Unstructured Finite Volume Algorithms for Compressible Multiphase Flow

Gerard Serge Bruno Lebon

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

August 2011

DECLARATION

I certify that this work has not been accepted in substance for any degree, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigations except where otherwise identified by references and that I have not plagiarised the work of others.

....

Bruno Lebon (PhD Student)

Date:

....

Pr Koulis Pericleous (Supervisor)

Date:

....

Dr Mayur Patel (Supervisor)

Date:

....

Dr Georgi Djambazov (Supervisor)

Date:

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors Professor Koulis Pericleous and Dr Mayur Patel for their advice, encouragement and guidance throughout my PhD. I would like to thank Dr Georgi Djambazov for helping me to get started with PHYSICA, and valuable support while using the CFD package. I wish to mention Darren Wise, the Unix support guy, who does a fantastic job maintaining the Linux cluster, on which most of the cases presented in this thesis have been run.

I would like to thank my parents for their support during these years. I wish to thank my fiancée Caroline for her encouragement, understanding and support during this overseas study. I also thank my in-laws for the encouragement they provided me with.

I would like to acknowledge the School of Computing and Mathematical Sciences which provided the funding for my bursary.

Last, but not least, I thank God for His everlasting support in the difficult times I have been through during these last years.

ABSTRACT

This research presents novel algorithms for computing flow within an unstructured, collocated, finite volume solver in the presence of non-orthogonality and compressibility in order to extend the range of problems which can be modelled with the University's in-house CFD code: PHYSICA.

A new non-orthogonality diffusion correction relaxation parameter has been successfully introduced and tested with benchmarks from the literature. Cases involving geometries meshed with commercial packages have been successfully run with the diffusion correction methods, variable bounding and proper under-relaxation practices. The applicability of a pressure interpolation method has also been tested with these cases.

A procedure for solving compressible flow within a finite volume, pressure correction type scheme, has been devised and successfully implemented in different test cases. This method is however prone to numerical diffusion in the presence of shocks, but does work even in the presence of skewed meshes. The method was then tested with the case of an oxygen jet entering a heated furnace, for which experimental data is available for comparison. The method was successful in predicting the axial variables of the jet, and used to develop a turbulence modification model for such jets.

The method was finally used to model the deformation of a free surface impinged by a compressible jet, using a novel zonal method called zonal Gas And Liquid Analyser (GALA). Convergence was achieved with the

method developed in this research, together with the application of the counter diffusion method to model the moving interface.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

Abbreviations

CDM	Counter Diffusion Method
CFD	Computational Fluid Dynamics
CVFEM	Control-Volume Finite-Element Method
FEM	Finite Element Method
GALA	Gas and Liquid Analyser
LES	Large Eddy Simulation
RANS	Reynolds-Averaged Navier Stokes
SEA	Scalar Equation Algorithm
SIMPLE	Semi-implicit Method for Pressure-Linked Equations
SIMPLEC	SIMPLE-Consistent
SIMPLER	SIMPLE Revised
TVD	Total Variation Diminishing
VOF	Volume of Fluid

Greek Symbols

α	Relaxation constant
Γ	Diffusion coefficient

γ	Specific heat capacities ratio	1.4
κ	Heat conductivity	$\mathrm{W\,m^{-1}\,K^{-1}}$
λ	Surface tension	$\mathrm{N\,m^{-1}}$
μ	Dynamic viscosity	$\mathrm{Pa\,s}$
μ_t	Turbulent dynamic viscosity	$\mathrm{Pa\,s}$
ν	Kinematic viscosity	$\mathrm{m^2\,s^{-1}}$
ν_t	Kolmogorov-Prandtl expression for turbulent kinematic viscosity	$\mathrm{m^2\,s^{-1}}$
Φ	Free surface variable	
ϕ	Conserved variable	
ρ	Density	$\mathrm{kg\,m^{-3}}$
σ_t	Turbulent Prandtl number	
θ	Non-orthogonal angle	$^\circ$
Υ	Residual	
ε	Turbulent kinetic energy dissipation rate	$\mathrm{m^2\,s^{-3}}$
ς	Non-orthogonality diffusion correction relaxation parameter	
ξ	Vector joining to adjacent cell centres	

Roman Symbols

Pe	Peclet number	
Pr	Nusselt number	
Ra	Rayleigh number	
Re	Reynolds number	
A	Area	$\mathrm{m^2}$
a	Speed of sound	$\mathrm{m\,s^{-1}}$

NOMENCLATURE

$a_{P,\Lambda,N,E,S,W}$	Linear equations coefficients	
B	Body force	N
$C_1, C_\mu, C_{2\varepsilon}, C_3$	Dimensionless constants for turbulent viscosity calculation	
C_v	Specific heat at constant volume	$\mathrm{J\,kg^{-1}\,K^{-1}}$
D	Diffusion conductance	
F	Strength of convection	
G	Turbulent generation rate	
g	Acceleration due to gravity	$\mathrm{m\,s^{-2}}$
G_b	Turbulent generation due to buoyancy	
h	Specific enthalpy	$\mathrm{J\,kg^{-1}}$
J	Jacobian	
k	Turbulent kinetic energy generation rate	$\mathrm{m^2\,s^{-2}}$
M	Mach number	
m	Mass	kg
p	Pressure	Pa
R	Ideal gas constant	$8.314472\ \mathrm{J\,K^{-1}\,mol^{-1}}$
S_ϕ	Source term for ϕ conservation equation	
T	Temperature	K
t	Time	s
u, v, w	Velocity vector components in the x, y and z directions	$\mathrm{m\,s^{-1}}$
V	Volume	$\mathrm{m^3}$
$\mathbf{n}$	Normal vector	
$\mathbf{u}$	Velocity vector	$\mathrm{m\,s^{-1}}$

Subscripts

f	General face value
n, e, s, w	Face values
P, Λ, N, E, S, W	Value at respective nodes
T	Total conditions
t	Turbulent value
U	Upwind value
UU	Upwind-upwind value

Chapter 1

INTRODUCTION

1.1 Research objectives

The main objective of this PhD study is to develop stable numerical methods to handle compressibility and non-orthogonality in finite volume computations of engineering processes involving fluid flow, both single phase and involving two phases separated by a free surface. Current pressure-correction type finite volume formulations as used in the in-house multiphysics package PHYSICA [15] developed by Chow [16] and Croft [17] suffer from solution instabilities with skewed meshes generated with commercial meshing packages. No compressibility framework was available in the package at the beginning of the research.

Such behaviour from a multi-physics code is not acceptable, since it restricts the range of engineering problems that can be tackled. New algorithms, building on the previous research at Greenwich with respect to finite volume code development [16; 17; 18; 19; 20], their application to free surface deformation [21] and published research from the literature are proposed to

1. analyse the non-orthogonality implementation of the code and improve on the code to tackle divergence issues in non-orthogonal geometries whose mesh have been generated using a commercial package,

2. introduce a compressibility computation framework to handle supersonic flows and
3. handle free surface modelling of an incompressible liquid impinged upon by a compressible gas, while handling density variations in the compressible gas at the same time.

This research begins with a literature review of the state of the art non-orthogonality handling and compressibility codes, and sifting methods applicable to a finite volume, collocated scheme. For skewness investigation, particular attention is paid to the resolution of face diffusion flux and the effect of non-orthogonality on pressure correction term, which is prone to divergence.

For compressibility, the pressure correction method is revised to account for density variation, and the turbulence model is modified to account for compressible effects. The pressure-correction procedure is also modified in the case of a compressible jet impinging on a liquid surface, to handle mass conservation in two separate media.

These selected methods are then tested and validated against either analytic solutions, benchmarks from the literature and experimental data. The results of these investigation are presented in the thesis, and the conclusions drawn from these test cases will present a series of advices and new procedures on how to tackle non-orthogonality and compressibility in a collocated, pressure-correction type finite volume code.

1.2 Introduction to the finite volume method

1.2.1 Governing equations

The transport equation for a conserved quantity ϕ is given by

$$\frac{\partial\left(\rho\phi\right)}{\partial t}+\nabla\cdot\left(\rho\boldsymbol{u}\phi\right)=\nabla\cdot\left(\Gamma\nabla\phi\right)+S_{\phi} \tag{1.1}$$

where ρ is density, $\boldsymbol{u}$ is velocity, Γ is the diffusion coefficient and S_ϕ contains the source term(s).

The mass and momentum conservation equations for the transient flow of an incompressible Newtonian fluid are obtained by setting $\phi = 1$ and $\phi = u$,v or w respectively, with $\Gamma = \mu$, dynamic viscosity, for the momentum equations. The heat balance equation is written in a slightly different form:

$$\frac{\partial\left(\rho h\right)}{\partial t} + \nabla\cdot\left(\rho\boldsymbol{u}h\right) = \nabla\cdot\left(\kappa\nabla T\right) + S_h \tag{1.2}$$

A turbulence model is used for closure in the Reynolds-Average Navier-Stokes (RANS) formulation. When RANS equations are used, the turbulent contribution to the effective viscosity is calculated using the standard $k-\varepsilon$ model [22; 23]:

$$\frac{\partial\left(\rho k\right)}{\partial t} + \nabla\cdot\left(\rho\boldsymbol{u}k\right) = \nabla\cdot\left(\left[\mu + \frac{\rho\nu_t}{\sigma_k}\right]\nabla k\right) + \rho\nu_t G + G_b - \rho\varepsilon \tag{1.3}$$

$$\frac{\partial\left(\rho\varepsilon\right)}{\partial t} + \nabla\cdot\left(\rho\boldsymbol{u}\varepsilon\right) = \nabla\cdot\left(\left[\mu + \frac{\rho\nu_t}{\sigma_\varepsilon}\right]\nabla\varepsilon\right) + C_1\rho\nu_t G\frac{\varepsilon}{k} + C_3\frac{\varepsilon}{k}G_b - C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} \tag{1.4}$$

where G is the turbulent generation rate, G_b is the generation due to buoyancy and ν_t is the Kolmogorov-Prandtl expression for turbulent kinematic viscosity.

$$\begin{aligned} G \;=\; & 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right] + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 \\ & + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right)^2 \end{aligned} \tag{1.5}$$

$$G_b = -\nu_t\frac{\boldsymbol{g}\cdot\nabla\rho}{\sigma_t} \tag{1.6}$$

The solutions for k and ε are used to calculate the eddy viscosity using:

$$\mu_t = \rho C_\mu\frac{k^2}{\varepsilon} \tag{1.7}$$

where C_μ is a dimensionless constant which is usually taken to be 0.09 for incompressible, low-temperature-gradient flows. The values for the other constants used are

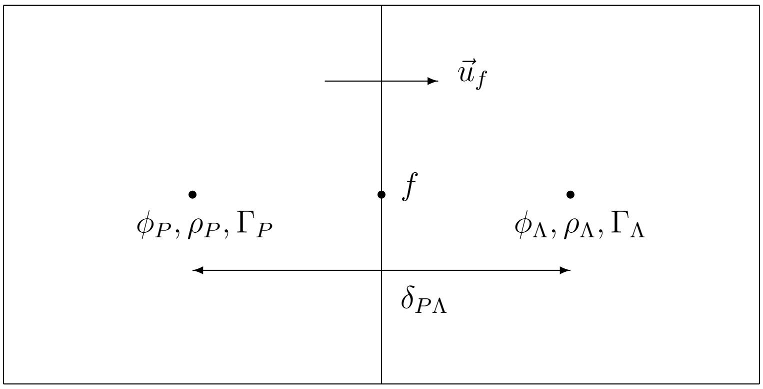
$\vec{u}_f$
f
ϕ_P, ρ_P, Γ_P
$\phi_\Lambda, \rho_\Lambda, \Gamma_\Lambda$
$\delta_{P\Lambda}$

Figure 1.1: Cell centres P and Λ, and face f to define convection and diffusion fluxes

$C_1 = 1.44$, $C_{2\varepsilon} = 1.92$ and $C_3 = 0.4$ [15]. The kinematic turbulent viscosity ν_t is μ_t divided by density.

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{1.8}$$

1.2.2 Discretisation procedure

The finite volume method is used to discretise the transport equation (1.1). The domain is divided into cells called finite volumes. The governing equations are integrated over each control volume. The convection and diffusion terms are then cast in surface integral form using Gauss's theorem. The transport equation then becomes:

$$\int\limits_V \frac{\partial\left(\rho\phi\right)}{\partial t}\,\mathrm{d}V + \int\limits_A \left(\rho\boldsymbol{u}\phi\right)\cdot\mathrm{d}\mathbf{A} = \int\limits_A \left(\Gamma\nabla\phi\right)\cdot\mathrm{d}\mathbf{A} + \int\limits_V S_\phi\,\mathrm{d}V \tag{1.9}$$

The application of a suitable differencing scheme leads to the following set of linear equations, which are to be solved simultaneously [24; 25].

$$a_P\phi_P = \sum_{neighbours} a_{nb}\phi_{nb} + b \tag{1.10}$$

The neighbours are grid nodes adjacent to the control volume. The coefficients a_P and a_{nb} contain convective and diffusive fluxes, together with some stabilisation terms depending on how the source is handled to help convergence. The term b contains the linearised source term arising from the discretisation procedure.

Table 1.1: Differencing schemes

Differencing Scheme	$\mathrm{Fn}\left(\lvert\mathrm{Pe}_f\rvert\right)$
Central Difference Scheme (CDS)	$1 - 0.5\lvert\mathrm{Pe}_f\rvert$
Upwind	1
HYBRID	$\max\left(0, 1 - 0.5\lvert\mathrm{Pe}_f\rvert\right)$

1.2.3 Differencing schemes

The strength of convection F_f and diffusion conductance D_f in the convection and diffusion terms arising after volume integration as in equation (1.9) are given by

$$F_f = A_f \rho_f \left(\boldsymbol{u} \cdot \hat{\boldsymbol{n}}\right)_f \tag{1.11}$$

$$D_f = A_f \frac{\Gamma_f}{\delta_{P\Lambda}} \tag{1.12}$$

The Peclet number Pe is an appropriate measure to determine the relative strengths of convection and diffusion in the computational domain.

$$\mathrm{Pe} = \frac{F_f}{D_f} = \frac{\rho_f u_f \delta_{P\Lambda}}{\Gamma_f} \tag{1.13}$$

1.2.3.1 Two point schemes

Using the notation in Figure 1.1 (where the face velocity u_f can be positive, zero or negative) and the variables defined above, the discretised convection-diffusion terms can be expressed in the following form

$$\left[D_f \mathrm{Fn}\left(\lvert\mathrm{Pe}_f\rvert\right) + \max\left(-F_f, 0\right)\right]\left(\phi_P - \phi_\Lambda\right) + F_f \phi_P \tag{1.14}$$

where the function $\mathrm{Fn}\left(\lvert\mathrm{Pe}_f\rvert\right)$ depends on the differencing scheme used.

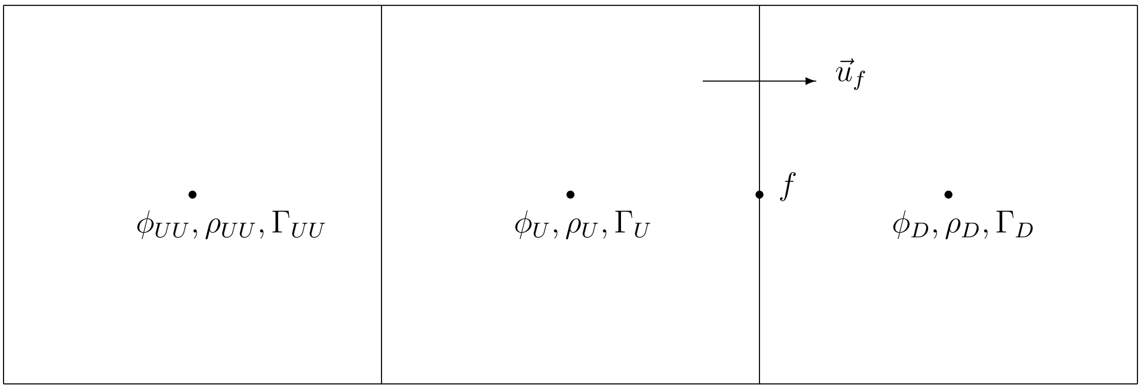
$\vec{u}_f$
f
$\phi_{UU}, \rho_{UU}, \Gamma_{UU}$
ϕ_U, ρ_U, Γ_U
ϕ_D, ρ_D, Γ_D

Figure 1.2: Discretisation for higher order schemes

Central Difference Scheme (CDS) In central differencing, the face value is taken as the average of the two surrounding nodal values.

Upwind The upwind scheme was suggested by Courant *et al.* in the early fifties [26]. The diffusion term is unaltered but the convection term uses the upwind nodal value for ϕ [15].

HYBRID The HYBRID scheme discriminates between diffusion driven flow and convection driven flow to choose how to resolve face values [27]. At small length scales, or low speed flow, transport is driven mainly by diffusion and a central difference scheme is used to calculate face values from the neighbouring nodal values. For high speed flow, transport is dominated by convection, and an upwind value is appropriate for the face.

When $|Pe_f| > 2$, convection dominates the flow and the scheme is equivalent to upwinding without diffusion. For low Peclet numbers, the values at the faces are taken as the average of the two nodal values, just like in CDS.

1.2.3.2 Higher order schemes

Higher order schemes need more than two nodes to determine a face value: a third "upwind upwind" is required. Upwind is used for faces near a boundary where there is no "upwind upwind" value.

SMART The SMART scheme by Gaskell and Lau [28] calculates face values as

$$\phi_f = \phi_U + 0.5\Psi\left(r\right)\left(\phi_U - \phi_{UU}\right) \tag{1.15}$$

using the notation from Figure 1.2, where

$$r = \frac{\phi_D - \phi_U}{\phi_U - \phi_{UU}} \tag{1.16}$$

and

$$\Psi\left(r\right) = \max\left[0, \min\left(2r, 0.75r + 0.25, 4\right)\right] \tag{1.17}$$

The following source contribution is added to both adjacent elements.

$$-fa0.5\Psi\left(r\right)\left(\phi_U - \phi_{UU}\right) \tag{1.18}$$

where fa is the convection flux out of the element [15].

1.2.4 Unstructured meshes and momentum interpolation

For unstructured grid arrangements, it is more convenient to store fluid velocities at cell centres. This collocated arrangement is efficient, with a large number of faces per cell a common feature of problems with complex geometries. A higher order dissipation term is then required to prevent non-physical oscillatory solutions: this dissipation term is introduced using a momentum interpolation practice originally introduced by Rhie and Chow [29].

Since a co-located velocity arrangement is adopted in our present code, a momentum interpolation method similar to that proposed by Rhie and Chow [29] is required to ensure correct pressure-velocity coupling and avoid non-physical solutions [25; 30].

Momentum interpolation, as implemented in PHYSICA [15], is given by

$$u_f = \overline{u_f} + \overline{d_f}\left(\overline{\nabla p_f \cdot \boldsymbol{x}} - \nabla p_f \cdot \boldsymbol{x}\right) \tag{1.19}$$

where

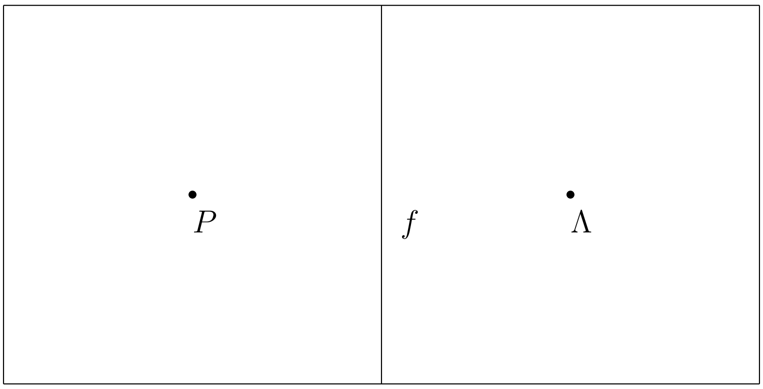

Figure 1.3: Cell centres P and Λ, and face f contained in equation for face pressure in the presence of body forces.

$$\begin{aligned}
\overline{u_f} &= 0.5u_P + 0.5u_\Lambda \\
\overline{\nabla p_f \cdot \boldsymbol{x}} &= 0.5\nabla p_P \cdot \boldsymbol{x} + 0.5\nabla p_A \cdot \boldsymbol{x} \\
\nabla p_f \cdot \boldsymbol{x} &= A_f n_x \left(P_\Lambda - P_P\right) \\
\overline{d_f} &= \left(0.5a_P + 0.5a_\Lambda\right)^{-1}
\end{aligned}$$

u_f is the face velocity required to compute the mass source term in the pressure correction equation used in Section 1.2.5.

1.2.4.1 Body forces in pressure interpolation

If body forces are applied in the momentum equation, they will be balanced by the pressure gradient term. This yields a larger pressure gradient equivalent to

$$\frac{\boldsymbol{B}_\Lambda}{V_\Lambda} + (\nabla p)_{U\Lambda} \tag{1.20}$$

where $\boldsymbol{B}_\Lambda$ is the integrated body force in cell Λ, with volume V_Λ. $(\nabla p)_{U\Lambda}$ is some underlying pressure gradient.

Assuming a constant value of $(\nabla p)_{U\Lambda}$ between element centres, the face pressure is given by

$$\begin{aligned} p_f &= p_\Lambda + \alpha \boldsymbol{d} \cdot \frac{\boldsymbol{B}_\Lambda}{V_\Lambda} + \alpha d \frac{\partial p}{\partial d} \\ &= p_B + \alpha \boldsymbol{d} \cdot \frac{\boldsymbol{B}_B}{V_B} - \alpha d \frac{\partial p}{\partial d} \end{aligned} \tag{1.21}$$

where $\boldsymbol{d}$ is the vector from cell centre Λ to cell centre B and α is the weighting factor based on relative distances from the face.

Rearranging equation (1.21),

$$\frac{\partial p}{\partial d} = \left(p_B - p_\Lambda - (1-\alpha)\, \boldsymbol{d} \cdot \left(\frac{\boldsymbol{B}_B}{V_B} \right) - \alpha \boldsymbol{d} \cdot \left(\frac{\boldsymbol{B}_\Lambda}{V_\Lambda} \right) \right) / d \tag{1.22}$$

Substituting equation (1.22) into (1.21),

$$p_f = \alpha p_\Lambda - (1-\alpha)\, p_B + \alpha (1-\alpha)\, \boldsymbol{d} \cdot \left(\frac{\boldsymbol{B}_\Lambda}{V_\Lambda} - \frac{\boldsymbol{B}_B}{V_B} \right) \tag{1.23}$$

[15]

1.2.5 Pressure handling

Pressure and velocity are both unknown quantities to be solved. However, pressure and velocity are strongly coupled in the momentum equation and pressure only appears as a gradient in the momentum equation. Patankar and Spalding [31] devised a special implicit procedure to calculate pressure, called SIMPLE (Semi-implicit Method for Pressure-Linked Equations). The outline of this iterative method is as follows:

1. The momentum equations are solved with a guessed pressure field p^*.

2. Mass continuity errors are computed to form a pressure correction equation, which is solved.

3. The corrections are applied to pressure and velocities.

4. Steps 1 - 3 are repeated until the convergence criterion is reached.

5. The energy conservation equation, and other scalar equations, are solved using the velocity values obtained.

6. The whole process is repeated until convergence is achieved.

A more detailed description of the SIMPLE method, including the equations and corrections in a collocated scheme, can be found in Appendix A. The pressure correction procedure has been enhanced by Van Doormaal and Raithby [32], and their SIMPLEC (SIMPLE Consistent) algorithm is used for pressure calculation, as described in section A.2. A good summary of different SIMPLE-like schemes, like SIMPLER [24] and PISO [33], can be found in a publication from Acharya et al. [34].

Extension of the SIMPLEC algorithm on significantly non-orthogonal geometries results in a complex pressure-correction term containing extra terms. Including these extra terms in a numerical experiment increases the robustness of the code at the expense of performance [35]. This research investigates alternative methods which can be used to avoid these extra terms.

1.2.6 Boundary source terms

Boundary sources are expressed in coefficient-value form:

$$S = \left(C_\phi + < \dot{m}, 0 >\right)\left(\phi_{ext} - \phi_P\right) \tag{1.24}$$

where C_ϕ is the coefficient of ϕ, ϕ_{ext} is the external value of ϕ, ϕ_P is the value of ϕ inside cell P and the mass inflow across the boundary face is given by $\dot{m}$. Breaking sources in coefficient-value form leads to larger diagonal terms in the coefficients matrix, improving the convergence behaviour of the iterative process used to solve the linear system of equations (1.10) [36].

1.2.7 Source linearisation for $k - \varepsilon$ turbulence model

The source terms for the kinetic energy of turbulence and dissipation rate - equations (1.3) and (1.4) - are non linear. Assuming a coefficient-value form $S_C + S_P\phi$ similar as described in Section 1.2.6 for sources, two methods are available to linearise the sources:

Method 1

k source terms

$$S_C = \nu_t \rho G_k \tag{1.25}$$

$$S_P = -C_\mu \frac{k\rho}{\nu_t} \tag{1.26}$$

ε source terms

$$S_C = C_\mu \rho k C_{1\varepsilon} G_k \tag{1.27}$$

$$S_P = -C_{2\varepsilon} C_\mu \frac{k\rho}{\nu_t} \tag{1.28}$$

Method 2

k source terms

$$S_C = 0.5 \nu_t \rho G_k \tag{1.29}$$

$$S_P = -1.5 \rho \frac{\varepsilon}{k} \tag{1.30}$$

ε source terms

$$S_C = \frac{1}{3} C_{2\varepsilon} \rho \varepsilon \frac{\varepsilon}{k} + C_{1\varepsilon} \rho G_k \nu_t \frac{\varepsilon}{k} \tag{1.31}$$

$$S_P = -\frac{4}{3} C_{2\varepsilon} \rho \frac{\varepsilon}{k} \tag{1.32}$$

Method 1 offers quicker convergence at the expense of numerical stability [15].

1.2.8 Momentum false time step

Inertial under-relaxation via the introduction of a momentum false time step term is a well known method used to stabilize the iterative solution process. This is achieved by introducing a source term

$$\frac{\rho V}{\delta t_f} \left(\phi^* - \phi \right) \tag{1.33}$$

where ϕ^* is the previous iteration value of ϕ. When the solution has converged, $\phi^* = \phi$ thus making this source term equal zero [36]. If δt_f is made very small, the value of ϕ is over-relaxed and will not deviate from ϕ^* by a large amount at every iteration.

The appropriate false-time step value is the smallest time taken for a fluid particle to leave one control volume: this value is used to achieve fast convergence.

Using the notation of Section 1.2.6 for source terms,

$$C_\phi = \frac{\rho V}{\delta t_f} \tag{1.34}$$

and

$$\phi_{ext} = \phi^* \tag{1.35}$$

where the * superscript denotes the previous iteration value.

1.2.9 Residuals calculation

At the end of each iteration, the residual Υ is calculated to determine if convergence has been achieved.

$$\Upsilon = \phi - \phi^* \tag{1.36}$$

where the * superscript denotes the previous iteration value. The convergence criterion is taken as having the norm of the residual vector being below than a threshold level (dependent on the problem solved). The vector norm for residual calculation is taken as

$$\Upsilon_{max}\sqrt{\sum_i \left(\frac{\Upsilon_i}{\Upsilon_{max}}\right)^2} \tag{1.37}$$

1.3 Thesis contributions

This study has been concerned with the convergence behaviour of a collocated finite volume code in the face of non-orthogonality and compressibility. The novel contributions of this thesis are as follows:

1. A new parameter ς, presented in Section 2.7, is used as an adjusting relaxation parameter for the non-orthogonal diffusion correction term. This parameter can be used to fine tune the level of relaxation when applying a deferred correction type [37] iterative technique to correct diffusion fluxes in non-orthogonal meshes.

2. A pressure-interpolation procedure is presented in Section 2.4.1.1 to handle extreme cases where convergence cannot be achieved with a deferred correction approach to the diffusion flux. The results of the solver are ignored in cells with large mass imbalances. Pressure corrections are indirectly interpolated from pressure values in an attempt to recover an accurate solution far from the interpolated cells.

3. A modified SIMPLEC [32] procedure is presented in Section 3.2.4 to handle pressure correction in compressible cases. A convection coefficient term is added to the pressure correction equation to both accurately reflect mass conservation in the presence of a variable density field and point the pressure correction procedure towards physical mass conservation.

4. A false time step term for pressure correction is introduced in Section 3.2.5 to relax the pressure correction procedure in compressible computations should the modified SIMPLEC procedure in 3 deem to be unstable.

5. A modified turbulence model applicable to compressible jets entering an environment with higher ambient temperature than the jet temperature is presented in Section 3.3.4.1. This model was developed based on empirical results from Sumi *et al.* [14] and is to be used in conjunction with the HYBRID difference scheme when solving the momentum equations.

6. A novel pressure correction approach as applied to problems involving a free surface is presented in Section 4.2.1.3 to be used in cases where a compressible jet impinges on a free surface. This numerical method, called zonal GALA (Gas And Liquid Analyser), treats the jet as a compressible gas, and the liquid as an incompressible fluid, within the same domain and in the same linear set of pressure correction equations.

1.4 Thesis outline

This thesis consists of 6 chapters. Chapter 1 sets the background for undertaking this research and presents a basic finite volume method review upon which this research is built. A section describes the contributions of this research and finally outlines the other chapters contained in this thesis.

Chapters 2 to 4 all begin with a literature review of the current research surrounding the problems to be tackled, present the solutions to these challenges and the methods used in this thesis, and contain a final section with the results of the numerical methods proposed.

Chapter 2 begins by a literature review of non-orthogonality and skewed mesh handling in Computational Fluid Dynamics. The new numerical models developed to tackle non-orthogonality, namely the diffusion flux correction adjustment parameter and the pressure interpolation method, are described. A series of test cases - Poisseuille flow and moving lid cases - are run and results are compared with analytic solutions (if applicable) or benchmarks and other results in the literature.

Chapter 3 begins by a discussion of the literature review and existing models to handle compressibility in a computational framework. The new methods developed during this research are then presented: the compressible SIMPLEC procedure and the false time-step term of pressure correction. A series of test cases are run to validate the model. The first cases - the de Laval nozzle and oblique shock cases - have analytic solutions with which the results have been compared with. The supersonic flow in a channel with a circular bump has also been computed and compared with results from the literature. Finally, the case of an oxygen jet entering a hot environment is presented, and a new turbulence model is proposed to tackle this particular kind of flow problem.

Chapter 4 starts with a review of free surface methods and presents existing numerical methods to handle interfaces. A novel zonal GALA method is presented, and then used in the case of an axisymmetric oxygen jet impinging on a liquid surface. Finally, the changes in deformation of the free surface with different discrimination criteria for zonal GALA is investigated.

Chapter 5 concludes the thesis and Chapter 6 discusses possible work which can be conducted to extend this research.

Chapter 2

NON-ORTHOGONALITY

Fluid flow problems involving complex geometries are often encountered in engineering, requiring unstructured non-orthogonal meshes. However, solution instabilities are often encountered when solving the Navier-Stokes equations on non-orthogonal meshes in the Finite Volume formulation. These instabilities arise due to:

1. difficulties in discretising the diffusion term on unstructured meshes,
2. dropping out extra non-orthogonal terms in the pressure correction equation to speed up convergence, and
3. mesh quality issues and wrong aspect ratios when automatically generating meshes for complex geometries.

This behaviour, plus the cost of manually creating acceptable meshes, restricts the range of problems that can be modelled and much research has been devoted to the study of non-orthogonality in numerical fluid flow problems in the literature.

This chapter will first define non-orthogonality measures and present ways of tackling non-orthogonality in flow computations. The numerical method proposed to handle non-orthogonality will be described after the literature review, followed by test cases to validate the proposed solutions.

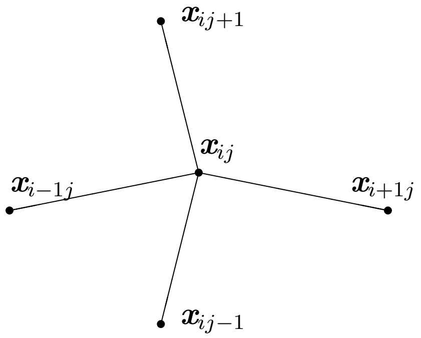

Figure 2.1: 2D node non-orthogonality measure geometry for node ij.

2.1 Non-orthogonality measures and effect on truncation error

A mesh quality measure needs to be formally defined to assess the extent of non-orthogonality or skewness, and its effect on truncation error. The criterion can be used to discriminate between corrections to be used, or if corrections are required at all, and to drive the mesh refinement process in mesh adaptation techniques. These measures are functions of angles at nodes or face normals.

2.1.1 Nodal measure

A measure of non-orthogonality between gridlines can be expressed by the scalar products between all the edges forming these angles. A nodal function for non-orthogonality is then obtained; in a 2D configuration as shown in Figure 2.1, this function can be written as [38]

$$\begin{aligned}\sigma_{ij} \;=\;& \left[(\boldsymbol{x}_{ij}-\boldsymbol{x}_{i+1j})\cdot(\boldsymbol{x}_{ij}-\boldsymbol{x}_{ij+1})\right]^2+\left[(\boldsymbol{x}_{ij}-\boldsymbol{x}_{i-1j})\cdot(\boldsymbol{x}_{ij}-\boldsymbol{x}_{ij+1})\right]^2\\ &+\left[(\boldsymbol{x}_{ij}-\boldsymbol{x}_{i+1j})\cdot(\boldsymbol{x}_{ij}-\boldsymbol{x}_{ij-1})\right]^2+\left[(\boldsymbol{x}_{ij}-\boldsymbol{x}_{i-1j})\cdot(\boldsymbol{x}_{ij}-\boldsymbol{x}_{ij-1})\right]^2 \qquad (2.1)\end{aligned}$$

Twelve angles are required in 3D for each node. If all angles surrounding the node are 90°, $\sigma_{ij}=0$ [38].

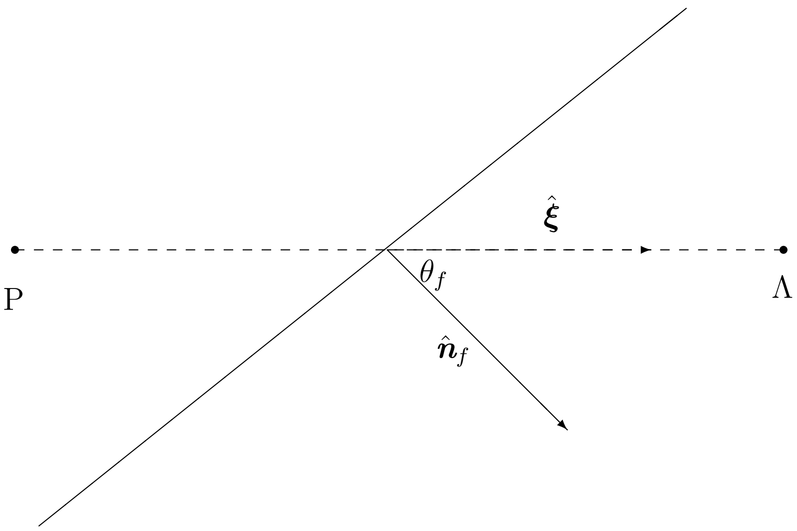

Figure 2.2: Face non-orthogonal angle.

2.1.2 Face non-orthogonality measure

The skewness of a face can be determined by the angle θ_f between the face normal $\hat{\boldsymbol{n}}_f$ and the vector joining the two adjacent face centres $\hat{\boldsymbol{\xi}}$. For a perfectly orthogonal face, $\theta_f = 0$.

2.1.3 Maximum normals skewness

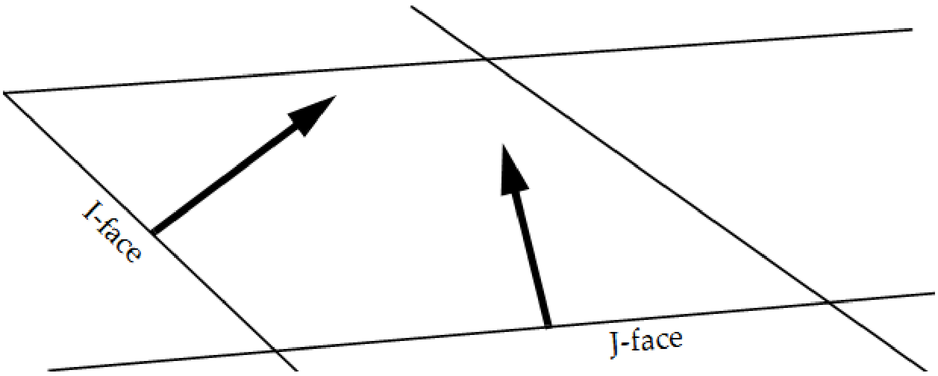

Figure 2.3: Measuring skewness value for two adjacent faces. Figure from Tecplot 360 User's Manual [39].

One mesh skewness measure used by the post-processing package Tecplot is given by the dot product of two faces in a cell. For example, IJ skewness as shown in Figure 2.3 is given by

$$|\hat{\boldsymbol{n}}_I \cdot \hat{\boldsymbol{n}}_J| \tag{2.2}$$

The maximum normal skewness (when comparing the dot products for the IJ, IK and KJ pairs) is a measure of the extent of non-orthogonality in a particular cell [39]. The range of this measure is $[0, 1)$.

2.1.4 Effect of non-orthogonality on truncation error

The order of a difference scheme refers to the rate of decrease of truncation error with mesh density increase. For uniform grids, this can be simply quantified by increasing the number of points (equivalent to decreasing the point spacing). For unstructured meshes, the interpretation of order is ambiguous since it can relate to either increasing the number of mesh points, or keeping the total number of points constant but changing their locations, or both [40].

Non-orthogonality also influences the truncation error of a difference scheme. Mastin [40] established that the error varies inversely with the sine of the angle between the coordinates lines of a curvilinear system. Non-orthogonal angles greater than 45° do not impact on truncation error with reasonable grid spacing within the middle of the domain. However, departure from non-orthogonality is problematic at the boundaries, where grids should be made as orthogonal as possible [40]. Orthogonality is therefore a desired grid characteristic since skewness increases truncation error [38].

2.2 Mesh deformation and analysis

Ideally, the mesh used to model a problem would be orthogonal, leading to a simple discretisation technique for the case to be solved. Orthogonal cases were investigated by the Computational Fluid Dynamics (CFD) community in the early days, using the finite difference method. Contrarily to the solid mechanics community who used the

Finite Element Method (FEM), the CFD community had to seek elegant techniques to fit complex geometries so that their discretisation methods could be applied [41].

2.2.1 Mapping physical space to a general curvilinear system

Since most discretisation techniques used by the early CFD community were based on orthogonal meshes, their first predictions were on Cartesian and curvilinear coordinate systems. Most curvilinear systems (e.g. cylindrical, spherical ...) can be considered as orthogonal from a programming point of view, and can be dealt with using a structured framework [42].

Conformal mapping is a transformation technique which is angle preserving [43]. Conformal mapping was first applied to CFD by Sells [44] who studied flow past an aerofoil by mapping the mesh to unit circle [41]. However, conformal mapping can be applied to two dimensional problems only, and alternative coordinate transformation techniques are required if one is to align a mesh with domain boundaries [41].

Karki and Patankar [45], and Davidson and Hedberg [46] presented good derivations of the discretised equations for non-orthogonal flow using a staggered body fitted co-ordinates in the finite volume formulation. Xu and Zhang [47] derived the equations for a non-staggered arrangement. The basis of this class of solution is (taking a steady state two dimensional problem as an example) to modify the Cartesian conservation equation

$$\frac{\partial\left(\rho u\phi\right)}{\partial x}+\frac{\partial\left(\rho v\phi\right)}{\partial y}=\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right)+R\left(x,y\right) \tag{2.3}$$

by introducing new independent variables ξ and η and using the general transformations $\xi=\xi\left(x,y\right)$ and $\eta=\eta\left(x,y\right)$, into

$$\begin{aligned}\frac{1}{J}\frac{\partial\left(\rho U\phi\right)}{\partial\xi}+\frac{1}{J}\frac{\partial\left(\rho V\phi\right)}{\partial\eta} &= \frac{1}{J}\frac{\partial}{\partial\xi}\left[\frac{\Gamma}{J}\left(q_1\frac{\partial\phi}{\partial\xi}-q_2\frac{\partial\phi}{\partial\eta}\right)\right]\\ &+\frac{1}{J}\frac{\partial}{\partial\eta}\left[\frac{\Gamma}{J}\left(q_3\frac{\partial\phi}{\partial\eta}-q_2\frac{\partial\phi}{\partial\xi}\right)\right]\end{aligned} \tag{2.4}$$

$$+ S(\xi, \eta)$$

where

$$\begin{aligned} U &= u\frac{\partial y}{\partial \eta} - v\frac{\partial x}{\partial \eta} && (2.5)\\ V &= v\frac{\partial x}{\partial \xi} - u\frac{\partial y}{\partial \xi} && (2.6)\\ q_1 &= \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 && (2.7)\\ q_2 &= \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta} && (2.8)\\ q_3 &= \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2 && (2.9)\\ J &= \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta} && (2.10) \end{aligned}$$

A plethora of articles successfully adopting other mapping techniques applied to both finite difference and finite volume methods can be found in the literature: older papers use the finite difference scheme in conjunction with a body-fitted system. Oliver and Miller [48] modelled the natural convection in a gas cooled cable, while Shyy *et al.* [49] applied this technique to recirculating flow problems. Using finite differences, particular attention has to be paid to the conservativeness of the scheme. This can be avoided by using a finite volume approach.

Coelho and Pereira [50] solved for three-dimensional steady laminar flows in a square diffuser, elliptical duct, an S-shaped duct and a moving lid cavity using a finite volume, non-staggered approach with generally good results. Kobayashi and Pereira [51] also used a non-staggered grid system with the finite volume method to model recirculating flow in a cavity, laminar flow through a pipe with a sudden contraction, laminar natural convection in an eccentric annulus and steady laminar flow round a cylinder. He and Salcudean [52] also investigated three dimensional flows in non-orthogonal geometries using the finite volume approach. They successfully applied this method to “moderately non-smooth grids” for flows in a moving lid cavity, a pipe with constriction and a duct with a smooth 90° bend. They however mentioned that they experienced convergence difficulties with “highly non-smooth grids” and recommended that such geometries should be avoided.

However, convergence can be achieved on highly non-orthogonal grids with this approach, with the application of complex face resolution approaches. Moulinec and Wesseling [53] paid particular attention to the scheme used to calculate face values from cell values. They compared the standard two point method used by previous authors with other elaborate interpolation methods - using path integrals and variants of bilinear interpolation, to model the Poisseuille flow in a duct using a moderately and a highly distorted grid. All schemes worked with the moderately distorted grid, though complex interpolation schemes were more accurate. However, for heavily distorted grids, only higher order schemes yield converged results.

2.2.2 Generating unstructured meshes for complex geometries

At the dawn of CFD, numerical methods were developed with structured meshes: systematic arrays of quadrilateral or hexahedral cells. These structured cells simplified the discretisation process and enabled the use of very efficient solvers. However, it is very difficult to generate structured meshes for the complex geometries that are commonly encountered in engineering [54].

Unstructured grids, on the other hand, have irregular connectivity, and can be easily generated by either numerically solving a partial differential equation, such as Thompson’s method [55] or by triangulation methods [41; 56]. Such an unstructured grid has been successfully used by Jameson *et al.* [57] to solve the flow over an aircraft using tetrahedral meshes and FEM.

The use of unstructured meshes is however at the expense of accuracy: predictions using RANS equations on a structured mesh have been found to be closer to experimental values than predictions computed on an unstructured mesh [41]. Nevertheless, the time consuming and problematic process of generating a structured mesh for complex geometries justify the use of unstructured grids, albeit having to resort to correction methods to improve results.

2.2.3 Correcting mesh defects

When grid generators do not produce a suitable mesh, it might be required to look into *a posteriori* methods to correct the mesh defects present using mesh smoothing techniques [38; 58]. This is generally achieved by minimising mesh error indicators such as non-orthogonality measures described in Section 2.1 [38; 40].

However, in multiphysics industrial problems one may not always be able to optimise the mesh for the flow simulation, since it will lead to conservation and stability issues if the mesh deformation is not properly handled. The mesh used for a simulation will always be a compromise between conflicting requirements of stability, accuracy and solution speed - so directly tackling ‘bad’ meshes is unfortunately sometimes required.

For the same reasons we cannot afford to build good meshes for capturing shocks but we want useful results that do not diverge due to the physical presence of shocks and other compressibility effects.

2.2.4 Control-Volume Finite-Element Method

Gradient determination is very complex if no line structure is devised. Difficulties in discretising diffusion fluxes or using higher order schemes can by avoided by using a finite element based method for discretisation. Of particular interest are control-volume finite-element methods (CVFEM) which combine the conservativeness of the finite volume methods and the geometrical flexibility of FEM [59; 60].

McBride *et al.* [61; 62] compared a vertex-based co-located method with co-located cell centred finite volume methods; they found that vertex-based methods could be used to solve a larger range of problems involving mesh skewness, e.g. the computational modelling of the mould filling process for a complex wheel casting [63].

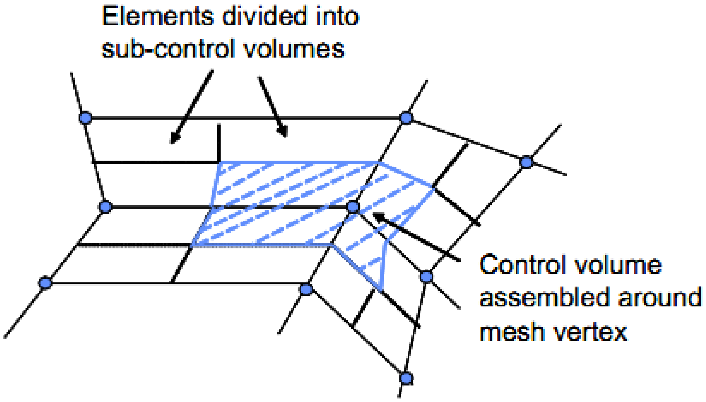
Elements divided into sub-control volumes
Control volume assembled around mesh vertex

Figure 2.4: Vertex-based control volume. Figure from [63]

In this particular technique, the conservation equations of the form (1.1) are discretised over a vertex-based control volume. Each finite volume cell is subdivided into sub-control volumes assembled around the vertices as shown in Figure 2.4. The local variation of ϕ is then described by piecewise polynomial functions. Diffusion fluxes can be directly evaluated on non-orthogonal meshes [61]. However, these vertex-based methods are very expensive computationally. McBride [19] found that a simple thermal analysis calculation requires 1.6 times the computational time of a cell-centered approach, requiring five times as much RAM space to store vertex values.

2.3 Handling of non-orthogonal meshes in the finite volume method

2.3.1 Handling of the pressure correction equation

When the SIMPLE [31] algorithm is extended to non-orthogonal configurations, the pressure correction equation becomes very complex, with a 9-point computational molecule in 2D and a 19-point computational molecule in 3D. Perić [35] analysed the performance of the full pressure correction equation and compared it with the simplified version on non-orthogonal grids. He found it necessary to use the full pressure correction equation for non-orthogonal angles, as defined in Figure 2.2, greater than 45°, together with low linear under-relaxation parameters ranging from 0.2 to 0.3 for pressure.

In 1994, Cho and Chung [64] proposed the decomposition of the full pressure correction equation into implicit and explicit parts; in their novel method, the explicit term is smaller than the implicit terms, so as to guarantee convergence. This method allowed a wide convergence range for relaxation factors, which was found to be independent of the skewness of the mesh. However, this stability is at the expense of convergence rate, which is slower than for simplified treatments.

Wang and Komori [65] investigated the laminar flow in a cavity with moving lid to demonstrate that the convergence behaviour for non orthogonal computations with SIMPLEC [32] is superior to that those using SIMPLE for pressure-velocity coupling. The convergence difficulties encountered when using SIMPLE can be avoided by using SIMPLEC. Using the non-orthogonal cross derivatives generally ignored in the original SIMPLE method reduces the CPU time by more than 60% [65].

2.3.2 Handling of the diffusion term in finite volume methods

Highly skewed meshes pose consistency problems when discretising terms of second derivative order for diffusive fluxes [1]. In a finite volume strategy, these problems originate from the approximation of the following diffusive fluxes.

$$\int\limits_{V} \nabla \cdot \nabla \phi \, \mathrm{d}V = \int\limits_{A} (\nabla \phi) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}A \tag{2.11}$$

where A is a typical surface on the volume cell V. For a generic cell, the integral can be written down as a summation:

$$\int\limits_{A} (\nabla \phi) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}A = \sum_{f} A_f \left(\nabla \phi\right)_f \cdot \hat{\boldsymbol{n}}_f \tag{2.12}$$

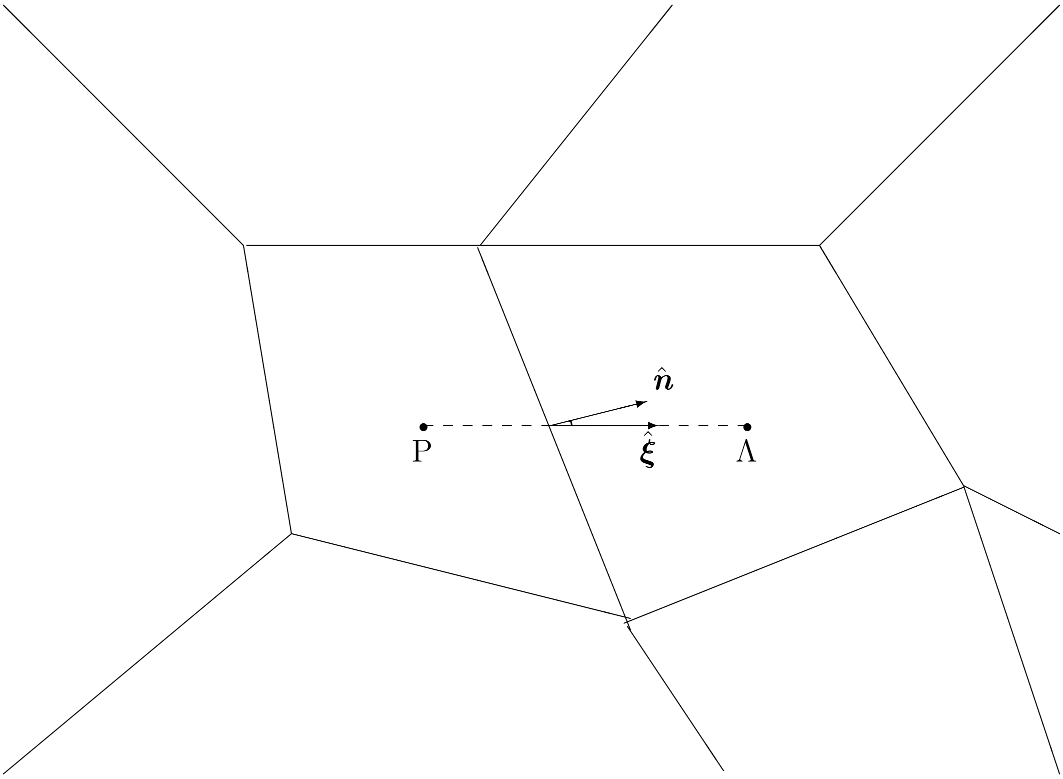

Figure 2.5: Description of a typical non-orthogonal face.

The difficulty lies in discretising $(\nabla\phi)_f \cdot \hat{\boldsymbol{n}}_f$ when meshes are non-orthogonal and therefore $(\nabla\phi)_f$ is not parallel to $\hat{\boldsymbol{n}}_f$ [1].

2.3.2.1 Deferred correction method

Ferziger and Perić [37] proposed an iterative process such that

$$A_f\left(\nabla\phi\right)_f^m \cdot \hat{\boldsymbol{n}}_f \longrightarrow A_f\left(\nabla\phi\right)_f \cdot \hat{\boldsymbol{n}}_f \text{ as m } \longrightarrow \infty \tag{2.13}$$

This is achieved by the treating higher order fluxes containing the non-orthogonality approximation explicitly. If D represents the diffusive flux, superscript ^{L} denotes the lower order approximation of non-orthogonality, and ^{H}, the higher order approximation, the flux is given by

$$D_f = D_f^L + \left(D_f^H - D_f^L\right)^{old} \tag{2.14}$$

where the subscript $_f$ represents a face value at face f.

This approach can be exemplified by considering a Poisson problem of the form

$$-\nabla\cdot\left(\nabla\phi\right) = g \tag{2.15}$$

After discretisation with the finite volume method and using the Green-Ostrogradsky theorem,

$$-\sum_f A_f\left(\nabla\phi\right)_f \cdot \hat{\boldsymbol{n}}_f = g\cdot V \tag{2.16}$$

The deferred correction iteration approach in long form, with m representing the current iteration, and using the notation of Figure 2.5, becomes

$$-\sum_f A_f\left(\nabla\phi\right)_f^m \cdot \hat{\boldsymbol{\xi}}_f = g\cdot V - \sum_f A_f\left(\nabla\phi\right)_f^{m-1}\cdot\left(\hat{\boldsymbol{\xi}}_f - \hat{\boldsymbol{n}}_f\right) \tag{2.17}$$

When the solution has converged, i.e. $(\nabla\phi)_f^m = (\nabla\phi)_f^{m-1} = (\nabla\phi)_f$, the iterative equation (2.17) becomes exactly (2.16).

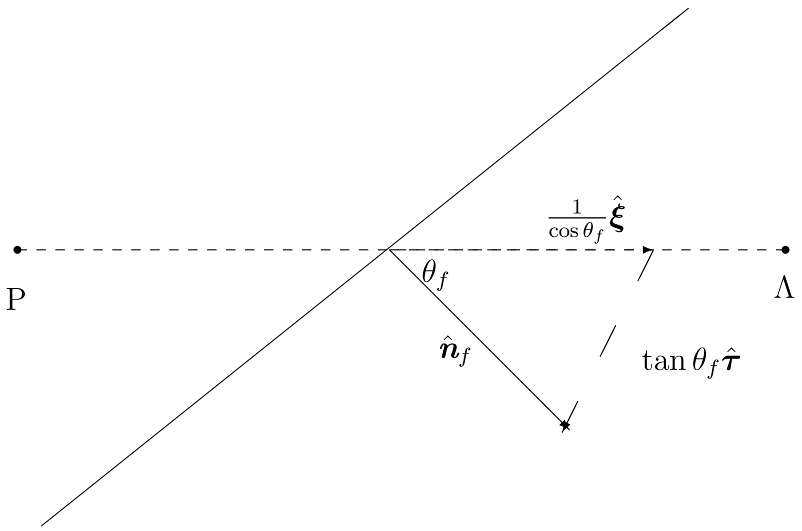
$\frac{1}{\cos\theta_f}\hat{\boldsymbol{\xi}}$
θ_f
P
Λ
$\hat{\boldsymbol{n}}_f$
$\tan\theta_f\hat{\boldsymbol{\tau}}$

Figure 2.6: Decomposition of $\hat{\boldsymbol{n}}_f$ into two components.

2.3.2.2 Ahipo and Traore's treatment for diffusion term [1; 2]

Ahipo and Traore improved the deferred correction technique applied to non-orthogonal diffusive fluxes by resolving $\hat{\boldsymbol{n}}_f$ into two components, one being along the direction of the vector joining the two adjacent cell centres, such that

$$\hat{\boldsymbol{n}}_f = \frac{1}{\cos\theta_f}\hat{\boldsymbol{\xi}}_f + \tan\theta_f\hat{\boldsymbol{\tau}}_f \tag{2.18}$$

The deferred correction iterative procedure to solve equation (2.15) then becomes

$$-\sum_f \frac{1}{\cos\theta_f} A_f \left(\nabla\phi\right)_f^m \cdot \hat{\boldsymbol{\xi}}_f = g\cdot V + \sum_f \tan\theta_f A_f \left(\nabla\phi\right)_f^{m-1} \cdot \hat{\boldsymbol{\tau}}_f \tag{2.19}$$

Expressing the iterative equation in terms of $\hat{\boldsymbol{\xi}}$ and $\hat{\boldsymbol{n}}$,

$$-\sum_f \frac{1}{\cos\theta_f} A_f \left(\nabla\phi\right)_f^m \cdot \hat{\boldsymbol{\xi}}_f = g\cdot V - \sum_f A_f \left(\nabla\phi\right)_f^{m-1} \cdot \left(\frac{1}{\cos\theta_f}\hat{\boldsymbol{\xi}}_f - \hat{\boldsymbol{n}}_f\right) \tag{2.20}$$

This method shows good convergence behaviour, even for cells having faces with $\theta_f > 60°$, whereas the standard deferred correction method diverges. They concluded that, for diffusion problems, errors on highly skewed meshes are mostly due to a wrong approach in solving the problem, rather than due to the non-orthogonal nature of the mesh itself [1].

2.3.2.3 Non-orthogonality correction for diffusion term

PHYSICA [15] proposes a similar correction method to handle these non-orthogonal cases: the normal component of the diffusion gradient is decomposed into two components

$$\boldsymbol{n} \cdot \nabla\phi = \lambda\boldsymbol{\xi} \cdot \nabla\phi + (\boldsymbol{n} - \lambda\boldsymbol{\xi}) \cdot \nabla\phi \tag{2.21}$$

where $\boldsymbol{n}$ is the normal to the face, $\boldsymbol{\xi}$ is a unit vector along the direction joining the adjacent cell centres on both sides of face f, as shown in Figure 2.5.

This formulation leads to the following the diffusion coefficient and extra source term:

$$D_f = A_f \frac{(\Gamma_\phi)_f}{d_{\Lambda P}} \lambda \tag{2.22}$$

$$S_{D_f} = A_f \left(\Gamma_\phi\right)_f (\boldsymbol{n} - \lambda\boldsymbol{\xi}) \cdot \nabla\phi \tag{2.23}$$

where $d_{\Lambda P}$ is the distance between the cell centres adjacent to the face.

The choice of the value of λ determines the level of relaxation with which the diffusion flux is to be evaluated [15]:

- A minimum residual approach is to make the last term in equation (2.23) as small as possible, by setting $\lambda = \boldsymbol{\xi} \cdot \boldsymbol{n}$. This method converges very fast, but is very unstable for highly skewed cases.
- Setting $\lambda = 1.0$ handles greater non-orthogonality at the expense of some convergence speed.
- When λ is set $\frac{1}{\boldsymbol{\xi} \cdot \boldsymbol{n}}$, the diffusion term (2.22) is large, making the code very stable. However, convergence is very slow.

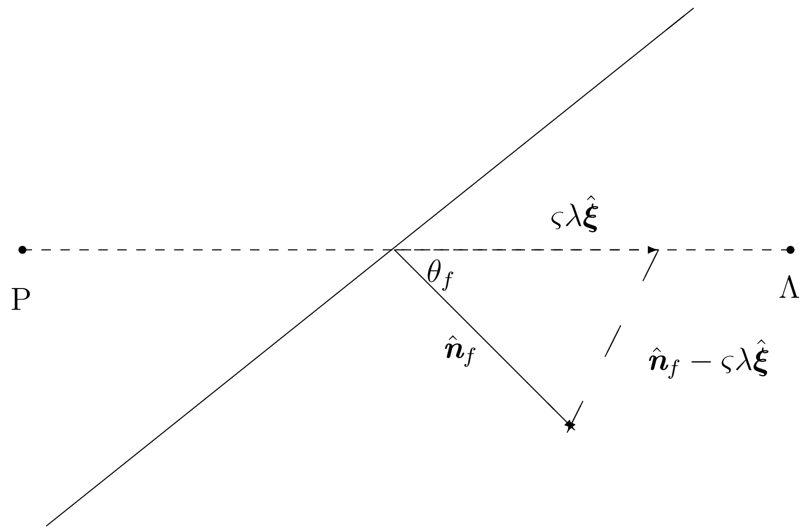
$\varsigma\lambda\hat{\boldsymbol{\xi}}$
P
θ_f
Λ
$\hat{\boldsymbol{n}}_f$
$\hat{\boldsymbol{n}}_f - \varsigma\lambda\hat{\boldsymbol{\xi}}$

Figure 2.7: Increasing the non-orthogonal vector length to adjust diffusion correction relaxation.

2.3.2.4 Adjustments to non-orthogonality corrections to diffusion term

Adjustments to the non orthogonality correction term can be effected by multiplying λ by a constant ς. When $\varsigma > 1.0$, this results in a larger diffusion coefficient term (2.22), stabilizing the code even further. On the other hand, decreasing ς will speed up convergence at the expense of stability.

The new diffusion coefficient then becomes

$$D_f = A_f \frac{(\Gamma_\phi)_f}{d_{\Lambda P}} \varsigma\lambda \tag{2.24}$$

and the new correction source term is

$$S_{D_f} = A_f \left(\Gamma_\phi\right)_f (\boldsymbol{n} - \varsigma\lambda\boldsymbol{\xi}) \cdot \nabla\phi \tag{2.25}$$

2.3.3 Non-conjunctionality

When estimating face values in the finite volume method, it is assumed that face value is constant. This face value is calculated by interpolating values from the two neighbouring cells at the face centroid. Non-conjunctionality arises when the face centre does not lie on the line joining the adjacent cell centres. In these cases, special care has to taken when interpolating these face values [17].

2.3.3.1 Special line structure to evaluate convective-diffusive transport across cell faces

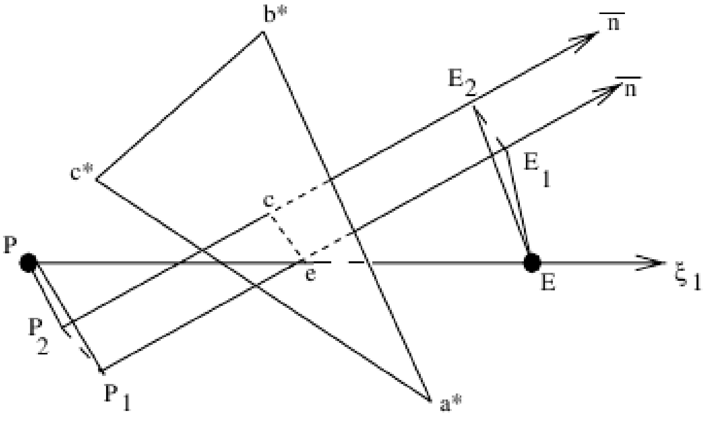

Figure 2.8: Construction of a line structure in non-conjuctional cases [8].

Date [8] carefully defined a line structure such that the face transport terms are correctly evaluated along the line passing through the face centroid. Instead of evaluating the face fluxes using cell nodes P and Λ, the fluxes are evaluated using fictitious points P_2 and Λ_2. The value of ϕ at the fictitious points are then interpolated from known values at cell centres.

This line structure has been tested in six non-orthogonal test cases by Pimpalnerkar *et al.* [66] including cases for free convection. Good results were obtained with moderately coarse grids for low Rayleigh number problems (Ra $= 10^5$), although the authors suggested that mesh refinement is still required for buoyancy driven flow in a tilted cavity with large Ra ($> 10^6$).

This method uses a large number of interpolations to resolve face values. The number of interpolations required for a consistent face value interpolation can be

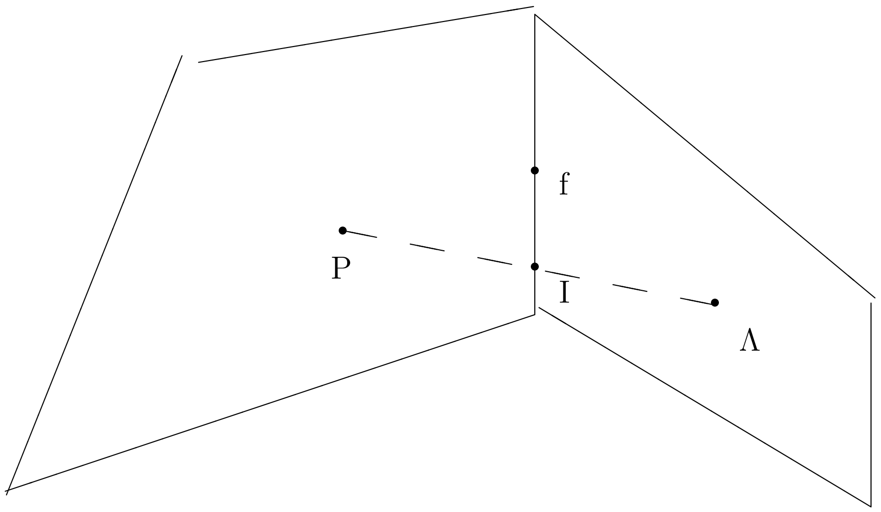

Figure 2.9: Non-conjunctional face.

reduced with a simpler non-conjuctionality correction procedure, as presented in the next section.

2.3.3.2 PHYSICA non-conjunctionality correction

Instead of interpolating nodal values, another non-conjunctionality correction method is to interpolate the face centroid value from another interpolated value from the face point I lying along the line joining the two adjacent nodes.

$$\phi_f = \phi_I + \boldsymbol{d}_{If} \cdot \nabla\phi \tag{2.26}$$

where $\boldsymbol{d}_{If}$ is the vector from I to the face centre.

Special care has to be taken when computing the gradients $\nabla\phi$, since they are themselves calculated from face values. A simple method for resolving this dependence is to store the previous gradients and use them to calculate the current iteration gradient, as follows [17]:

$$\left(\frac{\partial\phi}{\partial x}\right)^m = \frac{1}{V}\sum_f S_f n_x \left(\phi_I^m + \sum_j \left[\frac{\partial\phi}{\partial x_j}\right]_f^{m-1} (d_{If})_j\right) \tag{2.27}$$

2.4 Extreme cases

In some cases, a severe mesh defect may be causing divergence or a wrong solution in part of the domain. Sometimes, these defects are difficult to correct, mainly in meshes of complex geometries generated automatically using commercial packages (e.g. Harpoon [67]) and time is wasted when trying to manually adjust the position of some points in the mesh by trial and error. This section introduces a radical method to deal with such rogue cells. The methods described in this section are used only when diffusion corrections described in the previous sections are not sufficient to achieve convergence.

2.4.1 Variable interpolation in divergent cells

One way of dealing with rogue cells is by ignoring the standard solution procedure in these control volumes, and interpolating for solved variables using neighbouring cell values. Interpolation on an unstructured mesh is non trivial since data is not ordered in a tabular form and polynomial type interpolations cannot be directly implemented. Furthermore, data ordering in structured form is impossible in mixed-mode meshes comprising of both triangular and quadrilateral (or other polygonal) faces. Franke [68] listed interpolation methods available for unstructured data, and of particular interest is the inverse square distance interpolation method: Shepard's algorithm [69].

2.4.1.1 Shepard's method

The general inverse square distance interpolation method calculates an interpolated variable ϕ at a particular point using

$$\phi = \frac{\sum\limits_{neighbours} \frac{1}{r_i^2}\phi_i}{\sum\limits_{neighbours} \frac{1}{r_i^2}} \tag{2.28}$$

where r_i is the distance between the point where the interpolation value is required and the point i where the existing variable ϕ_i is picked up. In the context of a divergent cell, the weighing factor $\frac{1}{r_i^2}$ is the inverse square of the distance between the bad cell centre and the centre of a neighbouring cell i. The neighbouring cells include the NE, NW, SE and SW cells in 2D, and the 8 diagonal cells in 3D, if an interpolated cell is located in the middle of the computational domain. Interpolated control volumes at boundaries have less neighbours.

The algorithm used for handling bad cells using interpolation is illustrated by the flowchart in Figure 2.10. The main steps in the procedure are:

- Mass imbalance in a control volume is to be used as an indicator for invoking the interpolation routine. After solving the discretised momentum equations, the mass defect in each cell is computed using the resulting face velocities. The average mass defect in the whole domain for the iteration is then computed.
- The discrimination criterion for determining if a control volume is to be interpolated or not is the following: cells with a mass defect higher than the product of the average mass imbalance for the iteration and a certain factor are chosen for interpolation. This factor is determined by trial and error, and is case specific; a test case is first run without interpolation and the magnitude of mass imbalances within the domain are examined. The threshold is then chosen to one significant figure such that the pressure corrections in only a few cells are interpolated within each iteration.
- The pressure corrections in these control volumes are indirectly interpolated in the following manner: the new value for pressure in a rogue cell is interpolated from neighbouring last pressure values. The new pressure correction is computed as the difference between the interpolated pressure in the cell and the last pressure value in the same cell.
- In cases where there are rogue cells which are neighbours to each other, pressure values from adjacent rogue cells are skipped by the interpolation process. This technique therefore excludes cases involving large clusters of divergent control volumes, and can only be used for isolated misbehaving cells.

The procedure is inherently not mass conserving. It is an extreme measure used to artificially stop the solver from diverging in particularly bad cells in the hope that

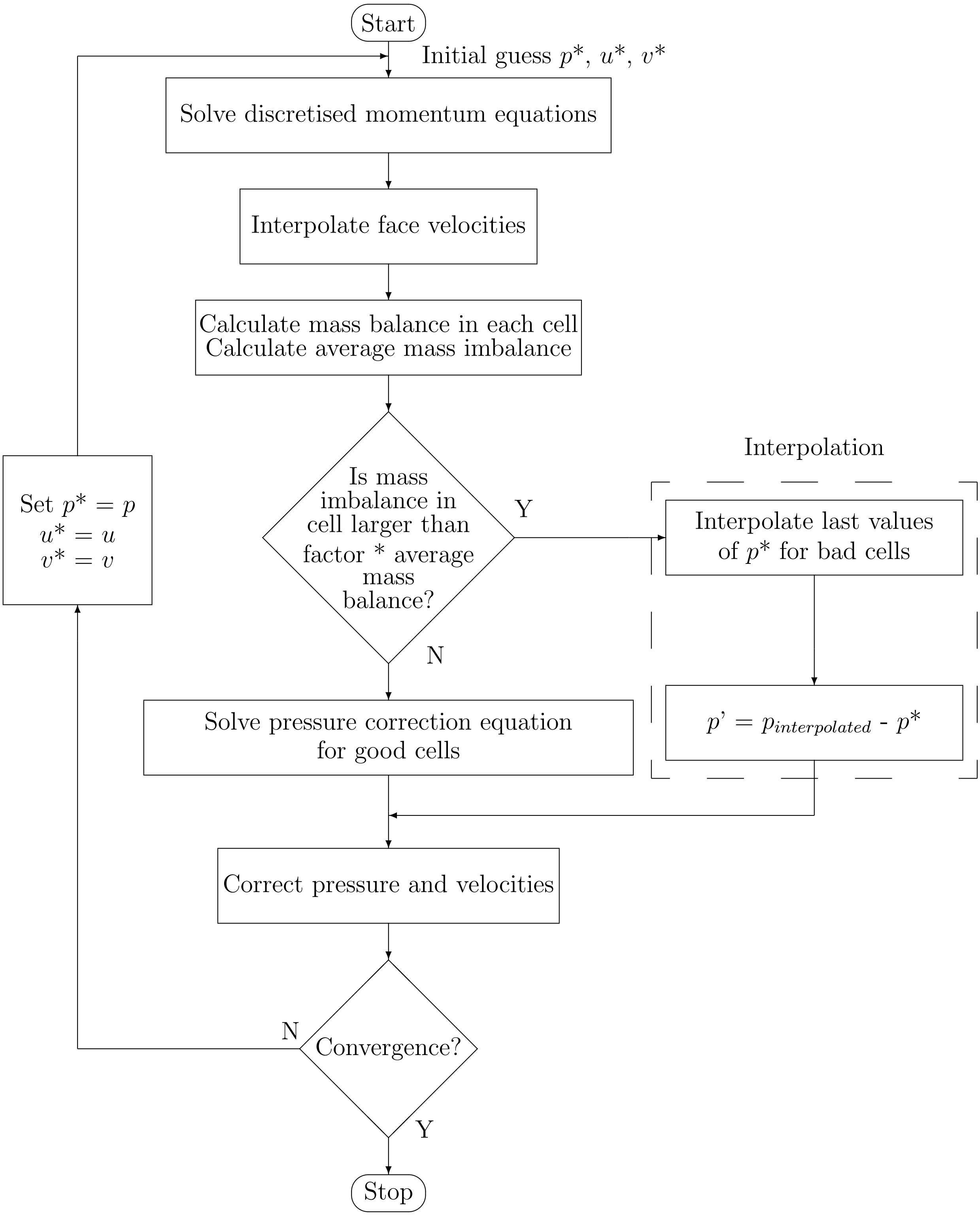
Start
Initial guess $p*$, $u*$, $v*$
Solve discretised momentum equations
Interpolate face velocities
Calculate mass balance in each cell
Calculate average mass imbalance
Is mass imbalance in cell larger than factor * average mass balance?
Y
N
Interpolation
Interpolate last values of $p*$ for bad cells
$p' = p_{interpolated} - p*$
Solve pressure correction equation for good cells
Correct pressure and velocities
Convergence?
N
Y
Stop
Set $p* = p$
$u* = u$
$v* = v$

Figure 2.10: Algorithm for pressure interpolation.

a solution can be recovered away from the interpolation region. In order to freeze residual growth, a variable bounding procedure as described in the next section is to be used in conjunction with the suggested interpolation method.

2.4.2 Bounding variables within a physically realistic range

A reasonable practice in Computational Fluid Dynamics is to limit the values of solved variables within a range imposed by the physics of the problem considered. These limits on variables prevent spurious oscillations during the calculation and help the code to converge to a physical value. Such practice can also used to handle convergence issues in cells with large mass defects: ignoring the results of the solver in these cells and fix the solved variables to a maximum or minimum value. An implementation of variable bounding is already present in the in-house code [15].

2.5 Orthogonality test cases

The results of the non-orthogonality investigations are described and discussed in this section. The mesh used in each case is presented and described, together with the non-orthogonality measure presented in Section 2.1.3. This measure has been chosen since fluxes are evaluated on faces in the finite volume method, and a face orthogonality indicator is more appropriate than a nodal indicator.

The non-orthogonal test cases are then run using the diffusion correction methods described in Sections 2.3.2.3 and 2.3.2.4. If these corrections do not work in obtaining a converged solution, the interpolation method of Section 2.4.1.1 and variable bounding method of Section 2.4.2 are used as last resort. Each result is compared with either analytic solutions (if available) or benchmarks from the literature. The convergence behaviour of each case, including mass balance plots, is also described and discussed.

2.5.1 Poisseuille flow

The first test case considered is the Poisseuille flow between two infinitely wide parallel plates separated by 0.01 m. The analytic solution to this problem is known: the

Table 2.1: Boundary conditions for Poisseuille flow case

Inlet	Pressure	$1.0 \times 10^5 + 1.0 \times 10^2$ Pa for Re = 1 $1.0 \times 10^5 + 1.0 \times 10^4$ Pa for Re = 100
Outlet	Pressure	1.0×10^5 Pa
Walls	Velocity	0 m s^{-1}

pressure profile decreases linearly downstream and the velocity profile is parabolic if the flow is laminar and fully developed, being zero at the plates and maximum at the mid-plane between the plates. The analytic velocity profile is given by

$$u = \frac{p_{in} - p_{out}}{2\mu L}\left(hz - z^2\right) \tag{2.29}$$

where p_{in} is the inlet pressure, p_{out} is the outlet pressure, h is the distance between the two plates and L is the length of the plates.

The aim of this series of test is to examine the existing code's behaviour in the face of non-orthogonality. This is done with the following approach:

1. A perfectly orthogonal mesh is first used to define the problem and verify if the code yields the analytic solution.
2. Defects are introduced in parts of the mesh to verify if the solution is recovered away from defects.
3. A large portion of the mesh is modified by moving points in the middle of the domain in a random manner, resulting in a large section where non-orthogonality is present.

The flow in the Poisseuille cases is perfectly laminar with boundary conditions chosen so that the Reynolds number is always well below 10^5. Two different range of Peclet numbers are chosen: Pe ~ 1, in which Re = 1, and ~ 100, for which Re = 100. In the cases with Pe ~ 1, diffusion is dominant and the central differencing scheme is used to solve for flow. For Pe ~ 100, the flow is convection driven and the predictions should be better than for Pe ~ 1.

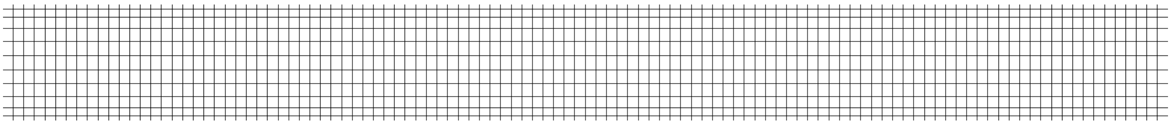

Figure 2.11: Orthogonal mesh for Poisseuille case.

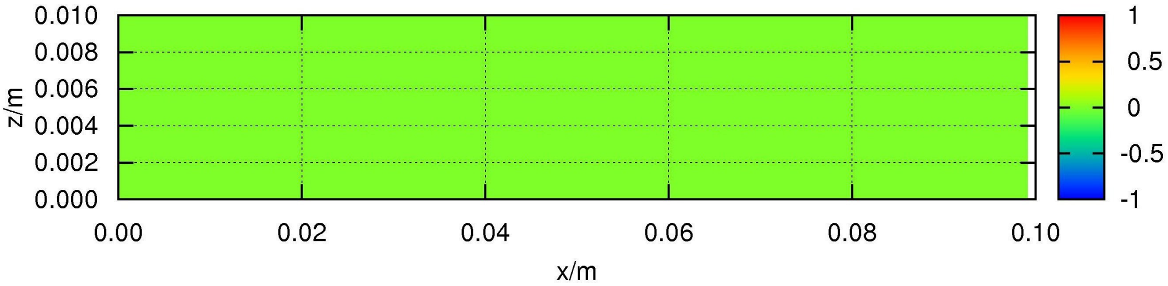

Figure 2.12: Mesh quality measure for orthogonal case.

Orthogonal mesh A 2D case is run at first with with a perfectly orthogonal mesh depicted in Figure 2.11 for two maximum Reynolds numbers, 1 and 100, to validate the numerical method. The mesh quality measure is shown in Figure 2.12. The boundary conditions for each case is described in Table 2.1. A no-slip velocity condition is imposed on each wall. The initial conditions are: pressure $= 1.0 \times 10^5$ Pa and velocity is set to the maximum analytical value in each node of the domain.

The fluid properties used are: density $\rho = 1.0$ kg m^{-3} and kinematic viscosity $\nu =$ 1.0 $\times 10^{-2}$ m^2 s^{-1}. The pressure difference across the 0.10 m long domain is 1.0 $\times 10^2$ Pa for the Re = 1.0 case, and 1.0 $\times 10^4$ Pa for the Re = 100 case. No under-relaxation factor is used for momentum in these cases. Pressure corrections are relaxed by a factor of 0.2. A momentum *false-time step* is added to the momentum equations to make the transport equation hyperbolic.

With this choice of fluid properties, the Reynolds numbers are of the same magnitudes as the velocity components in the direction of the flow, as shown in Figures 2.13 and 2.14. The Peclet numbers defined by equation (1.13) are depicted in Figures 2.15 and 2.16. In the Re = 100 case, the strength of convection is greater than diffusion, but for Re = 1, diffusion dominates at the walls, and is of comparable magnitude at

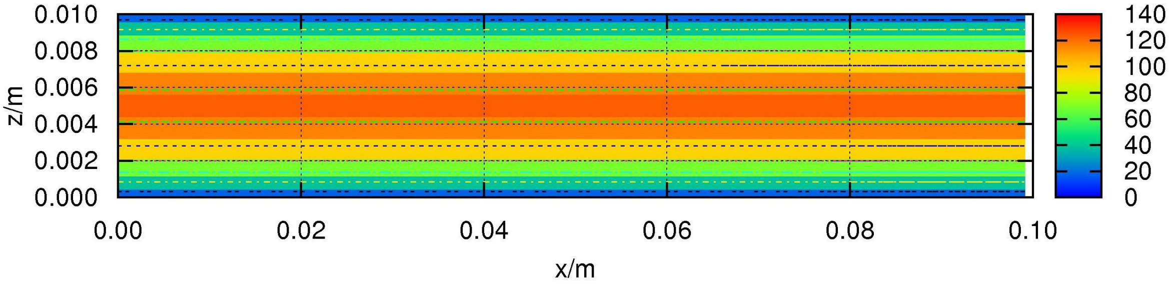

Figure 2.13: Reynolds number in Poisseuille flow case with orthogonal mesh and Re = 100.

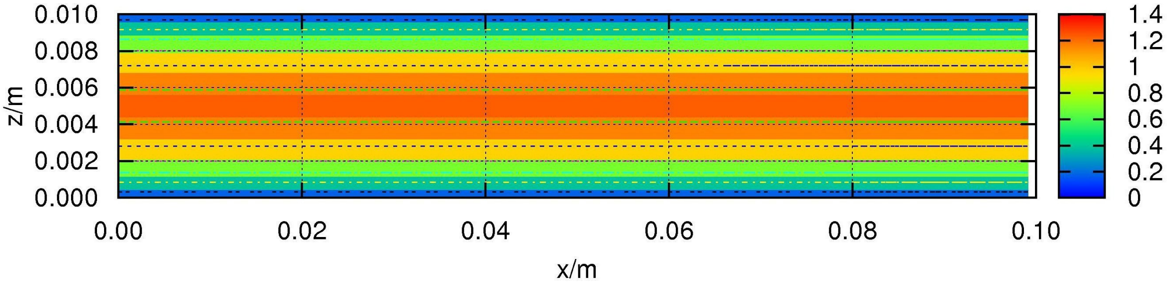

Figure 2.14: Reynolds number in Poisseuille flow case with orthogonal mesh and Re = 1.

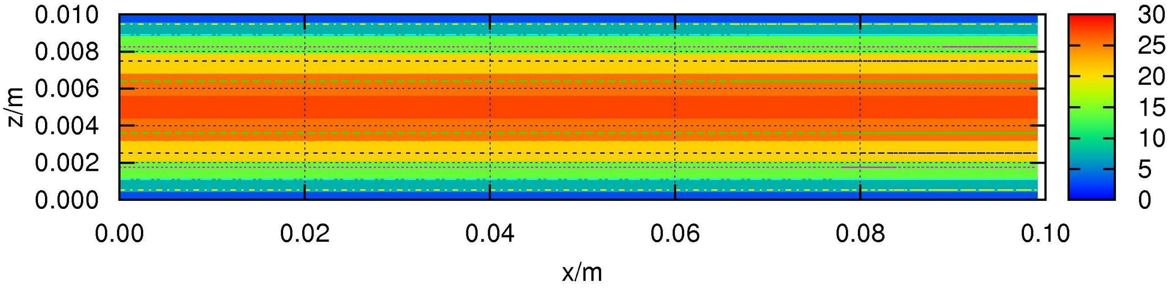

Figure 2.15: Peclet number contour for Poisseuille flow case with orthogonal mesh and Re = 100.

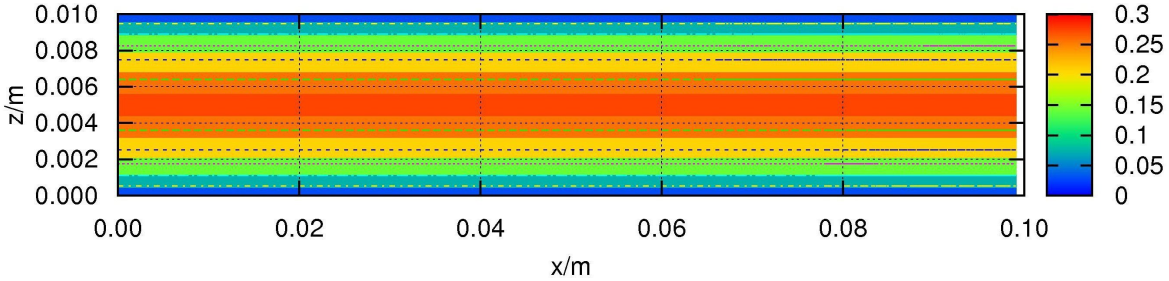

Figure 2.16: Peclet number contour for Poisseuille flow case with orthogonal mesh and Re = 1.

the centre of the domain.

The analytic solution is recovered in each case, as shown in Figures 2.17 and 2.19. The velocity contours for both cases are given in Figures 2.18 and 2.20. The analytic maximum of $u_{max} = 100$ ms^{-1} for Re = 100 and $u_{max} = 1$ ms^{-1} for Re = 1 are recovered. The linear pressure profile is also obtained as shown in Figures 2.21 and 2.22.

The mass imbalance for both cases is very small, as shown in Figures 2.23 and 2.24: both cases have converged.

Mesh with points moved The orthogonal mesh from the previous tests is then deformed by arbitrarily moving some points, resulting in non-orthogonality within localised areas in the domain as shown in Figure 2.25. The mesh skewness indicator is shown in Figure 2.26. The mesh has been deformed in different portions of the domain to reflect two different possibilities: non orthogonality at the centre where velocity is maximum, and skewness at the boundaries.

Again the choice of fluid properties renders the Reynolds numbers equal in the magnitude to the velocities, as shown in Figures 2.27 and 2.28. The Peclet numbers as defined by equation (1.13) are depicted in Figures 2.29 and 2.30. The minimum residual diffusion correction term of Section 2.3.2.3 is used to achieve convergence in this case.

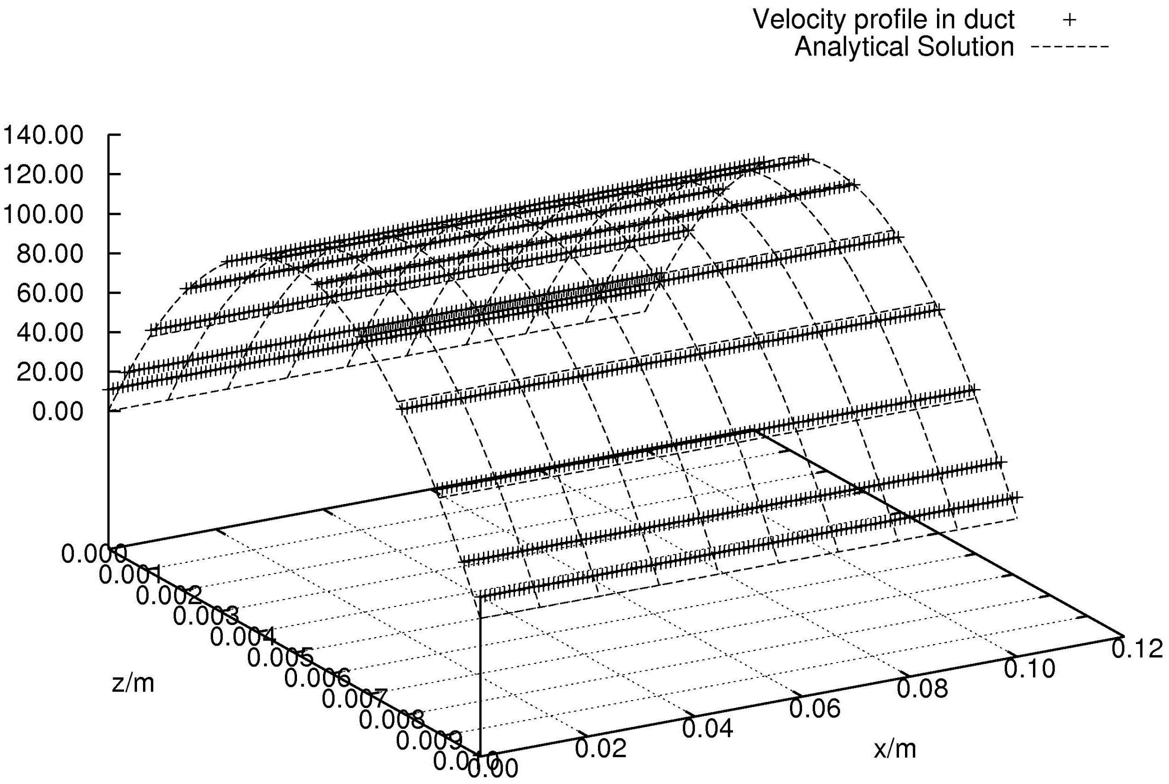

Figure 2.17: Comparison of velocity u in the duct with the analytic solution with orthogonal mesh for Re = 100.

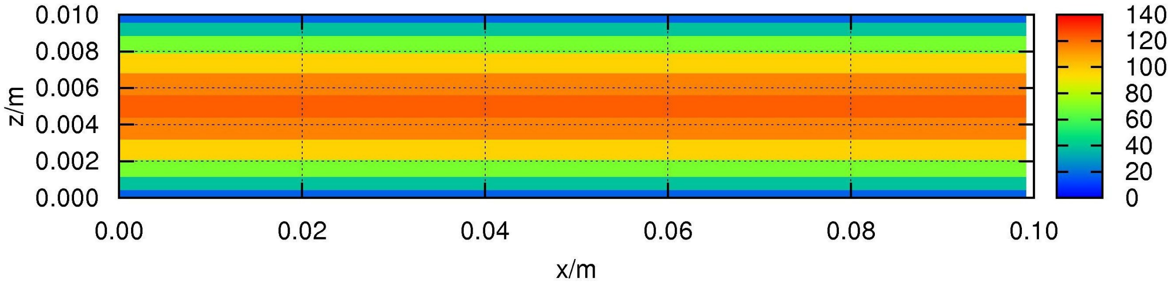

Figure 2.18: Velocity contour of Poisseuille flow case with orthogonal mesh and Re = 100.

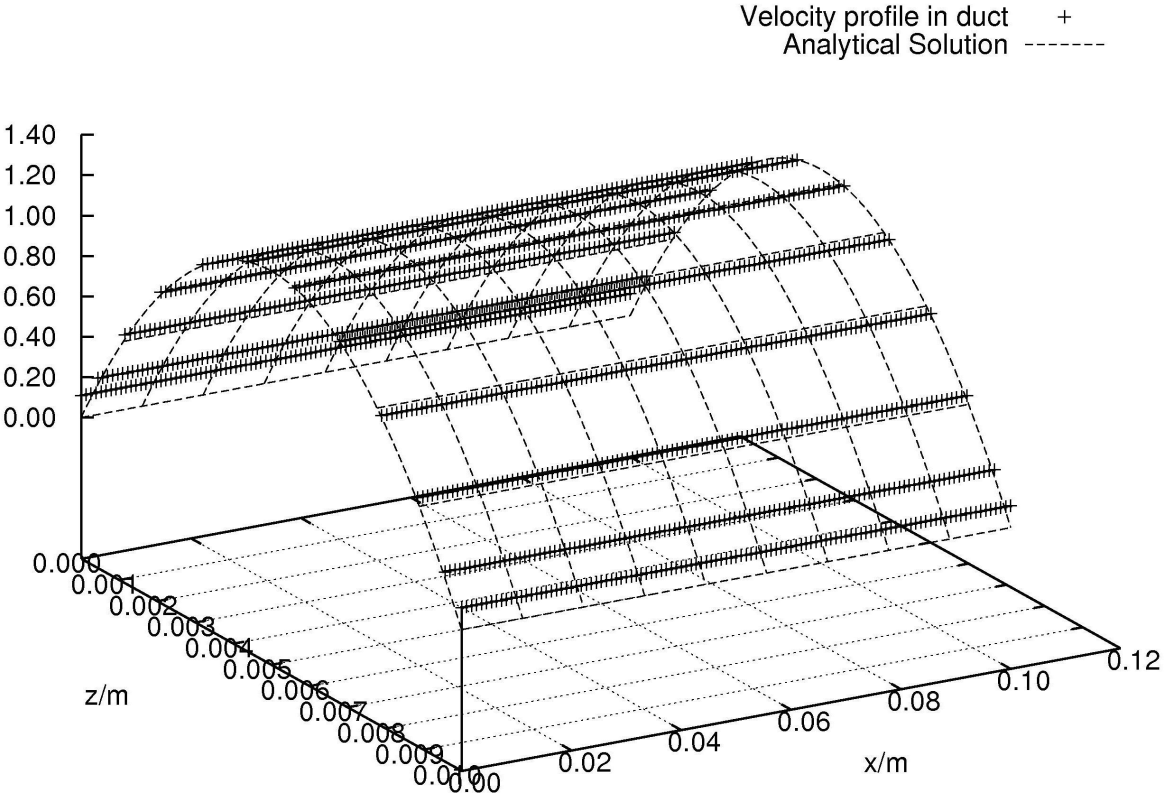

Figure 2.19: Comparison of velocity u in the duct with the analytic solution with orthogonal mesh for Re = 1.

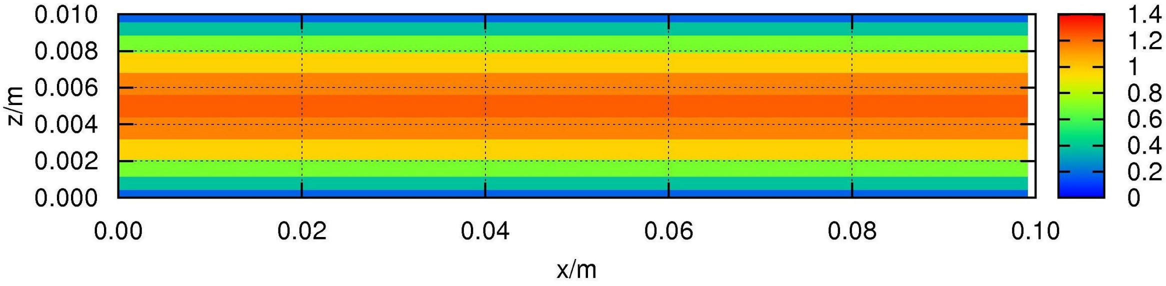

Figure 2.20: Velocity contour of Poisseuille flow case with orthogonal mesh and Re = 1.

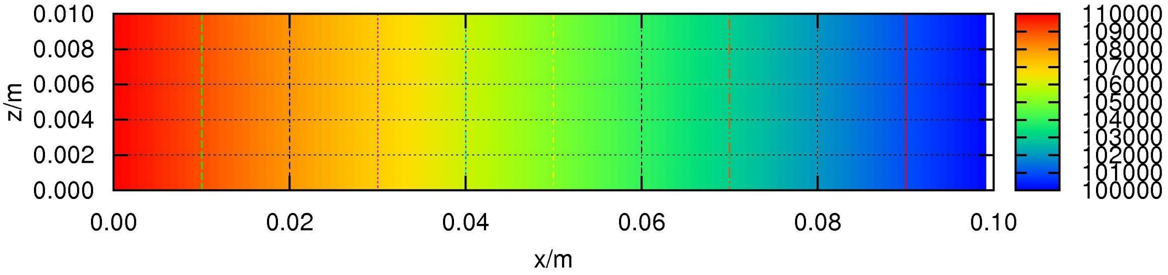

Figure 2.21: Pressure contour of Poisseuille flow case with orthogonal mesh and Re = 100.

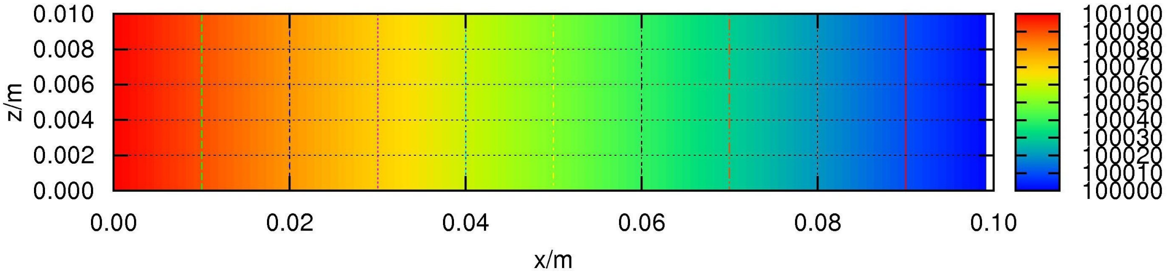

Figure 2.22: Pressure contour of Poisseuille flow case with orthogonal mesh and Re = 1.

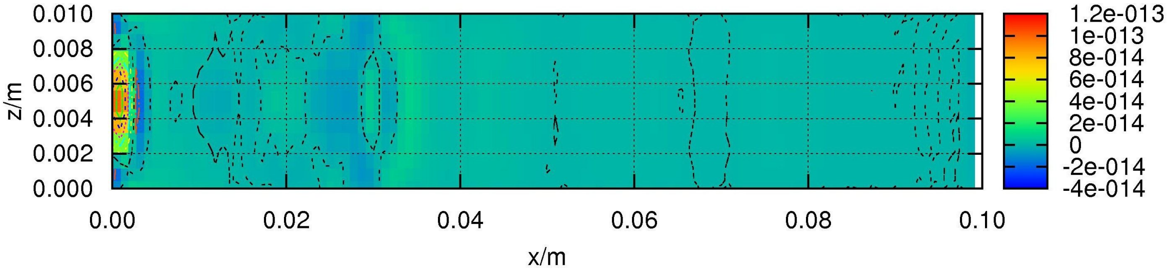

Figure 2.23: Mass contour of Poisseuille flow case with orthogonal mesh and Re = 100.

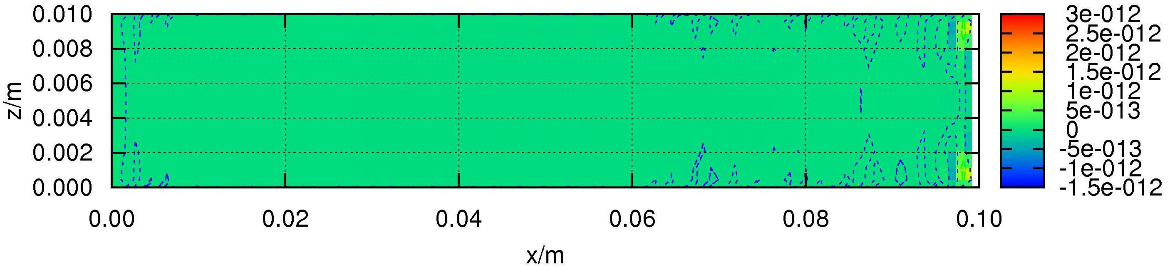

Figure 2.24: Mass contour of Poisseuille flow case with orthogonal mesh and Re = 1.

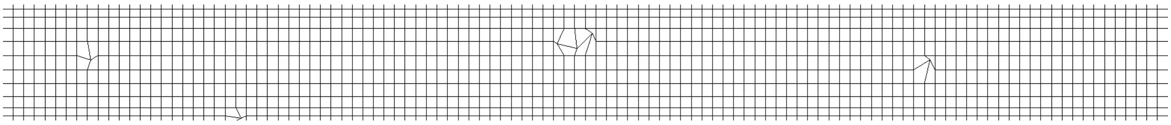

Figure 2.25: Poisseuille mesh with points moved.

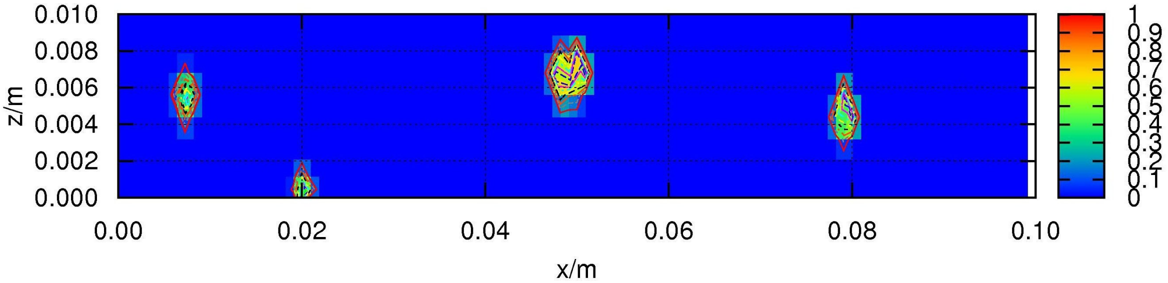

Figure 2.26: Mesh quality measure for points moved case.

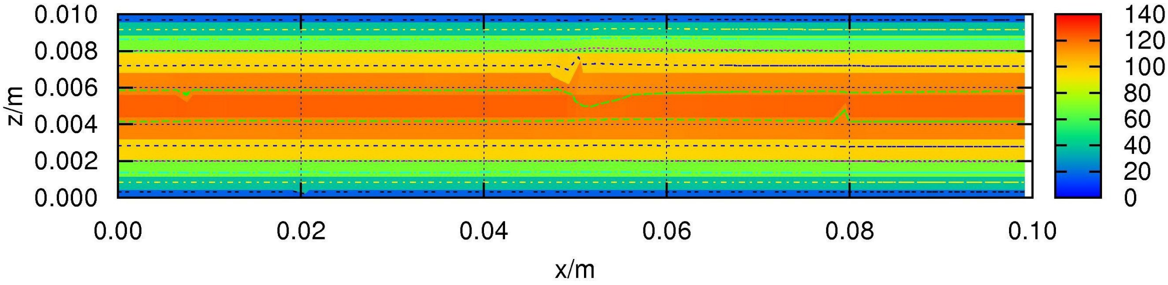

Figure 2.27: Reynolds number for Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 100.

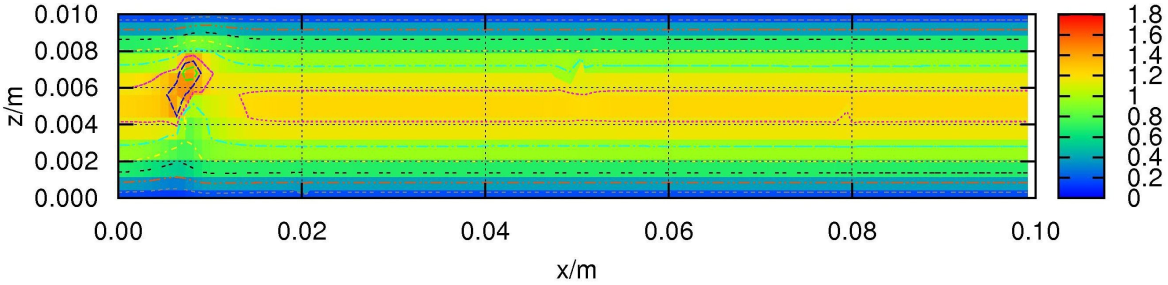

Figure 2.28: Reynolds number for Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 1.

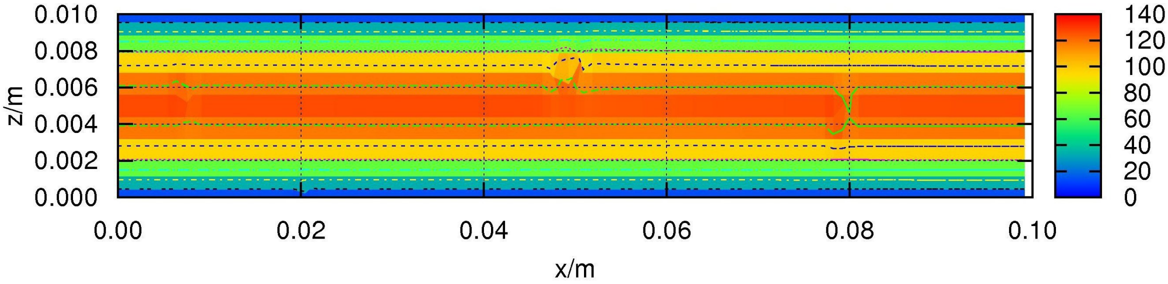

Figure 2.29: Peclet number contour for Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 100.

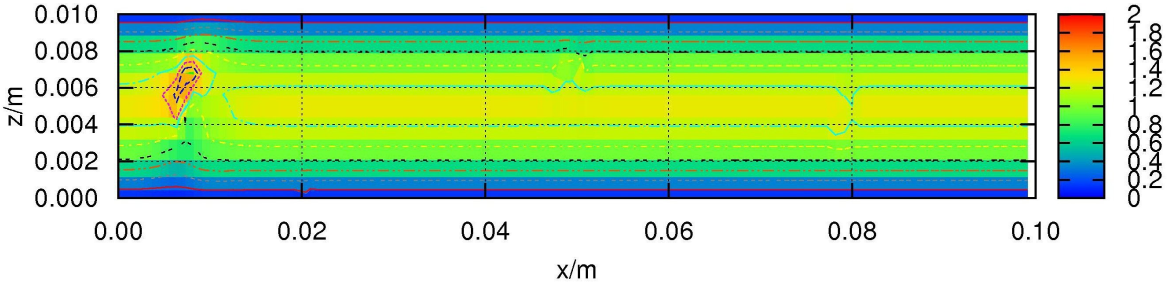

Figure 2.30: Peclet number contour for Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 1.

A similar parabolic profile is recovered, as shown in Figures 2.31 and 2.33 The velocity contours are shown in Figures 2.32 and 2.34. The pressure profile are given by Figures 2.35 and 2.36.

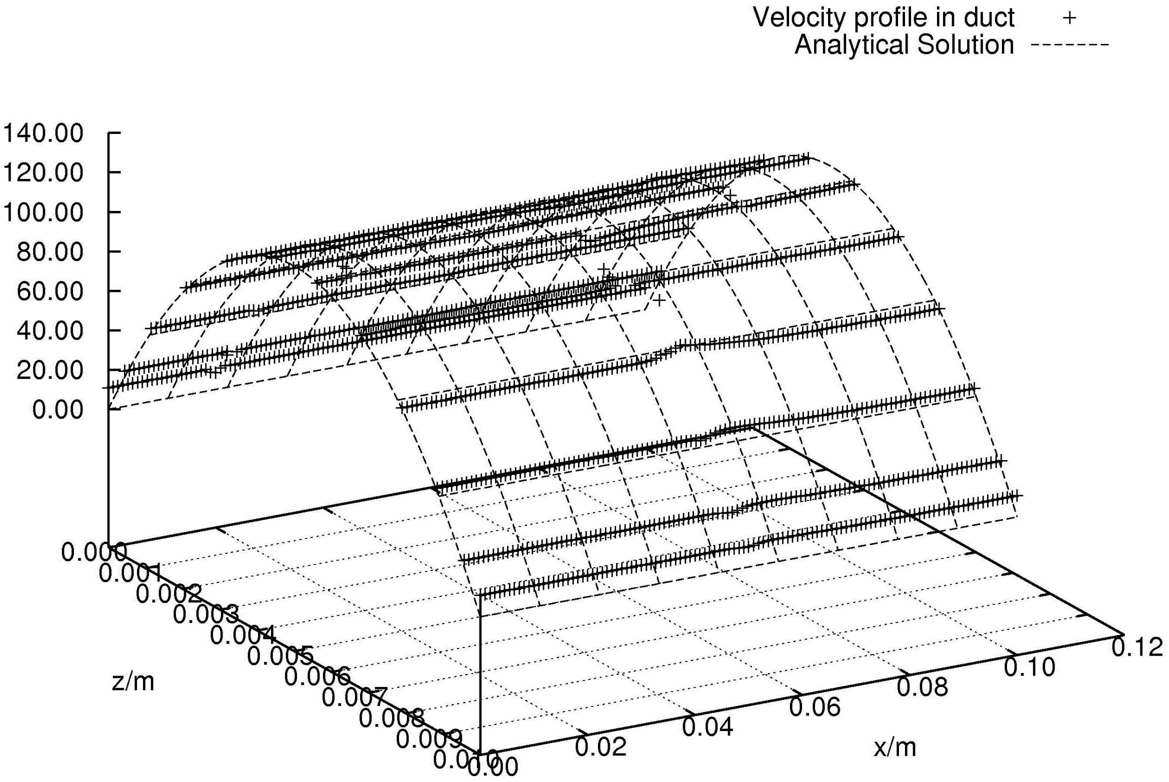

Figure 2.31: Comparison of velocity u in the duct with the analytic solution in deformed mesh with points arbitrarily moved in different regions of the domain for Re = 100.

The inaccuracies in the solution are localised at the mesh deformations, as shown in the mass imbalance plot in Figures 2.37 and 2.38.

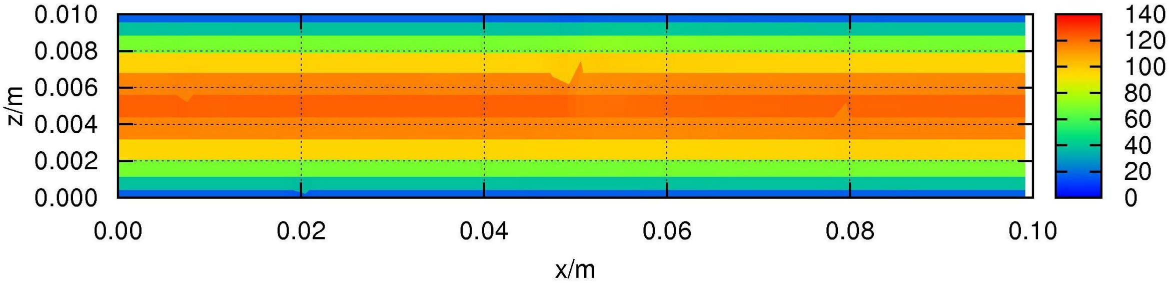

Figure 2.32: Velocity contour of Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 100.

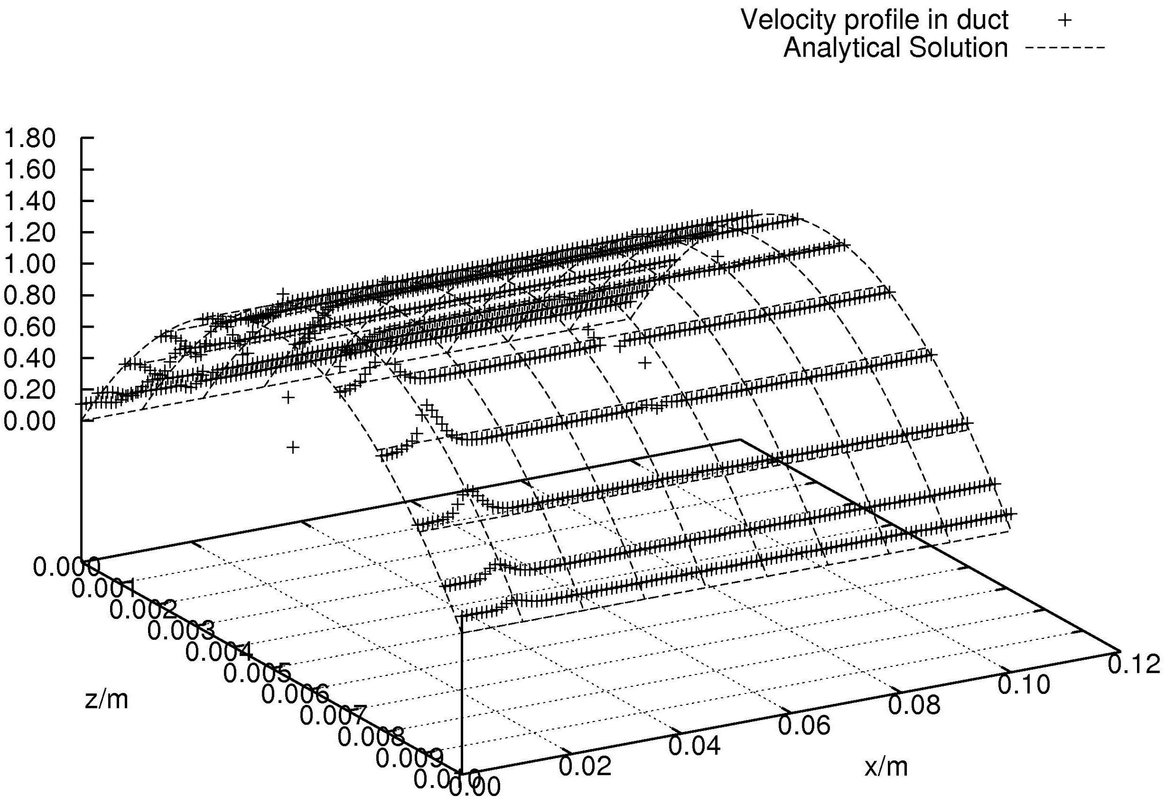

Figure 2.33: Comparison of velocity u in the duct with the analytic solution in deformed mesh with points arbitrarily moved in different regions of the domain for Re = 1.

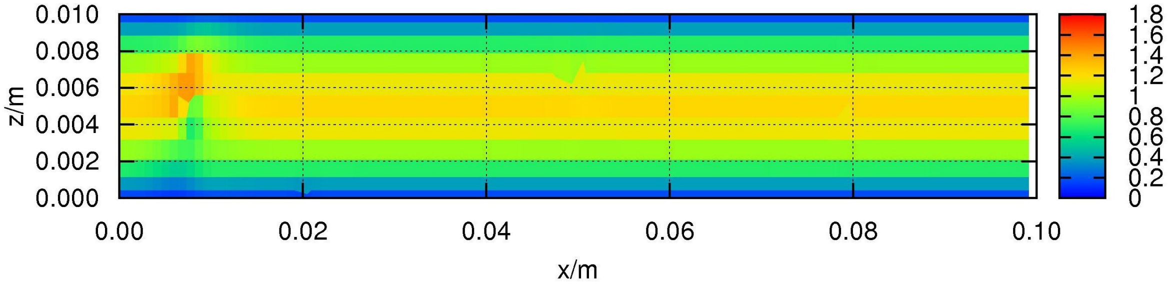

Figure 2.34: Velocity contour of Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 1.

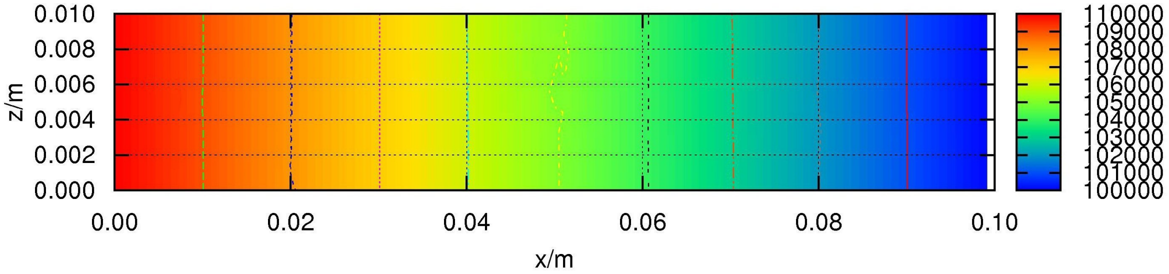

Figure 2.35: Pressure contour of Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 100.

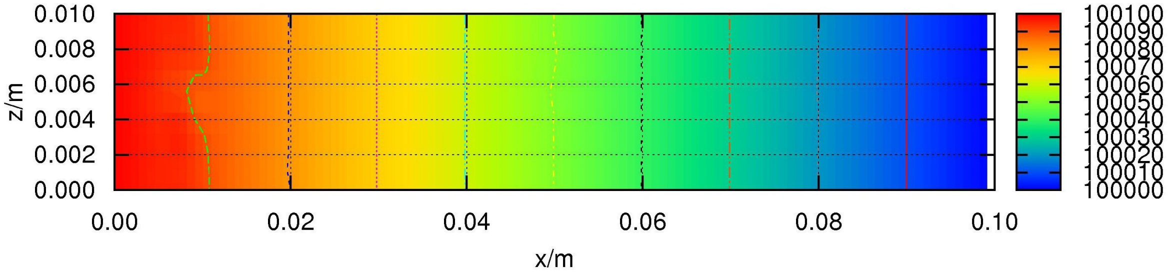

Figure 2.36: Pressure contour of Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 1.

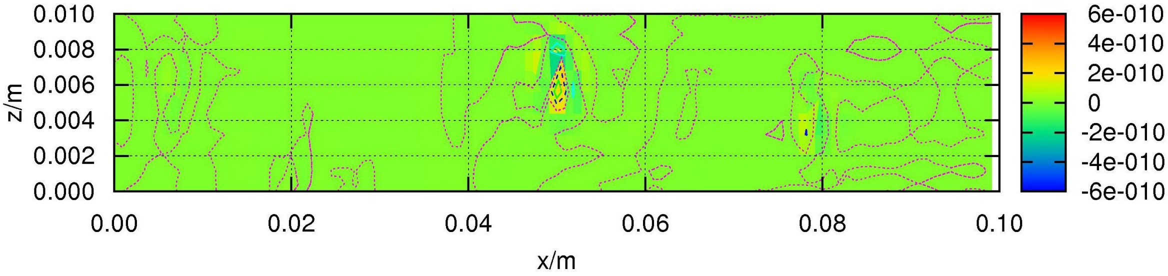

Figure 2.37: Mass contour of Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 100.

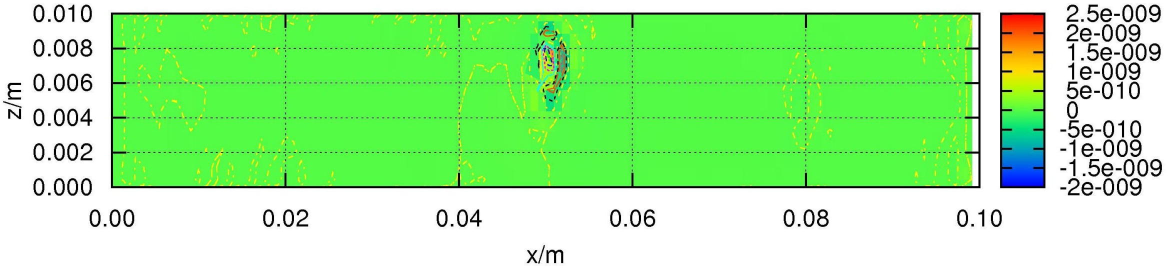

Figure 2.38: Mass contour of Poisseuille flow case with deformed mesh with points arbitrarily moved in different regions of the domain and Re = 1.

Randomly deformed mesh The mesh is further tampered with by introducing random deformations, illustrated by Figure 2.39, in x and z within a central portion of the domain. The deformations in z are, on average, 10 times smaller than those in x since the ratio of $x : z = 10 : 1$ for the domain. The mesh orthogonality measure is given in Figure 2.40. The extent for skewness is over a larger region and cells in the middle of the randomly deformed area are surrounded by other non-orthogonal cells. This behaviour is studied.

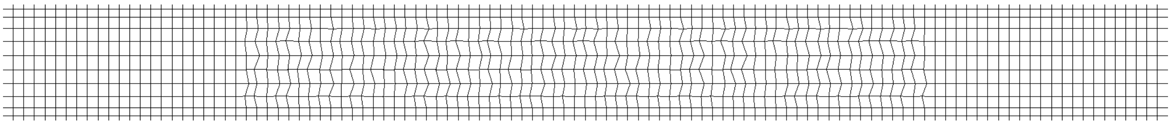

Figure 2.39: Poisseuille mesh with random deformations.

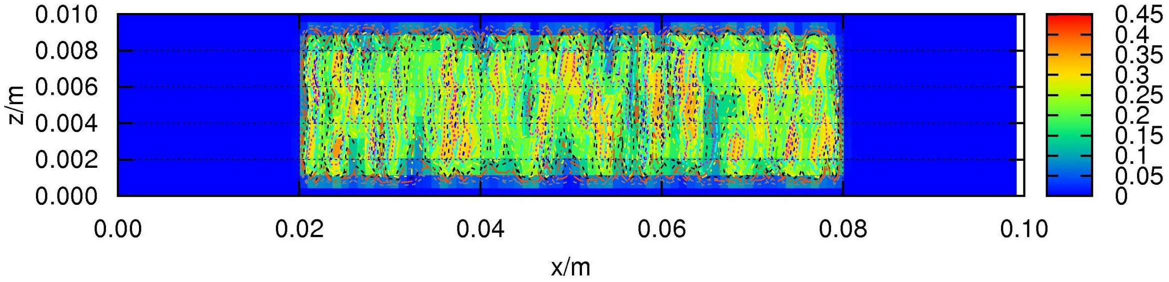

Figure 2.40: Mesh quality measure for case with randomly deformed mesh.

With the same choice of fluid properties, the Reynolds numbers are numerically equal to the velocity magnitudes, as shown in Figures 2.41 and 2.42. The Peclet numbers as defined by equation (1.13) are depicted in Figures 2.43 and 2.44. The cases are run with the diffusion correction term of Section 2.3.2.3.

The analytical solution is also recovered in the case with Re = 100, cf Figures 2.45, 2.46 and 2.49, with larger inaccuracies next to mesh deformations. However, for the case with Re = 1 corresponding to Pe $\sim$ 1 - where the strength of diffusion is more significant than convection, the case converges to a wrong solution as shown in Figures 2.47, 2.48 and 2.50. Diffusion non-orthogonality correction from section 2.3.2.3

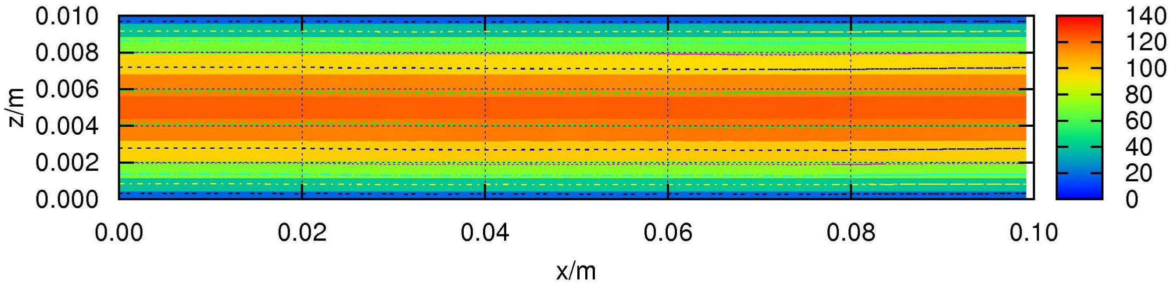

Figure 2.41: Reynolds number for Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 100.

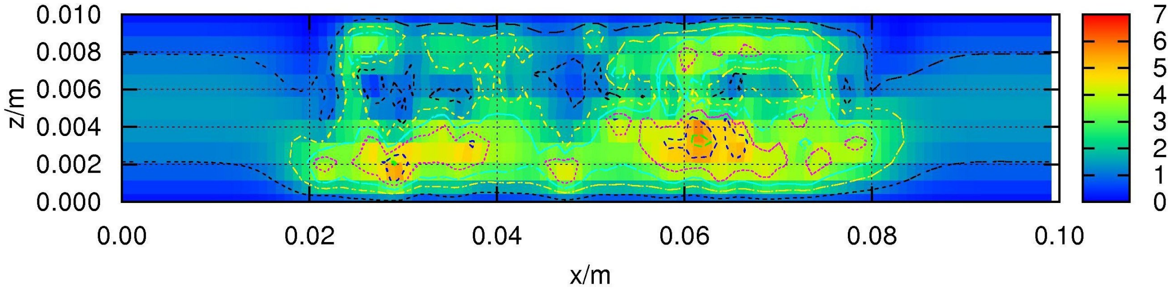

Figure 2.42: Reynolds number for Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 1.

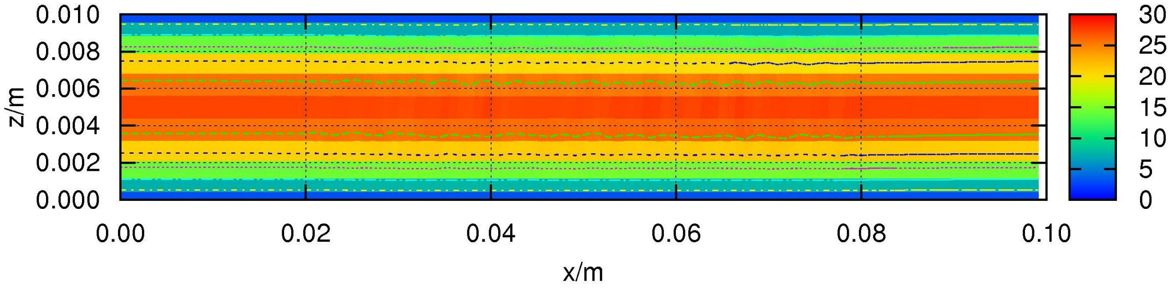

Figure 2.43: Peclet number contour for Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 100.

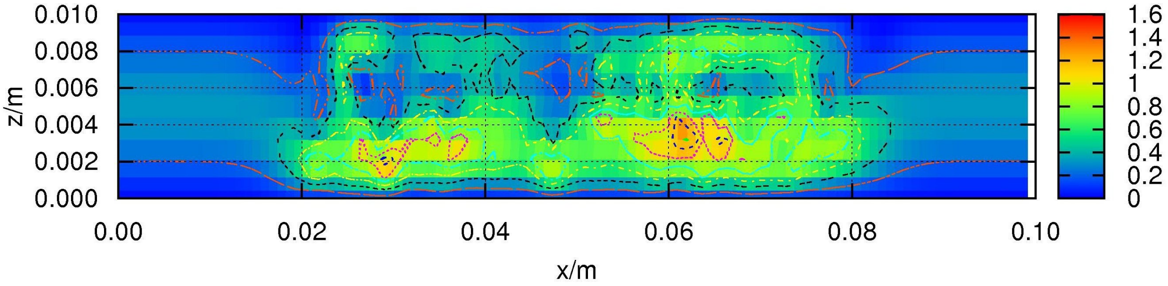

Figure 2.44: Peclet number contour for Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 1.

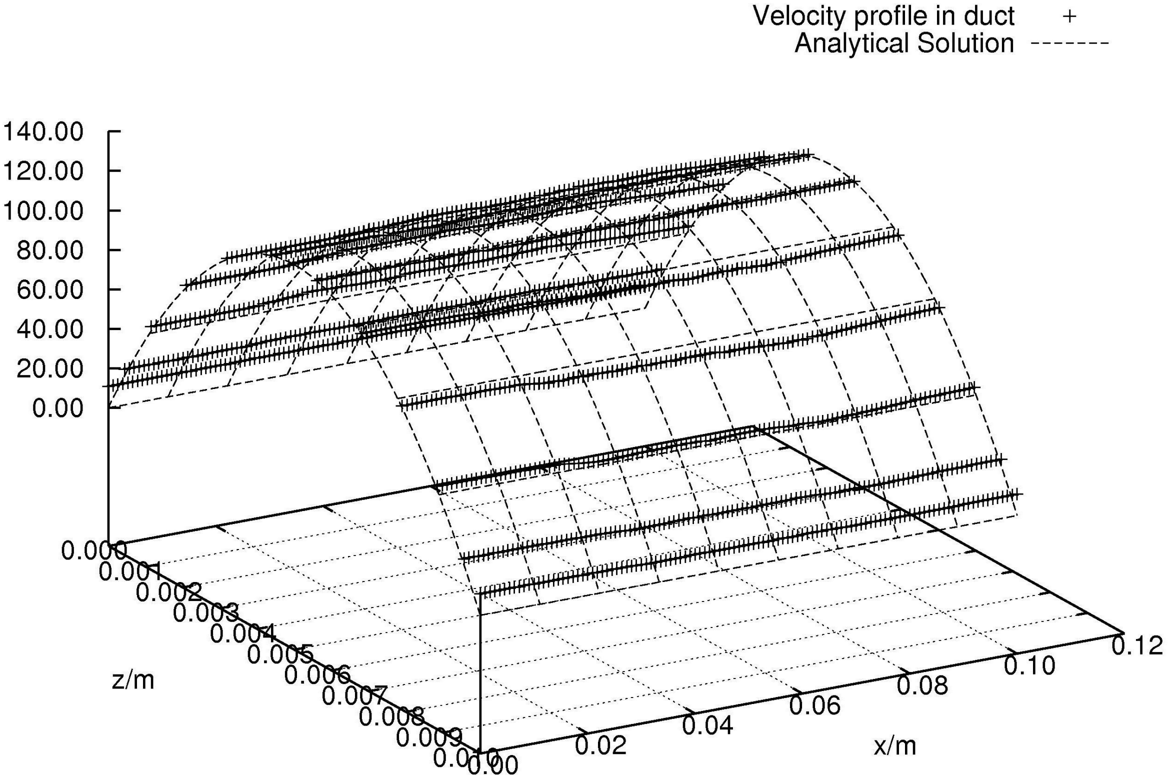

Figure 2.45: Comparison of velocity u in the duct with the analytic solution in randomly deformed mesh for Re = 100.

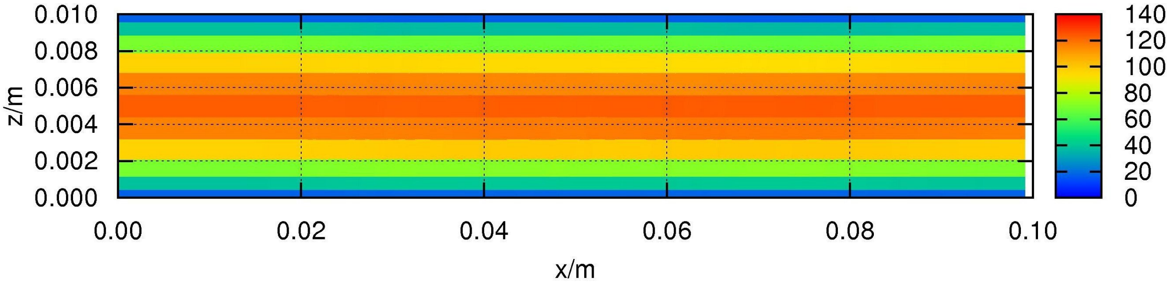

Figure 2.46: Velocity contour of Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 100.

therefore results in inaccurate solutions for diffusion dominated problems in presence of strong, random non-orthogonality in a large cluster of cells.

The inaccuracies seem to be localised at the mesh deformations, as shown in the mass imbalance plot in Figure 2.51 and 2.52.

The line plots in Figures 2.53 and 2.54 depict the effect of the mesh on the results. The effect of non-orthogonality is severe for cases with low Peclet number, which which non-orthogonality correction does not yield the analytic solution.

Convergence behaviour of Poisseuille cases Figures 2.55 and 2.56 show the convergence behaviour for the three cases. The convergence behaviour worsens as the mesh is further deformed. The residuals remain high when the diffusion non-orthogonality correction term is invoked, even though an accurate velocity profile has been recovered in most cases, without having recourse to variable bounding or interpolation.

Flows where diffusion is dominant, i.e. where Pe $\sim$ 1, have higher residuals than cases where Pe $\sim$ 100. This is expected since non-orthogonal errors arise mostly in flows where the magnitude of diffusion coefficients, which are not accurately evaluated, are comparable with those of convection coefficients.

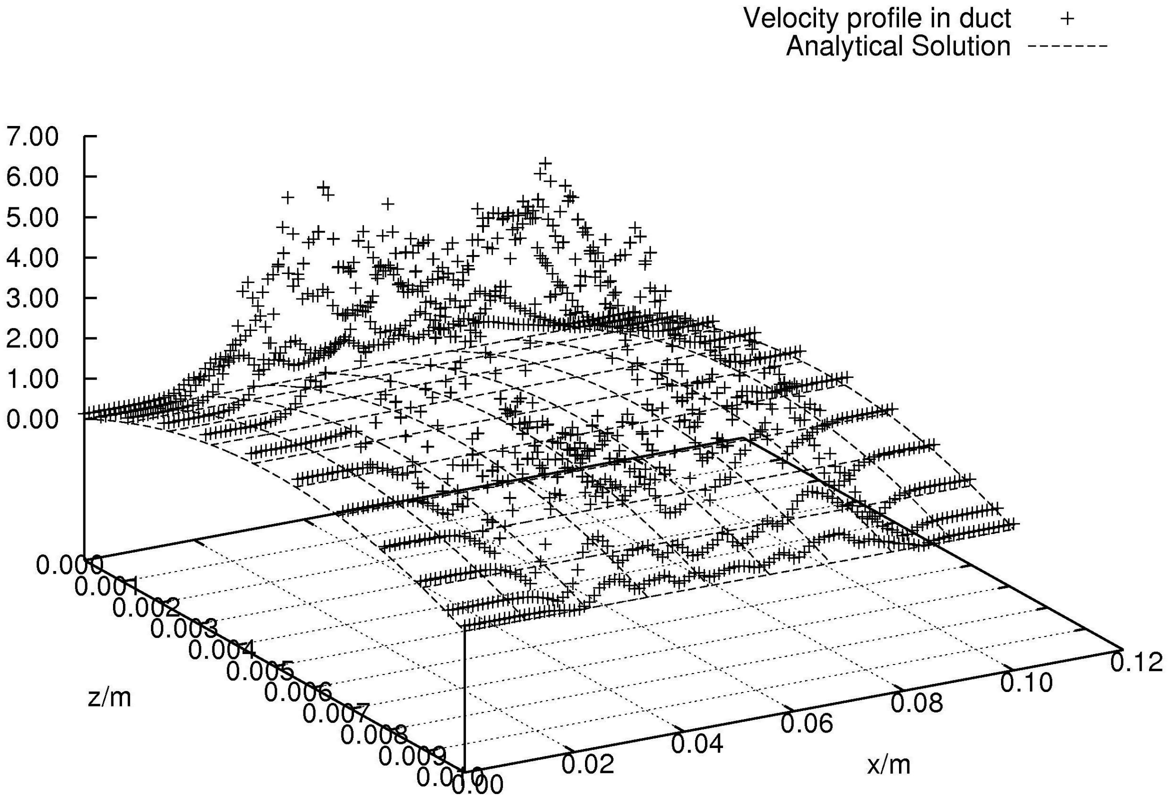

Figure 2.47: Comparison of velocity u in the duct with the analytic solution in randomly deformed mesh for Re = 1.

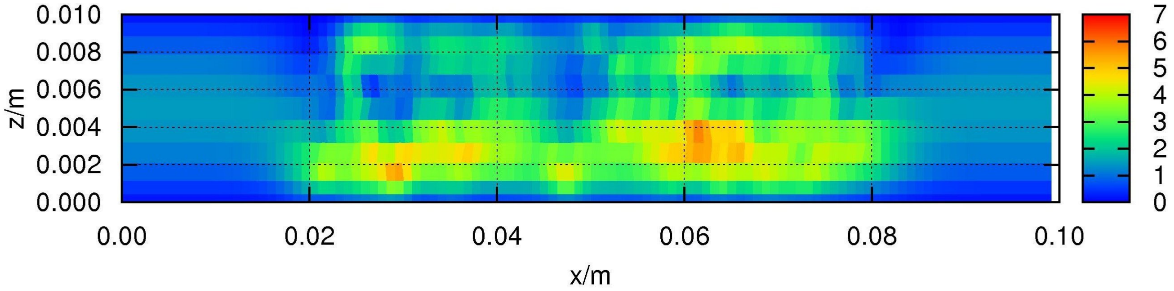

Figure 2.48: Velocity contour of Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 1.

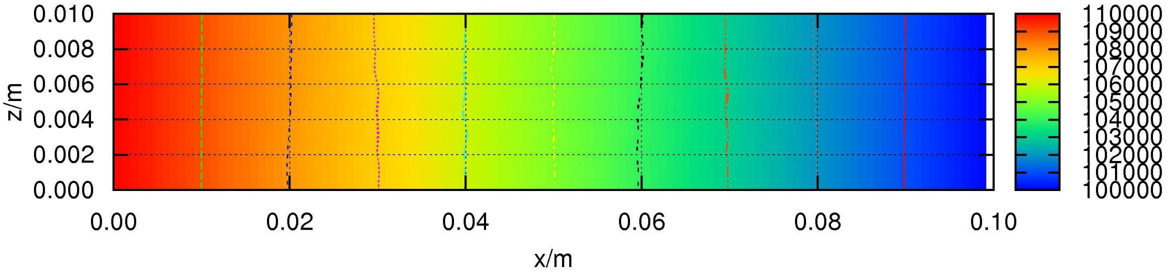

Figure 2.49: Pressure contour of Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 100.

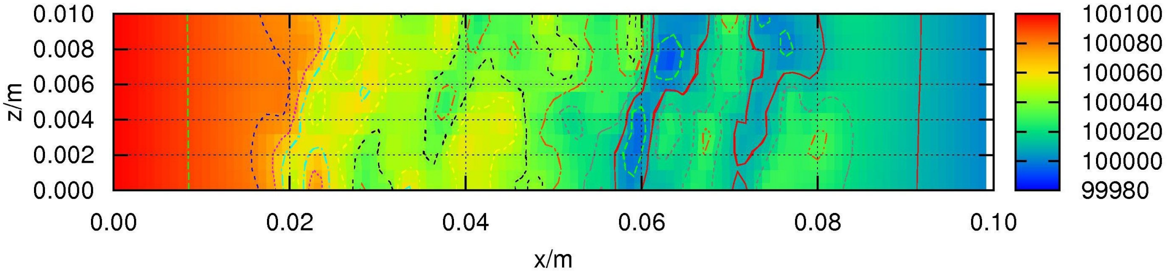

Figure 2.50: Pressure contour of Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 1.

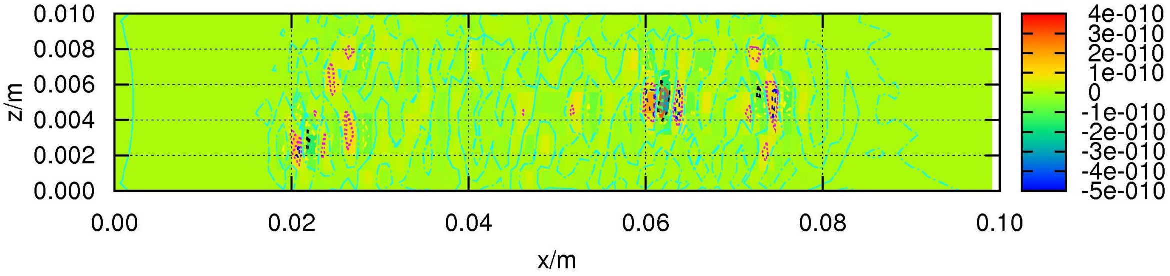

Figure 2.51: Mass contour of Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 100.

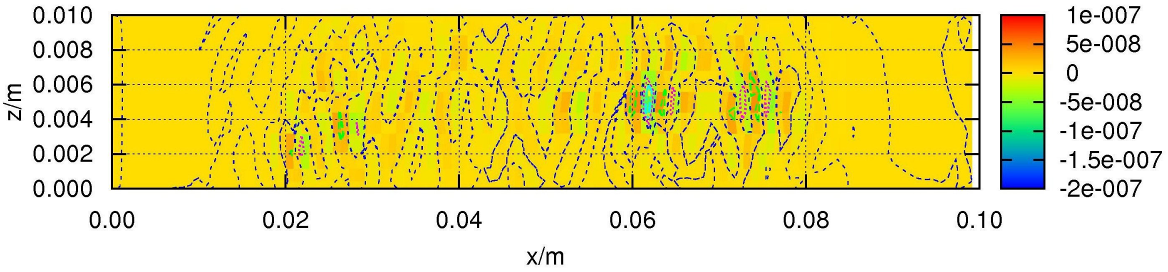

Figure 2.52: Mass contour of Poisseuille flow case with deformed mesh with points randomly moved in centre of the domain and Re = 1.

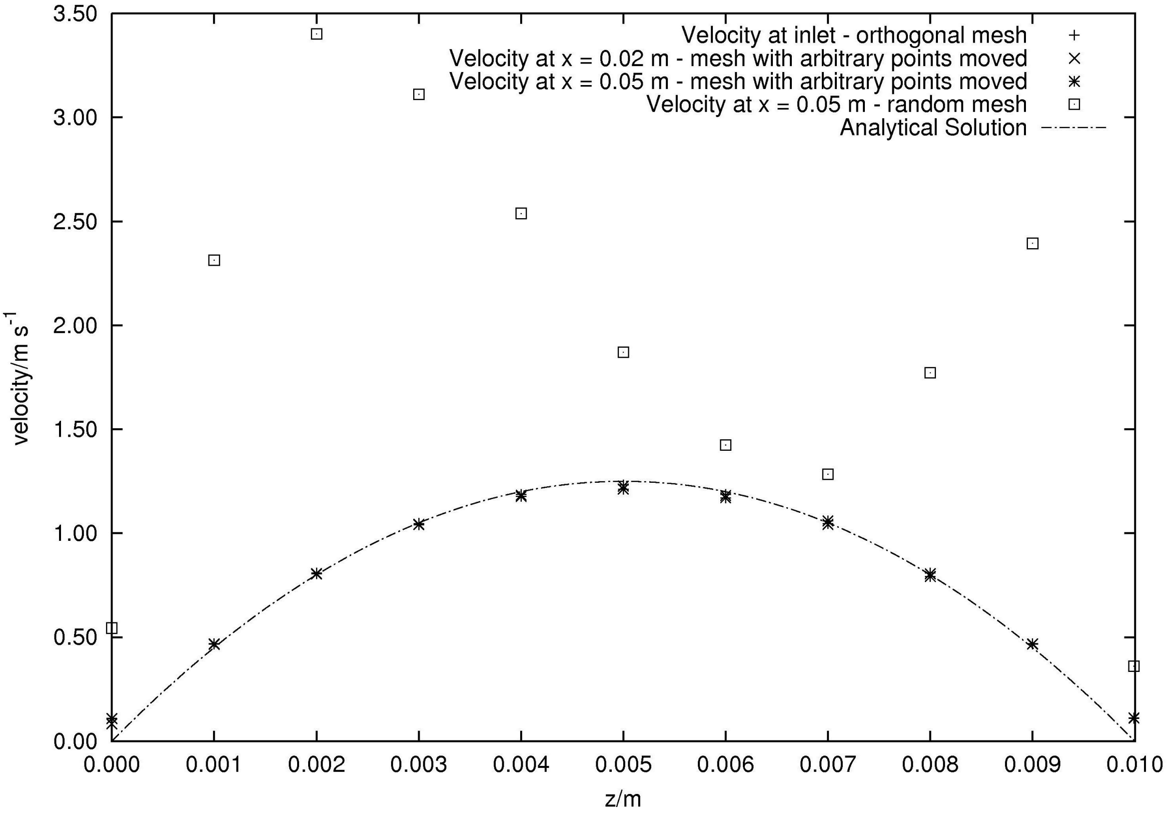

Figure 2.53: Velocity comparison for different meshes for the Re = 1 case.

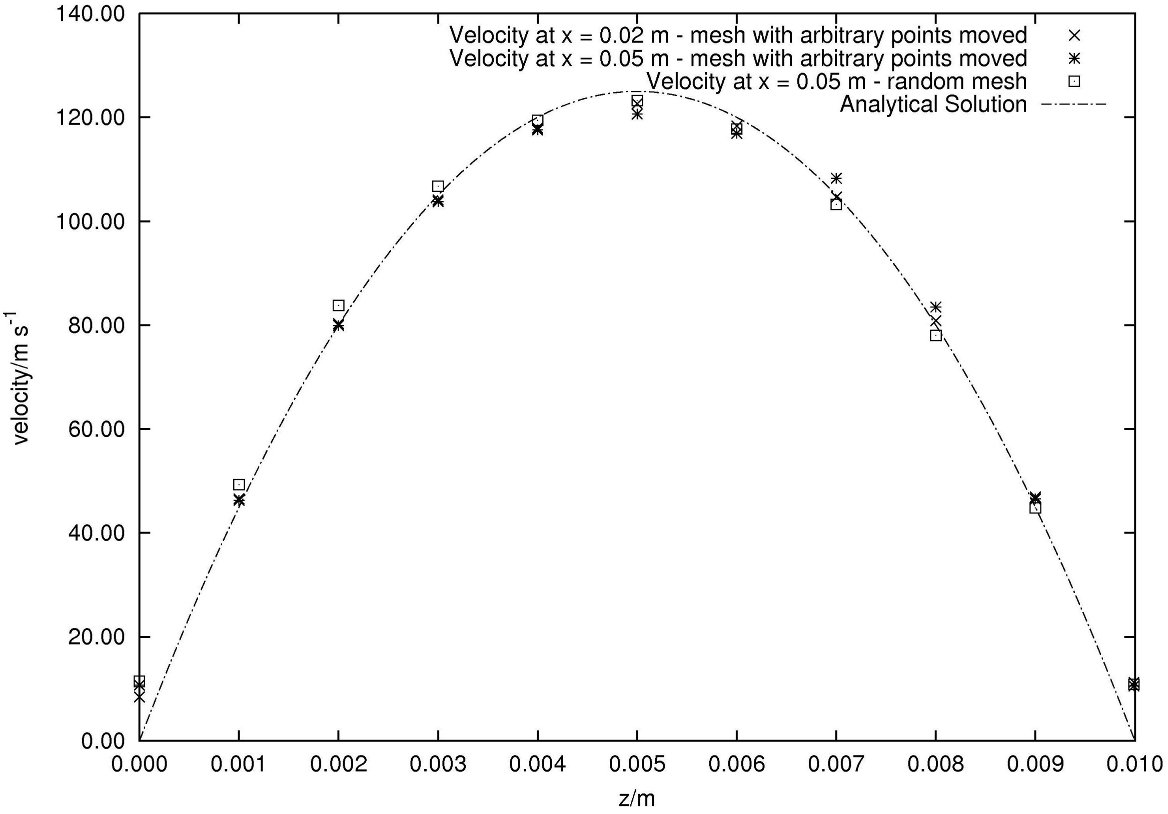

Figure 2.54: Velocity comparison for different meshes for the Re = 100 case.

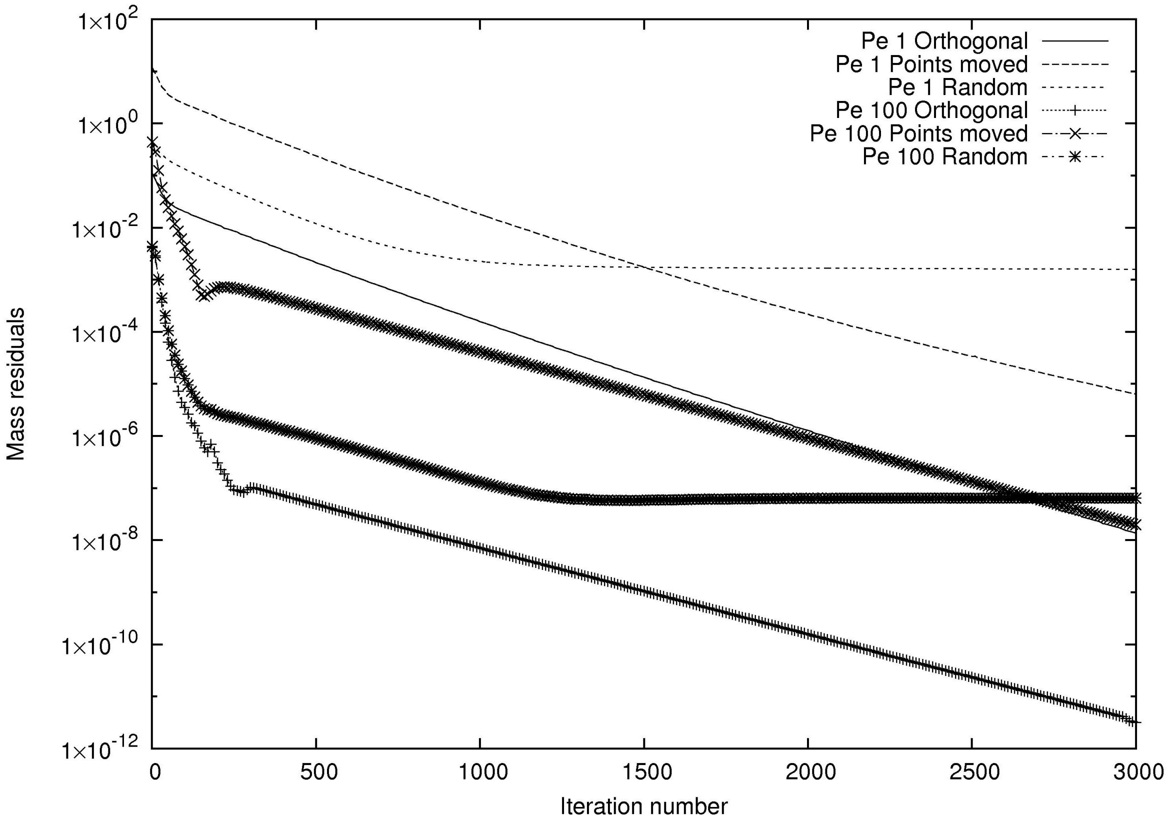

Figure 2.55: Mass residuals normalised with inlet mass entry for Poisseuille flow case. Residual calculation as described in Section 1.2.9. Resulting residual divided by mass entering domain at inlet.

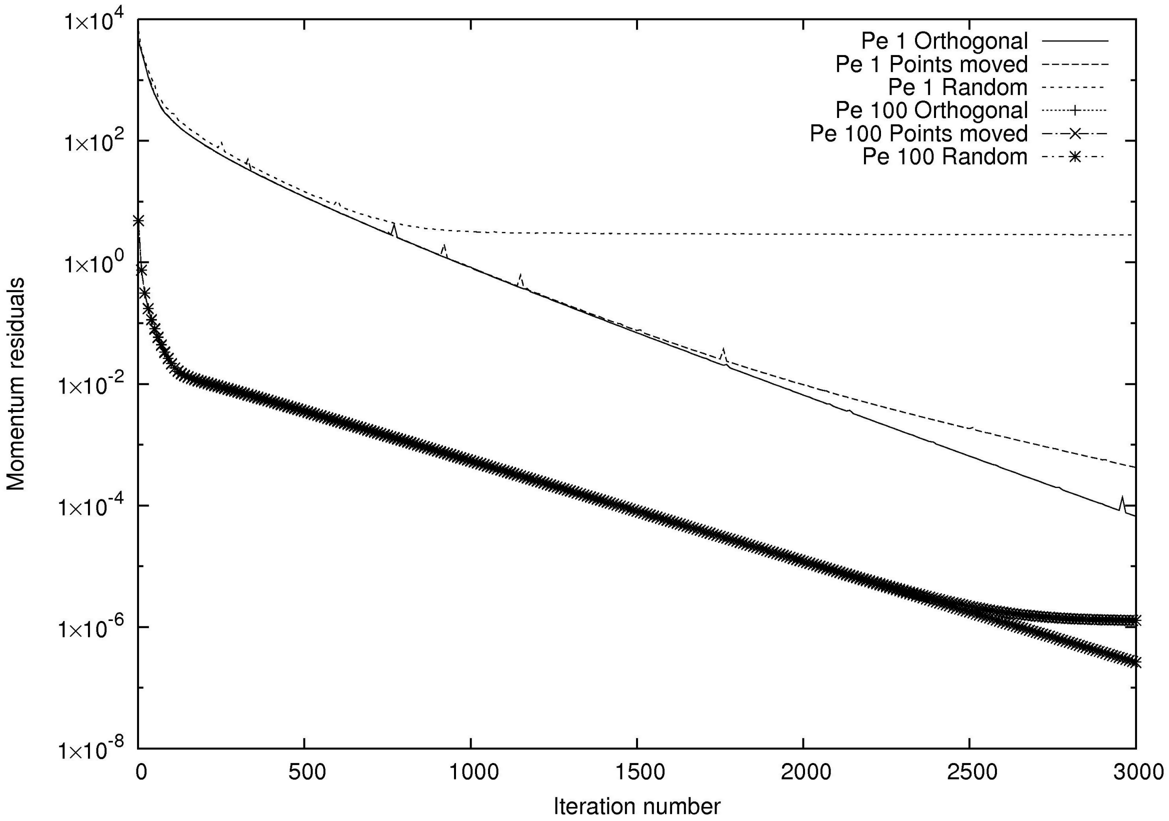

Figure 2.56: Momentum residuals normalised with inlet momentum for Poisseuille flow case. Residual calculation as described in Section 1.2.9. Resulting residual divided by momentum at inlet.

2.5.2 Moving lid cavity flow

Moving lid cavity flow cases, with the geometry illustrated by Figure 2.67, are used to analyse convergence behaviour in the presence of non-orthogonality. In each case, the density $\rho = 1.0$ kg m^{-3} and the speed of the lid is set to 1 m s^{-1}. Each side of the cavity is equal to 1 m. The Reynolds number is varied by changing the dynamic viscosity μ. Except for the relaxation factor study, under-relaxation parameters of 0.5 and 0.8 are used to relax pressure and velocity respectively.

The benchmarks used to validate the numerical experiments presented in this report were taken from Ghia *et al.* [9] for the tilted orthogonal cavity and Demirdžić *et al.* [10] for skewed domains.

After testing the standard non-orthogonality procedure using diffusion corrections only, the effect of the adjustment variable ς on the speed of convergence is tested, to see if significant improvement in speed can be obtained with this parameter.

A similar geometry is then meshed using a commercial package [67] and the application of extreme measures described in Section 2.4 is studied in the case of large mass defects in some cells within the domain.

2.5.2.1 Orthogonal cavity flow

A Cartesian mesh of grid density 80×80 was generated. The schematic diagram for the case is shown in Figure 2.67, with $\theta = 90°$. Diffusion is significant for this case, since the Peclet numbers are of the order of 1.0 or less in the domain, as shown in Figure 2.57.

The expected recirculating flow is obtained as shown in Figures 2.58 and 2.59. The top wall represents the moving lid at 1 m s^{-1}. The fluid at the top is entrained by the lid and hit the right wall; with an enclosed cavity, the fluid can only move downwards upon impinging the right surface and a circulating region is created. The region of highest pressure is therefore at the top right corner of the domain, as shown in Figure

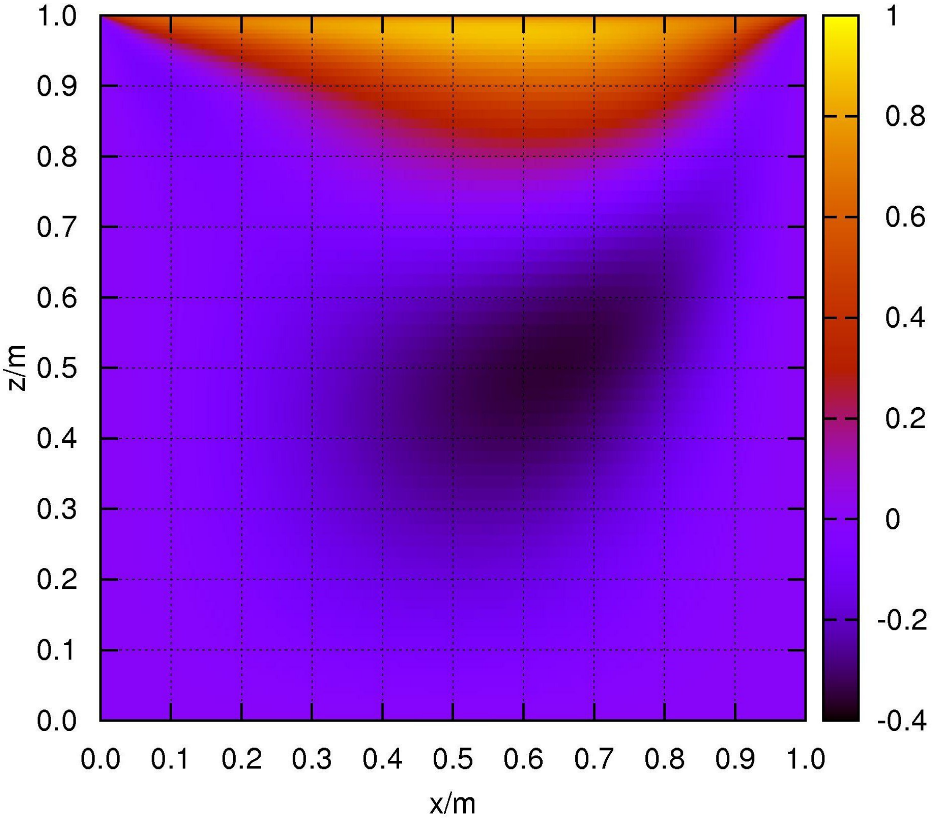

Figure 2.57: Peclet number contour for orthogonal moving lid case.

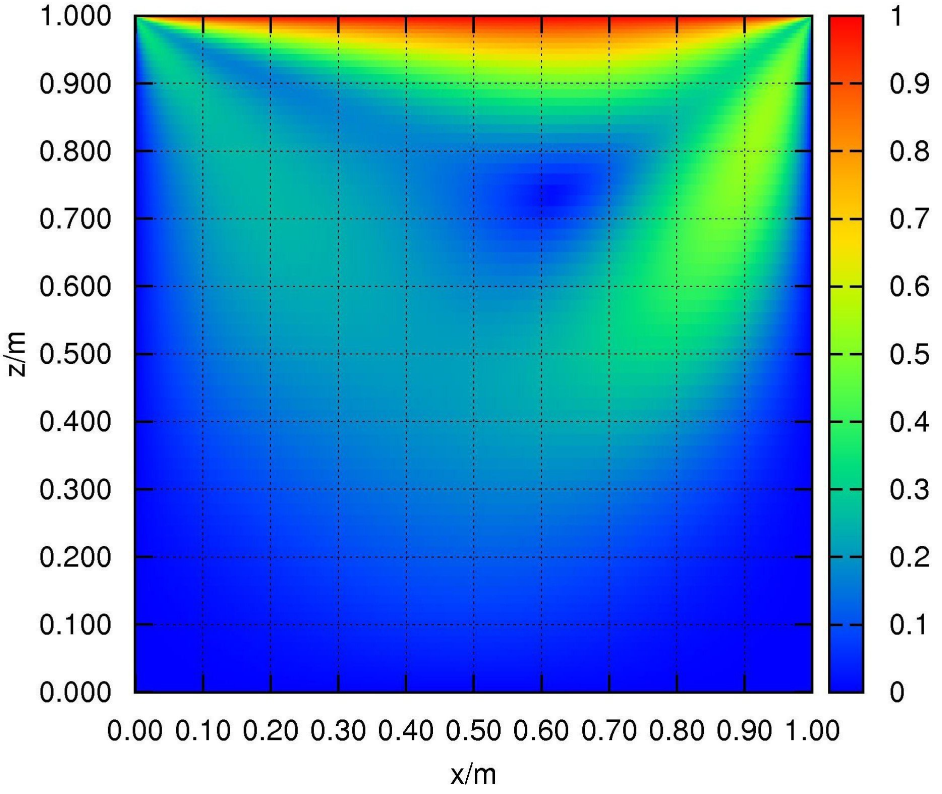

Figure 2.58: Velocity contour for orthogonal moving lid case.

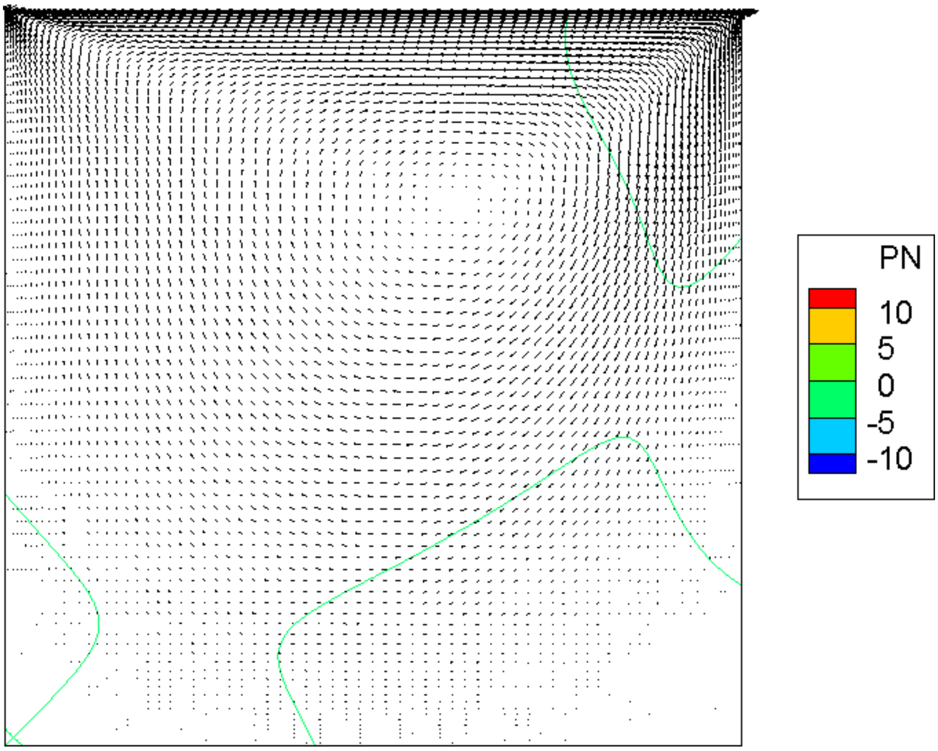

Figure 2.59: Pressure contour lines and velocity vectors for orthogonal moving lid case.

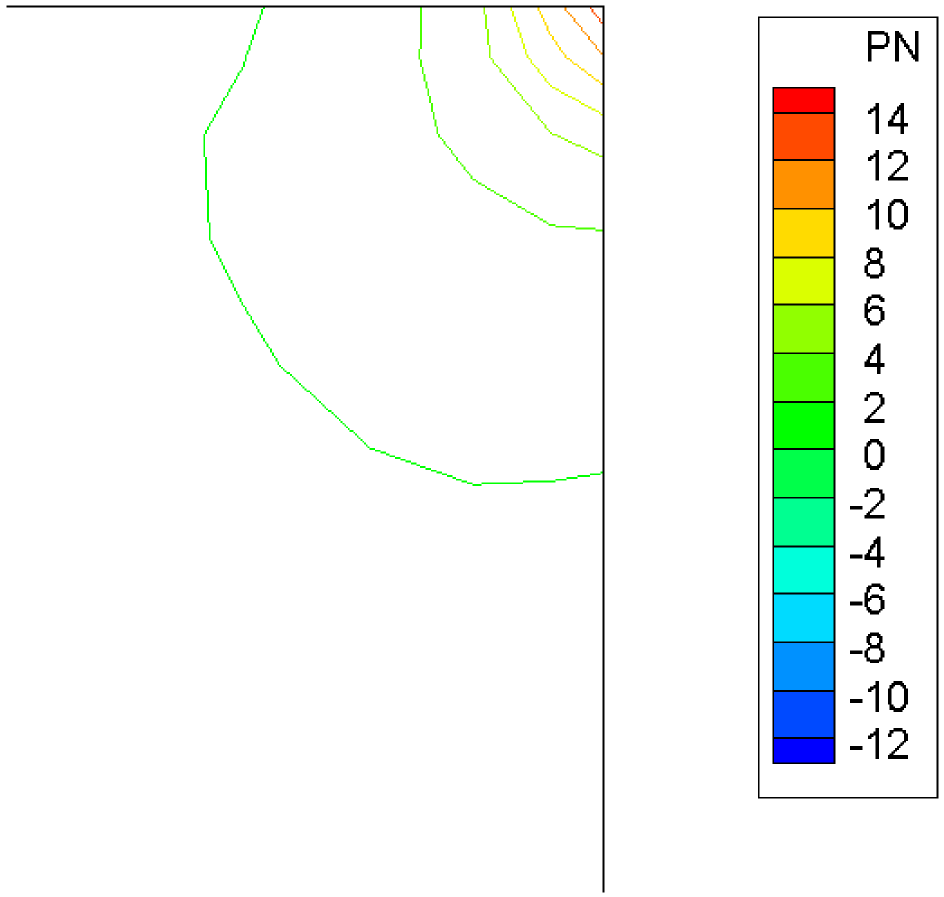

Figure 2.60: Pressure contour lines for top right edge of orthogonal moving lid case.

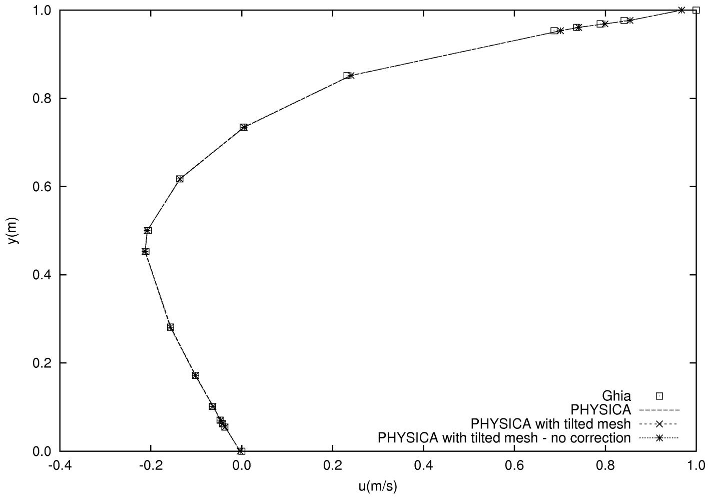

Figure 2.61: Comparison of velocity u along line $x = 0.5$ m for Re = 100 with benchmark from Ghia *et al.* [9].

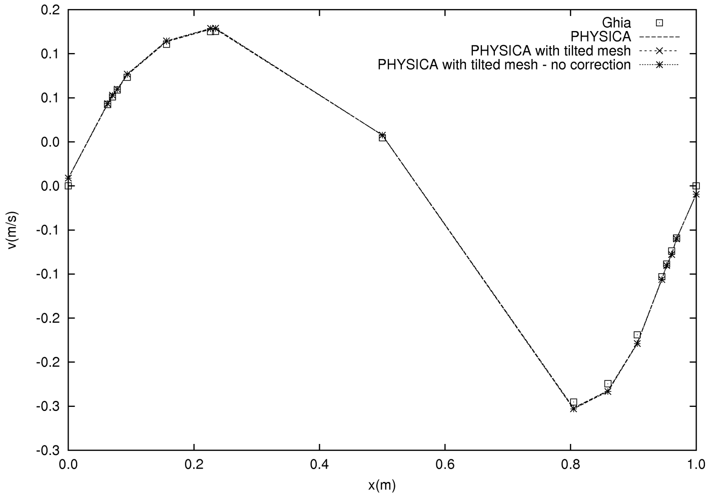

Figure 2.62: Comparison of velocity v along line $y = 0.5$ m for Re = 100 with benchmark from Ghia *et al.* [9].

2.60.

The calculated u velocity along the vertical line $x = 0.5$ m and the v velocity along the horizontal line $y = 0.5$ m are shown in Figure 2.61 and Figure 2.62. The results obtained with the above method are in perfect agreement with the benchmark from Ghia *et al.* [9]. The convergence behaviour in the orthogonal case is shown in Figure 2.63.

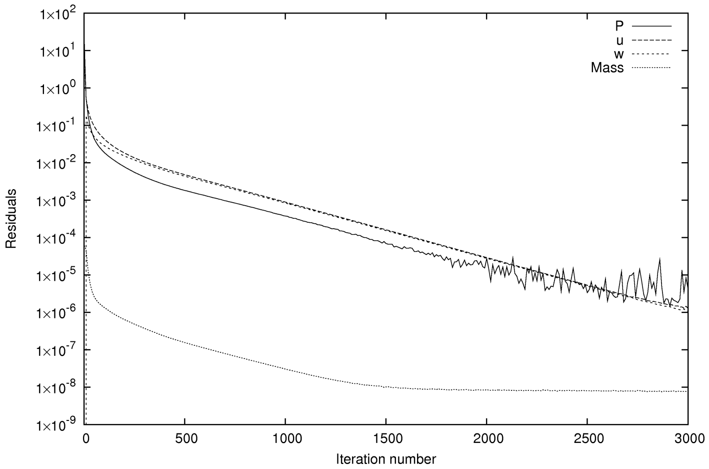

Figure 2.63: Residuals for orthogonal moving lid cavity. Residual calculation as described in Section 1.2.9.

The code solves for Cartesian velocity components, independently of mesh orientation. To eliminate component resolution as a cause for instability and check the code for consistency, the mesh was then rotated anticlockwise by 45°. The case was solved, invoking the diffusion correction for non-orthogonality, described in Section 2.3.2.3. There are no visible differences between results obtained from the original mesh and this new rotated case. However, the residuals for pressure and momentum in the rotated mesh case - shown in Figure 2.64 - remain high, even though both meshes yield accurate solutions at the symmetry lines.

The difference in mass residuals is due to the norm used to calculate residuals - which uses the absolute values of the difference between the previous and the current

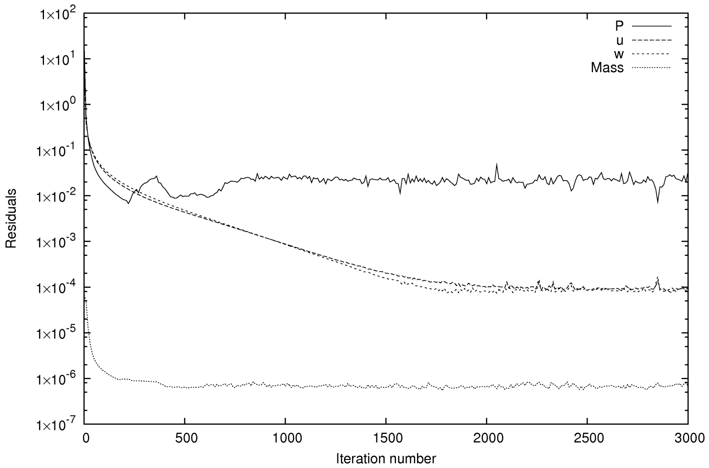

Figure 2.64: Residuals for orthogonal moving lid cavity tilted by 45°.

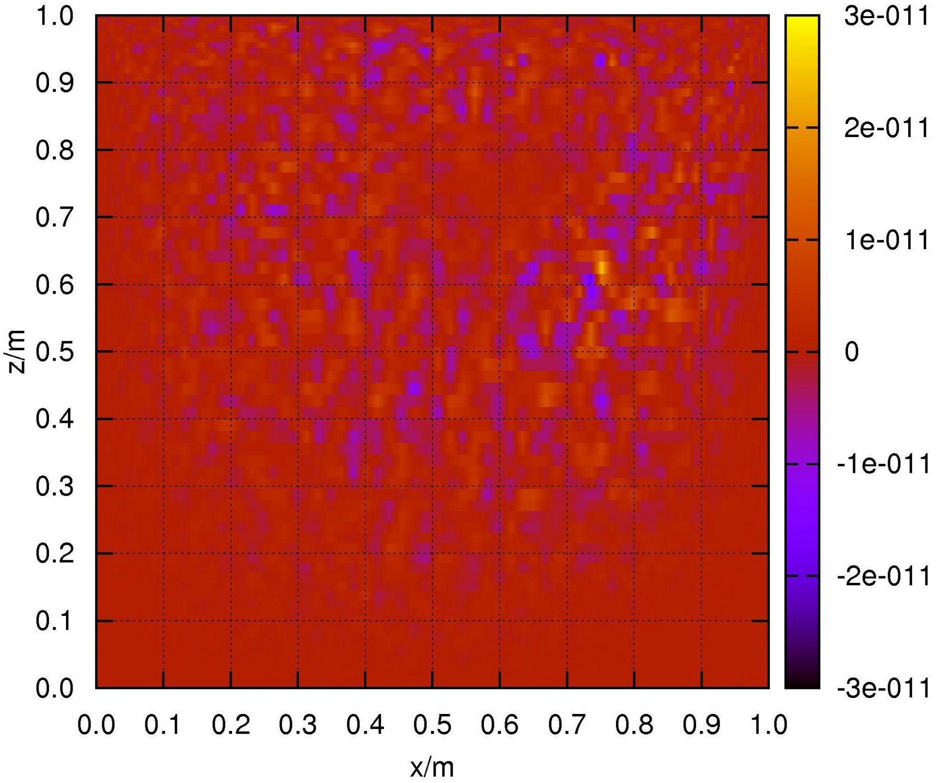

Figure 2.65: Mass imbalance contour for orthogonal moving lid case.

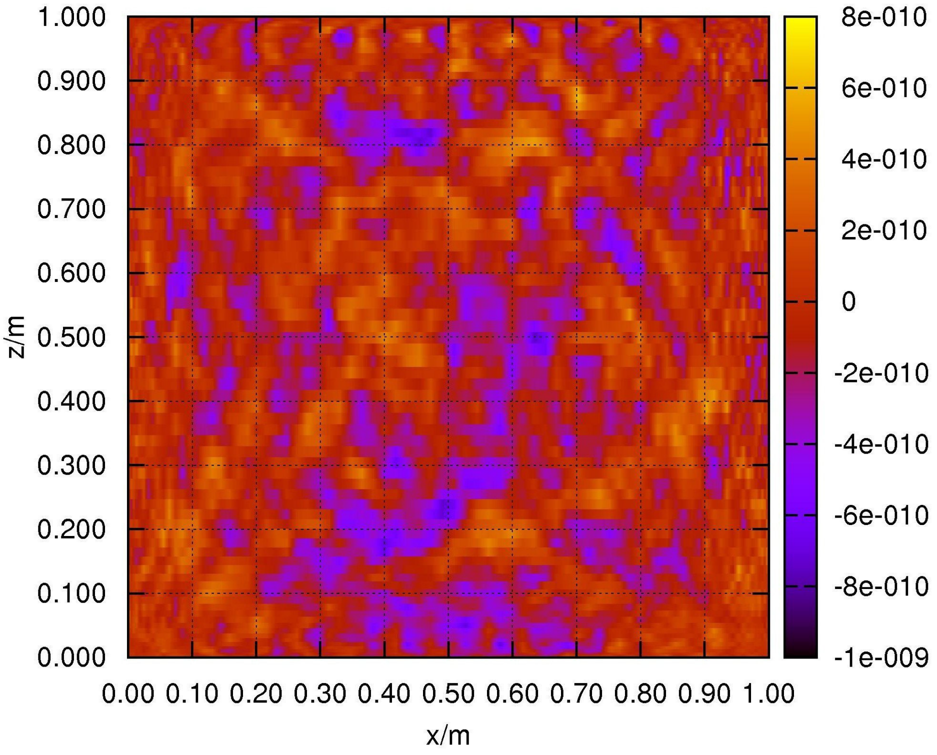

Figure 2.66: Mass imbalance contour for orthogonal moving lid case tilted by 45°.

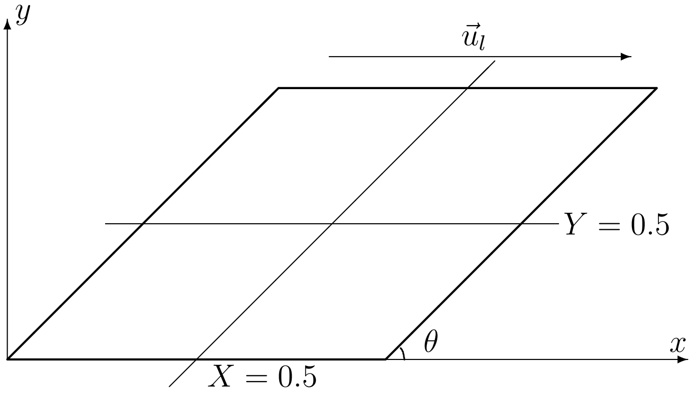

Figure 2.67: Geometry of skewed moving lid cavity.

iteration value. The net mass error in the domain in both cases is zero: a mass contour plot for each case - Figures 2.65 and 2.66 - show that the mass errors in the domain are both positive and negative. A summation of all errors in the domain yield a small mass imbalance in both cases.

Having established that accurate results can be obtained in a perfectly orthogonal moving lid case, the code is now tested with skewed geometries used as standard test case in the literature.

2.5.2.2 Skewed cavity flow

The skewed cavity depicted in Figure 2.67 was constructed by simple deformation of the orthogonal Cartesian mesh used in Section 2.5.2.1 by multiplying the coordinate of each grid point with the matrix $\begin{pmatrix} 1 & \cos\theta \\ 0 & \sin\theta \end{pmatrix}$. The mesh density used is depicted in Figure 2.68 - the mesh is finer at the wall where larger velocity gradients are expected. Two test angles, 30° and 45°, were used in this section. This standard test case has also been used by various authors in the literature to test their non-orthogonality handling method [70; 71; 72; 73].

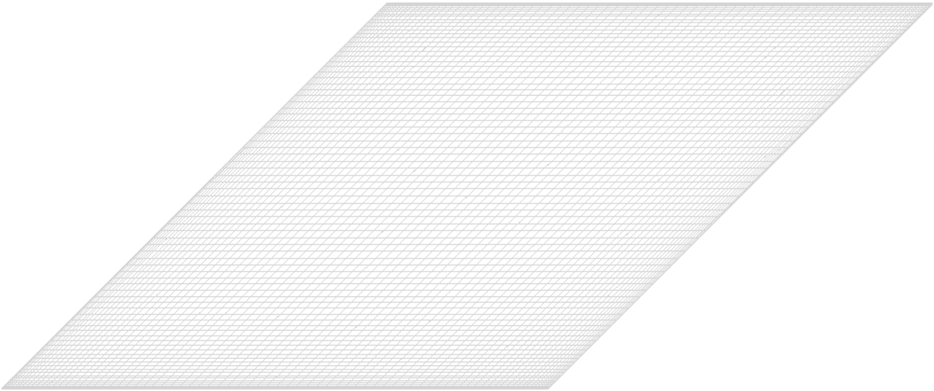

Figure 2.68: Mesh used skewed moving lid cavity. 80×80 grid

Four test cases were run by permuting the possible combinations of using cavities deformed by 30° and 45°, with two Reynold numbers of 100 and 1000. The Reynold number was varied by changing the kinematic viscosity of the fluid in the cavity. For cases with Re = 100, density $\rho = 1.0$ kg m^{-3} and kinematic viscosity $\nu = 0.01$ m^2 s^{-1}. For Re = 1000, the density is still 1.0 kg m^{-3} while $\nu = 0.001$ m^2 s^{-1}. The HYBRID differencing scheme is used to discretise the continuity equations in the results presented in this section. The diffusion non-orthogonality methods described in Sections 2.3.2.3 and 2.3.2.4 are tested in the following subsections.

Mesh independence analysis using the diffusion correction method of Section 2.3.2.3 The results for $\theta = 45°$ are shown in Figure 2.72 and Figure 2.73. The convergence criterion for these cases is that the total residuals for each solved parameter P, u or v - as calculated using equation (1.37) - must fall below 10^{-5}. The results obtained with a finite volume, collocated, pressure-correction based approach, coupled with the diffusion correction method of Section 2.3.2.3, with $\lambda = \frac{1}{\boldsymbol{\xi} \cdot \boldsymbol{n}}$, agree perfectly with the benchmark of Demirdžić *et al.* [10].

The velocity profiles in both cases are consistent with a recirculation created due to the moving lid as shown in Figure 2.69. The velocity is 1 m s^{-1} beneath the lid and the flow circulates clockwise around the domain. Results for an 80×80 mesh are already grid independent. The mass imbalance in the domain is negligible, as shown in Figure 2.70. The Peclet number is small in this case, as shown in Figure 2.71.

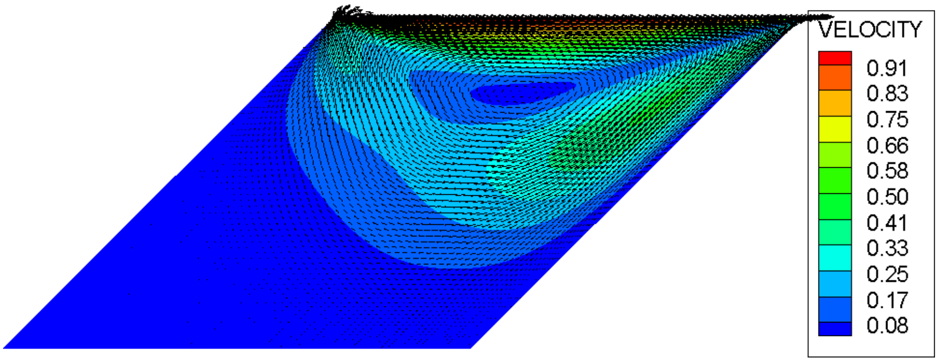

Figure 2.69: Velocity contour and vectors for skewed cavity with $\theta = 45°$ and Re = 100. 80×80 grid

Other cases with $\theta = 45°$ and Re = 1000, $\theta = 30°$ and Re = 100 and $\theta = 30°$ and Re = 1000 has been run and compared with the benchmarks from Demirdžić *et al.* [10]. The results are shown in Figures 2.86, 2.87, 2.88, 2.89, 2.90 and 2.91. The difference in the flow predictions is pronounced when the turbulence level and mesh skewness is altered. As expected, mesh independence is achieved at higher mesh densities for the skewer and more turbulent cases. For the profile of v along the central horizontal line, the results are completely wrong for mesh densities lower than 160x160 in the

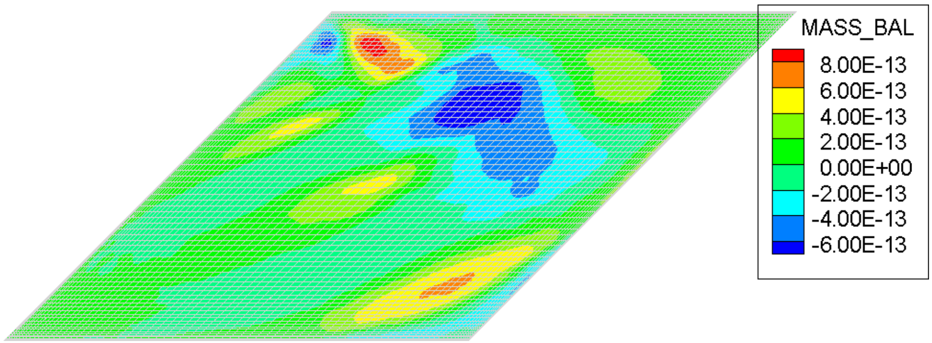

Figure 2.70: Mass contour and mesh for skewed cavity with $\theta = 45°$ and Re = 100. 80×80 grid

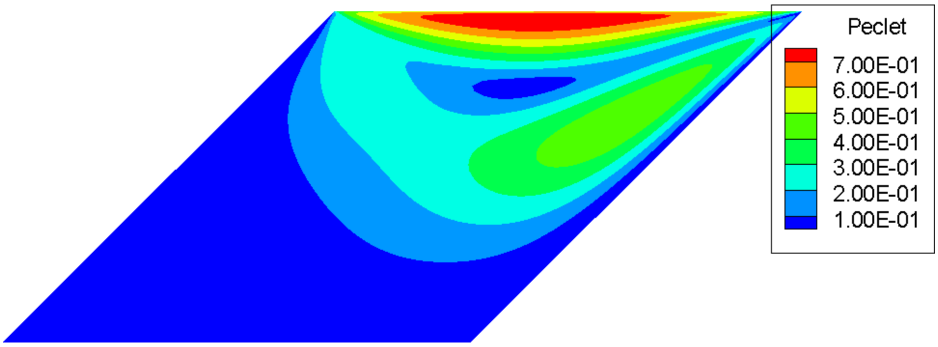

Figure 2.71: Peclet number contour for skewed cavity with $\theta = 45°$ and Re = 100. 80×80 grid

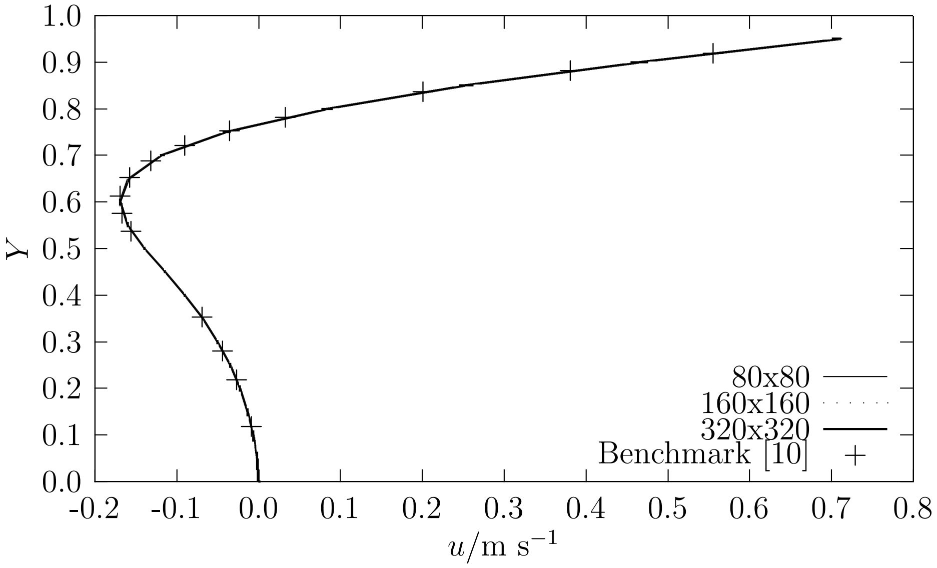

Figure 2.72: Comparison of velocity u along $Y = 0.5$ for $\theta = 45°$ and Re $= 100$ with benchmark from Demirdžić *et al.* [10]

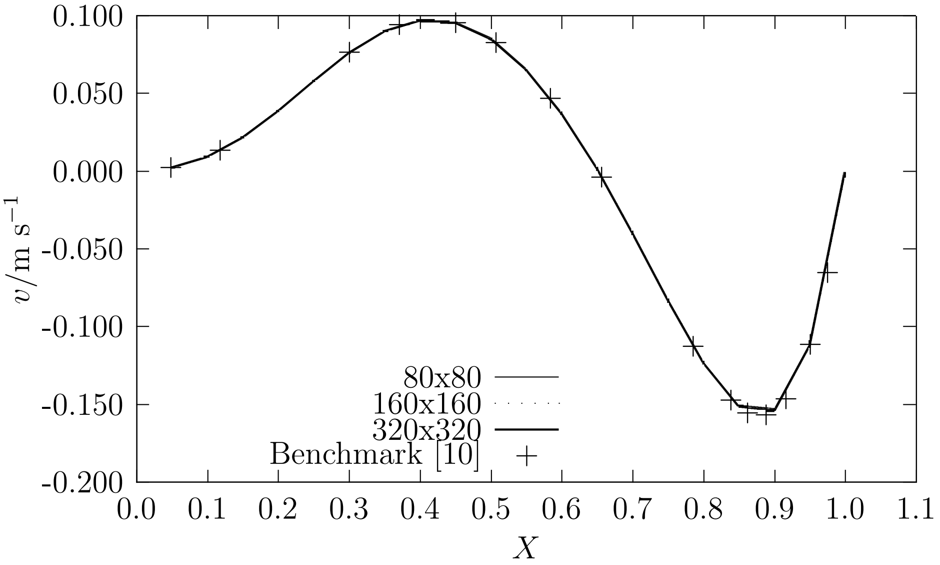

Figure 2.73: Comparison of velocity v along $X = 0.5$ for $\theta = 45°$ and Re $= 100$ with benchmark from Demirdžić *et al.* [10]

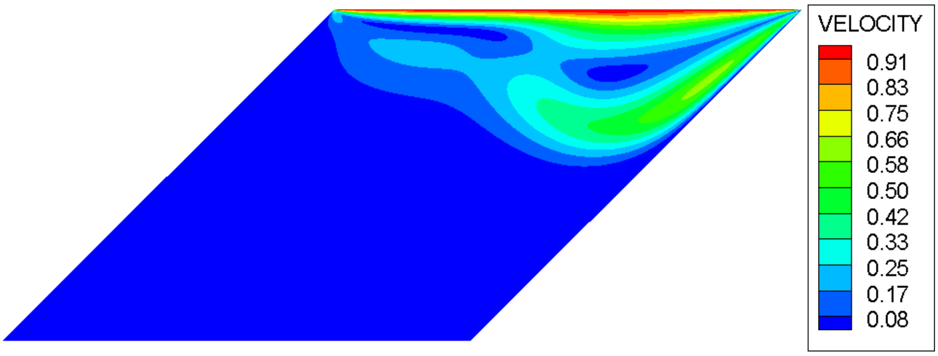

Figure 2.74: Velocity contour and vectors for skewed cavity with $\theta = 45°$ and Re = 1000. 160×160 grid

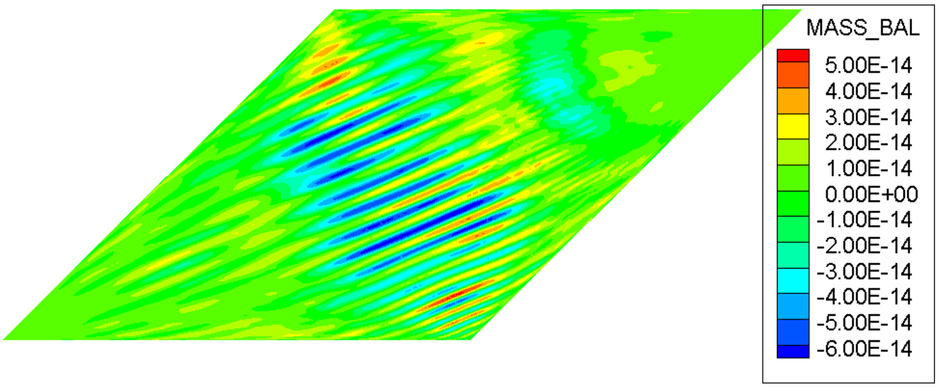

Figure 2.75: Mass contour and mesh for skewed cavity with $\theta = 45°$ and Re = 1000. 160×160 grid

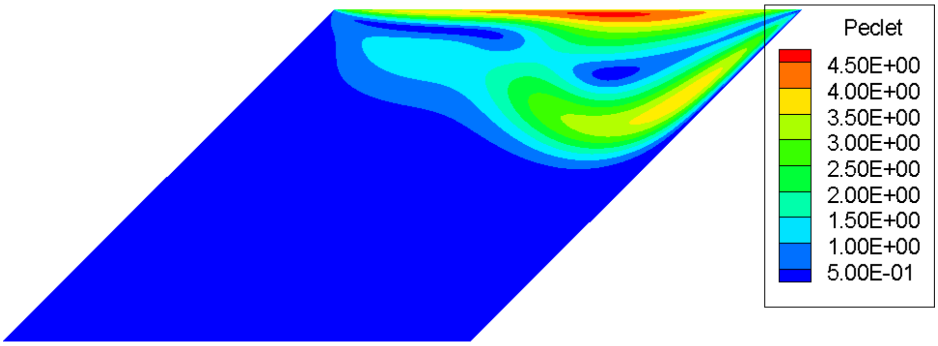

Figure 2.76: Peclet number contour for skewed cavity with $\theta = 45°$ and Re = 1000. 160×160 grid

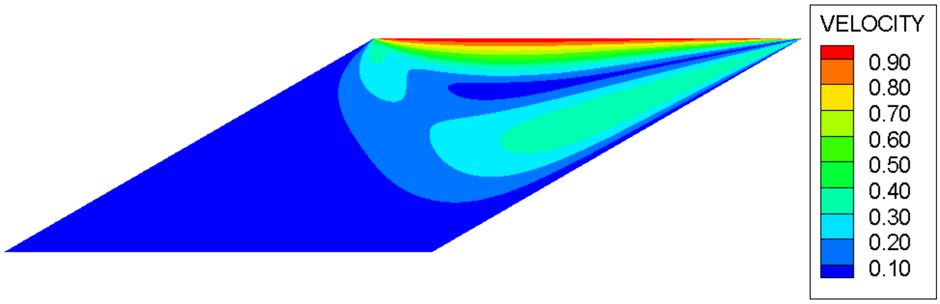

Figure 2.77: Velocity contour and vectors for skewed cavity with $\theta = 30°$ and Re = 100. 160×160 grid

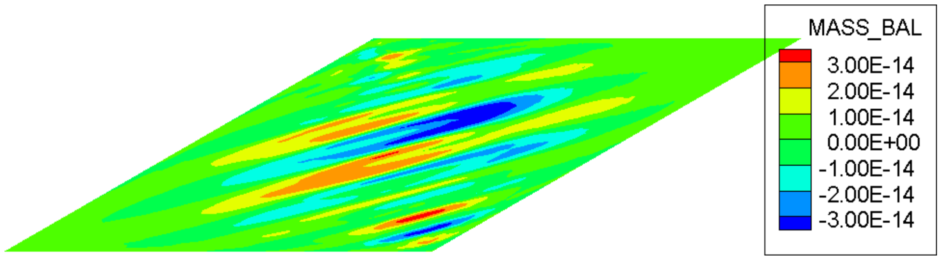

Figure 2.78: Mass contour and mesh for skewed cavity with $\theta = 30°$ and Re = 100. 160×160 grid

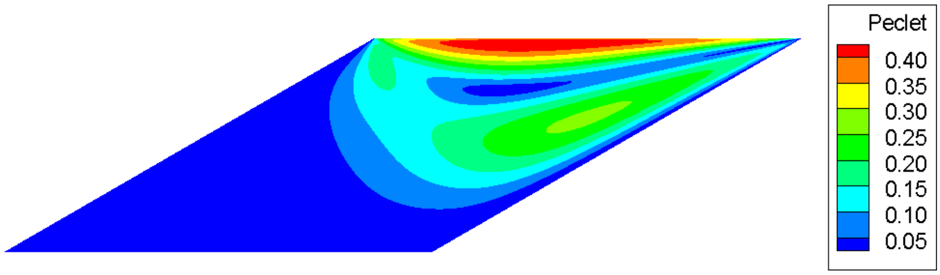

Figure 2.79: Peclet number contour for skewed cavity with $\theta = 30°$ and Re = 100. 160×160 grid

$\theta = 30°$ and Re $= 1000$ case. A high mesh density is therefore required for turbulent and highly non-orthogonal flows.

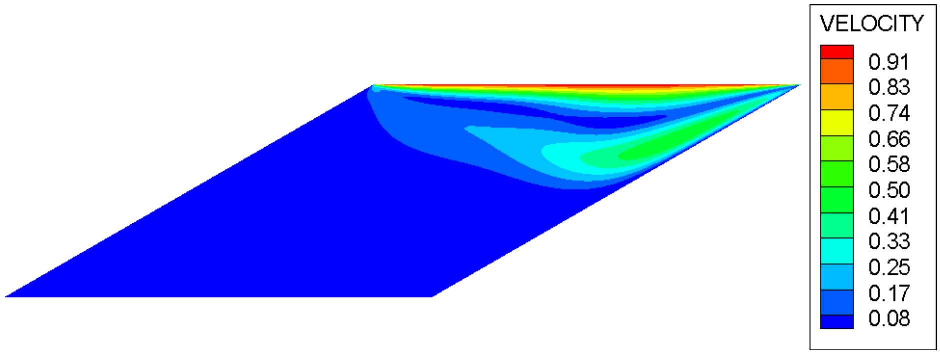

Figure 2.80: Velocity contour and vectors for skewed cavity with $\theta = 30°$ and Re = 1000. 80×80 grid

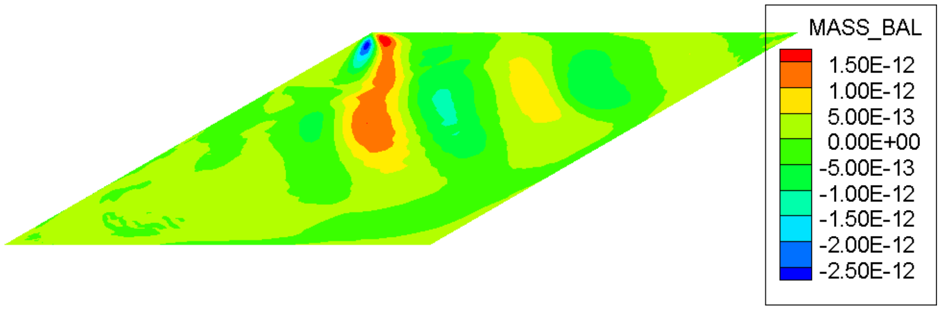

Figure 2.81: Mass contour and mesh for skewed cavity with $\theta = 30°$ and Re = 1000. 80×80 grid

Influence of linear relaxation parameters The impact of the choice of relaxation factors α_P and α_u on the convergence rate of the four cases each run on an 80×80 mesh are shown in Figures 2.92, 2.93, 2.94 and 2.95. In each plot, the ordinates represents the number of iterations required to achieve a converged solution for that particular case; the convergence criterion is selected as achieving at most a maximum residual value of 10^{-5} for all solved variables. The different values for the linear relaxation parameter for pressure correction α_P are represented by abscissae. Each line represents

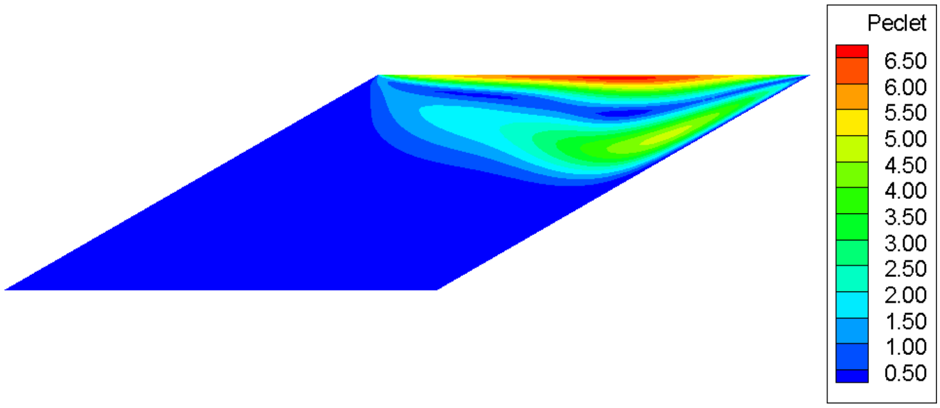

Figure 2.82: Peclet number contour for skewed cavity with $\theta = 30°$ and Re = 1000. 80×80 grid

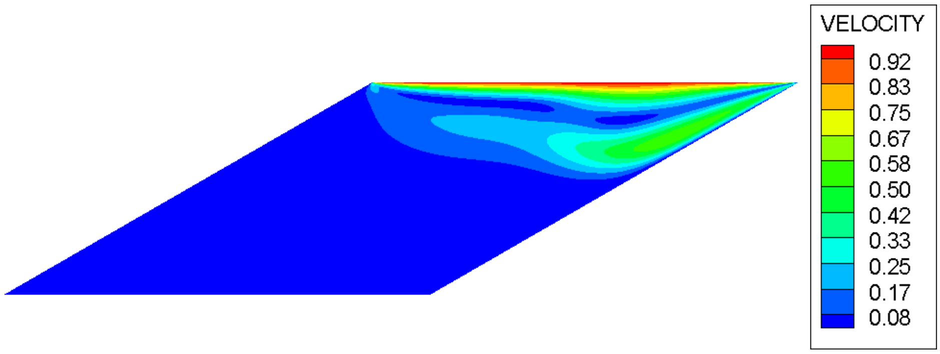

Figure 2.83: Velocity contour and vectors for skewed cavity with $\theta = 30°$ and Re = 1000. 320×320 grid

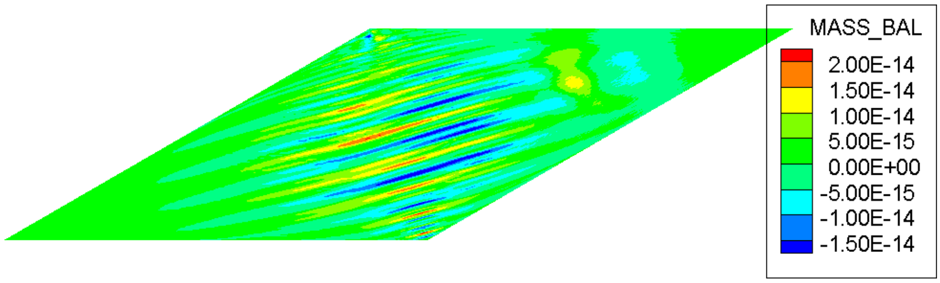

Figure 2.84: Mass contour and mesh for skewed cavity with $\theta = 30°$ and Re = 1000. 320×320 grid

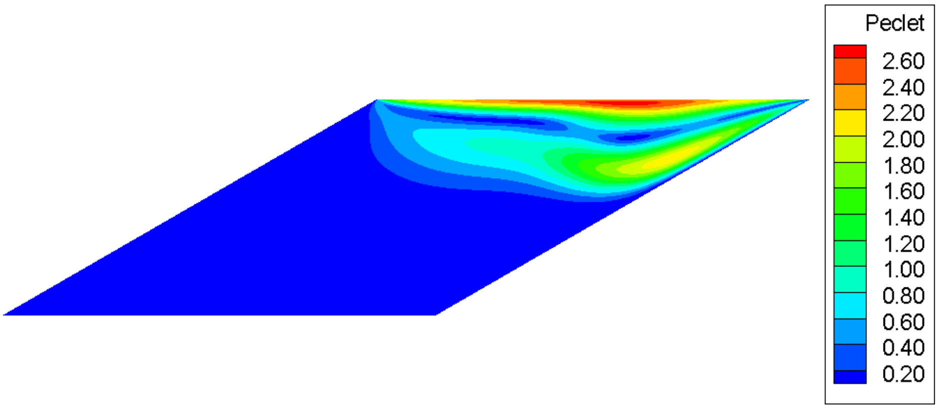

Figure 2.85: Peclet number contour for skewed cavity with $\theta = 30°$ and Re = 1000. 320×320 grid

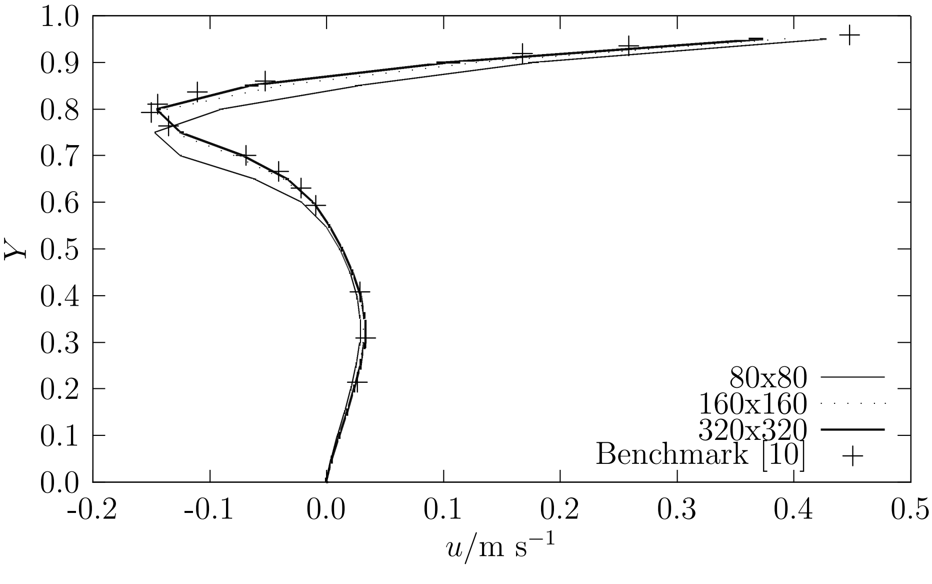

Figure 2.86: Comparison of velocity u along $Y = 0.5$ for $\theta = 45°$ and Re = 1000 with benchmark from Demirdžić *et al.* [10]

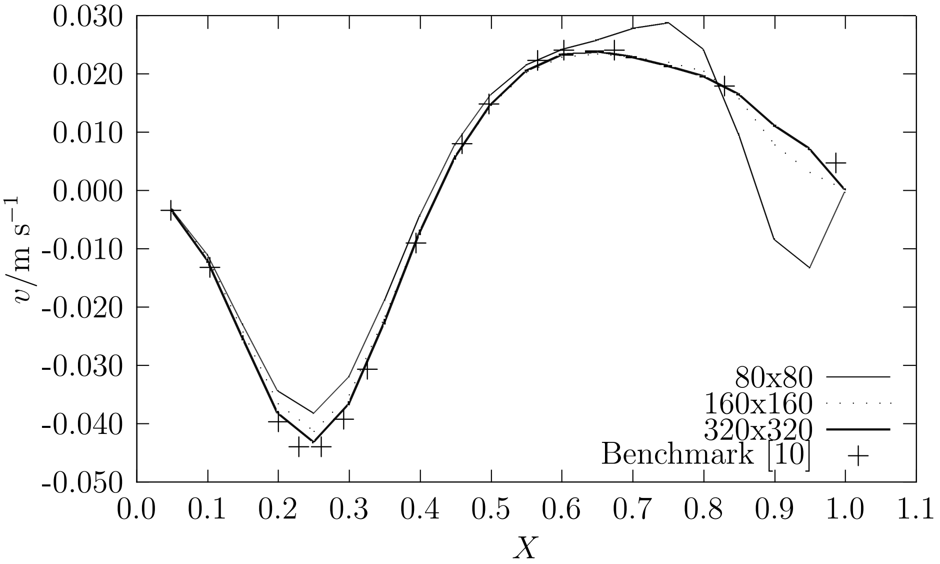

Figure 2.87: Comparison of velocity v along $X = 0.5$ for $\theta = 45°$ and Re $= 1000$ with benchmark from Demirdžić *et al.* [10]

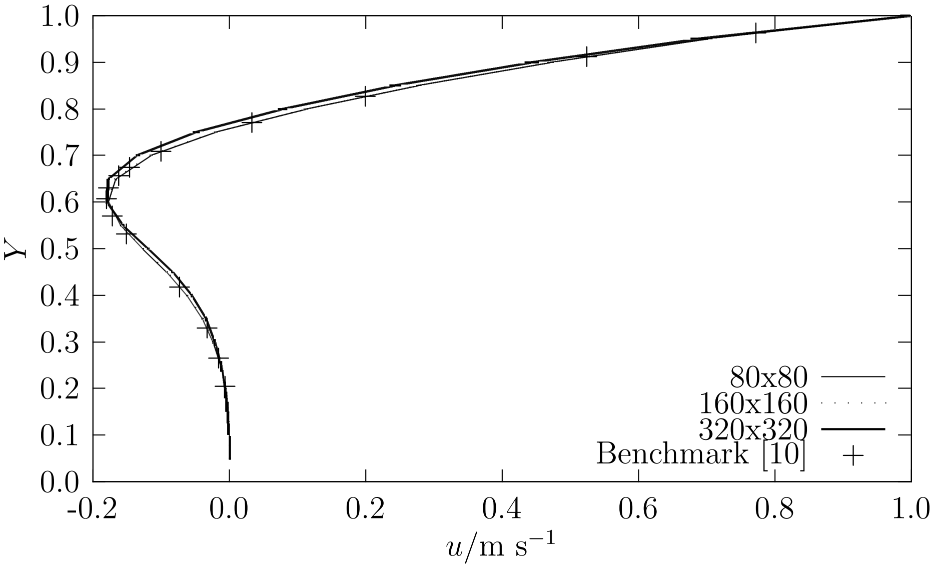

Figure 2.88: Comparison of velocity u along $Y = 0.5$ for $\theta = 30°$ and Re $= 100$ with benchmark from Demirdžić *et al.* [10]

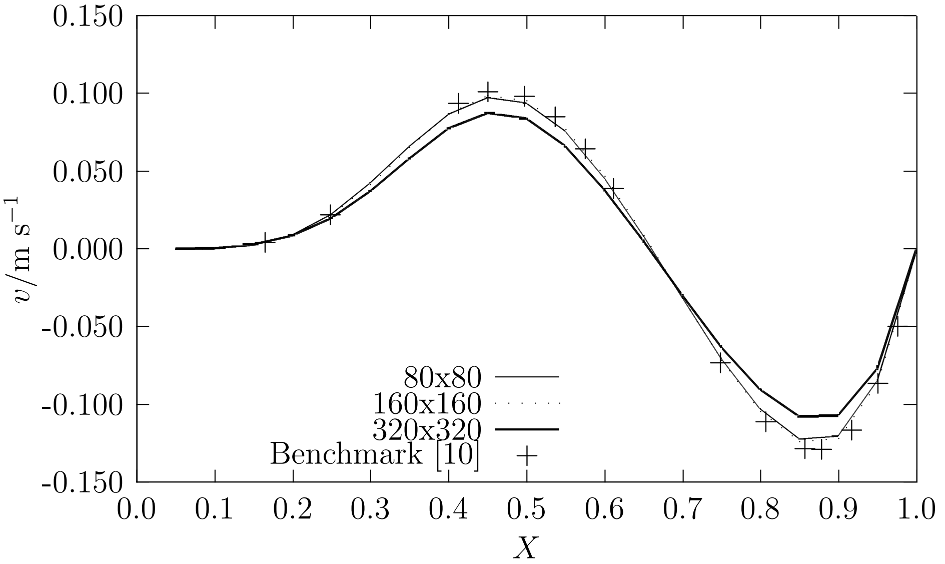

Figure 2.89: Comparison of velocity v along $X = 0.5$ for $\theta = 30°$ and Re $= 100$ with benchmark from Demirdžić *et al.* [10]

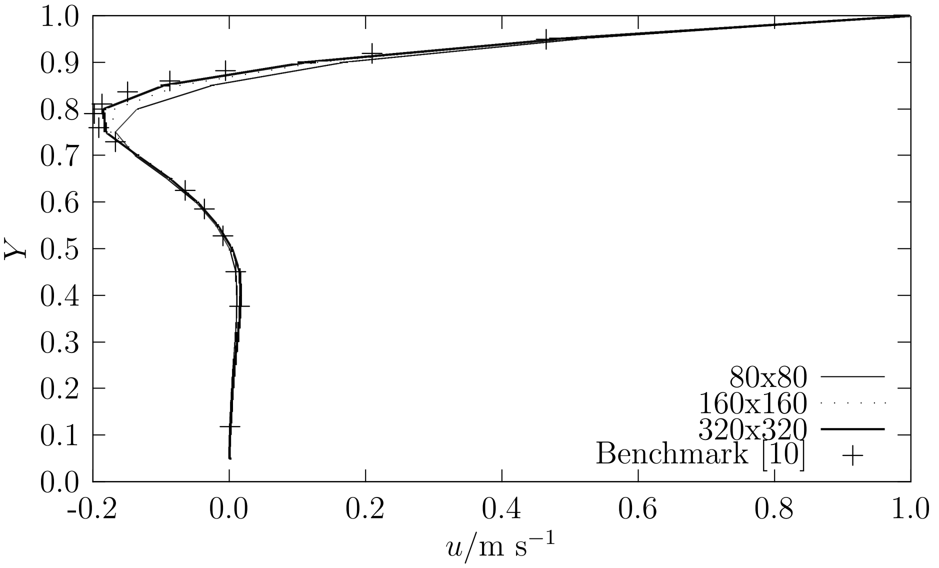

Figure 2.90: Comparison of velocity u along $Y = 0.5$ for $\theta = 30°$ and Re $= 1000$ with benchmark from Demirdžić *et al.* [10]

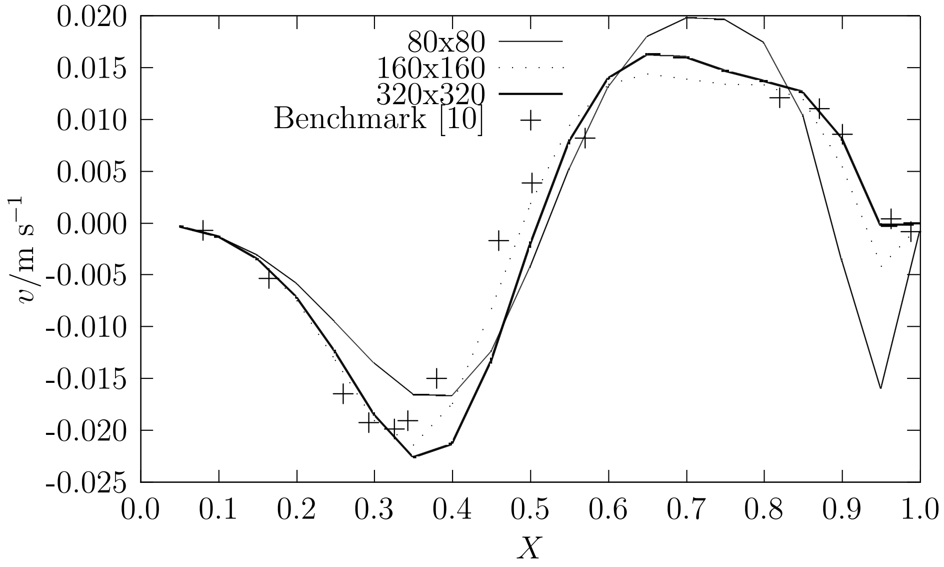

Figure 2.91: Comparison of velocity v along $X = 0.5$ for $\theta = 30°$ and Re = 1000 with benchmark from Demirdžić *et al.* [10]

Table 2.2: Linear relaxation parameters resulting in fastest convergence

Case	α_u	α_P
$\theta = 45°$ and Re = 100	0.9	0.4
$\theta = 45°$ and Re = 1000	0.9	0.1
$\theta = 30°$ and Re = 100	0.9	0.2
$\theta = 30°$ and Re = 1000	0.9	0.3

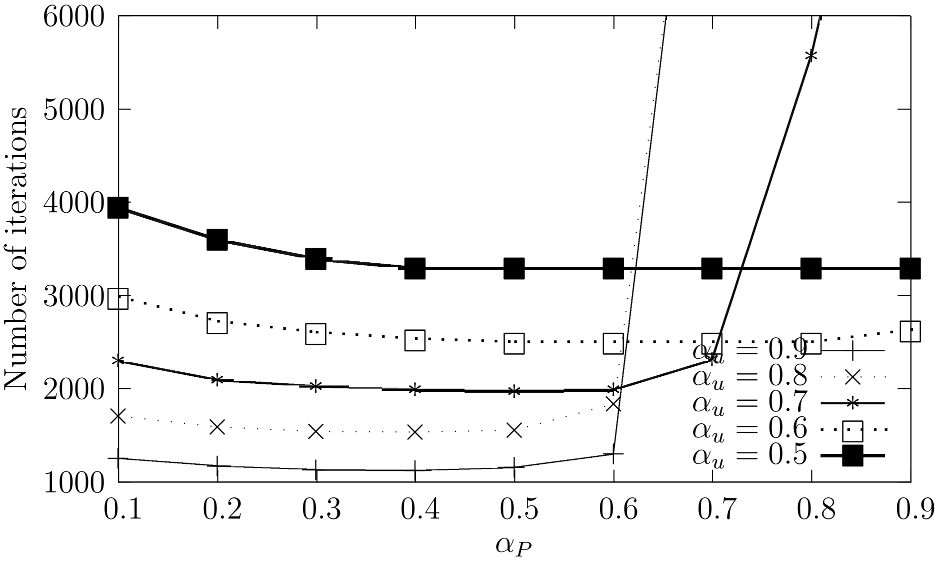

Figure 2.92: Influence of relaxation parameters on the convergence rate of the case for $\theta = 45°$ and Re $= 100$

a different value of α_u, the momentum linear relaxation parameter, as described by the key. Points out of the plotting scale represent divergence.

The choice of under-relaxation parameter for the pressure equation, α_P, is immaterial for low values momentum under-relaxation α_u. These require a large number of iterations to achieve convergence. Less iterations are required for high values of α_u; however, the case fails to converge if the pressure correction equation is not severely relaxed, e.g. with $\alpha_u = 0.8$, α_P must be lower than 0.5 for the case to converge. It is therefore good practice to keep a low value of the under-relaxation parameter for pressure.

Low values for the linear relaxation factor for pressure, α_P are required since the full pressure correction equation - including all cross derivatives required for non-orthogonal meshes - have not been included. Adding these terms would have resulted in a larger computation molecule, resulting in a more computationally intensive calculation: these cross derivatives are not essential since, in the semi-implicit approach

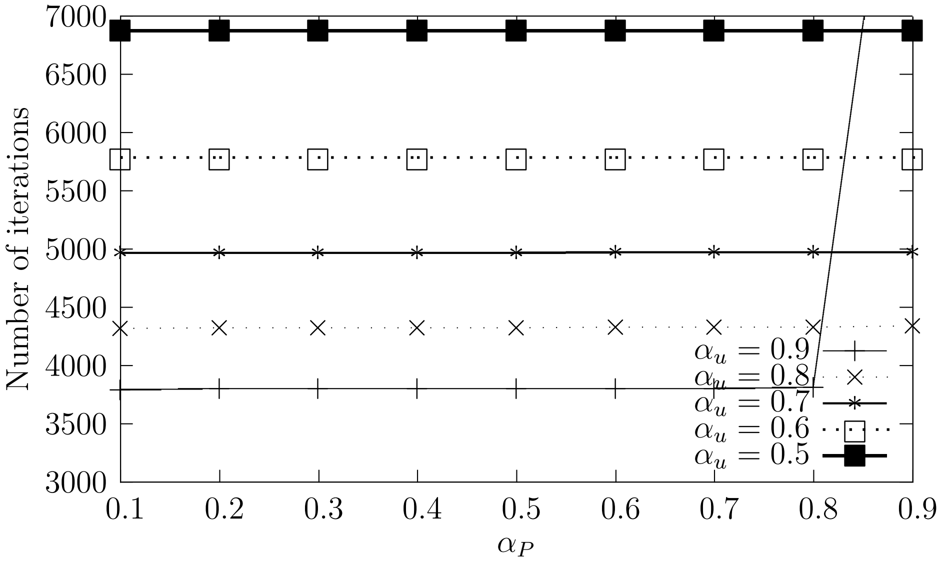

Figure 2.93: Influence of relaxation parameters on the convergence rate of the case for $\theta = 45°$ and Re $= 1000$

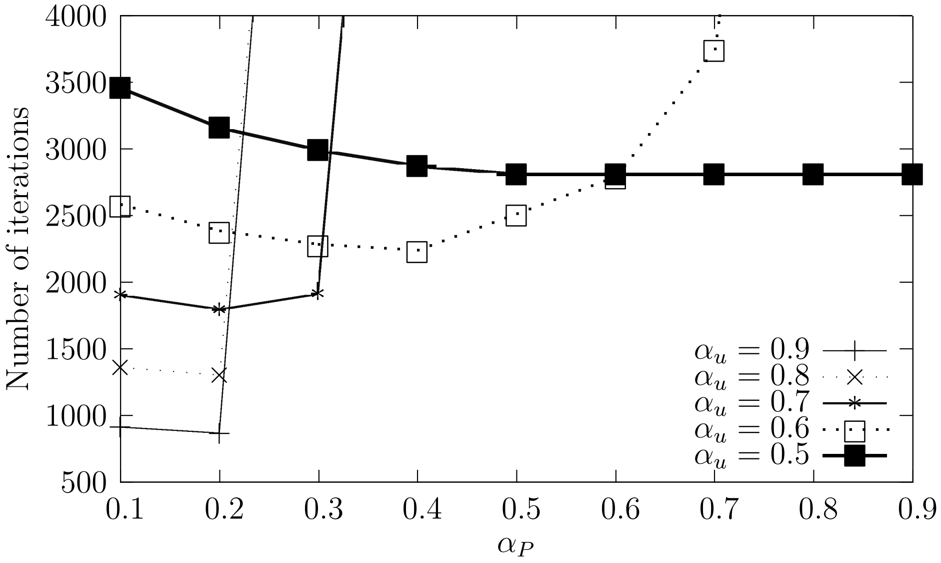

Figure 2.94: Influence of relaxation parameters on the convergence rate of the case for $\theta = 30°$ and Re $= 100$

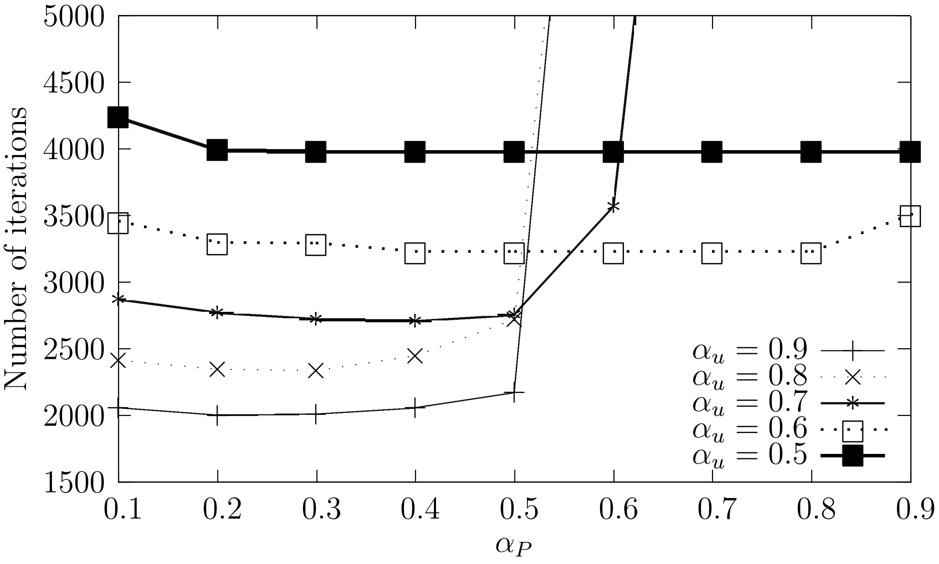

Figure 2.95: Influence of relaxation parameters on the convergence rate of the case for $\theta = 30°$ and Re = 1000

for pressure calculation, the pressure corrections becomes zero upon convergence. A low relaxation parameter will help stabilizing the pressure correction procedure in the absence of these non-orthogonal terms.

Influence of ς The influence of ς on convergence rate is shown in Figure 2.96. The ordinates represent the number of iterations to achieve convergence, i.e. when the maximum normalised residual for any solved variable is at most 10^{-5}. The abscissae represent the values of ς used in that particular simulation. Each line represents a case as described by the key. The linear relaxation parameters in each simulation were chosen as those which resulted in convergence the fastest as shown in Table 2.2.

Figure 2.96 shows the influence of ς on the convergence rate on each case. Increasing the value of ς corresponds to over-relaxation as expected. Decreasing the value of ς decreases the number of iterations required for convergence, but at the expense of stability. The speed up achieved by decreasing ς is disappointingly not significant.

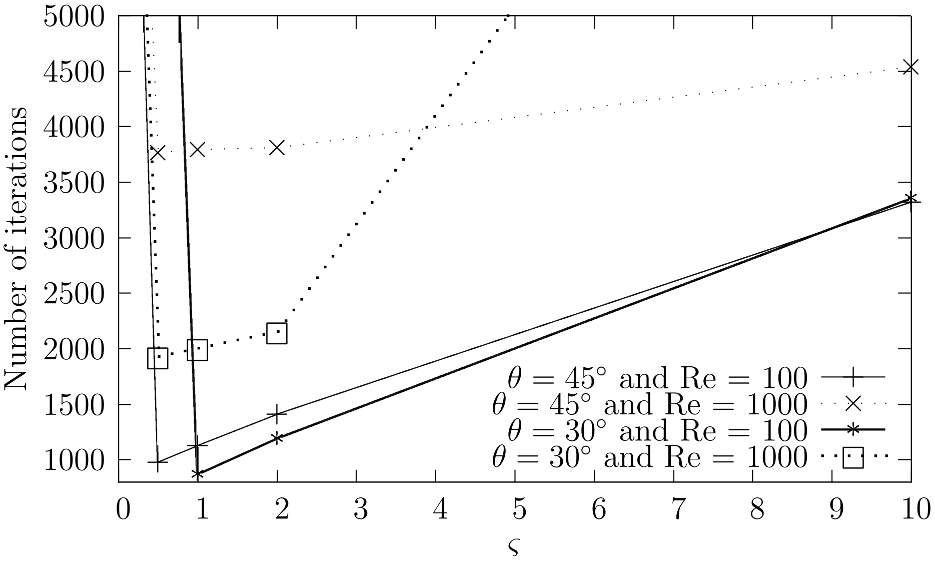

Figure 2.96: Influence of ς on the convergence rate of the skewed cavity case

Interestingly, a large over-relaxation factor, i.e. a large value of ς, cause divergence for the high turbulence and high skewness case - $\theta = 30°$ and Re = 1000. The new relaxation method described in Section 2.3.2.4 can be used in either highly skewed meshes without turbulence, or turbulent flows with mild skewness.

The non-orthogonality diffusion correction parameter of Section 2.3.2.3 is therefore sufficient to achieve convergence in consistently non-orthogonal geometries. The next section will investigate the convergence behaviour of meshes with varying levels of non-orthogonality, as expected from automatically generated meshes.

2.5.2.3 Meshes generated with a commercial package

Commercial packages such as Harpoon [67] are commonly used in industrial applications to generate meshes for complex geometries. It is essential for a good CFD code to work with these meshes generated from industry standard CAD/CAM and meshing software. The aim of this section is to test the code in the presence of cells generated with a commercial package, where sometimes convergence is not achieved due to mass imbalance in some cells only. With the moving lid cavity successfully modelled in the previous section, using a similar geometry for this study is appropriate, since convergence has already been established with the previous geometry. However, the meshes used are different in this section.

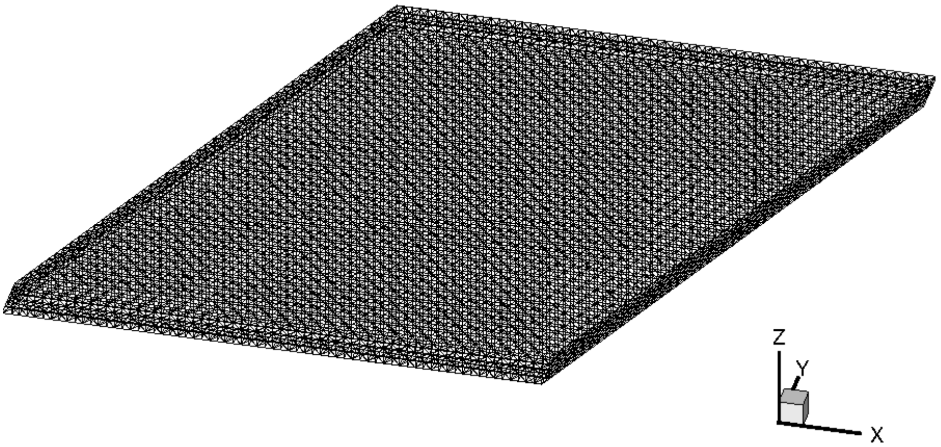

Figure 2.97: Tetrahedral mesh used for skewed moving lid cavity case.

The geometry investigated in this section is similar to the schematic depicted in Figure 2.67, with $\theta = 45°$ and Re =100. The same boundary conditions apply in this case. However, being three-dimensional, the cavity has a thickness of 0.1 m in the Y-direction. The velocities at the walls in the Y plane are fixed to zero. A 3D configuration is chosen to be able to test meshes made from tetraheda and hexahedra, as would be expected in industrial applications. Since no benchmark is available in this case, accuracy is assumed if the cases run with different meshes yield the same numerical results.

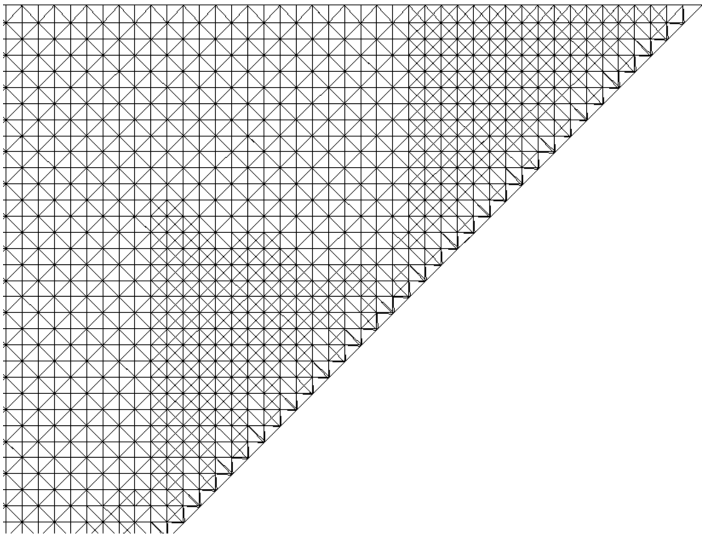

Figure 2.98: Corner where maximum velocity is expected in tetrahedral mesh used for skewed moving lid cavity case.

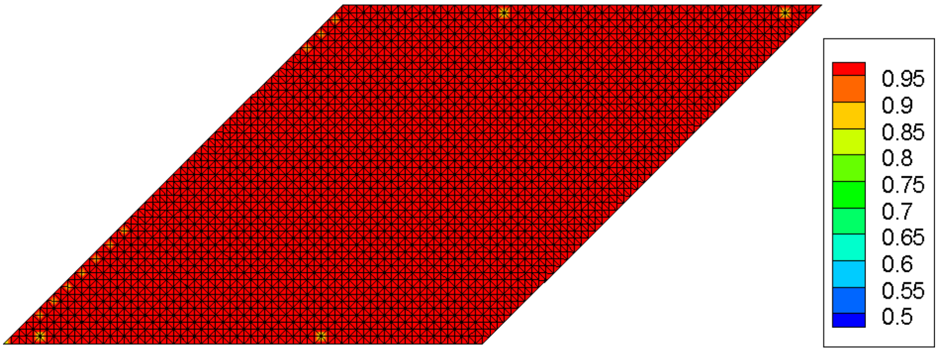

Figure 2.99: Mesh quality indicator for the tetrahedral mesh used for skewed moving lid cavity case. Quality indicator defined in Section 2.1.3.

The first mesh considered was generated by filling the domain with tetrahedra. The mesh is depicted in Figure 2.97. Of particular interest is the portion of the domain where maximum pressure is expected, as shown in Figure 2.98. This is the region where fluid will be forced downward upon impinging the wall after being entrained by the lid. The orthogonality indicator for this mesh is given in Figure 2.99.

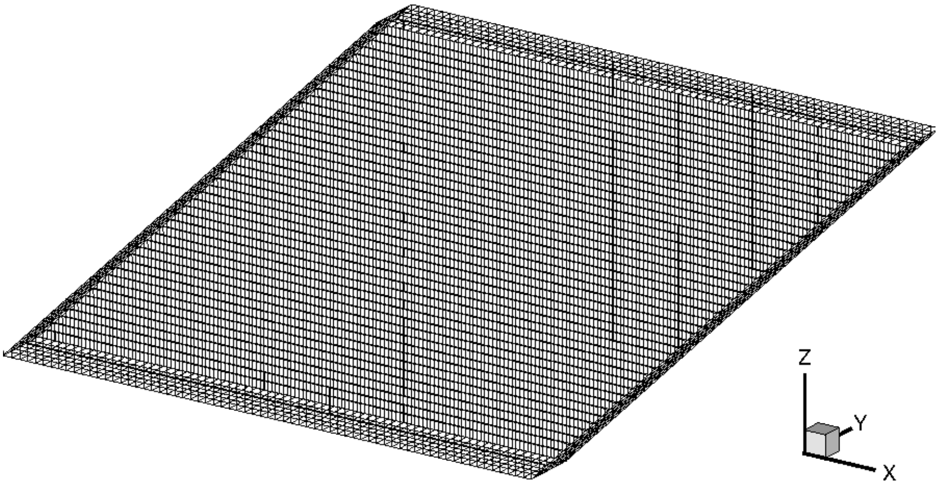

Figure 2.100: Mixed mesh used for skewed moving lid cavity case.

The other mesh investigated, shown in Figure 2.100, consists of hexahedra in the middle of the domain, and tetrahedra at the boundaries. The portion of the domain where maximum pressure is expected is shown in Figure 2.101. The mesh quality indicator plot in Figure 2.102 shows that deviation from non-orthogonality is significant only at the boundaries - a quality not desired as reported in Section 2.1.4.

The results for this section are depicted in Figures 2.106 and 2.107. The residuals are shown in Figures 2.103, 2.104 and 2.105. The meaning of the keys used in the figures are as follows:

- Hex. mesh: Case run with mesh from Figure 2.97, with interpolation from Section 2.4.1.1 and keeping the values of velocity components within physically realistic boundaries: [-1.0, 1.0] for u, v and w.

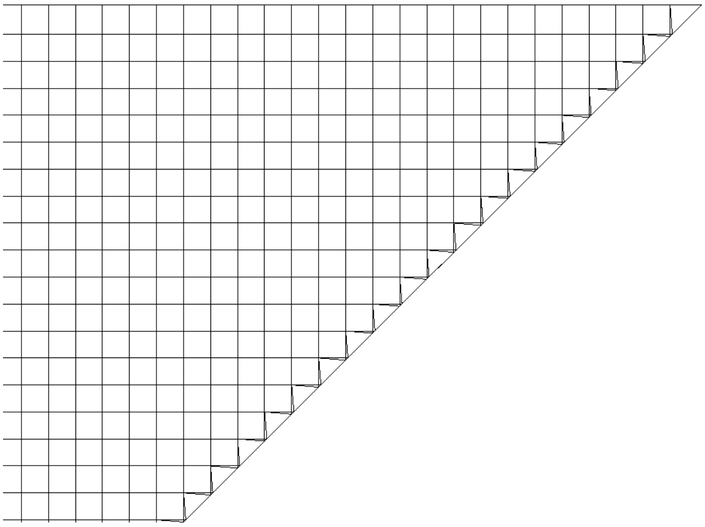

Figure 2.101: Corner where maximum velocity is expected in mixed mesh used for skewed moving lid cavity case.

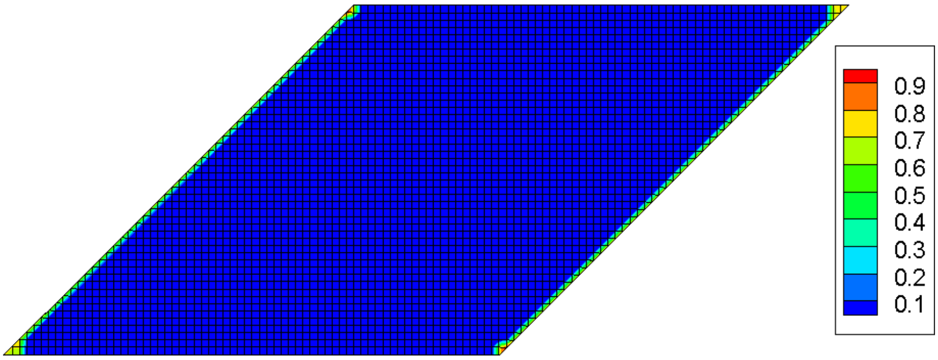

Figure 2.102: Mesh quality indicator for the mixed mesh used for skewed moving lid cavity case. Quality indicator defined in Section 2.1.3.

- Hex. mesh - no interpolation: Case run with mesh from Figure 2.97, keeping the values of velocity components within physically realistic boundaries: [-1.0, 1.0] for u, v and w.

- Hex. mesh - variable interpolation: Case run with mesh from Figure 2.97, with interpolation from Section 2.4.1.1, interpolating the value of the velocity components at bad cells where pressure is interpolated and keeping the values of velocity components within physically realistic boundaries: [-1.0, 1.0] for u, v and w.

- Tet. mesh - PHYSICA default: Case run with mesh from Figure 2.100 without any interpolation or limits on variables.

- Tet. mesh: Case run with mesh from Figure 2.100, with interpolation from Section 2.4.1.1 and keeping the values of velocity components within physically realistic boundaries: [-1.0, 1.0] for u, v and w.

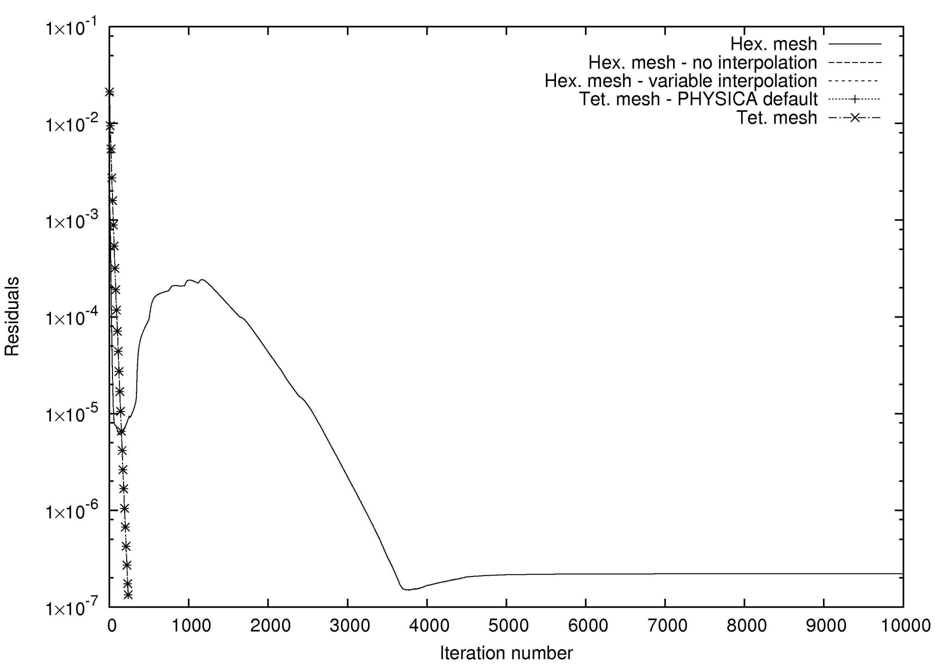

Figure 2.103: Mass residuals for Harpoon meshes. Residual calculation as described in Section 1.2.9.

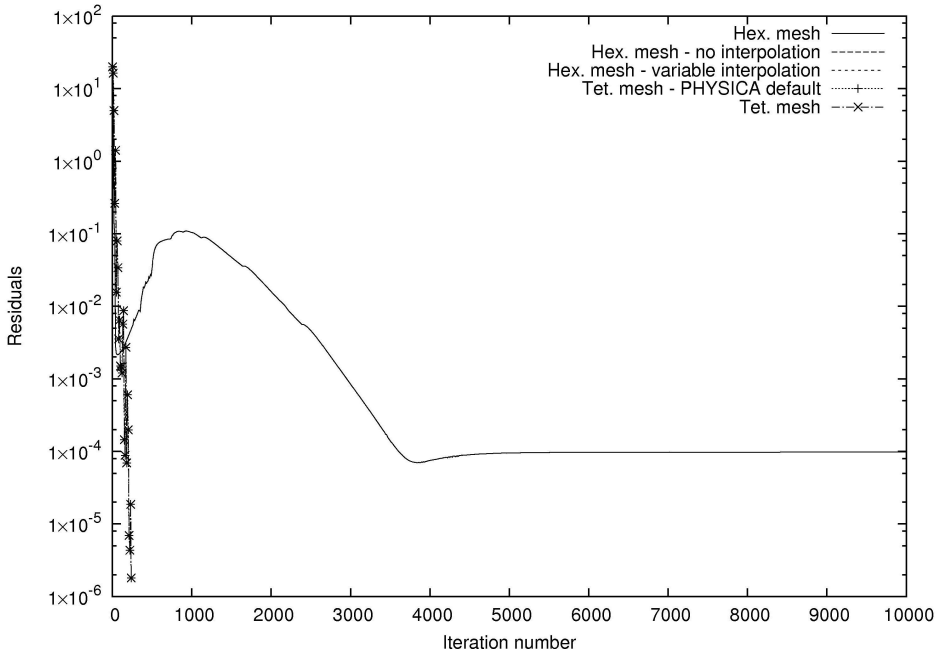

Figure 2.104: Pressure residuals for Harpoon meshes. Residual calculation as described in Section 1.2.9.

The cases run with a mesh from Figure 2.97 converge perfectly whether using a standard approach, or using the interpolation or variable bounding approaches of Section 2.4. With a mixed-mode mesh where non-orthogonality is present at the boundaries, as in Figure 2.100, the standard approach leads to divergence (residuals not shown) and results are obtained using treatments from Section 2.4 only. Moreover, the same result can be recovered when running the case without interpolation, but with variable bounding, suggesting that using an interpolation method to recover a correct solution is overkilling the problem.

This divergence can be explained by the nature of the cell at the top right corner of the geometry, as shown in Figure 2.108. The mass imbalance in this cell is several orders of magnitude higher than in other parts of the domain since only one face is not a fixed boundary, and since flux can only either come in or out of the cell (no flux can go out of boundary walls), mass conservation is not enforced in the cell. This case is therefore a suitable candidate for using the procedure described in Section 2.4.

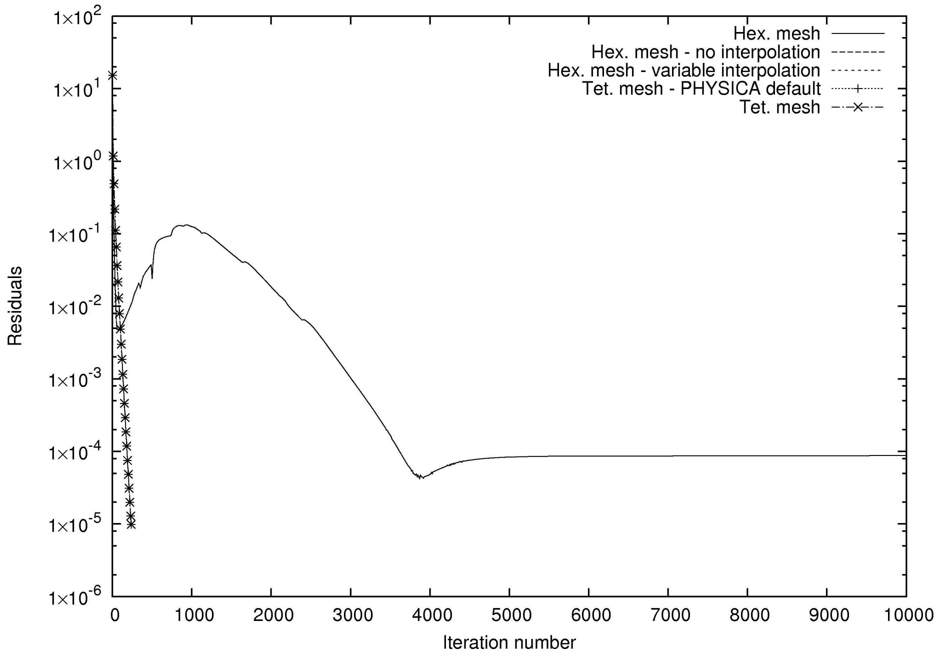

Figure 2.105: U Momentum residuals for Harpoon meshes. Residual calculation as described in Section 1.2.9.

Application of the interpolation method to the mixed mode mesh After determining that the rogue cells contain mass imbalances of around 100 the order of the average mass imbalance at each iteration, the factor 100×average mass imbalance is used to mark control volumes for interpolation. The pressure interpolation method of Section 2.4.1.1 together with variable bounding as in Section 2.4.2 is used to force a solution within the domain meshed with hexahedra and tetrahedra. The interpolation routine was invoked only at the corners with large mass residuals as shown in Figures 2.109 and 2.110.

The mass imbalances were computed at each iteration and the corner cells were automatically picked up using the suggested threshold. The pressure corrections for these cells where then indirectly interpolated as described in Figure 2.10, and these corrections were used for the interpolated cells pressures and velocities.

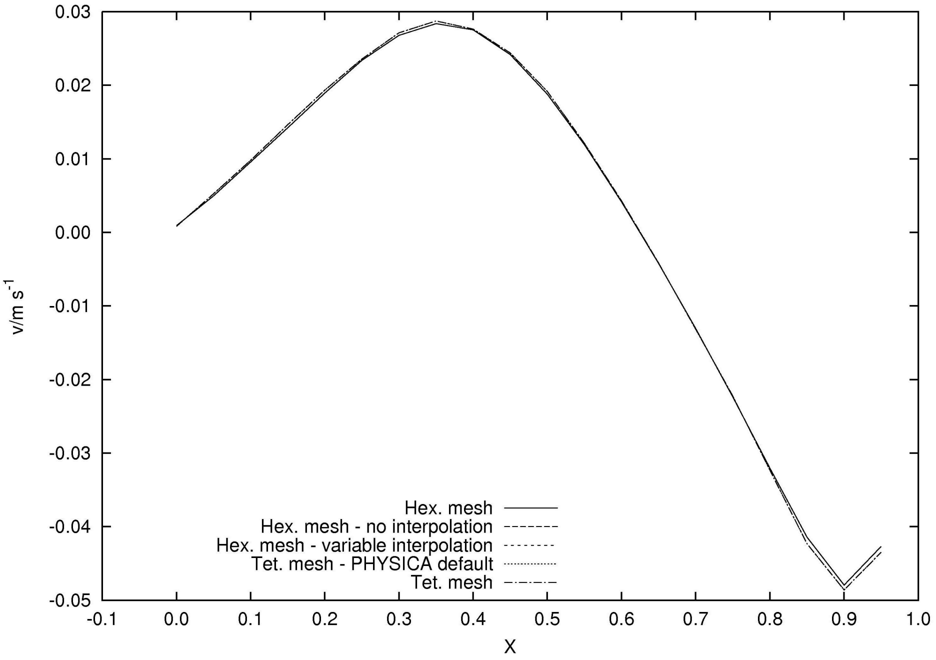

Figure 2.106: Comparison of velocity v along $X = 0.5$ for $\theta = 45°$ and Re = 100. Harpoon generated meshes

The flow fields at the middle of the domain for each run are almost identical, as shown in Figures 2.106 and 2.107, suggesting that the correct result for this case has been achieved. The similarity of the results imply that the variable bounding method of Section 2.4.2 and interpolation method of Section 2.4.1.1 are merely tools to achieve the correct result: once the code is set on the path of convergence with these methods, or if mass imbalances are ignored far from the domain of interest, they will yield the correct solution as though the case was run on a defect free mesh.

The velocity contours for the case run with a tetrahedral mesh and with interpolation are shown in Figure 2.111. The pressure contour lines in Figure 2.112 and a close up on the maximum pressure edge in Figure 2.113. The contours for only one case are presented here since the other contours are identical.

This series of test case demonstrate that sometimes, diffusion corrections alone are

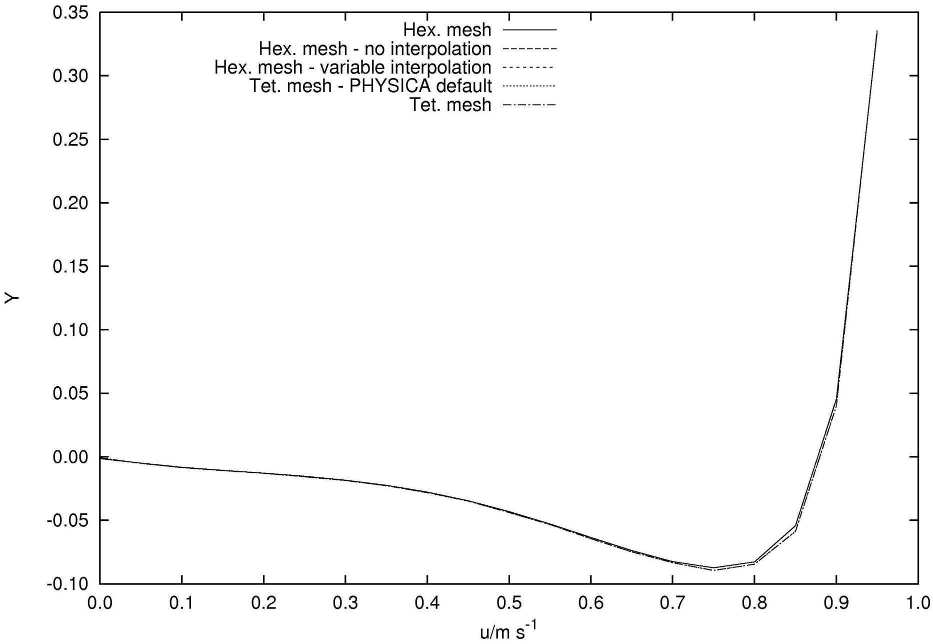

Figure 2.107: Comparison of velocity u along $Y = 0.5$ for $\theta = 45°$ and Re = 100. Harpoon generated meshes

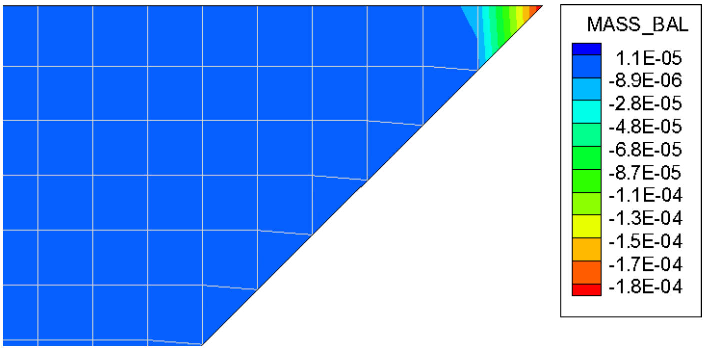

Figure 2.108: Mass imbalance at corner of domain for mixed mode mesh case.

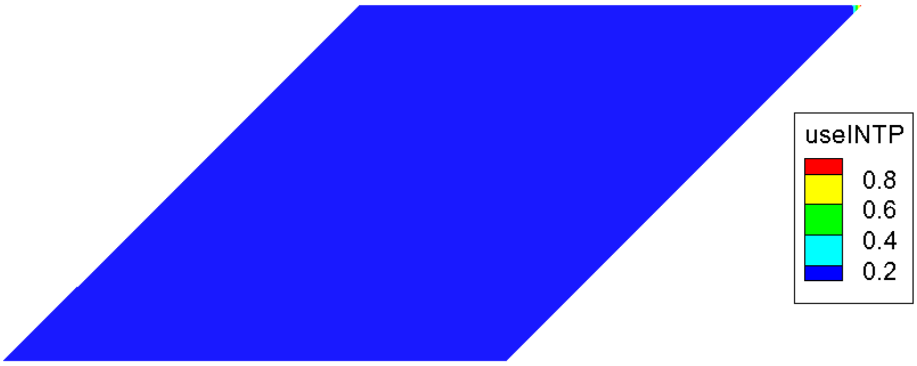

Figure 2.109: Cell distribution where interpolation algorithm is invoked. 1.0 means interpolated cell.

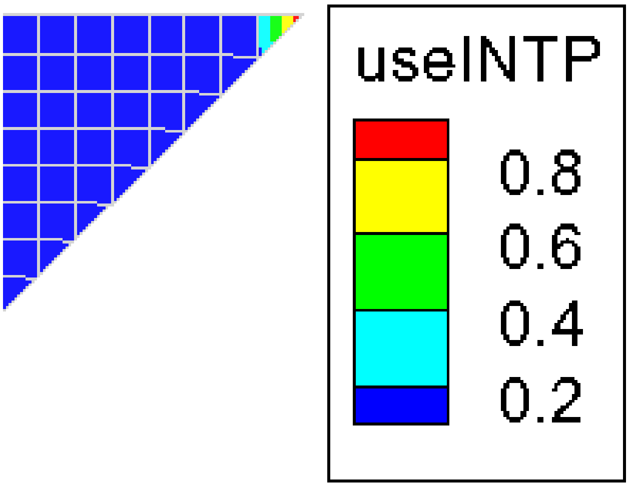

Figure 2.110: Zoom of corner where interpolation algorithm is invoked. 1.0 means interpolated cell.

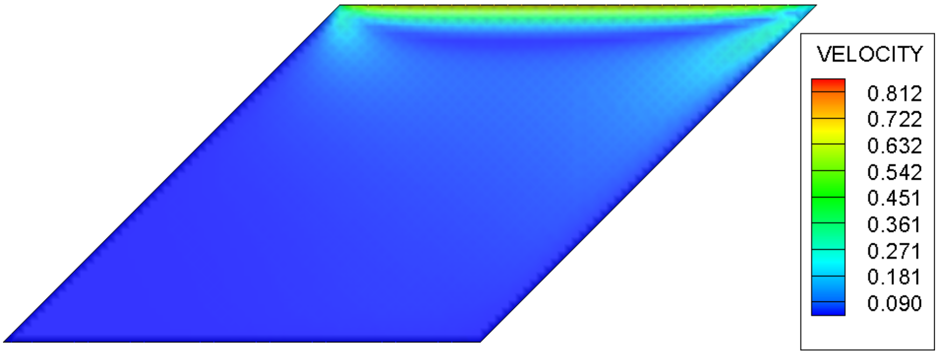

Figure 2.111: Velocity contour for skewed cavity with $\theta = 45°$ and Re = 100. Plane $y = 0.05$ m

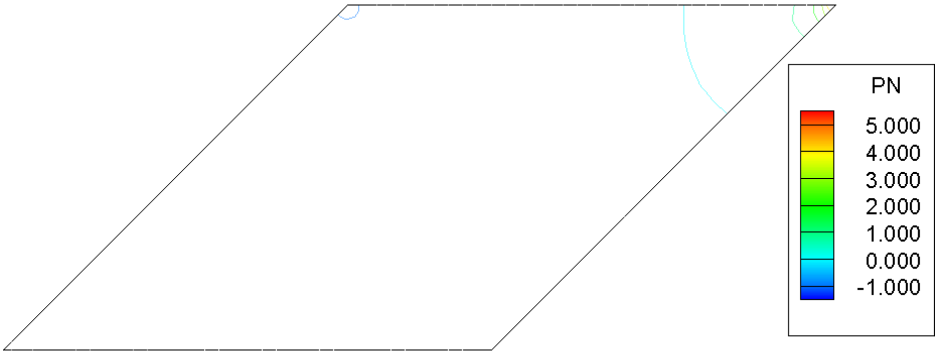

Figure 2.112: Pressure contour lines for skewed cavity with $\theta = 45°$ and Re = 100. Plane $y = 0.05$ m

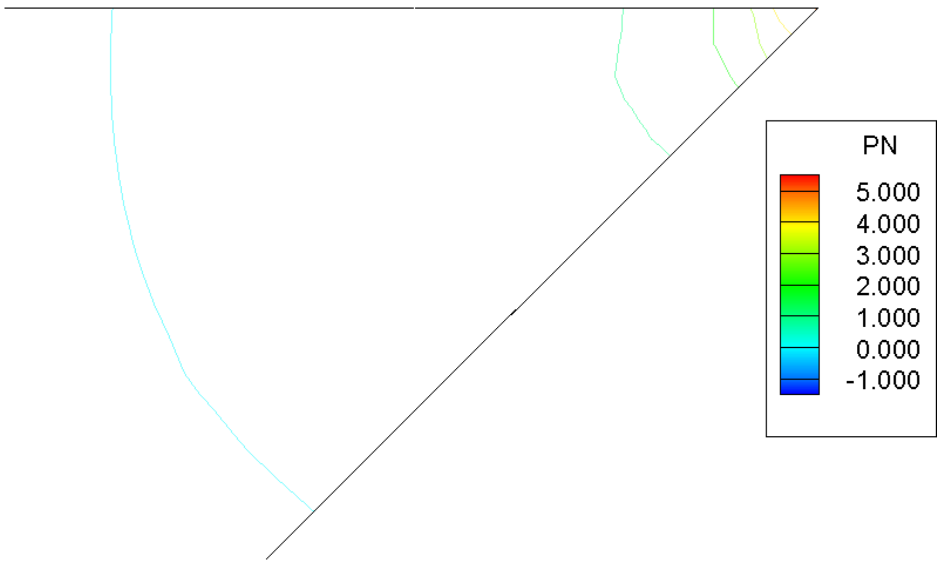

Figure 2.113: Pressure contour lines at top right edge of skewed cavity with $\theta = 45°$ and Re = 100. Plane $y = 0.05$ m

not sufficient to achieve convergence. When mass defects occur only within a small number of rogue cells, the methods described in Section 2.4 can be used to recover a correct solution.

2.5.3 Buoyancy driven flow in skewed cavity

The last series of tests involved heat transfer using a popular test case for non-orthogonal algorithms: the buoyancy driven cavity depicted in Figure 2.114. This test case is run to test whether any special procedure might be required when the heat equation is solved, together with the Navier-Stokes equations, in the presence of non-orthogonality. The flow is driven by density differences arising due to temperature differences in the domain. The Boussinesq approximation, where the gravity term is a simple function of temperature, is used since density differences are expected to be small. The gravity source term in the vertical momentum equation is given by:

$$S_v = \rho_f g \beta \left(T - T_f\right) \tag{2.30}$$

The case described by Demirdžić *et al.* [10] has been repeated here and compared with the proposed benchmark. The top and bottom walls are adiabatic, and the

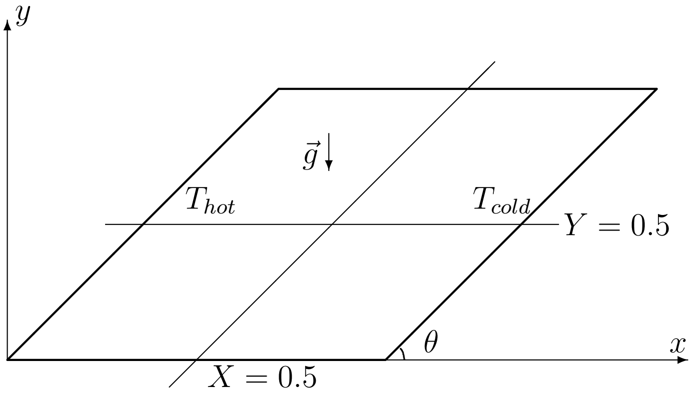

Figure 2.114: Geometry of skewed moving lid cavity.

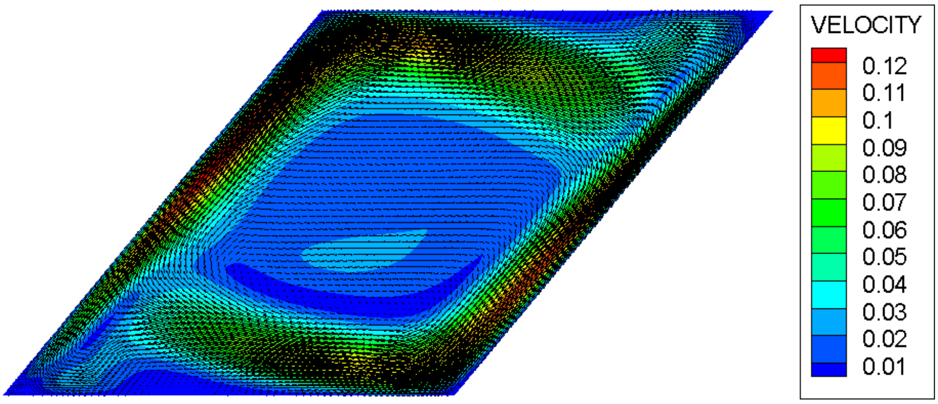

Figure 2.115: Velocity contour for buoyancy driven flow in skewed cavity when Pr = 0.1.

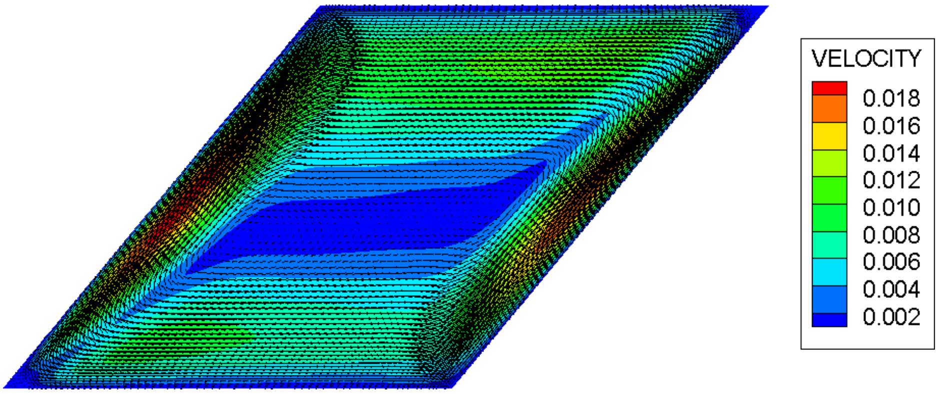

Figure 2.116: Velocity contour for buoyancy driven flow in skewed cavity when Pr = 10.

velocity at the walls is zero and gravity acts downwards. The angle $\theta = 45°$ and two cases are run for flow of Rayleigh number, $\mathrm{Ra} = 10^6$ where

$$\mathrm{Ra} = \frac{g\beta}{\nu^2}\left(T_{hot} - T_{cold}\right) L^3 \mathrm{Pr} \tag{2.31}$$

β is the thermal expansion coefficient, L the cavity length and Pr the Prandlt number defined as

$$\mathrm{Pr} = \frac{\nu}{\varrho} \tag{2.32}$$

where ϱ is thermal diffusivity.

The cases studied are of Prandlt numbers $\mathrm{Pr} = 0.1$ and 10. Only the diffusion corrections from Section 2.3.2.3 are required for convergence in the cases in this section, which have been run with the same meshes as in Section 2.5.2.2. The Rayleigh number is made as a function of kinematic viscosity and Prandtl number only by using $g = 1$, $\beta = 0.1$, $T_{hot} = 1$ and $T_{cold} = 0$, leaving

$$\mathrm{Ra} = \frac{0.1}{\nu^2}\mathrm{Pr} \tag{2.33}$$

For the case with $\mathrm{Pr} = 0.1$, $\nu = 10^{-4}$, and with $\mathrm{Pr} = 10$, the viscosity becomes 10^{-3} .

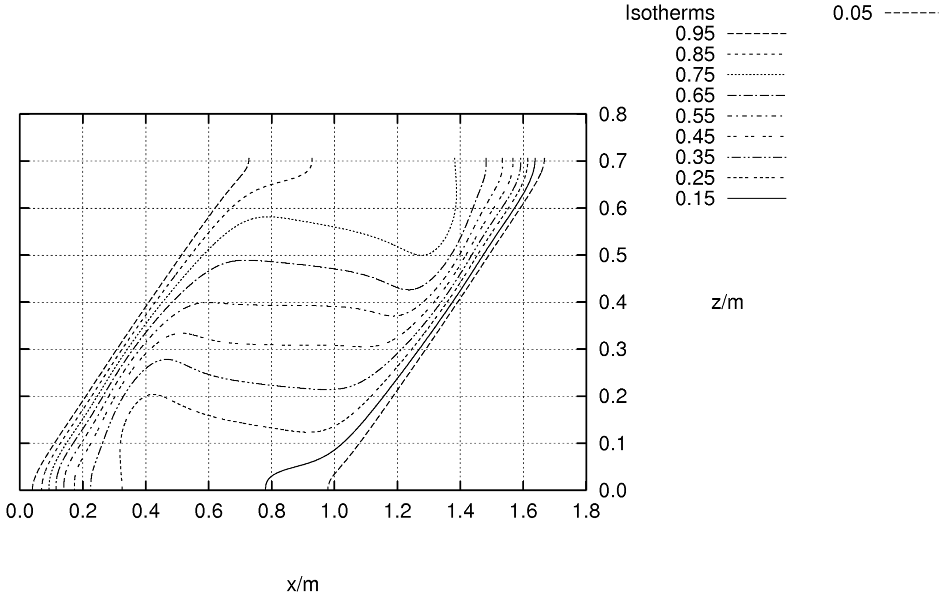

Figure 2.117: Isotherms for Pr = 0.1.

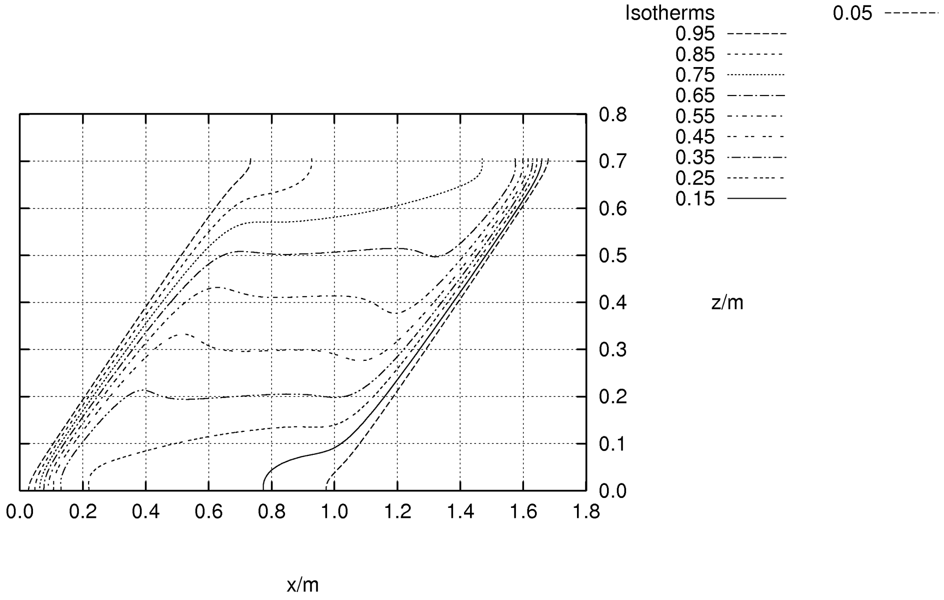

Figure 2.118: Isotherm for Pr = 10.

The results obtained for both cases are shown in Figures 2.115 and 2.116. The Prandtl number has a large influence on the flow pattern, as shown by the contour plots. With Pr = 10, the flow is concentrated along the walls, while for Pr = 0.1, a larger recirculation pattern extends to the top and bottom lids. These results are obtained without having recourse to interpolation and variable bounding from Section 2.4.

The recirculation pattern is also illustrated by the isotherms plotted in Figure 2.117 and Figure 2.118. In both cases, convection results in a large temperature gradient near the hot and cold walls - where velocities are higher than in other parts of the domain.

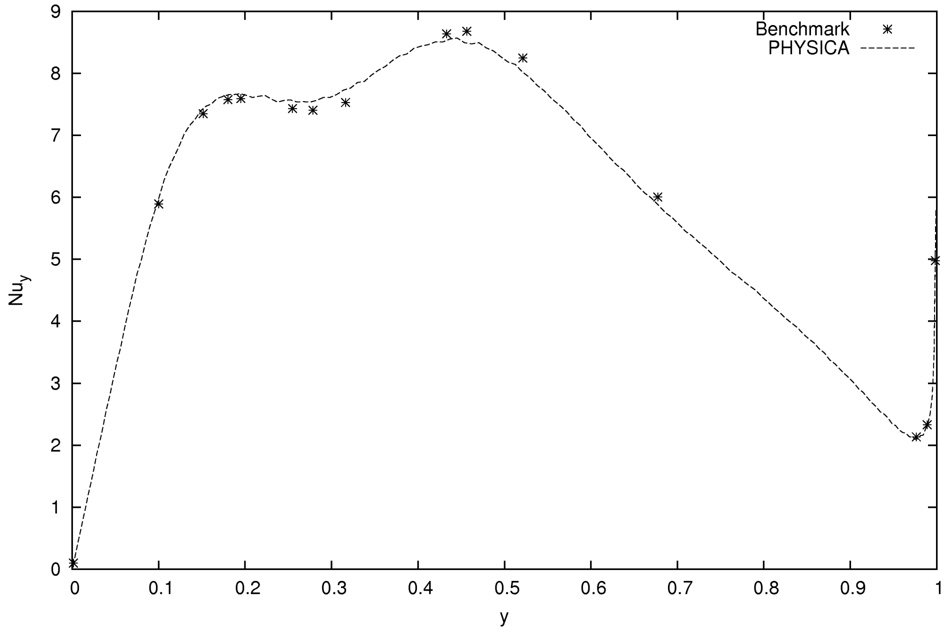

Figure 2.119: Comparison of calculated Nusselt number along cold wall with benchmark from Demirdžić *et al.* [10] when Pr = 0.1.

The profiles of the local Nusselt number, Nu - ratio of convective heat transfer to diffusive (conductive) heat transfer - along the cold wall for each case are shown in Figures 2.119 and 2.120. The Nusselt number is given by

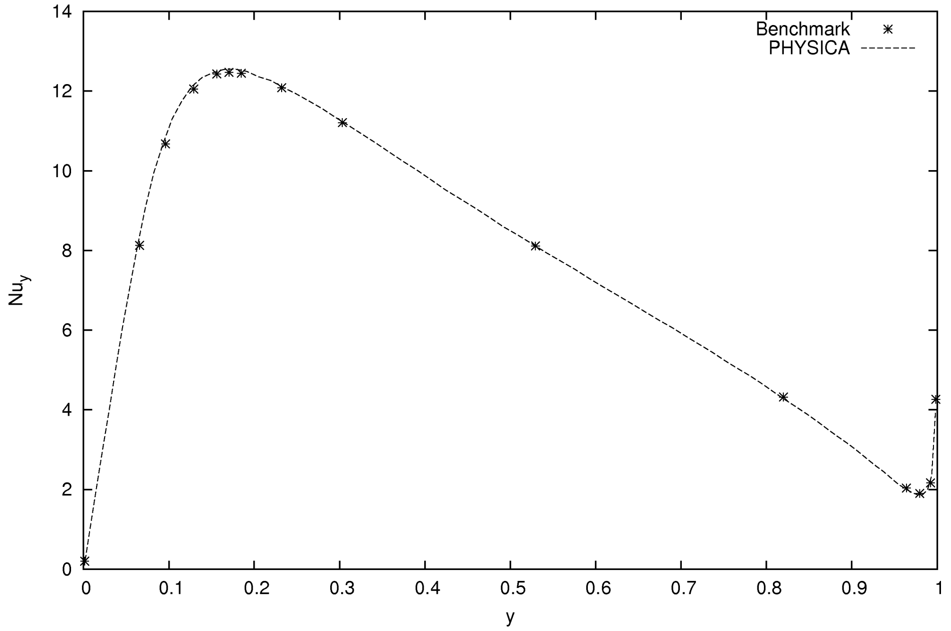

Figure 2.120: Comparison of calculated Nusselt number along cold wall with benchmark from Demirdžić *et al.* [10] when Pr = 10.

$$\mathrm{Nu} = \frac{hL}{\kappa} \tag{2.34}$$

where h is the convective heat coefficient and L the characteristic length of the problem. Both results agree very well with the benchmark from Demirdžić *et al.* [10].

Good agreement is obtained between the numerical method used and the benchmark from Demirdžić *et al.* [10]. The influence of the under-relaxation factors is shown in Figure 2.121. An under-relaxation factor of $\alpha_P = 0.4$ is the optimum choice for most values of α_u. The influence of α_P decreases when α_u is made smaller.

2.6 Summary

This chapter has presented the problem of non-orthogonality within the finite volume framework and remedial algorithms to tackle convergence instabilities in the presence of mesh skewness. The objective of this chapter was to present the validation of the

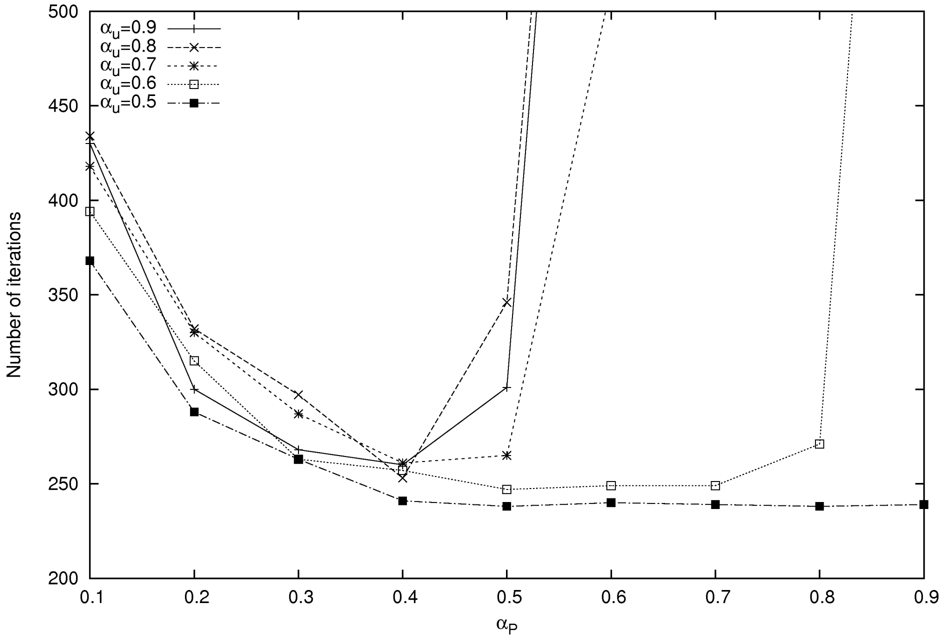

Figure 2.121: Influence of relaxation parameters on the convergence rate of the skewed buoyancy-driven cavity for Pr = 10. Sudden departure indicates divergence for value of α_P and above.

numerical methods developed in the research to tackle instabilities arising in modelling flow in complex geometries.

The results of the computations in this chapter show general good agreement with benchmarks and analytic solutions for the cases of laminar flow between parallel plates and circulation in a cavity due to a moving lid or buoyancy.

A diffusion correction relaxation parameter has been introduced as an adjustment parameter for the diffusion flux correction term. Adjustments made with this parameter do not offer significant speed up to justify a widespread application.

An indirect pressure interpolation method and variable bounding have been used in the extreme case of a three dimensional moving lid involving a “trapped cell” at one extremity. The correct solution was recovered far from the interpolated values. However, the pressure correction interpolation procedure is deemed to be an over-kill since variable bounding suffices to point the iterations to the correct solution.

Chapter 3

COMPRESSIBILITY

Compressible flows are of important engineering interest, e.g. for the prediction of jet flows at high ambient temperatures [5; 74; 75; 76; 77]. The current version of PHYSICA does not support compressibility by default, and users encounter convergence difficulties when working with flows with variable density. The aim of this chapter is to describe a stable implementation of compressibility in a cell-centred, non-staggered CFD code.

This implementation is to be effected on an unstructured, collocated, finite volume, pressure-correction type code. Viscosity is to be modelled since the diffusion coefficients are used in parts of the code to handle boundary conditions. Particular attention is to be paid to:

1. the interpolation of face property values e.g. density, viscosity ... from cell centres,
2. the order of calculation for property evaluation and solved variables,
3. any changes to the pressure correction technique due to variable density,
4. changes to turbulence modelling in compressible flow, and
5. extra source terms in the energy equation arising due to a variable density.

After implementing these changes, the numerical methods developed are validated against compressible cases with either analytic solutions, benchmarks from the literature or empirical data. The convergence behaviour and accuracy of each case is studied. The effect of non-orthogonality on the code is tested by randomly deforming the meshes in some test cases and comparing the results with those obtained using an orthogonal mesh.

3.1 Review

Various techniques for solving compressible flow have been reported in the literature. Most of these reported methods use either Euler equations or the Navier-Stokes equation with density as a solved variable and pressure calculated from the equation of state [78; 79]. However, these techniques are not directly applicable to low speed flows where the pressure-density coupling becomes weak, [80] unless using artificial compressibility [81].

Pressure correction techniques, such as the SIMPLE algorithm [31], are however very stable for incompressible flows, and a modified SIMPLE algorithm has been devised by Demirdžić *et al.* [80] and others [12; 13; 37; 82] to solve flows at all speeds. Such a technique is particularly attractive when modelling fluid flow at different speeds. For example, in a high speed compressible jet, the same algorithm can be used to model the compressible zone in the jet, and the low speed fluid in the ambient surroundings. Such a method has applications in modelling rocket nozzles or oxygen jets in blast furnaces. Pressure is the solved variable, since pressure variation remains significant in both compressible and incompressible flows.

In compressible flows, velocities are generally expressed in term of Mach number M defined by

$$M = \frac{u}{a} \tag{3.1}$$

where the speed of sound a is

$$a = \sqrt{\gamma R_w T} \tag{3.2}$$

3.2 Implementation of compressibility on an incompressible code

The implementation of compressibility requires some modifications to the standard incompressible procedure. The boundary conditions are different, since all values at supersonic inlets must be prescribed. The energy balance equation is to be modified to include the differences due to fluid kinetic energy. Turbulence models for compressible flow are still under investigation and care has to be taken when invoking them for supersonic flows.

3.2.1 Energy conservation equation for compressible cases

The energy equation for compressible flow is given by

$$\begin{aligned}\frac{\partial\left(\rho h_t\right)}{\partial t} + \nabla\cdot\left(\rho\boldsymbol{u}h_t\right) &= \nabla\cdot\left(\kappa\nabla T\right) + \frac{\partial p}{\partial t}\\ &+ \frac{\partial\left(u\tau_{xx}\right)}{\partial x} + \frac{\partial\left(u\tau_{yx}\right)}{\partial y} + \frac{\partial\left(u\tau_{zx}\right)}{\partial z}\\ &+ \frac{\partial\left(v\tau_{xy}\right)}{\partial x} + \frac{\partial\left(v\tau_{yy}\right)}{\partial y} + \frac{\partial\left(v\tau_{zy}\right)}{\partial z}\\ &+ \frac{\partial\left(w\tau_{xz}\right)}{\partial x} + \frac{\partial\left(w\tau_{yz}\right)}{\partial y} + \frac{\partial\left(w\tau_{zz}\right)}{\partial z} + S_{h_t}\end{aligned} \tag{3.3}$$

where h_t in equation (3.3) is total enthalpy given by

$$h_T = C_v T + \frac{p}{\rho} + \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{3.4}$$

For Newtonian fluids, the nine viscous stress components are

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} + \lambda\nabla\cdot\boldsymbol{u} \tag{3.5}$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda \nabla \cdot \boldsymbol{u} \tag{3.6}$$

$$\tau_{zz} = 2\mu \frac{\partial w}{\partial z} + \lambda \nabla \cdot \boldsymbol{u} \tag{3.7}$$

$$\tau_{xy} = \tau_{yz} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \tag{3.8}$$

$$\tau_{xz} = \tau_{zx} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \tag{3.9}$$

$$\tau_{yz} = \tau_{zy} = \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial z} \right) \tag{3.10}$$

The second viscosity λ is approximated as $-\frac{2}{3}\mu$ [83].

The extra terms arising due to work done by viscous stresses in the compressible heat imbalance equation are implemented as sources. The gradients of each viscous stress are calculated in each cell using the Newtonian assumption and these gradients are added to the source array for each control volume.

The specific enthalpy h is given by

$$h = C_v T + \frac{p}{\rho} \tag{3.11}$$

The specific energy E is given by

$$E = C_v T + \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{3.12}$$

3.2.1.1 Temperature from enthalpy

If enthalpy is a solved variable, temperature has to be updated using the relationship

$$T = \frac{E - \frac{1}{2}\left(u^2 + v^2 + w^2\right)}{C_v} \tag{3.13}$$

3.2.2 Boundary conditions for compressible cases

For a compressible case, all solved variables are to be prescribed at the inlet and extrapolated at the outlet [37; 80].

3.2.2.1 Fixed value boundary condition implementation

When the face value is known, the coefficient C_ϕ as defined in equation (1.24) is taken as the diffusion link between the face and neighbouring element [15]

$$C_\phi = \frac{A\Gamma_\phi}{d} \tag{3.14}$$

where d is the distance between the element and the face. The fixed boundary condition is implemented using a source of the form

$$\left[C_\phi + \max\left(-F_\phi \boldsymbol{u} \cdot \boldsymbol{n} A, 0.0\right)\right]\left(\phi_f - \phi_P\right) \tag{3.15}$$

where F_ϕ is the convection term coefficient [15].

3.2.2.2 Total conditions at inlet

When solving for compressible flows, all values are prescribed at the inlet. The inlet velocity is indirectly prescribed from the Mach number calculated from total pressure given by

$$p_T = p\left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}} \tag{3.16}$$

The total pressure p_T and the flow direction are known and prescribed at the inlet. The pressure is extrapolated from the adjacent cell in the domain, and equation (3.16) is used to calculate the Mach number M. The velocity components are then calculated using M, the speed of sound at the inlet (dependent on T) and the flow direction. The coefficient of the velocity source is then given by the contents of the first brackets in equation (3.15). ϕ_f is the velocity component.

If the flow at the inlet is supersonic, the pressure p is also prescribed at the inlet, which effectively fixes M to the prescribed value. The pressure correction term at the inlet is therefore fixed to zero, since pressure need not to be corrected for.

The gas enthalpy is prescribed at the inlet as a fixed value, using the same formulation as in Section 3.2.2.1. The temperature at the inlet is then calculated using equation (3.13).

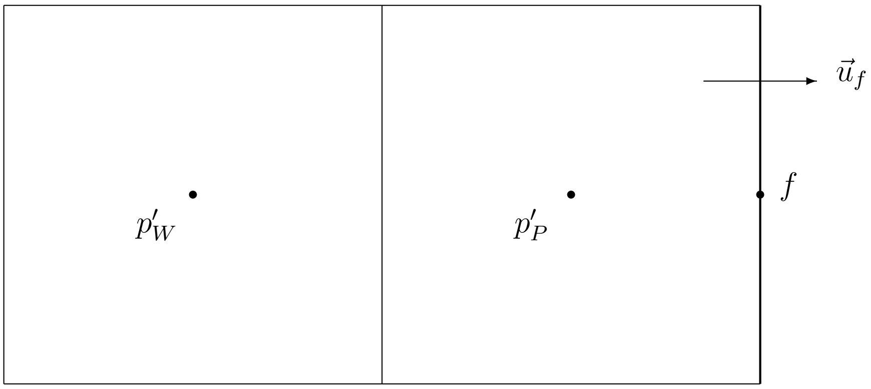

Figure 3.1: Extrapolated boundary condition for outlet f.

3.2.2.3 Extrapolated boundary condition

Solved variables are usually extrapolated at the outlet for compressible problems if the flow is supersonic [78]. This boundary condition is implemented by fixing the face values at the outlet using the nearest cell values. For a boundary face f as shown in Figure 3.1, the pressure correction is to be extrapolated using a linear function of p'_P and p'_W [80].

Using the notation of Section 3.2.2.1, the coefficient of the source at the outlet is given by $C_\phi + \max\left(-F_\phi \boldsymbol{u} \cdot \boldsymbol{n} A, 0.0\right)$ and the face value is

$$\phi_f = \frac{\frac{\phi_P}{d_P} + \frac{\phi_W}{d_W}}{\frac{1}{d_P} + \frac{1}{d_W}} \tag{3.17}$$

where d is the element to face distance.

3.2.3 Density update

The material property density is to be updated at each iteration either assuming isentropic conditions or using the ideal gas law.

3.2.3.1 Isentropic relation between pressure and density

The isentropic relations between absolute pressure and density for steady compressible flows are given by

$$\frac{\rho}{\rho_T} = \left(\frac{p}{p_T}\right)^{\frac{1}{\gamma}} \tag{3.18}$$

where γ is the ratio of specific heats and the subscript $_T$ denotes a total variable.

3.2.3.2 Ideal gas law

The ideal gas equation also governs the physics of the problem for compressible cases.

$$p = \rho R_w T \tag{3.19}$$

where R_w is given by

$$R_w = \frac{R}{\text{mol. weight of gas}} \tag{3.20}$$

R is the ideal gas constant equal to 8.314472 J mol^{-1} K^{-1}.

Since C_p is related to C_v by

$$C_p = C_v - R_w \tag{3.21}$$

the expression for total enthalpy (3.4) can be conveniently reduced to

$$h_T = C_p T + \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{3.22}$$

3.2.4 Modifications to pressure correction procedure

The pressure correction equation, as derived in Appendix A, is modified to account for variable density. The derivation of the compressible pressure correction equation was left as an exercise in Patankar [24]. Demirdžić *et al.* [80] derived the case for an isentropic flow. Date [12] derived a similar pressure correction equation in a curvilinear coordinate system. Here, we will derive the general expression for a compressible ideal gas for a collocated finite volume solver.

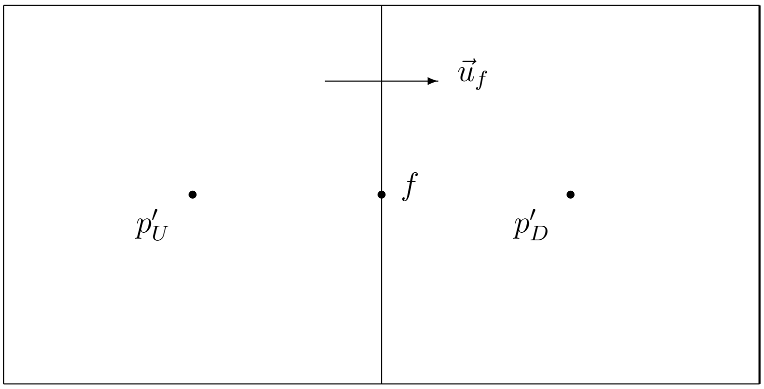

Figure 3.2: Upwind and downwind cells for face f.

Consider the 1D configuration below (extension to other dimensions is straightforward).

The face velocity and density can be expressed as the sum of two terms: a variable field not satisfying continuity, denoted by a $*$ superscript and a correction term, denoted by $'$.

$$u_f = u_f^* + u_f' = u_f^* + d_f\left(p_U' - p_D'\right) \tag{3.23}$$

where $d_f = \frac{1}{a_P} A_f n_x$ as in equation (A.9) for the derivation of an incompressible pressure correction procedure.

$$\rho_f = \rho_f^* + \rho_f' = \rho_f^* + \iota p_f' \tag{3.24}$$

where $\iota = \frac{1}{R_w T_f}$.

A volume integration of the continuity equation, as applied to a configuration as shown in Figure 3.3, followed by the application of Gauss's theorem results in an equation of the form:

$$\left(\rho A u\right)_e - \left(\rho A u\right)_w = 0 \tag{3.25}$$

The following approximation is used

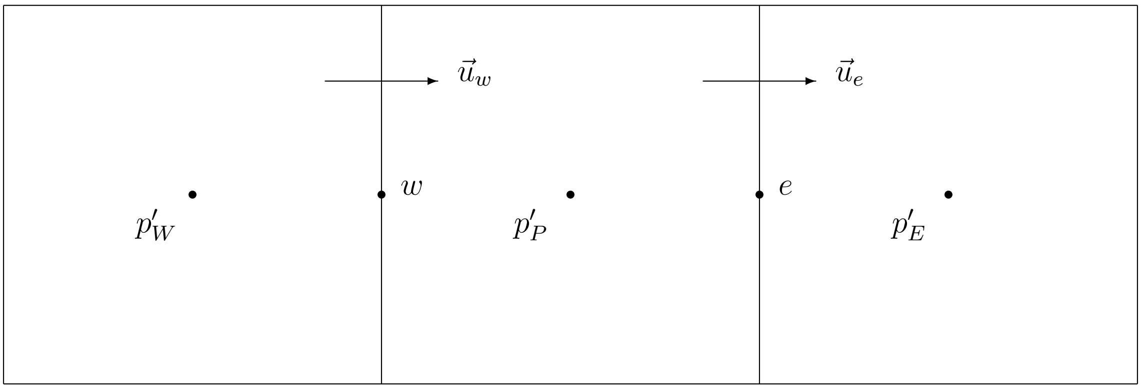

Figure 3.3: Derivation of 1D pressure correction equation for compressible flow.

$$(\rho u) = (\rho^* + \rho')(u^* + u') \approx \rho^* u^* + \rho' u^* + \rho^* u' \tag{3.26}$$

The second order term $\rho' u'$ is dropped.

Dropping the area terms for the sake of clarity, equation (3.25) then becomes

$$\rho_e^* u_e^* + \iota_e p_e' u_e^* + \rho_e^* d_e (p_P' - p_E') - \rho_w^* u_w^* - \iota_w p_w' u_w^* - \rho_w^* d_w (p_W' - p_P') = 0 \tag{3.27}$$

Velocities are large for compressible cases, resulting in large Peclet numbers. The upwind scheme is appropriate to approximate these face velocities. For flows flowing from west to east, the above equation becomes

$$\rho_e^* u_e^* + \iota_P p_P' u_e^* + \rho_e^* d_e (p_P' - p_E') - \rho_w^* u_w^* - \iota_W p_W' u_w^* - \rho_w^* d_w (p_W' - p_P') = 0 \tag{3.28}$$

Recognising that $A_f \rho_f^* d_f$ is a diffusion term and $\iota_{upw} u_f^* A_f$ a convection term, the notation

$$D_f = A_f \rho_f^* d_f \tag{3.29}$$

and

$$F_f = \iota_{upw} u_f^* A_f \tag{3.30}$$

can be used to generalise equation (3.28) to

$$\sum_{f}\left[D_f+\max\left(\mp F_f,0\right)\right]\left(p'_P-p'_A\right)\pm F_f p'_P=m^*_P \tag{3.31}$$

Expressing the pressure equation in the linear form (1.10)

$$a_P p'_P=\sum_{i} a_i p'_i-m^*_P \tag{3.32}$$

In the 1D example presented in this section,

$$a_w=D_w+\max\left(F_w,0\right) \tag{3.33}$$

$$a_e=D_e+\max\left(-F_e,0\right) \tag{3.34}$$

$$a_P=a_e+a_w+\left(F_e-F_w\right) \tag{3.35}$$

$$m^*_P=\left(\rho^*A^*u^*\right)_e-\left(\rho^*A^*u^*\right)_w \tag{3.36}$$

The convection terms for compressible pressure correction equation can be implemented in a collocated code by storing ι in all cell centres, and then assembling the face convection terms using upwinding.

Convergence behaviour of modified pressure correction procedure The derived pressure correction equation with corrections due to variable density contains the undesired feature that a_P is not equal to the sum of the neighbouring coefficients when mass convergence is not achieved. This would mean that, for a solution which is not mass conserved, a cell with neighbours having the same pressure correction value would have a different pressure correction value after solving the linear equations (3.32). However, this is not worrying in the context of most compressible computations since [80]:

1. The pressure correction equation is not a conserved variable equation, but merely a pointer towards mass conservation. The equation from the current section has been derived using the ideal gas equation which is the physical link between

density, pressure and temperature for compressible gas flows. The corrections described in this sections will therefore point the solution towards the correct mass conservation, which takes into account density variations.

2. At convergence, all pressure corrections and mass sources will become zero, and a_P will be equal to the sum of neighbouring coefficients since the algebraic sum of all convection terms will be zero due to mass conservation, i.e. $F_e - F_w = 0$.

However, the way the pressure correction is implemented, with density correction terms in the pressure correction equation, excludes cases where indirect pressure interpolation from Section 2.4.1.1 is invoked. Mass conservation is not ensured when this interpolation method is used, and an under-relaxation term is developed in Section 3.2.5 to address stabilisation concerns if mass conservation is not achieved.

3.2.4.1 Resolution method for face density

Moukalled and Darwish [13] suggested that using a high-resolution scheme to calculate m_P^* in pressure correction equations of the form (3.32) enhance the shock-capturing property of the algorithm. They tested their method with the case of flow inside a channel with a circular bump: the same case will be presented in Section 3.3.3 where the code developed in this chapter will be compared against theirs.

A possible way of implementing a similar higher-order resolution method is to use the SMART scheme presented in Section 1.2.3.2 to calculate face density values.

$$\rho_f = \rho_U + 0.5\Psi\left(r\right)\left(\rho_U - \rho_{UU}\right) \tag{3.37}$$

These face values are then used to calculate the estimated mass imbalance m^* in each cell. Where an “upwind-upwind” value is not available for a face, the upwind scheme is invoked.

3.2.5 False time step term for compressible cases

A false time step term is added to the pressure correction equation to act as under-relaxation should numerical instability arise with the pressure correction procedure of Section 3.2.4. The transient equivalent of this term is

$$\frac{\rho^o V^o - \rho V}{\Delta t} \tag{3.38}$$

The old and new cell volumes are identical if no mesh adaptation technique is used. The density difference can be treated as

$$\rho^o - \rho = \rho' = \iota p' \tag{3.39}$$

where ι is a constant derived from the ideal gas equation ($\frac{1}{R_w T}$). The false time step term is therefore

$$\frac{\iota V}{\delta t_f} p' \tag{3.40}$$

When expressed in source coefficient-value form, the coefficient for the pseudo-time step term is $\frac{\iota V}{\delta t_f}$ and the value is 0. At convergence, this source term becomes zero, since all pressure corrections are zero when mass conservation is achieved.

3.2.6 Solution procedure for compressible cases

The solution procedure is outlined in the flowchart shown in Figure 3.4. The pressure correction equation used for the compressible SIMPLEC algorithm is described in Section 3.2.4.

In particular, the density needs to be updated just before solving for momentum and pressure, so that the same velocities and densities with which mass conservation has been achieved are used to solve for heat balance. Therefore, density updates always occur after solving the heat balance equation within each iteration. Also, care has to be taken in using the absolute values of pressures and temperatures (in kelvins) for density updates.

No relaxation is used to limit the density changes at each iteration, since the density in each loop should match the physical density as computed from pressure and temperature within each loop. Relaxing the densities would lead to a non-physical value of density, which deviates from the ideal gas law used to derive the pressure correction procedure of Section 3.2.4.

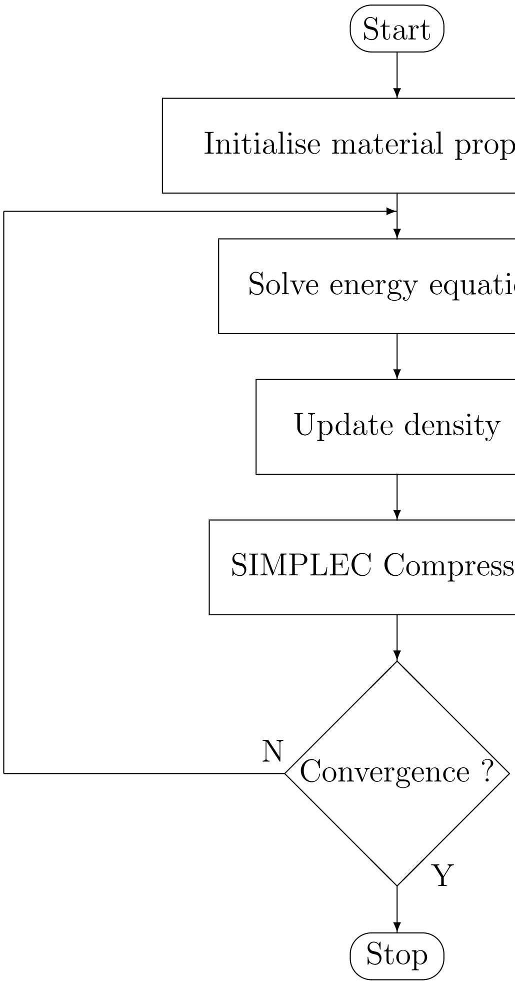

Figure 3.4: Algorithm for steady state (or iterations within a time-step of) compressible cases.

3.2.7 Turbulence modelling in compressible flows

Turbulence modelling is required in the computation of a variety of large Reynolds number flows, including high speed flows and large temperature cases. Each different physical situation requires a different treatment for turbulence modelling. Of particular interest in this research is the effect of turbulence on compressible flows with large temperature gradients, which is currently under extensive investigation from different authors [4; 5; 76].

Two main approaches for modelling the complex physical phenomenon which is turbulence are the Large Eddy Simulation (LES) approach and solving Reynolds-Averaged Navier Stokes (RANS) equations. The former is yet too computationally expensive, even for simple nozzles at low Reynolds number [77; 84]. Running a Large Eddy Simulation would require full 3-D modelling of cases which could be run with an axisymmetric assumption, and very small time steps. However, the processing requirements of RANS methods are practical enough [4].

The most common turbulence models based on time averaged Reynolds equations include Prandtl's mixing length model, two-equation models such as $k - \varepsilon$ model [22; 23], Reynolds stress equation models and algebraic stress models [25].

The mixing length model is generally well validated for axisymmetric jets [85] and is the least expensive form of turbulence modelling. However, this model is too simple and cannot describe flows with recirculation [25].

Reynolds stress equation models use six partial differential equations to model the transport of each of the six independent Reynolds stresses arising in the momentum conservation equations - hence catering for anisotropic flows. In the algebraic stress equation models, the convection and diffusion terms in the Reynolds stress equations are either removed or modelled, such that the six equations reduce to a simple set of linear equations to be solved simultaneously [25]. Tago and Higuchi [74] used the Reynold Stress Model (RSM) to cope with the anisotropy of μ_t. However, Abdol-Hamid *et al.* argue that there is no need to use these complex models, since studies show that more complex algebraic stress models do not impact much on mixing effect [4]. Turbulent mixing is of paramount importance to jet simulations, as presented in the next section.

We are therefore left with the well-established two-equation models.

3.2.7.1 Changes to the $k - \varepsilon$ turbulence model for jet flows

A jet is a region of high speed fluid flow surrounded by a static fluid [25]. When a supersonic jet interacts with the ambient gas, it produces a region of turbulent mixing as shown in Figure 3.5 [86]. The jet expands to its surroundings and the velocity decreases further down the jet. The effect of compressibility on jets is to reduce turbulent mixing, resulting in a longer core length [87]. Pamamoschou and Roshko [88] conducted an empirical investigation of the compressible turbulent shear layer and quantified this compressibility effect with a parameter: the convective Mach number - a parameter dependent on the speeds of sound and densities in the jet and in the ambient gas [86].

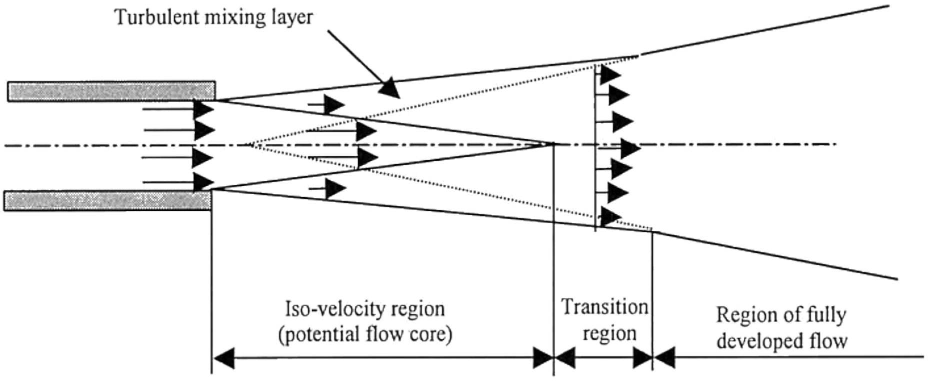

Figure 3.5: Schematic diagram of the different regions of jet flow. A jet consists of an initial flow core, followed by a decreasing velocity profile. Figure from Allemand *et al.* [86]

Two equation models like the $k - \varepsilon$ model [22; 23] are known to be inaccurate when modelling axisymmetric jets [89]. The empirical constants used in these models have been chosen to produce good results for modelling subsonic flows at room temperature, and are not directly applicable to high temperature, supersonic flows [89]. The standard model cannot be trusted to correctly predict the growth rate of a compressible jet. Sarkar *et al.* [90] recognised that another dissipation term needs to be modelled in high speed flows: compressible dissipation.

The following subsections will review different modifications to the $k - \varepsilon$ model to account for the effect of compressibility on turbulent mixing and growth.

3.2.7.2 Heinz's method [3]

Sarkar [91] suggested a turbulence parameter in compressible shear flows: the gradient Mach number M_g defined as

$$M_g = \frac{Sl}{a} \tag{3.41}$$

where S is the constant mean shear rate, l is the integral lengthscale of velocity fluctuations in the direction of the shear and a, the speed of sound. Sarkar [91] determined that the turbulent energy growth rate decreases significantly with increasing gradient Mach number. Heinz [3] used Sarkar's results [91] to derive a new value for C_μ as

$$C_\mu = 0.07 \exp\left(-0.4 M_g\right) \tag{3.42}$$

3.2.7.3 Abdol-Hamid *et al.*'s method [4]

Although there have been attempts to make the $k - \varepsilon$ model sensitive to temperature fluctuations [92; 93; 94], no other model incorporated temperature fluctuations at high speed flows. In order to account for both high speed flows and temperature variations, Abdol-Hamid *et al.* [4] made C_μ a variable dependent on two parameters: the total temperature gradient normalised by the turbulence length scale, T_g:

$$T_g = \sqrt{\left(\frac{\partial T_t}{\partial x_i}\right)^2} \frac{\left(k^{\frac{3}{2}}/\varepsilon\right)}{T_t} \tag{3.43}$$

and the turbulence Mach number M_τ:

$$M_\tau = \frac{\sqrt{2k}}{a} \tag{3.44}$$

Abdol-Hamid *et al.*'s correction to C_μ assumes the form

$$C_\mu = 0.09 C_T \tag{3.45}$$

$$C_T = 1 + \frac{T_g^3}{0.041 + f\left(M_\tau\right)} \tag{3.46}$$

with

$$f\left(M_\tau\right) = \left(M_\tau^2 - M_{\tau 0}^2\right) H\left(M_\tau^2 - M_{\tau 0}^2\right) \tag{3.47}$$

$H(x)$ is the Heaviside function and $M_{\tau 0} = 0.1$. The constants in equation (3.45) have been determined from experimental data. C_μ reduces to the usual value of 0.09 for no compressibility and no temperature gradients.

3.2.7.4 Alam *et al.*'s method [5]

Alam *et al.* [5; 76] were interested in a cold jet entering a hot environment and modified Abdol-Hamid *et al.*'s model [4] by dividing 0.09 by C_T in order to recover the observed decrease in growth rate of mixing.

$$C_\mu = \frac{0.09}{C_T} \tag{3.48}$$

They determined new constants for C_T by matching experimental data. In addition to using this modification, they used a turbulent Prandtl number σ_T of 0.5 as recommended by Wilcox [95] in order to increase the heat transfer from the surrounding to the jet, via the turbulent shear layer [76].

Both models from Abdol-Hamid *et al.* [4] and Alam *et al.* [5] can be singly expressed as a modification of C_μ in the form

$$C_\mu = 0.09 C_T^n \tag{3.49}$$

where

$$n = \begin{cases} +1 & \text{if } T_{jet} > T_{ambient} \\ -1 & \text{otherwise} \end{cases} \tag{3.50}$$

This formulation is general and covers both cases of a hot jet entering a cold environment, and vice versa.

3.2.7.5 Turbulent Prandtl number

The turbulent Prandtl number σ_t appears in the effective thermal diffusion term Γ_t as

$$\Gamma_t = \frac{\mu_t}{\sigma_t} \tag{3.51}$$

Since μ_t is calculated using C_μ, the different turbulence modifications reviewed above are, as far as turbulent heat diffusion is concerned, also equivalent to keeping the standard $k - \varepsilon$ model and modifying the turbulent Prandtl number in regions of large Mach number and temperature gradient variation [4].

3.3 Compressibility test cases

The results of the compressibility investigations are described and discussed in this section. The mesh used in each case is presented and described, together with the non-orthogonality measure presented in Section 2.1.3 for cases where skewness is significant. The indirect pressure interpolation method of Section 2.4.1.1 is not used for these test cases. The compressible pressure correction method from Section 3.2.4 and the boundary conditions from Section 3.2.2 are tested in this section. Of particular significance, the full Navier Stokes equation is being solved in each case, not just the inviscid Euler formulation.

An oxygen jet case is finally presented, and run with the modifications to the turbulence model proposed by [5]. These results are compared with another model developed specifically for cold jets entering in hot environments - derivation of which is presented in Section 3.3.4.1. The convergence behaviour of each case, including mass balance plots, is also described and discussed.

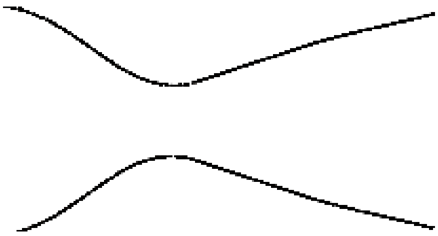

Figure 3.6: de Laval nozzle.

3.3.1 1D converging-diverging nozzle

The de Laval nozzle (also known as convergent-divergent nozzle) is a tube consisting of a converging section followed by a diverging section [96]. The fluid inside the nozzle is accelerated inside the nozzle. If properly designed, the speed of the fluid should reach the speed of sound at the throat (the region where the converging and diverging sections meet).

The de Laval nozzle is one of the simplest benchmarks which can be used to test the accuracy of the numerical methods proposed in the previous sections, bar turbulence modifications. The computed results from this case will be compared with an analytic solution. The nozzle case covers the transition range, with the Mach number expected to be in the range of [0.3,2].

By applying mass conservation to 1D isentropic flow, it can be shown that the area ratio of the nozzle is given by

$$\left(1 - M^2\right) \frac{1}{u} \frac{\mathrm{d}u}{\mathrm{d}x} = -\frac{1}{A} \frac{\mathrm{d}A}{\mathrm{d}x} \tag{3.52}$$

where M, u and A are the Mach number, velocity and area at point x respectively.

When the flow is subsonic, i.e. $M < 1$, $\frac{\mathrm{d}u}{\mathrm{d}x}$ and $\frac{\mathrm{d}A}{\mathrm{d}x}$ are of opposite signs: constricting the nozzle in this regime will accelerate the gas. For supersonic flow, i.e. $M > 1$, $\frac{\mathrm{d}u}{\mathrm{d}x}$ and $\frac{\mathrm{d}A}{\mathrm{d}x}$ have the same sign: increasing the cross-section will accelerate the flow. $\frac{\mathrm{d}A}{\mathrm{d}x} = 0$ at the throat, where $M = 1$.

Table 3.1: Boundary conditions for nozzle case

Inlet	Total Pressure	107826.788 Pa
	Pressure	1.0135×10^5 Pa
	Mach number	0.3
	Temperature	288 K
	Enthalpy	3.00114×10^5 J
Outlet	Pressure	$P_{interior}$ (Extrapolated)
	Velocity	$u_{interior}$ (Extrapolated)
	Enthalpy	$H_{interior}$ (Extrapolated)

Table 3.2: Material properties for gas in nozzle case

Viscosity (dynamic)	3.39×10^{-8} Pa s
Specific heat	1006.43 J kg^{-1} K^{-1}
Conductivity (thermal)	0.0257 W m^{-1} K^{-1}
Molecular weight	0.028966 kg mol^{-1}

The Mach number M along the nozzle is related to the cross sectional area A according to

$$\frac{A}{A_{th}} = \left(\frac{\gamma+1}{2}\right)^{-\frac{\gamma+1}{2(\gamma-1)}} \frac{\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma+1}{2(\gamma-1)}}}{M} \tag{3.53}$$

A derivation of this equation can be found in Appendix B.

The computations were performed for a de Laval nozzle whose cross sectional area varies as

$$A_x = A_{th} + (A_i - A_{th})\left(1 - \frac{x}{5}\right)^2 \tag{3.54}$$

where the inlet area $A_i = 2.035$ m^2 and throat area $A_{th} = 1.000$ m^2, and 0.0 m$\leqslant x \leqslant 10.0$ m. This is the same geometry used by Demirdzic *et al.* for validating their code in 1D [80].

A summary of the boundary conditions used in this case is found in Table 3.1. The inlet enthalpy and pressure were prescribed as 3.00114×10^5 J and 1.0135×10^5 Pa respectively. The inlet velocity of Mach 0.3 was prescribed at the inlet from the total pressure computed as 107826.788 Pa using equation (3.16). The material properties for the gas are given in Table 3.2.

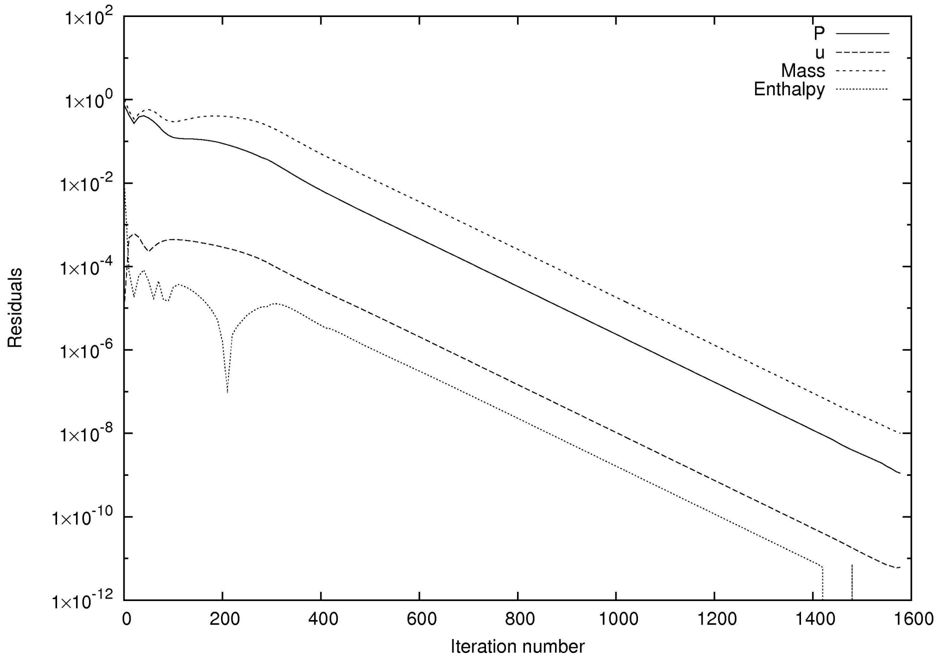

Figure 3.7: Residuals for de Laval nozzle case. Residual calculation as described in Section 1.2.9.

The results of the computations are shown in Figures 3.8, 3.9, 3.12, 3.13 and 3.14. The flow is not completely isentropic as shown in Figures 3.10 and 3.11 since total pressure and total temperature are not constant along the nozzle axis. The solutions for the solved variables do not therefore perfectly match the analytic solution, which has been derived assuming isentropic flow.

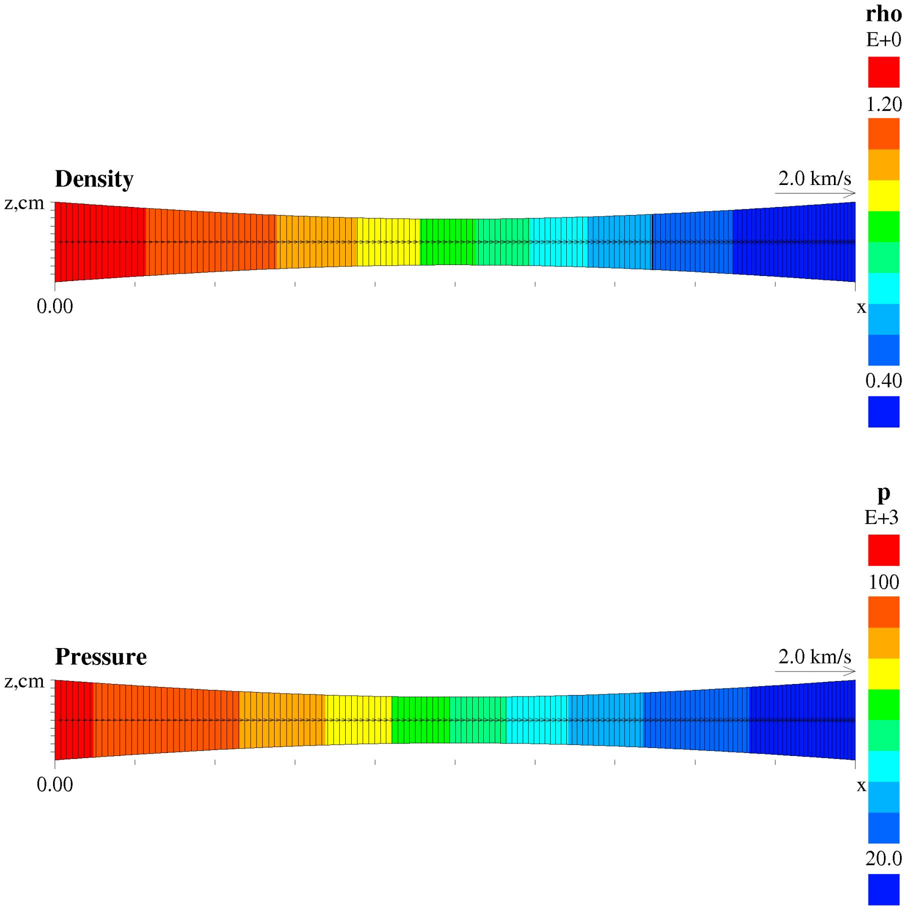

Figure 3.8: Density and Pressure contours for de Laval nozzle case.

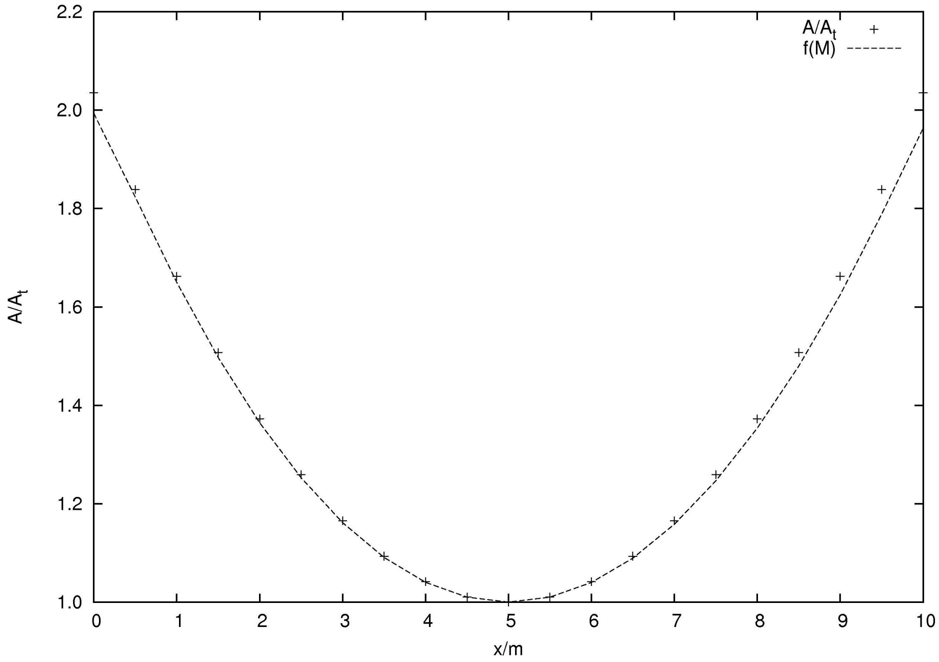

Figure 3.9: Comparison of result with analytic model for de Laval nozzle case.

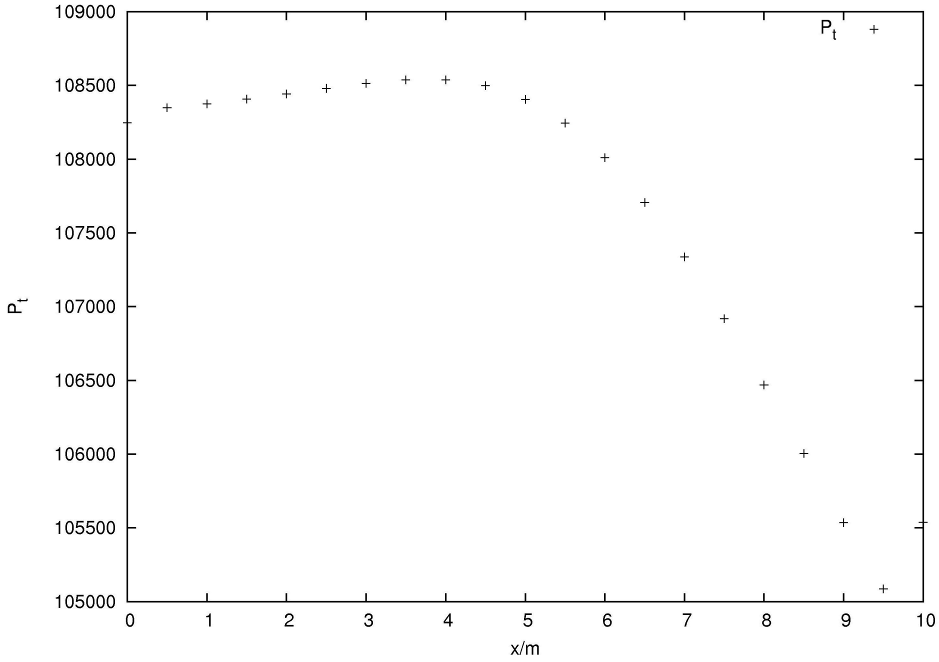

Figure 3.10: Total pressure along de Laval nozzle axis.

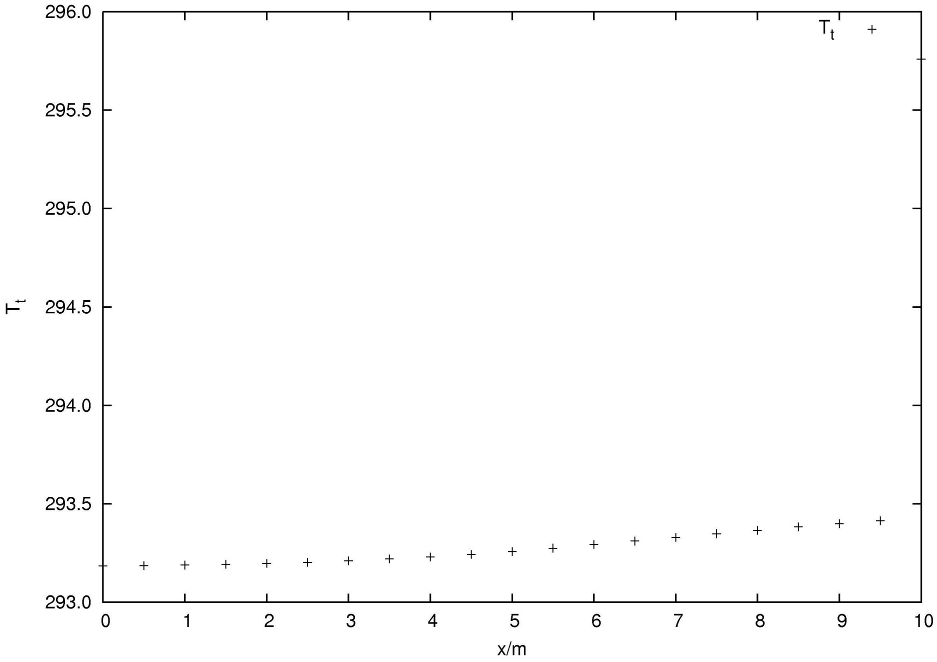

Figure 3.11: Total temperature along de Laval nozzle axis.

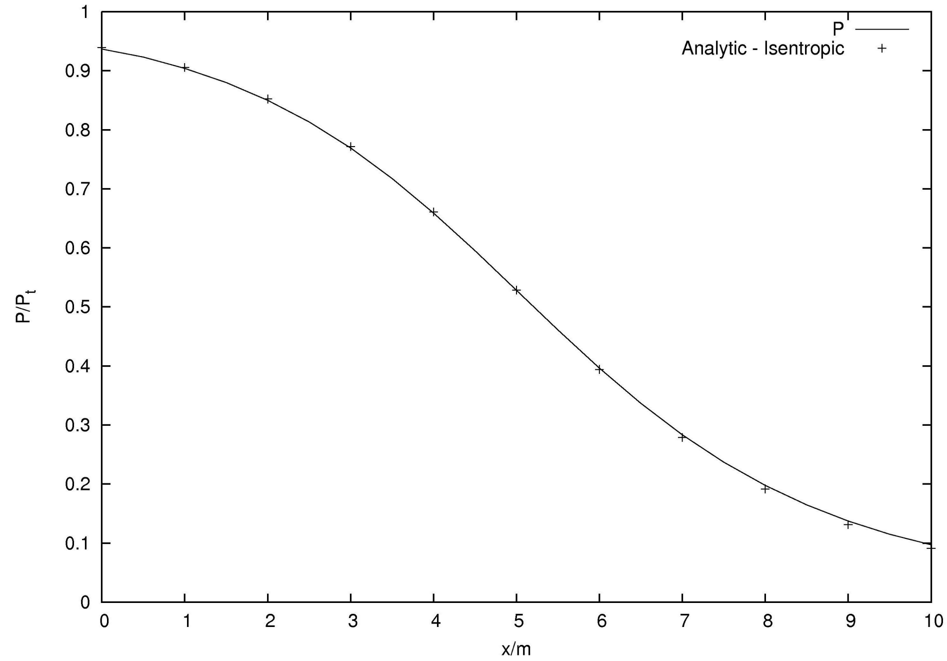

Figure 3.12: Pressure result for de Laval nozzle case.

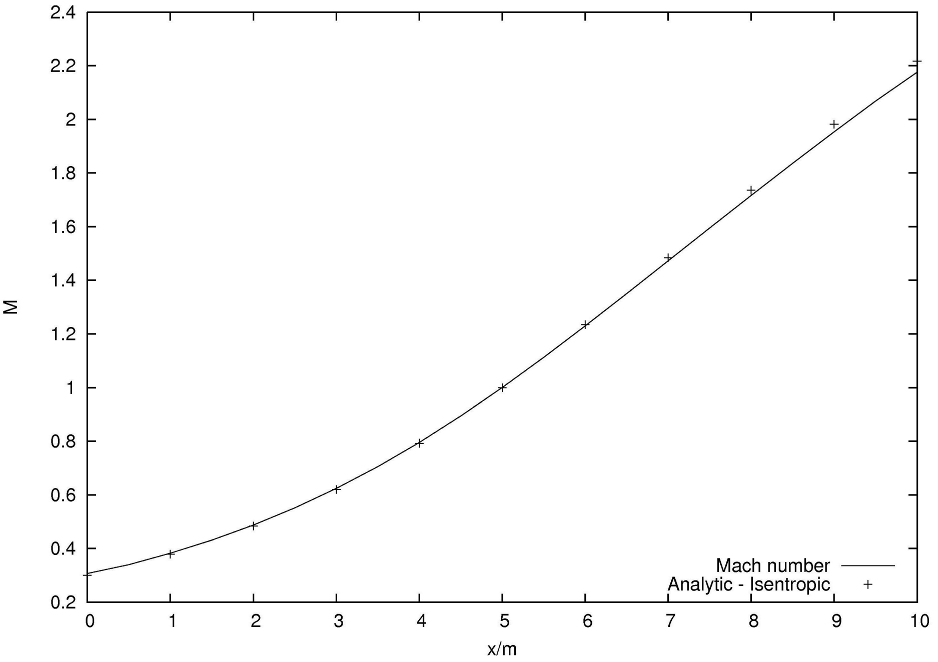

Figure 3.13: Mach result for de Laval nozzle case.

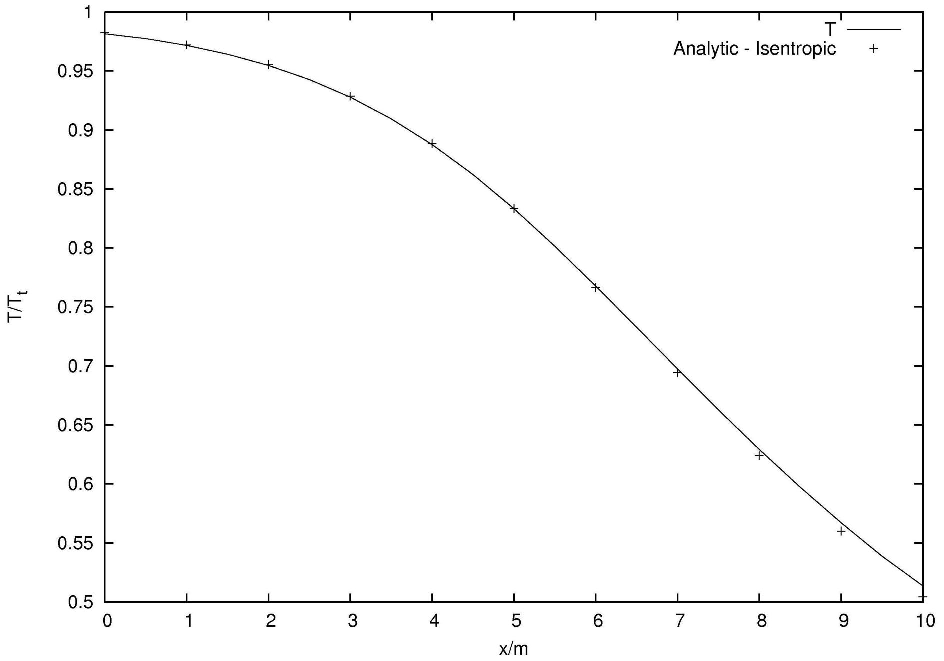

Figure 3.14: Temperature result for de Laval nozzle case.

The speed of sound, i.e. $M = 1.0$ is recovered at the throat, where the flow is choked. The analytic solutions for the other variables are determined from the analytic Mach number. Even though the results are from a non-isentropic flow, the Mach number, temperature and pressure do not deviate significantly from the isentropic case, suggesting that the deviation from non-isentropy does not significantly affect predictions using our numerical method.

The solutions have converged, with the normalised residuals as calculated using the procedure described in Section 1.2.9, going down as shown in Figure 3.7. As expected by the compressible SIMPLEC procedure which relies on mass conservation for stability, the convergence rate is slower at the beginning of simulation. However, as mass conservation is achieved, the solution of the discretised equations speed up.

The implementation of the procedure in Sections 3.2.6 therefore yields accurate results in one dimension, whilst covering the transition range of M = [0.3, 2.0].

3.3.2 Oblique shock

Having established that the suggested numerical method for compressibility works in 1D in the previous section, the next step is to test the code in 2D. The test case chosen is a fully supersonic case whose analytic solution is known: the oblique shock case. A uniform supersonic stream impinges on a wedge: this results in an oblique shock wave separating two regions of uniform flow. A schematic diagram of the case is given below in Figure 3.15 [11].

The solutions for the Mach number M_2, density ρ_2 and pressure p_2 in the shock region, and for the angle of shock δ, should agree with the following relations [11]:

$$\frac{1}{\tan\epsilon} = \left[\left(\frac{\gamma+1}{2}\right)\left(\frac{M_1^2}{M_1^2\sin^2\delta - 1}\right) - 1\right]\tan\delta \tag{3.55}$$

$$\frac{p_2}{p_1} = \left(\frac{2\gamma}{\gamma+1}\right)M_1^2\sin^2\delta - \left(\frac{\gamma-1}{\gamma+1}\right) \tag{3.56}$$

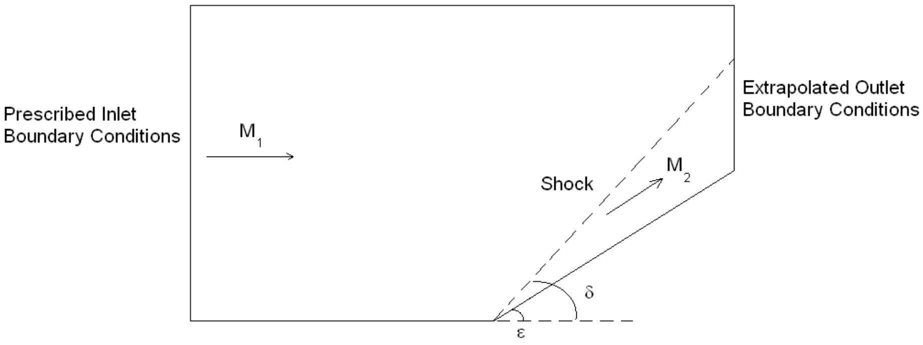

Figure 3.15: Schematic diagram for oblique shock case [11].

Table 3.3: Boundary conditions for oblique shock case

Inlet	Total Pressure	3.7229×10^6 Pa
	Pressure	1.0135×10^5 Pa
	Mach number	3.0
	Enthalpy	815112 J
Outlet	Pressure	$P_{interior}$ (Extrapolated)
	Velocity	$u_{interior}$ (Extrapolated)
	Enthalpy	$H_{interior}$ (Extrapolated)

$$\frac{\rho_2}{\rho_1} = \frac{\tan\delta}{\tan\epsilon} = \frac{(\gamma+1)\,M_1^2\sin^2\delta}{2+(\gamma-1)\,M_1^2\sin^2\delta} \tag{3.57}$$

$$\frac{|\overline{v}_2|}{|\overline{v}_1|} = \frac{\sin\delta}{\sin(\delta-\epsilon)}\left[\frac{2}{(\gamma+1)\,M_1^2\sin^2\delta} + \left(\frac{\gamma-1}{\gamma+1}\right)\right] \tag{3.58}$$

The following test case is computed for a wedge angle of $\epsilon = 15°$ and $\gamma = 1.4$. The inlet pressure for the case is 1.0135×10^5 Pa and the inlet enthalpy is prescribed as 815112 J. An inlet velocity of Mach 3.0 is prescribed by fixing the total pressure at the inlet as 3.7229×10^6 Pa. The boundary conditions for this computation is summarised in Table 3.3 and the material properties for the gas modelled are given in Table 3.4.

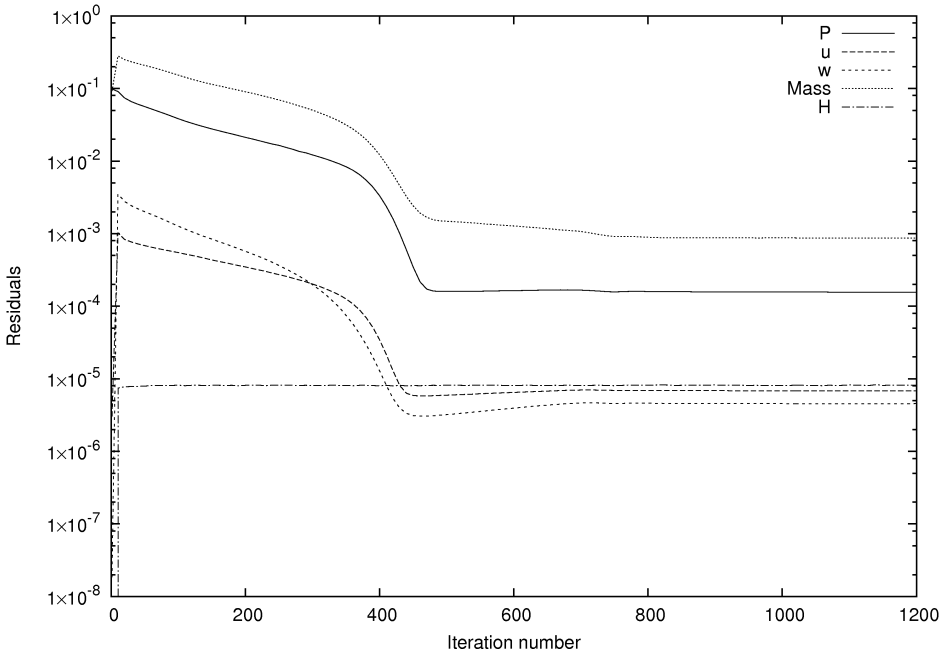

Figure 3.16: Residuals for the oblique shock case case. Residual calculation as described in Section 1.2.9.

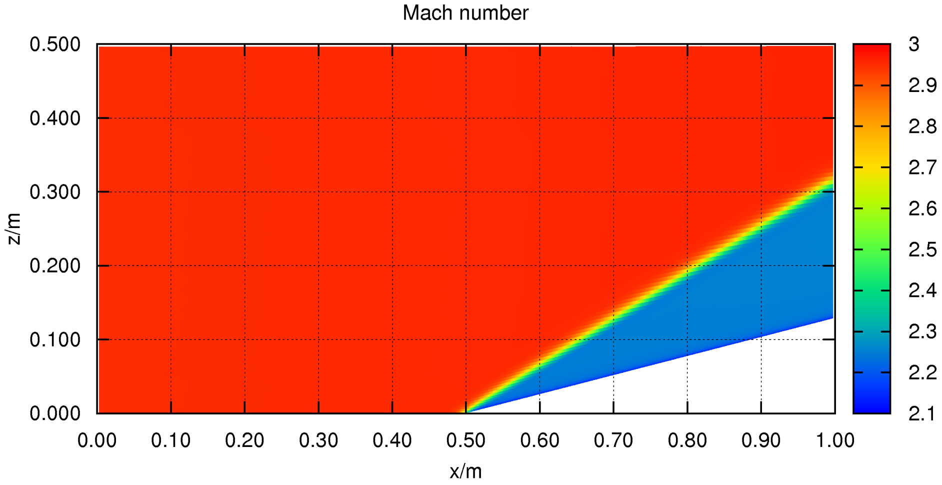

Figure 3.17: Mach number contour for the oblique shock case.

Table 3.4: Material properties for gas in oblique shock case

Viscosity (kinematic)	3.39×10^{-5} m^2 s^{-1}
Specific heat	1024 J kg^{-1} K^{-1}
Conductivity (thermal)	0.038 W m^{-1} K^{-1}
Molecular weight	0.028966 kg mol^{-1}

Table 3.5: Results for oblique case and comparison with analytic solution

	M	p	ρ	v
Region 1	3.00	101350	1.226	1020
Region 2	2.29	276517	2.452	909
Region 1 (deformed)	3.00	101403	1.228	1020
Region 2 (deformed)	2.28	275419	2.456	911
Ratio $\frac{Region2}{Region1}$	0.763	2.728	2.000	0.891
Ratio (deformed)	0.760	2.716	2.001	0.893
Analytic Ratio	0.750	2.787	2.034	0.888

The residuals and contour results for the case are illustrated in Figures 3.16 and 3.17. After convergence, the computed shock angle is $\delta = 31.34°$ while the analytic solution is $\delta = 32.2°$. The following table illustrates the results for the solved variables and comparison with the analytic solution. The computed results compare well with the theoretical solution, hence establishing that the code works for supersonic flow in two dimensions.

3.3.2.1 Randomly deformed mesh

The oblique case was re-run on a randomly deformed mesh using the same boundary conditions as in the previous case, cf Table 3.3. A diffusion correction as described in Section 2.3.2.4 is used to get a converged result. The mesh for this particular case

Table 3.6: Material properties for gas in the channel bump case

Viscosity (kinematic)	3.39×10^{-5} m^2 s^{-1}
Specific heat	1024 J kg^{-1} K^{-1}
Conductivity (thermal)	0.038 W m^{-1} K^{-1}
Molecular weight	0.028966 kg mol^{-1}

is shown in Figure 3.18. The same analytic solution is recovered again, as shown in Figure 3.21. This suggests that the numerical method described in this chapter works in compressible cases with a non-orthogonal mesh. The computed shock angle in this highly non-orthogonal case, $\delta = 32.1°$, is close to the analytic solution, $\delta = 32.2°$. No pressure interpolation is required to achieve convergence in this case.

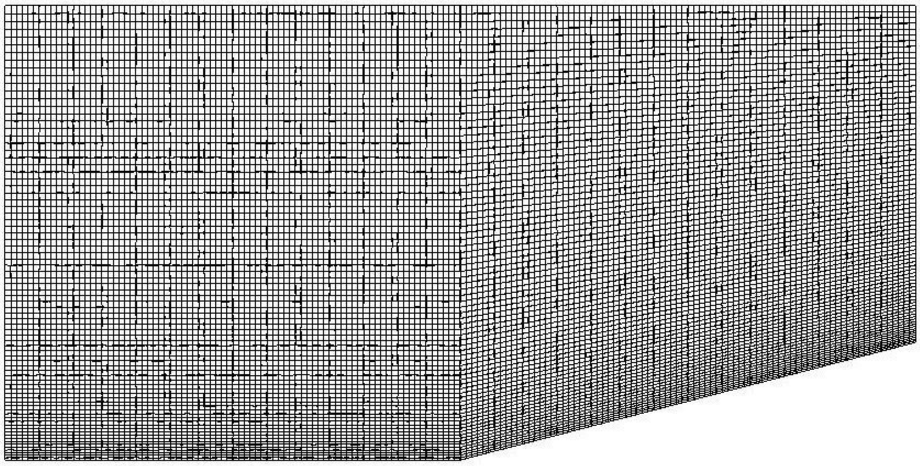

Figure 3.18: Randomly deformed mesh for oblique shock case.

3.3.3 Channel bump

This section presents the results when the numerical method developed in this thesis is tested against another standard test, one proposed at the GAMM conference [97]: the flow in a channel over a circular bump. The test case covers a large range of Mach numbers in 2D, from subsonic flow, through transition and up to supersonic flows. In

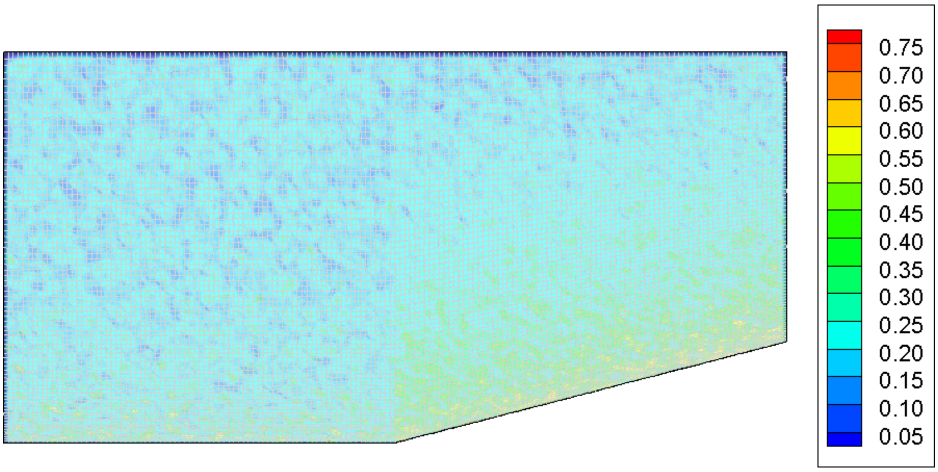

Figure 3.19: Mesh quality indicator for randomly deformed mesh used for oblique shock case. Mesh indicator from Section 2.1.3

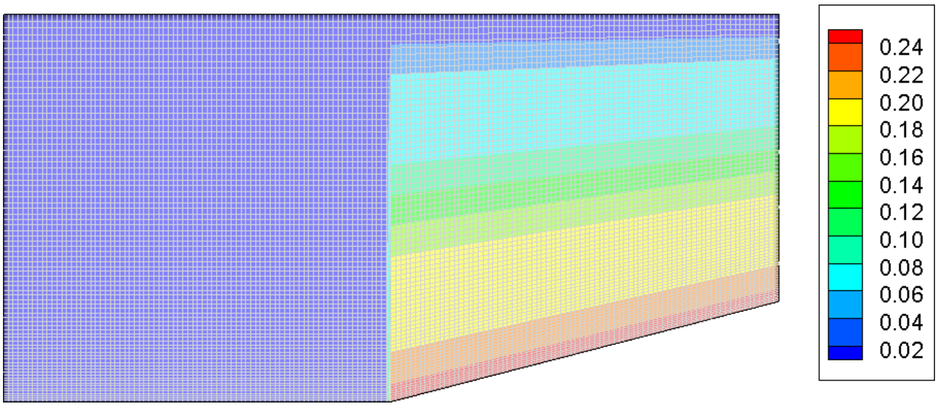

Figure 3.20: Mesh quality indicator for mesh used for oblique shock case. Mesh indicator from Section 2.1.3

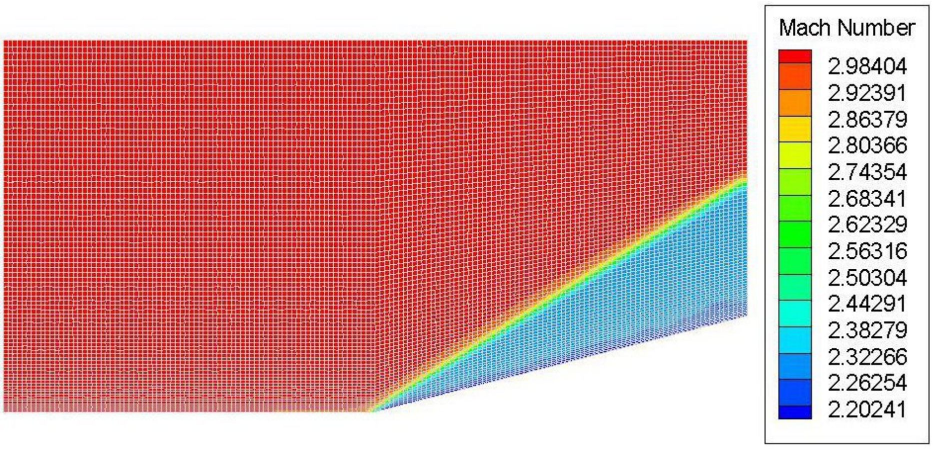

Figure 3.21: Mach number contour for randomly deformed mesh oblique shock case.

this particular section, the higher-order resolution scheme to compute face densities proposed in Section 3.2.4.1 is also tested to investigate the impact of face interpolation in the shock-capturing property of our numerical method.

The channel height is equal to the length of the bump and the channel length is three times the length of the channel bump. The results are compared with that of other authors [12; 13]. The material properties of the gas used in this section is given in Table 3.6.

3.3.3.1 Subsonic flow

For subsonic flow, the thickness-to-chord ratio of the bump is 10%. The mesh orthogonality for this geometry is given in Figure 3.22, with non-orthogonality being significant near the circular bump. The boundary conditions for this case are given in Table 3.7.

The mesh independence analysis study is shown in Figures 3.23 and 3.24. The cases are run using three different approaches:

1. using an incompressible finite volume code, with the HYBRID difference scheme for all variables,

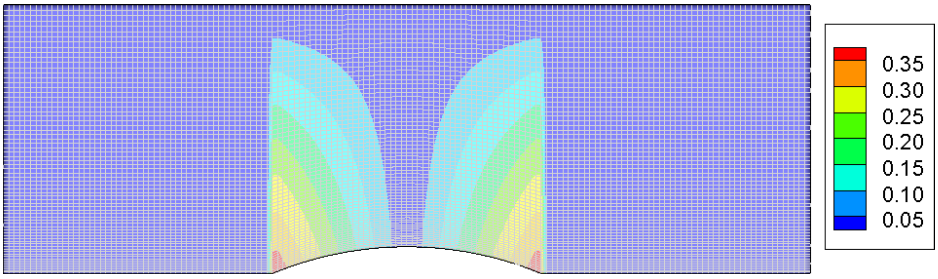

Figure 3.22: Mesh quality indicator for mesh used in subsonic channel case. Mesh indicator from Section 2.1.3

Table 3.7: Boundary conditions for subsonic bump case

Inlet	Total Pressure	1.02010×10^5 Pa
	Pressure	Not prescribed
	Mach number	0.10
	Enthalpy	2.9550×10^5 J
Outlet	Pressure	1.0135×10^5 Pa
	Velocity	$u_{interior}$ (Extrapolated)
	Enthalpy	$H_{interior}$ (Extrapolated)

2. using a compressible code with the modifications to the pressure correction equation as detailed in Section 3.2.4, with a false time step term from Section 3.2.5 and the HYBRID scheme for all solved variables, and

3. same as the above, but with the mass imbalance calculated using a higher order scheme as described in Section 3.2.4.1.

The results are not significantly different for the case when using 40×90 and 70×150 meshes. The same profile is recovered whether using a compressible code with a low order face density calculation scheme, a compressible code with a higher order scheme as described in Section 3.2.4.1, or simply using an incompressible code.

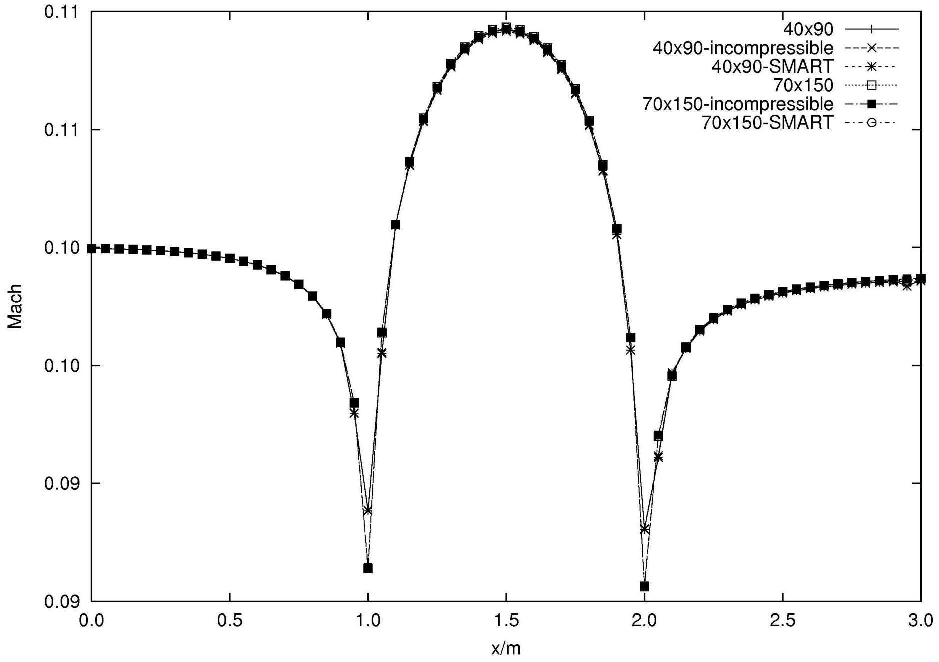

Figure 3.23: Mach number along the bottom of a channel with circular bump - subsonic case. Inlet Mach number = 0.10. Mesh independence analysis.

The Mach profile as shown in Figure 3.25 is almost symmetrical, as expected for this low speed case. A 'shock' is obtained at the outlet since the whole face is prescribed to be at atmospheric pressure.

With a maximum non-orthogonality factor of 0.35 as calculated using the indicator from Section 2.1.3, the non-orthogonality correction from Section 2.3.2.3 is used for diffusion terms. This may have been redundant if the mesh density is coarse, as shown in the Peclet contour plot.

Another subsonic case is run with Mach 0.5 at the inlet, and the total variables adjusted for the new inlet velocity. A benchmark [37] - which assumes an inviscid compressible fluid in its calculations - is available for comparison in this case. The results are shown in Figures 3.26 and 3.27. The results agree well with the benchmark from Ferziger and Perić [37], even though the case run is not inviscid. The flow after the bump is significantly different, but the difference from the benchmark case decreases

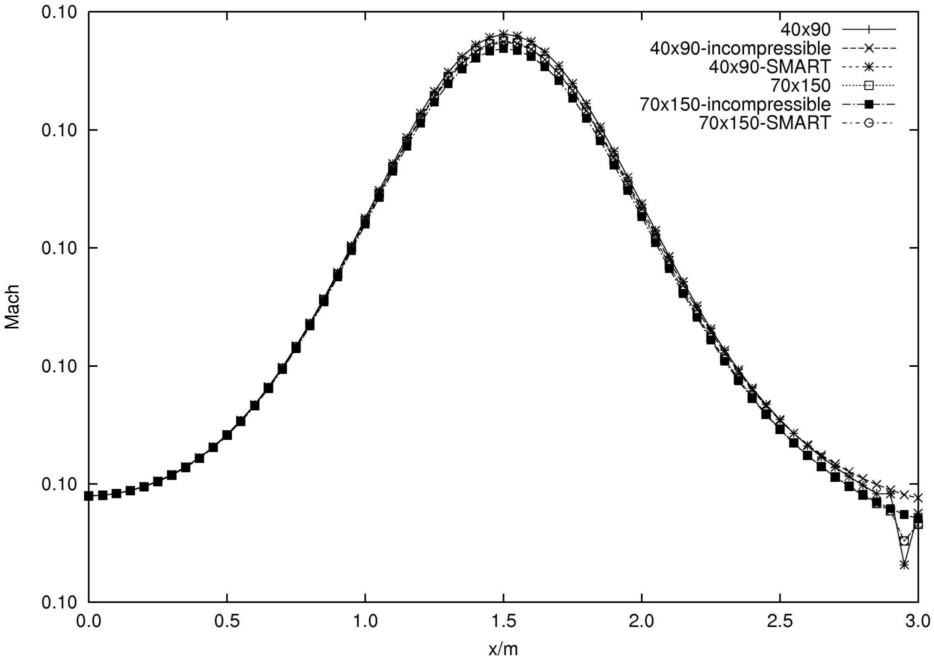

Figure 3.24: Mach number along the upper of a channel with circular bump - subsonic case. Inlet Mach number = 0.10. Mesh independence analysis.

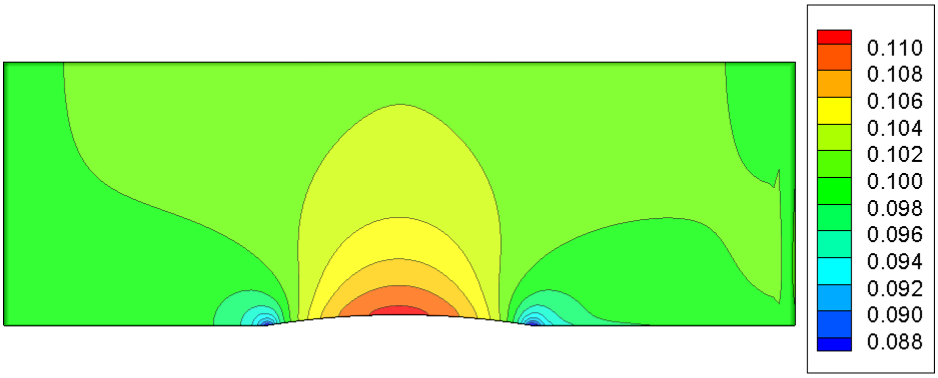

Figure 3.25: Mach contour for channel with bump for subsonic case. Inlet Mach number = 0.10

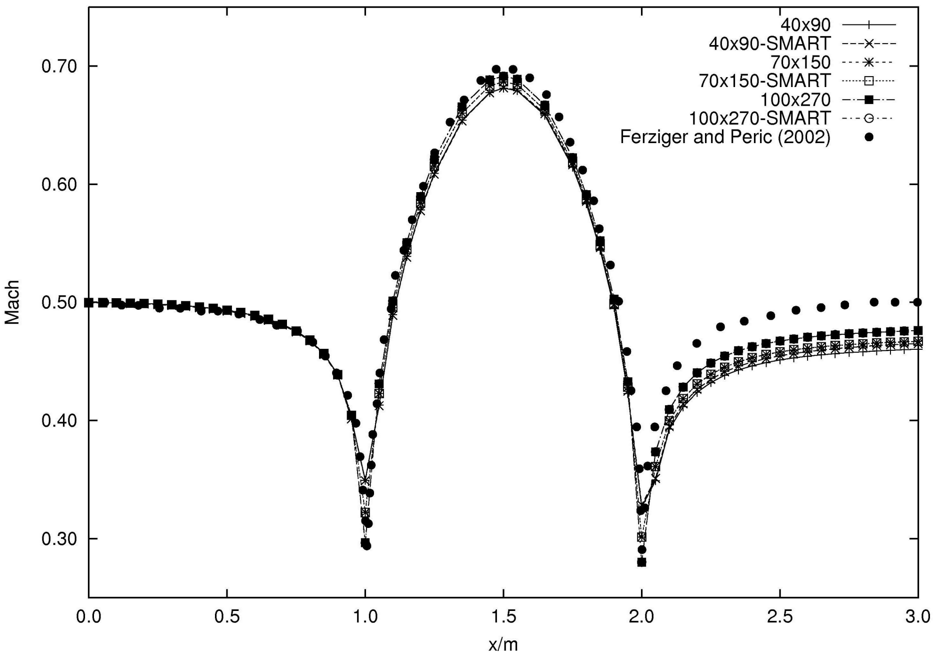

Figure 3.26: Mach number along the bottom of a channel with circular bump - subsonic case. Inlet Mach number = 0.50. Mesh independence analysis and comparison with benchmark from Ferziger and Perić [37].

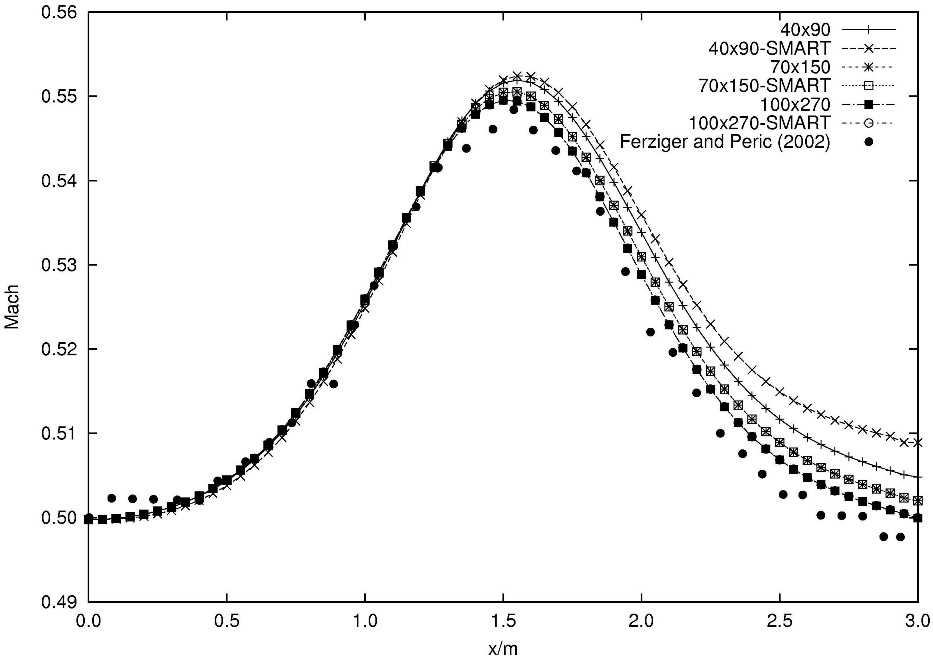

Figure 3.27: Mach number along the upper of a channel with circular bump - subsonic case. Inlet Mach number = 0.50. Mesh independence analysis and comparison with benchmark from Ferziger and Perić [37].

as the mesh is refined. Numerical energy dissipation is present on low density grids.

3.3.3.2 Transonic flow

The transition case is tested in this section. The same mesh as for subsonic tests is used to compute flow predictions, with only inlet conditions changed to match an inlet Mach number of 0.675. The expected Mach number profile is given by Figure 3.28, with a shock expected on the downstream half of the bump [37].

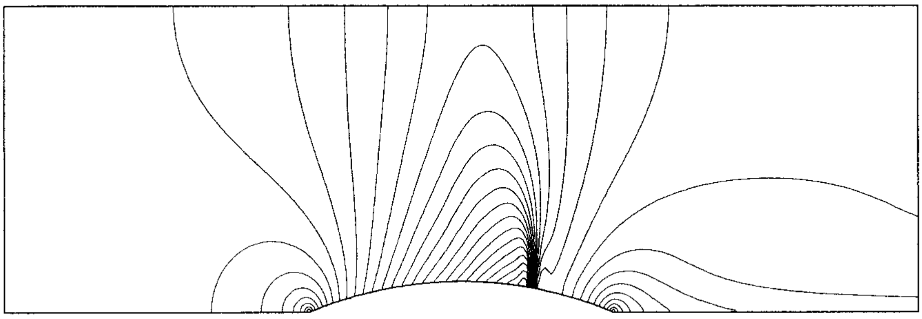

Figure 3.28: Mach contour lines for channel with bump for transonic case. Inlet Mach number = 0.675. Taken from Ferziger and Perić [37].

The numerical method proposed in this research fails to predict the profile predicted by the inviscid code of Ferziger and Perić [37], with the wrong profile given by Figure 3.29 predicted by the current method.

3.3.3.3 Supersonic flow

For supersonic flow, the thickness-to-chord ratio of the bump is 4%. The boundary conditions for this case are given in Table 3.8. Since all variables are prescribed at the inlet, the pressure correction at the inlet is fixed to zero. The orthogonality indicator plot for this case is shown in Figure 3.30.

The mesh independence analysis study is shown in Figures 3.24 and 3.24. The shocks are rendered in sharper fashion as the grid density is increased. The cases are run using two different approaches:

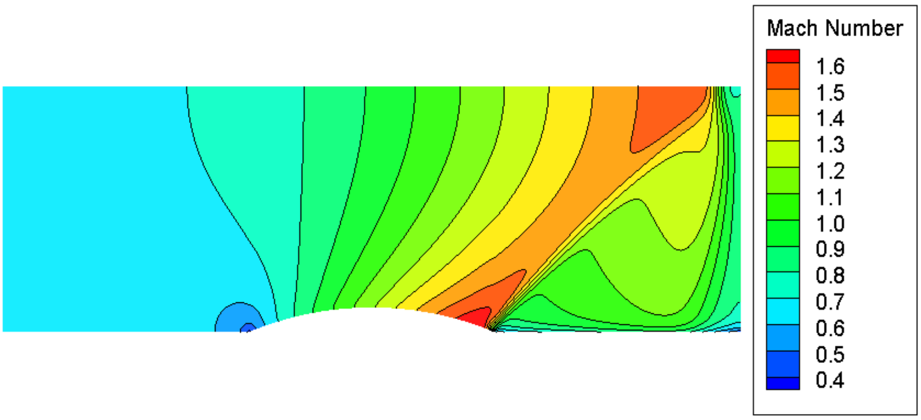

Figure 3.29: Mach contour for channel with bump for transonic case. Inlet Mach number = 0.675.

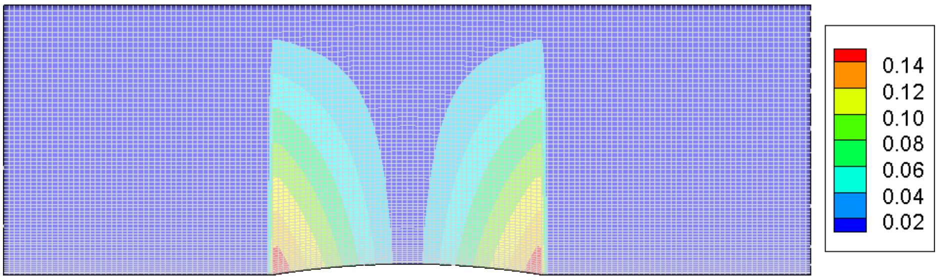

Figure 3.30: Mesh quality indicator for mesh used in supersonic channel case. Mesh indicator from Section 2.1.3

Table 3.8: Boundary conditions for supersonic bump case

Inlet	Total Pressure	4.6384×10^5 Pa
	Pressure	1.0135×10^5 Pa
	Mach number	1.65
	Enthalpy	3.6801×10^5 J
Outlet	Pressure	$P_{interior}$ (Extrapolated)
	Velocity	$u_{interior}$ (Extrapolated)
	Enthalpy	$H_{interior}$ (Extrapolated)

1. using a compressible code with the modifications to the pressure correction equation as detailed in Section 3.2.4, with a false time step term from Section 3.2.5 and the hybrid scheme for all solved variables, and
2. same as the above, but with the mass imbalance calculated using a higher order scheme as described in Section 3.2.4.1.

The supersonic case shows slow mesh convergence rate, and requires a large number of cells to achieve mesh independence after the second shock.

The Mach number profile along the lower and upper walls for the channel for the supersonic case are shown in Figures 3.33 and 3.34. The numerical method used in this study produces results with agree generally well with the codes from Date [12] and Moukalled and Darwish [13].

The contour is shown in Figure 3.35. Shocks are reflected on the upper wall, with numerical diffusion of the reflected shock. The face density values used to calculate mass balance terms in the pressure correction equation were computed using SMART [28] - it was thought that using a higher order scheme to obtain face values would help to attenuate the effect of diffusion, but without significant effect.

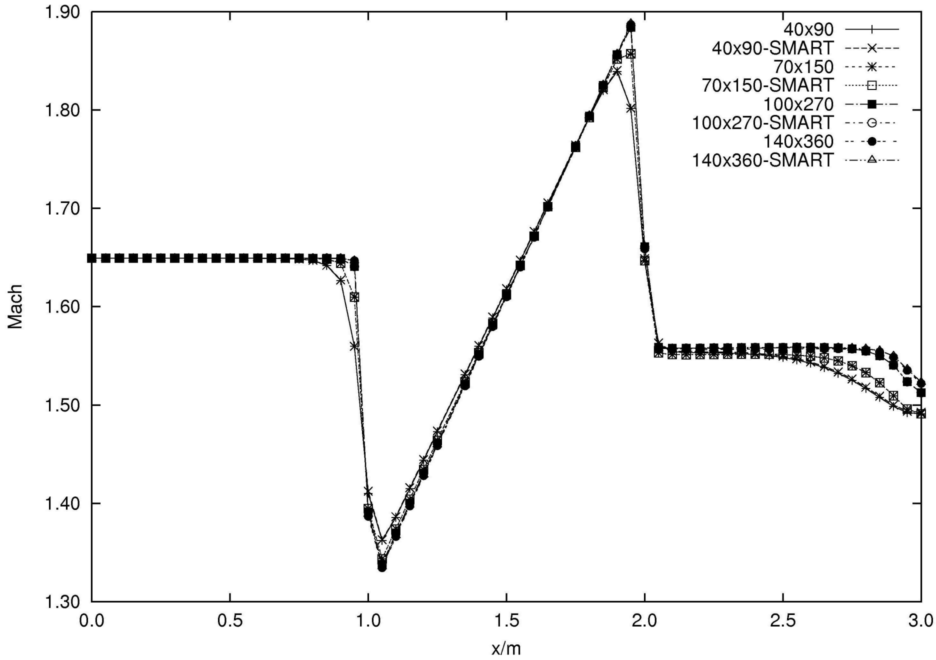

Figure 3.31: Mach number along the bottom of a channel with circular bump - supersonic case. Inlet Mach number = 1.65. Mesh independence analysis.

Supersonic case with Mach 2.0 Another supersonic case is run with the same geometry and code implementation, but with a different inlet Mach number of 2.0, in order to test the applicability of the code around Mach 2.0. Since no benchmark is available for this case, convergence is assumed if mesh independence of results is obtained. The results for this case are shown in Figures 3.36, 3.37 and 3.38. As in the Mach 1.65 case, the shocks become sharper as the mesh is refined - which is analogous to using a higher-order differencing scheme to solve conserved variables. No issue arises when running the case with a higher inlet Mach number of 2.0, suggesting that the code implementation is well behaved for supersonic flows.

3.3.3.4 Channel bump results overview

The numerical method developed for compressible flows within a collocated, pressure-correction type solver works for the subsonic and supersonic regions, including Mach 2.0. However, further work is required to address transonic flow in two dimensions. Suggestions as to how to address this are given in Chapter 6.

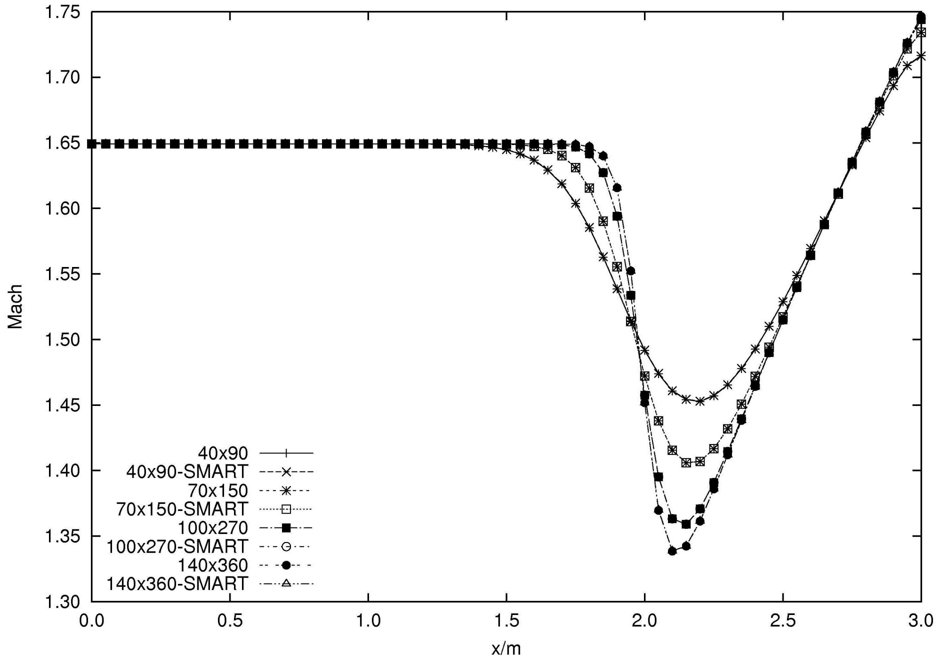

Figure 3.32: Mach number along the upper of a channel with circular bump - supersonic case. Inlet Mach number = 1.65. Mesh independence analysis.

3.3.4 Oxygen jet

The behaviour of the oxygen jet is crucial factor in steel-making processes. Oxygen jets are under current research [5; 14; 86] since their behaviour is not fully understood. Of particular interest is the behaviour of an oxygen jet in a high temperature field. Sumi *et al.* [14] conducted an empirical study of oxygen jets in a pre-heated furnace at different ambient temperature: 285 K, 772 K and 1002 K. They provide the empirical data against which the following results will be validated.

The setup used to model the experimental configuration is shown in Figure 3.39. Three mesh densities were used to verify mesh independence: 6188, 13830 and 24752 cells. Oxygen enters the domain through the nozzle exit at Mach 1.72 and a temperature of 190 K. The inlet velocity is prescribed. The walls are closed at the top and right of the domain, and symmetry boundary conditions are imposed at the axis. The pressure is extrapolated at the outlet. The temperature of the walls is fixed to the am-

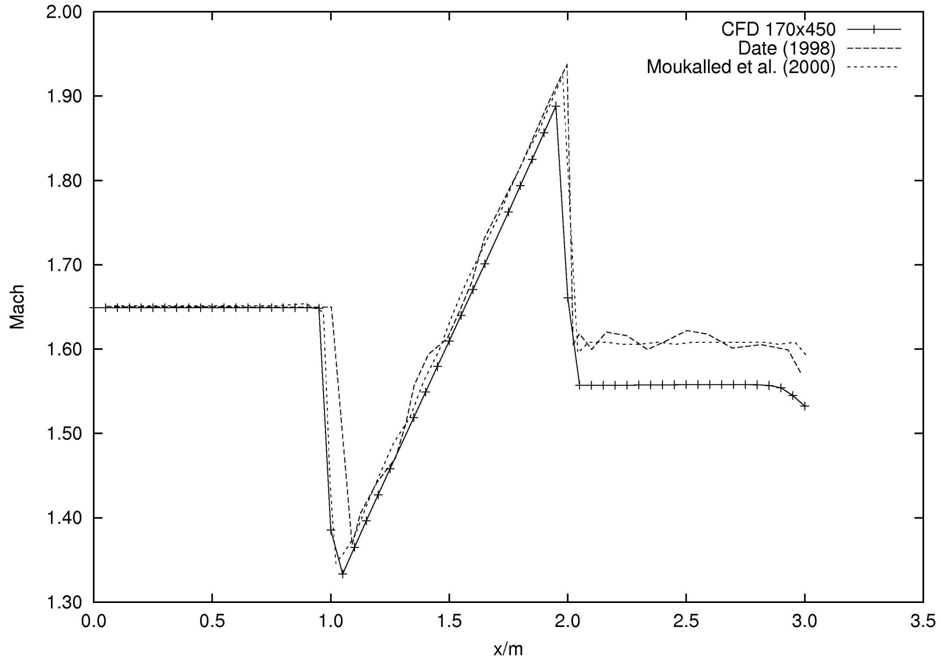

Figure 3.33: Mach number along bottom wall of channel with circular bump for supersonic case and comparison with predictions from [12] and [13]. Inlet Mach number = 1.65.

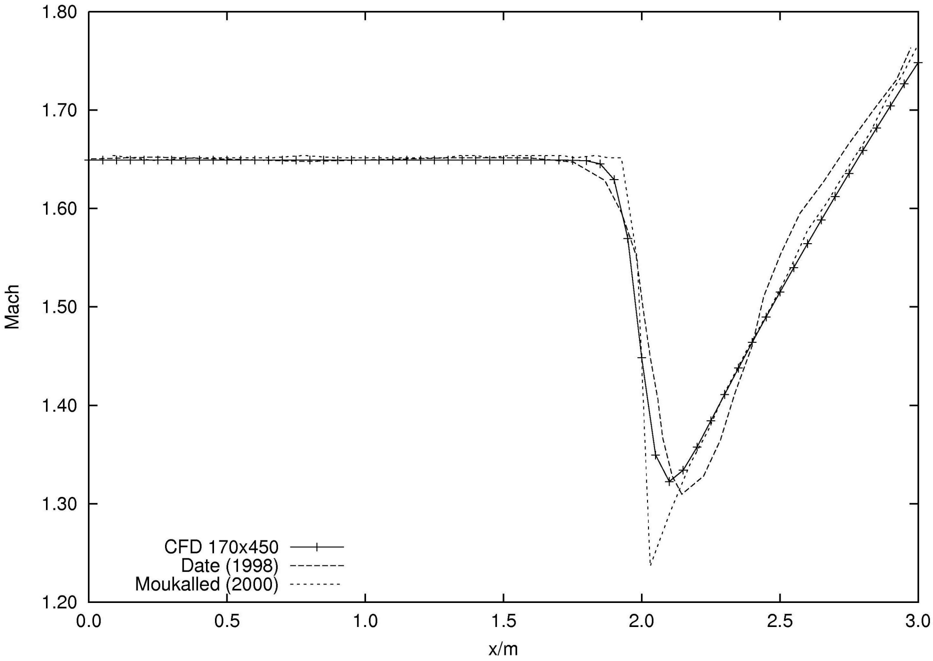

Figure 3.34: Mach number along upper wall of channel with circular bump for supersonic case and comparison with predictions from [12] and [13]. Inlet Mach number = 1.65.

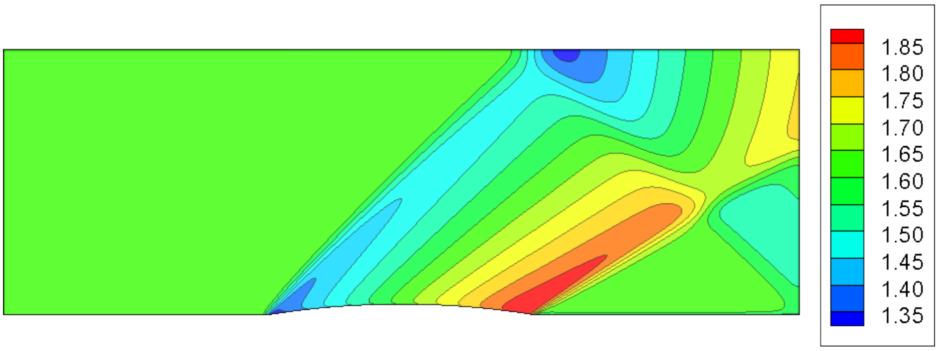

Figure 3.35: Mach contour for channel with bump for supersonic case. Inlet Mach number = 1.65.

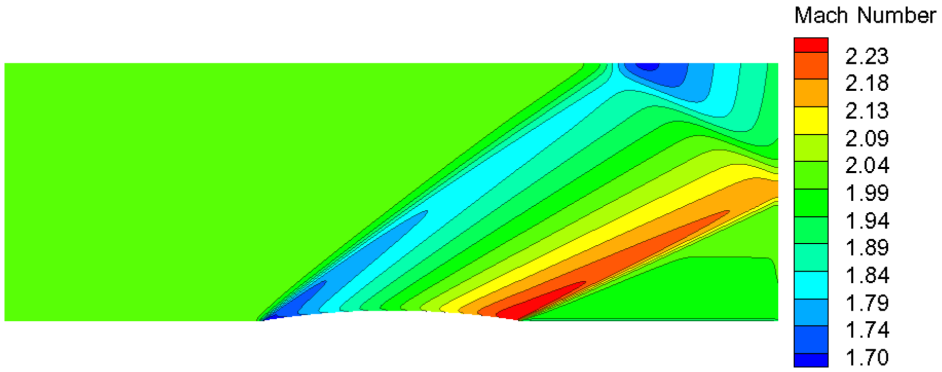

Figure 3.36: Mach contour for channel with bump for supersonic case. Inlet Mach number = 2.00.

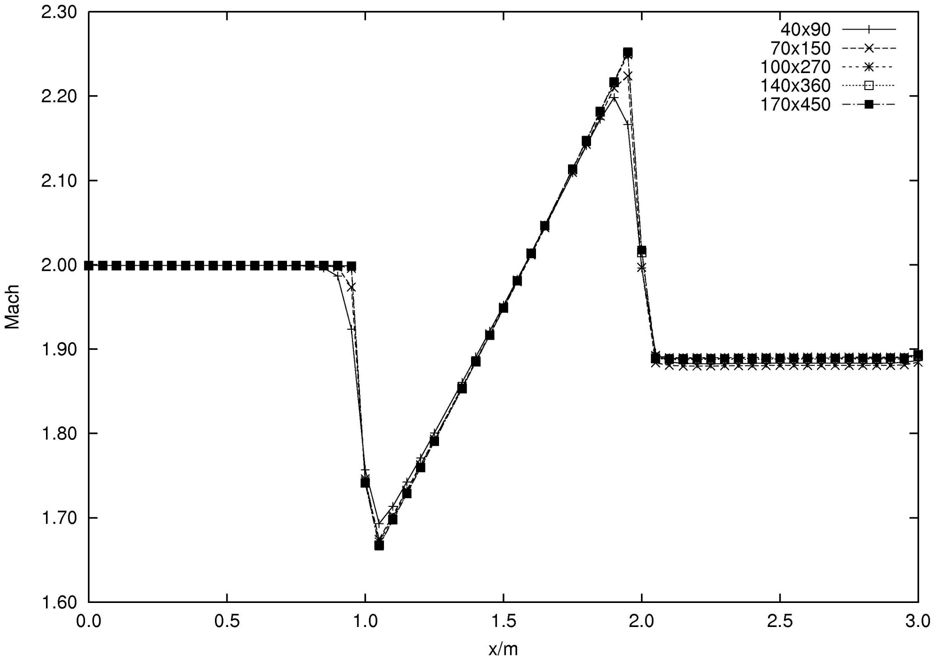

Figure 3.37: Mach number along the bottom of a channel with circular bump - supersonic case. Inlet Mach number = 2.00. Mesh independence analysis.

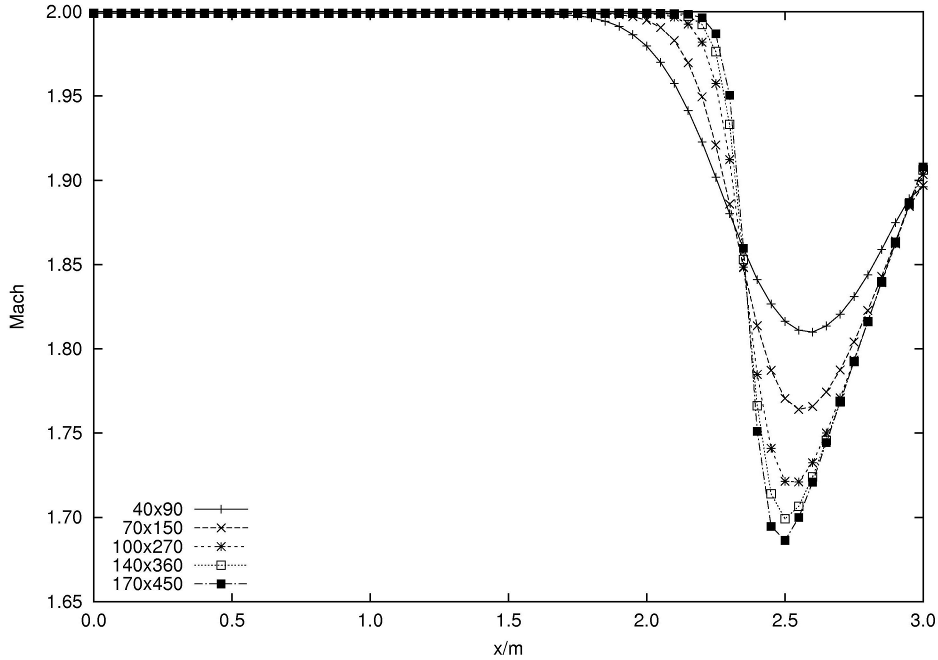

Figure 3.38: Mach number along the upper of a channel with circular bump - supersonic case. Inlet Mach number = 2.00. Mesh independence analysis.

bient temperatures of 285 K, 772 K and 1002 K respectively to match the experimental conditions. These temperatures are enforced at the outlets by fixing the recalculating the enthalpy at the outlet after each iteration to match these temperatures.

Turbulence is modelled using Alam's modifications to the $k - \varepsilon$ model [5], with a turbulent Prandtl number of 0.5. The SMART [28] differencing scheme is used to discretise the momentum equations, while upwinding is used for all other equations. The modified pressure correction procedure described in Section 3.2.4 is used to impose mass continuity.

When modelling turbulence, it is required to know a measure of the turbulent intensity, or turbulence level, I at the inlet, to be fed as boundary condition when solving for both k and ε.

$$I = \frac{u'}{U} \tag{3.59}$$

Table 3.9: Boundary conditions for oxygen jet case

Inlet	Total Pressure	4.97695×10^5 Pa
	Pressure	1.0135×10^5 Pa
	Mach number	1.72
	Temperature	190 K
	Turbulent intensity	2%
Outlet	Pressure	$P_{interior}$ (Extrapolated)
	Velocity	$u_{interior}$ (Extrapolated)
	Temperature	285 K, 772 K, 1002 K

Table 3.10: Material properties for oxygen jet case

Viscosity (dynamic)	1.4×10^{-5} Pa s
Specific heat	910 J kg^{-1} K^{-1}
Conductivity (thermal)	0.0238 W m^{-1} K^{-1}
Molecular weight	0.032 kg mol^{-1}

where u' is the root-mean-square value of the velocity fluctuations, and U is the Reynolds-averaged mean velocity. An empirical model used for fully developed pipe flow is [98]

$$I = 0.16\mathrm{Re}^{1/8} \tag{3.60}$$

where Re is the Reynolds number of the flow, given by

$$\mathrm{Re} = \frac{\rho U L}{\mu} \tag{3.61}$$

where L is a characteristic length of the problem.

The boundary conditions for such an inlet are then taken as

$$k = \frac{3}{2}\left(IU\right)^2 \tag{3.62}$$

$$\varepsilon = C_\mu^{3/4}\frac{k^{3/2}}{\ell} \tag{3.63}$$

The mixing length ℓ depends on the flow problem being investigated. For an axisymmetric jet, the mixing length is taken as

$$\ell = 0.09D \tag{3.64}$$

where D is the jet half width [25].

The development of a particular jet, with an ambient temperature of 1002 K, is shown in the plates in Figure 3.40 below. The jet takes about 15 ms to develop, after which the jet profile can be treated as steady state.

The results are shown in Figures 3.41 and 3.42 below. These axial velocities and temperatures are time averaged, after 15 ms - time after which the jet can be treated as steady state. The model slightly under-predicts the core length of the jet in each case. The axial temperature profile is also under-predicted. The results are mesh independent for T = 285 K as shown in Figure 3.43, but not at high temperatures as shown in Figures 3.44 and 3.45.

3.3.4.1 Development of a new turbulence model for the cold jet case using the HYBRID difference scheme

SMART does not show good mesh convergence behaviour at high temperatures as shown in Figures 3.44 and 3.45. A new turbulence model based on ambient temperatures and modification of C_μ, specifically designed for HYBRID, is proposed in this section. Following the modifications to the turbulence in the literature, C_μ is freely modified since no empirical value for compressible, high speed flows is available in the literature, and the $k-\varepsilon$ model has been devised for high Reynold number flows, but at low Mach numbers. Using HYBRID also results in faster convergence than when using SMART, and mesh refinement is similar to increasing the order of discretisation

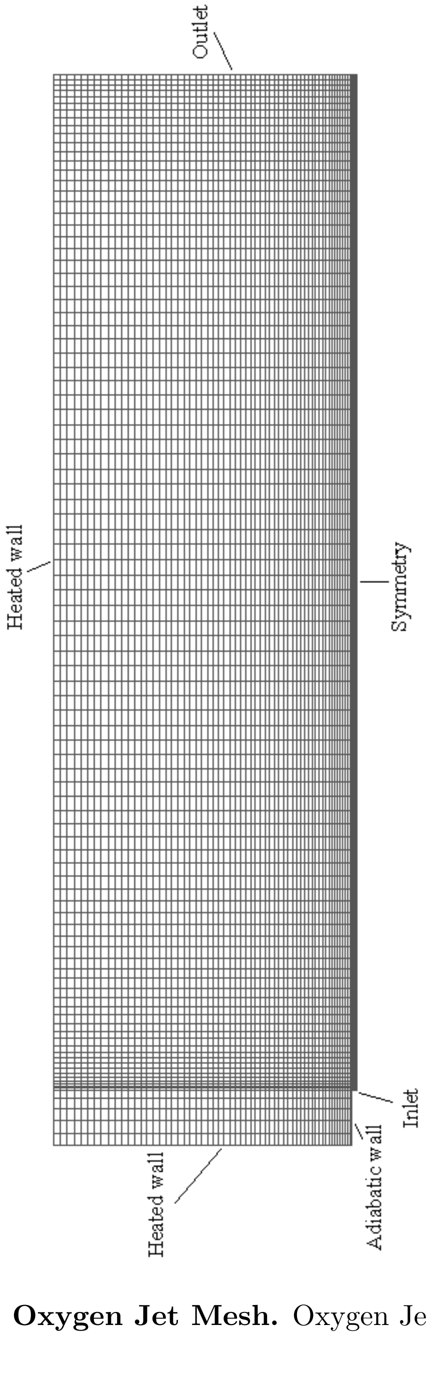

Figure 3.39: Oxygen Jet Mesh. Oxygen Jet Mesh - 6188

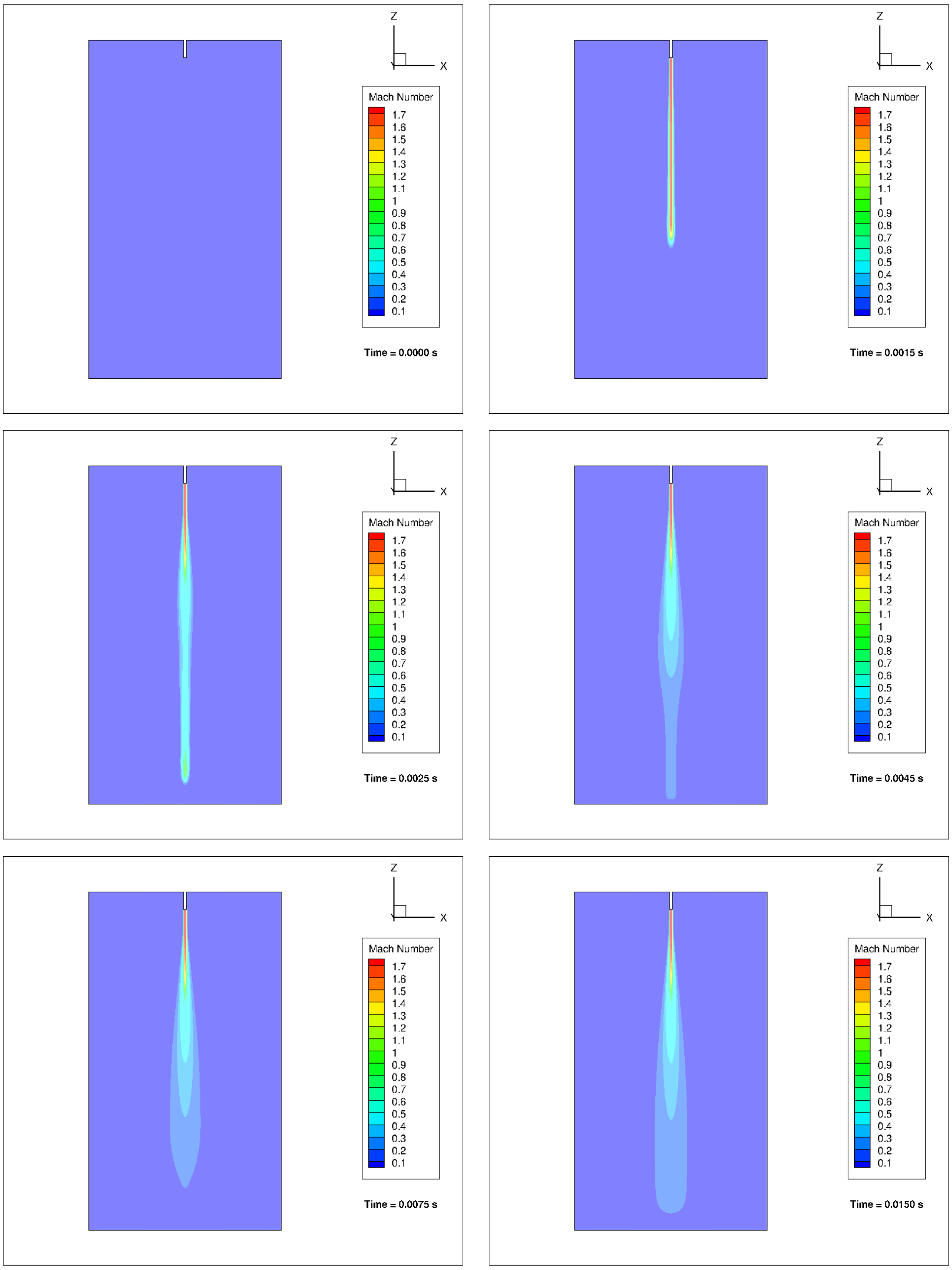

Figure 3.40: Oxygen Jet. Transient

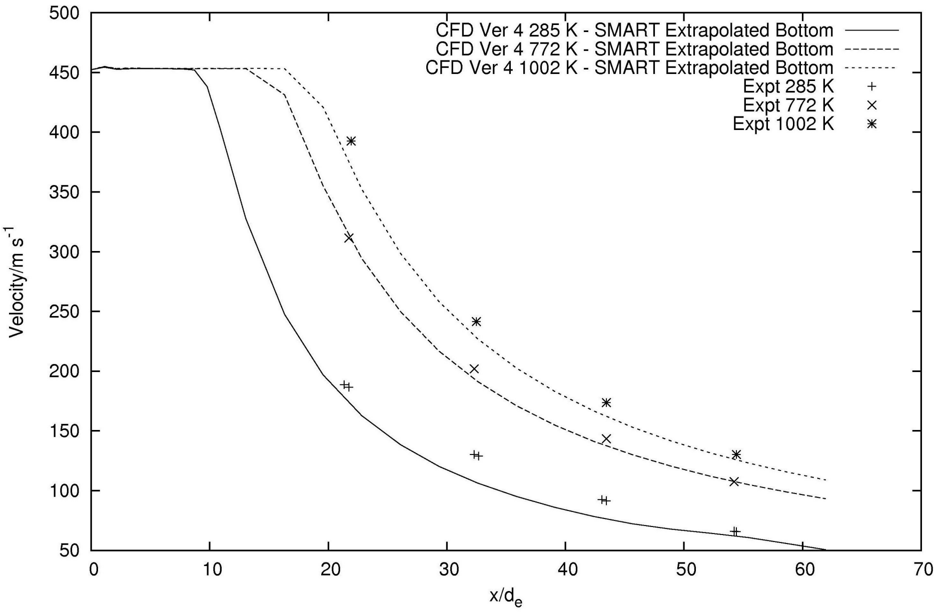

Figure 3.41: Comparison of axial velocity with experimental data from Sumi *et al.* [14]. Mesh - 24752

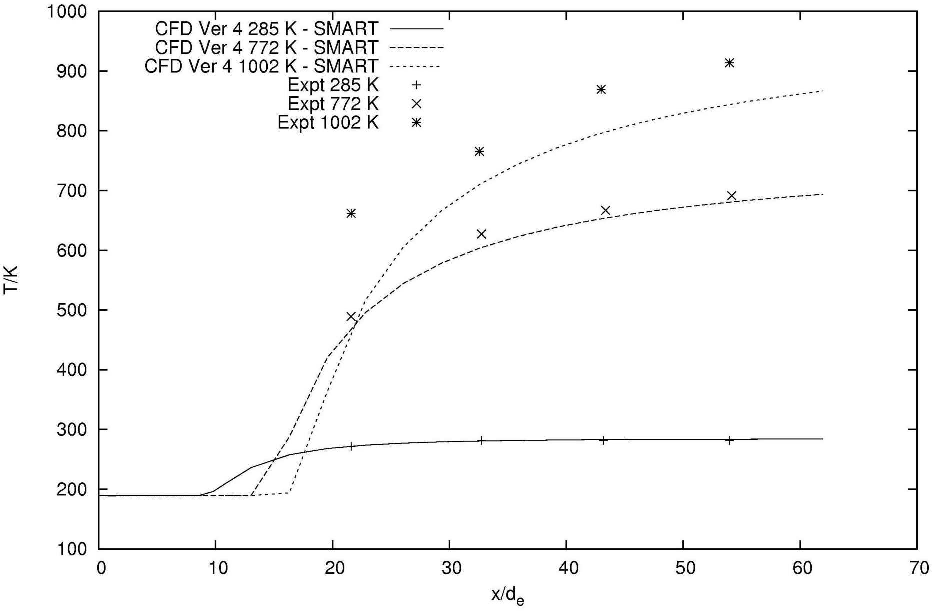

Figure 3.42: Comparison of axial temperature with experimental data from Sumi *et al.* [14]. Mesh - 24752

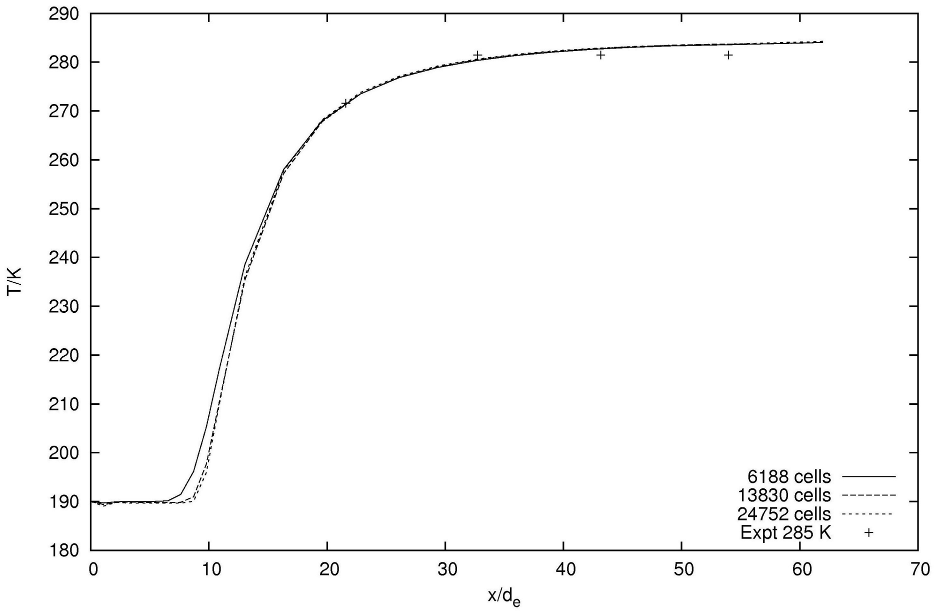

Figure 3.43: Mesh independence analysis for ambient temperature = 285 K. SMART for momentum discretisation, turbulence modification using [5] and $\sigma_T = 0.5$

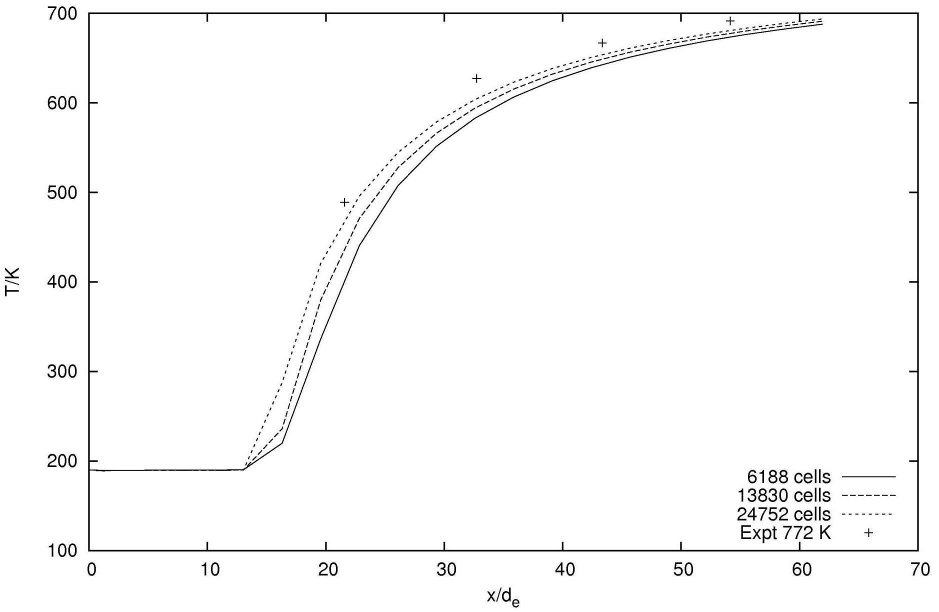

Figure 3.44: Mesh independence analysis for ambient temperature = 772 K. SMART for momentum discretisation, turbulence modification using [5] and $\sigma_T = 0.5$

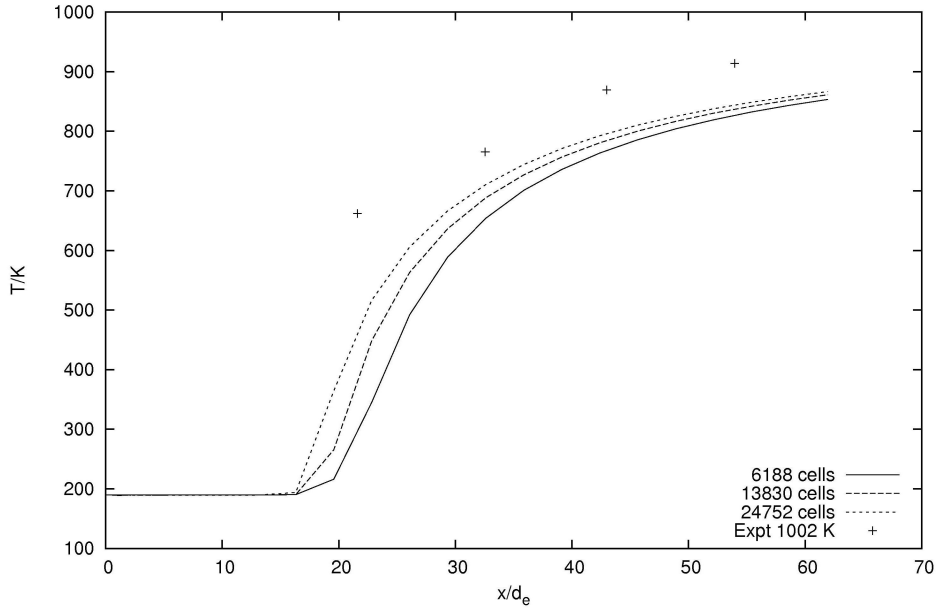

Figure 3.45: Mesh independence analysis for ambient temperature = 1002 K. SMART for momentum discretisation, turbulence modification using [5] and $\sigma_T = 0.5$

- therefore, this model can be used in the case of high speed flow jets.

To match empirical data from [14], the oxygen case is run for all three ambient temperatures of 285 K, 772 K and 1002 K, at different values of C_μ. The results for each case is shown in Figures 3.46, 3.47 and 3.48 below.

The best values of C_μ for each ambient temperature are given in Table 3.11. This is expected, since a larger ambient temperature restrains the jet expansion to the surrounding, thereby decreasing the turbulent mixing zone. Effectively, μ_t should be of lower value than that predicted by the standard $k-\varepsilon$ model at high ambient temperatures. Since C_μ is proportional to μ_t, C_μ should be decreased as ambient temperature increases to reflect this physical behaviour. Also, when the jet angle is narrowed due to decreased turbulent mixing, the core length of the jet is increased - resulting in larger core lengths and slower rate of decay of axial velocity at large ambient temperatures.

Assuming an exponential function fits the data, the following model can be used

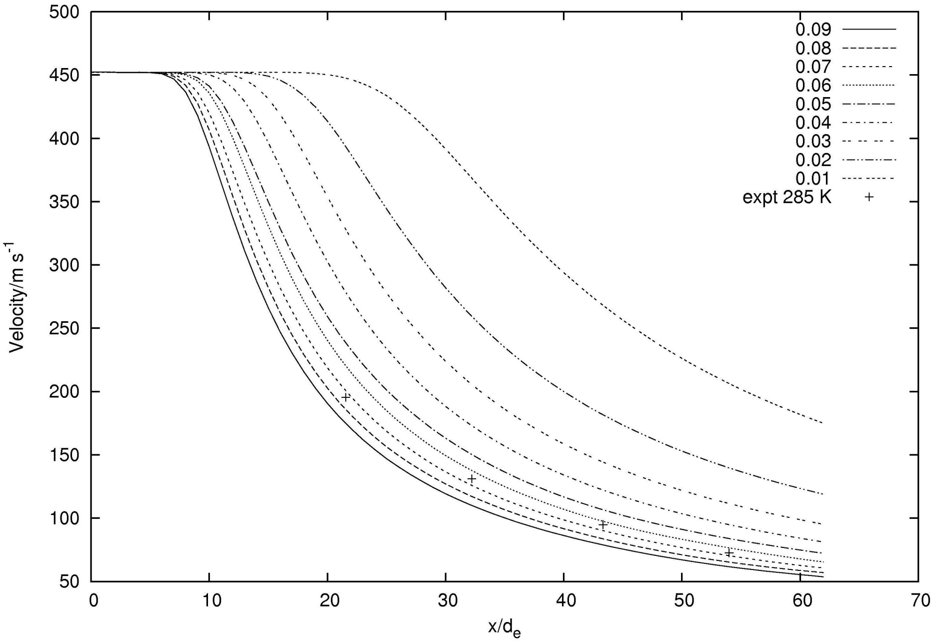

Figure 3.46: Dependence of axial velocity on C_μ. Temperature - 285 K

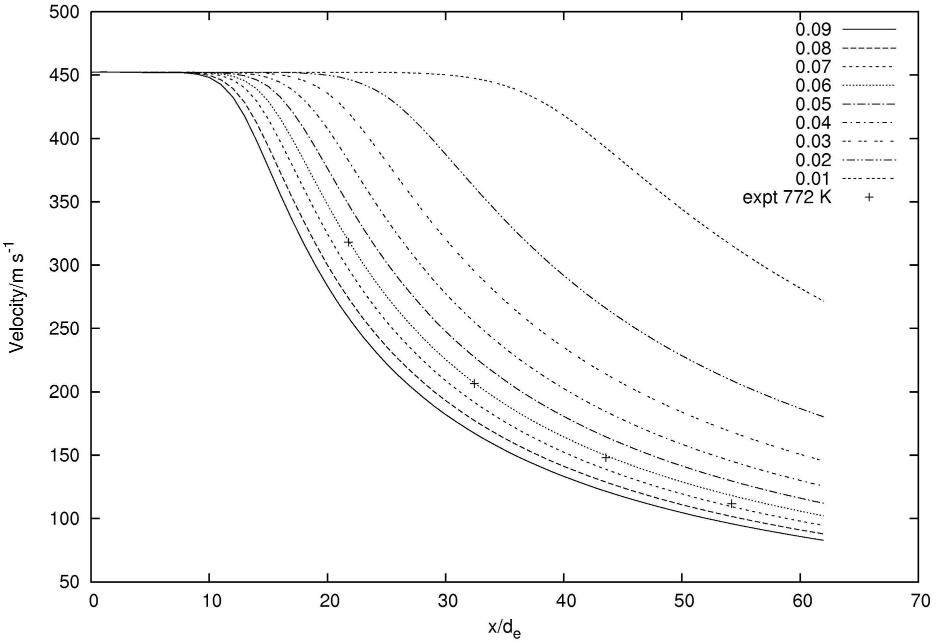

Figure 3.47: Dependence of axial velocity on C_μ. Temperature - 772 K

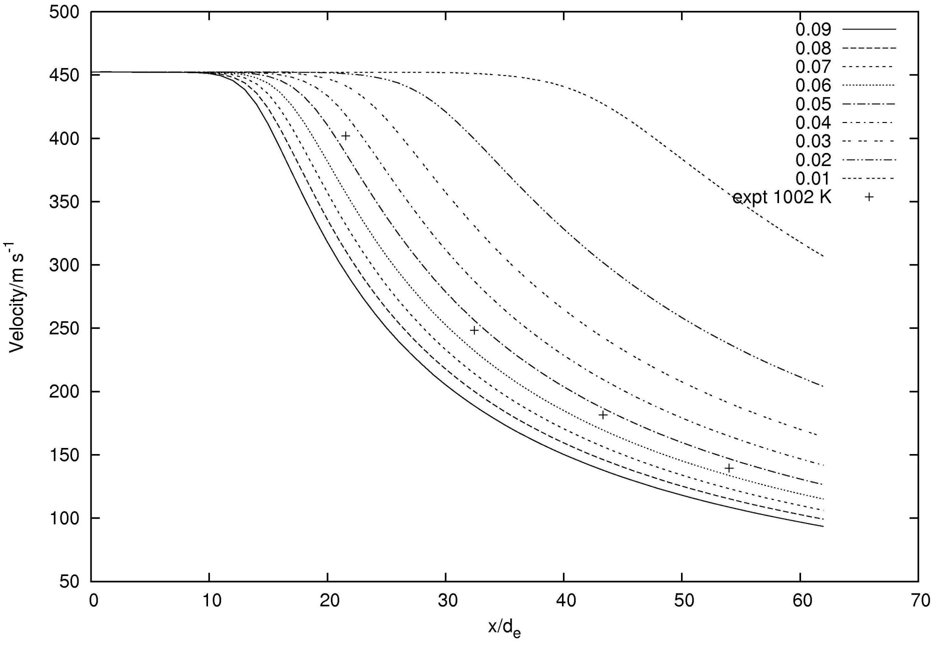

Figure 3.48: Dependence of axial velocity on C_μ. Temperature - 1002 K

Table 3.11: Best values of C_μ for different ambient temperatures

Temperature / K	C_μ
285	0.07
772	0.06
1002	0.05

to determine C_μ at each ambient temperature.

$$C_\mu = A \exp\left(-b\frac{h_{amb}}{h_{jet}}\right) \tag{3.65}$$

where $A = 0.081$, $b = 0.14$, h_{amb} is the enthalpy of the ambient gas and h_{jet} the enthalpy of the gas coming out of the nozzle. Enthalpy is used as a model parameter, since energy, and not velocity or temperature alone, is of paramount importance in compressible flows. Enthalpy being both a function of velocity (and therefore Mach number) and temperature is a suitable parameter for compressible, high temperature flows.

The coefficient of this model as ambient enthalpy decreases to zero is not 0.09 as in the incompressible $k - \varepsilon$ model [22; 23]: since this model has been derived for a specific case - a compressible, cold jet entering a hot environment - it is dangerous to extrapolate this model out of the experimental range of the data from which it has been derived, i.e. [285 K,1002 K].

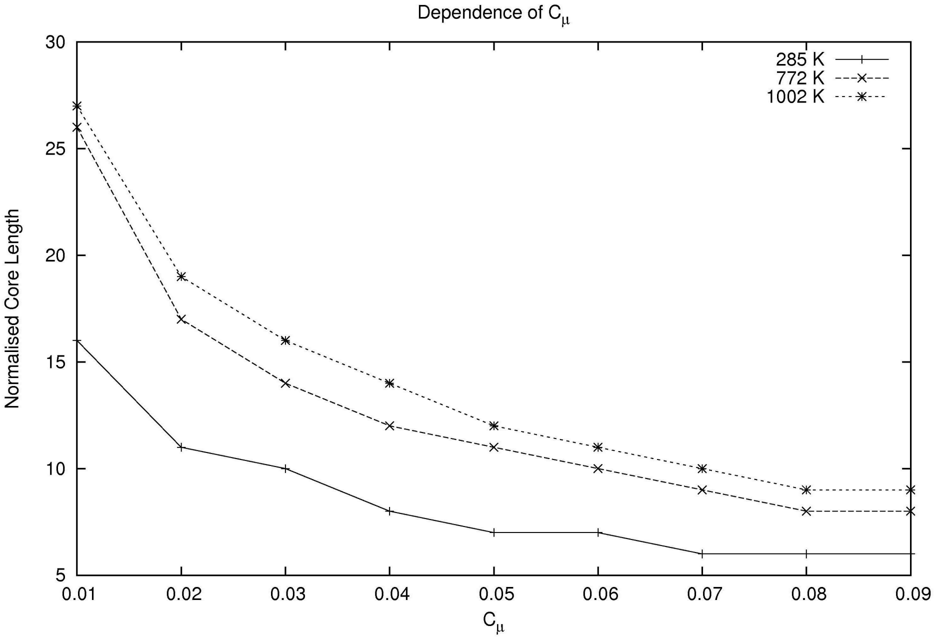

Figure 3.49: Dependence of core length on C_μ. All temperatures

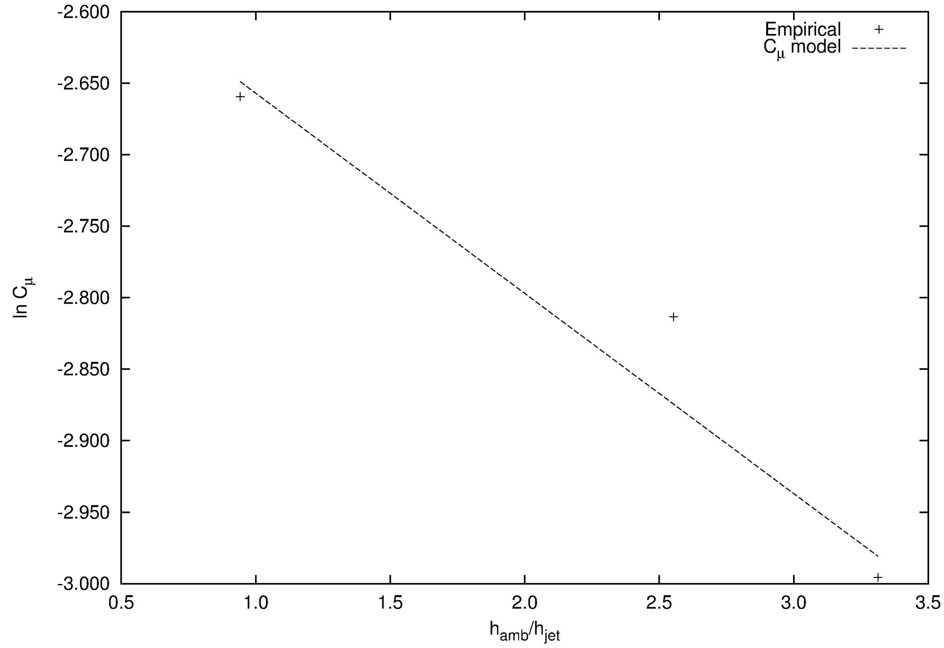

Figure 3.50: C_μ **model.** The line $\ln C_\mu = -0.14\frac{h_{amb}}{h_{jet}} - 2.517$ fits the best C_μ values for each different ambient temperature case, with coefficient of determination $R^2 = 0.9377$.

This model, used in conjunction with the HYBRID scheme, yields mesh independent results for a low number of cells in the computational domain, as shown in Figures 3.51, 3.52, 3.53, 3.54, 3.55 and 3.56.

The predicted results for the axial velocities and temperatures are shown in Figures 3.57 and 3.58. Again the model under-predicts temperature for large values of ambient temperatures. This temperature profile is difficult to predict, since we have to match the configuration of a transient experiment. In the experiment from Sumi *et al.* [14], the cold jet enters a pre-heated furnace, which will obviously cool down as time marches. The paper did not provide the times at which these measurements took place, making it difficult at which (computational) time the profile could be averaged. It is therefore unclear whether this discrepancy is due to a wrong prescription of thermal properties, or simply not replicating the same experiment numerically.

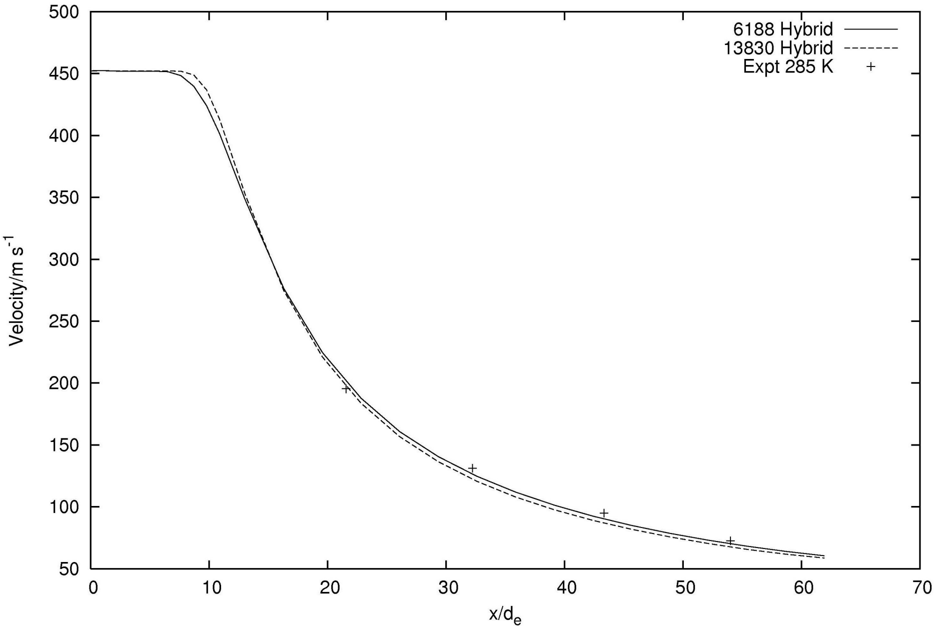

Figure 3.51: Mesh independence analysis for ambient temperature = 285 K - Velocity. HYBRID $C_\mu = 0.07$

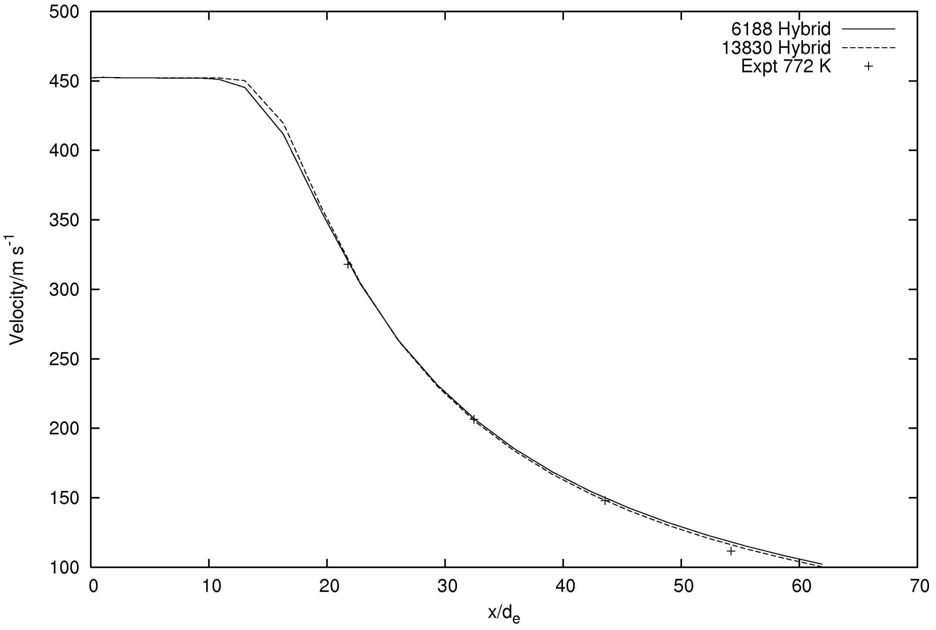

Figure 3.52: Mesh independence analysis for ambient temperature = 772 K - Velocity. HYBRID $C_\mu = 0.06$

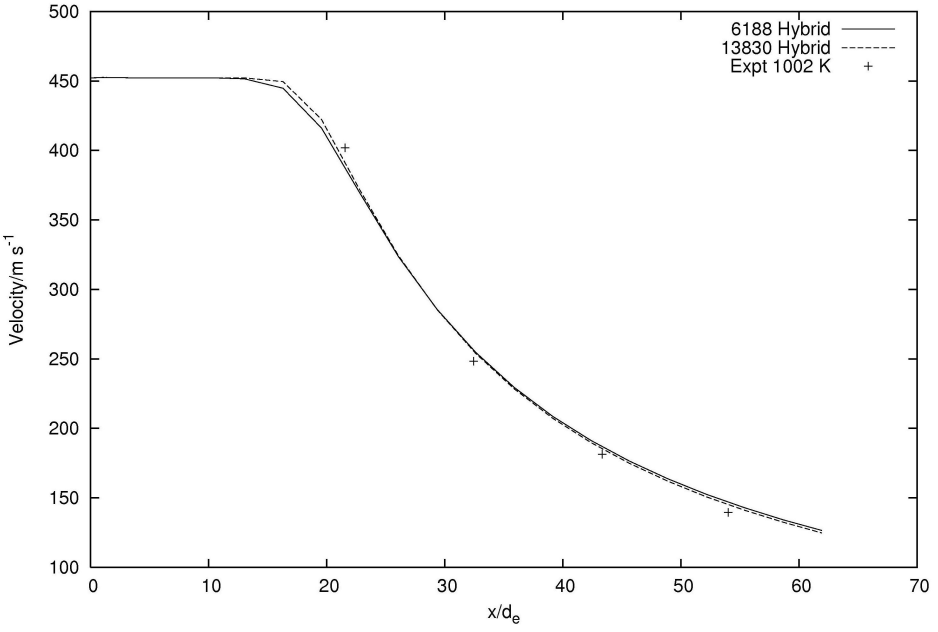

Figure 3.53: Mesh independence analysis for ambient temperature = 1002 K - Velocity. HYBRID $C_\mu = 0.05$

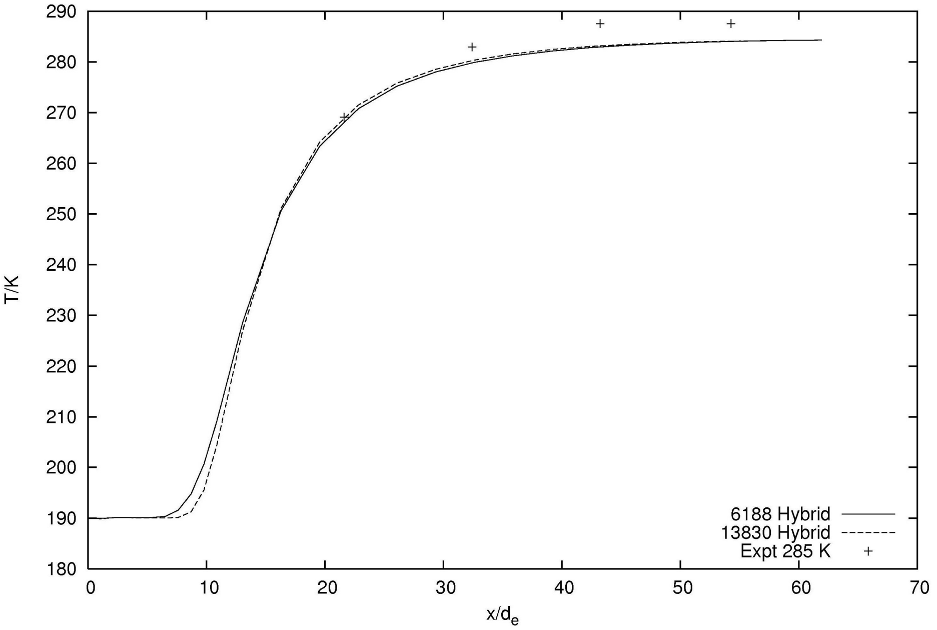

Figure 3.54: Mesh independence analysis for ambient temperature = 285 K - Temperature. HYBRID $C_\mu = 0.07$

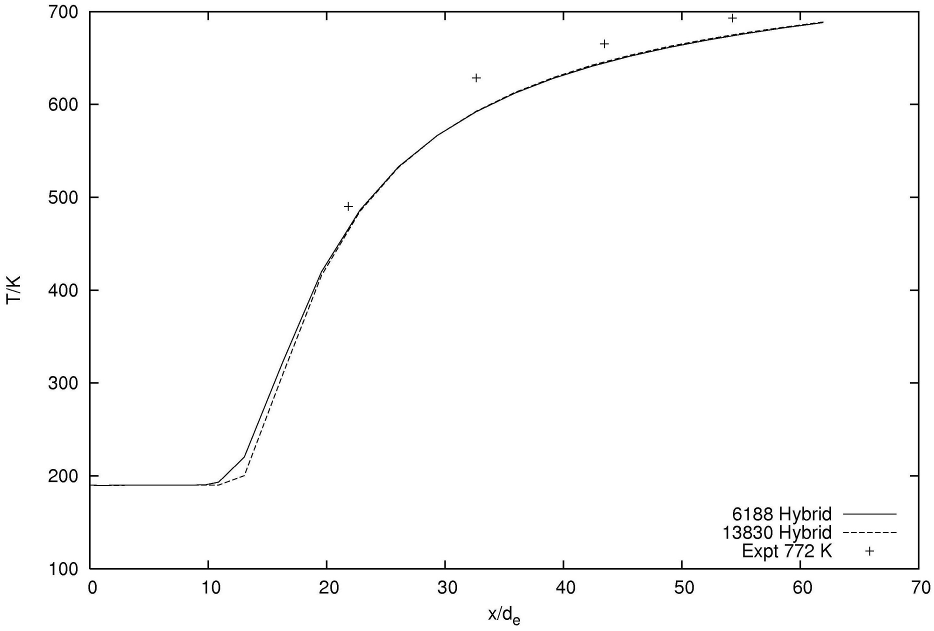

Figure 3.55: Mesh independence analysis for ambient temperature = 772 K - Temperature. HYBRID $C_\mu = 0.06$

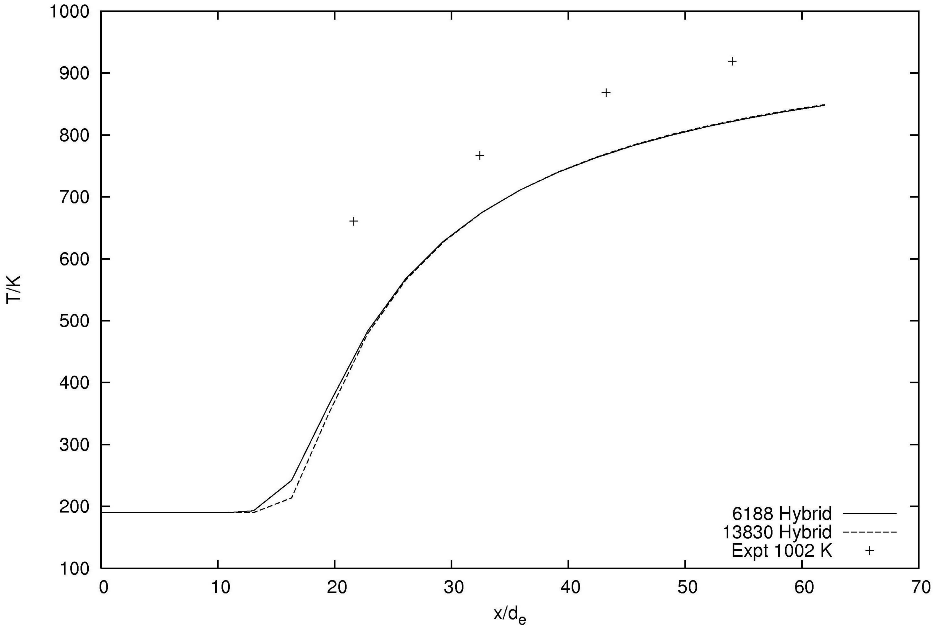

Figure 3.56: Mesh independence analysis for ambient temperature = 1002 K - Temperature. HYBRID $C_\mu = 0.05$

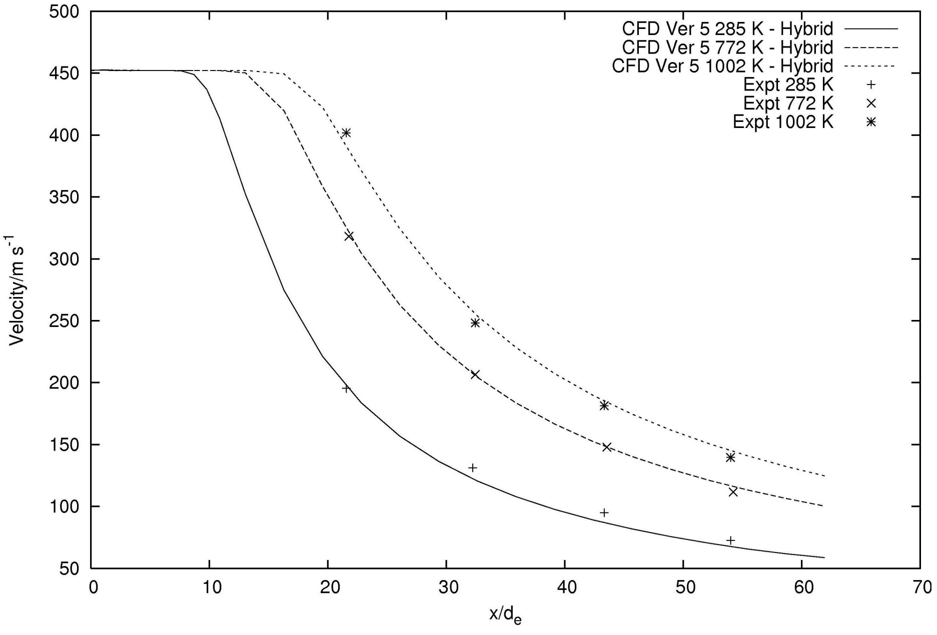

Figure 3.57: Axial velocity results. Mesh - 13830

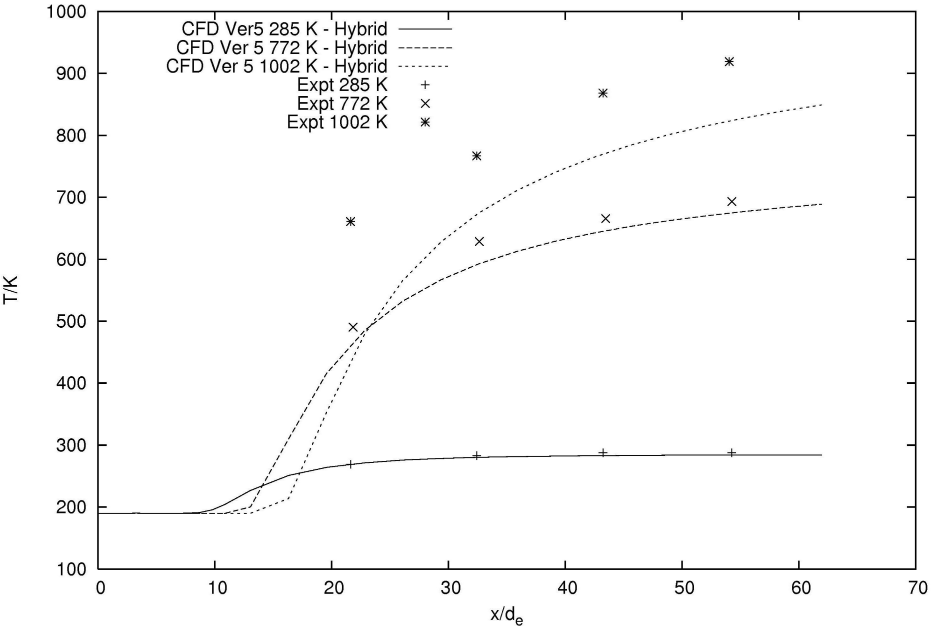

Figure 3.58: Axial temperature results. Mesh - 13830

3.3.4.2 Comparison of CFD results with model from Ito *et al.* [6]

Ito and Muchi [6] proposed the following jet model

$$-\frac{1}{2\ln(1-U_m)} = \alpha\sqrt{\frac{\rho_a}{\rho_n}}\frac{z}{d_n} - \beta \tag{3.66}$$

where $U_m = \frac{U}{U_n}$, the n subscript denote a variable measured at the nozzle exit and the a subscript a variable in the ambient surroundings. $\alpha = 0.0841$ and $\beta = 0.6035$.

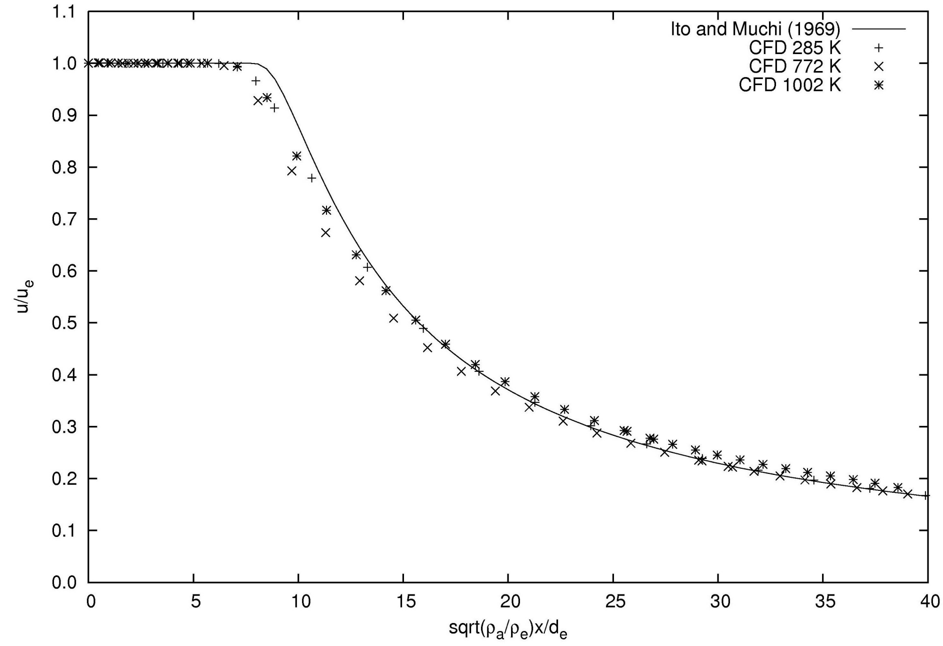

Figure 3.59: Comparison of CFD result with model from Ito and Muchi [6].

The computational results compare well with the empirical model for the all ambient temperatures, as shown in Figure 3.59.

3.4 Summary

This chapter presented the problem of compressibility in computational fluid dynamics and a procedure to solve compressible flows using a collocated, unstructured, pressure-based finite volume method. The results presented in this chapter compare well with analytic solutions and benchmarks in the literature.

Of particular interest was a problem of the oblique shock generated when a high speed flow impinging on a wedge is run using a randomly deformed mesh. The analytic solution was recovered even though the mesh was artificially deformed, implying the numerical scheme used is robust and can be applied to complex engineering problems.

Although the numerical method works well with supersonic cases, including those with inlet velocities at Mach 2.0, transition cases in two dimensions have not been accurately modelled, suggesting further work is required to capture shocks arising during transition.

Finally a new turbulence model to be used with a compressible jet entering a hot environment is developed using empirical data available in the literature. This model is used in the next chapter when hitting this modelled jet on a free surface.

Chapter 4

SUPERSONIC JET IMPINGEMENT ON A LIQUID SURFACE

4.1 Review

Experimental studies of axisymmetric jets hitting surfaces using cold models: with an air jet impinging on water, can be found in the literature [99; 100; 101]. These results are used to validate free surface codes before they are applied to high temperature cases - since there is no empirical measurement publicly available for the deformation of molten liquid surfaces by compressible gas jets.

Of interest here is the case of a high speed compressible jet hitting a hot free surface of a liquid metal. Numerical models simulating the deformation of the surface of a molten liquid bath due to a high speed gas jet are not common in the literature, although some papers related to the topic can be found [102; 103; 104; 105; 106].

Ersson *et al.* [105] used a volume of fluid scheme developed by Nichols and Hirt [107] - a scheme defining Φ as unity in a liquid and zero elsewhere, to track the liquid and gas interface. They surprisingly found that their predictions for the gas penetration depth in the liquid were independent of the turbulence model used. Their model does not include thermodynamics.

Nguyen and Evans [103] also used the volume of fluid method in their numerical experiment. They modelled the deformation of the free surface using the Young-Laplace equation by using the pressure jump across the interface.

4.2 Free surface model

The free surface flow formulation was first proposed by Harlow and Welch [108] to track an interface of water moving in a computational domain. The basis of the free surface model is to treat the variable Φ in the conservation equation (1.1) as a marker to track the volume fraction of one of the fluids in a particular control volume cell. All coefficients and sources are set to 1.0, leading to the following equation.

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\Phi) = 0 \tag{4.1}$$

Pericleous *et al.* suggested the Scalar Equation Algorithm (SEA) [109] to model flow involving a free surface between a gas and a liquid. Φ here represents the liquid volume fraction in a cell. Therefore $\Phi = 1.0$ when the cell is full of liquid, $\Phi = 0.0$ if the cell is completely filled with gas, and a value between 0.0 and 1.0 calculated as a linear function of the volume concentration of each fluid if the cell contains both fluids. A material property χ in each cell is then calculated using

$$\chi = \Phi \chi_{liq} + (1 - \Phi)\, \chi_{gas} \tag{4.2}$$

The governing equations of the form (1.1) need to be carefully discretised taking into account each material property is dependent on Φ.

4.2.1 Continuity handling

The mass conservation equation is given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{4.3}$$

where density is now a function of Φ.

$$\rho = \Phi \rho_{liq} + (1 - \Phi)\, \rho_{gas} \tag{4.4}$$

There are two ways in which continuity can be handled: either by volume conservation, or by mass conservation.

4.2.1.1 Gas and Liquid Analyser (GALA)

Substituting equation (4.4) into (4.3), and assuming that both gas and liquid are incompressible, the following equation is obtained.

$$(\rho_{liq} - \rho_{gas}) \left[\frac{\partial \Phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\Phi) \right] + \rho_{gas} \nabla \cdot \boldsymbol{u} = 0 \tag{4.5}$$

Using equation (4.1), (4.5) reduces to

$$\nabla \cdot \boldsymbol{u} = 0 \tag{4.6}$$

This is the volume conservation formulation of continuity used by default in PHYSICA [15]. If one or both of the two fluids is compressible, another source term equal to

$$-\frac{\rho_{liq} - \rho_{gas}}{\rho} \left[\frac{\partial \Phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\Phi) \right] \tag{4.7}$$

is added to the pressure correction equation [15].

Another way of expressing the source term to account for the large density difference between the gas and the liquid is

$$-\frac{\partial (\ln \rho)}{\partial t} - \boldsymbol{u} \nabla (\ln \rho) \tag{4.8}$$

When the Rhie-Chow interpolation [29] is performed in GALA, the weighing factor α is based on both the relative distances from the face, and the buoyancy force in the neighbouring cells.

$$\alpha = \left(\frac{\alpha \boldsymbol{d} \cdot \boldsymbol{B}_A}{V_A}\right) / \left(\frac{(1-\alpha)\, \boldsymbol{d} \cdot \boldsymbol{B}_B}{V_B} + \frac{\alpha \boldsymbol{d} \cdot \boldsymbol{B}_A}{V_A}\right) \tag{4.9}$$

4.2.1.2 Pressure gradients with GALA

The discontinuity of properties across the interface requires careful consideration when evaluating gradients across the interface. Of particular interest is the calculation of pressure gradients across the free surface. The technique used to evaluate pressure gradients is given as follows: [15]

Pressure gradients are estimated using the following expression.

$$\int\limits_V \frac{\partial p}{\partial x} \, \mathrm{d}V = \sum_f A_f p_f n_x \tag{4.10}$$

Face pressures p_f is calculated using

$$p_f = \alpha p_P + (1-\alpha)\, p_A \tag{4.11}$$

where, in the presence of a free surface,

$$\alpha = \frac{d_{AF}\left(\rho_A - \rho_{ref}\right)}{d_{AF}\left(\rho_A - \rho_{ref}\right) + d_{PF}\left(\rho_P - \rho_{ref}\right)} \tag{4.12}$$

d_{AF} is the perpendicular distance from cell centre A and the face f.

4.2.1.3 Zonal GALA

A novel technique to handle continuity in compressible free surface flows is presented in this thesis. A zonal GALA method is introduced in the following manner: GALA is used for most part of the domain, and the compressible procedure described in Section 3.2.4 is used for a compressible region of interest. The criteria used to discriminate between the two zones are described and discussed in the results sections.

At initialisation, the zone where GALA is to be used is defined using the criterion to be obeyed. For example, if GALA is to be used in cells for which Mach number is less than 0.3, the following procedure is followed at initialisation:

1. A flag array equal to the total number of cells is initialised: this flag is to hold the switch determining where GALA is to be used.
2. The Mach number is calculated for each cell in the domain.
3. For each cell where the GALA criterion is obeyed, the flag is set to 1. Other cells are given a flag of 0. Thus the zone where GALA is to be invoked consists of cells with a flag of 1.

The following changes are then to be effected on the pressure correction procedure:

- A temporary place-holder for "density" is used for each cell. The density used for the zone where GALA is invoked, i.e. with a flag of 1 is set to a value of 1.0: in these cells, only volume conservation matters. The physical density is used for the cells in which the flag is 0.
- The convection coefficients are to be zero in cells where GALA is to be used, as in the compressible case. The array storing values of ι are then made to be zero for these cells, so no convection term is to appear in the resulting discretisation equation for pressure correction in those cells.

In effect, the pressure correction equation for cases involving zonal GALA points towards

- mass conservation in the compressible region defined by the 0 zone, and
- volume conservation in the region where GALA is invoked. This region should in fact consists of cells where Φ is significantly large, i.e. $\Phi > 0.001$ for cases consisting of a liquid whose density is 1000 times greater than the gas density.

It is expected that p' will become 0 everywhere at convergence and that, in effect, volume conservation where liquid fraction is significant, is actually mass conservation, as in the standard GALA procedure. In order to check issues with mass conservation, especially at the interface of the GALA zone, the proper mass imbalance is computed using physical densities for each cell to consider if mass conservation is indeed achieved.

The zones are then updated in the following manner:

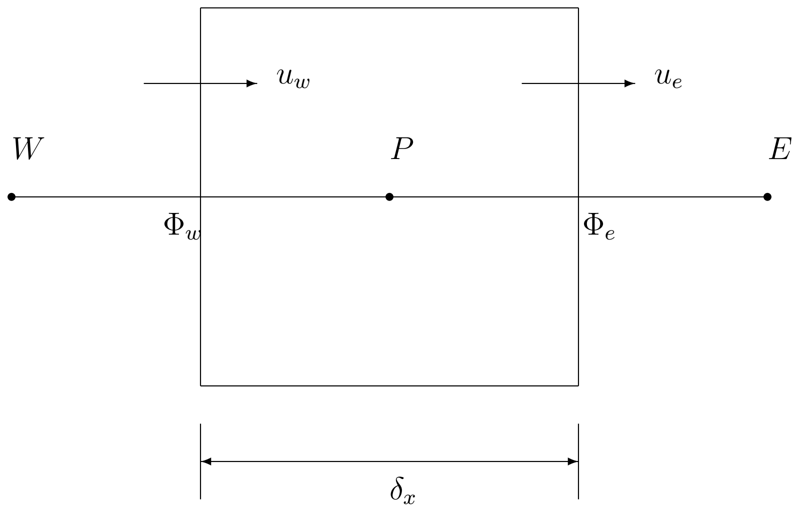

Figure 4.1: Control volume cell to explain van Leer differencing scheme

1. at the end of each time step for time explicit calculations, and
2. at the end of each iteration for time implicit calculations.

This zonal discrimination section is pertinent when the density differences between the two fluid involved is huge. For the gas, compressibility is important since it leads to significant density differences, particularly around the jet. However, compressibility is irrelevant for a liquid or regions far away from the jet. A zonal method can be used to apply the slower compressible code only in the gas region and make use of the stable GALA method for the liquid.

4.2.2 Van Leer

The numerical smearing of the gas-liquid interface is the main disadvantage of using fixed-grid free surface methods. In order to mitigate smearing, the SEA algorithm uses a second order Total Variation Diminishing (TVD) scheme suggested by Bram van Leer [110] to discretise equation (4.1).

The van Leer scheme is derived from the solution of the equation

$$\frac{\partial \Phi}{\partial t} + u\frac{\partial \Phi}{\partial x} = 0 \tag{4.13}$$

for the convection of Φ by velocity u. Referring to Figure 4.1,

$$\Phi_P|^{n+1} = \Phi_P|^n - \frac{\delta t}{\delta x}\left(u_e\Phi_e - u_w\Phi_w\right)|^n \tag{4.14}$$

The van Leer scheme computes the face value Φ_e as

$$\begin{aligned} \Phi_e &= \Phi_P + \frac{\delta x}{2}\left[\frac{\partial\Phi}{\partial x}\right]_P\left[1 - \frac{u_e\delta t}{\delta x}\right] \text{for } u_e > 0 \\ \Phi_e &= \Phi_P - \frac{\delta x}{2}\left[\frac{\partial\Phi}{\partial x}\right]_P\left[1 + \frac{u_e\delta t}{\delta x}\right] \text{for } u_e < 0 \end{aligned} \tag{4.15}$$

The gradient $\left[\frac{\partial\Phi}{\partial x}\right]_P$ is given by

$$\left[\frac{\partial\Phi}{\partial x}\right]_P = \begin{cases} \frac{2sgn(\delta\Phi_e)}{\delta x}\min\left(|\delta\Phi_e|, \frac{1}{2}\left(|\delta\Phi_e| + |\delta\Phi_w|\right), |\delta\Phi_w|\right) & \text{if } sgn\left(\delta\Phi_e\right) = sgn\left(\delta\Phi_w\right) \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

where

$$\begin{aligned} \delta\Phi_e &= \Phi_E - \Phi_P \\ \delta\Phi_w &= \Phi_P - \Phi_W \end{aligned} \tag{4.17}$$

$$\begin{aligned} sgn\left(\delta\Phi_e\right) &= +1 \text{ if } \delta\Phi_e \geq 0 \\ sgn\left(\delta\Phi_e\right) &= -1 \text{ if } \delta\Phi_e \leq 0 \end{aligned} \tag{4.18}$$

If an internal face does not having an upwind-upwind element associated with it, the face value of Φ is given by the upwind value itself. The above scheme is time explicit: the values of Φ at the beginning of each time-step are required.

To avoid instabilities arising due to time explicit calculations, the interface must move by only a fraction of a cell at each time step. This condition - the Courant-Friedrichs-Lewy criterion [111] - is implemented mathematically by limiting the time step value according to

$$\delta t < \min\left(|\frac{\delta x}{u}|, |\frac{\delta y}{v}|, |\frac{\delta z}{w}|\right) \tag{4.19}$$

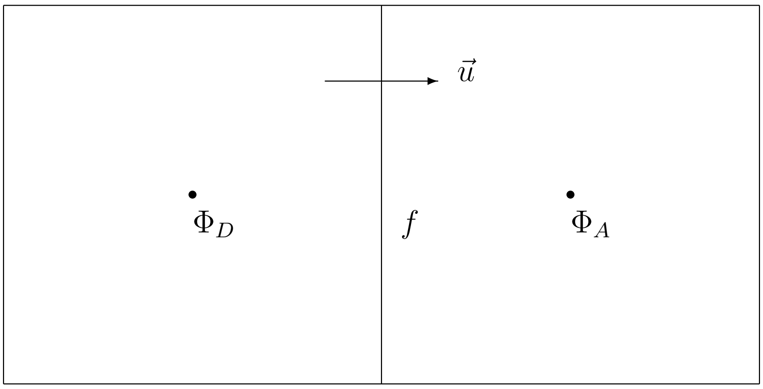

Figure 4.2: Donor Acceptor description

4.2.3 Donor acceptor

The Donor Acceptor method [112] is another way of computing face value terms for Φ. The upwind cell of the face value where Φ is being evaluated is referred to as the donor cell (with subscript $_D$), and the downwind cell is the acceptor cell (with subscript $_A$). The value of Φ at face f is

$$\Phi_f = \max\left(0, \min\left(\frac{d_\Phi}{cr}, 1\right)\right) \tag{4.20}$$

where

$$cr = |\boldsymbol{u} \cdot \hat{\boldsymbol{n}}| \cdot \delta t \tag{4.21}$$

$$d_\Phi = \min\left(\Phi_{AD} \cdot cr + C_\Phi, \Phi_D \frac{A}{V_D}\right) \tag{4.22}$$

$$C_\Phi = \max\left(0, (1 - \Phi_{AD}) \cdot cr - (1 - \Phi_D) \cdot \frac{A}{V_D}\right) \tag{4.23}$$

A is the face area, V_D is the donor cell volume and the subscript $_{AD}$ indicates either a donor or acceptor cell value.

4.2.4 Level set

The level set method for Osher and Sethian [113] is another technique used to track free surfaces. In the level set method, the zero level set of continuous, smooth and monotonic variable Φ corresponds to the position of the propagating interface. The evolution of Φ is calculated using a velocity field in conjunction with the advection equation [15].

One of the earliest uses of the level set method to compute incompressible two-phase flow is can be found here [114]. Since the level set method is prone to mass loss, Sussman and Puckett [115] combined the volume of fluid method with level set. Enright *et al.* [116] proposed a particle level set method by passively adding advected marker particles near the interface.

More accurate solutions with the level set methods were sought via the use of adaptive mesh refinement methods, quadtree based (2D) and octree (3D) based level methods [117; 118; 119; 120; 121].

4.2.5 Counter Diffusion Method [7]

Computational modelling of real phenomena involve discretizing physical equations in space and time. These discretized equations are more prone to diffusion than the original differential equations. Numerical diffusion is even more problematic for free surface flows simulated with a free surface variable, as compared with using a moving mesh. This numerical diffusion leads to the smearing of the sharp interface between the two fluids [21].

The Counter Diffusion Method is a recent time-implicit free surface model proposed by Pericleous *et al.* [7] to counter the smearing of the interface by false diffusion. This method was inspired by the gravity sifting of droplets and bubbles at an interface. If the interface is taken to be $\Phi = 0.5$, any value of $\Phi < 0.5$ corresponds to droplets, and $\Phi > 0.5$ represents bubbles. A normal counter diffusion flux, analogous to the slip velocity of real bubbles and droplets, is computed on each internal face, and applied as source term in a conservative way so as to separate these two regions.

The scalar Φ which takes values in the range $[0, 1]$ is solved as a conserved variable:

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\Phi) = S_{\Phi} \tag{4.24}$$

The source S_{Φ} contains the corrections from the computed counter diffusion flux Q computed at each internal face.

$$Q = C|\boldsymbol{u} \cdot \hat{\mathbf{n}}|A\,(1 - \Phi_{down})\,\Phi_{up} \tag{4.25}$$

where C is a scaling factor used to adjust the strength of the counter diffusion. $|\boldsymbol{u} \cdot \hat{\mathbf{n}}|A$ is the face-normal area velocity product. Φ_{up} and Φ_{down} are the upwind and downwind values of Φ. If the downwind cell is completely full of liquid, i.e. $\Phi_{down} = 1.0$, the $(1 - \Phi_{down})$ term ensures that the interface does not move down. Similarly, the Φ_{up} factor ensures that no counter diffusion flux is taken from cells full of gas.

The counter diffusion method has been validated by Pericleous *et al.* [7] and Wang [21], who compared the prediction of CDM computations against a collapsing water column experiment by Martin and Moyce [122]. Wang *et al.* [123] successfully applied the counter diffusion method to model the tilt casting process.

4.2.6 Surface tension

Across any line drawn on the liquid interface, there is a surface tension of magnitude γ per unit length, pulling the top liquid molecules in [124]. This force arises since the top layer of molecules do not have top neighbours to form bonds with and are pulled towards the inside. The expression for the force due to surface tension is given by

$$F_{\gamma} = \oint_{l} \vec{\tau} \cdot \vec{\gamma}\,\mathrm{d}l \tag{4.26}$$

where $\vec{\tau}$ is the unit vector tangential to the interface and normal to line l drawn on the interface. There is also a pressure difference across the interface, which balances with the surface tension. This balance is described by the Young Laplace equation

Table 4.1: Boundary conditions for oxygen jet on free surface case

Inlet	Total Pressure	3.6733×10^6 Pa
	Mach number	1.72
	Temperature	300 K
	Turbulent intensity	2%
Outlet	Pressure	1.0135×10^5
	Velocity	$u_{interior}$ (Extrapolated)
	Temperature	300 K

$$\triangle p = \frac{\gamma}{J} \tag{4.27}$$

where J is the surface curvature.

4.2.6.1 Surface tension source term in momentum equations

The momentum source term due to surface tension is given by [18]

$$\boldsymbol{S}_\gamma = \boldsymbol{n}\gamma J|\nabla\Phi| \tag{4.28}$$

The normal $\boldsymbol{n}$ is computed by

$$\boldsymbol{n} = \frac{\nabla\Phi}{|\nabla\Phi|} \tag{4.29}$$

and the curvature J using

$$J = \nabla \cdot \boldsymbol{n} \tag{4.30}$$

4.3 Oxygen jet impinging on a free surface

A series of transient test cases are conducted using the boundary conditions detailed in Table 4.1 and initial conditions from Table 4.2. The material properties for both

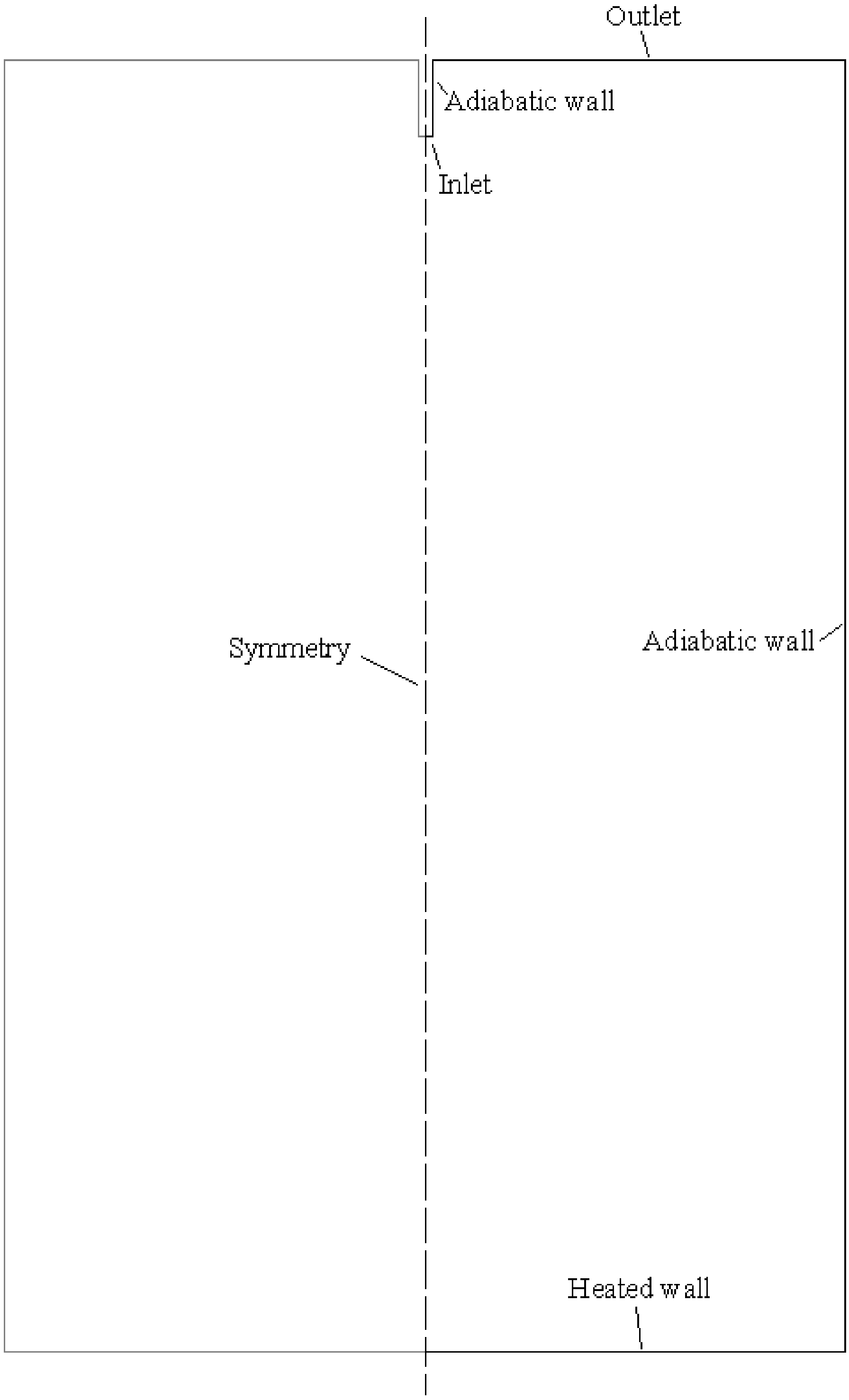

Figure 4.3: Schematic of the geometry used for the oxygen jet impinging on liquid surfae cases.

Table 4.2: Initial conditions for oxygen jet on free surface case

Gas	Pressure	1.0135×10^5 Pa
	Mach number	0
	Temperature	300 K
Liquid	Pressure	1.0135×10^5 Pa
	Mach number	0
	Temperature	300 K

Table 4.3: Material properties for oxygen jet on free surface case

Gas	Viscosity (dynamic)	1.4×10^{-5} Pa s
	Specific heat	910 J kg^{-1} K^{-1}
	Conductivity (thermal)	0.0238 W m^{-1} K^{-1}
	Molecular weight	0.032 kg mol^{-1}
Liquid	Viscosity (dynamic)	0.5×10^{-6} Pa s
	Specific heat	611 J kg^{-1} K^{-1}
	Conductivity (thermal)	30 W m^{-1} K^{-1}
	Density	7150 kg m^{-3}

fluids are given in Table 4.3. A schematic diagram of the geometry is given in Figure 4.3.

4.3.1 Incompressible jet

A test run is first conducted using an incompressible jet hitting the free surface. The counter diffusion method (CDM) of Section 4.2.5 is used to model the interface between the gas and the liquid. The SIMPLEC algorithm is used to solve for velocity components and pressure. The results of this case are shown in Figure 4.4.

The jet develops within a few ms and impinges on the free surface, thereby deforming it and forming a cavity of depth of about 16 nozzle diametres after 0.6 s.

4.3.2 Compressible jet

The following section presents the results for the case of a compressible oxygen jet impinging on the liquid surface, using the zonal GALA method developed in Section 4.2.1.3. Two zonal GALA criteria are described in the following sections: a Mach number discrimination criterion and a fluid phase criterion.

4.3.2.1 Zonal GALA: Mach number criterion

Mach number is used to discriminate between using GALA or not as shown in Figure 4.5. Good convergence is obtained when GALA is used in the liquid and interface region only, while the full compressible algorithm is used in the gas region where the Mach number exceeds 0.3. The boundary conditions of this case are as described in Table 4.1 and the initial conditions are given in Table 4.2. Good convergence is obtained, as expected since, for the liquid part, only volume need to be conserved, since mass due to liquid is much larger than any minute contribution from any gas bubble or speck entering the free surface. In the region for which the gas velocity is less than Mach 0.3, the gas behaves like an incompressible fluid and behaves in the same way as in the incompressible run of Section 4.3.1. On the other hand, full treatment for compressibility is required for the gas, as was required in Section 3.3.4, where only the jet was modelled. The results are shown in Figure 4.6.

The deformation of the interface is similar to that of Section 4.3.1. The jet develops quickly, within a few milliseconds and impinges on the free surface. Liquid metal is pushed aside to the vertical walls forming a cavity directly beneath the nozzle. The cavity depth in this case is slightly lower than when modelling an incompressible jet, being about 14 nozzle diameters after the cavity is "stabilized" 0.6 s after the initial jet impingement.

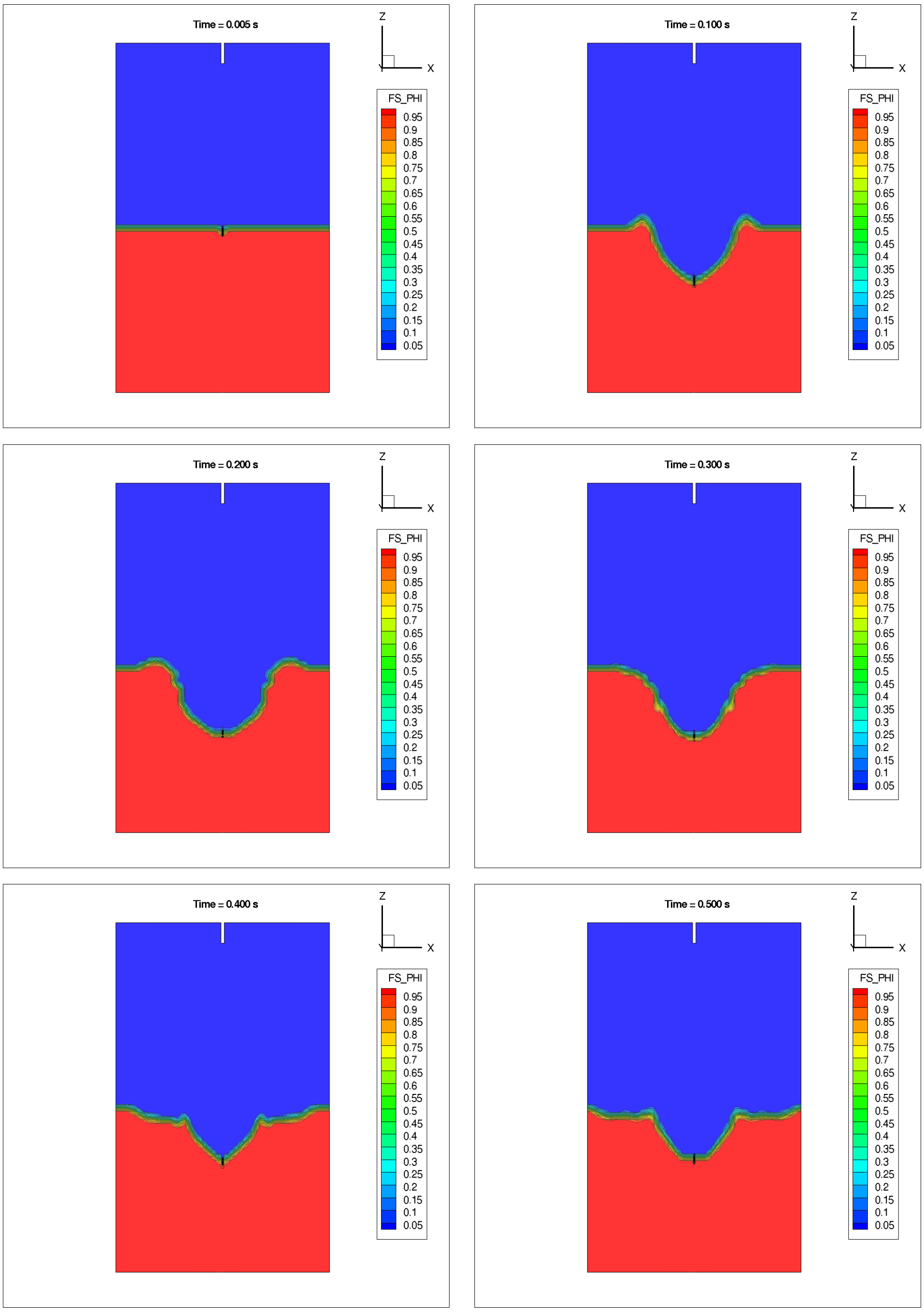

Figure 4.4: Deformation of interface with an incompressible jet - Transient

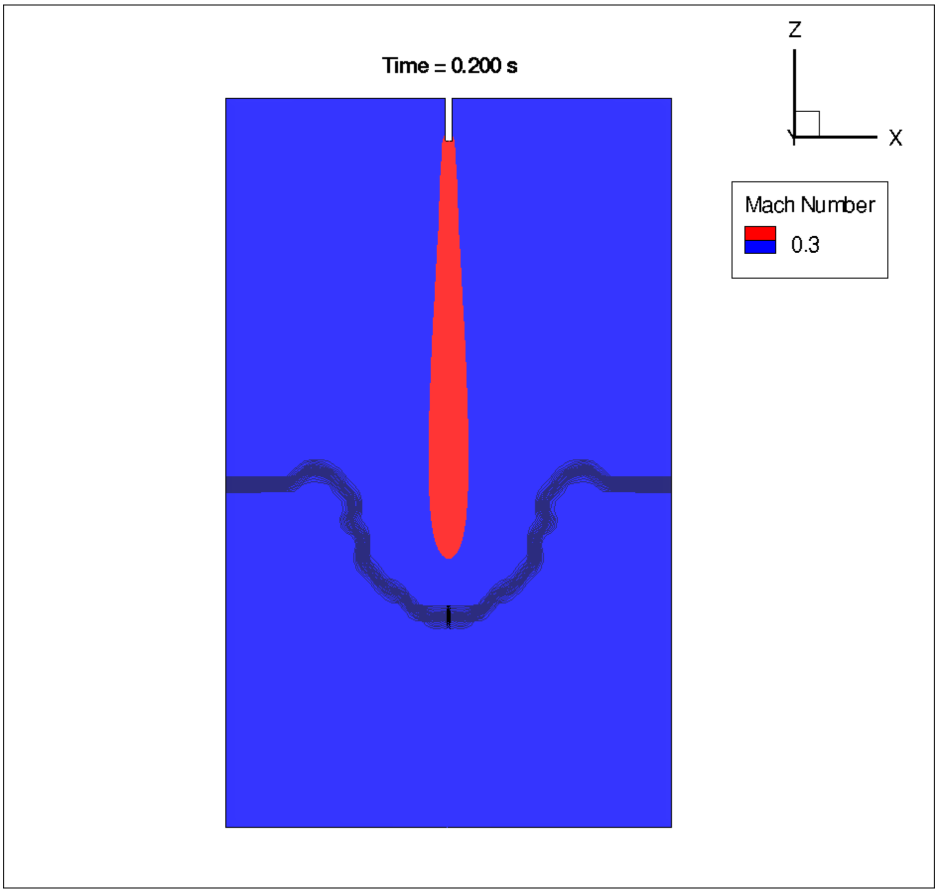

Figure 4.5: Zonal GALA setup for Mach number criterion. GALA is used in the blue region while compressibility procedure from Section 3.2.4 is used in the red region.

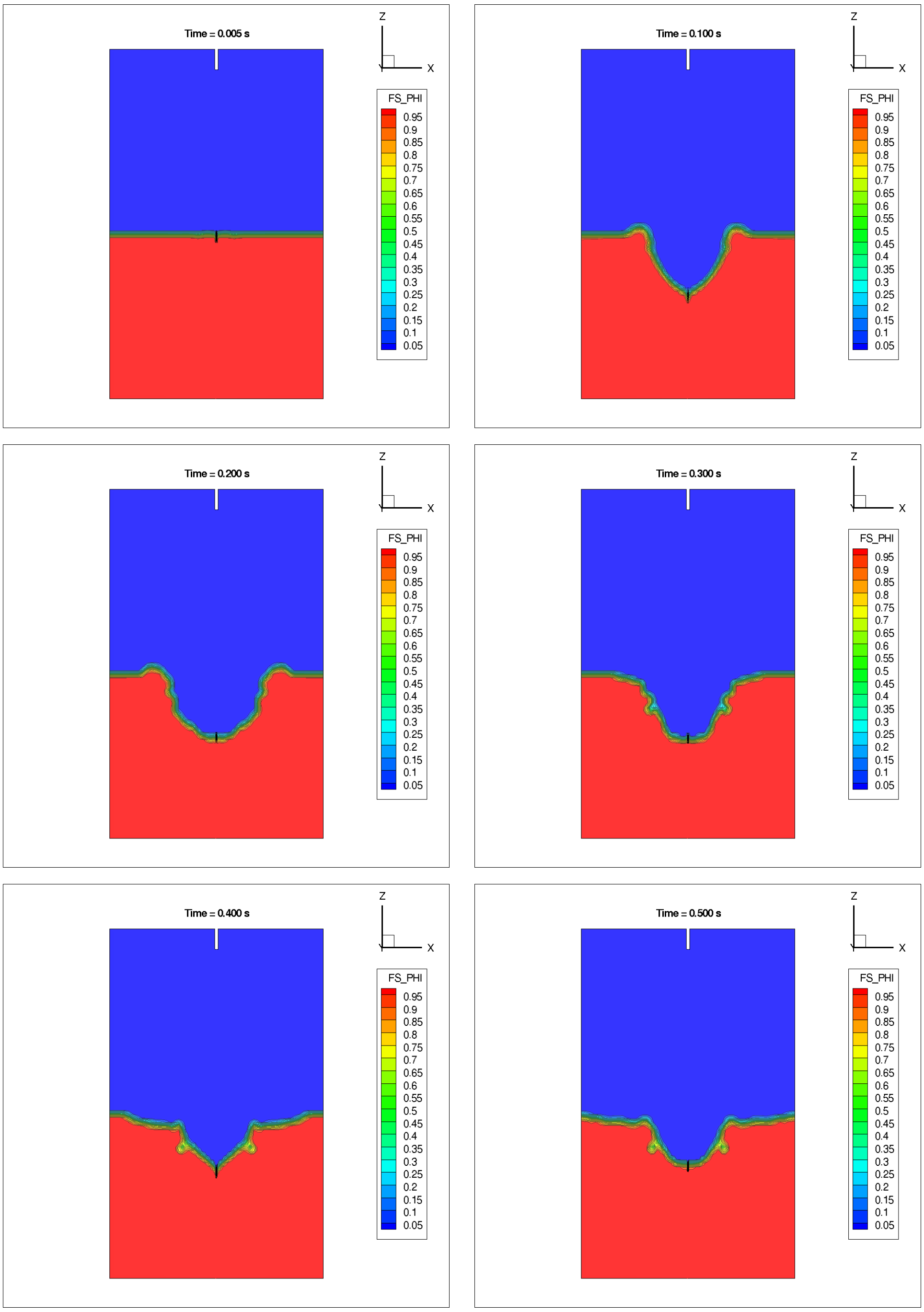

Figure 4.6: Deformation of interface when using the turbulence model from Section 3.3.4.1 and Mach number as zonal GALA criterion - Transient

4.3.2.2 Zonal GALA: Φ criterion

Φ is used to discriminate between using GALA or not as shown in Figure 4.7. Good convergence is obtained when GALA is used in the liquid and interface region only, while the full compressible algorithm is used in the gas region. This is expected since, for the liquid part, only volume need to be conserved, since mass due to liquid is much larger than any minute contribution from any gas bubble or speck entering the free surface. Hence, GALA is applicable for mass conservation in this region. On the other hand, full treatment for compressibility is required for the gas, as was required in Section 3.3.4, where only the jet was modelled. The discontinuity of the physics at the interface need not be a worrying aspect, since the liquid-gas interface is discontinuous in reality. Also, pressure corrections will be zero when the case converge: whether GALA or the compressible pressure correction procedure from Section 3.2.4 are used to achieve a mass conserving solution is irrelevant.

The results obtained using the turbulence model described in Section 3.3.4.1 for the jet, and the counter-diffusion method for the free surface are shown in Figure 4.8. The results are identical to the case from Section 4.3.2.1.

4.3.2.3 Comparison of cavity depth

A comparison of cavity depth with time for each different method is shown in Figure 4.9. There is a small difference between the predictions of the cavity depth using an incompressible jet and a compressible model for the gas when using the Hybrid difference scheme. The zonal criterion for GALA is irrelevant, since both simulations with a Mach number based and a Φ based discrimination criterion predict the same cavity deformation profile. This is expected since, at convergence, all pressure corrections are zero, and the correct mass conserving solution is achieved.

4.4 Summary

This chapter presented the case of a high speed oxygen jet hitting a free surface. The gas jet was made compressible using the numerical method presented in the previous

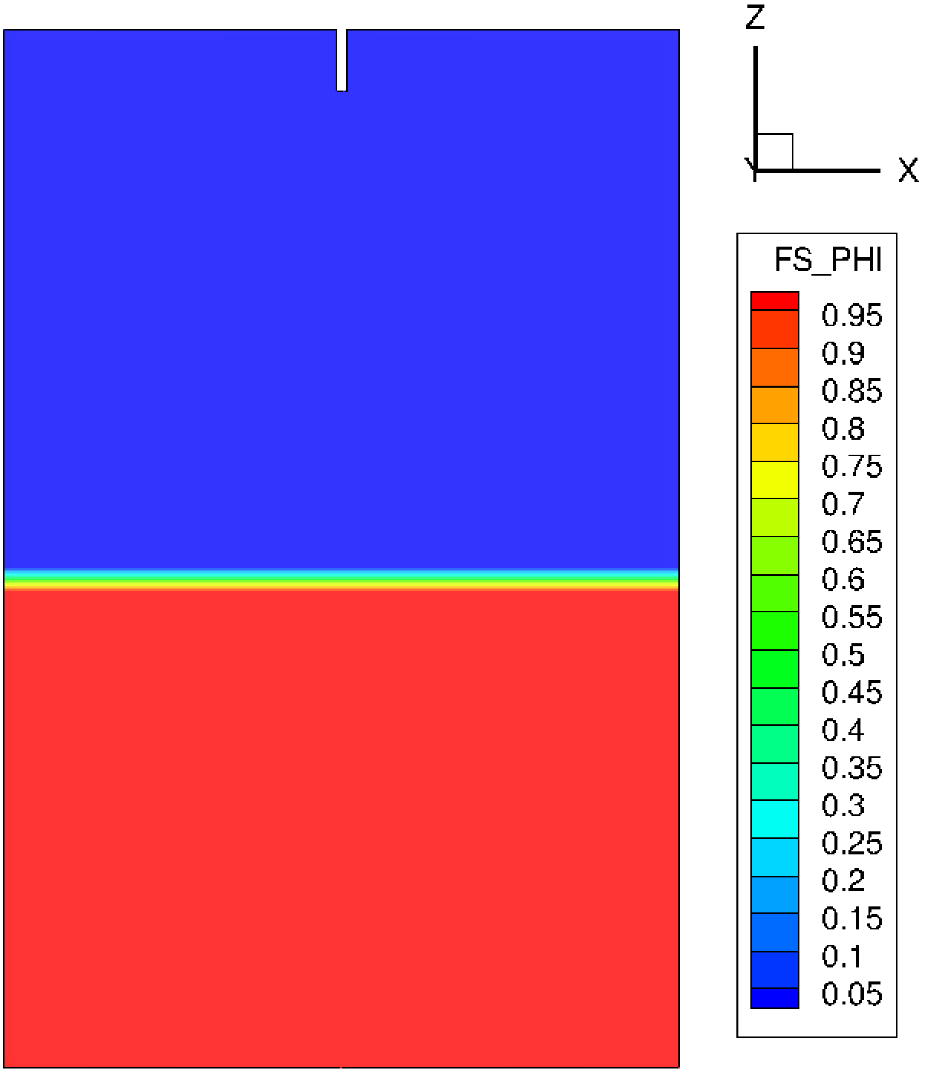

Figure 4.7: Zonal GALA setup for Φ criterion. GALA is used in red region while compressibility procedure from Section 3.2.4 is used in blue region.

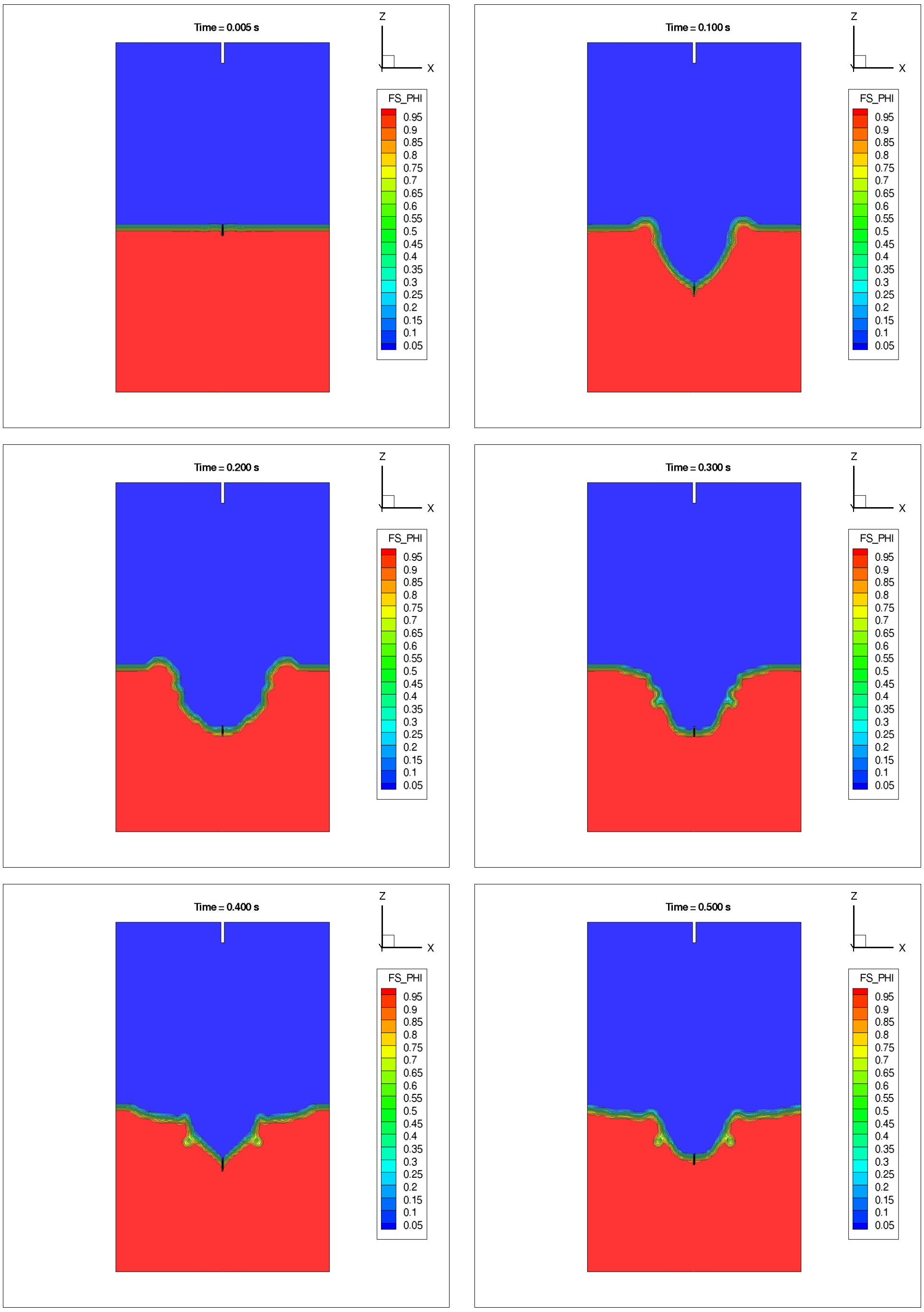

Figure 4.8: Deformation of interface when using the turbulence model from Section 3.3.4.1 - Transient

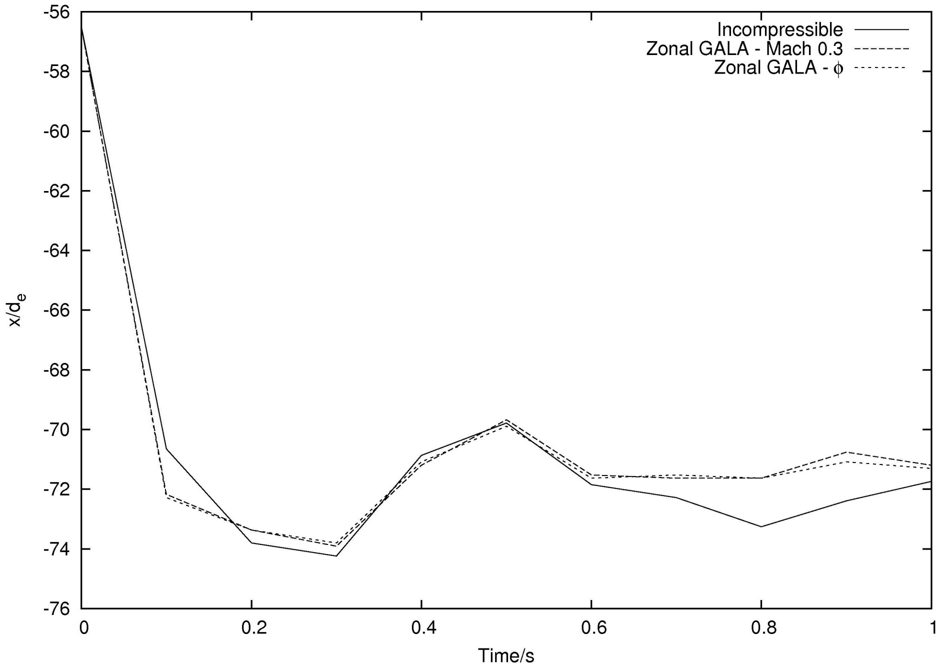

Figure 4.9: Variation of cavity depth normalised with nozzle diameter with time for the different methods described in this chapter. All using the same initial and boundary conditions and material properties as given in Tables 4.1, 4.2 and 4.3.

chapter. A novel zonal GALA method was used to handle the difference between density variations in the liquid and gas regions. Converged solutions were obtained for all cases presented when using the CDM method in conjunction with zonal GALA.

Chapter 5

CONCLUSIONS

This research has been conducted to increase the applicability range of an unstructured, collocated, finite volume code to a wider range of industrial problems. Two problems were given particular attention: stability in the presence of mesh skewness, and variable density. The results obtained have been promising.

5.1 Non-orthogonality

Mesh non-orthogonality problems have been tackled using two approaches: relaxing the non-orthogonality diffusion flux correction procedure, and the introduction of a pressure interpolation method for difficult cases. The non-orthogonality diffusion relaxation parameter was devised by an extension of the well-known deferred correction method for computing fluxes. A new variable ς has been introduced to fine tune the relaxation of the diffusion corrections in the presence of non-orthogonality. However, this parameter does not offer significant speed up to justify its widespread use.

The diffusion correction method works well with benchmarks present in the literature, with good predictions obtained for the case of a deformed moving lid cavity. With meshes generated with a commercial package, this correction is not enough, and diffusion correction over-relaxation does not work. A pressure interpolation method and variable bounding are used to “freeze” the errors in bad cells; this leads to the recovery of a good solution far from the interpolated cells.

5.2 Compressibility

The SIMPLEC algorithm was modified to accommodate a pressure correction convection term derived from the ideal gas law, and a false time step term for pressure correction was introduced to apply the finite volume method to compressible problems. The results obtained with the de Laval nozzle and the oblique shock case show good agreement with their analytic solutions.

The method works perfectly with a randomly deformed mesh for the oblique shock problem. This suggests that the code can be applied to industrial problems involving non-orthogonality.

The method is prone to numerical diffusion when reflected shocks are involved, as demonstrated by the channel bump case. Although the method works fine at subsonic speeds, the predicted shocks are smeared with an Mach number of 1.65, but the shocks become sharper at Mach 2.0. The code however does not predict the correct profile for transition cases in two dimensions.

The case of an oxygen jet entering a hot furnace was successfully modelled using the numerical method developed in this research. Modified turbulence models were also introduced, giving good agreement with experimental data from the literature.

5.3 Free surface modelling in the presence of compressibility

Finally, a modified turbulence model for cold jets and the numerical method for compressible flow were successfully applied to the case of a high-speed jet impinging on a free surface. A novel zonal GALA method has been introduced to handle pressure corrections in both the compressible region in the jet, and the incompressible region where liquid is significantly present.

The deformations predicted with a compressible jet are not very different from using an incompressible jet, when using the model in our report. This suggests that computational time can be saved in modelling such cases - making compressibility modelling redundant for blast furnace cases.

Chapter 6

FUTURE WORK

Further research is required to improve the numerical methods presented in this thesis and increase the range of problems which can be numerically modelled.

6.1 Turbulence and non-orthogonality

Invoking the diffusion non-orthogonality corrections lead to divergence when they are used in conjunction with the $k - \varepsilon$ turbulence model. This restricts the applicability of non-orthogonal problems modelled to low Reynolds numbers. This area requires further investigation.

6.2 Transonic flows

When the case described in Section 3.3.3.1 is run with an inlet Mach number of 0.675, the flow is expected to be transonic - starting at subsonic conditions at the inlet and resulting in a shock on the circular bump as shown in Figure 3.28 [37]. However, the current code fails to predict the shock and leads to a wrong velocity profile, as shown in Figure 3.29. in order to address this difficulty, two routes are possible:

1. make the code inviscid, necessitating a major re-write of the solver and boundary conditions handling, and

2. implementing more Total Variation Diminishing (TVD) schemes in the code to handle the shocks.

6.3 Mesh Adaptation for shocks and effect on non-orthogonality

Numerical diffusion plagues shock reflection on walls. A possible solution to this would be through the use of mesh adaptation techniques which are commonly used to capture shocks. These involve refining the mesh in shock regions, requiring additional terms in discretised equations to handle changes in volume. These mesh changes will certainly impact on the skewness of the problem, and trigger an interesting investigation on the applicability of the numerical methods presented in this thesis to these problems.

REFERENCES

[1] Yves Marcel Ahipo and Philippe Traore. **Nouvelle approche pour la discrtisation de flux diffusifs en volumes finis forte obliquit**. *Comptes Rendus Mcanique*, **334**(6):380 – 386, 2006. vii, 25, 27, 28, 29

[2] Yves Marcel Ahipo and Philippe Traore. **A robust iterative scheme for finite volume discretization of diffusive flux on highly skewed meshes**. *Journal of Computational and Applied Mathematics*, **231**(1):478 – 491, 2009. vii, 28

[3] S. Heinz. **A model for the reduction of the turbulent energy redistribution by compressibility**. *Physics of Fluids*, **15**(11):3580 – 3583, 2003. viii, 120

[4] Khaled S. Abdol-Hamid, S. Paul Pao, Steven J. Massey, and Alaa Elmiligui. **Temperature Corrected Turbulence Model for High Temperature Jet Flow**. *Journal of Fluids Engineering*, **126**(5):844–850, 2004. viii, 118, 119, 120, 121, 122

[5] M. Alam, J. Naser, and G.A. Brooks. **CFD simulation of supersonic oxygen jet behaviour inside a high temperature field**. In *Seventh International Conference on CFD in the Minerals and Process Industries*, Melbourne, Australia, December 2009. viii, 105, 118, 121, 122, 146, 150, 156, 157

[6] S. Ito and I. Muchi. **Transport phenomenon of supersonic jet in oxygen top blowing converter**. *Iron & Steel Inst Japan-J (Tets-To-Hagane)*, **55**:1152 – 1163, 1969. ix, xix, 166

[7] Pericleous, K., Djambazov, G., Lebon, B., Croft, T.-N., Domgin, J.-F., and Gardin, P. **Time-dependent modelling and experimental validation of the metal/flux interface in a continuous casting mould**. *Rev. Met. Paris*, (1):33–43, 2008. ix, 176, 177

[8] A.W. Date. **Solution of transport equations on unstructured meshes with cell-centered colocated variables. Part I: Discretization**. *International Journal of Heat and Mass Transfer*, **48**(6):1117 – 1127, 2005. xi, 31

[9] 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**(3):387 – 411, 1982. xiv, 60, 65, 66

[10] I. Demirdžić, Ž. Lilek, and M. Perić. **Fluid flow and heat transfer test problems for non-orthogonal grids: Bench-mark solution**. *International Journal for Numerical Methods in Fluids*, **15**(3):329–354, 1992. xiv, xv, xvii, 60, 71, 73, 78, 79, 80, 81, 97, 101, 102

[11] S. Bayyuk and S. Habchi. **Verification Test Case Example: Steady-state oblique shock wave**. Technical report, AIAA. xvii, 130, 131

[12] A. W. Date. **Solution of Navier-Stokes equations on nonstaggered grid at all speeds**. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, **33**(4):451 – 467, 1998. xviii, 106, 111, 136, 144, 147, 148

[13] F. Moukalled and M. Darwish. **A high-resolution pressure-based algorithm for fluid flow at all speeds**. *J. Comput. Phys.*, **168**:101–133, March 2001. xviii, 106, 115, 136, 144, 147, 148

[14] I. Sumi, Y. Kishimoto, Y. Kikuchi, and H. Igarashi. **Effect of high-temperature field on supersonic oxygen jet behaviour**. *ISIJ International*, **46**(9):1312 – 1317, 2006. xviii, 13, 146, 155, 157, 161

[15] T.N. Croft, K. Pericleous, and M. Cross. **PHYSICA: A multiphysics environment for complex flow processes**. *Numerical Methods in Laminar and Turbulent Flow '95*, **IX**:1269–1280, 1995. ISBN 0 906674 86 7. 1, 4, 6, 7, 9, 11, 29, 36, 109, 170, 171, 176

[16] P. M-Y. Chow. *Control volume unstructured mesh procedure for convection-diffusion solidification process.* PhD thesis, University of Greenwich, 1993. 1

[17] T.N. Croft. *Unstructured mesh-finite volume algorithms for swirling, turbulent, reacting flows.* PhD thesis, University of Greenwich, 1998. 1, 31, 32

[18] D. Wheeler. *Computational Modelling of Surface Tension Phenomena in Metals Processing.* PhD thesis, University of Greenwich, 2000. 1, 178

[19] D. McBride. *Vertex based discretisation of thermal-fluid flow phenomena.* PhD thesis, University of Greenwich, 2003. 1, 24

[20] Mehmet Shala. *Unstructured Staggered Mesh Discretisation Methods for Computational Fluid Dynamics.* PhD thesis, University of Greenwich, 2006. 1

[21] H. Wang. *Numerical modelling of the tilt casting process.* PhD thesis, University of Greenwich, 2008. 1, 176, 177

[22] W.P Jones and B.E Launder. **The prediction of laminarization with a two-equation model of turbulence**. *International Journal of Heat and Mass Transfer*, **15**(2):301 – 314, 1972. 3, 118, 119, 160

[23] B. E. Launder and D. B. Spalding. **The numerical computation of turbulent flows**. *Computer Methods in Applied Mechanics and Engineering*, **3**(2):269 – 289, 1974. 3, 118, 119, 160

[24] S. Patankar. *Numerical heat transfer and fluid flow.* Taylor & Francis, 1st edition, Jan 1980. 4, 10, 111

[25] H.K. Versteeg and W. Malalasekera. *Introduction to computational fluid dynamics: The finite volume method.* Pearson Prentice Hall, 1st edition, 1995. 4, 7, 118, 119, 152

[26] Richard Courant, Eugene Isaacson, and Mina Rees. **On the solution of nonlinear hyperbolic differential equations by finite differences**. *Communications on Pure and Applied Mathematics*, **5**(3):243–255, 1952. 6

[27] 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**(4):551–559, 1972. 6

[28] 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**(6):617–641, 1988. 7, 144, 150

[29] 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, November 1983. 7, 170, 210

[30] A. W. Date. **Fluid dynamical view of pressure checkerboarding problem and smoothing pressure correction on meshes with colocated variables**. *International Journal of Heat and Mass Transfer*, **46**(25):4885 – 4898, 2003. 7

[31] 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, 1972. 9, 24, 106

[32] J.P. Van Doormaal and G.D. Raithby. **Enhancements of the SIMPLE method for predicting incompressible fluid flows**. *Numerical Heat Transfer*, **7**:147–163, 1984. 10, 13, 25

[33] R. I. Issa. **Solution of the implicitly discretised fluid flow equations by operator-splitting**. *Journal of Computational Physics*, **62**(1):40 – 65, 1986. 10

[34] S. Acharya, B. R. Baliga, K. Karki, J. Y. Murthy, C. Prakash, and S. P. Vanka. **Pressure-Based Finite-Volume Methods in Computational Fluid Dynamics**. *Journal of Heat Transfer*, **129**(4):407–424, 2007. 10

[35] M. Perić. **Analysis of pressure-velocity coupling on nonorthogonal grids**. *Numerical Heat Transfer, Part B*, **17**:63 – 82, 1990. 10, 24

[36] K. A. Pericleous. **MATH1071 Lecture 8: Sources and Boundary Conditions**, 2010. University of Greenwich. 10, 11

[37] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer-Verlag, 3rd revised edition, 2002. 13, 27, 106, 108, 138, 140, 141, 142, 193

[38] Olivier-Pierre Jacquotte. **Grid Optimization Methods for Quality Improvement and Adaptation**. In Joe Thompson, Bharat Soni, and Nigel Weatherill, editors, *Handbook of Grid Generation*, pages **33**–1 – **33**–33. CRC Press, 1998. 17, 19, 23

[39] Tecplot, Inc., Bellevue, WA. *Tecplot 360 2010 User's Manual*, 2010. 18, 19

[40] C. Wayne Mastin. **Truncation Error on Structured Grids**. In Joe Thompson, Bharat Soni, and Nigel Weatherill, editors, *Handbook of Grid Generation*, pages **32**–1 – **32**–10. CRC Press, 1998. 19, 23

[41] Timothy J. Baker. **Mesh generation: Art or science?** *Progress in Aerospace Sciences*, **41**(1):29 – 63, 2005. 20, 22, 23

[42] Joe Thompson, Bharat Soni, and Nigel Weatherill. **Preface: An Elementary Introduction**. In Joe Thompson, Bharat Soni, and Nigel Weatherill, editors, *Handbook of Grid Generation*, pages ix – xxvi. CRC Press, 1998. 20

[43] E. W. Weisstein. **Conformal Mapping**, 2011. From MathWorld – A Wolfram Web Resource. 20

[44] C. C. L. Sells. **Plane Subcritical Flow Past a Lifting Aerofoil**. *Royal Society of London Proceedings Series A*, **308**:377–401, January 1969. 20

[45] K. C. Karki and S. V. Patankar. **Calculation procedure for viscous incompressible flows in complex geometries**. *Numerical Heat Transfer*, **14**:295–307, 1988. 20

[46] L. Davidson and P. Hedberg. **Mathematical derivation of a finite volume formulation for laminar flow in complex geometries**. *International Journal for Numerical Methods in Fluids*, **9**:531–540, May 1989. 20

[47] H. Xu and C. Zhang. **Study of the effect of the non-orthogonality for non-staggered grids - The theory**. *International Journal for Numerical Methods in Fluids*, **28**(9):1265–1280, 1998. 20

[48] A.J. Oliver and A.J. Miller. **A method for predicting 2-dimensional flows and heat transfer in complex geometries and its application to natural convection in a gas cooled cable**. In *Numer Methods Laminar Turbul Flow*, pages 740 – 749, 1983. 21

[49] W. Shyy, S. S. Tong, and S. M. Correa. **Numerical Recirculating Flow Calculation Using a Body-Fitted Coordinate System**. *Numerical Heat Transfer Part B - Fundamentals*, **8**:99–113, 1985. 21

[50] P. Coelho and J. C. F. Pereira. **Calculation procedure for 3-D laminar flows in complex geometrics using a nonstaggered nonorthogonal grid system**. *Applied Mathematical Modelling*, **17**(11):562 – 576, 1993. 21

[51] M. H. Kobayashi and J. C. F. Pereira. **Calculation of incompressible laminar flows on a nonstaggered, nonorthogonal grid**. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, **19**(2):243 – 262, 1991. 21

[52] P. He and M. Salcudean. **A numerical method for 3D viscous incompressible flows using non-orthogonal grids**. *International Journal for Numerical Methods in Fluids*, **18**(5):449–469, 1994. 21

[53] C. Moulinec and P. Wesseling. **Colocated schemes for the incompressible Navier-Stokes equations on non-smooth grids for two-dimensional problems**. *International Journal for Numerical Methods in Fluids*, **32**:349–364, February 2000. 22

[54] T. J. Barth, editor. *Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations*, May 1992. 22

[55] 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–+, July 1974. 22

[56] Joe F. Thompson and Nigel P. Weatherill. **Fundamental Concepts and Approaches**. In Joe Thompson, Bharat Soni, and Nigel Weatherill, editors, *Handbook of Grid Generation*, pages **1**–1 – **1**–30. CRC Press, 1998. 22

[57] A Jameson, T. J. Baker, and N. P. Weatherill. **Calculation of Inviscid Transonic Flow Over a Complete Aircraft**. In *AIAA Paper 86-0103*, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986. 22

[58] Hristo N. Djidjev. **Force-Directed Methods For Smoothing Unstructured Triangular And Tetrahedral Meshes**. In *IN PROCEEDINGS OF THE 9TH INTERNATIONAL MESHING ROUNDTABLE*, pages 395–406, 2000. 23

[59] 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, September 1983. 23

[60] M. Piller and E. Stalio. **Development of a mixed control volume - Finite element method for the advection-diffusion equation with spectral convergence**. *Computers & Fluids*, **40**(1):269 – 279, 2011. 23

[61] D. McBride, T.N. Croft, and M. Cross. **Combined vertex-basedcell-centred finite volume method for flows in complex geometries**. In *Third International Conference on CFD in the Minerals and Process Industries*, Melbourne, Australia, December 2003. 23, 24

[62] D. McBride, T.N. Croft, and M. Cross. **Finite volume method for the solution of flow on distorted meshes**. *International Journal of Numerical Methods for Heat & Fluid Flow*, **17**(2):213 – 239, 2007. 23

[63] D. McBride, T.N. Croft, and M. Cross. **A coupled finite volume method for the computational modelling of mould filling in very complex geometries**. *Computers & Fluids*, **37**(2):170 – 180, 2008. 24

[64] M. J. Cho and M. K. Chung. **New Treatment of Nonorthogonal Terms in the Pressure-Correction Equation**. *Numerical Heat Transfer Part B - Fundamentals*, **26**:133–145, September 1994. 25

[65] Y. Wang and S. Komori. **On the improvement of the SIMPLE-like method for flows with complex geometry**. *Heat and Mass Transfer*, **36**:71–78, 2000. 25

[66] S. Pimpalnerkar, M. Kulkarni, and A.W. Date. **Solution of transport equations on unstructured meshes with cell-centered colocated variables. Part II: Applications**. *International Journal of Heat and Mass Transfer*, **48**(6):1128 – 1136, 2005. 31

[67] Sharc. **Harpoon - The Extreme Mesher: Version 3.2**, 2008. 33, 60, 86

[68] R. Franke. **Scattered data interpolation: test of some methods**. *Mathematics of Computation*, **38**(157):181–200, 1982. 33

[69] D. Shepard. **A two-dimensional interpolation function for irregularly-spaced data**. In *Proceedings of the 1968 23rd ACM national conference*, 1968. 33

[70] T. Lehnhäuser and M. Schäfer. **Efficient discretization of pressure-correction equations on non-orthogonal grids**. *International Journal for Numerical Methods in Fluids*, **42**(2):211–231, 2003. 69

[71] S. R. Mathur and J. Y. Murthy. **A pressure-based method for unstructured meshes**. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, **31**(2):195 – 215, 1997. 69

[72] D. Ghosh Roychowdhury, Sarit Kumar Das, and T. Sundararajan. **An efficient solution method for incompressible N-S equations using non-orthogonal collocated grid**. *International Journal for Numerical Methods in Engineering*, **45**(6):741 – 763, 1999. 69

[73] Y.Y. Tsui and Y.F. Pan. **A pressure-correction method for incompressible flows using unstructured meshes**. *Numerical Heat Transfer, Part B*, **49**:43–65, 2006. 69

[74] Yukari Tago and Yoshihiko Higuchi. **Fluid flow analysis of jets from nozzles in top blown processes**. *ISIJ International*, **43**(2):209 – 215, 2003. 105, 118

[75] R. Sambasivam and F. Durst. **Characteristics of supersonic jets in LD steelmaking**. *Ironmaking & Steelmaking*, **37**:195–203(9), 2010. 105

[76] Morshed Alam, Jamal Naser, and Geoffrey Brooks. **Computational Fluid Dynamics Simulation of Supersonic Oxygen Jet Behavior at Steelmaking Temperature**. *Metallurgical and Materials Transactions B*, **41**:636–645, 2010. 10.1007/s11663-010-9341-0. 105, 118, 121

[77] J. R. Debonis and J. N. Scott. **Large-Eddy Simulation of a Turbulent Compressible Round Jet**. *AIAA Journal*, **40**:1346–1354, July 2002. 105, 118

[78] C. Hirsch. *Numerical Computation of Internal and External Flows Volume 1 Fundamentals of Computational Fluid Dynamics*. Butterworth-Heinemann, second edition, 2007. 106, 110

[79] **Gas Dynamics: The Riemann Problem and Discontinuous Solutions: Application to the Shock Tube Problem**. In Ionut Danaila, Pascal Joly, Sidi Mahmoud Kaber, and Marie Postel, editors, *An Introduction to Scientific Computing*, pages 213–233. Springer New York, 2007. 106

[80] I. Demirdžić, Ž. Lilek, and M. Perić. **A collocated finite volume method for predicting flows at all speeds**. *International Journal for Numerical Methods in Fluids*, **16**:1029–1050, 1993. 106, 108, 110, 111, 114, 125

[81] W. Y. Soh and John W. Goodrich. **Unsteady solution of incompressible Navier-Stokes equations**. *J. Comput. Phys.*, **79**:113–134, November 1988. 106

[82] Kiril S. Shterev and Stefan K. Stefanov. **Pressure based finite volume method for calculation of compressible viscous gas flows**. *J. Comput. Phys.*, **229**:461–480, January 2010. 106

[83] H. Schlighting. *Boundary-Layer Theory.* McGraw-Hill, seventh edition edition, 1979. 108

[84] Nicholas J. Georgiadis, James Iwan David Alexander, and Eli Reshotko. *Development of a Hybrid RANS/LES Method for Compressible Mixing Layer Simulations.* NASA Center for Aerospace Information, 2001. 118

[85] D. B. Spalding. *Combustion and mass transfer A textbook with multiple-choice exercises for engineering students.* Pergamon Press, Oxford, 1st edition, 1979. 118

[86] B. Allemand, P. Bruchet, C. Champinot, S. Melen, and F. Porzucek. **Theoretical and experimental study of supersonic oxygen jets. Industrial application in EAF**. *Rev. Met. Paris*, (6):571–587, 2001. 119, 146

[87] K. C. Schadow, E. Gutmark, and K. J. Wilson. **Compressible spreading rates of supersonic coaxial jets**. *Experiments in Fluids*, **10**:161–167, December 1990. 119

[88] Dimitri Papamoschou and Anatol Roshko. **The compressible turbulent shear layer: an experimental study**. *Journal of Fluid Mechanics*, **197**:453–477, 1988. 119

[89] S. B. Pope. **An explanation of the turbulent round-jet/plane-jet anomaly**. *AIAA Journal*, **16**:279–281, March 1978. 119, 120

[90] S. Sarkar, G. Erlebacher, M. Y. Hussaini, and H. O. Kreiss. **The analysis and modelling of dilatational terms in compressible turbulence**. *Journal of Fluid Mechanics*, **227**:473–493, 1991. 120

[91] S. Sarkar. **The stabilizing effect of compressibility in turbulent shear flow**. *Journal of Fluid Mechanics*, **282**:163–186, 1995. 120

[92] R. M. C. So and T. P. Sommer. **An explicit algebraic heat-flux model for the temperature field**. *International Journal of Heat and Mass Transfer*, **39**(3):455 – 465, 1996. 120

[93] Masoud Rokni and Thomas B. Gatski. **Predicting turbulent convective heat transfer in fully developed duct flows**. *International Journal of Heat and Fluid Flow*, **22**(4):381 – 392, 2001. 120

[94] K. Abe, T. Kondoh, and Y. Nagano. **A two-equation heat transfer model reflecting second-moment closures for wall and free turbulent flows**. *International Journal of Heat and Fluid Flow*, **17**(3):228 – 237, 1996. 120

[95] D. C. Wilcox. *Turbulence Modeling for CFD.* DCW Industries Inc., La Cañada, CA, 1993. 121

[96] Wikipedia. **De Laval nozzle — Wikipedia, The Free Encyclopedia**, 2010. [Online; accessed 27-July-2010]. 123

[97] Arthur. Rizzi, Henri. Viviand, and Gesellschaft fur Angewandte Mathematik und Mechanik. *Numerical methods for the computation of inviscid transonic flows with shock waves : a GAMM workshop / Arthur Rizzi, Henri Viviand, editors.* Vieweg, Braunschweig :, 1981. 134

[98] B.E. Launder, W.C. Reynolds, W. Rodi, J. Mathieu, and D. Jeandel. *Turbulence models and their applications*, **2**. Eyrolles, Paris, 1984. 151

[99] R. B. Banks and D. V. Chandrasekhara. **Experimental investigation of the penetration of a high-velocity gas jet through a liquid surface**. *Journal of Fluid Mechanics*, **15**:13–34, 1963. 168

[100] Frank R. Cheslak, J. Arthur Nicholls, and Martin Sichel. **Cavities formed on liquid surfaces by impinging gaseous jets**. *Journal of Fluid Mechanics*, **36**(01):55–63, 1969. 168

[101] Annie Nordquist, Nitesh Kumbhat, Lage Jonsson, and Pär Jönsson. **The effect of nozzle diameter, lance height and flow rate on penetration depth in a top-blown water model**. *Steel Research International*, **77, s. 82-90**, 2006. 168

[102] O. Olivares, A. Elias, R. Sanchez, M. Diaz-Cruz, and R. Morales. **Physical and mathematical models of gas-liquid fluid dynamics in LD converters**. *Steel Research*, **73**:44 – 51, 2002. 168

[103] Anh V. Nguyen and Geoffrey M. Evans. **Computational fluid dynamics modelling of gas jets impinging onto liquid pools**. *Applied Mathematical Modelling*, **30**(11):1472 – 1484, 2006. Selected papers from the Third International Conference on CFD in the Minerals and Process Industries - 3rd International Conference on CFD. 168, 169

[104] H. J. Odenthal, Jens Kempken, J. Schlüter, and Emling W. H. **Advantageous numerical simulation of the converter blowing process**. *Iron & Steel Technology*, **4**:71 – 89, 2007. 168

[105] Mikael Ersson, Anders Tilliander, Lage Jonsson, and Pär Jönsson. **A Mathematical Model of an Impinging Air Jet on a Water Surface**. *ISIJ International*, **48**(4):377–384, 2008. 168

[106] J. Solórzano-López, R. Zenit, and M.A. Ramírez-Argáez. **Mathematical and physical simulation of the interaction between a gas jet and a liquid free surface**. *Applied Mathematical Modelling*, **35**(10):4991 – 5005, 2011. 168

[107] C. W. Hirt and B. D. Nichols. **Volume of fluid (VOF) method for the dynamics of free boundaries**. *Journal of Computational Physics*, **39**(1):201 – 225, 1981. 168

[108] Francis H. Harlow and J. Eddie Welch. **Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface**. *Physics of Fluids*, **8**(12):2182 – 2189, 1965. 169

[109] K. A. Pericleous, K. S. Chan, and M. Cross. **Free surface flow and heat transfer in cavities: the SEA algorithm**. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, **27**(4):487 – 507, 1995. 169

[110] Bram Van Leer. **Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection**. *Journal of Computational Physics*, **23**(3):276 – 299, 1977. 173

[111] R. Courant, K. Friedrichs, and H. Lewy. **On the partial difference equations of mathematical physics**. *IBM J. Res. Dev.*, **11**:215–234, March 1967. 174

[112] John D. Ramshaw and John A. Trapp. **A numerical technique for low-speed homogeneous two-phase flow with sharp interfaces**. *Journal of Computational Physics*, **21**(4):438 – 453, 1976. 175

[113] Stanley Osher and James A. Sethian. **Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations**. *J. Comput. Phys.*, **79**:12–49, November 1988. 176

[114] Mark Sussman, Peter Smereka, and Stanley Osher. **A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow**. *Journal of Computational Physics*, **114**(1):146 – 159, 1994. 176

[115] Mark Sussman and Elbridge Gerry Puckett. **A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows**. *Journal of Computational Physics*, **162**(2):301 – 337, 2000. 176

[116] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. **A Hybrid Particle Level Set Method for Improved Interface Capturing**. *Journal of Computational Physics*, **183**(1):83 – 116, 2002. 176

[117] Marsha J Berger and Joseph Oliger. **Adaptive mesh refinement for hyperbolic partial differential equations**. *Journal of Computational Physics*, **53**(3):484 – 512, 1984. 176

[118] M.J. Berger and P. Colella. **Local adaptive mesh refinement for shock hydrodynamics**. *Journal of Computational Physics*, **82**(1):64 – 84, 1989. 176

[119] Frank Losasso, Ronald Fedkiw, and Stanley Osher. **Spatially adaptive techniques for level set methods and incompressible flow**. *Computers & Fluids*, **35**(10):995 – 1010, 2006. 176

[120] John Strain. **Tree Methods for Moving Interfaces**. *Journal of Computational Physics*, **151**(2):616 – 648, 1999. 176

[121] John Strain. **Fast Tree-Based Redistancing for Level Set Computations**. *Journal of Computational Physics*, **152**(2):664 – 686, 1999. 176

[122] J. C. Martin and W. J. Moyce. **Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane**. *Royal Society of London Philosophical Transactions Series A*, **244**:312–324, March 1952. 177

[123] H. Wang, G. Djambazov, K.A. Pericleous, R.A. Harding, and M. Wickins. **Modelling the dynamics of the tilt-casting process and the effect of the mould design on the casting quality**. *Computers & Fluids*, **42**(1):92 – 101, 2011. 177

[124] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, 1967. 177

Appendix A

Pressure correction algorithms in collocated code

A.1 SIMPLE

A pressure correction term p' is defined as the difference between the correct pressure field p and a guessed field p^* for the first iteration, and previous iteration values for subsequent iterations.

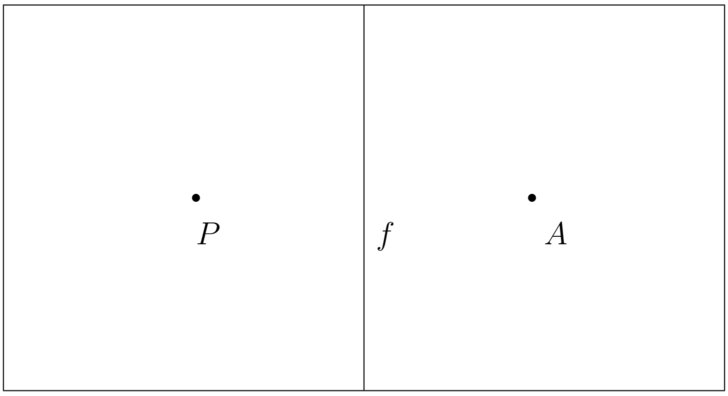

Figure A.1: Cell centres P and A, and face f.

$$p = p^{*} + p' \tag{A.1}$$

Similarly, velocity is expressed as the sum of two terms: a variable field not satisfying continuity, denoted by a $*$ superscript, and a correction term, denoted by $'$.

$$u = u^{*} + u' \tag{A.2}$$

A pressure field p^{*} is initially guessed and the discretised momentum equation for a collocated scheme is solved

$$a_P u_P^{*} = \sum_{neighbours} a_{nb} u_{nb}^{*} - \nabla p_P \cdot \boldsymbol{x} + b \tag{A.3}$$

A pressure correction equation is then solved. The correction values point pressure towards a mass conserving solution. This pressure correction equation is derived from the mass continuity equation.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{A.4}$$

Upon discretisation,

$$\frac{\rho_P V_P - \rho_P^{o} V_P^{o}}{\Delta t} + \sum_{f} A_f \left(\rho \boldsymbol{u} \cdot \hat{\boldsymbol{n}}\right)_f = 0 \tag{A.5}$$

In a collocated scheme, the face values are calculated using the Rhie-Chow interpolation method [29]

$$u_f = \overline{u_f} + \frac{1}{a_P}\left(\overline{\nabla p_f \cdot \boldsymbol{x}} - \nabla p_f \cdot \boldsymbol{x}\right) \tag{A.6}$$

Using the guessed pressure variables with superscript *,

$$u_f^{*} = \overline{u_f^{*}} + \frac{1}{a_P}\left(\overline{\nabla p_f^{*} \cdot \boldsymbol{x}} - \nabla p_f^{*} \cdot \boldsymbol{x}\right) \tag{A.7}$$

Subtracting (A.7) from (A.6), and using equation (A.2),

$$u_f' = \overline{u_f'} + \frac{1}{a_P}\left(\overline{\nabla p_f' \cdot \boldsymbol{x}} - \nabla p_f' \cdot \boldsymbol{x}\right) \tag{A.8}$$

The main approximation in SIMPLE is to drop the first two term in (A.8), and use

$$\begin{aligned} u'_f &= -\frac{1}{a_P}\left(\nabla p'_f \cdot \boldsymbol{x}\right) \\ &= \frac{1}{a_P} A_f n_x \left(p'_P - p'_A\right) \end{aligned} \tag{A.9}$$

Substituting equation (A.9) into discretised equation (A.5), the pressure correction equation is obtained

$$\sum_f \rho_f A_f^2 \frac{n_i^2}{a_i}\left(p'_P - p'_A\right)_f = \frac{\rho_P^o V_P^o - \rho_P V_P}{\Delta t} - \sum_f A_f \left(\rho \boldsymbol{u}^* \cdot \hat{\boldsymbol{n}}\right)_f = 0 \tag{A.10}$$

The subscript $_i$ indicates a summation over the coordinate directions. The equation can be written in the linear form (1.10)

$$a_P p'_P = \sum_{neighbours} a_{nb} p'_{nb} + b_{p'} \tag{A.11}$$

Solving a system of equations (A.11) gives the pressure correction values from which pressure can be adjusted using (A.1). However, the velocity corrections cannot be calculated from (A.9) in a collocated scheme. Instead, using the correct discretised nodal velocity field

$$a_P u_P = \sum_{neighbours} a_{nb} u_{nb} - \nabla_x p_P + b, \tag{A.12}$$

and subtracting equation (A.3) from (A.12), and using equation (A.2), we obtain

$$a_P u'_P = \sum_{neighbours} a_{nb} u'_{nb} - \nabla_x p'_P \tag{A.13}$$

The summation term is dropped and the velocity correction terms are obtained as

$$u'_P = -\frac{1}{a_P} \nabla_x p'_P \tag{A.14}$$

A.2 SIMPLEC

The derivation of the SIMPLEC algorithm is similar to that of SIMPLE, except that $\sum\limits_{neighbours} a_{nb}u'_P$ is subtracted from both sides of equation (A.13), leading to

$$u'_P = -\frac{1}{a_P - \sum\limits_{neighbours} a_{nb}} \nabla_x p'_P \tag{A.15}$$

Appendix B

Derivation of de Laval nozzle area ratio

The mass flow rate through a nozzle cross sectional area is given by

$$\dot{m} = \rho \boldsymbol{u} \cdot (A\hat{\boldsymbol{n}}) \tag{B.1}$$

For a 1D case where velocity is perpendicular to cross section

$$\dot{m} = \rho u A \tag{B.2}$$

The density of the gas is related to pressure and temperature by the ideal gas equation (3.19),

$$\rho = \frac{p}{R'T} \tag{B.3}$$

The velocity of the gas can be expressed in terms of Mach number

$$u = M \cdot a \tag{B.4}$$

where a is the speed of sound as defined in equation (3.2).

Replacing equations (B.4) and (B.3) into (B.2), we obtain

$$\begin{aligned} \dot{m} &= \frac{p}{R'T} M \cdot \sqrt{\gamma R' T} A \\ &= A M \sqrt{\gamma R'} \frac{p}{\sqrt{T}} \end{aligned} \tag{B.5}$$

Expressing pressure p and temperature T in terms of total variables using equation (3.16) and

$$T_t = T \left(1 + \frac{\gamma - 1}{2} M^2 \right), \tag{B.6}$$

we obtain

$$\dot{m} = A M \sqrt{\gamma R'} \frac{p_t \left(1 + \frac{\gamma-1}{2} M^2\right)^{-\frac{\gamma}{\gamma-1}}}{\sqrt{T_t} \left(1 + \frac{\gamma-1}{2} M^2\right)^{-\frac{1}{2}}} \tag{B.7}$$

At the nozzle throat A_{th}, the Mach number is 1.0

$$\dot{m} = A_{th} \sqrt{\gamma R'} \frac{p_t \left(1 + \frac{\gamma-1}{2}\right)^{-\frac{\gamma}{\gamma-1}}}{\sqrt{T_t} \left(1 + \frac{\gamma-1}{2}\right)^{-\frac{1}{2}}} \tag{B.8}$$

Dividing equation (B.7) by (B.8), the nozzle area ratio is obtained:

$$\frac{A}{A_{th}} = \left(\frac{\gamma + 1}{2} \right)^{-\frac{\gamma+1}{2(\gamma-1)}} \frac{\left(1 + \frac{\gamma-1}{2} M^2\right)^{\frac{\gamma+1}{2(\gamma-1)}}}{M} \tag{B.9}$$