A study of the characteristic based split scheme (CBS), considering the incorporation of 'speed up' techniques.

Hickey, Raymond

Swansea University

A study of the Characteristic Based Split scheme (CBS), considering the incorporation of 'speed up' techniques.

By
Ray Hickey

Supervisor
Prof. P. Nithiarasu

School of Engineering
2008

This thesis is submitted in partial requirement for the degree of
Doctor of Philosophy

Acknowledgements

I would like to sincerely express my gratitude to Prof. P. Nithiarasu, a man of infinite patience, for all his kind help, support and guidance throughout this project. It is a pleasure to know and have had as a supervisor such a kind man as he.

I am grateful for the provision of both three dimensional meshes and mesh generation software by Prof. O. Hassan and Dr. J.W. Jones. I would also like to thank the acknowledged for many conversations relating to the above.

I would like to thank all other members of staff and fellow research students for their helpful discussions and contributions.

This work received some funding from NASA, Engineering Computational and E.S.P.R.C

Lastly but not least I would like to thank my family and friends for all the love and support I received during my time in Swansea. A phone call in troubled times can make the world of difference.

Summary

The scope of this dissertation centers on the application of the Characteristic Based Split (CBS) algorithm to the governing equations of compressible flow, represented by the conservation form of the Navier Stokes equations.

It is proposed that the inherent stability of split algorithm alongside the explicit nature of the CBS scheme utilised in this dissertation will produce accurate results. To study the robustness and accuracy of the CBS scheme, various unstructured and hybrid meshes have been used. The method has been extended to three dimensions and the code parallelised to accelerate the solution. In addition to laminar compressible flows, at various Mach numbers, the code has been extended to incorporate RANS based turbulence modeling.

The performance of the CBS scheme is established by studying a variety of flow problems. The problems considered include test cases of an inviscid, viscous and turbulent nature for flows past both two dimensional and three dimensional solid geometries. Computed results are compared to those published in literature.

In addition the use and benefits of artificial diffusion, shock capturing and structured viscous layers are discussed at relevant points of this dissertation.

Declaration and Statements

Declaration

This work has not been previously accepted in substance for any degree and is not being currently submitted in candidature for any degree.

Signed_____.(Candidate)
Date_____29/9/18_____.

Statement 1

This dissertation is being submitted in partial fulfilment of the requirements for the degree of doctor of philosophy.

Signed_____.(Candidate)
Date_____29/9/18_____.

Statement 2

This dissertation is the result of my own independent work/investigation, excepts where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. References are appended.

Signed_____.(Candidate)
Date_____29/9/18_____.

Statement 3

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed_____.(Candidate)
Date_____29/9/18_____.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

1.1 General background

Computational Fluid Dynamics like other computational fields, reduces the continuum form of equations into discrete forms [1]. The fluid flow equations are discretised into a number of algebraic equations or ordinary differential equations which in turn are solved by setting the necessary initial and boundary conditions.

The majority of fluid flow formulations are based upon the Navier Stokes equations, and such equations are difficult to solve with analytical methods without recourse to various assumptions and approximations [2]. In order to convert the Navier Stokes equations into a form solvable by computational methods Galerkin procedures are often applied to the governing equations.

The standard Galerkin forms give symmetric matrices for self-adjoint diffusion dominated problems and give asymmetric matrices for the non self-adjoint convection terms. The non self-adjoint terms can also develop instability in the form of oscillations for convection dominated problems if the standard Galerkin method is used[3].

To overcome such instability, many types of stabilisation procedures have been suggested, including the Taylor Galerkin [4] method, the streamline upwind Petrov Galerkin method[5], Galerkin least squares [6] method and characteristic Galerkin method[7]. Artificial diffusion is added by the schemes in order to counteract the instability present.

Alongside the algorithm used for the Navier Stokes equations, the choice of

spatial and temporal discretisation procedures are also important in obtaining an accurate solution [8]. The finite element method is used in this dissertation to spatially discretise the the conservation variables.

The Characteristic Galerkin method[9] considers the movement of flow along the characteristics of the problem leading to self adjoint form of the momentum equation, discussed in more detail in chapter three. An approximate integration backwards leads to extra second order terms. The Characteristic Based Split or CBS scheme is described in detail in the third chapter. CBS was originally developed from work by Zienkiewicz and Codina[10] and it allows a matrix free solution of the Navier Stokes equations. This multi-step scheme operates by firstly removing the pressure terms from the momentum equation to obtain the resultant intermediate momentum variables using a lumped mass matrix to ensure an explicit solution. The pressure values are then calculated from density variables found from the intermediate momentum variables. The intermediate momentum variables are then updated using these new found pressure values.

The characteristic Galerkin form of the momentum equations stabilise the solution and the combination of both the pressure split with the characteristic Galerkin is the main attribute of the characteristic based split algorithm [11, 12].

Should the pressure term be completely removed from the momentum equation then a first order error is introduced into the momentum equation. Alternatively the pressure term may instead be treated as a source quantity and remain in the momentum equation. The Ladyzhenskaya-Babuska-Brezzi' (LBB) condition concerning zero diagonal terms must then be explicitly satisfied by other means[13]. The LBB condition is applied at near incompressibility as instability becomes unavoidable should the same interpolation functions be used for both the pressure and velocity variables [4]. The LLB condition is circumvented by use of a full 'split' of pressure from the momentum equation thus avoiding the formulating of a matrix. The split was first introduced for incompressible fluids[14] but later extended to compressible flows for both the non conservation form and then the conservation form of the Navier Stokes equations [15][16].

Turbulence modelling is a key area in modern CFD. Due to the complex nature of turbulence, obtaining a numerical model that approximates the physical behaviour of turbulent flows is a difficult task and has led to the establishment of

the sub-field of turbulence modelling within the CFD community.

Turbulence overview

Turbulence, is three dimensional and time dependent by nature, and thus extremely expensive to model to the lowest length scales. Therefore, to obtain a solution, models using time averaged equations have been formulated. These models are referred to as the Reynolds Averaged Navier Stokes R.A.N.S. models. Other more expensive solution methods include the Direct Numerical Solution method (D.N.S.) and the Large Eddy Simulation (L.E.S.) [17]. The (R.A.N.S.) method discussed in this dissertation consists of assuming time averaged velocities in place of the fluctuating velocity components[18].

In 1877, Boussinesq developed a mathematical description of the turbulent stresses in order to mimic the molecular gradient-diffusion process by introducing the concept of a so-called eddy viscosity. The Boussinesq eddy-viscosity approximation is the basis of the algebraic or zero equation models such as the Cebeci and Smith model[19] and the Baldwin and Lomax model[20].

Further progress was made when Prandtl discovered the existence of the boundary layer and extending this discovery to model turbulent flow. Prandtl introduced the concept of a mixing length describing the simulation of the mean free path of a gas [21]. Prandtl then proposed an equation for computing the eddy viscosity in terms of the mixing length. Later Prandtl proposed a new kind of turbulence model in which the eddy viscosity is related to the turbulent kinetic energy or the kinetic energy of the turbulent fluctuations. The examples of the one equation models includes the Spalart and Allmaras model[22] and the Baldwin Barth model [23].

Kolmogorov proposed what is known as the first complete model for the modelling of turbulence. In addition to having a equation for modelling the kinetic turbulence energy κ, Kolmogorov introduced a second parameter into his turbulence model that he referred to as ω 'the rate of dissipation of energy in unit volume and time.' Although Prandtl's model provided more physically realistic results by bonding eddy viscosity to the turbulent kinetic energy, it neglected the turbulent length scale [24]. The turbulent length scale can be thought of the characteristic

eddy size that deviates for different flows although introduction of another variable leads to an increase in computational cost. Models that do not provide a length scale are contrastingly termed 'incomplete'. In Kolmogorv's model known as k-ω model is the basis of the two equation turbulence models, examples include the $\kappa\epsilon$ model of Jones and Launder and the $\kappa\omega$ model of Saffman and Wilcox [25][26].

In this thesis, only the Spalart-Allmaras model is considered.

1.2 Research Aim

The objective of this thesis is twofold. Firstly, to study the performance of the Finite Element Method (F.E.M.) based Characteristic Based Split (C.B.S.) scheme with regards to both inviscid and viscous flow cases by comparing the accuracy of the scheme to results obtained by other authors.

Secondly, it will be ascertained whether the CBS can successfully incorporate the adoption of the speed up techniques of parallelisation and the edge based formulation technique which is currently a highly desired attribute of modern industry

1.3 Thesis outline

A brief outline description of the forthcoming chapters is given here.

The second chapter of this dissertation introduces the governing equations of fluid flow namely the compressible form of the Navier Stokes equations and the initial and boundary equations. The non dimensionalisation of the Navier Stokes equations and the non dimensional parameters obtained are also included in this chapter.

The FEM formulation and theory of the Characteristic Based Scheme algorithm, artificial damping and residual smoothing are introduced in chapter three.

The purpose of the short fourth chapter is to give the reader an understanding of the manner in which the computer code is written.

Chapter five examines acceleration techniques. This chapter contains information on the adoption and performance of both parallelization and the incorporation of the edge based data structure.

Inviscid flow problems are covered in chapter six which contains results obtained by the CBS scheme and the comparisons with corresponding published results.

The laminar viscous results obtained by the CBS scheme and the corresponding discussions are provided in chapter seven.

Chapter eight contains the results obtained from a turbulent flow cases utilising the one equation Spalart Allmaras model.

The conclusions, bibliography and appendixes are then included.

Chapter 2

Governing equations

2.1 The Navier Stokes equations

In recent years, considerable progress has been made in numerical modeling of fluid flow. Further research continues in the CFD field to improve both the efficiency and the accuracy of numerical schemes. This thesis discusses one such CFD scheme, the Characteristic Based Split (CBS) scheme [4] that models compressible flows and attempts to improve both the efficiency and the accuracy of this scheme.

The CBS scheme, like the majority of the other flow solvers developed in the past, is based on either the Navier Stokes or the Euler equations. The Euler equations are used to model inviscid flow problems while the Navier Stoke equations are used to model the viscous flow problems, the difference between these flow cases being the presence or absence of the diffusion flux vector [27, 28] which is discussed in more detail later in this chapter.

The Navier Stokes equations for calculating compressible fluid flow may be written in general conservative form[29] as.

$$\frac{\partial \mathbf{R}}{\partial t} + \frac{\partial \mathbf{F}_j}{\partial x_j} + \frac{\partial \mathbf{G}_j}{\partial x_j} + \mathbf{Q} = 0 \tag{2.1}$$

The notation is as follows, $\mathbf{R}$ represents the vector of conservation variables, $\mathbf{F}$ represents the convective or Euler flux vector, $\mathbf{G}$ represents the diffusion flux vector and $\mathbf{Q}$ represents the source vector.

These individual vector may be expressed in indicial notation as follows

$$\mathbf{R} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{pmatrix} \quad , \quad \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_1 u_j + p\delta_{1j} \\ \rho u_2 u_j + p\delta_{2j} \\ \rho u_3 u_j + p\delta_{3j} \\ \rho H u_j \end{pmatrix}$$

$$\mathbf{G}_j = \begin{pmatrix} 0 \\ -\tau_{1j} \\ -\tau_{2j} \\ -\tau_{3j} \\ -\tau_{ji}u_i - k\frac{\partial T}{\partial x_j} \end{pmatrix} \quad , \quad \mathbf{Q} = \begin{pmatrix} 0 \\ \rho g_1 \\ \rho g_2 \\ \rho g_3 \\ \rho g_j u_j - q_H \end{pmatrix}$$

The term ρ denotes the density, x_j are the spatial coordinate directions, u_j are the velocity components in these directions, p is the pressure, E is the total energy per unit mass, H is the enthalpy, k is the thermal conductivity , T is the temperature and g is the acceleration due to gravity. The Kronecker delta is denoted by δ_{ij}.

The source vector $\mathbf{Q}$ consists of changes to the conservation of the flow due to the motion and the work done by the body forces present from gravity, while the q_H term represents any energy that is generated or consumed by any chemical reaction that is present.

The enthalpy H may be written in terms of pressure, density and E, the total energy per unit mass; in addition the total energy E can be in turn be written in terms of the intrinsic energy per unit mass denoted by a lowercase e and the velocity.

$$H = E + \frac{p}{\rho} \quad , \quad E = e + \frac{1}{2}u_j u_j$$

The viscous stress τ_{ij} in the diffusive flux vector is defined by the following equation assuming that the fluid is a Newtonian fluid.

$$\tau_{ji} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{2.2}$$

The unknown μ is the dynamic viscosity and is calculated using the Sutherland's relation given below.

$$\mu = \mu_0 \frac{T_0 + C}{T + C} \left(\frac{T_0}{T}\right)^{\frac{3}{2}}$$

The terms μ_0, T_0 and C are all known constants for the specific fluid under consideration. The values of the constants for air are $1.716 * 10^{-5}$, 273.15 and $1.458 * 10^{-6}$ respectively.

The particular case of the fluid dynamic equations formed known as the Euler equations assumes that the flow is both non-viscous and not heat conducting, thus the τ_{ij} term and the heat flux term have a zero contribution which removes the diffusion flux vector $\mathbf{G}$ from the equation completely.

A more simplified Euler flow may also assume that there is no influence from neither the body forces nor from any chemical reaction present in the fluid. This further simplifies the Euler equation by also removing the source term vector $\mathbf{Q}$ from the equation, thus reducing Equation (2.1) to the following simplified form.

$$\frac{\partial \mathbf{R}}{\partial t} + \frac{\partial \mathbf{F_j}}{\partial x_j} = 0 \tag{2.3}$$

The temperature T can be written in terms of the internal energy e or alternatively in terms of the density and pressure values by using the ratio of specific heat at constant pressure c_p to the specific heat at constant volume c_v.

$$e = c_v T = \frac{1}{\gamma - 1} \frac{p}{\rho}$$

$$h = e + \frac{p}{\rho} = c_p T = \frac{\gamma}{\gamma - 1} \frac{p}{\rho}$$

$$\gamma = \frac{c_p}{c_v}$$

Note that the above terms can be written in terms of the total energy E by simply including an additional $\frac{1}{2} u_j^2$ term.

As is evident from the Navier Stokes equations, there are a total of six unknowns to be determined which are(P, ρ, u_{1-3}, E). In order to close the system, a sixth equation is therefore needed. The sixth equation used here is the 'ideal gas law'

given by.

$$\mathrm{p} = \rho \mathrm{RT}$$

The coefficient R is the ideal gas constant which may be written in terms of the two specific heats.

$$R = c_p - c_v \quad \text{or} \quad c_p = \frac{\gamma}{\gamma - 1} R$$

The Navier Stokes equations are also commonly rewritten in terms of the individual constituent equations, which are listed below.
The conservation of mass equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{2.4}$$

The conservation of momentum equation

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j u_i) + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i = 0 \tag{2.5}$$

The conservation of energy equation

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j H) - \frac{\partial}{\partial x_j}\left(k \frac{\partial T}{\partial x_j}\right) - \frac{\partial (\tau_{ji} u_i)}{\partial x_j} + \rho g_j u_j - q_H = 0 \tag{2.6}$$

This form of the equations shall be non-dimensionalised before being used in chapter three as the starting point for formulation of the CBS algorithm.

2.2 Non dimensional form

The Navier Stokes equations are non dimensionalised in order to reduce the number of parameters. The non dimensional form allows the unknown quantities to be written as a ratio of the unknown values to a reference value. Let the velocity be taken as an example below.

$$u_j^* = \frac{u_j}{u_\infty}$$

The term u_∞ denotes the free stream velocity value and u_j^* denotes the non dimensional ratio.

Substituting the above velocity relation into equations (2.4), (2.5) and (2.6)

along with the other non dimensional scaling ratios (which are all provided in Appendix A should the reader wish to refer to them), the non dimensional form of the governing equations are obtained as.

$$\frac{\partial \rho^*}{\partial t^*} + \frac{\partial}{\partial x_j^*}\left(\rho^* u_j^*\right) = 0 \tag{2.7}$$

$$\frac{\partial \left(\rho^* u_i^*\right)}{\partial t^*} + \frac{\partial}{\partial x_j^*}\left(\rho u_j^* u_i^*\right) + \frac{\partial p^*}{\partial x_i^*} - \frac{1}{R_e}\frac{\partial \tau_{ij}^*}{\partial x_j^*} + \rho^* g_i^* = 0 \tag{2.8}$$

$$\frac{\partial \left(\rho^* E^*\right)}{\partial t^*} + \frac{\partial}{\partial x_j^*}\left(\rho u_j^* H^*\right) - \frac{1}{P_r R_e}\frac{\partial}{\partial x_j^*}\left(k^*\frac{\partial T^*}{\partial x_j^*}\right) - \frac{1}{R_e}\frac{\partial \left(\tau_{ji}^* u_i^*\right)}{\partial x_j^*} + \rho^* g_j^* u_j^* - q_H^* = 0 \tag{2.9}$$

The superscript * denotes a non dimensional quantity. The non dimensional viscous deviatoric stress is similarly given by.

$$\tau_{ji}^* = \mu^*\left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} - \frac{2}{3}\frac{\partial u_k^*}{\partial x_k^*}\delta_{ij}\right) \tag{2.10}$$

The non dimensional dynamic viscosity is denoted by the term μ^*.

The two new non dimensional parameters appearing in the above forms of the momentum and energy equations are the Reynolds number R_e and the Prandtl number P_r which are defined as.

$$\begin{aligned} R_e &= \frac{u_\infty L}{\nu_\infty} \\ P_r &= \frac{c_p\,\mu_\infty}{k_\infty} \end{aligned}$$

It is evident that as the Reynolds number increases, the contributions from both the deviatoric stress terms and the conductive heat term in equations (2.5) and (2.6) decrease. This dissertation assumes, that when both of the above terms become negligibly small, an Euler flow is formed. The value of the Reynolds number assumed in this thesis to trigger the reduction to Euler flow is 10^8.

Another initial non dimensionalised parameter called the Mach number M is introduced at this stage which is defined below. Should the value of the Mach number be less than one, then the flow is subsonic and contrastively should the Mach number be greater than one then the flow is supersonic. The value of the

Mach number at free stream is defined as.

$$M_\infty = \frac{u_\infty}{c_\infty}$$

The term c_∞ is the speed of sound through the medium at the free stream; should constant entropy be assumed in the absence of any shock behaviour, the velocity of sound may be defined by

$$c^2 = \left(\frac{\partial p}{\partial \rho}\right)_s = \gamma RT = \frac{\gamma p}{\rho} \tag{2.11}$$

Note that both the non dimensional velocity and density are equal to unity in free stream.

2.3 Boundary and Initial conditions

The application of the boundary conditions will now be considered. Firstly, the components of the velocity are prescribed as the free stream values at the inlet boundaries. In the case of a supersonic flow, the density is also prescribed as the free stream density value at the inlet while in cases where subsonic flows are considered, the density is instead prescribed as the free stream density value at the exit boundary due to the different characteristic conditions between these two flow types.

The symmetric surface boundary condition consists of calculating the magnitude of the velocity vector and redistributing it in the tangential directions to the boundary face while the component of velocity normal to the boundary face is prescribed as zero.

The boundary condition for energy is applied at the inlet boundary, this value for energy at the inlet boundary is evaluated from these two relations where c is calculated from the Mach number.

$$E = c_v T + \frac{1}{2}u_i^2$$

$$T = \frac{p}{\rho R} = \frac{c^2}{\gamma R}$$

Substitution gives:

$$E = \frac{c^2}{\gamma(\gamma-1)} + \frac{1}{2}u_i^2$$

The speed of sound at inlet (free stream) is calculated from the free stream Mach number.

Lastly, the no-slip condition is applied at the boundary surfaces that represent solid geometries in the case of viscous flow. The no-slip condition involves the velocity components being prescribed as zero at the nodes on the solid surface. In the case of an Euler flow problem, the slip condition is applied where only the normal of the velocity is adjusted in the same manner as for the symmetric conditions.

The initial conditions are applied by prescribing the free stream values for the density, energy and the velocity components throughout the domain. These prescribed initial values are then used to calculate the initial values for pressure viscosity and temperature.

2.4 Chapter Summary

The Navier Stokes and Euler equations for fluid flow have been presented in this chapter. The non dimensional form of the governing equations as well as the boundary conditions are also included in this chapter.

The conservation form of the Navier Stokes equations were chosen in this dissertation due to the unsuitability of the non-conservation form of the equations in dealing with the prediction of shocks. The treatment of shocks is discussed in more detail in the next chapter.

The characteristic Based Split (CBS) scheme is employed in this thesis. In the following chapter the CBS scheme and it's advantages over other competing methods are discussed in more detail.

Chapter 3

The CBS Scheme and turbulence modelling

3.1 Characteristic Galerkin Method (CG)

Before the algorithms used in this dissertation are explained in depth, the physical theory behind the treatment of the Navier Stokes equations shall be discussed in more detail.

The Characteristic Galerkin or (CG) methods are based on the wave nature of the Navier Stokes equations and the use of wave characteristics[7]. The characteristic Galerkin method will be demonstrated by applying it to the simple one dimensional convection diffusion equation[30].

$$\frac{\partial R}{\partial t} + u\frac{\partial R}{\partial x} - \frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right) = 0 \tag{3.1}$$

Applying this equation to a fluid moving at a speed equal to the velocity u and assuming that the velocity is in a direction that is the same as the path of the flow's characteristic wave then the convection term of Equation (3.1) disappears.

Although this approach adds the complexity of a moving coordinate system denoted here by the supercript $'$, the approach has the advantage of eliminating the convective term leaving the equation purely diffusive in nature, in the process removing any convective spatial oscillations that may have been present. Should x be the current position of the moving coordinate and Δt denotes the time step,

Equation (3.1) may be rewritten as.

$$\frac{R^{n+1}|_{x_1} - R^n|_{x-\Delta x}}{\Delta t} = \theta \frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)^{n+1} + (1-\theta)\frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)^{n}|_{x-\Delta x} \tag{3.2}$$

Let a value of θ equal to 0.5 be assumed.

$$\frac{R^{n+1}|_{x_1} - R^n|_{x-\Delta x}}{\Delta t} = \frac{1}{2}\frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)^{n+1} + \frac{1}{2}\frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)^{n}|_{x-\Delta x} \tag{3.3}$$

Applying Taylor expansions to the both the $x - \Delta x$ terms denoting the previous spatial position results in the following equations. Note that the resultant third and higher order terms are neglected.

$$R^n|_{x-\Delta x} = R^n - \Delta x\frac{\partial R^n}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 R^n}{\partial x^2} + ... \tag{3.4}$$

$$\frac{1}{2}\frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)^{n}|_{x-\Delta x} = \frac{1}{2}\frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)^{n} - \frac{\Delta x}{2}\frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)\right]^{n} + ... \tag{3.5}$$

However Δx may be written as

$$\Delta x = \hat{u}\Delta t \tag{3.6}$$

The term $\hat{u}$ is the average velocity along the characteristic in the time Δt. $\hat{u}$ may be approximated in terms of u^{n+1} and u^n with the following relation

$$\hat{u} = \frac{u^{n+1} + u^n|_{x-\Delta x}}{2} \tag{3.7}$$

The term $u^n|_{x-\Delta x}$ also undergoes a Taylor expansion.

$$u^n|_{x-\Delta x} = u^n - \Delta x\frac{\partial u^n}{\partial x} + ... \tag{3.8}$$

Equations (3.4 - 3.8) are now substituted into Equation (3.3) and the third and higher order terms are again neglected. The resultant equation obtained is

$$\Delta R = R^{n+1} - R^n = \; - \; \Delta t\left[u^{n+\frac{1}{2}}\frac{\partial R^n}{\partial x} - \frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right)^{n+\frac{1}{2}}\right]$$

$$+ \quad \frac{\Delta t^2}{2} u^{n+\frac{1}{2}} \frac{\partial}{\partial x} \left[u^{n+\frac{1}{2}} \frac{\partial R^n}{\partial x} - \frac{\partial}{\partial x} \left(k \frac{\partial R}{\partial x} \right)^n \right] \tag{3.9}$$

The $n + \frac{1}{2}$ terms are approximated in the above equation as

$$\begin{aligned} \frac{\partial}{\partial x} \left(k \frac{\partial R}{\partial x} \right)^{n+\frac{1}{2}} &= \frac{1}{2} \frac{\partial}{\partial x} \left(k \frac{\partial R}{\partial x} \right)^{n+1} + \frac{1}{2} \frac{\partial}{\partial x} \left(k \frac{\partial R}{\partial x} \right)^{n} \\ u^{n+\frac{1}{2}} &= \frac{u^{n+1} + u^n}{2} \end{aligned} \tag{3.10}$$

The advantage of this extra second order term is that it both increases the time accuracy and also acts as an extra diffusion term which stabilises the solution. Should the terms existing at the time $n + \frac{1}{2}$ be approximated to their values at time n then the equation becomes explicit and may be extended to the three dimensional explicit form shown here.

$$\begin{aligned} \Delta R = R^{n+1} - R^n = \quad & - \quad \Delta t \left[\frac{\partial (u_j R)}{\partial x_j} - \frac{\partial}{\partial x_i} \left(k \frac{\partial R}{\partial x_i} \right) \right]^n \\ & + \quad \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k} \left[\frac{\partial (u_j R)}{\partial x_j} - \frac{\partial}{\partial x_i} \left(k \frac{\partial R}{\partial x_i} \right) \right]^n \end{aligned} \tag{3.11}$$

This approximation to obtain Equation (3.11) shall be used in the following section.

3.2 The Characteristic Based Split Scheme

To solve and accurately model the realistic problems that simulate the flow past very complex geometries of interest to the aerospace industry, three-dimensional unstructured grids must be used in the discretisation of the flow domains to achieve this goal in a cost effective manner. In achieving this goal, a dedicated and flexible unstructured mesh based solver will be needed, that can be applied to both unstructured meshes as well as the new types of automatically generated and adaptive meshes that current and future advances in mesh generation will produce.

The Characteristic Based Split or CBS scheme described in this section is a general scheme suitable for the solution of these realistic problems, the split procedure itself was initially developed by Chorin et al[14][31] about 35 years ago, however the actual application of the CBS algorithm to the finite element based

method considering compressible flow problems has been a much more recent occurrence. The first attempt to the author's knowledge in using the split procedure with the finite element method for the modeling of compressible flows was made by Zienkiewicz and Wu [32], in further research Zienkiewicz et al[33] developed the unified algorithm that could solve both compressible and incompressible flow problems in both shallow water simulation, solid dynamic computational models[34] as well as the more traditional aerodynamic problems. Many further improvements have been made since then to the CBS scheme, most recently in these cited papers [35][36][37].

The CBS method as mentioned is applicable to a wide scope of fluid dynamics models relative to aeronautics ranging from incompressible flow to compressible flow problems with flow speeds of subsonic to hypersonic flow. Although each step of the CBS method will be discussed in more detail later, a brief summary of these steps is given here.[27].

There are four main steps that make up the content of the explicit CBS algorithm, these may be described as follows

1. In the first step, the pressure terms are removed from the momentum equation. The resultant pressureless equation is then solved to calculate the temporary intermediate momentum conservation variables using a lumped mass matrix to ensure an explicit solution.

2. In the second step, the pressure values are calculated from density variables that were themselves found from the conservation of mass equation which incorporated the use of the intermediate momentum conservation variables that calculated from the previous step.

3. In the third step of the CBS scheme, the momentum variables are updated using these pressure values calculated from the previous step two.

4. Finally in the fourth step, the energy values are updated using the energy equation. These values will be used again in the first step of the scheme once the next cycle of the scheme is iterated.

3.3 CBS Theory and formulation

The Characteristic Based Split algorithm may be carried out in one of two distinct ways known as either Split A and Split B in literature. The difference between these two CBS approaches lies in the manner in which the intermediate momentum variables are formed by omitting the pressure terms. The first approach in calculating the intermediate momentum variables omits the pressure terms in their entirety, which is known as split A.

The other approach which is known as the Split B variation of the CBS scheme is based on an aspect of the pressure field after it is decoupled from the other variables. The Split B procedure formulates the intermediate momentum variables with the addition of the p^n terms so that only the p^{n+1} terms are omitted from the momentum equation.

The Split A procedure of the CBS scheme is used in this dissertation, however the second order pressure term is treated in the manner in which the Split B procedure would have treated this term by only omitting the p^{n+1} terms from the momentum equation.

The advantage of handling the second order pressure term in the manner described in the previous paragraph is that an additional stabilisation is created in the pressure field while a the Split B would have remained more difficult to stabilise due to the possibility of a zero term appearing in the discrete matrix of the continuity equation [4, 38, 39]. The described treatment of the decoupled pressure terms will be discussed in more mathematical detail at a later stage.

For convenience the governing equations will be repeated at this point from the previous chapter as the starting point for the derivation of the CBS scheme used in this dissertation. The governing equations may be solved in a time increment Δt as the only coupling which exists is through the speed of sound denoted by the term c which is present in the boundary condition of the energy equation, the value of the speed of sound can be taken as the value at the time t_n due to the explicit nature of the CBS algorithm.[40]

Therefore the following equations are the starting point for the discretisation of the CBS scheme.

The continuity equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{3.12}$$

The momentum equation

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j u_i) + \frac{\partial p}{\partial x_i} - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j} = 0 \tag{3.13}$$

The energy equation

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j H) - \frac{1}{P_r R_e}\frac{\partial}{\partial x_j}\left(k\frac{\partial T}{\partial x_j}\right) - \frac{1}{R_e}\frac{\partial (\tau_{ji} u_i)}{\partial x_j} = 0 \tag{3.14}$$

Note that the source terms are neglected from the momentum and energy equations for the sake of simplicity.

3.3.1 Temporal Discretisation

Firstly the temporal Discretisation will be considered. The continuity equation repeated below is the simplest of the governing equations to discretise.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{3.15}$$

By applying the forward difference time integral, the temporal discretisation of the conservation of mass equation is complete.

$$\rho^{n+1} = \rho^n - \Delta t \frac{\partial}{\partial x_j}(\rho u_j)^n \tag{3.16}$$

The second equation, the momentum equation will now be discretised by applying the Characteristic Galerkin procedure. Note that the momentum transport quantity U_i has been introduced at this point to replace the terms ρu_i which have been used previously in the equations.

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j}(U_i u_j) + \frac{\partial p}{\partial x_i} - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j} = 0 \tag{3.17}$$

Note that the momentum equation has been written in the form of a convection diffusion equation similar to Equation (3.1) shown below.

$$\frac{\partial R}{\partial t} + u\frac{\partial R}{\partial x} - \frac{\partial}{\partial x}\left(k\frac{\partial R}{\partial x}\right) = 0 \tag{3.18}$$

A Characteristic Galerkin procedure undertaken on Equation (3.1) results in Equation (3.11) which was derived at the start of this chapter.

$$\begin{aligned} \Delta R = R^{n+1} - R^n = & - \Delta t\left[\frac{\partial u_j R}{\partial x_j} - \frac{\partial}{\partial x_i}\left(k\frac{\partial R}{\partial x_i}\right)\right]^n \\ & + \frac{\Delta t^2}{2}u_k^n\frac{\partial}{\partial x_k}\left[\frac{\partial u_j R}{\partial x_j} - \frac{\partial}{\partial x_i}\left(k\frac{\partial R}{\partial x_i}\right)\right]^n \end{aligned} \tag{3.19}$$

A Characteristic Galerkin approximation of the similar momentum equation will therefore result in Equation (3.20) been formed.

$$\begin{aligned} U_i^{n+1} - U_i^n = & - \Delta t\left[\frac{\partial}{\partial x_j}(U_i u_j)^n + \frac{\partial p}{\partial x_i} - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j}^n\right] \\ & + \frac{\Delta t^2}{2}\left[u_k^n\frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j}(U_i u_j)^n + \frac{\partial p}{\partial x_i} - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j}^n\right)\right] \end{aligned} \tag{3.20}$$

The third order term is assumed to be neglible as the contribution from this term would have been quite small.

$$\begin{aligned} U_i^{n+1} - U_i^n = & - \Delta t\left[\frac{\partial}{\partial x_j}(U_i u_j)^n + \frac{\partial p}{\partial x_i} - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j}^n\right] \\ & + \frac{\Delta t^2}{2}\left[u_k^n\frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j}(U_i u_j)^n + \frac{\partial p}{\partial x_i}\right)\right] \end{aligned} \tag{3.21}$$

At this stage the split is now introduced into our formulation, as mentioned earlier the pressure gradient terms shall now be removed.
To accomplish the above, an auxiliary intermediate variable U_i^* is introduced, such that.

$$\Delta U_i^* = U_i^* - U_i^n = -\Delta t\left[\frac{\partial}{\partial x_j}(U_i u_j) - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j}\right]^n + \frac{\Delta t^2}{2}\left[u_k^n\frac{\partial}{\partial x_k}\frac{\partial}{\partial x_j}(U_i u_j)\right]^n \tag{3.22}$$

The superscript $*$ denotes an intermediate quantity.

The intermediate approximation for momentum is now calculated and is in turn is updated once the pressure gradient has been evaluated from the aforementioned intermediate momentum field.

The pressure correction U_i^{**} needed for updating the intermediate velocity is deduced by the difference between Equations (3.21) and (3.22).

$$U_i^{**} = -\Delta t \frac{\partial p}{\partial x_i} + \frac{\Delta t^2}{2} u_k^n \frac{\partial^2 p}{\partial x_k \partial x_i} \tag{3.23}$$

The pressure term may be treated as a known independent (source type) quantity, provided that a de-coupled process for the evaluation of the pressure term exists. In order to de-couple the pressure field, the pressure is approximated using a time increment $t = t^n + \theta_2 \Delta t$. This approximation results in the decoupled pressure correction shown below.

$$U_i^{**} = -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \frac{\Delta t^2}{2} u_k^n \frac{\partial^2 p^{n+\theta_2}}{\partial x_k \partial x_i} \tag{3.24}$$

The decoupled pressure is rewritten as

$$\frac{\partial p}{\partial x_i}^{n+\theta_2} = \theta_2 \frac{\partial p}{\partial x_i}^{n+1} + (1-\theta_2) \frac{\partial p}{\partial x_i}^{n} \tag{3.25}$$

Or alternatively as

$$\frac{\partial p}{\partial x_i}^{n+\theta_2} = \frac{\partial p}{\partial x_i}^{n} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \tag{3.26}$$

The change in pressure from the above equation is defined as

$$\Delta p = p^{n+1} - p^n \tag{3.27}$$

At this stage there is a choice in how to proceed with the CBS split, the above pressure correction can be maintained which is known as 'Split A' in CBS literature or the p^n terms from the above pressure de-coupling terms can be incorporated into the intermediate velocity Equation (3.22), this re-incorporation is known as 'Split B' in literature.

As mentioned earlier in a previous section of this chapter, Split A is used in this

dissertation. However, the second order pressure term is treated as it would have been in split B as it would otherwise have become a third order term in Split A derivation. The intermediate velocity and pressure correction equations therefore becomes the following.

$$\begin{aligned}\Delta U_i^* = U_i^* - U_i^n = & - \Delta t \left[\frac{\partial}{\partial x_j}\left(U_i u_j\right) - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j}\right]^n \\ & + \frac{\Delta t^2}{2}\left[u_k^n \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j}\left(U_i u_j\right) + \left(1-\theta_2\right)\frac{\partial p}{\partial x_i}\right)\right]^n\end{aligned} \tag{3.28}$$

The pressure correction is thus given by the following relation where the higher second order term is neglected due to the fact that it will develop into a third order term eventually

$$U_i^{**} = -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \theta_2 \frac{\Delta t^2}{2} u_k^n \frac{\partial^2 \Delta p}{\partial x_k \partial x_i} = -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \tag{3.29}$$

In order to calculate the pressure correction, the calculation of an intermediate density field is first necessary, which is calculated with the following procedure. It can be seen that Equation (3.30) gives a relation for U^{n+1}

$$\begin{aligned}\Delta U_i = U_i^{n+1} - U_i^n &= \Delta U_i^* + \Delta U_i^{**} \\ &= \Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i}\end{aligned} \tag{3.30}$$

And the conservation of mass equation (3.16) gives a relation of density to intermediate velocity.

$$\Delta \rho = -\Delta t \frac{\partial U_i^{n+\theta_1}}{\partial x_i} = -\Delta t \left[\left(1-\theta_1\right)\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial U_i^{n+1}}{\partial x_i}\right] \tag{3.31}$$

Rearranging Equation (3.16) and Equation (3.30) after replacing U_i^{n+1}, the following equation is formed.

$$\Delta \rho = -\Delta t \left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t \theta_1 \left(\frac{\partial^2 p^n}{\partial x_i \partial x_i} + \theta_2 \frac{\partial^2 \Delta p}{\partial x_i \partial x_i}\right)\right] \tag{3.32}$$

Equation (3.32) shall now be solved subsequently by an explicit timestep which

is applied to the discretised form enabling a full solution ($\theta_2 = 0$). The standard Galerkin type procedure may now be optimally used for the spatial approximation of the second Step of the CBS scheme which will be discussed in more detail in the next section.
Therefore Equation (3.32) becomes the following.

$$\Delta\rho = -\Delta t \left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \left(\frac{\partial U_i^*}{\partial x_i} - \frac{\partial U_i^n}{\partial x_i} \right) - \Delta t \theta_1 \frac{\partial^2 p^n}{\partial x_i \partial x_i} \right] \tag{3.33}$$

Now that the density has been calculated, the ideal gas law establishes the relation of pressure to density

$$p = \rho \left[RT \right]^n = \rho \frac{\gamma - 1}{\gamma} T^n \tag{3.34}$$

Since the temperature is given by

$$T = \gamma \left(E - \frac{1}{2} u_i u_i \right) \tag{3.35}$$

The pressure field can be formulated in terms of the momentum U, by the following equation from the above two relations.

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \frac{U_i U_i}{\rho} \right) \tag{3.36}$$

Once the pressure field from Equation (3.36) is calculated, the "pressure correction" given in Equation (3.30) below updates the intermediate velocity variable to find the actual velocity variable.

$$U_i^{n+1} = U_i^* + \Delta U_i^{**} = U_i^* - \Delta t \frac{\partial p^n}{\partial x_i} \tag{3.37}$$

The treatment of the energy equation follows a similar approach to both the treatment of the momentum equation and the conservation of mass equations. The energy equation is repeated here again for convenience as the following.

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j H) - \frac{1}{P_r R_e} \frac{\partial}{\partial x_j} \left(k \frac{\partial T}{\partial x_j} \right) - \frac{1}{R_e} \frac{\partial (\tau_{ji} u_i)}{\partial x_j} = 0 \tag{3.38}$$

However utilising the relation of H to E.

$$H = E + \frac{p}{\rho}$$

And applying forward difference in time to Equation (3.38), Equation (3.39) is formed.

$$\frac{\rho E^{n+1} - \rho E^n}{\Delta t} = -\frac{\partial}{\partial x_j} u_j \left(\rho E + p\right)^n + \frac{1}{P_r R_e} \frac{\partial}{\partial x_j} \left(k \frac{\partial T^n}{\partial x_j} \right) + \frac{1}{R_e} \frac{\partial \left(\tau_{ji} u_i\right)^n}{\partial x_j} \quad (3.39)$$

The Characteristic Galerkin procedure is applied to the energy equation above, this application results in Equation (3.40) being formed which is used to calculate the value of the energy term in the computational code.

The intermediate stages of the characteristic Galerkin expansion are omitted for brevity as the energy equation's treatment is the same as the procedure in which the momentum equation was treated earlier

$$\begin{aligned} \Delta\left(\rho E\right) = & - \Delta t \frac{\partial}{\partial x_j} u_j \left(\rho E + p\right)^n + \frac{\Delta t^2}{2} u_k^n \frac{\partial}{\partial x_k} \left[\frac{\partial}{\partial x_j} u_j \left(\rho E + p\right)^n \right] \\ & + \frac{\Delta t}{P_r R_e} \frac{\partial}{\partial x_j} \left(k \frac{\partial T^n}{\partial x_j} \right) + \frac{\Delta t}{R_e} \frac{\partial \left(\tau_{ji} u_i\right)^n}{\partial x_j} \end{aligned} \quad (3.40)$$

The characteristic based split scheme will be summarised with respect to the temporal discretisation, before moving on to the spatial discretisation section. The four steps in this split are sorted by procedure as

1. Remove the pressure terms from the momentum equation and solve the altered momentum Equation (3.22) for the intermediate momentum conservation variables. ΔU^*

2. Solve Equation (3.32) for pressure correction terms ΔU^{**} using the obtained values of ΔU^*

3. The momentum variables ΔU for the current timestep may now be obtained from solving Equation (3.37) using the obtained values for Δp

4. Solve Equation (3.40) for the internal energy before proceeding to the next timestep

3.3.2 Spatial Discretisation

From the previous section, the derived equations numbered (3.22), (3.32), (3.37) and (3.40) which represent the four steps of the CBS step will now be discretised in space using Galerkin spatial approximation.

To accomplish this Galerkin spatial discretisation, the following spatial approximations shown below are utilised to find the elemental values where **N** represents the elemental shape functions.

$$\begin{aligned} u_i = \mathbf{N}\overline{\mathbf{u_i}} \qquad U_i &= \mathbf{N}\overline{\mathbf{U_i}} \qquad \rho = \mathbf{N}\overline{\rho} \\ p = \mathbf{N}\overline{\mathbf{p}} \qquad E &= \mathbf{N}\overline{\mathbf{E}} \qquad \mathbf{T} = \mathbf{N}\overline{\mathbf{T}} \end{aligned} \tag{3.41}$$

A term which an overbar denotes a nodal quantity and a term in bold font is a vector based quantity.

Step 1 Discretisation

First let Equation (3.22) be considered as the derivation starting point which is repeated below for convenience.

$$\begin{aligned} \Delta U_i^* = U_i^* - U_i^n = \; &- \Delta t \left[\frac{\partial}{\partial x_j}(U_i u_j) - \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j} \right]^n \\ &+ \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k} \left[\frac{\partial}{\partial x_j}(U_i u_j) + \frac{\partial p}{\partial x_i} \right]^n \end{aligned} \tag{3.42}$$

The Galerkin weak form of the above equation establishes the intermediate momentum variables of Step 1 as follows.

$$\begin{aligned} \int_\Omega \mathbf{N}^T \Delta U_i^* d\Omega = \; &- \Delta t \left[\int_\Omega \mathbf{N}^T \frac{\partial}{\partial x_j}(u_j U_i)\, d\Omega - \int_\Omega \mathbf{N}^T \frac{1}{R_e}\frac{\partial \tau_{ij}}{\partial x_j} d\Omega \right] \\ &+ \frac{\Delta t^2}{2} \left[\int_\Omega \mathbf{N}^T u_k \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial p}{\partial x_i} \right) d\Omega \right] \end{aligned} \tag{3.43}$$

Since the viscous stress τ_{ij} is defined by the equation below, Equation (3.45) is therefore formed.

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \right) \tag{3.44}$$

Note that the variable u_k^e is also introduced at this point which is an elemental averaged approximation for the velocity vector.

$$\begin{aligned}\int_\Omega \mathbf{N}^T \Delta U_i^* d\Omega = \quad & \Delta t \left[\int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j}(u_j U_i)\, d\Omega - \frac{\Delta t}{2}\int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_k} u_k^e \left(\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial p}{\partial x_i}\right) d\Omega\right] \\ & -\Delta t \left[\int_\Gamma \left(\mathbf{N}^T (u_j U_i)\right) n_j d\Gamma - \frac{\Delta t}{2}\int_\Gamma \mathbf{N}^T u_k^e \left(\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial p}{\partial x_i}\right) n_j d\Gamma\right] \\ & -\Delta t \frac{\mu}{Re}\left[\int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) d\Omega\right] \\ & +\Delta t \frac{\mu}{Re}\left[\int_\Gamma \mathbf{N}^T \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) n_j d\Gamma\right] \end{aligned} \tag{3.45}$$

At this stage the continuous velocity field variables in Equation (3.45) are replaced with the nodal approximations found in Equation (3.41), the resultant matrix form formed is given as

$$\mathbf{M}_1 \overline{\mathbf{\Delta U}_i^*} = \Delta t \left[\mathbf{C}_1 \overline{\mathbf{u}_j \mathbf{U}_i} - \mathbf{f}_1 + \frac{\Delta t}{2}\mathbf{f}_2 - \frac{\Delta t}{2}\mathbf{K}_1 \left(\overline{\mathbf{u}_j \mathbf{U}_i} + \overline{\mathbf{p}}\right) - \mathbf{v}_1 + \mathbf{v}_2\right] \tag{3.46}$$

The component matrix terms are as follows, firstly the mass matrix $\mathbf{M}_1$ is given by the following for an element.

$$\mathbf{M}_1 = \int_\Omega \mathbf{N}^T \mathbf{N} d\Omega = \frac{V}{20}\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \approx \frac{V}{4}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Mass matrix $\mathbf{M}_1$ is lumped in the above equation in order to make the solution matrix free. The term $\mathbf{C}_1 \overline{\mathbf{u}_j \mathbf{U}_i}$ or the convective matrix term is the following.

$$\mathbf{C}_1 \overline{\mathbf{u}_j \mathbf{U}_i} = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_j} \mathbf{N} \overline{\mathbf{u}_j \mathbf{U}_i} d\Omega = \frac{V}{4}\begin{bmatrix} \frac{\partial N_1}{\partial x_j} \\ \frac{\partial N_2}{\partial x_j} \\ \frac{\partial N_3}{\partial x_j} \\ \frac{\partial N_4}{\partial x_j} \end{bmatrix} \sum_{s=1}^{nodes} \left[\overline{\mathbf{u}_j \mathbf{U}_i}\right]^s \quad j = 1, 2, 3$$

Note that the summation s is over the number of nodes and that the summation j is over the number of coordinate directions.
The first order boundary term $\mathbf{f}_1$ is defined as

$$\mathbf{f}_1 = \int_\Gamma \left(\mathbf{N}^T \mathbf{N}\overline{\mathbf{u}_j\mathbf{U}_i}\right)\mathbf{n}_j d\Gamma = \int_\Gamma \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ 0 \end{bmatrix} [N_1 \quad N_2 \quad N_3 \quad 0] \begin{bmatrix} \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^a \\ \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^b \\ \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^c \\ 0 \end{bmatrix} \mathbf{Cos}\alpha_j \, d\Gamma$$

$$= \frac{A}{12}\mathbf{Cos}\alpha_j \begin{bmatrix} 2\left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^a + \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^b + \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^c \\ \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^a + 2\left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^b + \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^c \\ \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^a + \left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^b + 2\left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)^c \\ 0 \end{bmatrix} \quad j = 1,2,3 \tag{3.47}$$

The superscripts a b and c in the above equation denote the three nodes of the triangular boundary face and $\mathbf{Cos}\alpha_j$ denotes the components of the normal vector on the boundary face.
The second order diffusion term $\mathbf{K}_1\left(\overline{\mathbf{u}_j\mathbf{U}_i} + \overline{\mathbf{p}}\right)$ is defined as

$$\mathbf{K}_1\left(\overline{\mathbf{u}_j\mathbf{U}_i} + \overline{\mathbf{p}}\right) = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_k} u_k^e \left[\frac{\partial}{\partial \mathbf{x}_j}\mathbf{N}\left(\overline{\mathbf{u}_j\mathbf{U}_i}\right) + \frac{\partial}{\partial \mathbf{x}_i}\mathbf{N}\overline{\mathbf{p}}\right] d\Omega$$

$$= u_k^e V \begin{bmatrix} \frac{\partial N_1}{\partial x_k} \\ \frac{\partial N_2}{\partial x_k} \\ \frac{\partial N_3}{\partial x_k} \\ \frac{\partial N_4}{\partial x_k} \end{bmatrix} \left[\frac{\partial \overline{\mathbf{p}}}{\partial \mathbf{x}_i} + \frac{\partial\left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)}{\partial \mathbf{x}_j}\right] \quad j,k = 1,2,3 \tag{3.48}$$

The corresponding second order boundary term $\mathbf{f}_2$

$$\mathbf{f}_2 = \int_\Gamma \mathbf{N}^T u_k^e \frac{\partial}{\partial \mathbf{x}_k}\left[\mathbf{N}\left(\overline{\mathbf{u}_j\mathbf{U}_i}\right) + \mathbf{N}\overline{\mathbf{p}}\right]\mathbf{n}_j d\Gamma$$

$$= u_k^e \frac{A}{3}\mathbf{Cos}\alpha_k \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \left[\frac{\partial \overline{\mathbf{p}}}{\partial \mathbf{x}_i} + \frac{\partial\left(\overline{\mathbf{u}_j\mathbf{U}_i}\right)}{\partial \mathbf{x}_j}\right] \quad j,k = 1,2,3 \tag{3.49}$$

The first order viscous term $\mathbf{v}_1$ is defined below as

$$\begin{aligned}\mathbf{v}_1 &= \frac{\mu}{Re}\int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_j}\left(\frac{\partial \mathbf{N}\overline{\mathbf{u}}_i}{\partial \mathbf{x}_j}+\frac{\partial \mathbf{N}\overline{\mathbf{u}}_j}{\partial \mathbf{x}_i}-\frac{2}{3}\frac{\partial \mathbf{N}\overline{\mathbf{u}}_k}{\partial \mathbf{x}_k}\delta_{ij}\right)d\Omega \\ &= \frac{1}{Re}V\begin{bmatrix}\frac{\partial N_1}{\partial x_j}\\ \frac{\partial N_2}{\partial x_j}\\ \frac{\partial N_3}{\partial x_j}\\ \frac{\partial N_4}{\partial x_j}\end{bmatrix}(\overline{\tau}_{j,1}+\overline{\tau}_{j,2}+\overline{\tau}_{j,3}) \quad j=1,2,3 \end{aligned} \tag{3.50}$$

The viscous boundary term $\mathbf{v}_2$ is defined as

$$\begin{aligned}\mathbf{v}_2 &= \frac{\mu}{Re}\int_\Gamma \mathbf{N}^T\mathbf{N}\left(\frac{\partial \overline{\mathbf{u}}_i}{\partial \mathbf{x}_j}+\frac{\partial \overline{\mathbf{u}}_j}{\partial \mathbf{x}_i}-\frac{2}{3}\frac{\partial \overline{\mathbf{u}}_k}{\partial \mathbf{x}_k}\delta_{ij}\right)\mathbf{n}_j d\Gamma \\ &= \frac{1}{Re}\frac{A}{3}\mathbf{Cos}\alpha_j\begin{bmatrix}1\\1\\1\\0\end{bmatrix}(\overline{\tau}_{j,1}+\overline{\tau}_{j,2}+\overline{\tau}_{j,3}) \quad j=1,2,3 \end{aligned} \tag{3.51}$$

Step 2 Discretisation

Let us now consider the spatial discretisation of the second step of the CBS scheme, Equation (3.32) is repeated below for convenience.

$$\Delta\rho = -\Delta t\left[\frac{\partial U_i^n}{\partial x_i}+\theta_1\frac{\partial \Delta U_i^*}{\partial x_i}-\Delta t\theta_1\left(\frac{\partial^2 p^n}{\partial x_i \partial x_i}+\theta_2\frac{\partial^2 \Delta p}{\partial x_i \partial x_i}\right)\right]$$

The Galerkin weak form of the equation that finds the density/pressure variables is as follows when $\theta_2 = 0$.

$$\begin{aligned}\int_\Omega \mathbf{N}^T\Delta\rho d\Omega &= -\Delta t\int_\Omega \mathbf{N}^T\left[\frac{\partial U_i}{\partial x_i}+\theta_1\frac{\partial \Delta U_i^*}{\partial x_i}-\Delta t\theta_1\frac{\partial^2 p}{\partial x_i \partial x_i}\right]d\Omega \\ &= -\Delta t\int_\Omega \mathbf{N}^T\frac{\partial}{\partial x_i}\left[U_i+\theta_1\Delta U_i^*-\Delta t\theta_1\frac{\partial p}{\partial x_i}\right]d\Omega \end{aligned} \tag{3.52}$$

After rewriting the equation in the above manner with regard to the derivative of x_i, an integration by parts is performed which results in Equation (3.53).

$$\begin{aligned}\int_\Omega \mathbf{N}^T \Delta\rho d\Omega &= \Delta t \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_i}\left[U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \frac{\partial p}{\partial x_i}\right] d\Omega \\ &-\Delta t \int_\Gamma \mathbf{N}^T \left[U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \frac{\partial p}{\partial x_i}\right] n_i d\Gamma \end{aligned} \tag{3.53}$$

The stage is reached where the continuous field variables in the above Equation (3.53) are substituted with the approximations given in Equation (3.43) to obtain the matrix form of the equation. An overbar denotes that the variable is nodal.

$$\mathbf{M}_2 \Delta\overline{p} = \Delta t \left[\mathbf{C}_2\left(\overline{\mathbf{U}}_i + \theta_1 \Delta \overline{\mathbf{U}}_i^*\right) - \mathbf{f}_3\right] - \Delta t^2 \theta_1 \left[\mathbf{K}_2 \overline{p} - \mathbf{f}_4\right] \tag{3.54}$$

The consistent matrix terms are stated as follows, starting with the lumped mass matrix term.

$$\mathbf{M}_2 = \int_\Omega \mathbf{N}^T \mathbf{N} d\Omega \approx \frac{V}{4}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The mass matrix is lumped above as it was in the previous step, the convection term $\mathbf{C}_2\left(\overline{\mathbf{U}}_i + \theta_1 \Delta \overline{\mathbf{U}}_i^*\right)$ is then given as the following, where once again the summation s is over the number of nodes and the summation i is over the number of coordinate directions.

$$\mathbf{C}_2\left(\overline{\mathbf{U}}_i + \theta_1 \Delta \overline{\mathbf{U}}_i^*\right) = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_i} \mathbf{N}\left(\overline{\mathbf{U}}_i + \theta_1 \Delta \overline{\mathbf{U}}_i^*\right) d\Omega, \tag{3.55}$$

$$= \frac{V}{4}\begin{bmatrix} \frac{\partial N_1}{\partial x_i} \\ \frac{\partial N_2}{\partial x_i} \\ \frac{\partial N_3}{\partial x_i} \\ \frac{\partial N_4}{\partial x_i} \end{bmatrix} \sum_{s=1}^{nodes} \left[\overline{\mathbf{U}}_i + \theta_1 \Delta \overline{\mathbf{U}}_i^*\right]^s = \frac{V}{4}\begin{bmatrix} \frac{\partial N_1}{\partial x_i} \\ \frac{\partial N_2}{\partial x_i} \\ \frac{\partial N_3}{\partial x_i} \\ \frac{\partial N_4}{\partial x_i} \end{bmatrix} \sum_{s=1}^{nodes} \left[\left(1-\theta_1\right)\overline{\mathbf{U}}_i + \theta_1 \overline{\mathbf{U}}_i^*\right]^s \quad i = 1,2,3$$

The diffusion term $\mathbf{K}_2\overline{\mathbf{p}}$ is given as

$$\mathbf{K}_2 = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_i}\mathbf{N}\frac{\partial \overline{\mathbf{p}}}{\partial \mathbf{x}_i}d\Omega = V \begin{bmatrix} \frac{\partial N_1}{\partial x_i} \\ \frac{\partial N_2}{\partial x_i} \\ \frac{\partial N_3}{\partial x_i} \\ \frac{\partial N_4}{\partial x_i} \end{bmatrix} \left(\frac{\partial \overline{\mathbf{p}}}{\partial \mathbf{x}_i}\right) \quad j = 1,2,3 \tag{3.56}$$

The boundary momentum term $\mathbf{f}_3$ is given by the following for $i = 1, 2, 3$.

$$\begin{aligned} \mathbf{f}_3 &= \int_\Gamma \mathbf{N}^T\mathbf{N}\left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)\mathbf{n}_j d\Gamma \qquad (3.57)\\ &= \frac{A}{12}\,\mathbf{Cos}\alpha_i \begin{bmatrix} 2\left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^a + \left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^b + \left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^c \\ \left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^a + 2\left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^b + \left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^c \\ \left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^a + \left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^b + 2\left((1-\theta_1)\,\overline{\mathbf{U}}_i + \theta_1\overline{\mathbf{U}}_i^*\right)^c \\ 0 \end{bmatrix} \end{aligned}$$

The superscripts a b and c denote the three nodes of the boundary face, finally the boundary term $\mathbf{f}_4$ which contains the pressure terms is as follows

$$\mathbf{f}_4 = \int_\Gamma \mathbf{N}^T\mathbf{N}\frac{\partial \overline{\mathbf{p}}}{\partial \mathbf{x}_i}\mathbf{n}_i d\Gamma = \frac{A}{3}\,\mathbf{Cos}\alpha_i \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \left(\frac{\partial \overline{\mathbf{p}}}{\partial \mathbf{x}_i}\right) \quad i = 1,2,3 \tag{3.58}$$

Where $\mathbf{Cos}\alpha_i$ is the normal coefficent summated over the three coordinate directions.

Step 3 Discretisation

Let us now consider the spatial discretisation of the third step of the CBS scheme where the intermediate velocity field is updated by the newly calculated pressure correction, Equation (3.37) representing the third step of the CBS scheme is re-

peated below for convenience where $\theta_2 = 0$.

$$U_i^{n+1} = U_i^* + \Delta U_i^{**} = U_i^* - \Delta t \frac{\partial p^n}{\partial x_i} \tag{3.59}$$

The Galerkin weak form of the equation is as follows.

$$\int_\Omega \mathbf{N}^T \left(U_i^{n+1} - U_i^*\right) d\Omega = -\Delta t \int_\Omega \mathbf{N}^T \frac{\partial p}{\partial x_i} d\Omega \tag{3.60}$$

Performing integration by parts of the weak form, results in Equation (3.61) defined as.

$$\int_\Omega \mathbf{N}^T \left(U_i^{n+1} - U_i^*\right) d\Omega = \Delta t \left[\int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_i} p \, d\Omega - \int_\Gamma \mathbf{N}^T p \, n_i d\Gamma\right] \tag{3.61}$$

Substitution the continuous field variables with the approximations given by Equation (3.43), the matrix form of Equation (3.61) is therefore the following. An overbar denotes a nodal quantity.

$$\mathbf{M}_3 \left(\overline{\mathbf{U}}_i^{n+1} - \overline{\mathbf{U}}_i^*\right) = \Delta t \left[\mathbf{C}_3 \overline{\mathbf{p}} - \mathbf{f}_5\right] \tag{3.62}$$

The individual component matrix terms are as follows, firstly the mass matrix has been lumped above as it was previously

$$\mathbf{M}_3 = \int_\Omega \mathbf{N}^T \mathbf{N} d\Omega \approx \frac{V}{4} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The convection term $\mathbf{C}_3 \, \overline{\mathbf{p}}$ is given as follows, where once again the summation s is over the number of nodes and the summation i is over the number of coordinate directions.

$$\mathbf{C}_3 \, \overline{\mathbf{p}} = \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_i} \mathbf{N} \, \overline{\mathbf{p}} = \frac{V}{4} \begin{bmatrix} \frac{\partial N_1}{\partial x_i} \\ \frac{\partial N_2}{\partial x_i} \\ \frac{\partial N_3}{\partial x_i} \\ \frac{\partial N_4}{\partial x_i} \end{bmatrix} \sum_{s=1}^{nodes} (\overline{\mathbf{p}})^s \quad i = 1, 2, 3 \tag{3.63}$$

And the boundary term $\mathbf{f}_5$ is

$$\mathbf{f}_5 = \int_\Gamma \mathbf{N}^T \mathbf{N}\overline{\mathbf{p}}\, \mathbf{n}_i d\Gamma = \frac{A}{12}\, \mathbf{Cos}\alpha_i \begin{bmatrix} 2\left(\overline{\mathbf{p}}\right)^a + \left(\overline{\mathbf{p}}\right)^b + \left(\overline{\mathbf{p}}\right)^c \\ \left(\overline{\mathbf{p}}\right)^a + 2\left(\overline{\mathbf{p}}\right)^b + \left(\overline{\mathbf{p}}\right)^c \\ \left(\overline{\mathbf{p}}\right)^a + \left(\overline{\mathbf{p}}\right)^b + 2\left(\overline{\mathbf{p}}\right)^c \\ 0 \end{bmatrix} \quad i = 1,2,3 \qquad (3.64)$$

The superscripts a b and c denote the three faces of the boundary edge as before.

Step 4 Discretisation

Let us now consider the spatial discretisation for the final step of the CBS scheme the update of the energy values. The conservation of energy equation (3.40) is repeated here for convenience.

$$\begin{aligned} \frac{\Delta(\rho E)}{\Delta t} = & - \frac{\partial}{\partial x_j} u_j (\rho E + p)^n + \frac{\Delta t}{2} u_k^n \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j} u_j (\rho E + p)^n\right) \\ & + \frac{1}{P_r R_e}\frac{\partial}{\partial x_j}\left(k\frac{\partial T}{\partial x_j}\right)^n + \frac{1}{R_e}\frac{\partial (\tau_{ji} u_i)^n}{\partial x_j} \end{aligned} \qquad (3.65)$$

The Galerkin weak form of the equation is shown below.

$$\begin{aligned} \int_\Omega \mathbf{N}^T \Delta(\rho E)\, d\Omega = & - \Delta t \int_\Omega \mathbf{N}^T \frac{\partial}{\partial x_j} u_j (\rho E + p)\, d\Omega \\ & + \frac{\Delta t^2}{2}\int_\Omega \mathbf{N}^T u_k \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j} u_j (\rho E + p)\right) d\Omega \\ & + \Delta t \int_\Omega \mathbf{N}^T \frac{1}{P_r R_e}\frac{\partial}{\partial x_j}\left(k\frac{\partial T}{\partial x_j}\right) d\Omega + \Delta t \int_\Omega \mathbf{N}^T \frac{1}{R_e}\frac{\partial (\tau_{ji} u_i)}{\partial x_j} d\Omega \end{aligned} \qquad (3.66)$$

Performing integration by parts of the above equation results in the following equation.

$$\int_\Omega \mathbf{N}^T \Delta(\rho E)\, d\Omega = \Delta t \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j} u_j (\rho E + p)\, d\Omega - \Delta t \int_\Gamma \mathbf{N}^T u_j (\rho E + p)\, n_j d\Gamma$$
$$-\frac{\Delta t^2}{2} u_k^e \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_k} \frac{\partial}{\partial x_j}(u_j (\rho E + p))\, d\Omega + \frac{\Delta t^2}{2} u_k^e \int_\Gamma \mathbf{N}^T \frac{\partial}{\partial x_j}(u_j (\rho E + p))\, n_k d\Gamma$$
$$+\Delta t \frac{1}{P_r R_e} \int_\Gamma \mathbf{N}^T \left(k \frac{\partial T}{\partial x_j}\right) n_j d\Gamma - \Delta t \frac{1}{P_r R_e} \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j} \left(k \frac{\partial T}{\partial x_j}\right) d\Omega$$
$$+\Delta t \frac{1}{R_e} \int_\Gamma \mathbf{N}^T (\tau_{ji} u_i)\, n_j d\Gamma - \Delta t \frac{1}{R_e} \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j} (\tau_{ji} u_i)\, d\Omega \qquad (3.67)$$

Note that the elemental averaged velocity vector denoted by the term u_k^e has been introduced into the energy equation at this point.

Substitution of the continuous field variables with the approximations given by Equation (3.43), results in the matrix form for step 4 of the CBS scheme. An overbar denotes a nodal quantity as before.

$$\begin{aligned} \mathbf{M}_4 \Delta\left(\overline{\rho \mathbf{E}}\right) &= \Delta t \left[\mathbf{C}_4 \overline{\mathbf{u}}_j \left(\overline{\rho \mathbf{E}} + \overline{\mathbf{p}}\right) - \mathbf{f}_6 - \mathbf{Q} + \mathbf{f}_Q - \mathbf{S}\left(\overline{\tau}_{ji} \overline{\mathbf{u}}_i\right) + \mathbf{f}_S\right] \\ &- \frac{\Delta t^2}{2} \left[\mathbf{K}_4 \overline{\mathbf{u}}_j \left(\overline{\rho \mathbf{E}} + \overline{\mathbf{p}}\right) + \mathbf{f}_7\right] \end{aligned} \qquad (3.68)$$

The consistent matrix terms are as follows, firstly the mass matrix has been lumped in the same manner as the preceding steps.

$$\mathbf{M}_4 = \int_\Omega \mathbf{N}^T \mathbf{N} d\Omega \approx \frac{V}{4} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The convection term $\mathbf{C}_4 \overline{\mathbf{u}}_j \left(\overline{\rho \mathbf{E}} + \overline{\mathbf{p}}\right)$ is shown below, where once again the summation s is over the number of nodes and the summation j is over the number of coordinate directions.

$$\begin{aligned}\mathbf{C}_4\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right) &= \int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_j}\mathbf{N}\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right) \\ &= \frac{V}{4}\begin{bmatrix}\frac{\partial N_1}{\partial x_j}\\ \frac{\partial N_2}{\partial x_j}\\ \frac{\partial N_3}{\partial x_j}\\ \frac{\partial N_4}{\partial x_j}\end{bmatrix}\sum_{s=1}^{nodes}\left[\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right]^s \quad j=1,2,3 \end{aligned} \tag{3.69}$$

The boundary term $\mathbf{f}_6$ corresponding to the first order convection term is given as

$$\begin{aligned}\mathbf{f}_6 &= \int_\Gamma \mathbf{N}^T\mathbf{N}\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\mathbf{n}_j d\Gamma \\ &= \frac{A}{12}\mathbf{Cos}\alpha_j\begin{bmatrix}2\left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^a+\left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^b+\left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^c\\ \left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^a+2\left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^b+\left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^c\\ \left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^a+\left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^b+2\left(\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right)^c\\ 0\end{bmatrix} \quad j=1,2,3 \end{aligned} \tag{3.70}$$

The superscripts a b and c are as before the nodes of the boundary face. The second order diffusion term $\mathbf{K}_4\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)$ is given as

$$\begin{aligned}\mathbf{K}_4\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right) &= \overline{\mathbf{u}}_k^e\int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_k}\mathbf{N}\frac{\partial}{\partial \mathbf{x}_j}\left[\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right]d\Omega \\ &= \overline{\mathbf{u}}_k^e\, V\begin{bmatrix}\frac{\partial N_1}{\partial x_k}\\ \frac{\partial N_2}{\partial x_k}\\ \frac{\partial N_3}{\partial x_k}\\ \frac{\partial N_4}{\partial x_k}\end{bmatrix}\left[\frac{\partial}{\partial \mathbf{x}_j}\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right] \quad j,k=1,2,3 \end{aligned} \tag{3.71}$$

The corresponding second order boundary term $\mathbf{f}_7$ is given as

$$\begin{aligned}\mathbf{f}_7 &= \overline{\mathbf{u}}_k^e \int_\Gamma \mathbf{N}^T\mathbf{N}\frac{\partial}{\partial \mathbf{x}_j}\left[\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right]\mathbf{n}_k d\Gamma \\ &= \overline{\mathbf{u}}_k^e \frac{A}{3}\,\mathbf{Cos}\alpha_j \begin{bmatrix}1\\1\\1\\0\end{bmatrix}\left[\frac{\partial}{\partial \mathbf{x}_j}\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}}+\overline{\mathbf{p}}\right)\right] \quad j,k=1,2,3 \end{aligned} \tag{3.72}$$

The temperature flux term $\mathbf{Q}$ is given below

$$\mathbf{Q} = \frac{1}{P_r R_e}\int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_j}\mathbf{N}\left(k\frac{\partial \overline{\mathbf{T}}}{\partial \mathbf{x}_j}\right)d\Omega = \frac{k}{P_r R_e}\,V \begin{bmatrix}\frac{\partial N_1}{\partial x_j}\\ \frac{\partial N_2}{\partial x_j}\\ \frac{\partial N_3}{\partial x_j}\\ \frac{\partial N_4}{\partial x_j}\end{bmatrix}\left(\frac{\partial \overline{\mathbf{T}}}{\partial \mathbf{x}_j}\right) \quad j=1,2,3 \tag{3.73}$$

The corresponding temperature flux $\mathbf{f}_Q$ boundary term is shown below where a b and c are the nodes of the boundary face

$$\mathbf{f}_Q = \frac{1}{P_r R_e}\int_\Gamma \mathbf{N}^T\mathbf{N}\left(k\frac{\partial \overline{\mathbf{T}}}{\partial \mathbf{x}_j}\right)\mathbf{n}_j d\Gamma = \frac{k}{P_r R_e}\frac{A}{3}\,\mathbf{Cos}\alpha_\mathbf{j} \begin{bmatrix}1\\1\\1\\0\end{bmatrix}\left(\frac{\partial \overline{\mathbf{T}}}{\partial \mathbf{x}_j}\right) \quad j=1,2,3 \tag{3.74}$$

The term representing the change in energy due to the viscous forces $\mathbf{S}\left(\overline{\tau}_{ji}\overline{\mathbf{u}}_i\right)$ is given by the following relation where τ_{ji} is once again calculated using Equation (3.44).

$$\begin{aligned}\mathbf{S}\left(\overline{\tau}_{ji}\overline{\mathbf{u}}_i\right) &= \frac{1}{R_e}\int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_j}\mathbf{N}^T\left(\overline{\tau}_{ji}\overline{\mathbf{u}}_i\right)d\Omega \\ &= \frac{1}{R_e}V\begin{bmatrix}\frac{\partial N_1}{\partial x_j}\\ \frac{\partial N_2}{\partial x_j}\\ \frac{\partial N_3}{\partial x_j}\\ \frac{\partial N_4}{\partial x_j}\end{bmatrix}\left(\overline{\tau}_{j1}u_1^e+\overline{\tau}_{j2}u_2^e+\overline{\tau}_{j3}u_3^e\right)\quad j=1,2,3\end{aligned} \tag{3.75}$$

Similarly the corresponding boundary term $\mathbf{f}_s$ is shown below where a b and c are the nodes of the boundary face

$$\mathbf{f}_s=\frac{1}{R_e}\int_\Gamma \mathbf{N}^T\left(\overline{\tau}_{ji}\overline{\mathbf{u}}_i\right)\mathbf{n}_j d\Gamma=\frac{1}{R_e}\frac{A}{3}\mathbf{Cos}\alpha_\mathbf{j}\begin{bmatrix}1\\1\\1\\0\end{bmatrix}\left(\overline{\tau}_{j1}u_1^e+\overline{\tau}_{j2}u_2^e+\overline{\tau}_{j3}u_3^e\right)\quad j=1,2,3 \tag{3.76}$$

The first step of the CBS split is always of explicit nature, however step two may be either of explicit or implicit nature depending on the choice taken regarding the defined value of the parameters θ_1 and θ_2. Should the fully explicit form of the split be adopted, then values of $\theta_2 = 0$ and θ_1 lying in the range 0.5 to 1.0 must be defined by the user.

In this dissertation a value of 0.5 is defined for θ_1.

3.3.3 Final CBS equations

The spatially discretised four steps comprising split A of the CBS scheme are summarized below. A lumped mass matrix is employed consistently at each step to ensure an explicit solution.

1. In the first step the pressure terms are omitted from the momentum equation, the resultant Equation (3.77) is solved to calculate the intermediate

momentum conservation variables.

$$\mathbf{M}_1\overline{\Delta\mathbf{U}}_i^* = \Delta t\left[\mathbf{C}_1\overline{\mathbf{u}_j\mathbf{U}_i} - \mathbf{f}_1 + \frac{\Delta t}{2}\mathbf{f}_2 - \frac{\Delta t}{2}\mathbf{K}_1\left(\overline{\mathbf{u}_j\mathbf{U}_i} + p\right) - \mathbf{v}_1 + \mathbf{v}_2\right] \tag{3.77}$$

2. In the second step of the split, the density values are found by using the values of the intermediate momentum conservation variables calculated from the preceding step. The pressure field is then updated using these density values.

$$\mathbf{M}_2\Delta\overline{p} = \Delta t\left[\mathbf{C}_2\left(\overline{\mathbf{U}}_i + \theta_1\Delta\overline{\mathbf{U}}_i^*\right) - \mathbf{f}_3\right] - \Delta t^2\theta_1\left[\mathbf{K}_2\overline{p} - \mathbf{f}_4\right] \tag{3.78}$$

3. In the third step of the CBS scheme, the intermediate momentum variables are updated to the next time step's momentum variables using the pressure field calculated from the second step.

$$\mathbf{M}_3\left(\overline{\mathbf{U}}_i^{n+1} - \overline{\mathbf{U}}_i^*\right) = \Delta t\left[\mathbf{C}_3\overline{p} - \mathbf{f}_5\right] \tag{3.79}$$

4. Finally in the fourth step of the split, the energy values are updated for use in the next iteration of the split.

$$\begin{aligned}\mathbf{M}_4\Delta\left(\overline{\rho\mathbf{E}}\right) &= \Delta t\left[\mathbf{C}_4\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}} + \overline{p}\right) - \mathbf{f}_6 - \mathbf{Q} + \mathbf{f}_Q - \mathbf{S}\left(\overline{\tau}_{ji}\overline{\mathbf{u}}_i\right) + \mathbf{f}_S\right] \\ &- \frac{\Delta t^2}{2}\left[\mathbf{K}_4\overline{\mathbf{u}}_j\left(\overline{\rho\mathbf{E}} + \overline{p}\right) + \mathbf{f}_7\right]\end{aligned} \tag{3.80}$$

This concludes the discretisation of the Characteristic Based Split Scheme.

3.4 Shock Capturing

The CBS scheme will find the exact prediction of shocks to be a difficult task in the present form. The accurate reproduction of shocks is impossible to achieve by a finite element approximation as the values of the variables will vary throughout the element opposed to reality where each shock will be present in only one location.

The best shock depiction that such a scheme can hope to predict is a region of very high variable gradient confined to a few mesh elements in length. These

regions of high gradient unfortunately produce local oscillations which may affect a much larger proportion of the domain should these oscillations propagate. Therefore in order to achieve an accurate shock prediction and overall solution, the application to the CBS scheme of additional shock treatment is necessary.

Although the application of additional shock treatment originally dates back to the 1950s [41], there are today two main regonised routes in the prediction of shocks. The first route is that of the 'shock capturing' method used in this dissertation, while the second route is the more computationally expensive 'shock fitting' technique which is generally limited to one dimensional problems due to the large computational cost of this method. In this dissertation the following described shock capturing method represented by Equation (3.81) below shall be used to predict shocks [42][43].

$$R_s^{n+1} = R^{n+1} + \Delta t \mu_a \frac{\partial}{\partial x_i} \left(\frac{\partial R}{\partial x_i} \right)^n \tag{3.81}$$

As is evident from the above equation, the smoothed variable R_s^{n+1} is found by applying an artificial diffusion term in the region of the shock location. The amount of artificial diffusion applied is itself controlled by the presence of the second derivative of pressure term incorporated in the μ_a term(see Equation (3.83)). The second derivative of pressure is used to denote the presence of regions of high gradient.

At this stage of the explanation, before moving on to the calculation of the artificial diffusion coefficient in more depth, the nodal pressure switch S_i is introduced which will utilised in the next section.

$$S_i = \frac{|\sum_{s=1}^{nodes} (p_i - p_s)|}{\sum_{s=1}^{nodes} |(p_i - p_s)|} \tag{3.82}$$

The term p_i denotes the pressure value at the node under consideration, while the p_s term denotes the values of pressure present at the other elemental nodes.

The nodal pressure switch S_i is an alternative method in determining the presence of a shock. The value of S_i will be asymptotically close to 1.0 should the pressure value at the considered node be vastly different from the pressure values at the surrounding elemental nodes.

Contrastingly should the pressure value at the considered node be the average

pressure value for nodes in the element then S_i will be calculated to have a zero value, thus denoting a linear variation of pressure across the element and the absence of any shock.

The elemental pressure switch S_e is calculated from the mean of the nodal pressure switches

3.4.1 Artificial diffusion

The artificial diffusion μ_a necessary for shock capturing is calculated in the following manner. MacCormak and Baldwin have proposed the following Equation (3.83) for μ_a [44], note that β is a user defined coefficient which normally lies in the range of zero to two.

$$\mu_a = \beta h^3 \frac{|\mathbf{u}+c|}{p^e} \left(\frac{\partial^2 p}{\partial x_i \partial x_i} \right)^e \tag{3.83}$$

The superscript e denotes an elemental quantity. In the above equation, the elemental pressure value p^e is calculated from the mean of the nodal pressure values.

The following approximation may be used to write a second derivative of a scalar value in terms of the mass matrixes[42].

$$\left(\frac{\partial^2 R}{\partial x_i \partial x_i} \right)^e = \frac{(\mathbf{M} - \mathbf{M}_L)}{h^2} \overline{\mathbf{R}} \tag{3.84}$$

Applying the above approximation to Equation (3.83), the following is given.

$$\mu_a = \beta h \frac{|\mathbf{u}+c|}{p^e} (\mathbf{M} - \mathbf{M}_L) \overline{\mathbf{p}} \tag{3.85}$$

Equation (3.85) can be further simplified by approximating the $(\mathbf{M} - \mathbf{M}_L)\overline{\mathbf{p}}/p^e$ term to the elemental pressure switch S^e resulting in Equation (3.86) below solving for the elemental artificial diffusion coefficient μ_a^e[30].

$$\mu_a^e = \beta h \left(|\mathbf{u}+c| \right) S^e \tag{3.86}$$

Since $(|\mathbf{u}+c|) = h\Delta t^e$ along a characteristic, Equation (3.86) and Equation (3.84) can be substituted into the original starting Equation (3.81) to form Equation

(3.87) used for the calculation of shock capture in this dissertation.

$$R_s^{n+1} = R^{n+1} + \Delta t \mathbf{M}_L^{-1} \frac{\beta S^e}{\Delta t^e} (\mathbf{M} - \mathbf{M}_L) \overline{\mathbf{R}} \tag{3.87}$$

The discretisation of the shock capturing technique is relatively straight forward. The consistent mass matrix $\mathbf{M}$ and the lumped mass matrix $\mathbf{M}_L$ are given by the following relations alongside the expanded form of $(\mathbf{M} - \mathbf{M}_L)$.

$$\mathbf{M} = \frac{V}{20}\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \mathbf{M}_L = \frac{V}{20}\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

$$(\mathbf{M} - \mathbf{M}_L) = \frac{V}{20}\begin{bmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{bmatrix} \tag{3.88}$$

Substituting these relations into Equation (3.87) results in Equation (3.89) the final form of the artificial diffusion equation that shall be implemented into the computational code.

$$\begin{aligned} R_s^{n+1} &= R^{n+1} + \Delta t \frac{V}{20}\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}^{-1} \frac{\beta S^e}{\Delta t^e}\frac{V}{20}\begin{bmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{bmatrix} \overline{\mathbf{R}} \\ &= R^{n+1} + \Delta t \frac{V}{20}\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}^{-1} \frac{\beta S^e}{\Delta t^e}\frac{V}{20}\begin{bmatrix} \left(R^b - R^a\right) + \left(R^c - R^a\right) + \left(R^d - R^a\right) \\ \left(R^a - R^b\right) + \left(R^c - R^b\right) + \left(R^d - R^b\right) \\ \left(R^a - R^c\right) + \left(R^b - R^c\right) + \left(R^d - R^c\right) \\ \left(R^a - R^d\right) + \left(R^b - R^d\right) + \left(R^c - R^d\right) \end{bmatrix} \end{aligned} \tag{3.89}$$

The superscripts a b c and d denote the four nodes of the tetrahedral element.

3.5 Variable Smoothing

It has been previously determined that at subsonic Mach numbers lower than the transonic range, it is necessary to apply variable smoothing treatment should the energy equation be coupled to other variables as it is in the CBS scheme [45].

In this dissertation, the transonic range has been assumed to begin at Mach numbers greater than 0.8, therefore those flows with Mach numbers lower than this number will undergo the following variable technique for the redistribution of each conservation variable $\mathbf{R}$ [46].

$$\mathbf{R} = \frac{1}{1+0.5\alpha}\mathbf{R} + \frac{\alpha}{1+0.5\alpha}\mathbf{M}_L^{-1}\left(\mathbf{M}-\mathbf{M}_D\right)\mathbf{R} \tag{3.90}$$

The different mass matrix types are denoted by the following terms, the term $\mathbf{M}$ denotes the consistent mass matrix, $\mathbf{M}_L$ denotes the lumped mass matrix and $\mathbf{M}_D$ denotes the diagonal of the consistent mass matrix. The term α is a user defined co-coefficient ranging from 0 to 0.05.

It is evident from Equation (3.90) that as the user defined coefficient increases, the weighting of $\mathbf{R}$ from the considered node decreases to be replaced by a higher weighting from the surrounding nodes in the element.

3.6 One equation turbulence model

As was already noted earlier there is no steady state solution possible in many computational simulations that model the Navier Stokes equations if the fluid portrays a turbulent nature. Fortunately it is possible to adapt the Navier Stokes equations by choosing and adopting a turbulence model to ensure a steady state solution may still be reached.

The Spalart Allmaras turbulence model [22] is one of the most commonly used one equation turbulence models due to the many advantages pertaining to this model, therefore this model was chosen as the one equation model to be used in this dissertation because of these advantages which shall now be discussed in further detail in the following paragraphs.

The Spalart Allmaras model operates by replacing the molecular laminar viscosity with a Reynold's averaged viscosity, this substitution of viscosities enables

the Navier Stokes equations to successfully model the averaged turbulent solution of the flow while refraining from increasing the mesh resolution to the impractical sizes that a three dimensional flow simulation incorporating a Direct Numerical Solution (DNS) would have otherwise required.

A further aspect of the Spalart Allmaras model which will be explained in more detail shortly is that the model includes a wall destructive source term dependent on the distance from the flow region to the wall surface. This wall destructive term allows the model to achieve greater stability as the large turbulent source term which could lead to a solution failure is not present throughout the domain but only present in the region of the wall's surface.

An additional advantageous attribute of the Spalart Allmaras model is the local or explicit nature of the calculated turbulence values, meaning that the values of the turbulence variables calculated by the model are not dependent on any other turbulence variable values existing elsewhere in the domain. This attribute allows the model to be compatible with any type of mesh whether the grid type be structured, unstructured or hybrid or whether the problem is of two or three dimensions, parallelised or serial in nature.

Another advantage of choosing the Spalart Allmaras model is that use of this model obtains a steady state solution at a computationally inexpensive cost on relatively course grids as the model performs well in high pressure gradient areas which has led to the model's usage in modeling transonic turbulent flow in complex industrial simulations [47].

3.6.1 Change in the momentum equation

The integration of the Spalart Allmaras turbulence model into the CBS scheme involves carrying out two alterations to the diffusive flux vector of the Navier Stokes equations. These two alterations will be now discussed as the laminar Navier Stokes equations are converted into the turbulent case [22, 48]. The diffusive flux vector

in Equation (2.1) now becomes

$$\mathbf{G_j} = \begin{pmatrix} 0 \\ -\tau^l_{1j} - \tau^t_{1j} \\ -\tau^l_{2j} - \tau^t_{2j} \\ -\tau^l_{3j} - \tau^t_{3j} \\ -\tau^l_{ji} u_i - k\frac{\partial T}{\partial x_j} - q^t \end{pmatrix} \tag{3.91}$$

The superscripts ^{l} and ^{t} represent an original laminar quantity and a newly added turbulent quantity respectively.

The first alteration in the diffusive flux vector occurs in the momentum equations where a turbulent viscosity stress term is added to the original laminar viscosity stress contribution. This summation of the laminar viscosity and the turbulent viscosity stress terms is known as the total viscous stress term τ^s_{ij}.

$$\tau^s_{ij} = \tau^l_{ij} + \tau^t_{ij} \tag{3.92}$$

Spalart and Allmaras use the following relation for the turbulent viscous stress in their paper [22].

$$\tau^t_{ij} = v^t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{3.93}$$

As the term v_t is a function of κ the turbulent kinetic energy, Spalart and Allmaras obtain a 'rough approximation' for κ by introducing both a diagonal contribution dependent on the stress part of the previous relation as well as a diagonal contribution dependent on κ.

This approximation leads to Equation (3.94).

$$\tau^t_{ij} = \mu^t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3}\mu^t \frac{\partial u_k}{\partial x_k}\delta_{ij} - \frac{2}{3}\rho\kappa\delta_{ij} \tag{3.94}$$

The diagonal contribution containing the turbulence kinetic energy term "κ" may be assumed to be equal to zero for flows with a thin shear layer. The above authors therefore neglect this term from the above equation for simplicity.

Since the laminar viscosity stress term τ^l_{ij} is calculated using the Boussinesq

Equation (2.2), Equation (3.92) becomes the following.

$$\tau_{ij}^{s} = \mu^{s}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{3.95}$$

Similarly the total viscosity is given by the summation of the laminar viscosity and the turbulent viscosity, these different viscosity terms are once again denoted by the superscripts s, l and t respectively.

$$\mu^{s} = \mu^{l} + \mu^{t}$$

The alteration of the momentum equations is easy to implement into the computer code as essentially only one extra scalar value is needed. This ease of implementation highlights another one of the advantages of choosing to use the Spalart Allmaras one equation turbulence model.

3.6.2 Change in the energy equation

The second alteration to occur to the laminar Navier Stokes equations by incorporating the Spalart Allmaras model is present in the energy equation. An additional heat flux term q^t is added to the diffusive flux vector, evident in Equation (3.91).

$$q^{t} = -\mu^{t}\frac{\mathrm{Cp}}{\mathrm{Pr}_t}\frac{\partial \mathrm{T}}{\partial x_j} \tag{3.96}$$

The term Pr_t is the turbulent Prandtl number which is normally assigned a value of 0.9. This additional heat flux term undergoes non-dimensionalisation along with the rest of the energy equation to form Equation (3.97).

$$\frac{\partial q^{t*}}{\partial x_j^*} = -\mu^{t*}\frac{1}{\mathrm{RePr}_t}\frac{\partial}{\partial x_j^*}\frac{\partial \mathrm{T}^*}{\partial x_j^*} \tag{3.97}$$

The superscript $*$ denotes a non-dimensionalised quantity.

3.6.3 Calculation of the turbulent viscosity

It has been shown that once the turbulent Prandtl number is prescribed the only other variable necessary for the alteration of the momentum and energy equations

is the turbulent eddy viscosity term μ^t, which is calculated by the Spalart Allmaras model through the following procedure.

Firstly the turbulent dynamic viscosity μ^t is related to turbulent kinematic viscosity ν^t as follows

$$\nu_t = \frac{\mu_t}{\rho} \tag{3.98}$$

The turbulent kinematic viscosity ν^t in turn is related to a variable newly introduced by the Spalart Allmaras model. This variable is known as the eddy viscosity and is denoted by the notation $\tilde{\nu}$.

$$\nu_t = \tilde{\nu} f_{vi} \tag{3.99}$$

The coefficient f_{vi} is defined as

$$f_{vi} = \frac{x^3}{x^3 + C_{vi}^3} \tag{3.100}$$

The constant C_{vi} is equal to a value of 7.1, while the variable x is itself a ratio of eddy viscosity to turbulent kinematic viscosity.

$$x = \frac{\tilde{\nu}}{\nu_t}$$

The eddy viscosity itself is obtained by solving the following differential equation.

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} &= c_{b1}\left(1 - f_{t2}\right)\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left(\left(\nu + \tilde{\nu}\right)\frac{\partial \tilde{\nu}}{\partial x_j}\right) + C_{b2}\frac{\partial \tilde{\nu}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j}\right] \\ &- \left[C_{w1} f_w - \frac{C_{b1}}{\tilde{\kappa}^2} f_{t2}\right]\left[\frac{\tilde{\nu}}{d}\right]^2 + f_{t1}\Delta u^2 \end{aligned} \tag{3.101}$$

In order to solve Equation (3.101), the following set of equations (3.102) must first be solved and the empirical constants in Table (3.1) be prescribed.

$$\begin{aligned} \tilde{S} = \tilde{\omega} + \frac{\tilde{\nu}}{\tilde{\kappa}^2 d^2} f_{v2} \quad &, \quad f_{v2} = 1 - \frac{x}{1 + x f_{v1}} \\ f_w = g\left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right]^{\frac{1}{6}} \quad &, \quad g = \gamma + C_{w2}\left(\gamma^6 - \gamma\right) \\ \gamma = \min\left(\frac{\tilde{\nu}}{\tilde{S}\tilde{\kappa}^2 d^2}, 10\right) \quad &, \quad f_{t2} = C_{t3}\exp\left(-C_{t4}x^2\right) \end{aligned}$$

$$f_{t1} = C_{t1} g_t \exp\left(-C_{t2}\frac{\tilde{w}_t^2}{\Delta u^2}\left[d^2 + g_t^2 d_t^2\right]\right) \quad , \quad g_t = \min\left(0.1, \frac{\Delta u}{\tilde{w}_t \Delta x}\right) \tag{3.102}$$

The solution of the above terms in Equation (3.102) requires in turn the calculation of the following variables.

- The calculation of $\tilde{w}$, the vorticity magnitude which is given by

$$\tilde{w} = \left(\left(\frac{\partial u_3}{\partial x_2} - \frac{\partial u_2}{\partial x_3}\right)^2 + \left(\frac{\partial u_1}{\partial x_3} - \frac{\partial u_3}{\partial x_1}\right)^2 + \left(\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}\right)^2\right)^{0.5} \tag{3.103}$$

- The calculation of the variable $\tilde{w}_t$, which is the magnitude of the vorticity present at the trip curve.

- The calculation of the variable Δu, which is the difference in velocity between the considered point in the flow domain and the velocity present at the trip curve.

- The calculation of the variable Δx, which is the size of the surface grid spacing at the trip curve.

- The calculation of the variable d, which is the distance from the considered point in the flow domain to the nearest point on a boundary surface that represents a solid geometry otherwise known as the shortest distance to the wall.

- The calculation of the variable d_t, which is the distance between the considered point in the flow domain and the nearest point located along the trip curve(s) on the solid geometries. The trip curve(s) is the position on the surface that the turbulent boundary layer is assumed to begin.

The empirical constants are to be prescribed are presented in the following table below [22]. The final constant C_{w1} is calculated from the empirical constants by the following formula.

$$C_{w1} = \frac{C_{b1}}{\tilde{\kappa}^2} + \frac{(1 + C_{b2})}{\sigma} \tag{3.104}$$

The procedure in calculating the wall distance variable d will be explained in more detail before discussing the turbulence model any further. An initial wall distance

Constant		Constant		Constant	
C_{b1}	0.1355	σ	$\frac{2}{3}$	C_{w3}	2.0
C_{t1}	1.0	C_{t3}	1.1	c_{b2}	0.622
$\tilde{\kappa}$	0.41	C_{w2}	0.3	C_{v1}	7.1
C_{t2}	2.0	C_{t4}	2.0		

Table 3.1: The Empirical constants for the Spalart Allmaras model

value is set for each point in the domain which is larger than the total span of the domain then the following loop is performed.

- Loop over the solid geometry nodes.
- Calculate the distance between the considered domain node and the current solid geometry node.
- Check if this distance calculated is less than the stored wall distance value.
- Replace the stored distance with the calculated distance, should this distance be less, else take no action should the calculated distance be greater.
- End loop

The wall distance variable d_t is calculated in the same manner with the exception that the loop is over the number of trip points on the solid geometry surfaces rather than the number of solid wall points.
The Spalart Allmaras model is non-dimensionalised before implementation, the non-dimensionalised form of Equation (3.101) is

$$\begin{aligned}\frac{\partial \tilde{\nu}^*}{\partial t^*} + u_j^* \frac{\partial \tilde{\nu}^*}{\partial x_j^*} &= c_{b1}\left(1 - f_{t2}^*\right)\tilde{S}^*\tilde{\nu}^* + \frac{1}{Re}\frac{1}{\sigma}\left[\frac{\partial}{\partial x_j^*}\left(\left(\nu^* + \tilde{\nu}^*\right)\frac{\partial \tilde{\nu}^*}{\partial x_j^*}\right) + C_{b2}\frac{\partial \tilde{\nu}^*}{\partial x_j^*}\frac{\partial \tilde{\nu}^*}{\partial x_j^*}\right] \\ &- \frac{1}{Re}\left[C_{w1} f_w^* - \frac{C_{b1}}{\tilde{\kappa}^2} f_{t2}^*\right]\left[\frac{\tilde{\nu}^*}{d^*}\right]^2 + Re\, f_{t1}^* \Delta u^{*2} \qquad (3.105)\end{aligned}$$

The superscript $*$ denotes a non-dimensionalised quantity, similarly the relations in Equation (3.102) are non-dimensionalised to form the set of relations found in

Equation (3.106).

$$\begin{aligned}
\tilde{S}^* = \tilde{\omega}^* + \frac{1}{Re}\frac{\tilde{\nu}^*}{\tilde{\kappa}^2 d^{*2}} f_{v2}^* \quad &, \quad f_{v2}^* = 1 - \frac{x}{1 + x f_{v1}^*} \\
f_w^* = g^* \left[\frac{1 + C_{w3}^6}{g^{*6} + C_{w3}^6}\right]^{\frac{1}{6}} \quad &, \quad g^* = \gamma^* + C_{w2}\left(\gamma^{*6} - \gamma^*\right) \\
\gamma^* = \min\left(\frac{1}{Re}\frac{\tilde{\nu}^*}{\tilde{S}^*\tilde{\kappa}^2 d^{*2}}, 10\right) \quad &, \quad f_{t2}^* = C_{t3}\exp\left(-C_{t4}x^2\right) \\
f_{t1}^* = C_{t1} g_t^* \exp\left(-C_{t2}\frac{\tilde{w}_t^{*2}}{\Delta u^{*2}}\left[d^{*2} + g_t^{*2} d_t^{*2}\right]\right) \quad &, \quad g_t^* = \min\left(0.1, \frac{\Delta u^*}{\tilde{w}_t^* \Delta x^*}\right)
\end{aligned} \tag{3.106}$$

Lastly the non-dimensionalised form of the vorticity is given by Equation (3.107).

$$\tilde{w}^* = \left(\left(\frac{\partial u_3^*}{\partial x_2^*} - \frac{\partial u_2^*}{\partial x_3^*}\right)^2 + \left(\frac{\partial u_1^*}{\partial x_3^*} - \frac{\partial u_3^*}{\partial x_1^*}\right)^2 + \left(\frac{\partial u_2^*}{\partial x_1^*} - \frac{\partial u_1^*}{\partial x_2^*}\right)^2\right)^{0.5} \tag{3.107}$$

3.6.4 Discretisation

The turbulent eddy viscosity equation is ready to be implemented since the non-dimensionalisation procedure has taken place. The temporal discretisation is once again implemented ultisating a forward difference in time matching the previous temporal treatment of the laminar equations, therefore Equation (3.105) now becomes the following.

Note, that the additional Characteristic Galerkin terms are also included in the formulation but neglected here for the sake of simplicity.

$$\begin{aligned}
\frac{\tilde{\nu}^{n+1} - \tilde{\nu}^n}{\Delta t} + u_j^n \frac{\partial \tilde{\nu}^n}{\partial x_j} &= c_{b1}\left(1 - f_{t2}\right)\tilde{S}\tilde{\nu}^n + \frac{1}{Re}\frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left(\left(\nu + \tilde{\nu}\right)\frac{\partial \tilde{\nu}}{\partial x_j}\right)^n + C_{b2}\frac{\partial \tilde{\nu}^n}{\partial x_j}\frac{\partial \tilde{\nu}^n}{\partial x_j}\right] \\
&- \frac{1}{Re}\left[C_{w1} f_w - \frac{C_{b1}}{\tilde{\kappa}^2} f_{t2}\right]\left[\frac{\tilde{\nu}^n}{d}\right]^2 + Re\; f_{t1}\Delta u^{n2}
\end{aligned} \tag{3.108}$$

Utilising Galerkin procedure to perform the spatial discretisation of the equation results in the Galerkin weak form of Equation (3.109) where the bold font again denotes a vector quantity.

$$\begin{aligned}\int_\Omega \mathbf{N}^T \Delta\tilde{\nu} d\Omega = \; & \Delta t \left[-\int_\Omega \mathbf{N}^T u_j \frac{\partial \tilde{\nu}}{\partial x_j} d\Omega + \int_\Omega \mathbf{N}^T c_{b1} \left(1 - f_{t2}\right) \tilde{S}\tilde{\nu} \, d\Omega \right] \\ + \; & \Delta t \left[\frac{1}{\sigma Re} \int_\Omega \mathbf{N}^T \left[\frac{\partial}{\partial x_j} \left(\left(\nu + \tilde{\nu}\right) \frac{\partial \tilde{\nu}}{\partial x_j} \right) + C_{b2} \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} \right] d\Omega \right] \\ + \; & \Delta t \left[\int_\Omega \mathbf{N}^T Re \; f_{t1} \Delta u^2 d\Omega - \frac{1}{Re} \int_\Omega \mathbf{N}^T \left[C_{w1} f_w - \frac{C_{b1}}{\tilde{\kappa}^2} f_{t2} \right] \left[\frac{\tilde{\nu}}{d} \right]^2 d\Omega \right]\end{aligned} \tag{3.109}$$

Performing integration by parts on the first part of the diffusion term results in Equation (3.110).

$$\begin{aligned}\int_\Omega \mathbf{N}^T \Delta\tilde{\nu} d\Omega = \; & \Delta t \left[-\int_\Omega \mathbf{N}^T u_j \frac{\partial \tilde{\nu}}{\partial x_j} d\Omega + \int_\Omega \mathbf{N}^T c_{b1} \left(1 - f_{t2}\right) \tilde{S}\tilde{\nu} \, d\Omega \right] \\ + \; & \Delta t \left[\frac{1}{\sigma Re} \int_\Gamma \mathbf{N}^T \left(\nu + \tilde{\nu}\right) \frac{\partial \tilde{\nu}}{\partial x_j} d\Gamma - \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j} \left(\nu + \tilde{\nu}\right) \frac{\partial \tilde{\nu}}{\partial x_j} d\Omega \right] \\ + \; & \Delta t \left[\frac{1}{\sigma Re} \int_\Omega \mathbf{N}^T C_{b2} \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} d\Omega \right] \\ + \; & \Delta t \left[\int_\Omega \mathbf{N}^T Re \; f_{t1} \Delta u^2 d\Omega - \frac{1}{Re} \int_\Omega \mathbf{N}^T \left[C_{w1} f_w - \frac{C_{b1}}{\tilde{\kappa}^2} f_{t2} \right] \left[\frac{\tilde{\nu}}{d} \right]^2 d\Omega \right]\end{aligned} \tag{3.110}$$

At this stage the continuous eddy viscosity field variable in Equation (3.110) is replaced with the nodal approximation $\tilde{\nu} = \mathbf{N}\overline{\tilde{\nu}}$, the resultant matrix form of the eddy viscosity equation is given below.
An overbar denotes a nodal quantity as before.

$$\mathbf{M}_t \overline{\Delta\tilde{\nu}} = \Delta t \left[\mathbf{C}_t - \mathbf{K}_t + \mathbf{f}_t + \mathbf{S} \right] \tag{3.111}$$

The component matrix terms are as follows, firstly the mass matrix $\mathbf{M}_t$ is given by the following. The mass matrix is lumped as before in order to retain a matrix free solution

$$\mathbf{M}_t = \int_\Omega \mathbf{N}^T \mathbf{N} d\Omega = \frac{V}{20} \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \approx \frac{V}{4} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The convection term $\mathbf{C}_t$ is given as the following where the summation j is over the number of coordinate directions.

$$\mathbf{C}_t = \mathbf{u}_j^e \int_\Omega \mathbf{N}^T \frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \bar{\tilde{\nu}} d\Omega = \mathbf{u}_j^e \frac{V}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \sum_{s=1}^{nodes} \left[\frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \bar{\tilde{\nu}} \right]^s \quad j = 1,2,3 \tag{3.112}$$

The superscript e denotes an elemental averaged quantity, the diffusion term $\mathbf{K}_t$ is given as

$$\begin{aligned} \mathbf{K}_t &= \frac{1}{\sigma Re} \left[(\nu + \tilde{\nu})^e \int_\Omega \frac{\partial \mathbf{N}^T}{\partial \mathbf{x}_j} \frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \bar{\tilde{\nu}} d\Omega - C_{b2} \int_\Omega \mathbf{N}^T \left(\frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \bar{\tilde{\nu}} \right)^2 d\Omega \right] \\ &= \frac{1}{\sigma Re} (\nu + \tilde{\nu})^e V \begin{bmatrix} \frac{\partial N_1}{\partial x_j} \\ \frac{\partial N_2}{\partial x_j} \\ \frac{\partial N_3}{\partial x_j} \\ \frac{\partial N_4}{\partial x_j} \end{bmatrix} \sum_{s=1}^{nodes} \left[\frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \bar{\tilde{\nu}} \right]^s \\ &\quad - \frac{C_{b2}}{\sigma Re} \frac{V}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \left(\sum_{s=1}^{nodes} \left[\frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \bar{\tilde{\nu}} \right]^s \right)^2 \quad j = 1,2,3 \end{aligned} \tag{3.113}$$

The boundary term $\mathbf{f}_t$ which is generated from the integration by parts of the diffusion term is given as

$$\begin{aligned} \mathbf{f}_t &= \frac{1}{\sigma Re} (\nu + \tilde{\nu})^e \int_\Gamma \mathbf{N}^T \frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \bar{\tilde{\nu}} d\Gamma \\ &= \frac{1}{\sigma Re} (\nu + \tilde{\nu})^e \frac{A}{3} \mathbf{Cos}\alpha_j \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \sum_{s=1}^{nodes} \left[\frac{\partial \mathbf{N}}{\partial \mathbf{x}_j} \right]^s \quad j = 1,2,3 \end{aligned} \tag{3.114}$$

Lastly the source term $\mathbf{S}_t$ is given by Equation (3.115).

$$\begin{aligned}
\mathbf{S}_t &= \left\{c_{b1}\left(1-f_{t2}\right)\tilde{S}-\frac{1}{Re}\left[C_{w1}f_w-\frac{C_{b1}}{\tilde{\kappa}^2}f_{t2}\right]\left[\frac{\tilde{\nu}^e}{d^2}\right]\right\}\int_\Omega \mathbf{N}^T\mathbf{N}\bar{\tilde{\nu}}d\Omega + Re\, f_{t1}\Delta u^2\int_\Omega \mathbf{N}^T d\Omega \\
&= \left\{c_{b1}\left(1-f_{t2}\right)\tilde{S}-\frac{1}{Re}\left[C_{w1}f_w-\frac{C_{b1}}{\tilde{\kappa}^2}f_{t2}\right]\left[\frac{\tilde{\nu}^e}{d^2}\right]\right\}\frac{V}{20}\begin{bmatrix} 2\bar{\tilde{\nu}}^a+\bar{\tilde{\nu}}^b+\bar{\tilde{\nu}}^c+\bar{\tilde{\nu}}^d \\ \bar{\tilde{\nu}}^a+2\bar{\tilde{\nu}}^b+\bar{\tilde{\nu}}^c+\bar{\tilde{\nu}}^d \\ \bar{\tilde{\nu}}^a+\bar{\tilde{\nu}}^b+2\bar{\tilde{\nu}}^c+\bar{\tilde{\nu}}^d \\ \bar{\tilde{\nu}}^a+\bar{\tilde{\nu}}^b+\bar{\tilde{\nu}}^c+2\bar{\tilde{\nu}}^d \end{bmatrix} \\
&+ Re\, f_{t1}\Delta u^2\frac{V}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
\end{aligned} \tag{3.115}$$

The superscripts a b c and d denotes the four nodes of the tetrahedral element.

3.6.5 Boundary and Initial conditions

Finally the imposed boundary conditions of the Spalart Allmaras turbulence model shall now be discussed. Firstly the value of eddy viscosity $\tilde{\nu}$ at a inlet boundary must be prescribed and in this dissertation the prescription is a value of 10% that of the prescribed value for the laminar viscosity. This percentage relation of the prescribed viscosity values is the suggested boundary condition by Spalart and Allmaras in their paper. This prescription of 10% is furthermore also utilised as the initial condition for the eddy viscosity field throughout the domain at the commencement of a new computational simulation.

At a solid surface or a symmetric boundary, the value of turbulent viscosity ν is set to zero for all the nodes present on these boundaries.

At outlet boundaries for subsonic flow cases, Spalart and Allmaras recommend that the value prescribed on the boundary for the eddy viscosity be extrapolated from values taken inside the domain unless a large domain is used so that the possibility of any inaccuracies percolating back into the domain are prevented. Since the outlet boundary is at a minimum distance of ten chord lengths from the airfoil in the meshes used in this dissertation, it was decided that this distance of ten chord lengths was large enough to avoid the requirement of prescribing the

eddy viscosity at the exit.

3.7 Chapter Summary

In this chapter, the mathematical formulation of the Characteristic Based Scheme (CBS) has been examined in entirity from the starting point of the non-dimensionalised Navier Stokes equations. In addition both the temporal and spatial discretisation of the CBS scheme has been explained in depth.

Furthermore the additional shock capturing procedure used in this dissertation that utilises artificial diffusion has been explained in this chapter. The principle of adding variable smoothing to a computational scheme that consists a coupled energy equation has also been explained.

Finally, the details for modeling turbulence with the one equation Spalart Allmaras model have been included in this chapter. Every aspect of the Spalart Allmaras model has been mentioned from the alterations in the diffusive flux vector of the Navier Stokes equations to the non-dimensionalisation and discretisation of the additional equation necessary to find the eddy viscosity $\tilde{\nu}$. The boundary conditions and initial conditions for the tubulence model have also been stated in the penultimate section.

Chapter 4

Program implementation

The purpose of this chapter is to give an outline of the manner, in which, the CBS code was implemented computationally. The computational code may be broken into three basic modules or stages which are identified as the preprocessing, solution and postprocessing stages. The computational code was written using the Fortran programming language and compiled using the Fortran compiler [49, 50].

The layout for the computational code consists of a number of successive operations carried out for each element or edge of the domain in what is known as a 'loop over elements' or a 'loop over edges'. In this chapter, a serial element based version of the code will be used for the demonstration purpose rather than an edge based version. (The parallel implementation of the code will be discussed later in the next chapter.)

4.1 Preprocessing Module

Firstly the actions undertaken by the code in the preprocessing stage of the program will be considered. The constituent actions of preprocessing are listed below.

- All the required parameters for the successful modeling of the problem are read into the computer memory by the program from input files. The input protocol itself involves the use of four input files, these four files are a file containing the mesh geometry, a file containing the surface boundary flags, a file containing the current unknowns in the case of a computational restart

and lastly, a control file containing all the necessary run parameters such as the free stream values which will be used to initialize certain arrays.

- The next step undertaken by the computational code involves identifying the mesh elements that exist on a boundary surface of the mesh domain. These identified mesh elements are then checked to see whether the mesh element exists on more than one boundary or in other words, the mesh element is checked to see whether one or more of the elemental faces exist on different boundary surfaces of the domain. Those mesh elements that do exist on one or more boundary surfaces are then stored by the program along with their corresponding boundary face or faces.

- The variables are then changed from non-conservation to conservation form, thus obtaining the values of the momentum component variables.

- The elemental shape function derivatives along with the corresponding lumped mass matrix for each mesh element are then calculated and stored for later use in the solution loops. These calculations are undertaken at this stage and not later in the solution stage in order to avoid unnecessarily performing the calculations more than once.

- Additionally, the direction cosines are calculated for the normal of each face that exists on a boundary surface. The direction cosines are allocated to their parent element and stored.

- The nodes that exist on the boundary surfaces that are solid wall geometries are identified and stored, accomplished by looping over the boundary faces and assembling the points found in any boundary face that contain the solid surface boundary flag read from the input file, the averaged normals of these solid wall faces are also calculated and stored.

- Depending on which turbulence model is being incorporated into the version of the code if at all, the necessary turbulent preprocessing is also undertaken at this stage. An example in the case of the one equation Spalart Allmaras turbulence model described earlier, the nearest wall distance for each node in the domain is calculated.

4.2 The Solution Module

The next stage of the computer code to be considered is the second solution module which involves both the solution algorithm containing the CBS scheme and the turbulence model placed inside a solution loop. A computational check against a specified convergence tolerance in order to determine whether a steady state solution has been found is also included in this module.

- The first task of the solution loop is to transfer the values calculated from the previous computational iteration to the designated arrays that store the variable values from the previous timestep.
- The decoupled pressure field is also updated at this point from the updated values of energy calculated by the energy equation in the previous computational iteration.
- The intermediate momentum conservation variables are then calculated, this calculation is achieved in the following manner, firstly the magnitude is calculated for every one of nodal time steps and then stored in the computer memory. Secondly, the inviscid fluxes and the diffusive fluxes are calculated should the flow be viscous and are added to the main equation along with the calculated artificial diffusion.
- Once this stage has been reached, step two of the CBS scheme is then implemented. The solution of the density values is found from the conservation of mass equation after it has been in turn computed from the intermediate momentum conservation variables calculated from the previous step. Artificial diffusion is added before the density values are updated
- Step three of the CBS scheme is undertaken by the code by correcting the momentum variables using the pressure values that were found using the density values calculated from the previous step of the CBS scheme.
- At this stage of the solution cycle, it is necessary to impose the symmetry, far field and wall specific boundary conditions to the variable values that were calculated from the previous three steps of the CBS scheme. Also the pressure field is updated at this point, by using the newly calculated

momentum variables. A vorticity correction may also be necessary as a boundary condition on the far field boundary depending on the size of domain used.

- Step four, the final step of the CBS scheme is then implemented at this point, the calculation of total energy from the energy equation. The energy is calculated by the implementation of the following procedure, the inviscid flux for the energy equation is calculated along with the corresponding diffusive flux should the flow be of viscous nature. Should artificial diffusion be present, it is added at this stage to either the inviscid or diffusive fluxes before updating the values of temperature and energy. The energy boundary condition is then applied to the energy values calculated by the code.

- The next action in the solution stage is to update the values of the turbulence variables, this will not be discussed in detail as the turbulence variables depend on which turbulence model is being implemented to the code. If boundary conditions exist for the turbulence model, the conditions are also applied by the code at this point.

- A full computational iteration of the solution loop has been completed by the code once this stage has been reached, therefore the code must now be checked to determine whether the simulation has converged to steady state solution. The convergence check is performed by comparing the value of the L2 norm of the density residual to that of a previously prescribed value. Depending on the result of this convergence check, the solution loop is either repeated should the residual be higher than the prescribed value or alternatively the program advances to the postprocessing stage should the residual is lower than the prescribed value.

4.3 Postprocessing

Once the solution has passed the convergence check or the solution stage of the code has completed a prescribed maximum number of computational iterations, the program enters the final module of the code identified as the postprocessing stage where the following actions are undertaken by the code.

- The pressure distribution C_p is calculated along the surface of the solid geometries that represent the airfoil under consideration and written to an output file.

- If the computational simulation is of a viscous flow case, then the skin friction C_f along the solid geometries representing the aerofoil is also calculated and then written to an external output file.

- Output files are also generated should the a restart of the solution be needed at a future date. A restart would be necessary should the user wish to converge the solution to a finer degree or alternatively a user may have other reasons for desiring a restart. The most recently calculated values for the density velocity and energy variables are written to this restart data file.

- Finally the unknown variables in the Navier Stokes equations (density, the velocity components and energy) along with the pressure are written to an external file in the required format that a external visualizer and plotter require to read and plot these values.

4.4 Chapter Summary

This is a short chapter explaining how the computational algorithm is implemented in the computer code for the preprocessing, solution and postprocessing stages. It is hoped that the reader has developed a basic understanding on the structure of the explicit CBS solver from this chapter.

Chapter 5

Techniques to acquire a faster solution

5.1 Overview

In this chapter, some computational techniques will be examined with the aim of obtaining a solution both quicker in real time and at a reduced computational cost. The advantages of achieving a faster solution becomes more pronounced as larger and more increasingly complex three dimensional problems are considered which would otherwise have required a run time of several days using a conventional code.

Two such computational techniques will be examined in this chapter with a view of obtaining lower computational cost. These two techniques are the edge based data structure and the application of code parallelisation.

5.2 The edge based data structure

5.2.1 Formulation

The edge based data structure[51] has been used in many problems including finite element and finite volume problems related to CFD based simulations. Due to the many advantages the edge based format posses, including the speeding up in real time of solution calculation and a lower usage of computer memory than the more traditional element based memory structure when considering certain types of flow

cases, edge based structure has become increasingly widespread throughout the CFD community in recent years.

The edge based data structuring is more suitable to an Euler flow case since as modeling the viscous terms of the full Navier Stokes equations possesses additional problems. The problem posed by use of the edge based structure centers around the treatment of the elemental quantities introduced during the standard Galerkin discretisation process of the momentum and energy equations discussed earlier in the CBS theory chapter.

In order to incorporate the edge based structure for viscous flows, a choice must be taken to either compute the aforementioned elemental quantities or use an alternatively more complicated discretisation procedure. Clearly, there will be a significant computational cost regardless of which one of the above options is used in determining a viscous solution.

The application of the edge based technique to a two dimensional Euler flow problem modeled by triangular elements will be considered to mathematically outline the format of the edge based data structure. Referring to problem theory chapter for a mathematical starting point, the Euler flow equations may be written again as the following.

$$\frac{\partial \mathbf{R}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = 0 \tag{5.1}$$

The term $\mathbf{R}$ denotes the conservation variable vector while the term $\mathbf{F}$ denotes the convective Euler flux vector. Applying Galerkin spatial approximation to the above equation, the element based formulation of the weak form may be written as Equation (5.2) below should an interior node be considered and assuming any stabilisation terms are omitted. The elemental shape functions are denoted by the term $\mathbf{N}$ and the terms in bold font denote vector quantities as these quantities were denoted before in the earlier theory chapter.

$$\int_\Omega (\mathbf{N})^T \frac{\partial R}{\partial t} d\Omega = -\int_\Omega (\mathbf{N})^T \frac{\partial F_i}{\partial x_i} d\Omega \tag{5.2}$$

As the fully explicit form of the CBS split solution procedure is considered in this dissertation, the left hand side of Equation (5.2) simply becomes $\mathbf{M}\Delta\mathbf{R}/\Delta t$. The right hand side of the equation for any node $\mathbf{I}$, may be rewritten in the form below for any interior node in the domain by interpolating F_i in each element after

Green's theorem is applied.

$$\sum_{E\epsilon I}\int_{A_E}\frac{\partial N^I}{\partial x_i}\left(N^n\mathbf{F}_i^n\right)d\Omega=\sum_{E\epsilon I}\left[\frac{A_E}{3}\frac{\partial N_I}{\partial x_i}\right]_E\left(F_i^a+F_i^b+F_i^c\right) \tag{5.3}$$

The term denoted by A is the area of any triangular element consisting of nodes with a b and c, while the subscript E denotes the element identity. In the case of

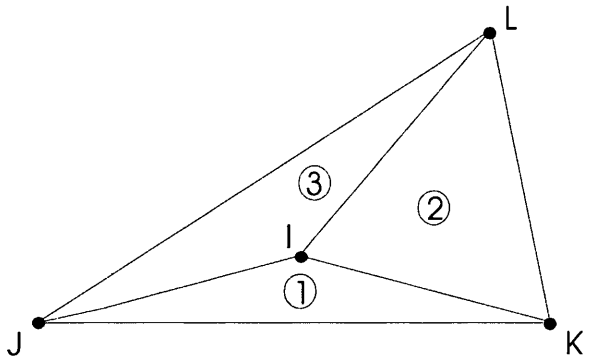


Figure 5.1: Inside Node and surrounding nodes

the interior node illustrated in Figure 5.1, the global nodes a b and c are replaced by three elements with local nodes I J K and L. The right hand side of the applied above elemental relation is now expanded as

$$\frac{A_1}{3}\frac{\partial N_I}{\partial x_i}\left(F_i^I+F_i^J+F_i^K\right)+\frac{A_2}{3}\frac{\partial N_I}{\partial x_i}\left(F_i^I+F_i^K+F_i^L\right)+\frac{A_3}{3}\frac{\partial N_I}{\partial x_i}\left(F_i^I+F_i^L+F_i^J\right) \tag{5.4}$$

The corresponding equation formulation used to produce the spatial approximation for this interior node using the edge based connectivity formulation is given in Equation (5.5) below.

$$\sum_{E\epsilon I}\int_{\Omega_E}\left[\frac{\partial N^I}{\partial x_i}\right]_E\left(N^n\mathbf{F}_i^n\right)d\Omega=\sum_{S=1}^{m_s}\left\{\sum_{E\epsilon II_s}\left[\frac{A_E}{3}\frac{\partial N_I}{\partial x_i}\right]_E\left(F_i^I+F_i^{I_s}\right)\right\} \tag{5.5}$$

The summation S in this equation is over the maximum number of edges connected to the node I, while the second summation $E\epsilon II_s$ represents the elements containing this edge with first node I and second node I_s.

The flux coefficient in this equation can be shown to be equal to zero for an interior node when the summation is over the total number of edges and elements giving the relation shown below.

$$\sum_{S=1}^{m_s} \sum_{E \epsilon I I_s} \left[\frac{A_E}{3} \frac{\partial N_I}{\partial x_i} \right]_E = 0 \tag{5.6}$$

As the flux coefficient above can be shown to be zero, Equation (5.5) can be said to be exactly identical to the corresponding elemental based formulation for an interior node.

The above procedure shows the manner in which an internal node is considered by the edge based method, a boundary node will now be treated in a similar manner by the method.

Considering again the discretisation of an Euler flow case from the starting point of Equation (5.1) when the equation is relating to a boundary node instead of an interior node, the right hand side of the equation becomes the following for any boundary node in the domain using elemental formulation after F_i is interpolated and after Green's theorem is applied (From Figure 5.2).

$$\sum_{B \epsilon I} \int_{\Gamma_B} N^I \left(N^n \mathbf{F}_i^n \right) d\Gamma n_B = - \sum_{B \epsilon I} \left[\frac{\Gamma_B}{6} \left(2F_i^a + F_i^b \right) \mathbf{n} \right]_B \tag{5.7}$$

The term $\mathbf{n}$ denotes the normal to the boundary edge, where the superscript a denotes the node under consideration and the superscript b denotes the second node forming the boundary edge.

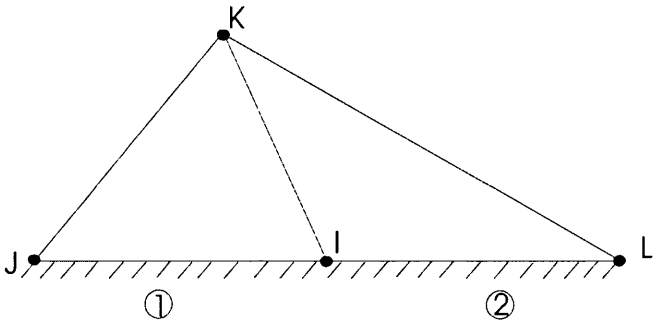


Figure 5.2: A Boundary Node and surrounding nodes

Let the boundary node I illustrated by Figure 5.2 now be considered. The right hand side of Equation (5.7) can be written in the expanded form below. Note that the terms Γ_{B1} and Γ_{B2} denote the corresponding edge lengths and the terms n_1 and n_2 denote the corresponding edge normals.

$$-\frac{\Gamma_{B1}}{6}\left(2F_i^I + F_i^J\right)\mathbf{n}_1 - \frac{\Gamma_{B2}}{6}\left(2F_i^I + F_i^L\right)\mathbf{n}_2 \tag{5.8}$$

It can be shown for a boundary node that the flux coefficient calculated by the edge based formulation for a boundary node in Equation (5.5) will result in a non zero value.

$$\sum_{S=1}^{m_s}\sum_{E\varepsilon II_s}\left[\frac{A_E}{3}\frac{\partial N_I}{\partial x_i}\right]_E \neq 0 \tag{5.9}$$

In order to calculate the flux coefficient as a zero value, the following correction is made to Equation (5.5) to form the end edge based structured equation for a boundary node which is given by the following equation.

$$\sum_{B\varepsilon I}\int_{\Gamma_B} N^I\left(N^n\mathbf{F}_i^n\right)d\Gamma n_B = \sum_{S=1}^{m_s}\left\{\sum_{E\varepsilon II_s}\left[\frac{A_E}{3}\frac{\partial N_I}{\partial x_i}\right]_E\left(F_i^I + F_i^{I_s}\right)\right\} - \frac{\Gamma_{B1}}{2}F_i^I\mathbf{n}_1 - \frac{\Gamma_{B2}}{2}F_i^I\mathbf{n}_2 \tag{5.10}$$

This equation can now be shown to be identical to the elemental based formulation when a boundary node is considered [29].

5.2.2 Performance

To compare the performance of the edge based formulation verus the performance of the elemental based formulation, a comparison was made between the run time taken to compute a set amount of computational iterations. A three dimensional domain of 18,356 nodes was adopted for use in this comparison, which was duely arranged into either 101,930 tetrahedral elements and 123,361 edges for the relative formulation cases. The domain used was an unstructured mesh which represented a NACA0012 aerofoil.

Once the formulations was completed, an inviscid problem was run on both the edge based and the elemental based organisations of this domain. The run time durations resulting from the different computational iterations are compared in Table (5.1).

To validate that the implementation of the edge based method was successful, the differences after one computational iteration between the density variable values were calculated between the different connectivity formulations. The maximum observed difference was of the order of magnitude of 10^{-7} using a single precision program code. It is therefore assumed that discrepancies between the methods are negligible and caused by truncation errors from the machine. This justification is due to the tiny size of the observed error values. Note that after the completion of the 5000 computational iterations, the maximum observed difference of variable values had risen to not more than an order of magnitude to 10^{-6}. This rise in the differences of values is attributed to a cumulative effect of the aforementioned truncation errors. The error magnitude was itself evaluated by a post processing program which compared the variable values printed in outputted data files.

Iterations	Edge time (s)	elemental time (s)	Scaling %
100	44.3	45.3	98
200	85.4	89.5	95
500	205.5	218.9	94
1000	392.8	421.3	93
5000	1960.3	2096.4	93

Table 5.1: Time analysis for the edge based formulation verus the elemental formulation. The scaling column portrays the edge based formulation run time as a percentage of the elemental formulation run time

As is evident from Table (5.1), the edge based formulation is observed to be quicker than the elemental based formulation, however the margin of speed up is not of a significant level. It is also evident from Table (5.1) that the observed margin of speed up increases as the number of computational iterations increase. The margin of speed up appears to converge asymptotically to a value of 93%. The low margin of speed up is attributed to the high preprocessing cost of the elemental based quantities needed by both the elemental and edge based formulations. An example of one of these elemental quantities being the derivatives of the

elemental shape functions. As more computational iterations are completed, the pre-processing stage has a lower influence on the total run time.

It was therefore concluded not to continue with the implementation of the edge based formulation of the Characteristic Based Split due to the low margin of speed up observed and the difficulties in extending the edge based formulation to a viscous flow case mentioned earlier. An additional computational expense would of course be created from the necessary treatment of the elemental quantities introduced by the discretisation process of the momentum and energy equations or resultant from an alternative discretisation of these equations. Any extra computational expense would further negatively effect the already observed low margin of speed up to an even smaller value or if the negative effect was extreme enough, it may cause the edge based formulation to have a slower run time than the elemental formulation defeating the objective of using a speed up technique.

5.3 Parallelization

5.3.1 Introduction

This section will explain the second speed up technique considered in this dissertation, the concept of parallel computing and detail the application of code parallelisation to the explicit Characteristic Based Split scheme. The use of parallel computing and the development of the high performance parallel computing platforms necessary for the operation of the parallel coding architectures allows the solution of much larger problems which would themselves have been impractical to solve on simpler machines that only use a single processor.

The principle behind the running of these larger sized simulations on parallel computing platforms which are also known as distributed memory machines, is to achieve a solution of the overall problem by dividing it into a smaller number of sub-problems. The parallel architecture then allocates a separate processor to solve each one of these sub-problems which can be solved in less real time, this principle of allocation to separate processors is illustrated in Figure (5.3).

The reductions in the necessary run time needed to solve the computational problem is acquired not only by the obvious splitting of the problem and spreading of the workload over several simultaneously running processors, but also from a

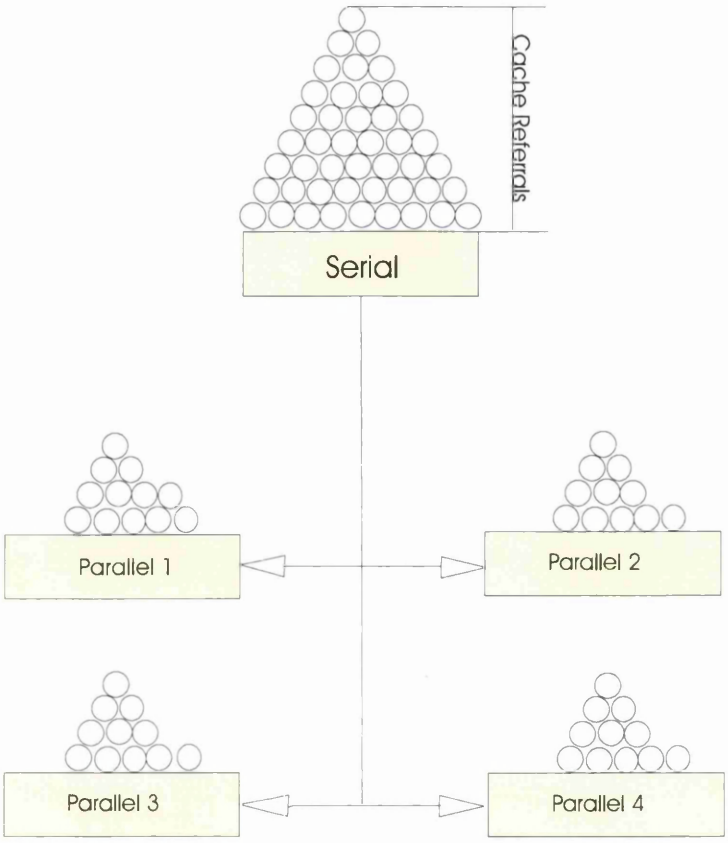


Figure 5.3: Representation of load distribution amongst processors

feature present from the manner in which computers are constructed.

To explain this fortuitous time saving feature, it is first brought to attention that the amount of memory that each processor must handle is reduced due to the smaller workload it is assigned. This smaller amount of the memory allows a much more efficient handling of the data as a larger percentage of the total required memory initialised by the problem is stored in the cache and the allocated RAM of the parallel processors which both remain constant regardless of the amount of assigned workload. This more efficient handling of memory will allow in some cases the occurrence known as super-linear speedup to propagate.

Super-linear speedup is defined as the level of speedup that occurs when the run duration of the parallel code is lower than the run duration of the serial code divided by the total amount of processors used by the parallel code. The concept

of super linear speedup is best explained by considering the following example.

In this example, let the computational problem require a run time of 80 minutes duration when run on a serial code to obtain the solution. The problem is then split and allocated by parallel architecture to a total of eight different processors and the corresponding parallel run time is also duely noted. In theory the run time of the parallel code should be equal to a minimum possible value of time which is calculated by dividing the original serial run time of 80 minutes by the eight processors present in the parallel architecture resulting in a minimum possible value of 10 minutes. This value of 10 minutes is stated as a minimum value, because in practise an exact 100 percent efficiency may not be obtained due to the demand that some additional time will be needed by the parallel code for the individual processors to communicate with each other in order to transfer data from subproblem to subproblem. Fortunately, run time is saved by the more efficient hardware handling of memory when a parallel code is run as was mentioned in the previous paragraph. This more efficient handling of memory can possibly offset the extra run time required by the processors in communicating to one another. Should the parallel code be observed in our example to require a run time of only nine minutes, this demonstrates that the more efficient memory handling did indeed save more time than the extra inter-processor communications cost, thus causing the run time of the parallel code to be less than the theoretical minimum of ten minutes. A lower than ten minute run time for in this example reflects a greater than 100% efficiency in the margin of speed up, this occurrence of a greater than 100% efficiency is referred to as super-linear speedup.

The solution of the parallelised problem is made possible by calculating the variables in each one of the sub divided problem parts in the manner that these variables would have normally been calculated by the serial code and then the communication of results between these subproblems.

An update of the variable values is necessary after each one of the CBS algorithm time steps for the nodal values that are present along the boundaries between the different sub divided problem domains. This updating of variables is achieved by allowing communication to occur between the two identified processors handling the two sub-problems existing at either side of the specific boundary considered. As the processors have been identified with the corresponding boundaries,

an individual processor will only communicate with another processor should the sub-problem allocated to the first processor share a boundary with the sub-problem allocated to the second processor. This communication condition is explained in Figure (5.4). The second processor from the left in Figure (5.4) handling the green marked sub-problem will communicate only with the first processor from the left handling the blue sub-problem and the third processor from the left handling the purple sub-problem. The second processor will not communicate with the fourth processor from the left handling the sub-problem denoted by the red colour as there is no boundary shared between these two processors.

The library of parallel commands adopted for these inter processor communications in addition to providing the other necessary parallel instructions was the well known Message Passing Interface (M.P.I.) Library. The original Message Passing Interface library was written in 1994 by a combined group of researchers incorporating representatives from both academia, government and industry. The version of M.P.I used in this dissertation consists of a 32 bitt library containing 129 functions that may be used in either C, C++ and Fortran programming languages and is flexible enough to be included by either one of the various Fortran compilers. The M.P.I. library was one of the first standards used for programming parallel computers and was the first parallel architecture developed that was based on message passing between the different processors[52].

Each individual inter-processor communication or message passing of a single scalar variable within the parallel architecture is unfortunately several orders of magnitude slower on most types of parallel platforms than the time taken to perform a simple floating point operation calculated within a single processor. The amount of inter-processor communications must therefore be kept to a minimum when organising the parallel code structure to avoid performing any of these slower inter-processor communications which may be superfluous or duplicated. This attribute of message passing also infers that certain numerical schemes will be ill suited in relative terms to undergo a parallelization procedure, such schemes would include highly implicit schemes such as those numerical schemes that incorporate direct Gaussian elimination.

Contrastingly, numerical schemes that imply explicit iterative techniques such as the Characteristic Based Split scheme incorporating an explicit fractional step

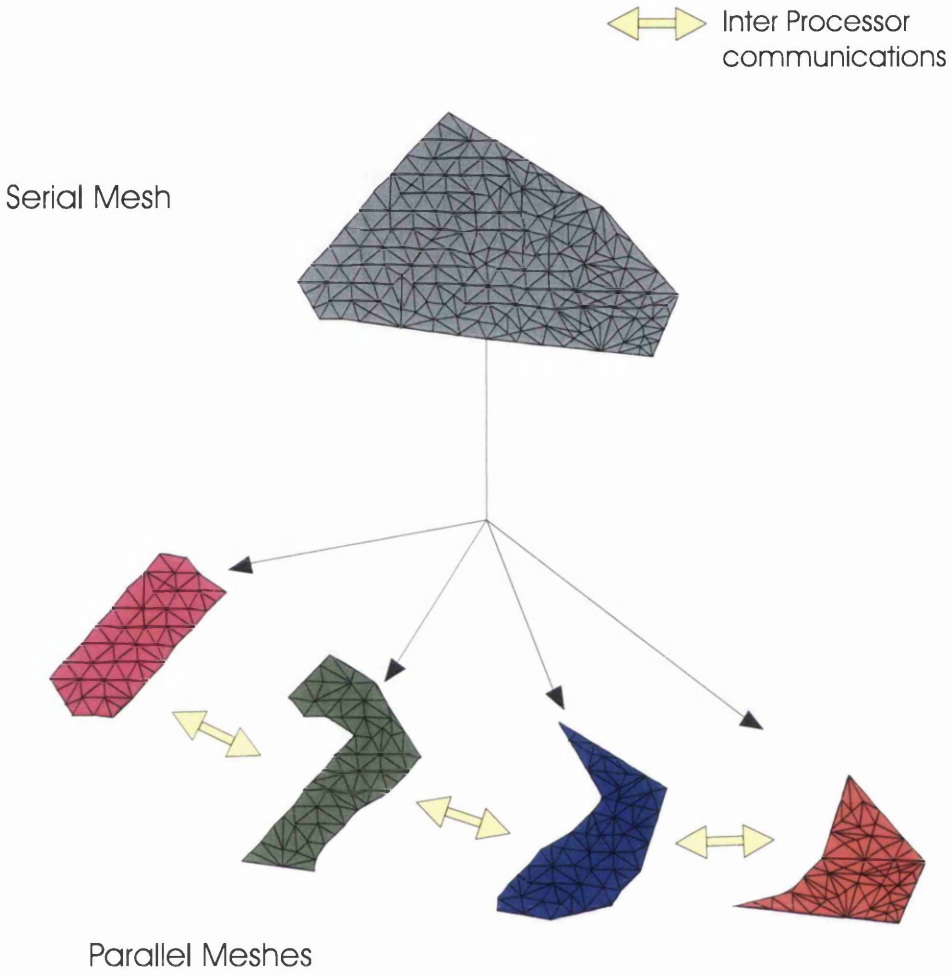


Figure 5.4: Domain decomposition and inter processor communication

are relatively simple to parallelise as the communication between the processors of a large implicit matrix is not involved.

5.3.2 Advanced Data Format

In order to prepare the problem under consideration for parallelization, the splitting of the problem into required sub-problems determines that the computational domain must also be split into corresponding sub-domains.

Rather than use a dedicated parallel mesh generator to create a dedicated parallel mesh directly, a decision was made instead to use a domain splitting technique

on an existing mesh previously created for and solved by the serial version of the code. The decision to use an existing mesh originally generated for serial use, possess the advantage of allowing a post-run comparison of variable values and run times calculated by the parallel code to those previously calculated by the serial code considering the same domain. This comparison of the two sets of variable values will therefore allow the possibility to validate the accuracy of the parallelisation procedure.

As sub-domains require the storage of the additional data to allow inter-processor communications and the necessary global nodal identifications, the standard mesh data file layout used by the serial code of connectivities, coordinates and boundary information is now insufficient. To store the original data that was contained in the standard mesh file along with the additional necessary inter processor communication and global information data, the author decided to adopt the highly versatile Advanced Data Format or (ADF) database library structure for use in this dissertation.

The A.D.F. file structure itself consists of a library of I/O subroutines which store information in a relatively simple hierarchical tree structured database rather than the standard linear data format of the traditional data files. An analogy may be made by likening the A.D.F. file structure to the root directory of a computer, the user may close which information to open at any time in any specific order that the user chooses. The linear data format of the traditional data file structure has a greater similarity to reading a book that is omitting any form of index, since all the information from the first page must be read by the user until the desired information is reached and found.

This greater freedom in choosing which order to read information from the A.D.F. file structure allows a greater flexibility in constructing the parallel code as no unnecessary computational cost will be spent by each one of the different processors in the parallel architecture reading unnecessary input data. The A.D.F. file structure is also described as being a self-describing file structure, meaning that the information stored in the file structure may be advantageously browsed or read by the user without the requirement of another external third party package. Another additional advantage to choosing the ADF structure is the ability of this file structure to add or remove additional information anywhere in the data tree

at a later incidence without the need of modifying the ADF compatible input subroutines found in any existing Fortran source codes.

A brief synopsis of the ADF file structure's background will be given in this paragraph, the ADF structure was originally developed as part of the CFD General Notation System or (CGNS) project[53]. The aims of this project were to define a new standard of information models for use in all Navier Stoke's equation based CFD technology, standardize the I/O information structure of the information models and develop all the necessary software that would be required in allowing the easy implementation of these aforementioned models into all existing and any future computer codes.

The operating system behind the CGNS protocol consists of the series of standard accepted conventions for the input and output of CFD data. A proposed adherence to the conventions of the CGNS protocol by all parties was intended to facilitate the exchange of CFD data between the various diverse sites interested in CFD. The adoption of an universal standard for the treatment of input/output data files would allow an ease of communication between the various different application codes such as the CFD solvers and grid generators across the multitude of different computing platforms available and expected.

The data models that were defined by the CGNS project are individually known as Standard Interface Data Structures or SIDS. After the project defined these models, the CGNS project then undertook the requirement that those SIDS should be replicated correctly on any harddrive or disk without the necessity of using an external program application. The ADF file structure is the summation of the resultant software generated by the CGNS project, having had some success to date taking hold throughout the CFD community due to these inherent advantages already discussed.

5.3.3 Domain splitting

As was mentioned before, that in order to prepare the problem under consideration for parallelization, it is necessary for the computational domain to be split into a corresponding number of sub-domains equal to the number of expected parallel processors.

The domain splitting performed in this dissertation is undertaken by using

a compiler based on the METIS library of domain decomposition algorithms, to split the complete domain into the required user specified number of sub-domains. The author would like to sincerely thank Dr. Jason Jones for provision of his METIS based domain decomposition software which is the sole mesh decomposition software used in this dissertation.

The METIS library or 'Serial Graph Partitioning and Fill-reducing Matrix Ordering' [54] library as it is known in full contains a set of serial programs having the capability for either the splitting of graphs, the splitting of matrices as well as the decomposition of finite element meshes. The METIS library functions in a topographical manner to propose the specified number of sub-domain boundaries that would intersect with the minimum number of element edges of the finite element mesh otherwise known as the 'minimum edge cut'. A METIS library based domain decomposer obtains the sub-domains quicker than a geometrical based decomposer would, however the topographical method may choose to create sub domains in locations in some unusual mesh refinement cases that a human operator would not choose if undertaking a manual decomposition. The advantage of creating sub-domains divided along the minimum edge cut principles is that the number of inter-processor communications containing nodal boundary information is kept to a minimum.

As the sub-domains are created, the nodes, elements and faces of the parent domain are distributed amongst the newly created sub-domains then flagged to which sub domain they are now allocated within. Once all this connective data have been allocated and flagged to a sub-domain, this data is then locally renumbered relative to the newly residing sub-domains. This renumbering will now allow a local looping over the elements and nodes in the sub domain to be possible, as these nodes or faces residing in the sub-domain thus create miniature meshes that will each be solved by the corresponding parallel processor, therefore each node and face will have both a local and global identity.

The nodes present on the boundaries between the sub-domains are duplicated by the domain decomposition process appearing in both sub-domains, meaning that each sub-domain stores a duplicate version of the nodes or triplicate version if along a boundary between three sub-domains. Those nodes which appear more than once are known as ghost nodes and are illustrated in Figure (5.5).

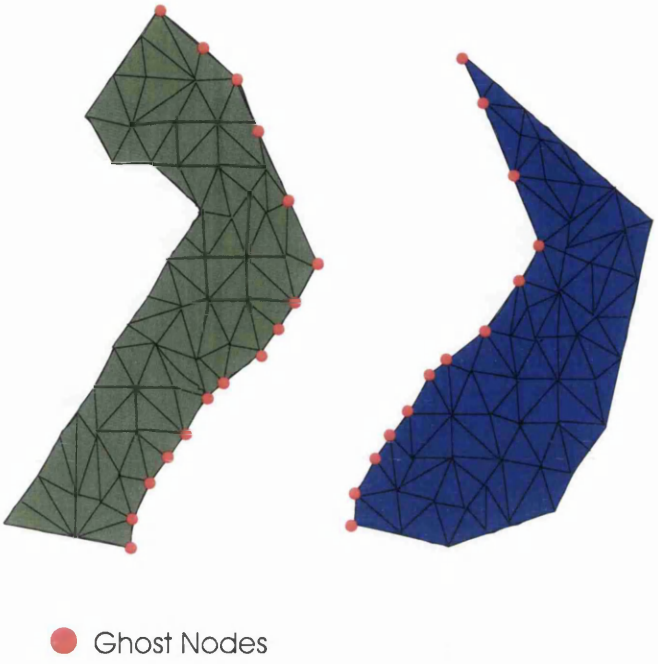


Figure 5.5: The boundary between two sub domains for a two dimensional mesh, note the nodes on the boundary denoted in red are duplicated while the boundary elements are only distributed to one sub domain

This duplication of ghost nodes along the boundaries results in the total number of local nodes when summated from every one of the sub domains exceeding the total number of nodes which are found in the original serial/global domain. The total numbers of exterior boundary faces and interior elements when summated from every one of the sub-domains are still equal to the corresponding number existent in the serial/Global domain as the newly created boundary faces between the different sub-domains are disgarded.

Note that additional interior boundary faces were generated by use of the METIS based domain decomposer along the borders between the sub-domain meshes. These additional faces were not necessary for the CBS finite element analysis and were subsequently removed by post processing of the parallel mesh.

Finally, the domain decomposing process organises the necessary inter processor communication data by identifying the local identities of the ghost nodes in each of the sub-domains and then storing this data in additional tree structured memory allocations using the previously described flexible ADF file structure format[55].

5.3.4 Parallel code

Parallel architecture consists of two kinds of processors, a controlling and calculation processor known as the master processor and a designated amount processors that calculate only, known as slave processors. The number of calculation processors is generally designated as a amount equal to two to the power of an integer, i.e. 2,4,8,16....N. This is a guideline rather than an actual hardware requirement.

The parallelised code is essentially built up of loops over the local elements and nodes present in both the slave processors and the master processor similar to the layout of loops in the serial version of the code. The main distinguishing attribute of the master processor is both to gather and organise the global variable values and secondly to check these variable values for convergence after each time step is completed. The master processor also distributes and gathers the I/O data in addition to sending the updated global values back to the slave processors for the next time step.

The sequence of events undertaken by the parallelised version of the code will now be described below, the unknowns mentioned in the program outlined refer to the values of density, velocity and energy.

1. Firstly all global input information is read by the master processor such as the flow parameters and the nodal global identifications. Any global preprocessing required by the code is calculated at this point by the master processor, these calculations include the determination of which nodes exist on a solid geometry as well as the positioning of the trip curve in the case of application of the one equation turbulence model.

2. This preprocessed global information is then broadcast to the slave processors, (see Figure (5.6)) for a topographical view of the organisation and distribution of this data. This broadcasted information includes the initial values for the prescribed unknowns along with the flow parameters necessary for the completion of the first time step. Once this data has been received by the corresponding slave processors, the calculation of local preprocessing is then undertaken by the slave processors. This local preprocessing includes the calculation of the shortest wall distances for each node in the sub-domain as well as the calculation of the elemental shape function derivatives for each

element contained in the sub-domain.

3. Once all slave processors have signaled that their corresponding local preprocessing stages have been completed, the signal for the time iteration loop to be commenced is then sent by the master processor.

4. Each slave processor calculates the contributions to the nodal values for the nodes contained in the considered sub-domain. These contributions are found from the calculation of the right hand side of the CBS explicit fractional step Step one equation.

5. The nodal contributions for the ghost nodes existing on the internal boundaries bordering other sub-domains are identified and sent to the corresponding sub domain that contains the duplicate ghost nodes.

6. The associated contribution values of the ghost nodes on the boundary calculated by the bordering sub-domain are then received by the corresponding processors that are allocated to these bordering sub domains.

7. Once all the sets of contributions are received from every linked sub-domain, the signal is then given to use these received values to update the values of the nodal unknowns calculated by the transmitting sub-domain. The solver is then advanced to the next stage of the CBS explicit fractional step by the master processor.

8. The parallelised code continues to calculate, send and receive nodal contributions until all the unknown values are updated for all four steps of the CBS scheme for the relevant time step been computed.

9. At this stage, it is now time for the turbulence model to be updated should a turbulence model be present. In the case where the one equation turbulence model is been solved, the contributions for the ghost nodes of the turbulence variable values are obtained by the slave processors in the same manner that was described above for the calculating the unknowns of the Navier Stokes equations. These turbulence variable values are then updated using the ghost node send and receive procedure as described above.

10. The calculation of convergence is measured in this dissertation using the equation for the calculation of the L_2 density norm below.

$$L_2 = \left[\frac{\sum_1^{nodes} \left(\frac{\rho^{n+1} - \rho^n}{\Delta t} \right)^2}{\sum_1^{nodes} \left(\rho^{n+1} \right)^2} \right]^{0.5} \tag{5.11}$$

A check for convergence requires that the locally calculated nodal values of density be sent from the slave processors to be assembled by the master processor. This assembly to obtain the global density values undertaken by the master processor will increase the computational cost and run time duration. Therefore in order to avoid a high computational cost in checking for convergence, a convergence check shall only be performed after a user specified number of computational iterations rather than after each and every single iteration.

Should the time step being solved by the parallelised code be a specified time step where a check for convergence does take place, the nodal variables for all the unknowns are collected by the master processor from every slave processor as is illustrated by Figure (5.6). Should the convergence tolerance of this check be met, the master processor then advances to the post processing stage.

11. The master processor otherwise advances the calculation to the next time step, should the convergence tolerance not be met or should no convergence check be made for this computational iteration. Steps four to ten of this schedule are then re-iterated to calculate the CBS scheme cycle again for the next time step iteration.

12. The post processing stage is reached, should the master processor complete the maximum number of time steps specified by the user or the convergence tolerance be reached. The values of both the nodal unknowns and the turbulence variables are collected by the master processor in the manner illustrated by Figure (5.6). These collated global values are then used for the postprocessing stage calculations, which include the calculation of the coefficient of pressure and skin friction distributions and then the outputting of these

coefficients to data files. The collected values of the global unknowns are also required for the creation of the output and restart files also written by the master processor, this concludes the postprocessing stage and end of the program code.

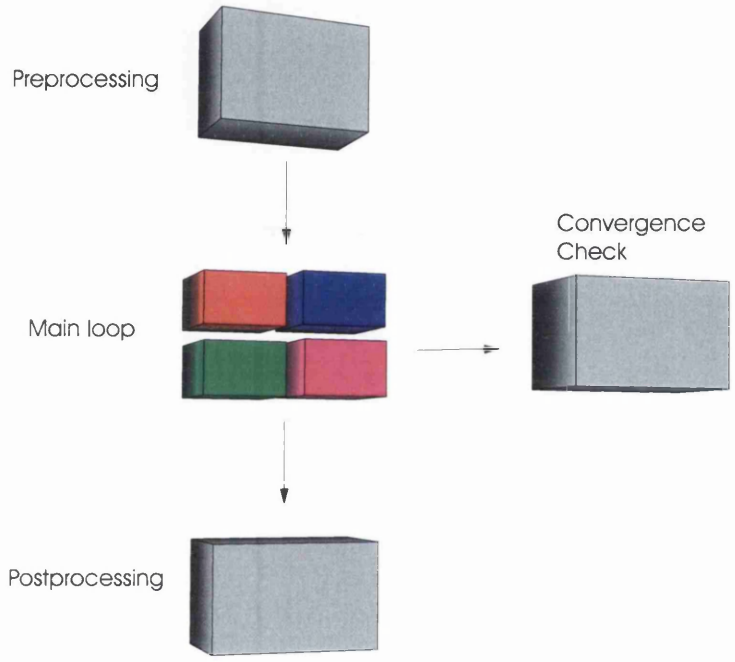


Figure 5.6: The dividing and sending of global information from the master processor into the local information for the slave processors

5.3.5 Speed up observations

The first problem that was considered using the parallelisation procedure described in the above subsections was of a viscous nature. This problem was represented by the courser of the two meshes discussed in this subsection, a domain which represented a NACA0012 airfoil modeled in three dimensions and comprising of approximately 500,000 mesh elements in size. A convergence check was specified to be performed by the parallelised code every ten time step iterations, for consistency the serial version of the code was adjusted so that it too checked for convergence every ten time step iterations in order not to skew the run time duration comparisons.

To validate and debug the parallelised version of the code, a comparison was made between the computed unknowns calculated by the parallelised code to those unknowns computed earlier by a serial version of the code which was run on the corresponding mesh before the domain decomposition procedure was performed. This comparison of values was undertaken by comparing the values of the unknowns by a post processing program that read the necessary data from the relevant restart files. The restart files were themselves created after one time step iteration, note that this is the same manner in which the edge/element based comparison found at the start of this chapter was performed.

Concluded from the comparison of these variable values was that validation between the parallel and serial codes was perfect apart from the slight expected truncation errors caused by machine code calculation discrepancies which were typically of the order magnitude of 10^{-7} when analysing a single precision versions of the parallel and serial code.

Run time durations were recorded both for the serial runs and for parallelised runs with an architecture of four, eight and twelve processors, these times are given in Table 5.2 and representing graphically in Figure (5.7). The machine used was the shared use Silicon Graphics or SGI parallel computing platform Onyx 3800 series, the Onyx 3,800 series has 32 R14000 type processing CPUs each of which run at a clock speed of 500MHz. Note that the values recorded in Table 5.2 and later in Table ??table1200000) are the mean values recorded from three separate run simulations all computed at a time when some of the R14000 processors remained idle to avoid the results being possibly influenced by the actions of other users.

A time and efficiency analysis is undertaken on these run times, the result of which is depicted in Table (5.3). The second column labeled 'total cost' in this table and later in Table (5.5) is the run time multiplied by the number of parallel processors initialised, obtaining the cost in processor seconds of the computational run. The 'Scaling' value in the penultimate column is the ratio of the cost in processor seconds to the corresponding time needed by the serial code to perform the same number of computational iterations. In the ultimate column the 'Efficiency' rating is portrayed, a value greater than 100% in this column is representative of the fact that super linear speed up has occurred.

Firstly, regarding the run times of the viscous case run modeled by the 500,000

Processors	Iterations	User time (s)	System time (s)	Real Time (s)
Serial	100	446.03	0.43	446.50
Serial	200	870.96	0.46	871.40
Serial	500	2173.92	0.77	2174.69
Serial	1000	4401.75	1.43	4403.18
Serial	5000	22130.88	5.56	22136.44
4	100	16.45	43.10	59.55
4	200	32.55	81.95	114.40
4	500	77.23	199.57	276.80
4	1000	164.37	419.91	584.30
4	5000	789.67	1938.91	2728.60
8	100	9.03	25.67	34.70
8	200	18.18	46.62	64.80
8	500	44.47	117.33	161.80
8	1000	90.41	234.79	325.20
8	5000	444.61	1121.89	1576.50
12	100	8.78	21.22	30.00
12	200	16.74	39.36	56.10
12	500	35.56	97.84	133.40
12	1000	74.46	190.33	264.80
12	5000	399.61	931.50	1331.10

Table 5.2: Run time durations in seconds against iterations for simulations with different numbers of processors considering a 500000 element mesh.

Processors	iterations	Total cost (s)	Scaling %	Efficiency %
Serial	100	446.50	100	n/a
Serial	200	871.40	100	n/a
Serial	500	2174.69	100	n/a
Serial	1000	4403.18	100	n/a
Serial	5000	22136.44	100	n/a
4	100	238.20	53	187
4	200	457.60	53	190
4	500	1107.20	51	196
4	1000	2337.20	53	188
4	5000	10914.40	49	203
8	100	277.60	62	161
8	200	518.40	59	168
8	500	1294.40	59	168
8	1000	2601.60	59	169
8	5000	12612.00	57	176
12	100	360.00	81	124
12	200	673.2	77	129
12	500	1600.80	74	136
12	1000	3177.60	72	139
12	5000	15973.20	72	139

Table 5.3: Time and efficiency analysis against iterations for simulations with different numbers of processors considering a 500000 element mesh. An efficiency value greater than 100% indicates that super linear speed up has occurred.

element mesh, it is seen from Table 5.2 that super linear speed up occurred for every one of the parallelised cases regardless of the number of processors commissioned nor the number of iterations for which the parallelised code was solved. The run duration times portray that the computational efficiency increased directly with the number of time iterations, thus indicating the trait of an increased time cost exhibited by parallel codes in the broadcasting of variable values between the slave processors at both the input and preprocessing stages.

The results also indicate that an aspect of diminishing returns does exist when a problem is parallelised. When splitting a problem into smaller and smaller sub-problems, a point will eventually be reached when the running time saved by splitting the main problem any further, diminishes to the extent that the extra inter-processor communications across an ever increasing amount of sub-domain boundaries are instead more computationally costly. This diminishing return will eventually increase to the extent that use of more processors will actually be also more expensive in real time. The actual point of diminishing return or resistance to speed is seen to be approached from the graph illustrated in Figure (5.7), as there is very little additional saving in real time between the cases where eight slave processors are being used and the last case where twelve slave processors are used.

These diminishing returns underline the reason that parallelisation is not justified on very small meshes, as logically the quickest run time duration will probably be that of the serial code if a small enough domain is used. The author expects that even in the case of this 500,000 element sized mesh, the real time run duration would soon increase in direct proportion to the amount of processors used by the parallel code as the number of commissioned processors rise above the twelve processors used here.

The effect of diminishing returns dramatically increasing the real time duration due to an increase of inter-processor communications from additional processors been used, illustrates again the philosophy in which parallel code should be written. It is vitallty important that a parallel computing programmer should write code in a manner that minimises the amount of inter-processor communications as the number of these communications will rise exponentially as the number of processors being commissioned are increased.

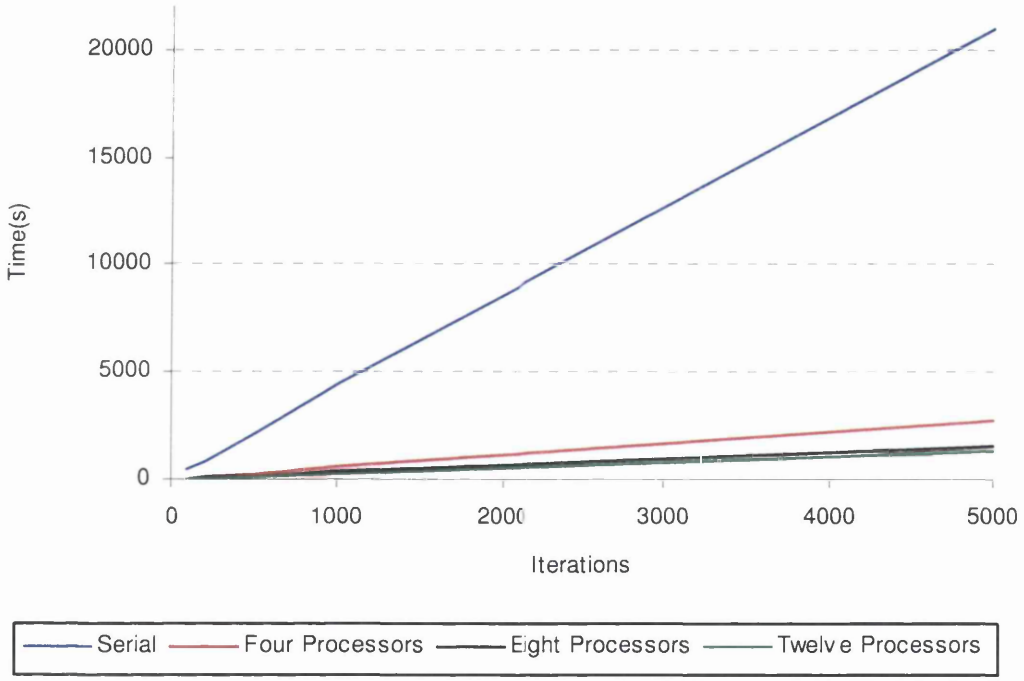


Figure 5.7: Graph showing total times in seconds against iterations for different numbers of processors on a 500000 element mesh

Processors	Iterations	User time (s)	System time (s)	Real Time (s)
Serial	100	535.80	0.96	536.76
Serial	200	1052.55	0.94	1053.50
Serial	500	2612.90	1.70	2614.60
Serial	1000	5260.94	2.57	5263.50
Serial	5000	26152.21	8.00	26160.21
4	100	38.09	94.40	132.50
4	200	71.49	180.46	251.95
4	500	177.83	455.22	633.05
4	1000	347.29	892.74	1240.03
4	5000	1758.29	4348.83	6107.12
8	100	20.56	49.05	72.61
8	200	36.90	93.81	133.71
8	500	98.21	242.61	343.83
8	1000	194.14	501.73	698.86
8	5000	932.73	2410.51	3346.24
12	100	14.61	36.69	51.30
12	200	27.76	69.80	97.57
12	500	65.98	168.59	234.58
12	1000	135.42	331.17	466.59
12	5000	646.06	1645.85	2291.91

Table 5.4: Run time durations in seconds against iterations for simulations with different numbers of processors considering a 1200000 element mesh

The second problem considered for parallelisation was an Euler flow problem around a NACA0012 airfoil represented by a three dimensional domain consisting of approximately 1,200,000 tetrahedral elements. The observed run time durations were again recorded for both the serial code and for parallel run cases consisting of four, eight and twelve processors, these run times are recorded in Table (5.4) and also shown graphically in Figure (5.8). In obtaining the run times for the Euler flow case, a convergence check was performed only after every 250 time iterations, this condition was again applied to both the serial and parallel run cases for consistency. Lastly the time and efficiency analysis of these results are shown in Table (5.5).

It is evident from considering the results in Table (5.5) that super linear speed up described previously has again occurred for the parallel case utilizing four processors. It is also noted that the amount of super linear speed up observed is not as great for the 1,200,000 element sized mesh as it was for the previous 500,000 sized element mesh. Contrary to the viscous case, the Euler cases run times utilizing both eight and twelve processors did not achieve any level of super linear speed up.

The run times in Table (5.4) and Table (5.5) do however exhibit the same trends that are present in Tables (5.2) and (5.3) for the 500,000 element sized mesh. These trends show that the level of efficiency or amount of super linear speed up is again seen to increase as the number of iterations increase. The trends also exhibit that a diminishing return is present, as a resistance to speed up is again visible once performance is seen to drop after the number of processors is steadily increased.

The lower observed performances of the 1,200,000 mesh Euler flow case when compare to the 500,000 mesh viscous flow case, may be attributed to two separate causes. The first cause is that since the actual size of the sub-domains in the 1,200,000 mesh are larger than the corresponding sub-domains of the 500,000 elemental mesh, a smaller percentage of the overall problem is stored in the caches of the processors on the parallel platform.

Secondly the Euler flow nature of the 1,200,000 element mesh case will affect the amount of speed up, as a smaller percentage of the computational cost is expended by the simplified inviscid solution loops,the parallelised efficiency is stronger in the solution stage relative to the preprocessing stage which is relatively quite costly in inter-processor communications.

Processors	iterations	Total cost (s)	Scaling %	Efficiency %
Serial	100	536.76	100	n/a
Serial	200	1053.50	100	n/a
Serial	500	2614.60	100	n/a
Serial	1000	5263.50	100	n/a
Serial	5000	26160.21	100	n/a
4	100	530.00	99	101
4	200	1007.80	96	104
4	500	2532.20	97	103
4	1000	4960.12	94	106
4	5000	24428.48	93	107
8	100	580.88	108	93
8	200	1069.68	102	97
8	500	2750.84	105	95
8	1000	5590.88	106	95
8	5000	26769.92	102	98
12	100	615.60	115	87
12	200	1170.84	111	90
12	500	2814.96	108	93
12	1000	5599.08	106	94
12	5000	27502.92	105	95

Table 5.5: Time and efficiency analysis against iterations for simulations with different numbers of processors considering a 1200000 element mesh. An efficiency value greater than 100% indicates that super linear speed up has occurred.

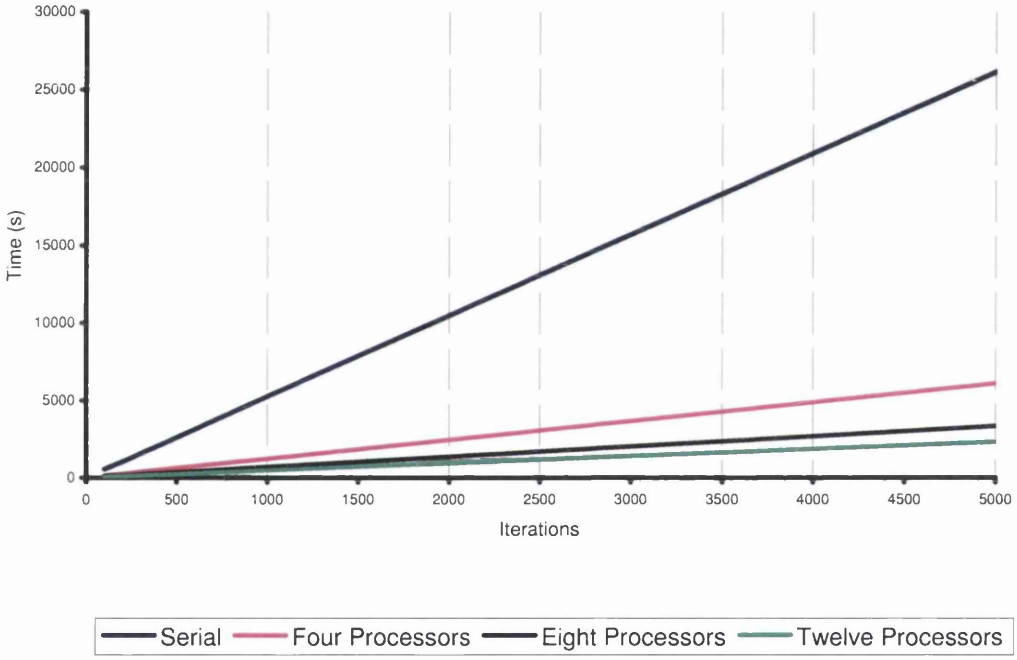


Figure 5.8: Graph showing total times in seconds against iterations for different numbers of processors on a 1200000 element mesh

5.4 Chapter Summary

Two computational techniques, the concept of parallel processing and the application of the edge based formulation, have both been considered as possible solution speed up techniques in this chapter. These two techniques were both successfully incorporated into the CBS explicit fractional step code used in this dissertation, illustrating the robustness and flexibility of the CBS scheme in being able to adopt both of these speed up procedures. The explicit fractional step nature of the CBS scheme allowed both of these two techniques to be applied in a straightforward manner.

The amount of speed up observed from the application of the first technique of edge based formulation was less than the author had originally expected for the flow cases studied in this chapter. This low level of speed up can be attributed to the high initial preprocessing cost from calculating elemental quantities such as the derivatives of the elemental shape functions. It is the author's opinion that the edge based formulation would achieve a better performance in regards to other flow problems such as those problems requiring a larger amount of iterative steps.

As a low amount of speed up was observed by the Edge based formulation in dealing with Euler flow, it was decided that extending the edge based formulation to viscous flows would not be cost effective for the reasons stated in the edge based formulation section of this chapter.

The amount of speed up observed by the application of the second computational technique, that of parallel processing, was very substantial when a comparison was made with the run times of the corresponding serial code. It was also noted that the number of processors used by the operator when running a parallel code should be carefully considered to maximise the level of speed up efficiency. Also observed was the importance that a parallel code should keep the amount of inter-processor communications to a minimum as the run time duration and computational cost for these inter-processor communications will increase exponentially as the number of processors used being commissioned increase.

Chapter 6

Inviscid flow Problems

6.1 Introduction

In this chapter, the commencement of analysing the results generated by the Characteristic Based Split will begin. A decision was made to begin these considerations and validations using results generated from Euler flow based simulations, recognised as one of the most basic and general flow configurations[56].

The simulated flow for a non-heat conducting non-viscous fluid described by the Euler equations, is obtained by neglecting all the shear stresses and heat conduction terms from the Navier Stokes equations discussed in detail in the problem theory chapter. This approximation is justified from Prandtl's boundary layer analysis which found that the Euler equations are valid for flows at high Reynold's numbers outside of any viscous regions that have developed in the vicinity of solid surfaces interacting with the flow[57].

Three important aspects to be considered when modeling a solution are stated as the following.

1. The choice taken and the implementation of the chosen boundary conditions present on the surface of the solid geometry and present at the farfield boundaries of the domain under consideration.

2. The consideration of the areas located at the leading and trailing edges of the airfoil has long being areas of interest for solvers. Some schemes[58, 59] give this area and the pursuing wake region additional treatment resulting in

an increase of computational cost. Universally, the required degree of mesh refinement at the leading and trailing edges are of importance to all numerical schemes. Additionally, the degree of accuracy achieved by a numerical scheme modeling the highly stretched elements that can appear at the trailing edge of an airfoil must also be considered .

3. Lastly, the procedure for modeling the accurate computation of any shock waves that may be present is relevant to obtaining a valid solution due to the nature of a shock being a non-linear event.

The farfield is set at a minimum distance of 10 chord lengths from the leading edge of the airfoil. It is assumed that this distance allows the near wall region to be void of any disturbance or reflection that could otherwise have been caused by interaction of the flow at the farfield.

The boundary condition at the inlet region is imposed by setting each nodal value present on the boundary to a prescribed free stream value at the commencement of each computational time step. Contrastingly for the outlet region should the flow be sub-sonic, the density is set to the free steam value on the boundary else no other action is taken due to the distance of the farfield to the solid geometry being deemed sufficient to avoid any reflection or interaction affecting the accuracy of the simulation.

Regarding the aspect of shock prediction as mentioned in this section, the 'lumped and consistent mass matrix' shock capturing technique is explained in detail in chapter three is the method used for the modeling of shocks.

In the proceeding sections of this chapter, various Euler flow problems will be presented, these inviscid solutions are both two dimensional and three dimensional in nature. The principle of this chapter is to show the robustness of the CBS scheme as it is applied to different flow cases of further increasing difficulty and complexity. There are four Euler flow problem modeled in this dissertation, these are that of flow around a NACA0012 airfoil, flow past a RAE2822 airfoil, flow over a ONERA M6 wing and finally the modeling of flow past the partial geometry of a Dassault Falcon business jet.[60, 61, 62, 63].

6.2 Flow past NACA0012 Airfoil

The first problem to be modeled is that of the simple Euler flow case past a NACA0012 airfoil which is aligned to the direction of flow, thus incorporating an angle of attack of zero.

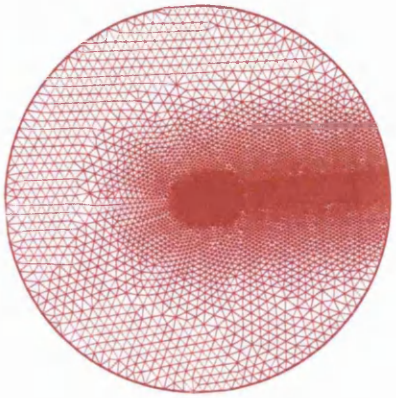
As was mentioned in the previous section, it is assumed that a distance of ten chord lengths from the farfield to the solid geometry is sufficient to avoid any reflection or interaction affecting the accuracy of the simulation. To validate this assumption which is posited throughout this dissertation, a comparison shall be made of the NACA0012 results obtained from simulations using varying distances between the farfield and solid geometry. These distances are measured in multiples of the chord length and are approximately ten, fifteen and twenty chord lengths in magnitude.

The meshes required to represent the flow domain surrounding the NACA0012 airfoil are created using a two dimensional advancing front based mesh generator. The resultant meshes produced by the advancing front generator[64] consist of a fully unstructured nature, meaning that there is no structured layer in the region of the solid surface nor has any manual refinement undertaken in the trailing and leading edge regions. The leading and trailing edge regions are illustrated in Figure (6.1), alongside additional views of the considered meshes. The level of mesh refinement in the region of the airfoil is the same for all meshes as only the position of the farfield was changed when writing the mesh generator data files.

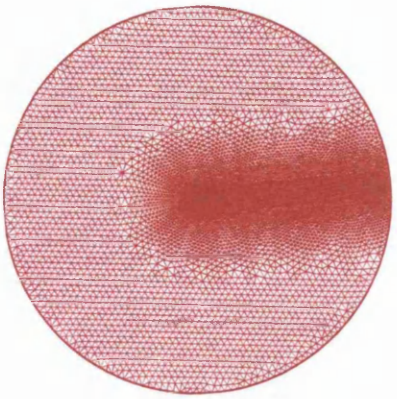
To validate the assumption that a minimum distance of ten chord lengths to the farfield will not affect the accuracy of the results, four different test case were employed covering a broad range of Mach numbers. An artificial diffusion constant of 0.7 was used in obtaining the following results alongside a residual smoothing coefficient of 0.01.

The test cases modeled included subsonic flow at a Mach number of 0.5, transonic flow at Mach numbers of 0.95 and 1.2 and lastly, supersonic flow was considered with a Mach number of 2.0. The computed pressure contours from these four test cases are shown in Figure (6.2). Observing Figure (6.2), the development of the bow shock and the trailing edge fishtail shock development are clearly evident as the flow speed increases from subsonic to supersonic.

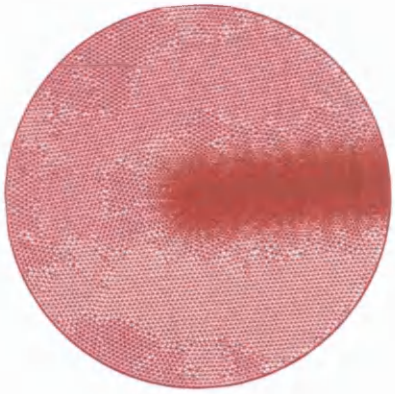
In order to check the quantitative accuracy of the results computed by the



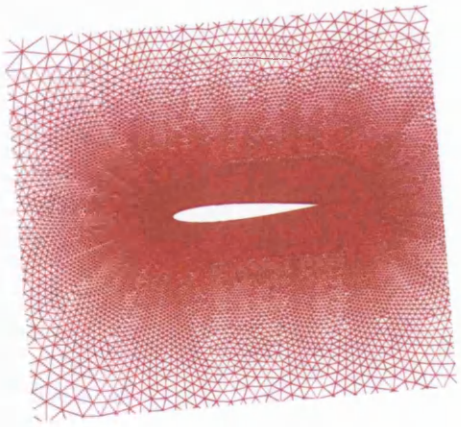
(a) A farfield distance of 10 chord lengths



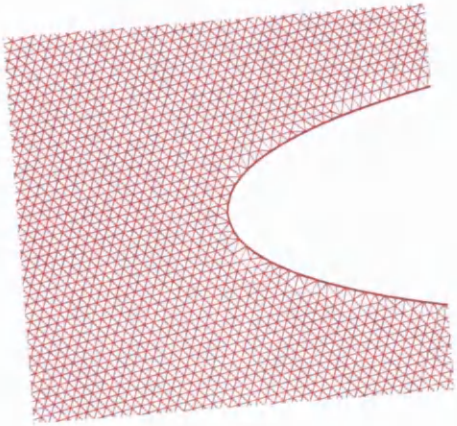
(b) A farfield distance of 15 chord lengths



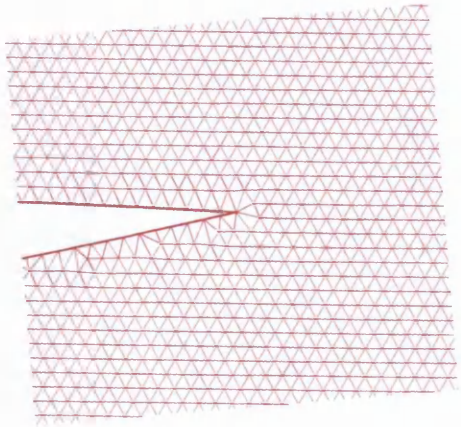
(c) A farfield distance of 20 chord lengths



(d) The NACA0012 airfoil

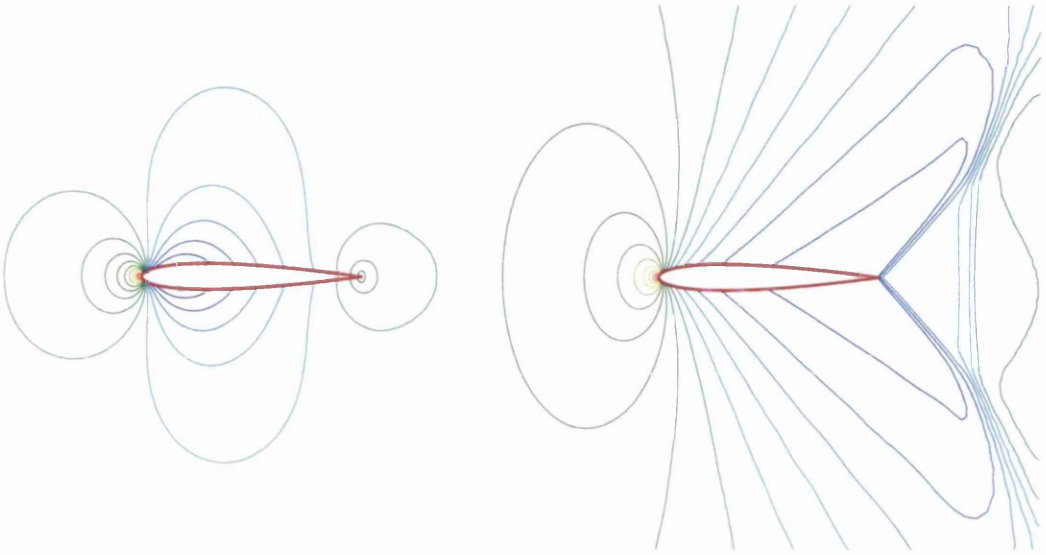


(e) A depiction of the leading edge



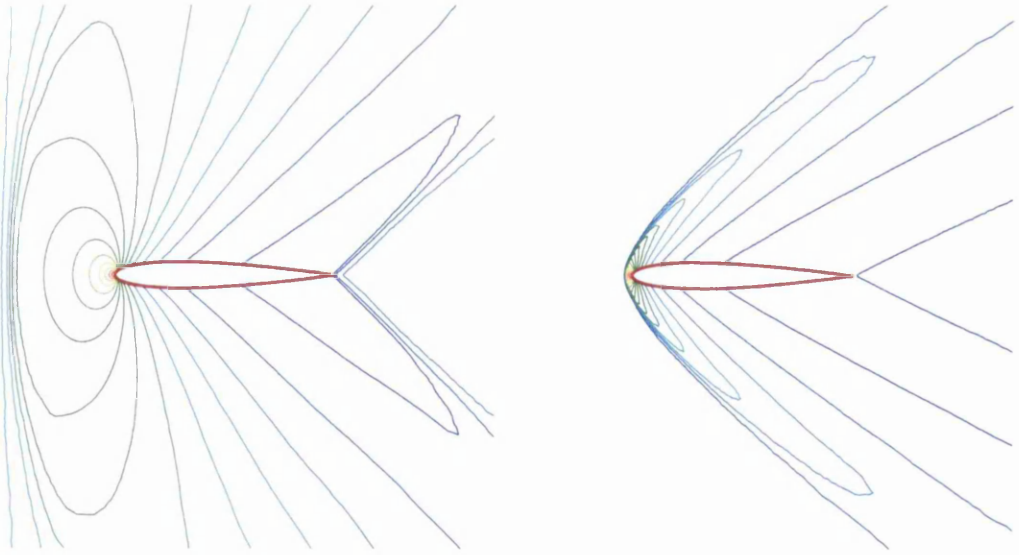
(e) A depiction of the trailing edge

Figure 6.1: An overview of the Meshes depicting the NACA 0012 aerofoil.



(a) Mach number of 0.50

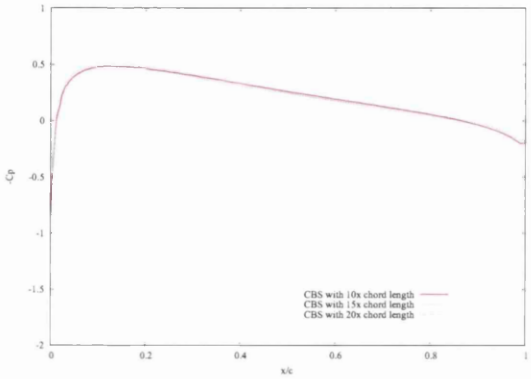
(b) Mach number of 0.95



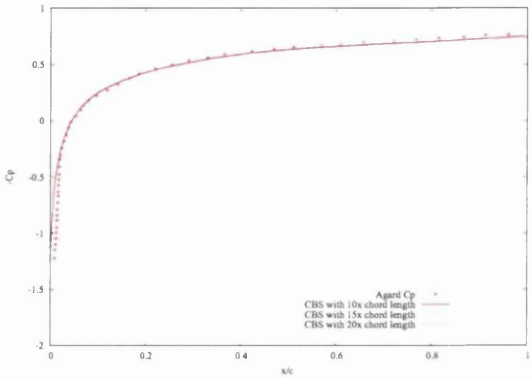
(c) Mach number of 1.20

(d) Mach number of 2.00

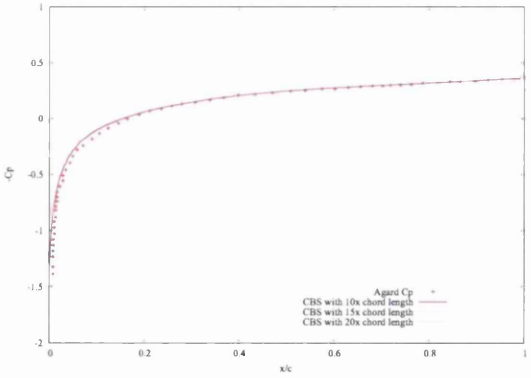
Figure 6.2: Pressure contours for the NACA0012 aerofoil. An area in blue denotes a low value while an area in red denotes a high value.



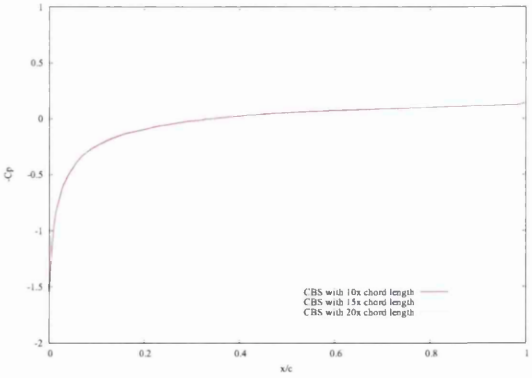
(a) Mach number of 0.5



(b) Mach number of 0.95



(c) Mach number of 1.20



(d) Mach number of 2.00

Figure 6.3: Pressure coefficients distributions (CP) for the NACA0012 aerofoil

CBS scheme, the coefficient of pressure (C_p) was calculated along the surface of the airfoil and then compared against results previously published by other researchers. The coefficient of pressure is obtained from the following equation.

$$C_{p\,i} = \frac{2\,(p_i - p_\infty)}{\rho_\infty u_\infty^2} \tag{6.1}$$

The subscript i denotes the node on the airfoil under consideration, p_i is the value of pressure at this node and p_∞ is the reference pressure value. The reference pressure value is taken as free stream pressure value.

The calculated pressure coefficients for each of the test cases is shown in Figure (6.3) for the different domain sizes. A comparison is also made in Figure (6.3) between the pressure coefficients computed by the CBS scheme and the corresponding results published by the AGARD project [65] for the test cases with flows of Mach numbers of 0.95 and 1.2.

The results computed by the CBS scheme are seen in Figure (6.3) to be consistent as the distance to the farfield is varied. This consistency for the considered flow problems, indicates that a distance of ten multiples of chord length to the farfield is sufficient to be ultilised in this dissertation. It is also evident from Figure (6.3) that the pressure coefficients computed by the CBS scheme when flows of Mach numbers 0.95 and 1.2 are considered, closely match the published pressure coefficients from the AGARD results. This close match of pressure coefficients thus validates the CBS code for this test problem and shows the strength of the CBS scheme in replicating the AGARD results that were obtained with use of a structured mesh.

It should also be noted that the pressure coefficients calculated by the CBS scheme were obtained from meshes that used approximately 254 wall points while the corresponding mesh used by the AGARD project incorporated 320 wall points, note only a trace of these 320 wall points is used to plot the representations of the AGARD pressure coefficients in Figure (6.3).

6.3 Flow past a RAE2822 airfoil

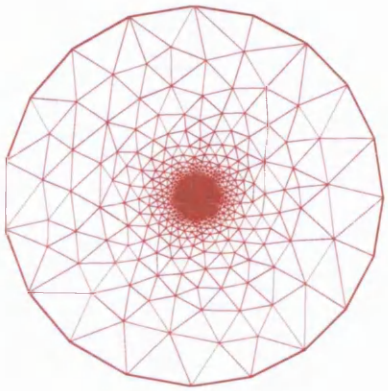
In order to verify the suitability of the CBS scheme to model flows that incorporate angle of attacks, flow past an inclined RAE2822 airfoil is considered here.

The domain surrounding the two dimensional RAE2822 airfoil is created by use of the same advancing front mesh generator that was utilised in the preceding example. Again an unstructured mesh is generated lacking any structured layer around the solid geometry nor any manual refinement treatment in the regions near the airfoil. The mesh representing the RAE2822 is shown in Figure (6.4) illustrating the unstructured nature. The number of nodes generated along the surface of the airfoil consisted of 889 wall nodes.

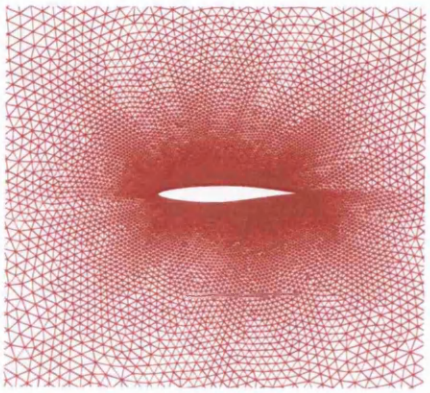
The flow parameters for the test case chosen to be simulated on the generated mesh are the following, the flow passing the airfoil is inclined at an angle of attack of three degrees to the horizontal direction and has a Mach number of 0.75. In additional, an artificial diffusion constant of 0.7 was used in obtaining a solution, a residual smoothing coefficient of 0.01 was also applied. The plotted pressure, density and velocity component contours are illustrated in Figure (6.5).

An observation of the plotted contours from Figure (6.5) shows the expected phenomena, an area of higher pressure located at the center of the airfoil's lower surface visible where the flow impinges on the airfoil's surface. A lower pressure region is observed located on the upper surface when compared to the higher pressure region located along the bottom surface of the airfoil illustrating the sheltered region created by the airfoil. The most distinctive and expected flow characteristic evident from Figure (6.5) is the presence of the shock located along the upper surface of the airfoil at approximately three quarters the length of the chord.

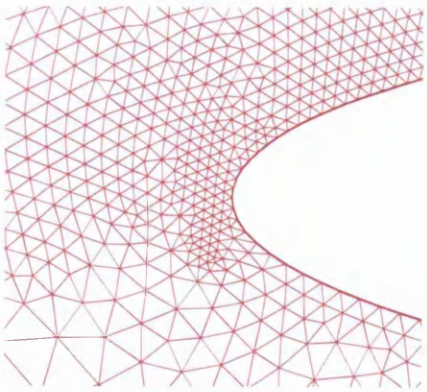
In order to analyse the quantitative accuracy of the results obtained by the CBS scheme, the pressure coefficient distribution is compared against corresponding results previously published by the AGARD project [65]. The AGARD project test case incorporated 320 wall nodes as before due to use of the same 'C' meshing topography mesh generator that created the NACA0012 mesh referred to the previous section. The structured mesh that was generated for use by the AGARD project extended the structuring process along the wake region for a combined total of 561 wall and wake points, although only a trace of these points is used for



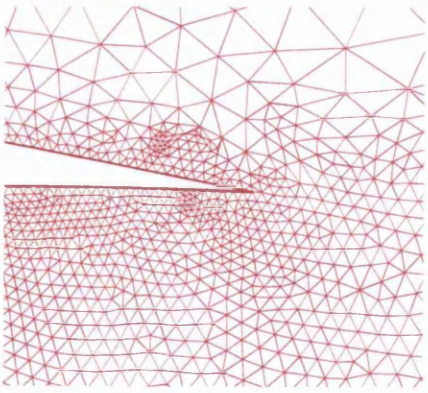
(a) The complete domain



(b) The RAE2822 airfoil

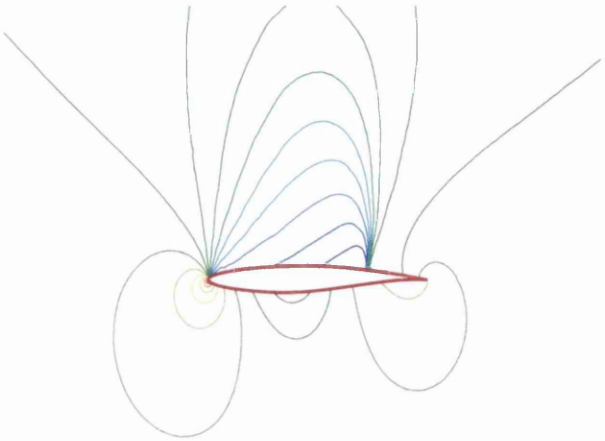


(c) The leading edge

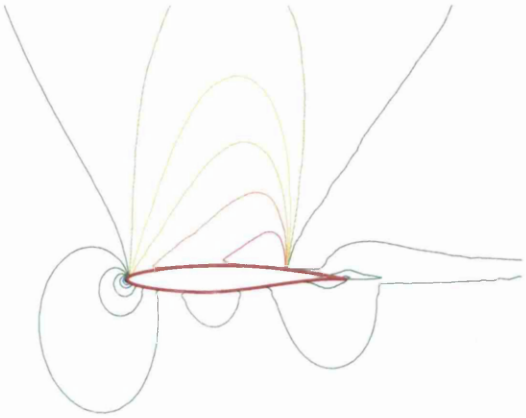


(d) The trailing edge

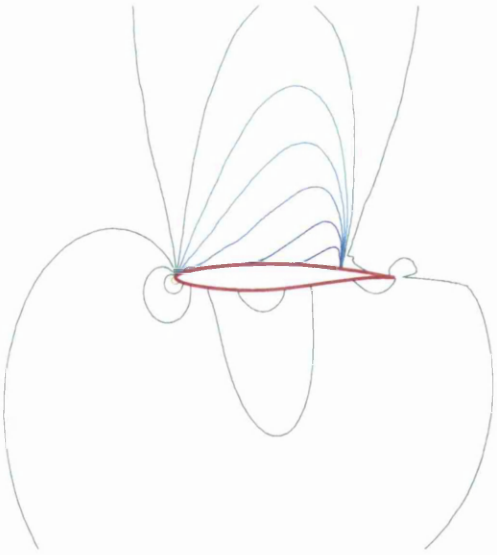
Figure 6.4: The Mesh representing the RAE2822 aerofoil that was used by the CBS scheme



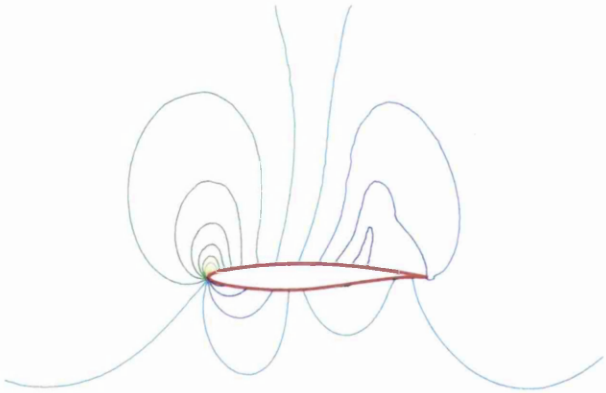
(a) The pressure contours



(b) The horizontal velocity contours



(c) The density contours



(d) The vertical velocity contours

Figure 6.5: Variable contours for the RAE2822 aerofoil at Mach 0.75 and angle of attack of three degrees. An area in blue denotes a low contour value while an area in red denotes a high value.

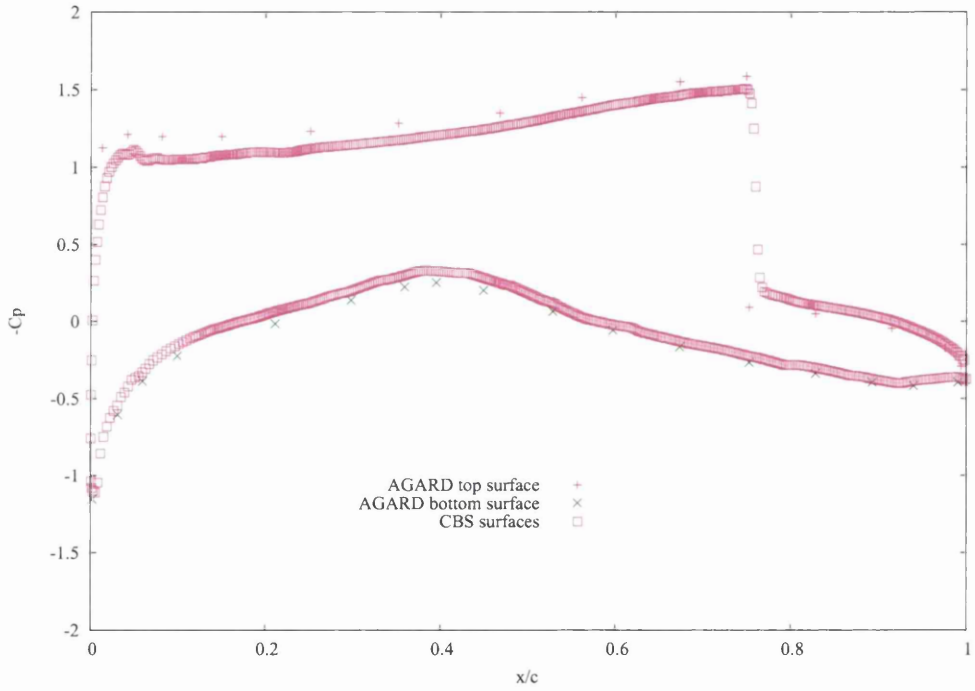


Figure 6.6: Pressure co-efficients for the RAE2822 aerofoil, plotted against a trace of the AGARD values

plotting the AGARD pressure distribution in this dissertation.

This aforementioned comparison between the CBS and AGARD distributions of pressure coefficient is shown in Figure (6.6). It is evident from observing Figure (6.6) that the CBS based code slightly underpredicts the magnitude of the pressure coefficient in the region of the leading edge when compared to the AGARD results. The reason for this underprediction by the CBS scheme can be attributed to the difficulty in providing the exact amount of artificial damping on the unstructured mesh.

The presence of the shock on the upper surface of the airfoil is predicted by the CBS in the same position as the shock predicted by the AGARD project. Although the shock predicted by the CBS scheme is not quite as sharp as that of the published AGARD result, the shock is still confined over a few mesh elements.

The discrepancies between the pressure coefficient distributions gradually disappear as the shock region is left behind and the trailing edge of the airfoil is approached.

6.4 Inviscid flow over an ONERA wing

The application of the CBS scheme will be extended in this section to an inviscid flow past a three dimensional non-planar geometry. The three dimensional geometry considered for this test case is that of a ONERA M6 wing, the mesh generated for the ONERA wing domain is again a fully unstructured mesh without any structured layers present in the region of the solid surface.

The ONERA mesh was generated using the 'Parallel Simulation User Environment' or (PSUE) mesh generator[66] from geometry files provided by Dr. J. Jones. The unstructured mesh generated contains a total of 1,469,666 tetrahedral elements, 25,2192 nodes and 33,768 triangular boundary faces, the surface mesh of this domain is illustrated in Figure (6.7).

The methodology used to model the tip of the ONERA wing is illustrated in Figure (6.8). An additional geometrical curve is prescribed at the centerline location where the upper and lower surfaces physically meet each other. Two additional surfaces are then created, thus connecting the aforementioned centerline curve with corresponding curves existing on both the upper and lower numerical

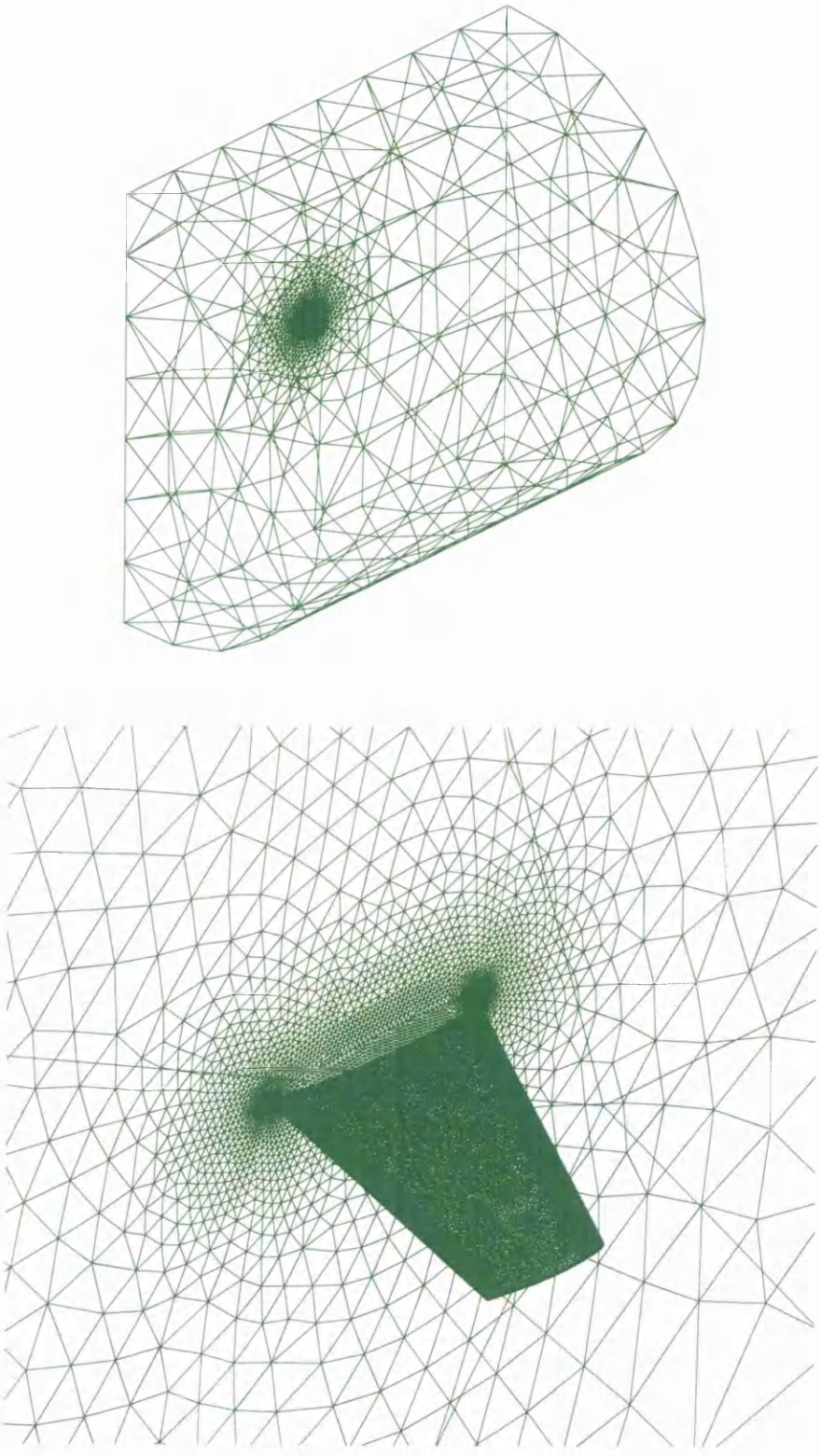


Figure 6.7: Surface mesh for the Onera aerofoil

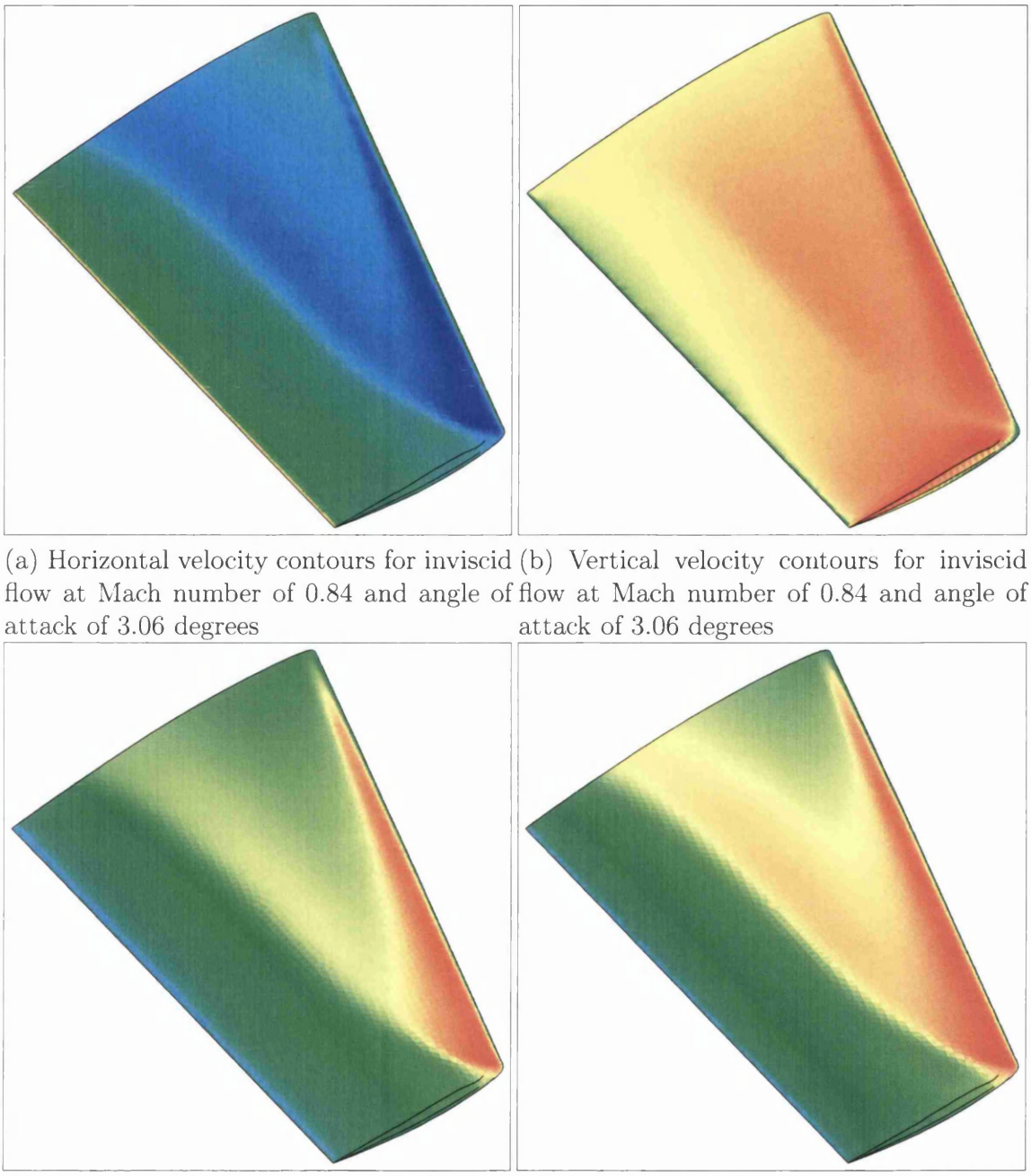
Figure 6.8: The treatment of the Onera wing tip

wing surfaces.

A transonic flow case is chosen for simulation of flow past the mesh of this ONERA M6 wing. The simulation of the transonic flow applied the following flow parameters: a Mach number 0.84 was prescribed to the flow past a wing which was inclined at an angle of attack of 3.06 degrees to the direction of flow approaching the wing's leading edge.

A solution of the above flow case was obtained using an artificial diffusion constant of 0.7 and a residual smoothing coefficient of 0.01. The contour plots for the test case values obtained by the CBS scheme for pressure, density and the velocity components are all shown in Figure (6.9). The expected phenomena can all be clearly seen from Figure (6.9), including the distinctive lambda (λ) shaped shock presence along the surface of the wing which is a well known aspect associated with ONERA problems.

To compare the quantitative accuracy of the results obtained by the CBS scheme, the computed values for the distribution of pressure are compared to previously published benchmark results[67] as before. As the published benchmark results for this test case are from experiments, they are viscous in nature. A comparison of the results is also made for this test case with inviscid results obtained by Sorenson[68]. The comparison between the pressure coefficients from



(a) Horizontal velocity contours for inviscid flow at Mach number of 0.84 and angle of attack of 3.06 degrees

(b) Vertical velocity contours for inviscid flow at Mach number of 0.84 and angle of attack of 3.06 degrees

(c) Density contours for inviscid flow at Mach number of 0.84 and angle of attack of 3.06 degrees

(d) Pressure contours for inviscid flow at Mach number of 0.84 and angle of attack of 3.06 degrees

Figure 6.9: Surface contours of computed values the Onera aerofoil

these three sources are taken at span-wise percentage lengths of 20%, 44%, 65%, 80%, 90% and 95% along the body of the wing, shown graphically in Figure (6.10).

It is evident from Figure (6.10) that, while similarities between the experimental results and those computed by the inviscid schemes are present, the match itself is not perfect with discrepancies along the first half of the airfoil's upper surface and in the trailing edge region. The discrepancy observed at the trailing edge is expected due to the lack of viscosity in the governing equation of the inviscid schemes while the underprediction of pressure at the leading edge may be attributed to an overdiffusive effect caused in degree by the artificial damping.

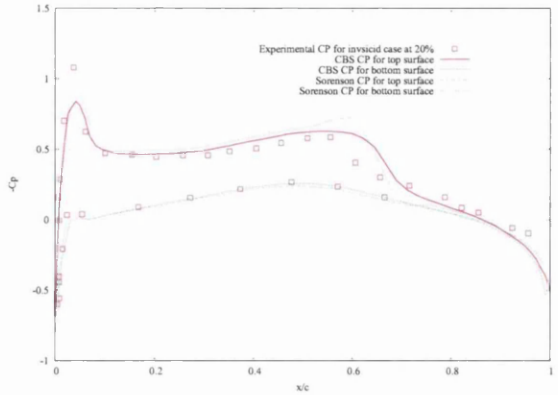
In respect to the respective performance of the inviscid schemes in relation to the upper surface of the wing, the CBS scheme obtains a closer match to the experimental results at lower spanwise percentage lengths in the regions where the ONERA wing would have been attached to the aircraft. Contrastingly, the inviscid code of Sorenson achieves a closer match to the experimental results at higher spanwise percentage lengths in the regions of the ONERA wing tip. In relation to the lower surface of the ONERA wing, both inviscid schemes perform to a similar standard in achieving a close match to the experimental results apart from the aforementioned trailing edge.

6.5 External inviscid flow over a Falcon aircraft

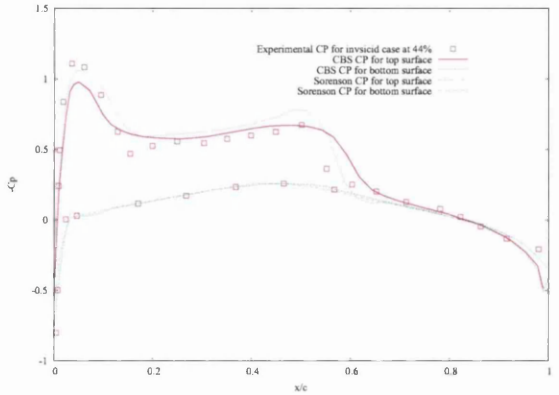
The final test case to be considered in this chapter is a more complex three dimensional geometry than the previous example of the ONERA wing. The reason is to examine the performance of the CBS scheme when applied to solving flow cases past the more demanding realistic geometries that an industrial based solver would expect to encounter. This examination will provide an estimate of the overall capability of the CBS scheme for use in any future possible applications.

The three dimensional geometry chosen for this test case is a representation of the Dassault Falcon business jet. However, only one half of the fuselage is considered here due to the symmetrical nature of this flow case.

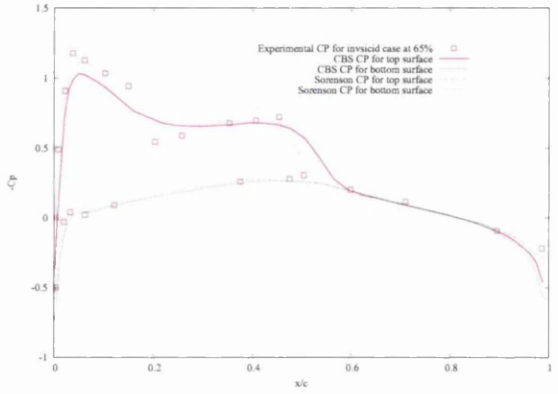
The mesh generated to represent the relevant domain was again generated by the PSUE application that was discussed in the previous section and provided by Prof. O. Hassan. This generated mesh comprises a total of 1,162,256 tetrahedral



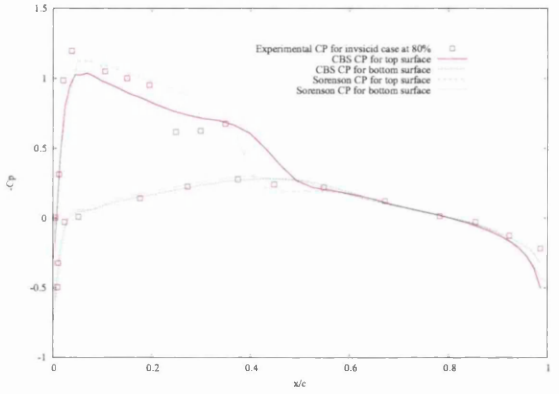
(a) Pressure distribution (Cp) at 0.2 of wing length



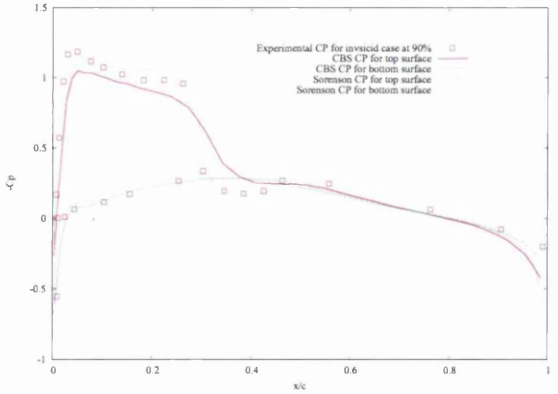
(b) Pressure distribution (Cp) at 0.44 of wing length



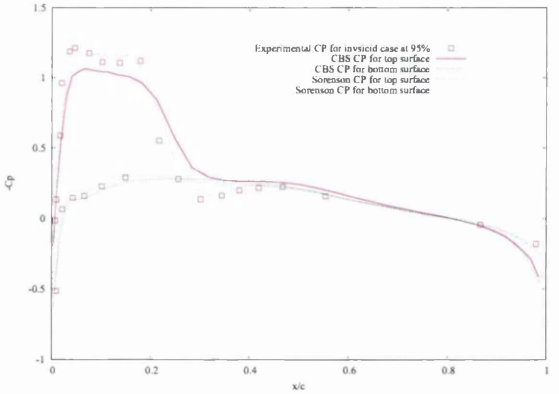
(c) Pressure distribution (Cp) at 0.65 of wing length



(d) Pressure distribution (Cp) at 0.8 of wing length



(e) Pressure distribution (Cp) at 0.9 of wing length



(f) Pressure distribution (Cp) at 0.95 of wing length

Figure 6.10: Coefficient of pressure distributions (Cp) for the Onera aerofoil at spanwise percentage lengths of 20% 44% 65% 80% 90% and 95%

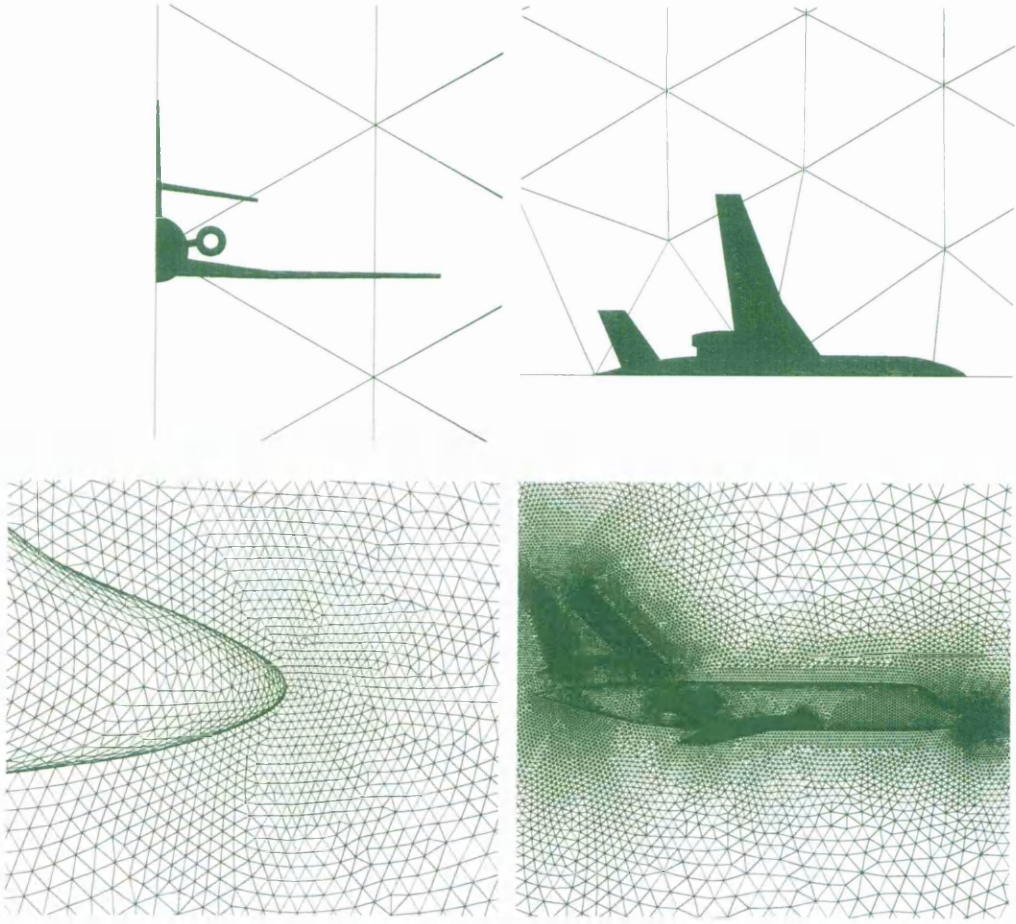


Figure 6.11: The three dimensional surface mesh that depicts the Dassault falcon

elements, 205,039 coordinate points and 52,964 triangular boundary edges, the surface mesh from this domain is shown graphically in Figure (6.11).

The following transonic flow case is modeled in this section. An angle of attack of 3.06 degrees is imposed on the direction of flow past the solid geometry, the Mach number of the flow is set as 0.84. An artificial diffusion constant of 1.2 was utilised in obtaining a solution. The resultant colour contour plots of the solution variables computed by the CBS scheme are shown in Figures (6.12) - (6.14).

The contours portrayed in Figures (6.12) - (6.14) are those of the density, horizontal velocity and pressure contours. An area of blue represents a high variable value contour region while an area in red denotes a low variable value region. It is evident from these figures that the expected flow behaviour has been modeled, this flow behaviour shall now be discussed in more detail.

Let the flow behaviour at the nose of the aircraft first be considered. A velocity dip is observed at this point where the fluid first interacts with the aircraft geometry. In this region, an increase in density and pressure is observed as fluid is slower in leaving this obstructed region before being directed around the body of the aircraft. It can be seen that the velocity values soon increase after this initial area of contact is left behind, while the values of density and pressure are similarly observed to decrease as the flow passes around the body of the aircraft and approaches the roof and side walls of the fuselage.

Let the observation of the flow behaviour at the wing now be considered. Examining the frontal section of the wing's upper surface, the velocity is seen to accelerate as it travels over the wings longer profile. The values of pressure and density thus decease due to this speeding up of the velocity, providing the necessary difference in pressure to exist between the upper and lower surfaces of the wing generating the required lift force to the allow the aircraft to remain airborne. Figure (6.13) also portrays a decrease in the horizontal velocity as the flow approaches the trailing edge of the wing's upper surface, this velocity decrease illustrates the velocity slowing as the flow leaves the wing to eventually become a value of velocity closer to the free stream velocity after the flow has fully separated from the geometry's surface.

Similar flow behaviour is observed to the discussed behaviour observed at the wing region is also seen to occur at the rear of the aircraft where the tail section

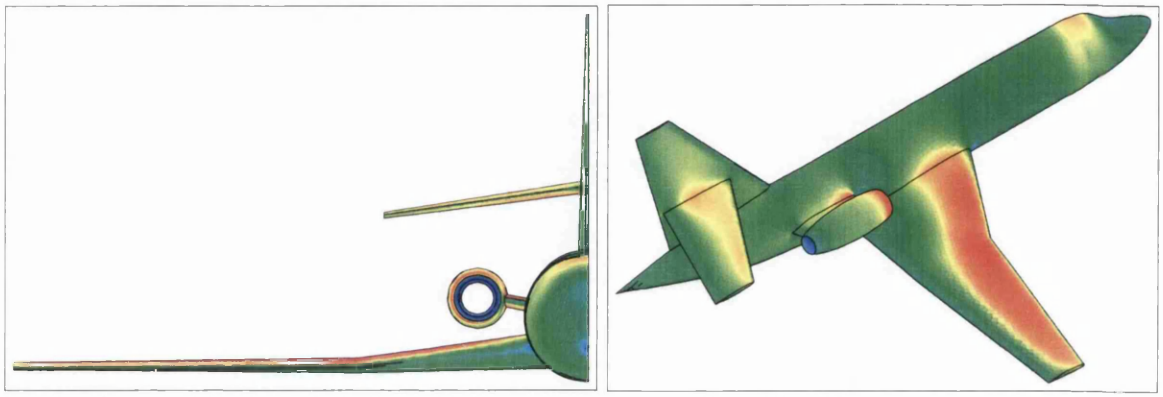


Figure 6.12: Density contours for the Dassault Falcon at 0.84

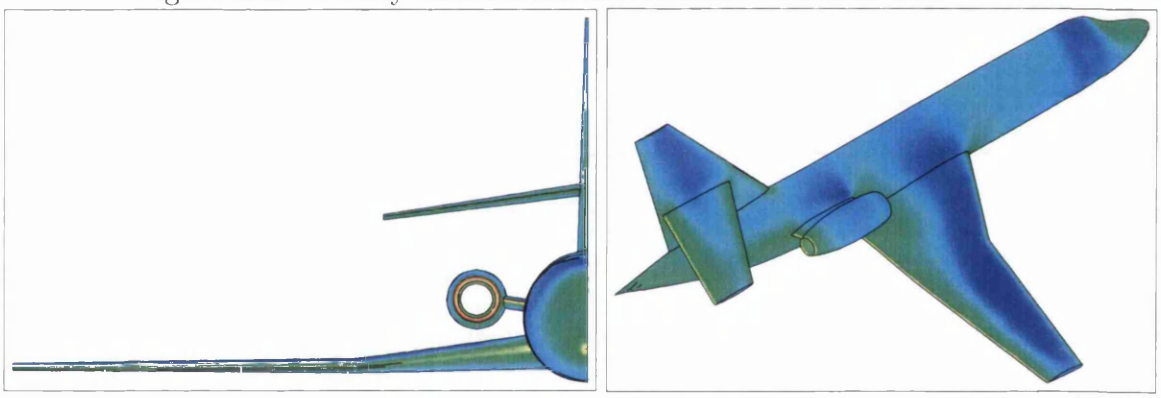


Figure 6.13: Horizontal velocity contours for the Dassault Falcon at 0.84

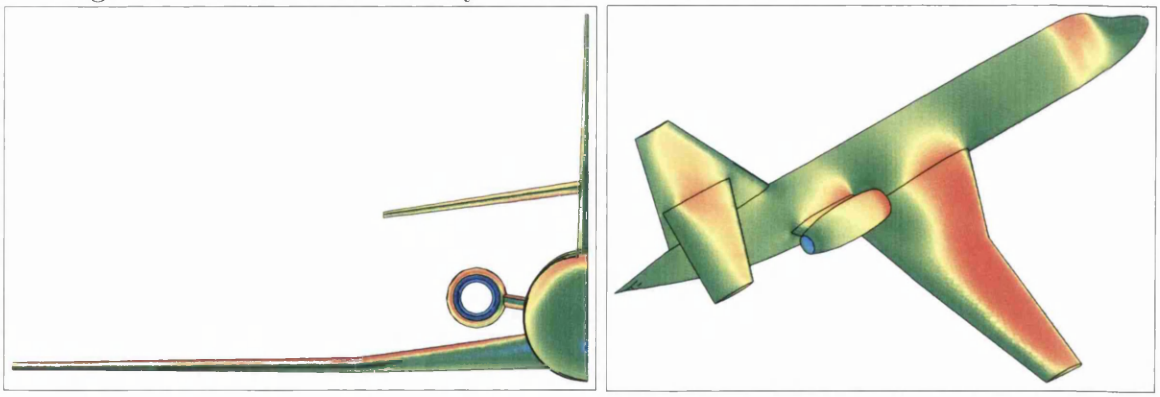


Figure 6.14: Pressure contours for the Dassault Falcon at 0.84

An area of blue represents a high variable value region while an area in red denotes a low variable value region.

is connected to the stabilisation fin and at the engine housing position.

6.6 Chapter Summary

The ability of the Characteristic Base Split (CBS) scheme to model of inviscid flow problems has been examined in this chapter. The application of the CBS scheme to four different test case problems, alongside the results duely generated by the CBS are detailed in the preceding sections. In order to validate the accuracy of the scheme, a selection of the test case results are compared to previously published numerical and empirical values.

The first test case solutions modeled by the CBS scheme are of two dimensional airfoils both incorporating and omitting an angle of attack. The inviscid flow prediction of the CBS scheme was then extended to three dimension tests, both the ONERA wing and Dassault Falcon test cases modeled incorporated an angle of attack.

In addition, the effect of distance to the farfield with relation to the performance of the scheme was examined in this chapter. This examination resulted in the validation that a distance of ten chord lengths to the boundary was sufficient to avoid any disruption to the solution accuracy for the considered flow problems.

Chapter 7

Laminar Viscous Flow Problems

7.1 Introduction

In this chapter the consideration of the CBS scheme's performance shall be extended to regard viscous flow cases. The difference between viscous flow cases of this chapter against the inviscid flow cases encountered earlier is that the simulated viscous flows consist of a real fluid rather than ideal inviscid fluid.[69]

The presence of a real fluid is represented by the diffusive flux vector of the Navier Stokes equations, added to the inviscid version of the code the CBS solver ultilised in the previous chapter. The diffusive vector is discussed in full expanded form in the second chapter of this dissertation denoted by the term G_j in the Navier Stokes equations and therefore shall not be repeated here.[70]

The viscous stress and heat conduction terms contained in the diffusive flux vector are necessary to simulate the viscosity dependent behaviour of a real fluid. Note that in certain cases, the nature of viscous flow problems found in the aerospace industry can be said to be comprised of two components. These two components are firstly the large regions of essentially inviscid flows and secondly the smaller regions existent closer to the solid surfaces where the viscous terms dominate due to the effect of the boundary layer.[57]

In the proceeding sections of this chapter, viscous flow problems shall be presented, commencing with basic viscous flow cases and then following viscous flow cases of increasing gradual complexity. The viscous solutions considered in this chapter include two dimensional and three dimensional flows past a NACA0012

airfoil and conclude with the prediction of flow past a double ellipsoid.

The results computed by the CBS scheme are compared to those published by other authors in order to compare the accuracy. Note that the published results plotted in this thesis are plotted from a trace of the results observed in literature rather than the actual output data files of these other authors.

7.2 Flow past the NACA0012 Airfoil

7.2.1 Two dimensional NACA0012 flow case

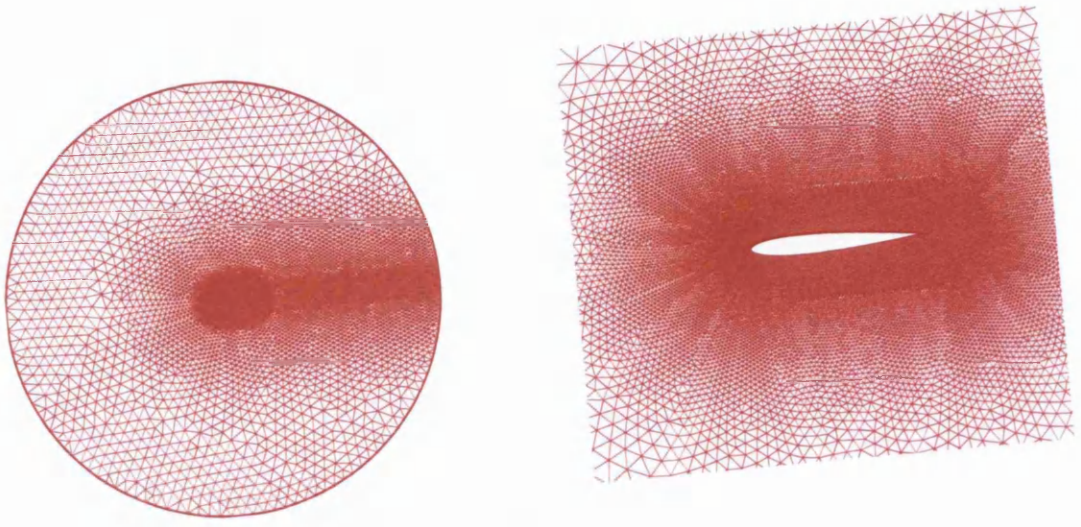
The first solution of a viscous problem to be considered by the CBS algorithm is that of the viscous flow case past a two dimensional NACA0012 airfoil incorporating no angle of attack to the directional components of flow.

The two dimensional domain representing the NACA0012 airfoil is generated utilising an advancing front based mesh generator. As a distance of ten chord lengths to the boundaries, was deemed to be sufficient to obtain a solution free of interference from a farfield for inviscid cases, it is also assumed that this distance shall be sufficient for viscous cases as the viscous effects manifest at the solid geometries.

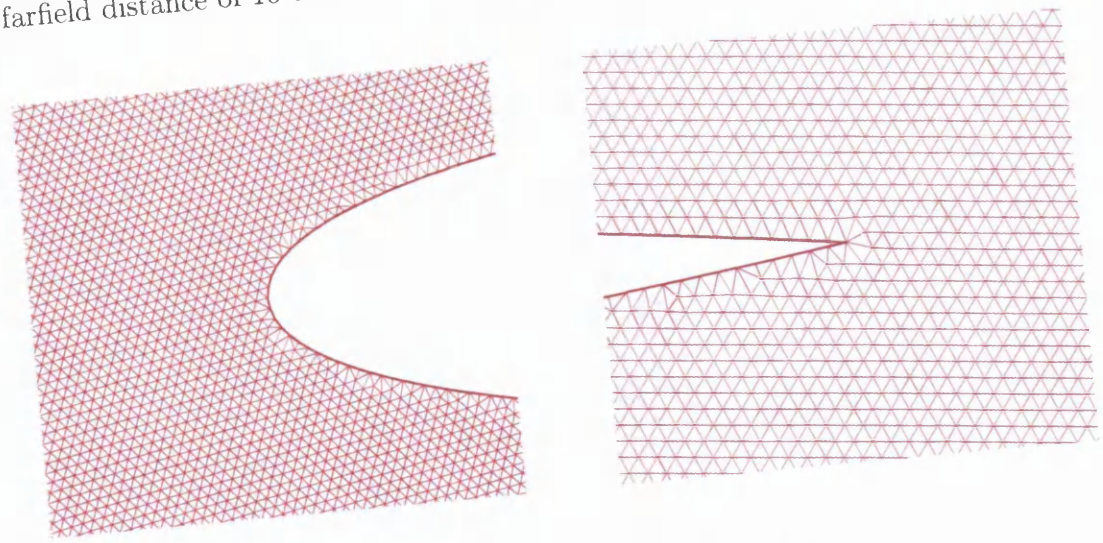
This generated mesh is of unstructured nature, comprising 58,840 triangular elements and 27,636 nodes. The mesh is illustrated in Figure (7.1) for convenience. Note that there are 254 wall points present in the mesh.

A subsonic flow problem is chosen to be the basis of the simple two dimensional flow case, the flow parameters of this flow case are as follows. A free stream speed Mach number of 0.85 is prescribed to simulate the subsonic flow while a Reynold's number of 2,000 establishes that the solution will be of viscous nature. In order to achieve a converged solution an artificial diffusion constant of 0.9 was used in obtaining the following results alongside a residual smoothing coefficient of 0.009. The computed variable contour plots of pressure, density, and horizontal velocity can be seen in Figure (7.2).

To check the quantitative accuracy of these results, the coefficient of pressure and the coefficient of friction calculated from the CBS results are compared against other published results. The coefficient of skin friction is obtained by using the following equation while the coefficient of pressure is given by the Equation (6.1)

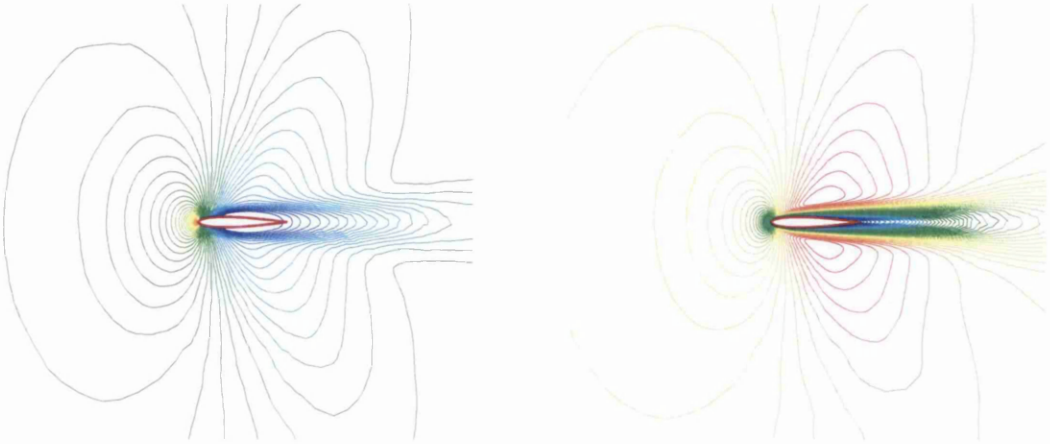


(a) A farfield distance of 10 chord lengths (b) The NACA0012 airfoil



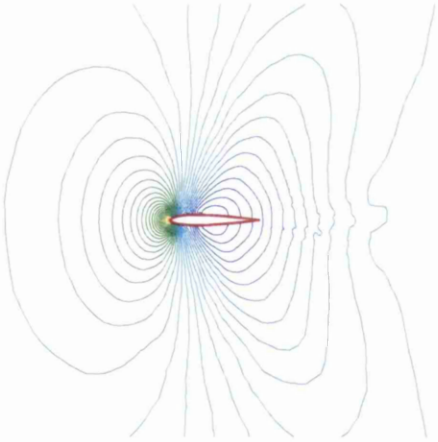
(c) The leading edge (d) The trailing edge

Figure 7.1: The Mesh representing the NACA0012 aerofoil.



(a) Density contours

(b) Horizontal velocity contours



(c) Pressure contours

Figure 7.2: Variable contours for the NACA 0012 aerofoil at Mach number of 0.85, angle of attack of zero degrees and Reynolds number of 2000

stated earlier.

$$C_f = \frac{2\tau_w}{\rho_\infty u_\infty^2} \tag{7.1}$$

The results obtained by the CBS scheme are compared to those primarily of Mittal [71] who obtained his results using a structured mesh comprising 19,014 nodes and 18,772 quadrilateral elements.

In addition to a comparison with the results obtained by Mittal, the coefficient of pressure is also compared with the explicit predictor-corrector algorithm based on the MacCormak's finite difference scheme of Cambier [72] and the combined method of lines and Runge-Kutta method of Satofuka [73]. This co-efficient of pressure comparison is shown in Figure (7.3). Note that while the prediction of the CBS scheme is found to have a good overall match with the other three schemes, there does appear to be some minor oscillations present between the positions of 10% and 30% of chord length along the airfoil. This may be due to a lack of sufficient artificial damping.

The co-efficient of skin friction calculated by the CBS scheme for this test case is compared to the corresponding coefficient of Mittal in Figure (7.4). It is evident from Figure (7.4), that the CBS scheme under-predicts the skin friction peak from an over-diffusive effect while minor oscillations are also present along the first 40% of chord length distance along the airfoil.

In a later section of this chapter, an attempt shall be made to improve the coefficient of skin friction calculated by the CBS scheme using structured layers in the region of the solid surface.

7.2.2 Three dimensional flow past the NACA0012 airfoil

Let a viscous flow case past a three dimensional NACA0012 airfoil wing now be considered. The mesh representing this NACA0012 airfoil wing is generated using the 'Parallel Simulation User Environment' or (PSUE) mesh generator software[66] comprising 1,019,305 tetrahedral elements formulated from 183,564 spatial nodes. This generated mesh is also of a unstructured nature and is illustrated in Figure (7.5).

A supersonic flow problem is chosen to be the basis of this flow case. The simulation parameters of this chosen flow case are stated as the following. A free

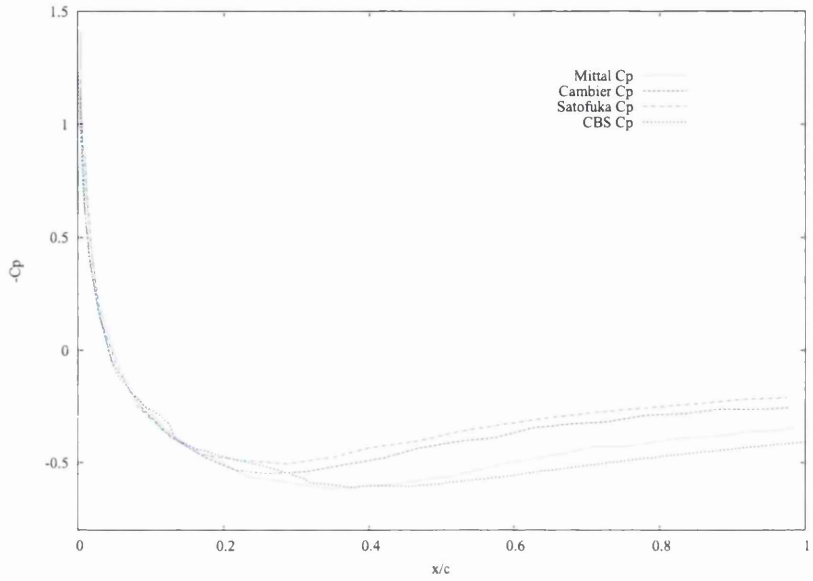


Figure 7.3: Pressure co-efficient for NACA 0012 aerofoil at a Mach number of 0.85, angle of attack of zero degrees and Reynolds number of 2000

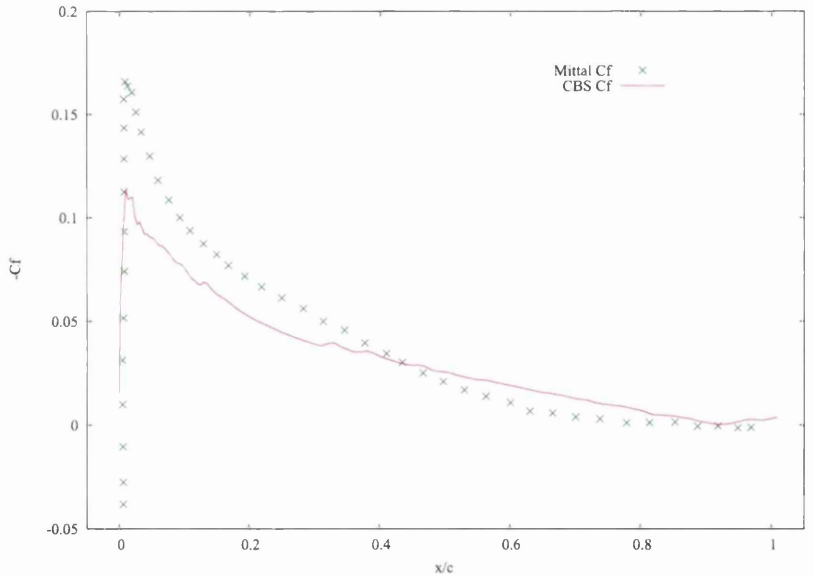


Figure 7.4: Skin friction co-efficient for NACA 0012 aerofoil at Mach number of 0.85, angle of attack of zero degrees and Reynolds number of 2000

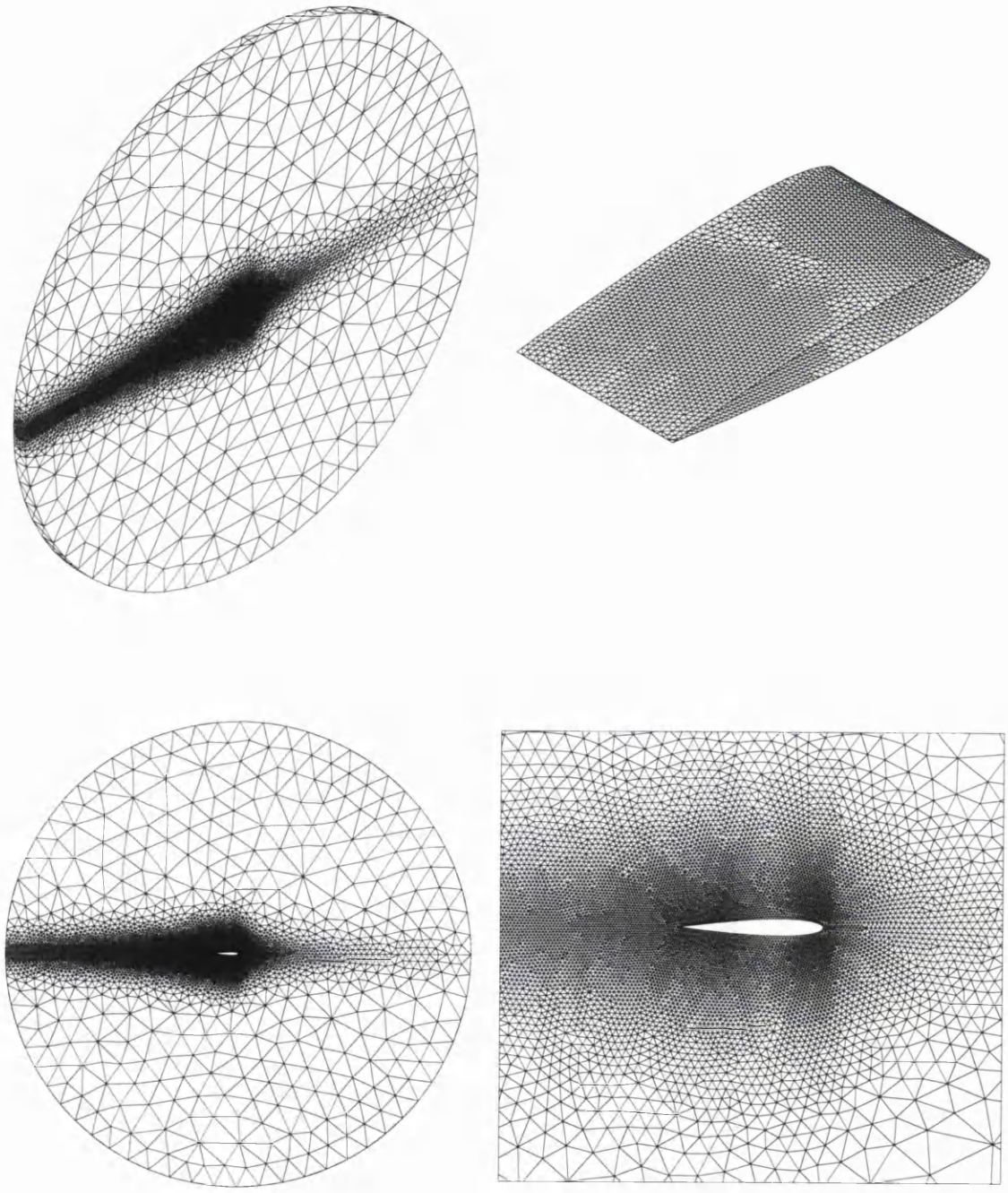
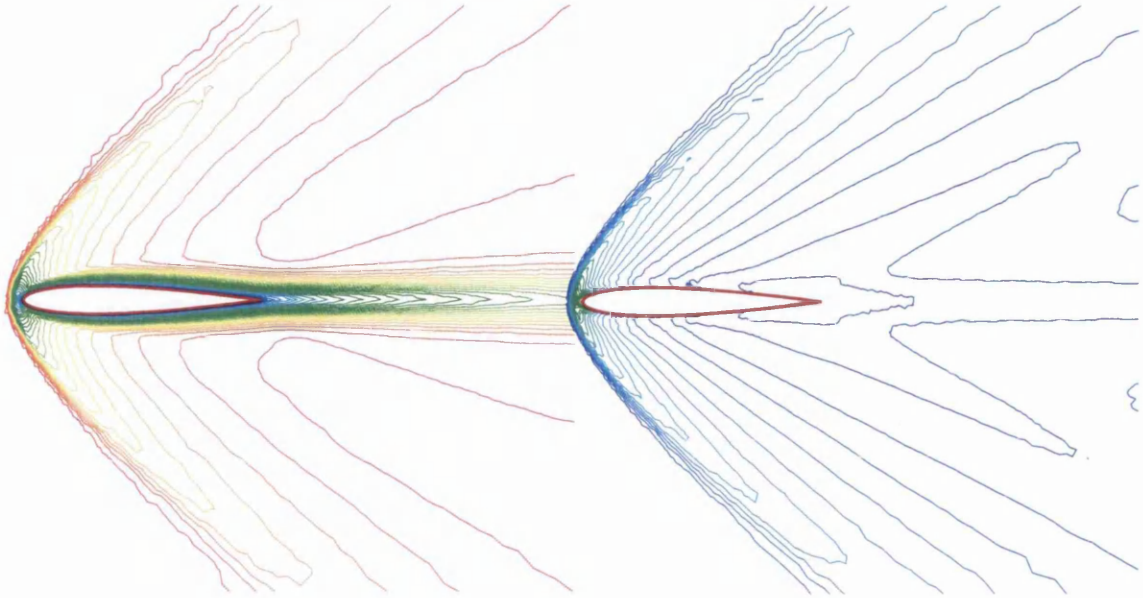
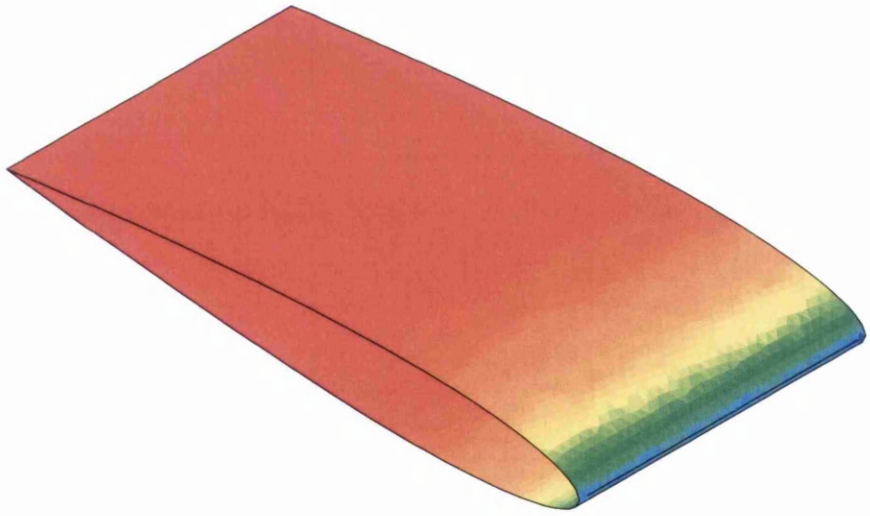


Figure 7.5: The 3D NACA 0012 mesh



(a) Horizontal velocity contours (b) Density contours



(c) Density surface contours

Figure 7.6: Variable contours for a Mach number of 2.0, and Reynolds number of 10000 with zero angle of attack

stream speed Mach number of 2.00 is prescribed for the supersonic flow, while a Reynold's number of 10,000 determines that the solution will be of viscous nature.

There is no angle of attack prescribed to either of the direction of flow components, an artificial diffusion constant of 0.2 is used alongside a residual smoothing coefficient of 0.001 to ensure a converged solution. The variable contour plots of pressure, density are shown in Figure (7.6) along with a three dimensional density surface depiction.

In order to compare the accuracy of the CBS scheme, the coefficient of pressure and the coefficient of skin friction are compared against the published results of Castro-Diaz et al[74] in Figures (7.7) and (7.8) respectively.

The mesh used by Castro-Diaz was of adaptive nature and was regenerated five times before obtaining the final coefficient of skin friction. The coefficient of skin friction obtained by Castro-Diaz before any adaption is also included in Figure (7.8).

There is a good match observed between the comparison of coefficients of pressure in Figure (7.7). Although, the curve of the CBS coefficient of pressure distribution is observed to oscillate mildly between the positions of 10% and 30% of chord length along the airfoil. This behaviour is similar to that observed from the previous test case and is likely caused by the under-diffusive nature of the solution.

7.3 Two dimensional flow case past a NACA0012 airfoil with an angle of attack

In the second two dimensional flow case considered, an angle of attack of ten degrees is applied to the direction of flow past the NACA0012 airfoil.

The domain representing the NACA0012 airfoil is the previously discussed unstructured mesh illustrated in Figure (7.1), comprising 58,840 triangular elements and 27,636 nodes.

The flow parameters for this test case are a Mach 0.85 and a Reynold's number of 10000 establishing a subsonic and viscous solution respectively. In obtaining a converged solution an artificial diffusion constant of 0.9 is employed alongside a residual smoothing coefficient of 0.009. The resulting variable contour plots, obtained by the CBS scheme, of density, pressure and horizontal velocity are illus-

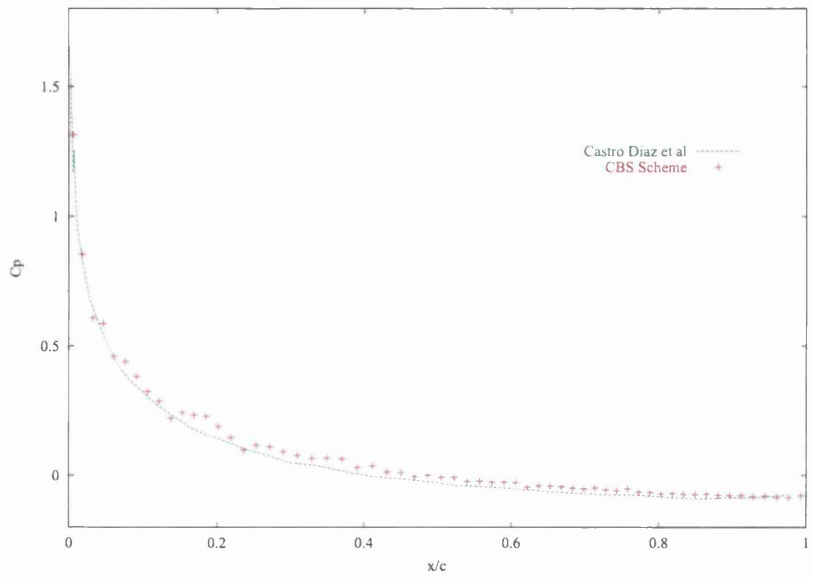


Figure 7.7: Pressure co-efficient for NACA0012 aerofoil at a Mach number of 2.00 and Reynolds number of 10000 with zero angle of attack

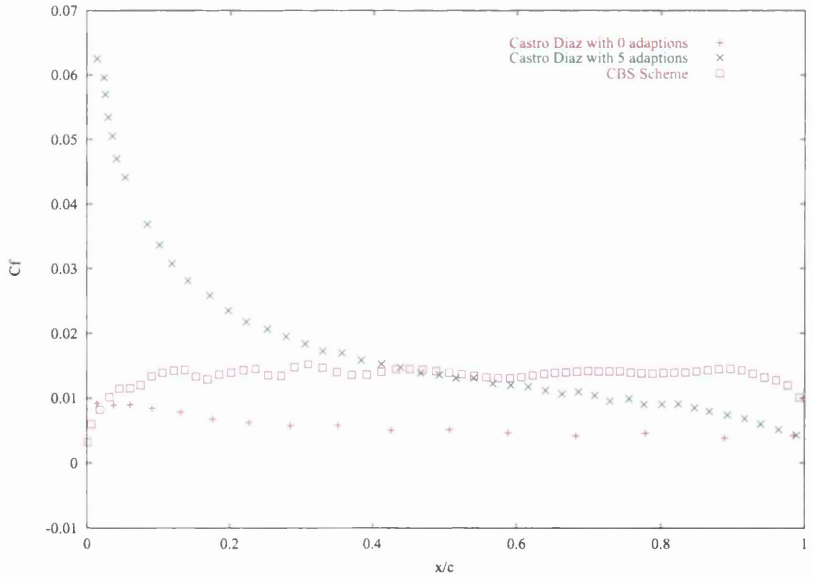


Figure 7.8: Skin friction co-efficient for NACA0012 aerofoil at a Mach number of 2.00 and Reynolds number of 10000 with zero angle of attack

trated in Figures (7.9).

To check the quantitative accuracy of these results, the coefficient of pressure and the coefficient of friction are compared against those published by Mittal in Figure (7.10) and in Figure (7.11).

The comparison of the CBS scheme's coefficient of pressure prediction along the lower surface of the airfoil is observed to be quite close to the published result. Although, there is again evidence of small oscillations present between the positions of 10% and 30% of chord length along the upper surface of the airfoil. In addition the CBS disagrees with the published work in the region of adverse pressure gradient near the rear section of the upper surface of the airfoil and at the trailing edge.

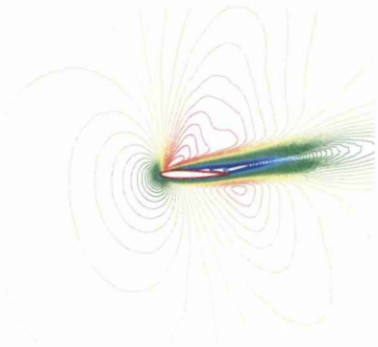
The prediction of co-efficient of skin friction by the CBS scheme is observed from Figure (7.11). In addition to the disagreement, oscillations are present along the first 20% of chord length along the airfoil.

In the next section of this chapter, an attempt shall be considered to improve the performance of the CBS scheme in predicting the coefficient of skin friction.

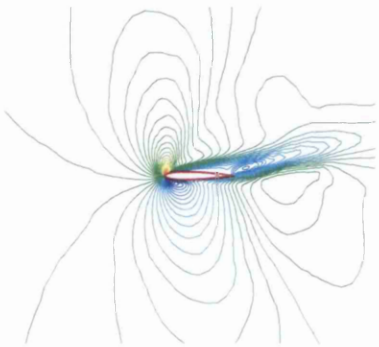
7.4 Skin friction distribution using structured layers in the boundary layer region

In the preceding sections of this chapter, it has been observed that the coefficient of skin friction distributions computed by the CBS scheme have been of lower quality when compared to those of other results. It is noted however that the results computed by the CBS scheme were obtained from unstructured non-adaptive meshes while the corresponding published results were computed by use of finely structured and adaptive meshes. The results will now be computed by the CBS scheme on meshes incorporating a structured layer. It is hoped that the presence of this structured layer shall improve the coefficient of skin friction predictions.

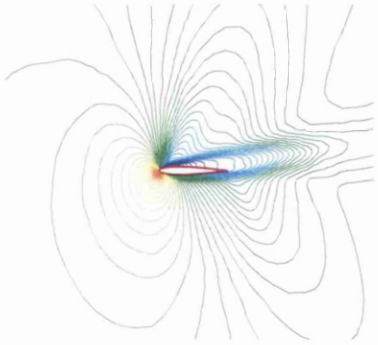
The use of an unstructured mesh incorporating a structured layer still maintains advantages over the adoption of a completely structured mesh. The structured layer is formed first in the generation to a user specified size and refinement, then the rest of the unstructured domain is generated and connected to the interfacing nodes of the structured layer. This manner of mesh generation allows a structured



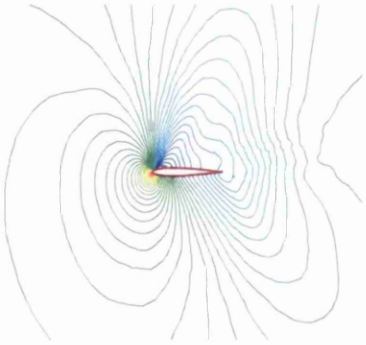
(a) Horizontal velocity contours



(b) Vertical velocity contours



(c) Density contours



(d) Pressure contours

Figure 7.9: Variable contours for ten degree angle of attack past the NACA0012 at Reynolds number of 10000 and Mach number of 0.85

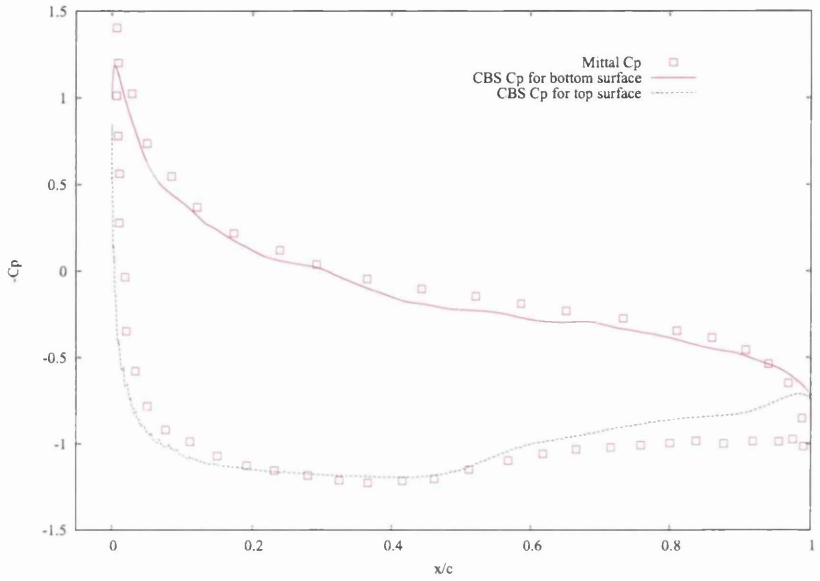


Figure 7.10: Pressure co-efficient for NACA 0012 aerofoil at a Mach number of 0.85, angle of attack of ten degrees and Reynolds number of 10000

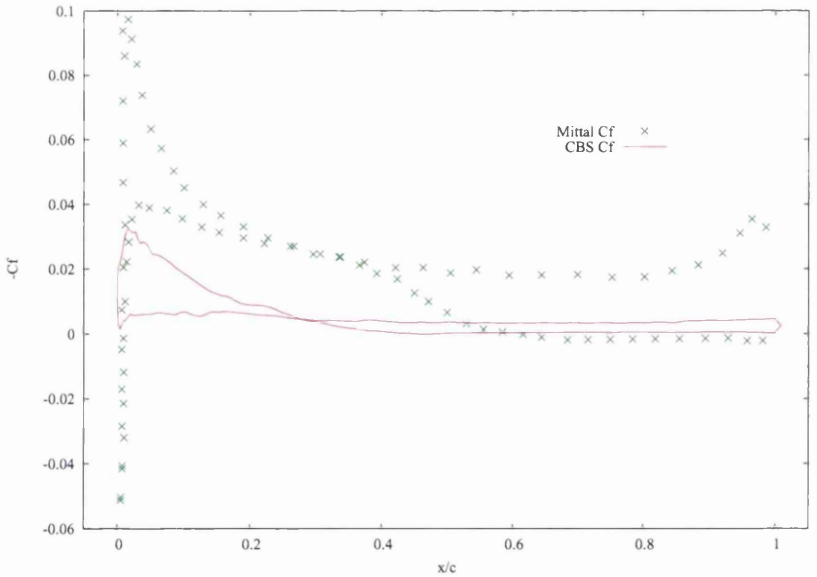


Figure 7.11: Skin friction co-efficient for NACA 0012 aerofoil at a Mach number of 0.85, angle of attack of ten degrees and Reynolds number of 10000

layered mesh to be generated much quicker than a completely structured mesh as the generation of the unstructured part of the mesh comprising the majority of the domain remains fully automatic.

The structured layered mesh used in this dissertation was generated in the following manner, firstly a three dimensional flow case was generated using a mesh generator developed by Hassan and coworkers [75, 76]. The three dimensional mesh was then reduced to two dimensions by a mesh convertor written by the author. This task was accomplished by selecting a symmetric boundary of the three dimensional mesh and renumbering the information to form a two dimensional mesh.

The resultant two dimensional version obtained by this process comprises approximately 56,858 elements and 28750 nodes as shown in Figure (7.12). Note that the size of the generated mesh is close to the size of the completely unstructured mesh utilised earlier. This similarity in size allows the comparison of accuracy to be based purely on the presence or absence of a structured layer rather than influence from noticeable differences in the sizes of the meshes.

The viscous flow cases discussed in the previous two dimensional sections are simulated on the structured layered mesh. The coefficient of skin friction distributions computed by the CBS scheme are given in Figure (7.13) and Figure (7.14). It is immediately evident that the incorporation of the structure layer has increased the accuracy of the predictions substantially compared to the previous predictions in Figure (7.11) and Figure (7.4).

7.5 Flow past a Double Ellipsoid

A more complicated three dimensional geometry of a double ellipsoid was chosen for the last viscous flow case. A double ellipsoid is a generic geometry utilised for representing the frontal section of a high speed aircraft such as a military jet or a space shuttle.

The mesh of the double ellipsoid used in this dissertation was also generated by the PSUE mesh generation application discussed earlier. The generated mesh comprises 441,760 tetrahedral elements, 79,023 coordinate points and 15,400 triangular boundary edges as illustrated in Figure (7.16).

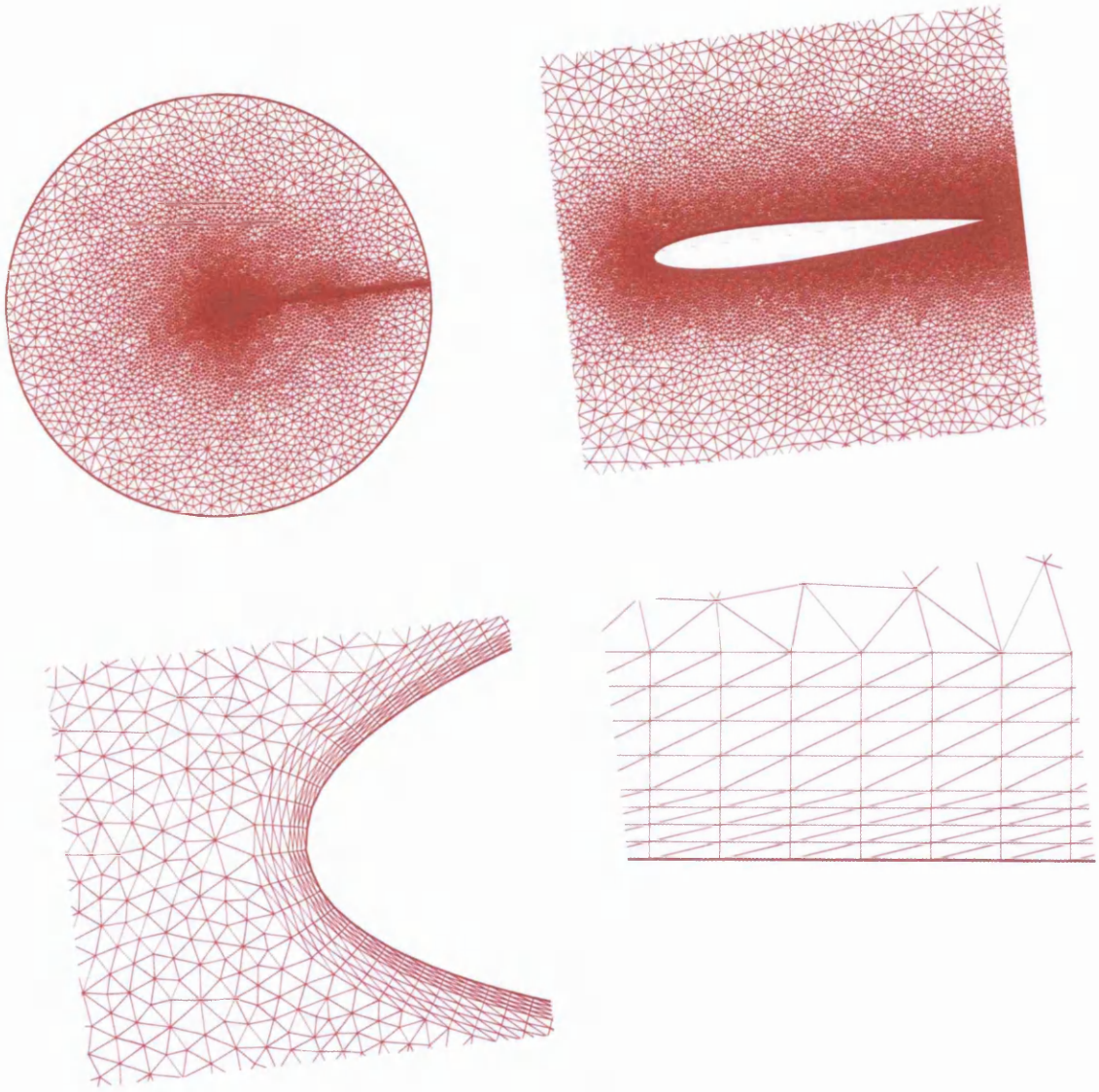


Figure 7.12: The structured layered mesh

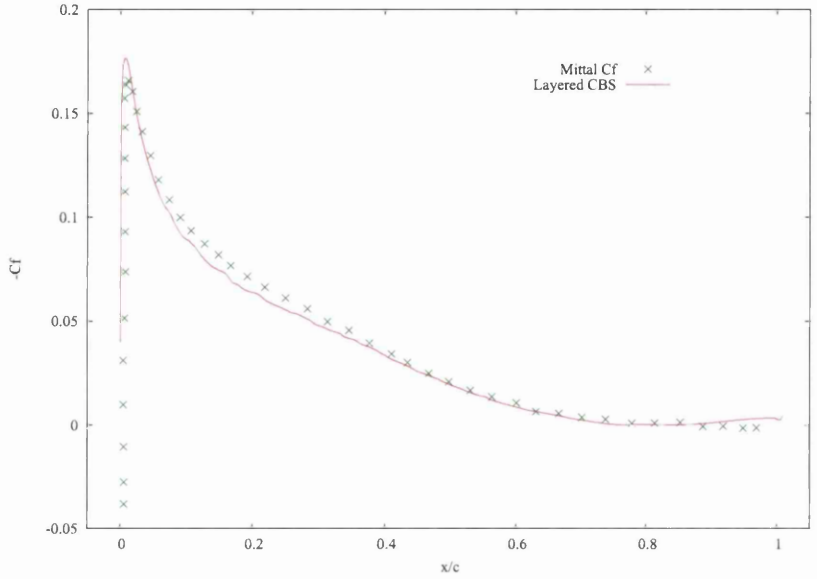


Figure 7.13: Skin friction co-efficient for NACA0012 aerofoil at a Mach number of 0.85 and Reynolds number of 2000 with a zero degree degree angle of attack, generated from a mesh incorporating a structured layer

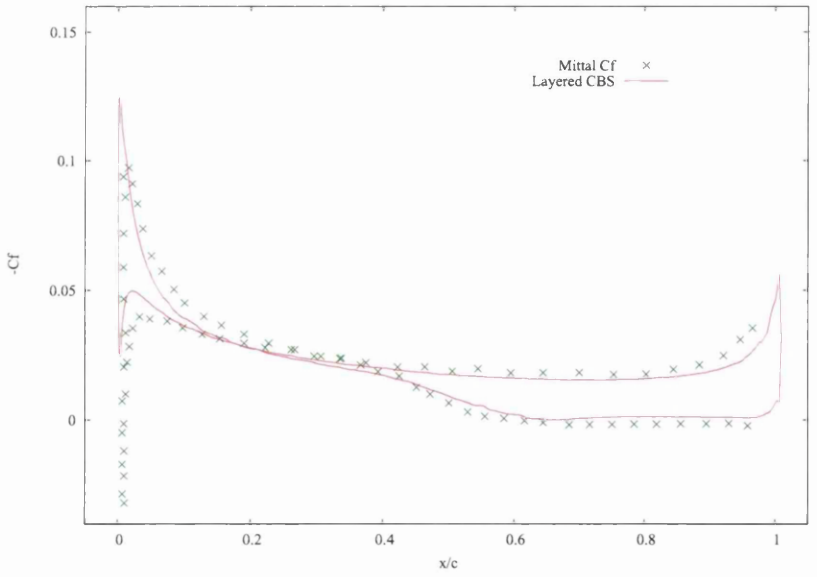


Figure 7.14: Skin friction co-efficient for NACA0012 aerofoil at a Mach number of 0.85 and Reynolds number of 10000 with ten degree angle of attack, generated from a mesh incorporating a structured layer

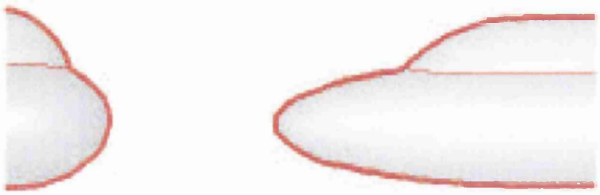


Figure 7.15: A Double ellipsoid

This hypersonic flow case considered here is representative of the space shuttle at the reentry stage of flight. The flow parameters of the flow case are as follows: firstly an angle of attack of 30.0 degrees is imposed on the flow direction past the lengthwise axis of the solid geometry to represent the inclination of the space shuttle at reentry. A high mach number of 8.125 is set as the free stream value to represent the hypersonic flow and a Reynolds number of 100,000 is applied. The resultant variable contour plots computed by the CBS scheme are shown in Figure (7.17) for both the horizontal and vertical velocity components along with the pressure and density contours.

Examination of Figure (7.17) shows that certain expected observations can be made, notably that the highest values of the density and pressure field have occurred at the nose position of the double ellipsoid/aircraft where the incoming flow would have perpendicularly intersected with the surface of the geometry. Relatively high values of pressure and density are also observed along the surface following the intersection point on the underside of the ellipsoid.

7.6 Chapter Summary

In this chapter, the ability of the characteristic base split scheme (CBS) to model both two dimensional and three dimensional viscous flow problems has been examined.

It has been shown that CBS scheme has been able to model viscous flow past airfoils represented by unstructured meshes simulating flows with and without an

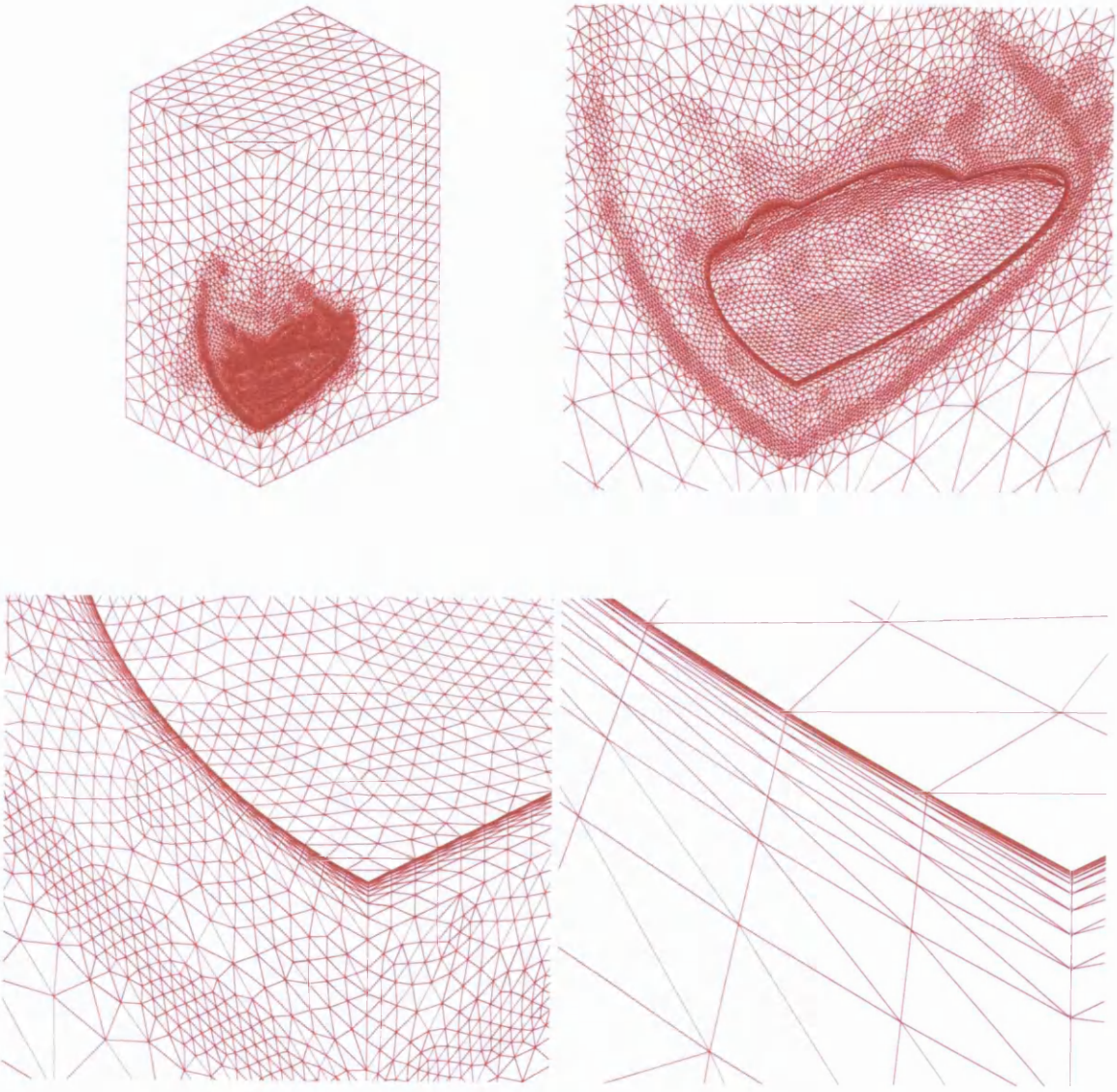
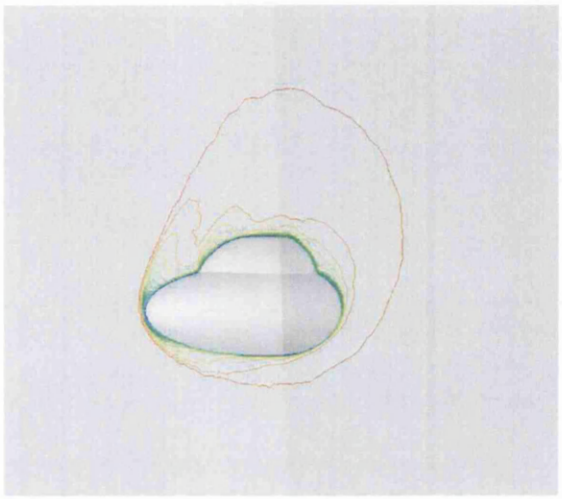
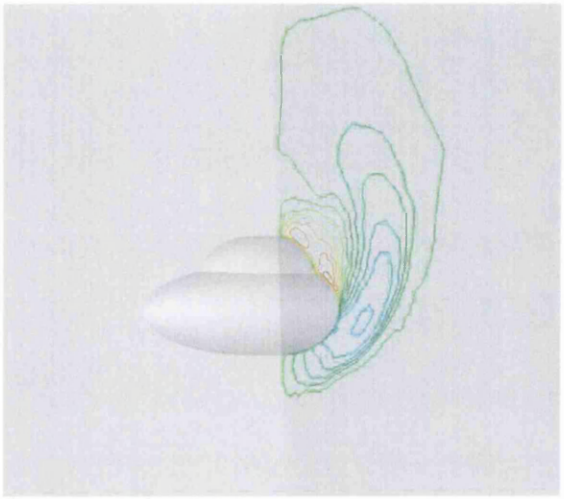


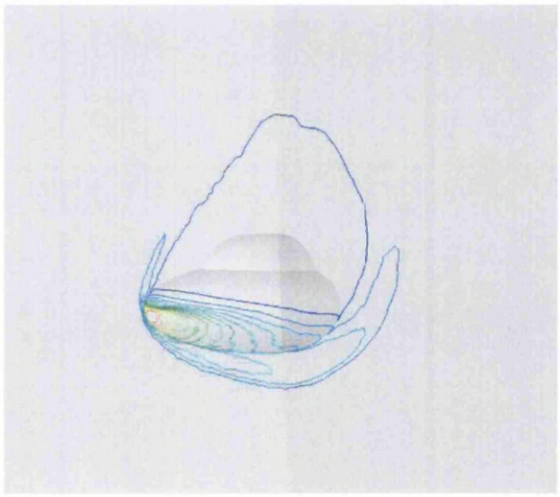
Figure 7.16: Surface mesh of the double ellipsoid



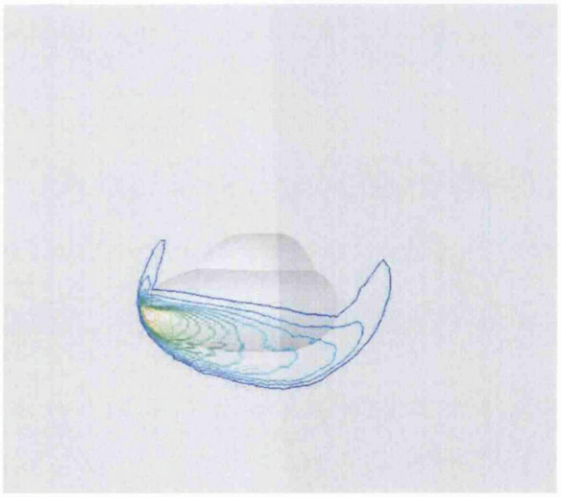
(a) Horizontal velocity contours



(b) Vertical velocity contours



(c) Density contours



(d) Pressure contours

Figure 7.17: Variable contours past the double ellipsoid at a Mach number of 8.125 and Reynolds number of 100,000 along with and angle of attack of 30 degrees

angle of attack.

The accuracy of coefficient of pressure distributions predicted by the CBS scheme using the unstructured meshes has been comparated for the cases without an angle of attack to the corresponding results published by other authors which were obtained on structured and adaptive meshes. The prediction of the coefficient of pressure by the CBS scheme when considering the flow case incorporating an angle of attack was found to disagree with the published results, most notably in the region of the adverse pressure gradient on the upper surface of the airfoil near the trailing edge.

The prediction of the coefficient of skin friction by the CBS scheme disagrees to the corresponding published results unless a structured layer was added to the mesh. Once the structured layered mesh was considered by the CBS scheme, accurate results for the skin friction distributions were obtained.

Lastly, a more complicated three dimensional problem of hypersonic flow past a double ellipsoid was also discussed in this chapter.

Chapter 8

Turbulent flows

The nature of turbulence is inherently time dependent and random, therefore a large amount of information is required to completely describe a turbulent flow leading to impossibly memory expensive solution methods [77, 78]. To obtain a solution, turbulence models such as the one equation Spalart Allmaras model described in chapter three are implemented.

In this chapter, turbulent flow around a NACA0012 aerofoil is modelled using a version of the CBS code incorporating the Spalart Allmaras model. The NACA0012 domain is generated in two dimesnions by the same approach described in previous chapters, using the PSUE simulator to create a three dimensional mesh then reducing that mesh to a two dimensional mesh comprising 11,266 triangular elements and 5,803 coordinate points illustrated in Figure (8.1).

The test case considered [79] is that of turbulent flow past the standard NACA0012 at a Mach number of 0.799, Reynolds number of 9,000,000 and angle of attack of 2.26 degrees. The contours of the computed variables are shown in Figure (8.2) and the resulting coefficient of pressure distribution is shown in Figure (8.3), note that the coefficient of pressure distribution is also compared to that of the Baldwin and Lomax model [80].

As can be seen from Figure (8.3), the pressure distribution calculated by the CBS scheme is of lower quality to those of the published results, calculated on structured meshes especially in the region close to the trailing edge of the aerofoil.

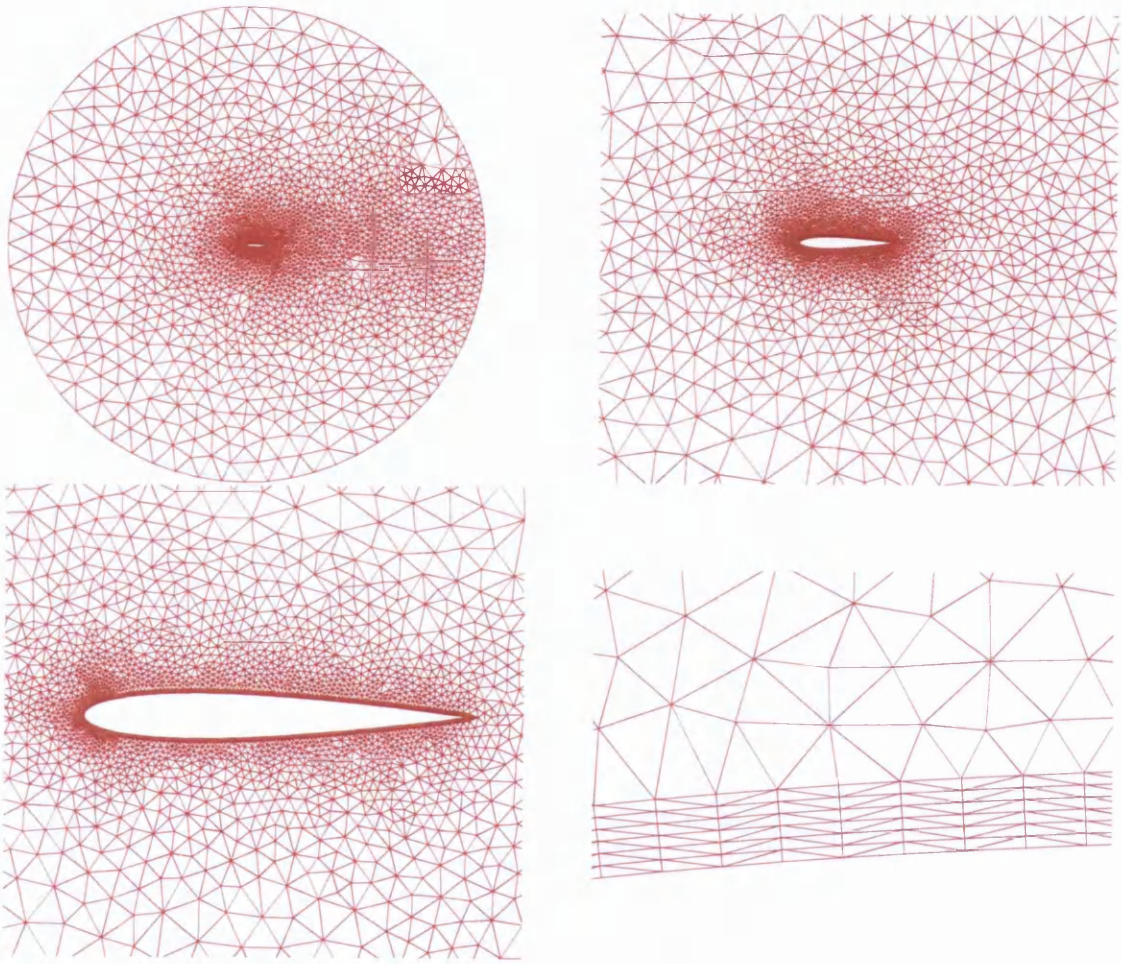
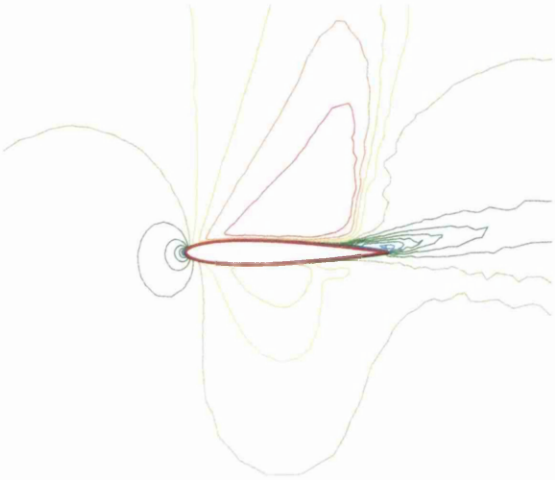
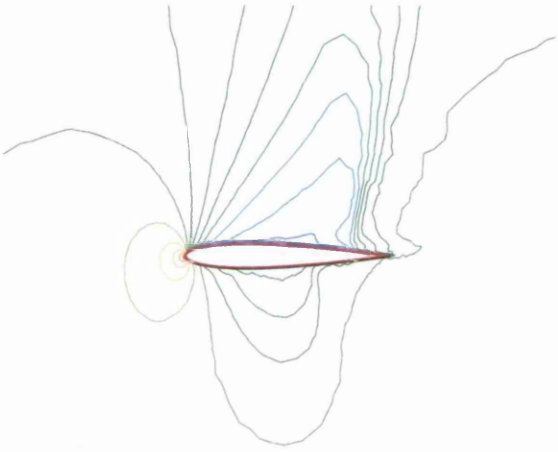


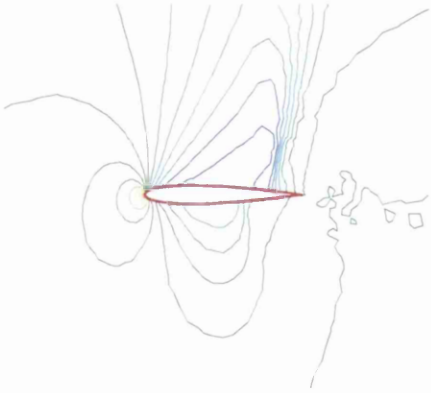
Figure 8.1: Mesh for the NACA0012 aerofoil,11266 triangular elements, 5803 co-ordinate points



(a) Horizontal velocity contours



(b) Density contours



(c) Pressure contours

Figure 8.2: Contours for the NACA0012 aerofoil turbulent flow case at Mach number of 0.799, Reynolds number of 9,000,000 and angle of attack of 2.26 degrees

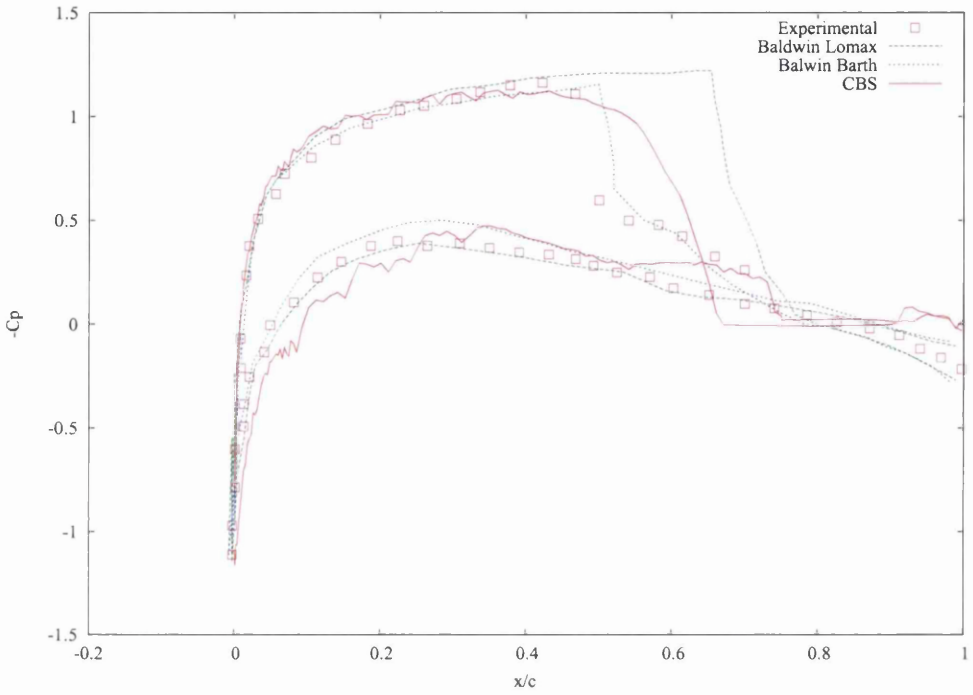


Figure 8.3: Pressure co-efficients for the NACA0012 aerofoil turbulent flow case

8.1 Chapter Summary

In this chapter, the ability of the characteristic base split scheme to model a turbulent flow problem has been examined. The prediction of the coefficient of pressure by the CBS scheme on this mesh was found to be of lower quality to the published results.

Chapter 9

Conclusions

The objective of this dissertation was twofold, firstly to examine the performance of the CBS scheme with particular emphasis on the scheme's robustness. The goal of this examination was to ascertain whether a solver could be created that would obtain results using unstructured meshes of comparable quality to published results obtained on both structured and adaptive meshes.

Secondly due to the explicit nature of the CBS scheme, whether the CBS could successfully incorporate the adoption of speed up techniques such as parallelisation and the edge based formulation technique which are both highly desired by modern industry.

In the course of these examinations, the following findings were concluded which are arranged in an itemised form below.

9.1 Findings

1. The two computational techniques used to obtain a quicker solution, namely parallelisation and the edge based formulation technique have both been successfully incorporated into the CBS explicit fractional step code used in this dissertation, illustrating the robustness and flexibility of the CBS scheme.

2. The level of speed up observed from the implementation of edge based formulation technique was limited due to the high initial preprocessing cost from calculating elemental quantities such as the derivatives of the elemental shape functions.

3. The amount of speed up observed by the application of parallelisation was very substantial when compared with the run times of the corresponding serial code. It was also observed that the number of processors mobilised by the operator when running a parallel code should be carefully considered to maximise the level of speed up efficiency. The importance that a parallel code should keep the amount of inter-processor communications to a minimum is evident, as the total computational cost for these inter-processor communications was seen to increase exponentially as the number of mobilised processors increased.

4. Four inviscid test cases are considered in this dissertation, incorporating two dimensional and three dimensional problems both with and without an angle of attack. The coefficient of friction prediction by the CBS scheme was seen to match well with the published results

5. The effect of distance to the farfield from the geometry also examined in the inviscid flow chapter. An examination found that a distance of ten chord lengths to the boundary was sufficient to avoid any interference to the accuracy from the presence of the exit boundary for the considered flow problems.

6. Regarding the modelling of viscous flow test cases, it was found that CBS scheme was able to obtain converged viscous flow solutions past airfoils represented by unstructured meshes simulating flow problems with and without an angle of attack.

7. The accuracy of the coefficient of pressure distributions predicted by the CBS scheme using the unstructured meshes was found to be close to the published results, obtained on structured and adaptive meshes. The prediction of the coefficient of pressure by the CBS scheme when considering the flow case incorporating an angle of attack was found to disagree to the published results, notably in the region of the adverse pressure gradient on the upper surface of the airfoil near the trailing edge.

8. The predicted coefficient of skin friction by the CBS scheme disagreed with the corresponding published results unless a structured layer was added to

the mesh. Once a layered mesh was used by the CBS scheme, accurate results for the skin friction distributions were obtained.

9. Lastly a turbulent flow case was examined illustrating the robustness of the CBS scheme in incorporating the one equation turbulence model

9.2 Possible further research

It is evident from this document that the Characteristic Based Split scheme obtains accurate laminar results assuming a correct type of mesh is used. Therefore, the main focus of any further work or extension of this work should be directed in increasing the accuracy of obtained turbulent results.

A possible direction may be to extend the CBS solver to incorporate alternative turbulence models, such as the two equation turbulence model which may provide better accuracy. Any increased accuracy obtained by use of a two equation turbulence model would of course arrive at an increased computational cost.

An alternative direction of further research would be the adoption of a Reynolds stress turbulence model or RSTM. The principle concept of the RSTM family of turbulence models is the transportation of the stresses. The advantage of the stress transport model being the inclusion of both nonlocal and historical effects which allow the RTSM to perform well when applied to flows that contain areas with severe adverse pressure gradients[81, 77]. RSTM are unfortunately more complex in nature and more difficult to implement, thus also increasing the computational cost.

Appendix A

Addendum

A.1 overview

In this appendix, the more basic equation that were used in this dissertation shall be referenced here. It was decided to place these equations and relations in this section that than overly extend the main chapters of this dissertation.

A.2 Non-dimensional relations

The complete set of non-dimensional relations shall be referenced in this section along with the non-dimensional quantities such as the Reynold's number that appear after the process of non-dimensionalisation has taken place.

In the followings relation the superscript $*$ denotes a non-dimensionalised quantity and the subscript j denotes the coordinate directions.

Non dimensional velocity is given by the following relation where u_∞ is the free stream velocity.

$$u_j^* = \frac{u_j}{u_\infty}$$

Non dimensional density is given by the following relation where ρ_∞ is the free stream density.

$$\rho^* = \frac{\rho}{\rho_\infty}$$

The non dimensional spatial term is given by the following relation where L is the

specific reference length.

$$x_j^* = \frac{x_j}{L}$$

The non dimensional temporal term used is given by the following relation where t_c is the characteristic time taken from a molecule of fluid to move along the characteristic from x to $x + \Delta x$.

$$t^* = \frac{t}{t_c}$$

Non dimensional pressure is given by the following relation where ρ_∞ is the free stream density and u_∞ is the free stream velocity.

$$p^* = \frac{p}{\rho_\infty u_\infty^2}$$

Non dimensional kinematic viscosity is given by the following relation where ν_∞ is the free stream kinematic viscosity.

$$\nu^* = \frac{\nu}{\nu_\infty}$$

The non dimensional energy is given by the following relation where u_∞ is the free stream velocity.

$$E^* = \frac{E}{u_\infty^2}$$

The non dimensional temperature is given by the following relation where C_p is the specific heat at constant pressure and u_∞ is the free stream velocity.

$$T^* = \frac{T\,C_p}{u_\infty^2}$$

The non dimensional heat capacity at constant pressure is given by the following relation.

$$C_p^* = \frac{C_p}{C_p}$$

Lastly, the non dimensional conductivity ratio is given by the following relation where k_∞ is the reference thermal conductivity.

$$k^* = \frac{k}{k_\infty}$$

A.2.1 Non-dimensional quantities

A number of non-dimensional quantities are created during the non dimensionalisation process of the equations in this dissertation, these quantities are listed and expanded overleaf.

The first quantity is the Reynolds number which is created in the process of deriving the non-dimensional form of the momentum equation, the energy equation and the one equation of the Spalart Allmaras turbulence model. The Reynold's number is shown below where u_∞ is the free stream velocity, ν_∞ is the free stream kinematic viscosity and L is the specific reference length.

$$R_e = \frac{u_\infty L}{\nu_\infty}$$

The Prandtl number is created in the process of deriving the non-dimensional form of the energy equation and is shown below where μ_∞ is the free stream dynamic viscosity, C_p is the heat capacity at constant pressure and k_∞ is the reference thermal conductivity.

$$P_r = \frac{c_p\, \mu_\infty}{k_\infty}$$

The Strouhal number is created in the process of non-dimensionalisating the one equation of the Spalart Allmaras turbulence model and is shown below where u_∞ is the free stream velocity, t_c is the characteristic time and L is the specific reference length.

$$S_t = \frac{t_c\, u_\infty}{L}$$

A.3 Miscellaneous equations

In this section, the basic equations and theorems that are used in the dissertation shall be listed for reference purposes.

A.3.1 Green's Theorem

Green's theorem is named after George Green and is used in this dissertation during discretisation and consequently in the formation of the boundary terms.

Green's theorem is stated as the following.

$$\int_C (\mathbf{L}dx + \mathbf{M}dy) = \int\int_D \left(\frac{\partial \mathbf{M}}{\partial x} - \frac{\partial \mathbf{L}}{\partial y}\right) dA$$

The integrals are over C which is a smooth simple curve and D the the region bounded by C. In order for the above equation to be true $\mathbf{M}$ and $\mathbf{L}$ must be functions of (x, y) defined in the region containing D and have continuous partial derivatives.

A.3.2 Integration by parts

The relation governing integration by parts is used throughout the dissertation and is shown of the two function of x being $f(x)$ and $g(x)$.

$$\int f(x)g'(x)dx = f(x)g(x) - \int f'(x)g(x)dx$$

Appendix B

Message Passing Interface Commands

Introduction

The Message Passing Interface (M.P.I.) was used in this dissertation for the control of the parallel programs. The procedure of implementing M.P.I. and the specific library command subroutines chosen by the author shall be listed and described in more depth in the following sections of this Appendix.

A pure complete version of M.P.I is a library of functions and macros that can be used in C, Fortran and C++ programs, however the commands listed in this document correspond only to the Fortran based subroutines found in the complete M.P.I. library.

Implementation

The implementation of M.P.I. is performed in the following manner (illustrated overleaf). At the beginning of the Fortran code, the text line "include 'mpif.h'" is added which is the command necessary for the compiler to include the relevant Fortran subroutines from the M.P.I. library. The first parallel and last M.P.I. commands stated by the programmer must be "call MPI_INIT(Ierr)" and "call MPI_FINALIZE(Ierr)" respectively. These two commands are responsible for both the initialisation and the termination of the parallel constituent of the current

computational run. The other stated parallel commands must be placed between these two commands

- start of Fortran program
- include 'mpif.h'
- call MPI_Init(Ierr)
- Parallel operations
- call MPI_Finalize(Ierr)
- end program

The "Ierr" flag is a user specific identifying number which shall be outputted back to be user should an error occur while the program executes the corresponding command in order to ease the debugging process.

Relevant MPI Commands

Although the M.P.I. library contains 128 separate subroutine functions, only eight of these are used in the parallelisation procedure of this dissertation, these eight subroutine functions shall now be explained in more detail.

The first two subroutine functions have been mentioned earlier and are utilisation and termination of the parallelise run.

call MPI_Init(Ierr)
call MPI_Finalize(Ierr)

The purposes of the next two function considered are to inform the Fortran code of the number of processors in the parallel architecture and the current processor that is handling the calling section of the code. The returnable information from the above two functions is necessary in the formation of loops over the sending and receiving of information between the individual processors and the required identifying of the processors.

call MPI_COMM_RANK(Comm, Rank, Ierr)

call MPI_COMM_SIZE(Comm, Processors, Ierr)

The arguments of the above commands are as follows, 'Processors' is the returnable value equal to the total number of processors in the parallel architecture. The argument 'Rank' corresponds to the returnable value that informs the calling code which individual processor has handled this request. The argument 'Ierr' is the user specified numerical identifier as before that shall be returned should an error be encounter by the run while attempting to call this function, this purpose of this argument remains unchanged in the following function explanations.

The last argument is the flag 'Comm', this argument is known as the communicator. This communicator argument informs the code which type of communication protocol is currently being used by the parallel architecture, essentially establishing which collections of processors are allowed to communicate with each other. In this dissertation the only communicator used is the well known MPI_COMM_WORLD communicator which may safely be substituted with any future argument 'Comm' encountered by the reader.

As the required information has been gathered using the MPI_COMM_RANK and MPI_COMM_SIZE functions to construct the send/receive statements, the next two function commands to be discussed are those actual functions controlling the sending and receiving of information between the individual processors.

call MPI_SEND(Message, Count, Datatype, Dest, Tag, Comm, Ierr)
call MPI_RECV(Message, Count, Datatype, Source, Tag, Comm, Ierr)

The method in which the MPI_SEND and MPI_RECV functions operate can be illustrated by explanation of the command arguments. The 'Message' argument contains the actual information while the 'Count' argument is a scaler value corresponding to the size of information contain in the message. The identities of the individual processors receiving and sending the message are prescribed by the arguments 'Dest' and 'Source' respectively, 'Comm' and 'Ierr' serve the same purpose as before. The last argument 'Datatype' determines the type of data being sent in the message, the data types of relevance to this dissertation are stated below. Note that addition data types are supported in the M.P.I. library.

MPI_INTEGER
MPI_REAL

MPI_DOUBLE_PRECISION
MPI_CHARACTER

In addition to the MPI_SEND and MPI_RECV functions, there is another way in which information is passed from one processor in other in this dissertation. This alternative method uses the MPI_BCAST below, MPI_BCAST can be used should data be sent from one processor to every other processor in the parallel architecture.

call MPI_BCAST(Buffer, Count, Datatype, Root, Comm, Ierr)

The arguments are similar to those previously encounter for the MPI_SEND and MPI_RECV functions. The argument 'Buffer' contains the information to be sent in the message, the argument 'Count' represents the size of the information to be sent, the 'Datatype' argument represents the type of data being sent in the message while the arguments 'Ierr' and 'Comm' served the same purpose as these arguments served earlier.

The last function to be considered is the function MPI_Barrier, the purpose of MPI_Barrier is to halt all operations and calculations on the processor until each of the other processors has also called the MPI_Barrier function.

call MPI_Barrier(Comm, Ierr)

The utilisation of MPI_Barrier allows the individual processors to be synchronised allowing all send and receive functions to be completed before the code is advanced to another step or time step.

Appendix C

Finite Elements

C.1 Overview

The Finite Element Method shall be discussed in more detail in the course of this appendix. The derivation of the shape functions will also be included in the following sections.

The Finite Element Method(F.E.M.) is a numerical tool utilised for the determination of approximate solutions to engineering problems, F.E.M. operates by dividing the engineering problem under consideration into a given amount of computer comprehendible constitutive problems [82, 83]. It is generally accepted that Clough was the developer of the present day finite element method and it was Clough himself that first coined the term 'finite element' in the context of discrete systems in order to study the stresses in complex airframe structures in his 1960 paper *The finite element in plane stress analysis.*

The presence of (F.E.M.) is found in many research fields and in many forms including direct continuum elements, piecewise continuous trial functions and variational finite differences, among others;[30][4][84].

C.2 Two Dimensional Formulation

The derivation of the Finite Element Method shall be explained by considering the application of the method to a two dimensional problem represented by the domain illustrated in Figure (C.1). The two coordinate directions that the consid-

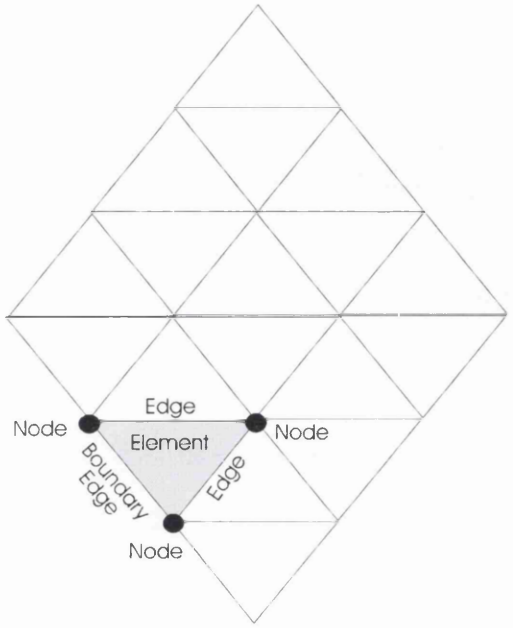


Figure C.1: The discretisation Process

ered domain is subjected to shall be represented by the notation x and y. Let an individual element of the domain now be considered for the mathematical formulation of the scheme. This individual element is illustrated in Figure (C.2) below, the function T denotes any arbitrary flow field variable while the three nodes of the triangular element are represented by the notation i, j and k.

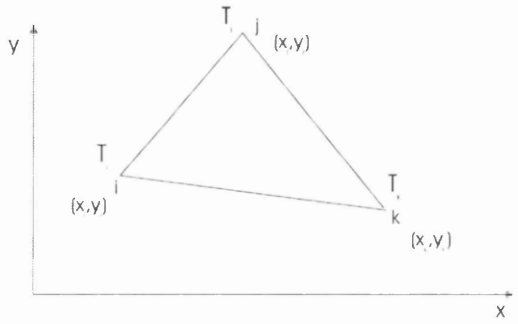


Figure C.2: A two dimensional triangular finite element

As the value of the flow variable T varies throughout the domain but is a given value at any single point, T can therefore said to be dependent on its position with

respect to x and y.

$$T(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{C.1}$$

Extending this relation in the above Equation (C.1) to the considered element illustrated in Figure (C.2), the values of the flow variable T at each of the nodes forming the triangular element may be found using the following set of relations.

$$\begin{aligned} T_i &= \alpha_1 + \alpha_2 x_i + \alpha_3 y_i \\ T_j &= \alpha_1 + \alpha_2 x_j + \alpha_3 y_j \\ T_k &= \alpha_1 + \alpha_2 x_k + \alpha_3 y_k \end{aligned} \tag{C.2}$$

A Solution of the above equations for the values of the alpha coefficients results in the following

$$\begin{aligned} \alpha_1 &= \frac{1}{2A}\left[(x_j y_k - x_k y_j)T_i + (x_k y_i - x_i y_k)T_j + (x_i y_j - x_j y_i)T_k\right] \\ \alpha_2 &= \frac{1}{2A}\left[(y_j - y_k)T_i + (y_k - y_i)T_j + (y_i - y_j)T_k\right] \\ \alpha_3 &= \frac{1}{2A}\left[(x_k - x_j)T_i + (x_i - x_k)T_j + (x_j - x_i)T_k\right] \end{aligned} \tag{C.3}$$

The term A denotes the elemental area which is evaluated from the determinant and rewritten as in the following manner.

$$\begin{aligned} 2A &= det\begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix} \\ &= (x_i y_j - x_j y_i) + (x_k y_i - x_i y_k) + (x_j y_k - x_k y_j) \end{aligned} \tag{C.4}$$

The value of the flow variable T within the considered element is dependent on the nodal values of T and the elemental shape functions denoted by N_*. Equation (C.5) below interpolates the value of T at any point in the element from the nodal values of T.

$$\begin{aligned} T &= N_i T_i + N_j T_j + N_k T_k \\ &= \begin{bmatrix} N_i & N_j & N_k \end{bmatrix} \begin{Bmatrix} T_i \\ T_j \\ T_k \end{Bmatrix} \end{aligned} \tag{C.5}$$

Substitution of the relations derived in Equation (C.2) and Equation (C.3), the values of the shape functions may be rewritten as the following.

$$\begin{aligned} N_i &= \frac{1}{2A}(a_i + b_i x + c_i y) \\ N_j &= \frac{1}{2A}(a_j + b_j x + c_j y) \\ N_k &= \frac{1}{2A}(a_k + b_k x + c_k y) \end{aligned} \tag{C.6}$$

The values of the individual constants are

$$\begin{aligned} a_i &= x_j y_k - x_k y_j \quad & b_i &= y_j - y_k \quad & c_i &= x_k - x_j \\ a_j &= x_k y_i - x_i y_k \quad & b_j &= y_k - y_i \quad & c_j &= x_i - x_k \\ a_k &= x_i y_j - x_j y_i \quad & b_k &= y_i - y_j \quad & c_k &= x_j - x_i \end{aligned} \tag{C.7}$$

The derivatives of the flow variable T with respect to the two coordinate directions can be written in terms of the corresponding shape functions derivatives which may be in turn written in terms of the coefficients stated in Equation (C.6).

$$\begin{aligned} \frac{\partial T}{\partial x} &= \frac{\partial N_i}{\partial x} T_i + \frac{\partial N_j}{\partial x} T_j + \frac{\partial N_k}{\partial x} T_k \\ &= \frac{b_i}{2A} T_i + \frac{b_j}{2A} T_j + \frac{b_k}{2A} T_k \end{aligned} \tag{C.8}$$

$$\begin{aligned} \frac{\partial T}{\partial y} &= \frac{\partial N_i}{\partial y} T_i + \frac{\partial N_j}{\partial y} T_j + \frac{\partial N_k}{\partial y} T_k \\ &= \frac{b_i}{2A} T_i + \frac{b_j}{2A} T_j + \frac{b_k}{2A} T_k \end{aligned} \tag{C.9}$$

Alternatively the two dimensional finite element formulation may be written in vector form.

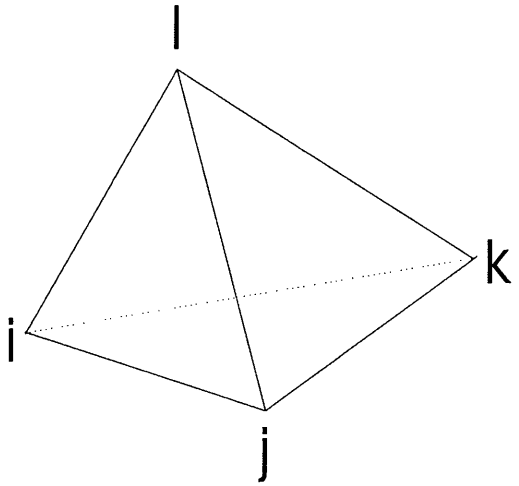
$$\{\mathbf{g}\} = \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix} \begin{Bmatrix} T_i \\ T_j \\ T_k \end{Bmatrix} = [\mathbf{B}]\{\mathbf{T}\} \tag{C.10}$$

The vector form is the more commonly encountered reference format.

C.3 Three Dimensional Formulation

The two dimensional finite element formulation will now be extended to add a third dimension, the extension to the three dimensional form shall omit certain steps for brevity.

Considering the domain in Figure (C.3) should a three dimensional problem be considered, an individual three dimensional element is chosen to be tetrahedral in shape formed using an additional fourth node l.



The value of the flow variable T now varies throughout the domain with respect to another third coordinate denoted by the notation z. The value of T is now said

to be dependent on its position with respect to x, y and z.

$$T(x,y,z) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \tag{C.11}$$

In similar manner to the two dimensional case, the values of the flow variable T at each of the four nodes forming the tetrahedral element may be found using the following set of relations.

$$\begin{aligned} T_i &= \alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 z_i \\ T_j &= \alpha_1 + \alpha_2 x_j + \alpha_3 y_j + \alpha_4 z_j \\ T_k &= \alpha_1 + \alpha_2 x_k + \alpha_3 y_k + \alpha_4 z_k \\ T_l &= \alpha_1 + \alpha_2 x_l + \alpha_3 y_l + \alpha_4 z_l \end{aligned} \tag{C.12}$$

In the rearranging of the above four relations in order to find the four alpha coefficients, the volume is obtained and is again evaluated from the determinant.

$$6V = det \begin{bmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_l & y_l & z_l \end{bmatrix} \tag{C.13}$$

Using the same approach as in the two dimensional formulation, the value of the flow variable T in the element is related to the elemental shape functions.

$$T = N_i T_i + N_j T_j + N_k T_k + N_l T_l \tag{C.14}$$

Substitution of the alpha coefficients into the four relations of Equation (C.12), results in the four shape functions listed below in Equation (C.15) being written in terms of the elemental volume and nodal positions.

$$
\begin{aligned}
N_i &= \frac{1}{6V}\left(a_i + b_i x + c_i y + d_i z\right) \\
N_j &= \frac{1}{6V}\left(a_j + b_j x + c_j y + d_j z\right) \\
N_k &= \frac{1}{6V}\left(a_k + b_k x + c_k y + d_k z\right) \\
N_l &= \frac{1}{6V}\left(a_l + b_l x + c_l y + d_l z\right)
\end{aligned} \tag{C.15}
$$

The constants of the three dimensional case are more complex than their two dimensional counterparts and are represented in full below.

$$
\begin{aligned}
a_i &= x_j\left(y_k z_l - y_l z_k\right) - x_k\left(y_j z_l - y_l z_j\right) + x_l\left(y_j z_k - y_k z_j\right) \\
a_j &= x_k\left(y_l z_i - y_i z_l\right) - x_l\left(y_k z_i - y_i z_k\right) + x_i\left(y_k z_l - y_l z_k\right) \\
a_k &= x_l\left(y_i z_j - y_j z_i\right) - x_i\left(y_l z_j - y_j z_l\right) + x_j\left(y_l z_i - y_i z_l\right) \\
a_l &= x_i\left(y_j z_k - y_k z_j\right) - x_j\left(y_i z_k - y_k z_i\right) + x_k\left(y_i z_j - y_j z_i\right) \\
b_i &= -\left(y_k z_l - y_l z_k\right) + \left(y_j z_l - y_l z_j\right) - \left(y_j z_k - y_k z_j\right) \\
b_j &= -\left(y_l z_i - y_i z_l\right) + \left(y_k z_i - y_i z_k\right) - \left(y_k z_l - y_l z_k\right) \\
b_k &= -\left(y_i z_j - y_j z_i\right) + \left(y_l z_j - y_j z_l\right) - \left(y_l z_i - y_i z_l\right) \\
b_l &= -\left(y_j z_k - y_k z_j\right) + \left(y_i z_k - y_k z_i\right) - \left(y_i z_j - y_j z_i\right) \\
c_i &= x_j\left(z_k - z_l\right) + x_k\left(z_l - z_j\right) + x_l\left(z_j - z_k\right) \\
c_j &= x_k\left(z_l - z_i\right) + x_l\left(z_i - z_k\right) + x_i\left(z_k - z_l\right) \\
c_k &= x_l\left(z_i - z_j\right) + x_i\left(z_j - z_l\right) + x_j\left(z_l - z_i\right) \\
c_l &= x_i\left(z_j - z_k\right) + x_j\left(z_k - z_i\right) + x_k\left(z_i - z_j\right) \\
d_i &= x_j\left(y_l - y_k\right) + x_k\left(y_j - y_l\right) + x_l\left(y_k - y_j\right) \\
d_j &= x_k\left(y_i - y_l\right) + x_l\left(y_k - y_i\right) + x_i\left(y_l - y_k\right) \\
d_k &= x_l\left(y_j - y_i\right) + x_i\left(y_l - y_j\right) + x_j\left(y_i - y_l\right) \\
d_l &= x_i\left(y_k - y_j\right) + x_j\left(y_i - y_k\right) + x_k\left(y_j - y_i\right)
\end{aligned} \tag{C.16}
$$

Alternatively the three dimensional finite element formulation may be written in

the following vector form.

$$[\mathbf{B}] = \frac{1}{6V}\begin{bmatrix} b_i & b_j & b_k & b_l \\ c_i & c_j & c_k & c_l \\ d_i & d_j & d_k & d_l \end{bmatrix} \tag{C.17}$$

Once again the vector form is the commonly encountered reference format due to the length and complexity of the three dimensional coefficients.

Bibliography

[1] Nakayama Y. and Boucher R.F. Introduction to Fluid Mechanics. *Arnold Publishing*, 1999.

[2] Murdock J. Fundamental Fluid Mechanics for the Practising Engineer. *Marcel Dekker*, 1993.

[3] Fletcher C. Computational techniques for fluid dynamics. *Springer Verlag*, 1988.

[4] Zienkiewicz O.C. and Taylor R.L. The Finite Element Method. The Basis. *Butterworth Heinemann*, 2000.

[5] Brooks A.N. and Hughes T.J.R. Streamline Upwind Petrov Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier Stokes equation. *Computer Methods in Applied Mechanics and Engineering vol 32 199-259*, 1982.

[6] Hughes T.J.R., Franca L.P., and Hulbert G.M. A new finite element formulation for computational fluid mechanics VIII. The Galerkin Least Squares method for advective-diffusion equations. *Computer Methods in Applied Mechanics and Engineering vol 73 173-189*, 1989.

[7] Lee J.H.W., Peraire J., and Zienkiewicz O.C. The Characteristic Galerkin method for advection-dominated problems-An assessment. *Computer Methods in Applied Mechanics and Engineering vol 61 359-369*, 1987.

[8] Anderson J.D. Computational Fluid Dynamics. The Basics with Applications. *McGraw Hill Publishers*, 1995.

[9] Lohner R., Morgan K., and Zienkiewicz O.C. The solution of nonlinear hyperbolic equations systems by the finite element method. *International Journal of Numerical Methods in Fluids vol 4 1043-1063*, 1984.

[10] Zienkiewicz O.C. and Codina R. A general algorithm for compressible and incompressible flow - Part I: The split, Characteristic Based Scheme. *International Journal for Numerical Methods in Engineering vol 20 869-885*, 1995.

[11] Zienkiewicz O.C., Nithiarasu P, Codina R., Vazquez M., and Ortiz P. The Characteristic Based Split procedure. An efficient and accurate algorithm for fluid dynamics. *International Journal for Numerical Methods in Engineering vol 31 359-392*, 1999.

[12] Liu C.B. and Nithiarasu P. The Characteristic Based Split (CBS) scheme for viscoeleastic flows past a circular cylinder. *International Journal of Numerical Methods in Fluids vol 57 157-176*, 2008.

[13] Babuska I. and Strouboulis T. The finite element method and its reliability. *Clarendon Press*, 2001.

[14] Nithiarasu P., Mathur J.S., Weatherill N., and Morgan K. Three dimensional incompressible flow calculations using the Characteristic Based Split (CBS) scheme. *International Journal for Numerical Methods in Fluids vol 44 1207-1229*, 1993.

[15] Zienkiewicz O.C., Szmelter J., and Peraire J. Compressible and incompressible flow, an algorithm for all seasons. *Computer Methods in Applied Mechanics and Engineering vol 78 105-121*, 1990.

[16] Zienkiewicz O.C. and Wu J. A general explicit or semi-explicit algorithm for compressible and incompressible flows. *International Journal for Numerical Methods in Engineering vol 35 457-479*, 1992.

[17] Hutchinson W. and Wu T.Y. Advances in applied mechanics. *Academic Press*, 1994.

[18] Leschziner M.A. and Drikakis D. Turbulence modelling and turbulent flow computation in aeronautics. *The Aeronautic Journal vol 106 349-384*, 2002.

[19] Cebeci T. and Smith A.M.O. Analysis of turbulent boundary layers. *Academic Press*, 1974.

[20] Mavriplis D.J. Algebraic turbulence modelling for unstructured and adaptive meshes. *AIAA Journal vol 29 2080-2093*, 1992.

[21] Tulapurkara E.G. Turbulence models for the computation of flow past airplanes. *Progress in Aerospace and Sciences vol 33 71-165*, 1997.

[22] Spalart P.R. and Allmaras S.R. A one equation turbulence model for aerodynamic flows. *AIAA paper 92-0439*, 1992.

[23] Baldwin B. and Barth T.J. A one equation turbulence transport model for high Reynolds number wall bounded flows. *AIAA Paper 91-0610*, 1991.

[24] Wilcox D.C. Reassessment of the scale determining equation for advanced turbulence models. *AIAA journal vol 26 1299-1310*, 1988.

[25] Jones W.P. and Launder B.E. The calculation of low Reynolds number phenomena with a two equation model of turbulence. *International Journal of Heat Mass Transfer vol 16 1119-1130*, 1973.

[26] Saffman P.G. and Wilcox D.C. Turbulence model predictions for turbulent boundary layers. *AIAA journal vol 12 541-546*, 1974.

[27] Zienkiewicz O.C., Gallagher R.H., and Hood P. Newtonian and non-Newtonian viscous incompressible flow, temperature induced flows. Finite element solutions. *Mathematics of finite elements and applications, Academic Press vol 2 235-267*, 1976.

[28] Hirsch C. Numerical computation of internal and external flows. Vol 2. computational methods for inviscid and viscous flows. *Wiley Publishing*, 1990.

[29] Zienkiewicz O.C. and Taylor R.L. The Finite Element Method. Fluid Dyanamics. *Butterworth Heinemann*, 2000.

[30] Lewis R.W., Nithiarasu P., and Seetharamu KN. Fundamentals of the finite element method for heat and fluid flow. *Wiley Publishing*, 2004.

[31] Chorin A.J. and Marsden J.E. A mathematical introduction to fluid mechanics. *Springler Verlag*, 1993.

[32] Zienkiewicz O.C. and Wu. J. A semi-explicit finite element scheme for coupled fluid/thermal problems / O. C. Zienkiewicz and J. Wu CR/777/93. *Institute for Numerical Methods in Engineering, University College of Swansea*, 1993.

[33] Massarotti N., Nithiarasu P., and Zienkiewicz O.C. Characteristic Based Split algorithm for incompressible flow problems with heat transfer. *International Journal of Numerical Methods for Heat and Fluid Flow vol 8 969-990*, 1998.

[34] Nithiarasu P. and Ravindran K. A new semi-implicit time stepping procedure for buoyancy driven flow in a fluid saturated porous medium. *Computer Methods in Applied Mechics and Engineering vol 165 147-154*, 1998.

[35] Nithiarasu P. and Zienkiewicz O.C. Analysis of an explicit and matrix free fractional step method for incompressible flows. *Computer Methods in Applied Mechanics and Engineering vol 195 5537-5551*, 2006.

[36] Nithiarasu P. and Liu C.B. An Artificial Compressibility based Characteristic Based Split (CBS) scheme for steady and unsteady turbulent incompressible flows. *Computer Methods in Applied Mechanics and Engineering vol 195 2961-2982*, 2006.

[37] Codina R., Nithiarasu P., and Zienkiewicz O.C. The Characteristic Based Split (CBS) scheme - a unified approach to fluid dynamics. *Internation Journal of Numerical Methods in Engineering vol 66 1514-1546*, 2006.

[38] Nithiarasu P. An efficient Artificial Compressibility (AC) scheme based on Characteristic Based Split (CBS) method for incompressible flows. *International Journal for Numerical Methods in Engineering vol 56 1815-1845*, 2003.

[39] Peraire J., Peiro J., Formaggia L., Morgan K., and Zienkiewicz O.C. Finite element computations in three dimensions. *International Journal for Numerical Methods in Engineering vol 26 2135-2159*, 2004.

[40] Zienkiewicz O.C and Morgan K. and Satya B. and Codina R. and Vazquez M. A general algorithm for flow. Part II: Tests on the explicit form. *International Journal of Numerical Methods in Fluids Vol 20 887-913*, 1995.

[41] Von Neumann J. and Richtmyer R.D. A method for the numerical calculations of hydrodynamical shocks. *Journal of Applied Mathmatical Physics vol 21 232-237*, 1950.

[42] Morgan K., Peraire J., Peiro J., and Zienkiewicz O.C. Adaptive remeshing applied to the solution of a shock interaction problem on a cyclindrical leading edge. *Computional Methods in Aeronautical Fluid Dynamics, Clarenden Press 327-344*, 1990.

[43] Peraire J., Morgan K., Peiro J., and Zienkiewicz O.C. An adaptive finite element method for high speed flows. *AIAA paper 87-0558*, 1987.

[44] MacCormak R.W. and Baldwin B.S. A numerical method for solving the Navier Stokes equations with application to shock boundary layer interaction. *AIAA paper 75-1*, 1975.

[45] Zienkiewicz O.C., Taylor R.L., and Nithiarasu P. The finite element method for fluid dynamics 6th edition. *Butterworth Heinemann*, 2005.

[46] Thomas C.G. and Nithiarasu P. Influences of element size and variable smoothing on inviscid compressible flow solutions. *International Journal for Numerical Methods in Heat and Fluid Flow Vol 15 420-428*, 2005.

[47] Deck S., Duveau P., D'Espiney P., and Guillen P. Development and application of Spalart-Allmaras one equation turbulence model to three-dimensional supersonic complex configurations. *Journal of Aerospace Science and Technology Vol 6 171-183*, 2002.

[48] Tritton D. Physical Fluid Dynamics. *Oxford science publications*, 1988.

[49] The fortran compiler, docmentation availible in download from homepage at. *www.fortran.com/fortran/market.html.*

[50] Compaq. Compaq fortran. language reference manual. *Compaq computer corporation*, 1999.

[51] Morgan K. and Peraire J. Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics Vol 61 569-638*, 1998.

[52] MPI users guide in fortran. introduction to message passing programming. available from. *www.hku.hk/cc/sp2/ftp/mpi.*

[53] CFD General Notation System Advanced Data Format (ADF) user's guide, version 1.1.1 available from. *http://www.grc.nasa.gov/WWW/cgns/adf/adf.pdf.*

[54] An education resource detailing the backround and operating principles of the METIS based library available from. *http://glaros.dtc.umn.edu/gkhomes/metis/metis/overview.*

[55] Karypis G. and Kumar V. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distribuated Computing vol 48 96-129*, 1998.

[56] Thompson P.A. Compressible fluid dynamics. *McGraw Hill*, 1972.

[57] Hirsch C. Numerical computation of internal and external flows. Vol 1. computational methods for inviscid and viscous flows. *Wiley Publishing*, 1988.

[58] Kuo C.H., Chiou L.C., and Chen C.C. Wake flow pattern modified by small control cyclinders at low Reynold's number. *International Journal for Fluids and Structures vol 23 938-956*, 2007.

[59] Afanasyev Y.D. and Korabel V.N. Wakes and vortex streets behind a localised force. Numerical simulations. *Communications in Non-linear Sciences and Numerical Simulations vol 13 1101-1111*, 2008.

[60] Zienkiewicz O.C. and Nithiarasu P. A universal algorithm for fluid dynamics. the Characteristic Based Split. Some tests on stability and boundary conditions. *Technical report, Swansea school of engineering*, 2000.

[61] Satya Sai B.V.K., Zienkiewicz O.C., Manzari M.T., Lyra P.R.M., and Morgan K. General purpose vs special algorithims for high speed flow with shocks. *International Journal of Numerical Methods in Fluids. Vol 27 57-80*, 1998.

[62] Zienkiewicz O.C. and Nithiarasu P. The Characteristic Based Split (CBS) algorithm, stability and boundary conditions. *Archive of Mechanics vol 52 857-887*, 2000.

[63] Nithiarasu P., Zienkiewicz O.C., Morgan K., Satya B.V.K.S., Codina R., and Vazquez M. Shock capturing viscosities for the general fluid mechanics algorithm. *International Journal of Numerical Methods in Fluids. vol 28 1325-1353*, 1998.

[64] Peraire J., Vahdati M., Morgan K., and Zienkiewicz O.C. Adaptive remeshing for compressible flow problems. *Journal of Computational Physics vol 72 449-466*, 1987.

[65] Pulliam T.H. and Barton J.T. AIAA 23rd Aerospace sciences meeting - Euler computations of AGARD working group 07 airfoil test cases. *AIAA Paper 85-0018*, 1985.

[66] Parallel Simulation User Environment (P.S.U.E) package and user documentation. *Department of Civil Engineering Swansea University.*

[67] Schmitt V. and Charpin F. Pressure distributions of the ONERA M6 wing at transonic Mach numbers. Experimental database for computer program assessment. *AGARD Report AR-138*, 1979.

[68] Sorensen K.A. A multigrid acceleration proceedure for the solution of compressible fluid flows on unstructured hybrid meshes. *PhD thesis, Swansea School of Engineering*, 2001.

[69] Chadwick A. and Morfett J. Hydraulics in civil and enviromental engineering. *E and Fn Spon*, 1997.

[70] Von Mises R. and Friedrichs K. O. Fluid dynamics. *Springer Verlag*, 1971.

[71] Mittal S. Finite element computation of unsteady viscous compressible flows. *Computer Methods in applied Mechanics and Engineering vol 157 151-175*, 1998.

[72] Cambier L. Computation of viscous transonic flows using an unsteady type method and a zonal grid refinement technique. *Numerical Simulation of Compressible Navier Stokes Flows, Vol18, Notes of Numerical Fluid Mechanics (Vieweg, Wiesbaden)*, 1987.

[73] Satofuka N., Morinishi K., and Nishida. Numerical solution of two dimensional compressible Navier Stokes equations using rational Runge Kutta method. *Numerical Simulation of Compressible Navier Stokes Flows, Vol18, Notes of Numerical Fluid Mechanics (Vieweg, Wiesbaden)*, 1987.

[74] Castro-Diaz M.J., Hecht F., Mohammadi B., and Pironneau O. Anisotropic unstructured mesh adaption for flow simulations. *International Journal of Numerical Methods in Fluids vol 25 475-491*, 1997.

[75] Hassan O. and et al. Unstructured mesh generation for viscous high speed flows. *School of engineering, University of Wales Swansea*, 1994.

[76] Hassan O. and et al. Generation and adaption of unstructured meshes. *School of engineering, University of Wales Swansea*, 1995.

[77] Pope S.E. Turbulent Flows. *Cambridge Academic Press*, 2000.

[78] Tennekes H. and Lumley J.L. A First Course in Turbulence. *MIT Academic Press*, 1972.

[79] Balwin S.B. and Barth T.J. A one-equation turbulence transport model for high Reynolds number wall bounded flows. *NASA Technical Memorandum 102847*, 1990.

[80] Balwin S.B. and Lomax H.L. Thin layer appromixation and algebraic model for separated flows. *AIAA paper 78-257*, 1978.

[81] Wilcox D.C. Turbulence modelling for CFD. *Griffin Printing*, 1993.

[82] Conte S.D. and De Boor C. Elementary numerical analysis. An algorithmic approach. *McGraw Hill*, 1980.

[83] Thomas J and Hughes R. The Finite Element Method-Linear Static and Dynamic Finite Element Analysis. *Dover*, 2000.

[84] Henwood D. and Bonet B. Finite elements, a gentle introduction. *Macmillan*, 1996.