighbours appearing in the pressure correction equation. This should have the advantage of spreading the continuity errors more quickly throughout the solution domain, consequently requiring less iterations of the linear solver.

§7.2.3 General Interpolation

The general interpolation scheme, for calculating velocity on a face, requires the use of massless particles to determine a set of flow axes which are used to interpolate the velocity field. This works only when there is prior knowledge of some details of these axes, for example the swirl is generated by a boundary condition. It would be preferable if a scheme could be formulated which allows each element to calculate its own axis and the corresponding radial, axial and tangential directions. This would mean that the scheme could be used in all cases, without the user requiring any kno ledge or guess at the expected solution. The cost of such an approach would be additional code complexity.

§7.2.4 Mesh Correction Terms

The terms added to account for the various forms of mesh skewness are correct to first order. Problems are frequently encountered early in simulations when the approximations made in calculating the corrections can lead to significant errors in their values. This can cause difficulties in the convergence of a simulation which frequently means more iterations and, as a consequence, a longer run time. Currently the terms are added into the discretised equation with little linearisation. It may be possible to reformulate the calculation of the errors to introduce a form of relaxation which does not significantly affect the overall simulation time.

§7.2.5 Starting and Stopping

Frequently there is a need to apply significant relaxation at the initial stages of a simulation to avoid divergence. The value of the relaxation is frequently one or two orders of magnitude greater than when performing a similar simulation in PHOENICS. This initial relaxation can be fairly quickly removed but it still leads to a requirement to use more iterations than should be necessary. An investigation into the causes of this excessive initial instability would be of benefit to the running of the code. As noted in section 7.2.2 the pressure correction algorithm can lead to oscillations in the continuity and momentum errors. The problems at the start of a simulation may just be a critical extension of this but there may be other reasons for the difficulties.

In addition, the recognition of convergence needs to be improved. In most complex cases CWNN reaches a stage in the solution procedure where the residuals just "bounce" around. At this stage the problem may still be moving towards the required solution and further iterations will benefit the accuracy of the numerical result. The difficulty arises in recognising how many extra iterations are necessary and when the simulation answer is sufficiently accurate. Some better measure of convergence is required. This measure may be a function of the residual value, element sizes, variable value and velocity field rather than just the residual value. In particular, a balance of the integrated sources of the problem, performed for each conserved variable, would provide good guidance to the state of convergence. In a practical problem this is often all that is required.

§7.2.6 Grid Generation

One of the major areas controlling the use of unstructured mesh codes is the ability to easily construct the meshes used by the code. One of the areas not investigated by this study is the use of elements with larger number of faces than the cube. The possibility that numerical diffusion could be reduced by the use of elements with large numbers of faces is of interest[41]. The reason why the area was not covered was that no mesh generator was available which could create meshes consisting of this type of element. Most mesh generators create meshes consisting of tetrahedral, pentahedral and hexahedral elements with occasionally pyramid shaped elements.

The second concern is the quality of the meshes produced from the mesh generators. As part of this study a number of mesh generation packages were used in order to generate the meshes. So far no package used during this research has consistently fulfilled the requirement of being able to generate a highly orthogonal mesh using high order elements (for all packages so far tested this meant hexahedral elements).

Appendix A

Shape Functions

§A.1 Brick Elements

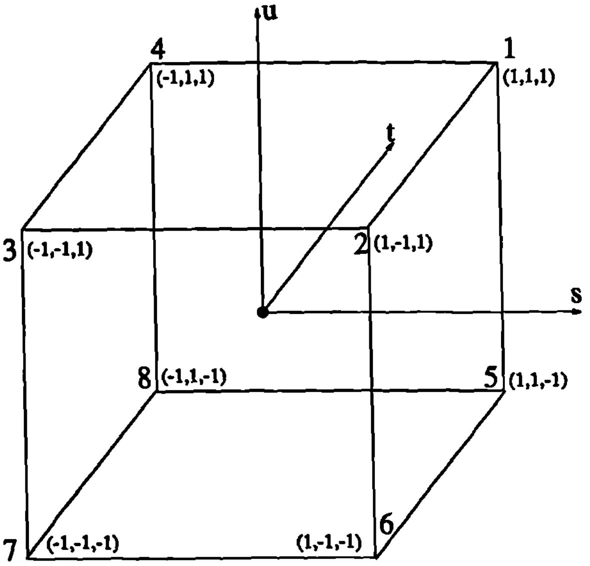

Figure A.1: Local Coordinate Brick Element

$$
\begin{aligned}
N_1(s,t,u) &= 0.125\,(1+s)\,(1+t)\,(1+u) \\
N_2(s,t,u) &= 0.125\,(1+s)\,(1-t)\,(1+u) \\
N_3(s,t,u) &= 0.125\,(1-s)\,(1-t)\,(1+u) \\
N_4(s,t,u) &= 0.125\,(1-s)\,(1+t)\,(1+u) \\
N_5(s,t,u) &= 0.125\,(1+s)\,(1+t)\,(1-u) \\
N_6(s,t,u) &= 0.125\,(1+s)\,(1-t)\,(1-u) \\
N_7(s,t,u) &= 0.125\,(1-s)\,(1-t)\,(1-u) \\
N_8(s,t,u) &= 0.125\,(1-s)\,(1+t)\,(1-u)
\end{aligned}
\tag{A.1}
$$

§A.2 Wedge Elements

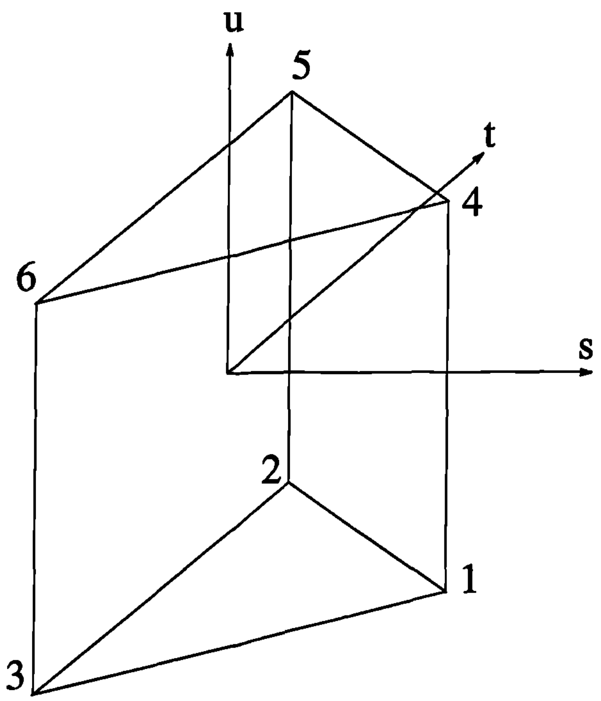

Figure A.2: Local Coordinate Wedge Element

$$
\begin{aligned}
N_1(s,t,u) &= (1+2s)(1-u)/6.0\\
N_2(s,t,u) &= (1-s+\sqrt{3}\,t)(1-u)/6.0\\
N_3(s,t,u) &= (1-s-\sqrt{3}\,t)(1-u)/6.0\\
N_1(s,t,u) &= (1+2s)(1+u)/6.0\\
N_2(s,t,u) &= (1-s+\sqrt{3}\,t)(1+u)/6.0\\
N_3(s,t,u) &= (1-s-\sqrt{3}\,t)(1+u)/6.0
\end{aligned}
\qquad (A.2)
$$

with the vertices located at

1	(1,	0,	-1)
2	(-1/2,	$\sqrt{3}/2$,	-1)
3	(-1/2,	-$\sqrt{3}/2$,	-1)
4	(1,	0,	1)
5	(-1/2,	$\sqrt{3}/2$,	1)
6	(-1/2,	-$\sqrt{3}/2$,	1)

§A.3 Tetrahedral Elements

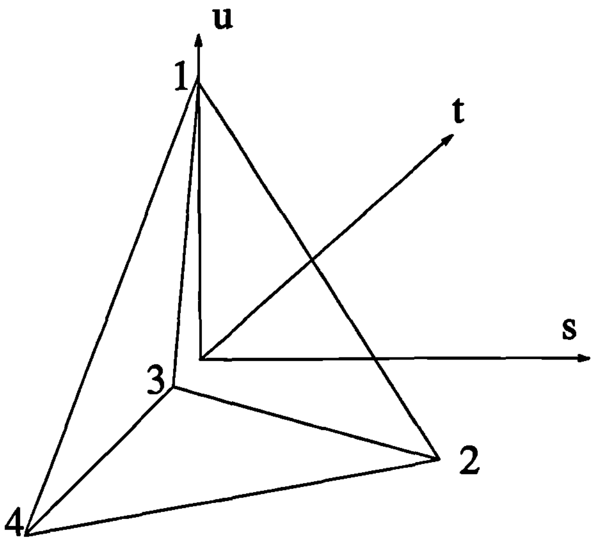

Figure A.3: Local Coordinate Tetrahedral Element

$$\begin{aligned}
N_1(s,t,u) &= (1/4) - (1/3)\,s + t/\sqrt{3} - u/(3.0\,\sqrt{2}) \\
N_2(s,t,u) &= (1/4) - (1/3)\,s - t/\sqrt{3} - u/(3.0\,\sqrt{2}) \\
N_3(s,t,u) &= (1/4) + u/\sqrt{2} \\
N_4(s,t,u) &= (1/4) + (2/3)\,s - u/(3.0\,\sqrt{2})
\end{aligned} \tag{A.3}$$

with the 4 vertices located at

1	(0,	0,	$3/(2\sqrt{2}))$
2	(1,	0,	$-1/(2\sqrt{2}))$
3	(-1/2,	$\sqrt{3}/2$,	$-1/(2\sqrt{2}))$
4	(-1/2,	$-\sqrt{3}/2$,	$-1/(2\sqrt{2}))$

Appendix B

Algorithms for the Linear Solver

This appendix contains the pseudo code for the various solvers discussed in section 3.6. The pseudo code for the ILU(0) decomposition of the matrix being solved is also included. The naming convention used is as follows

$\underline{\underline{A}}$	System Matrix
$\underline{b}$	Source Vector
$\underline{\phi}$	Solution Vector
$\underline{r}$	Residual Vector
$\underline{u}$	Search Direction
$\underline{q}$	Vector forming an orthonormal base for the Krylov subspace
$\underline{\underline{M}}$	Pre-Conditioning Matrix
$\underline{z}$	Solution of the Pre-Conditioning equation $\underline{\underline{M}}\,\underline{z} = \underline{r}$

Any quantity with a tilde, such as $\tilde{\underline{r}}$, is a shadow form of the vector. The superfix on any vector signifies the iteration number of the vector and any component of a vector or matrix will use a suffix indicating the row, and if relevant column, of the component.

§B.1 JOR

```
Select an initial guess $\underline{\phi}^0$
Select over relaxation factor $\omega$
k=0
do
   for all i
      $y_i \;=\; \frac{1}{A_{ii}} \left( b_i \;-\; \sum_{j=1,j\neq i}^{n} A_{i,j}\; \phi_j^k \right)$
   for end
   $\underline{\phi}^{k+1} \;=\; \underline{\phi}^k \;+\; \omega\; (\underline{y} - \underline{\phi}^k)$
   k=k+1
while not converged
```

§B.2 SOR

```
Select an initial guess $\underline{\phi}^0$
Select over relaxation factor $\omega$
k=0
do
   $\underline{y} = \underline{\phi}^k$
   for all i
      $z = y_i$
      $y_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1,j\neq i}^{n} A_{i,j}\, y_j \right)$
      $y_i = z + \omega\, (y_i - z)$
   for end
   $\underline{\phi}^{k+1} = \underline{y}$
   k=k+1
while not converged
```

§B.3 Residual JOR

```
Select an initial guess $\underline{\phi}^0$
Select m and over relaxation factor $\omega$
k=0
do
   $\underline{r} = \underline{b} - \underline{\underline{A}}\, \underline{\phi}^k$
   $\underline{y} = \underline{0}$
   for l from 0 to m-1
      $\underline{w} = \underline{y}$
      for all i
         $y_i = \frac{1}{A_{ii}} \left( r_i - \sum_{j=1,j\neq i}^{n} A_{i,j}\, w_j \right)$
      for end
      $\underline{y} = \underline{w} + \omega(\underline{y} - \underline{w})$
   for end
   $\underline{\phi}^{k+1} = \underline{\phi}^k + \underline{y}$
   k=k+1
while not converged
```

§B.4 Residual SOR

```
Select an initial guess $\underline{\phi}^0$
Select m and over relaxation factor $\omega$
k=0
do
   $\underline{r} = \underline{b} - \underline{\underline{A}}\, \underline{\phi}^k$
```

```
  $\underline{y} = \underline{0}$
  for l from 0 to m-1
     for all i
        $z = y_i$
        $y_i = \frac{1}{A_{ii}} \left( r_i - \sum_{j=1,j\neq i}^{n} A_{i,j} \; y_j \right)$
        $y_i = z + \omega \; (y_i) - z$
     for end
  for end
  $\underline{\phi}^{k+1} = \underline{\phi}^{k} + \underline{y}$
  k=k+1
while not converged
```

§B.5 Pre-Conditioned Conjugate Gradient

```
Select an initial guess $\underline{\phi}^0$
$\underline{r}^0 = \underline{b} - \underline{\underline{A}} \; \underline{\phi}^0$
$\underline{u}^{-1} = \underline{0}$
do
   k=0
   Solve $\underline{\underline{M}} \; \underline{z}^k = \underline{r}^k$
   if k = 0
      $\beta^k = 0$
   else
      $\beta^k = \frac{(\underline{r}^k)^T \; \underline{z}^k}{(\underline{r}^{k-1})^T \; \underline{z}^{k-1}}$
   end if
   $\underline{u}^k = \underline{z}^k + \beta^k \; \underline{u}^{k-1}$
   $\alpha^k = \frac{(\underline{r}^k)^T \; \underline{z}^k}{(\underline{u}^k)^T \; \underline{\underline{A}} \; \underline{u}^k}$
   $\underline{\phi}^{k+1} = \underline{\phi}^k + \alpha^k \; \underline{u}^k$
   $\underline{r}^{k+1} = \underline{r}^k - \alpha^k \; \underline{\underline{A}} \; \underline{u}^k$
while not converged
```

§B.6 Pre-Conditioned BiConjugate Gradient

```
Select an initial guess $\underline{\phi}^0$
$\underline{\tilde{r}}^0 = \underline{r}^0 = \underline{b} - \underline{\underline{A}} \; \underline{\phi}^0$
$\underline{\tilde{u}}^0 = \underline{u}^{-1} = \underline{0}$
do
   k=0
   Solve $\underline{\underline{M}} \; \underline{z}^k = \underline{r}^k$
   Solve $\underline{\underline{M}}^T \; \underline{\tilde{z}}^k = \underline{\tilde{r}}^k$
   if k = 0
```

```
        $\beta^k = 0$
    else
        $\beta^k = \dfrac{(\underline{\tilde{r}}^k)^T\ \underline{z}^k}{(\underline{\tilde{r}}^{k-1})^T\ \underline{z}^{k-1}}$
    end if
    $\underline{u}^k = \underline{z}^k + \beta^k\ \underline{u}^{k-1}$
    $\underline{\tilde{u}}^k = \underline{\tilde{z}}^k + \beta^k\ \underline{\tilde{u}}^{k-1}$
    $\alpha^k = \dfrac{(\underline{r}^k)^T\ \underline{z}^k}{(\underline{u}^k)^T\ \underline{\underline{A}}\ \underline{u}^k}$
    $\underline{\phi}^{k+1} = \underline{\phi}^k + \alpha^k\ \underline{u}^k$
    $\underline{r}^{k+1} = \underline{r}^k - \alpha^k\ \underline{\underline{A}}\ \underline{u}^k$
    $\underline{\tilde{r}}^{k+1} = \underline{\tilde{r}}^k - \alpha^k\ \underline{\underline{A}}\ \underline{\tilde{u}}^k$
    k=k+1
while not converged
```

§B.7 BiCGStab(l)

```
Select an initial guess $\underline{\phi}^0$
$\underline{\hat{r}}^0 = \underline{\tilde{r}} = \underline{r}^0 = \underline{b} - \underline{\underline{A}}\ \underline{x}^0$
k=0
$\rho = 1 \quad \omega = 1 \quad \alpha = 0$
$\underline{\hat{\phi}}^0 = \underline{\phi}^0$
do
    $\rho = -\omega\ \rho_0$
    for j from 0 to l-1
        $\rho_1 = (\underline{\tilde{r}})^T\ \underline{\hat{r}}^j$
        $\beta = \alpha\ \rho_1\ /\ \rho$
        $\rho = \rho_1$
        for i from 0 to j
            $\underline{\hat{u}}^i = \underline{\hat{r}}^i - \beta\ \underline{\hat{u}}^i$
        for end
        $\underline{\hat{u}}^{j+1} = \underline{\underline{A}}\ \underline{\hat{u}}^j$
        $\gamma = (\underline{\hat{u}}^{j+1})^T\ \underline{\tilde{r}}$
        $\alpha = \rho\ /\ \gamma$
        for i from 0 to j
            $\underline{\hat{r}}^i = \underline{\hat{r}}^i - \alpha\ \underline{\hat{u}}^i$
        for end
        $\underline{\hat{r}}^{j+1} = \underline{\underline{A}}\ \underline{\hat{r}}^j$
        $\underline{\hat{\phi}}^0 = \underline{\hat{\phi}}^0 + \alpha\ \underline{\hat{u}}^0$
    for end

    for j from 1 to l
        $\underline{q}^j = \underline{\hat{r}}^j$
        for i from 1 to j-1
```

```
            $\tau_{ij} = \frac{1}{\sigma_j}(\underline{q}^j)^T \; \underline{q}^i$
            $\underline{q}^j = \underline{q}^j - \tau_{ij} \; \underline{q}^i$
        for end
        $\sigma_j = (\underline{q}^j)^T \; \underline{q}^j$
        $\gamma_j^1 = \frac{1}{\sigma_j}(\hat{\underline{r}}^0)^T \; \underline{q}^j$
    for end

    $\gamma_l = \gamma_l^1$
    $\omega = \gamma_l$
    for j down from l-1 to 1
        $\gamma_j = \gamma_j^1 - \sum_{i=j+1}^{l} \tau_{ji}\gamma_i$
    for end
    for j from 1 to l-1
        $\gamma_j^2 = \gamma_{j+1} + \sum_{i=j+1}^{l-1} \tau_{ji}\gamma_{i+1}$
    for end
    $\hat{\underline{\phi}}^0 = \hat{\underline{\phi}}^0 + \gamma_1\hat{\underline{r}}^0$
    $\hat{\underline{r}}^0 = \hat{\underline{r}}^0 - \gamma_l^1\hat{\underline{r}}^l$
    $\hat{\underline{u}}^0 = \hat{\underline{u}}^0 - \gamma_l\hat{\underline{u}}^l$
    for j from 1 to l-1
        $\hat{\underline{\phi}}^0 = \hat{\underline{x}}^0 + \gamma_j^2\hat{\underline{r}}^j$
        $\hat{\underline{r}}^0 = \hat{\underline{r}}^0 - \gamma_j^1\hat{\underline{r}}^j$
        $\hat{\underline{u}}^0 = \hat{\underline{u}}^0 - \gamma_j\hat{\underline{u}}^j$
    for end
    k=k+1
    $\underline{u}^{k-1} = \hat{\underline{u}}^0$
    $\underline{r}^k = \hat{\underline{r}}^0$
    $\underline{\phi}^k = \hat{\underline{\phi}}^0$
while not converged
```

§B.8 Pre-Conditioned GMRES(k,m)

```
Select an initial guess $\underline{\phi}$
$\underline{r} = \underline{b} - \underline{\underline{A}} \; \underline{\phi}$
do
    Solve $\underline{\underline{M}} \; \underline{z} = \underline{r}$
    $\underline{q}^1 = \underline{z} \; / \; \|\underline{z}\|_2$
    for k from 1 to m
        Solve $\underline{\underline{M}} \; \underline{w} = \underline{\underline{A}} \; \underline{q}^k$
        for j from 1 to k
            $h_{j,k} = (\underline{w})^T \; \underline{q}^j$
            $\underline{w} = \underline{w} - h_{j,k} \; \underline{q}^j$
        for end
        $h_{k+1,k} = \|\underline{w}\|_2$
        $\underline{q}^{k+1} = \underline{w} \; / \; h_{k+1,k}$
```

```
  for end

  for k from 2 to m+1
     $\sigma_j = 0$
  for end
  $\sigma_1 = \|\underline{z}\|_2$

  for k from 1 to m
     $d = \sqrt{h^2_{k,k} + h^2_{k+1,k}}$
     $s = h_{k+1,k} / d$
     $c = h_{k,k} / d$
     $\sigma_{k+1} = -s \sigma_k$
     $\sigma_k = c \sigma_k$
     for j from k to m
        $t = c h_{k,j} + s h_{k+1,j}$
        $h_{k+1,j} = -s h_{k,j} + c h_{k+1,j}$
        $h_{k,j} = t$
     for end
  for end

  for j down from m to 1
     for k from j+1 to m
        $\sigma_j = \sigma_j - h_{j,k} \sigma_k$
     for end
     $\sigma_j = \sigma_j / h_{j,j}$
  for end

  $\underline{\phi} = \underline{\phi} + \sum_i^m \sigma_i \underline{q}^i$
  $\underline{r} = \underline{b} - \underline{\underline{A}} \underline{\phi}$
while not converged
```

§B.9 Incomplete Lower Upper Decomposition(0)

```
$\underline{\underline{M}} = \underline{\underline{A}}$
for i from 1 to n
   for k from 1 to i-1
      if $M_{i,k} \neq 0$
         $t = M_{i,k} M_{k,k}$
         $M_{i,k} = t$
         if $M_{k,i} \neq 0$
            $M_{i,i} = M_{i,i} - t M_{k,i}$
         end if
         for j from k+1 to n, j not equal to i
            if $M_{i,j} \neq - 0$
               $M_{i,j} = M_{i,j} - t M_{k,j}$
```

```
        end if
      for end
    end if
  for end
  $M_{i,i}$ = 1.0 / $M_{i,i}$
for end
```

Appendix C

Flow Diagram of the CWNN Code

- Read configuration file (called INFORM)
- Read geometry file.
- If necessary read restart database.
- Initialise variables
- Time Step Loop
    - Copy current values into old storage.
    - Outer Iteration Loop
        - Solve for Enthalpy
        - Solve for Radiosity
        - Solve for Chemical Species
        - Solve for Turbulence (k, ϵ)
        - Update Calculated Values (Species Mass Fractions, Turbulent Viscosity, Material Properties)
        - Solve for momentum.
        - Solve pressure correction equation.
        - Correct pressure and momentum fields.
        - Calculate convection fluxes, G_k, skin friction.
        - Test for Convergence
        - Display Monitor/Residual values.
        - Calculate the droplet tracks.
        - Calculate sources due to droplets.

- End of Outer Iteration Loop
- End of Time Step Loop
- Write restart database.
- Write result files.

References

[1] D. F. G. Durão, M. V. Heitor and A. L. N. Moreira, 'Flow Measurements in a Model Burner - Part 1', *Transactions of the ASME, Journal of Fluids Engineering*, **113**, 668–674 (1991).

[2] D. F. G. Durão, M. V. Heitor and A. L. N. Moreira, 'On the Stabilization of Flames on Multijet Industrial Burners', *Experimental Thermal and Fluid Science*, **5**, 736–746 (1992).

[3] D. F. G. Durão, M. V. Heitor and A. L. N. Moreira, 'Flow Measurements in a Model Burner - Part 2', *Transactions of the ASME, Journal of Fluids Engineering*, **115**, 309–316 (1993).

[4] J. Görres, U. Schnell and K. R. G. Hein, 'Trajectories of Burning Coal Particles in Highly Swirling Reactive Jets', *International Journal of Heat and Fluid Flow*, **15**, 440–450 (1995).

[5] M. Kashiwaya, K. Nakagawa and Y. Oshamoto, 'Fine Atomization Fuel Injector', *Hitachi Review*, **39(5)**, 289–294 (1990).

[6] H. Gou, G. A. Irons and W.-K. Lu, 'Mathematical Modelling of Postcombustion in a KOBM Converter', *Metallurgical Transactions B*, **24B(2)**, 179–188 (1993).

[7] T. O'Doherty, D. Froud, J. Bates and N. Syred, 'Characteristics of a Power Station', *Proceedings of the Institution of Mechanical Engineers*, **208**, 89–101 (1994).

[8] J. P. Moodie, M. P. Davis and M. Cross, 'Numerical Modelling for the Analysis of Direct Smelting Processes', in *7th Process Technology Conference Proceedings*, 55–64 (1988).

[9] G. J. Hardie, M. Cross, R. J. Batterham, M. P. Davis and M. P. Schwarz, 'The Role of Mathematical Modelling in the HIsmelt^{TM} Process', in *10th Process Technology Conference Proceedings*, 109–121 (1992).

[10] M. P. Davis, K. Pericleous, M. P. Schwarz and M. Cross, 'Mathematical Modelling Tools for the Optimisation of Direct Smelting Processes', in *Proceedings of the International Conference on CFD in Minerals and Metals Processing and Power Generation* (1997).

[11] M. P. Schwarz and P. Dang, 'Simulations of Blowthrough in Smelting Baths with Bottom Gas Injection', in *13th Process Technology Conference Proceedings*, 415–422 (1995).

[12] M. P. Schwarz, 'Simulation of Gas Injection into Liquid Melts', *Applied Mathematical Modelling*, **20**, 41–51 (1996).

[13] 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).

[14] 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).

[15] S. Whitaker, 'Advances in Theory of Fluid Motion in Porous Media', *Industrial and Engineering Chemistry*, **61(12)**, 14–28 (1969).

[16] W. J. Gordon and C. A. Hall, 'Construction of Curvilinear Co–ordinate Systems and Applications to Mesh Generation', *International Journal for Numerical Methods in Engineering*, **7**, 461–477 (1973).

[17] 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 Arbitrary Two–Dimensional Bodies', *Journal of Computational Physics*, **15**, 299–319 (1974).

[18] 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).

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

[20] 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).

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

[22] I. Demirdžić and M. Perić, '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).

[23] 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).

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

[25] 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).

[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] A. M. Winslow, 'Numerical Simulation of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh', *Journal of Computational Physics,* **1(2)**, 149–171 (1966).

[28] B. R. Baliga and S. V. Patankar, 'A New Finite–Element Formulation for Convection–Diffusion Problems', *Numerical Heat Transfer,* **3**, 393–409 (1980).

[29] 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).

[30] G. E. Schneider and M. J. Raw, 'Control Volume Finite–Element Method for Heat Transfer and Fluid Flow Using Colocated Variables - 1. Computational Procedure', *Numerical Heat Transfer,* **11(4)**, 363–390 (1987).

[31] R. D. Lonsdale, 'The ASTEC Code: An Algorithm for Solving Thermal–Hydraulic Equations in Complex Geometries', *Nuclear Science and Engineering,* **100**, 385–392 (1988).

[32] A. Jameson, T. J. Baker and N. P. Weatherill, 'Calculation of Inviscid Transonic Flow over a Complete Aircraft', *AIAA*, Paper AIAA-86-0103 (1986).

[33] P. Chow and M. Cross, 'An Enthalpy Control Volume–Unstructured Mesh (CV–UM) Algorithm for Solidification by Conduction Only', *International Journal for Numerical Methods in Engineering,* **35(9)**, 1849–1870 (1992).

[34] 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', *Applied Mathematical Modelling,* **15**, 639–645 (1991).

[35] J. A. Shaw, A. J. Peace and N. P. Weatherill, 'A Three–Dimensional Hybrid Structured–Unstructured Method: Motivation, Basic Approach and Initial Results', *Institute of Mathematics and its Applications Conference Series,* **44**, 157-201 (1994).

[36] P. N. Childs, J. A. Shaw, A. J. Peace and J. M. Georgala, 'SAUNA: A System for Grid Generation and Flow Simulation using Hybrid/Structured/Unstructured Grids', in *Computational Fluid Dynamics, Proceedings of the 1^{st} European CFD Conference,* Volume 2, 875–882 (1992).

[37] 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).

[38] C. Bailey, Y. D. Fryer, M. Cross and P. Chow, 'Predicting the Deformation of Castings in Moulds using a Control Volume Approach on Unstructured Meshes', in *Mathematical Modelling for Material Processing* (1993).

[39] G. T. Tomaich and P. L. Roe, 'Compact Schemes for Advection–Diffusion Problems on Unstructured Grids', *Modelling and Simulation*, **23(5)**, 2629–2636 (1993).

[40] H. Deconinck, R. Struijs and P. L. Roe, 'Fluctuation Splitting for Multidimensional Convection Problems: An Alternative to Finite Volume and Finite Element Methods', in *Computational Fluid Dynamics*, Volume 2 of *Lecture Series 1990–03.* von Karman Institute for Fluid Dynamics (1990).

[41] P. M–Y. Chow, 'Control Volume Unstructured Mesh Procedure for Convection-Diffusion Solidification Process', Ph.D. Thesis, University of Greenwich, London (1993).

[42] T. N. Croft, 'Software Design for Unstructured Mesh–Control Volume Analysis Codes in Three Dimensions', Master's Thesis, School of Mathematics, Statistics and Computing, University of Greenwich, London (1992).

[43] 'CWNN (Code With No Name)', In House Code, University of Greenwich.

[44] C. M. Rhie and W. L. Chow, 'Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation', *AIAA Journal*, **21(11)**, 1525–1532 (1983).

[45] D. B. Spalding, *GENMIX: A General Computer Program for Two-Dimensional Parabolic Phenomena.* Pergamon Press (1977).

[46] J. C. Slattery, *Momentum, Energy and Mass Transfer in Continua.* Robert E. Kreiger Publishing Company, Huntingdon, New York, Second Edition (1981).

[47] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow.* McGraw–Hill, New York (1980).

[48] R. Courant, E. Isaacson and M. Rees, 'On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Difference', *Communications in Pure and Applied Mathematics*, **5**, 243–255 (1952).

[49] G. D. Raithby, 'A Critical Evaluation of Upstream Differencing Applied to Problems Involving Fluid Flow', *Computer Methods in Applied Mechanics and Engineering*, **9**, 75–103 (1976).

[50] D. B. Spalding, 'A Novel Finite Difference Formulation for Differential Expressions Involving both First and Second Derivatives', *International Journal for Numerical Methods in Engineering*, **4**, 551–559 (1972).

[51] S. V. Patankar, 'A Calculation Procedure for Two Dimensional Elliptic Situations', *Numerical Heat Transfer*, **2** (1979).

[52] B. P. Leonard, A. P. Lock and M. K. MacVean, 'Extended Numerical Integration for Genuinely Multidimensional Advective Transport Insuring Conservation', in *Numerical Methods in Laminar and Turbulent Flows, Proceedings of the Ninth International Conference*, Volume 9, Part 1, 1–12 (1995).

[53] A. Staniforth and J. Côté, 'Semi–Lagrangian Integration Schemes for Atmospheric Models – A Review', *Monthly Weather Review*, **119**, 2206–2223 (1991).

[54] N. P. Waterson and H. Deconinck, 'A Unified Approach to the Design and Application of Bounded Higher–Order Convection Schemes', in *Numerical Methods in Laminar and Turbulent Flows, Proceedings of the Ninth International Conference*, Volume 9, Part 1, 203–214 (1995).

[55] 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(12)**, 2182–2189 (1965).

[56] 'Flow3d, Version 2.3.2', AEA Technology. Harwell, England.

[57] 'CFX4, Version 4.1', AEA Technology. Harwell, England.

[58] N. A. Shore, B. S. Haynes, D. F. Fletcher and A. A. Sola, 'Numerical Aspects of Swirl Flow Computation', in *Computational Techniques and Applications: CTAC95* (1995).

[59] R. B. Bird, W. E. Stewart and E. N. Lightfoot, *Transport Phenomena.* Wiley International Edition (1960).

[60] H. I. Rosten, D. B. Spalding and D. G. Tatchell, 'PHOENICS: A General–Purpose Program for Fluid–Flow, Heat–Transfer and Chemical–Reaction Processes', in *Proceedings of the 3^{rd} International Conference on Engineering Software*, 639–655 (1983).

[61] B. P. Leonard, 'A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation', *Computer Methods in Applied Mechanics and Engineering*, **19**, 59–98 (1979).

[62] N. C. Markatos and M. K. Patel, 'Evaluation of Eight Discretisation Schemes for Two- Dimensional Convection-Diffusion Equations', *International Journal for Numerical Methods in Fluids*, **6(3)**, 129–154 (1986).

[63] A. Pollard and A. L.-W. Siu, 'The Calculation of some Laminar Flows using Various Discretisation Schemes', *Computer Methods in Applied Mechanics and Engineering*, **35**, 293–313 (1982).

[64] P. H. Gaskell and A. K. C. Lau, 'Curvature-Compensated Convective Transport: SMART, A New Boundedness Preserving Transport Algorithm', *International Journal for Numerical Methods in Fluids*, **8**, 617–641 (1988).

[65] U. Ghia, K. N. Ghia and C. T. Shin, 'High–Re Solutions for Incompressible Flow Using the Navier–Stokes Equations and a Multigrid Method', *Journal of Computational Physics*, **48**, 387–411 (1982).

[66] M. R. Hestenes, 'Iterative Methods for Solving Linear Equations', *Journal of Optimization Theory and Applications*, **11(4)**, 323–334 (1973).

[67] E. Stiefel, 'Über Einige Methoden der Relaxationsrechnung', *Zeitschrift für Angewandte Mathematik und Physik*, **3(1)**, 1–33 (1952).

[68] M. R. Hestenes and E. Stiefel, 'Methods of Conjugate Gradients for Solving Linear Systems', *Journal of Research of the National Bureau of Standards*, **49(6)**, 409–436 (1952).

[69] R. Fletcher, 'Conjugate Gradient Methods for Indefinite Systems', *Lecture Notes in Mathematics*, **506**, 73–89 (1976).

[70] C. Lanczos, 'Solutions of Systems of Linear Equations by Minimized Iterations', *Journal Res. Nat. Bur. Stand*, **49**, 33–53 (1952).

[71] R. W. F. amd M. Gutknecht and N. M. Nachtigal, 'An Implementation of the Look-Ahead Lanczos Algorithm for Non-Hermitian Matrices', *SIAM, Journal of Scientific Statistical Computing*, **14(1)**, 137–158 (1993).

[72] B. N. Parlett, D. R. Taylor and Z. A. Liu, 'A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices', *Mathematics of Computation*, **44(169)**, 105–124 (1985).

[73] P. Sonneveld, 'CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems', *SIAM Journal of Scientific Statistical Computing*, **10**, 36–52 (1989).

[74] H. A. van der Vorst, 'Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems', *SIAM, Journal of Scientific Statistical Computing*, **13(2)**, 631–644 (1992).

[75] M. H. Gutknecht, 'Variants of BiCGStab for Matrices with Complex Spectrum', Technical Report 91–14, IPS Research Report (1991).

[76] G. L. G. Sleijpen and D. R. Fokkema, 'BICGSTAB(l) for Linear Equations Involving Unsymmetric Matrices with Complex Spectrum', *Electronic Transactions on Numerical Analysis*, **1**, 11–32 (1993).

[77] Y. Saad and M. H. Schultz, 'GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems', *SIAM, Journal of Scientific Statistical Computing*, **7(3)**, 856–869 (1986).

[78] D. B. Spalding, 'Proposal for a Diffusional Radiation Model', Unpublished Technical Memorandum, CHAM, London.

[79] K. G. Wilson, 'Renormalization Group and Critical Phenomena. I Renormalization Group and the Kandoff Scaling Picture', *Physics Review B*, **4(9)**, 3174–3183 (1971).

[80] K. G. Wilson, 'Renormalization Group and Critical Phenomena. II Phase–Space Cell Analysis of Critical Behaviour', *Physics Review B*, **4(9)**, 3184–3205 (1971).

[81] D. B. Spalding, 'A New Model of Turbulent Combustion', Technical Report HTS/76/10, Mechanical Engineering Department, Imperial College, London (1976).

[82] D. B. Spalding, 'The Influences of Laminar Transport and Chemical Kinematics on the Time–Mean Reaction Rate in a Turbulent Flame', in *Proceedings of the 17^{th} Combustion Symposium*, 431–440 (1976).

[83] N. C. Markatos and K. A. Pericleous, 'An Investigation of Three–Dimensional Fires in Enclosures', *American Society of Mechanical Engineers, Heat Transfer Division (Publication) HTD*, **25**, 115-124 (1983).

[84] R. Viskanta and M. P. Mengüç, 'Radiation Heat Transfer in Combustion Systems', *Progress in Energy and Combustion Science*, **13(2)**, 97–160 (1987).

[85] J. R. Howell, 'Thermal Radiation in Participating Media: The Past, the Present and Some Possible Futures', *Transactions of the ASME, Journal of Heat Transfer*, **110**, 1220–1229 (1988).

[86] H. C. Hamaker, 'Radiation and Heat Conduction in Light–Scattering Material: Part I Reflection and Transmission', *Philips Research Report*, **2**, 55–67 (1947).

[87] H. C. Hamaker, 'Radiation and Heat Conduction in Light-Scattering Material: Part II', *Philips Res. Rep.*, **2**, 103 (1947).

[88] A. Schuster, 'Radiation Through a Foggy Atmosphere', *The Astrophysical Journal*, **21(1)**, 1–22 (1905).

[89] A. D. Brent, V. R. Voller and K. J. Reid, 'Enthalpy–Porosity Technique for Modelling Convection–Diffusion Phase Change: Application to the Melting of a Pure Metal', *Numerical Heat Transfer*, **13**, 297–318 (1988).

[90] M. Cross, S. Johnson and P. Chow, 'Mapping Enthalpy–Based Solidification Algorithms onto Vector and Parallel Architectures', *Applied Mathematical Modelling*, **13**, 702–709 (1989).

[91] V. R. Voller and C. Prakash, 'A Fixed Grid Numerical Modelling Methodology for Convection–Diffusion Mushy Region Phase–Change Problems', *International Journal of Heat and Mass Transfer*, **30(8)** (1987).

[92] C. R. Swaminathan, 'Fixed Grid Approaches for Modelling Conduction Controlled Solidification Phase Change Problems', Ms report, University of Minnesota, Minneapolis (1990).

[93] V. R. Voller and C. R. Swaminathan, 'General Source–Based Method for Solidification Phase Change', *Numerical Heat Transfer, Part B*, **19(2)**, 175–189 (1991).

[94] J. Boussinesq, 'Theorie de L'Ecoulement Toubillant', *Mem. Acad. Sci.*, **23(46)** (1877).

[95] W. Rodi, ***Turbulence Models and their Application in Hydraulics.*** Delft International Institute for Hydraulic Research (1980).

[96] B. E. Launder and D. B. Spalding, 'The Numerical Computation of Turbulent Flows', *Computer Methods in Applied Mechanics and Engineering*, **3**, 269–289 (1974).

[97] M. Nallasamy, 'Turbulence Models and Their Application to the Prediction of Internal Flows: A Review', *Computers and Fluids*, **15(2)**, 151–194 (1987).

[98] V. Yakhot and S. A. Orszag, 'Renormalization Group Analysis of Turbulence. I. Basic Theory', *Journal of Scientific Computing*, **1(1)**, 3–51 (1986).

[99] V. Yakhot, S. A. Orszag, S. Thangam, T. B. Gatski and C. G. Speziale, 'Developments of Turbulence Models for Shear Flows by a Double Expansion Technique', *Physics of Fluids A*, **4(7)**, 1510–1520 (1992).

[100] P. W. C. Firth and R. K. Duggins, 'Turbulence Modelling of Swirling Flows', in *Numerical Methods in Laminar and Turbulent Flow, Proceedings of the Fourth International Conference*, 353–363 (1985).

[101] H. Schlichting, *Boundary–Layer Theory.* McGraw Hill, 4th Edition (1960).

[102] N. I. Kolev, 'Fragmentation and Coalescence Dynamics in Multiphase Flows', *Experimental Thermal and Fluid Science*, **6**, 211–251 (1993).

[103] I. P. Jones, 'A Comparison Problem for Numerical Methods in Fluid Dynamics, The 'Double–Glazing' Problem', in *Numerical Methods in Thermal Problems, Proceedings of the First International Conference*, 338–348 (1979).

[104] 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).

[105] S. J. Kline, B. J. Cantwell and G. M. Lilley, editors, *AFOSR–HTTM–Sanford Conference, Complex Turbulent Flows*, Volume I, II and III (1980–1981).

[106] 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).

[107] J. K. Eaton and J. P. Johnston, 'A Review of Research on Subsonic Turbulent Flow Reattachment', *AIAA*, Paper AIAA–80–1438 (1980).

[108] M. Nallasamy and C. P. Chen, 'Studies on Effects of Boundary Conditions in Confined Turbulent Flow Predictions', Technical Report CR-3929, NASA (1985).

[109] S. A. Syed and L. M. Chiappetta, 'Finite Difference Methods for Reducing Numerical Diffusion in TEACH–Type Calculations', *AIAA*, Paper AIAA–85–0056 (1985).

[110] TECHNICAL MEMORANDUM, 'Introduction to the ReNormalization Group Method and Turbulence Modeling', Technical Report TM-107, Fluent INC (1993).

[111] B. L. Cusack, G. J. Hardie and P. D. Burke, 'HIsmelt – 2nd Generation Direct Smelting', in *Second European Ironmaking Conference* (1991).

[112] B. R. Clayton and Y. S. M. Morsi, 'Determination of Principal Characteristics of Turbulent Swirling Flow along Annuli, Part 1: Measurement of Time Mean Parameters', *International Journal of Heat and Fluid Flow*, **5(4)**, 195–203 (1984).

[113] B. R. Clayton and Y. S. M. Morsi, 'Determination of Principal Characteristics of Turbulent Swirling Flow along Annuli, Part 2: Measurement of Turbulent Components', *International Journal of Heat and Fluid Flow*, **6(1)**, 31–41 (1985).

[114] Y. S. M. Morsi and B. R. Clayton, 'Determination of Principal Characteristics of Turbulent Swirling Flow along Annuli, Part 3: Numerical Analysis', *International Journal of Heat and Fluid Flow*, **7(3)**, 208–222 (1986).

[115] Unpublished Experimental Data, '1/3rd Scale, 60kPa Test (Reference Number CL1C) Run by CSIRO at Clayton'.

[116] 'MSC/Patran Version 5.0', MacNeil Schwendler Corporation, Los Angeles, USA.

[117] I. F. Taylor, 'Gas Blowthrough of Liquid Metal Bath by and Upward Directed Submerged Jet', *ISIJ International*, **33(7)**, 748–756 (1993).

[118] P. T. L. Koh and R. J. Batterham, 'Liquid Splashes from Submerged Gas Injection', *Chemical Engineering Research and Design*, **67**, 211–215 (1989).

[119] J. L. T. Azevedo and J. C. F. Pereira, 'Modeling of Multisizes Particle Laden Turbulent Low Swirling Free Jets', *Journal of Wind Engineering and Industrial Aerodynamics*, **46 & 47**, 713–720 (1993).

[120] K. Nishimura, S. Tashiro and H. Yoshiki, 'Velocity Distribution of Swirl Flow in a Circular Cylinder', in *Institution of Mechanical Engineers, Computers in Engine Technology, International Conference*, 63–68 (1991).