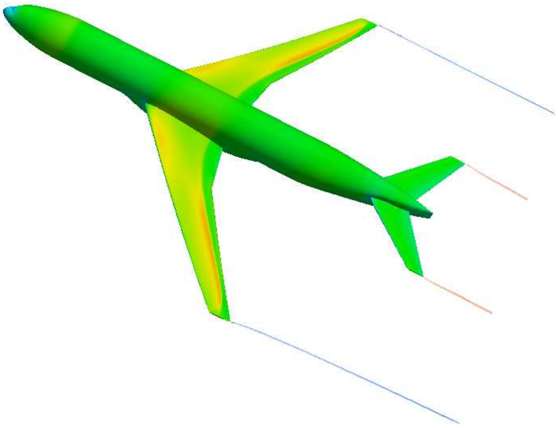
TOWARDS A HYBRID CFD PLATFORM FOR INVESTIGATING AIRCRAFT TRAILING VORTICES



Donovan M. Changfoot
Supervisor: Prof. A.G. Malan
Co-supervisor: Prof. J. Nordström

Dissertation submitted to the University of Cape Town in full fulfilment of the requirements for the degree of Masters of Science in Engineering.

August 2017

Ur e Town

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

This work is dedicated to my parents, Bertram and Muriel.

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed by candidate

Signature Removed

Signature: .... Date: August 31, 2017

Abstract

This dissertation outlines the development of a parallel 3D hybrid finite-volume-finite-difference solver. As motivation for such a scheme, the specific application area under consideration is modeling the trailing vortices shed from the wings of aircraft under transonic flight conditions. For this purpose, the *Elemental*® finite volume code is employed in the vicinity of the aircraft, while the *Essense* finite difference software is employed to accurately resolve the trailing vortices. The former method is spatially formally 2^{nd} order and the latter set to 6^{th} order accurate. The coupling of the two methods is achieved in a stable manner through the use of Summation-by-Parts operators and weak imposition of boundary conditions through Simultaneous-Approximation-Terms (SBP-SAT). Accordingly, a special parallel SBP-SAT interface library is developed in *Elemental*® . In addition, the code is extended to impose boundary conditions in a weak manner via the SBP-SAT framework; as well as interface volume definitions changed to allow coupling with the 6^{th} order code. The developed hybrid solver is successfully validated against analytical test-cases. This is followed by demonstrating its ability to model the flow field, including trailing vortex structures, around the NASA Common-Research-Model (CRM) under transonic flow conditions. Inviscid flow was assumed and the trailing vortices from both wing and horizontal stabiliser accurately resolved to 3 and 1 reference chords downstream of the lifting surface respectively. The robustness of the interface treatment is demonstrated by the smoothness of the flow solution across an interface boundary in the presence of high flow gradients and rapidly changing mesh topology. In addition, high vortex axial flow gradients were predicted while the vortex core speed is 6 % slower than free-stream.

Acknowledgements

I would like to thank Prof. A. G. Malan and Prof. J. Nordström for their guidance throughout the course of the project. Further thanks is given to my colleagues of the inCFD group for their support with *Elemental*® and Marco Kuppiainen for his support with *Essense* .

Many thanks for the financial assistance of the National Aerospace Centre (NAC) of South Africa; the AEROGUST project (funded by the European Commission under grant agreement number 636053); and the South African Research (SARChI) Chair in Industrial CFD. SARChI is funded by the Department of Science and Technology (DST) and the National Research Foundation (NRF). Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

The computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team as well as the Triolith computing cluster from the National Supercomputer Centre at Linköping University.

Lastly, a special thanks to Space Camp, my friends and parents for their undying support and encouragement that got me through the ups and downs of this journey.

Contents

The symbols listed in the nomenclature are non-dimensional unless otherwise stated. The repeated use of indices implies Einstein summation convention.

Nomenclature

Greek Symbols

α	Constant for advection diffusion equation boundary condition
β	Constant for advection diffusion equation boundary condition
ϵ	Error in functional between two mesh refinement levels
γ	Ratio of specific heats
Λ	Diagonal matrix of eigen-values
λ	Second viscosity coefficient
μ	Dynamic viscosity coefficient
ω	Vorticity vector
ϕ	Smooth scalar function
Ψ	van Albada flux limiter
ρ	Density
σ	SAT penalty parameters
τ	Stress tensor
Υ_{lm}	Edge connecting nodes l and m
ε	Non-dimensional scaling for the viscous flux vector
ξ	Curvilinear coordinate
k	Vortex strength co-efficient

Mathematical Operators

δ_{ij}	Kronecker delta function

$\otimes$	Kronecker product
π	Pi

Roman Symbols

$\mathbf{C}_{lm}$	Edge coefficient vector for edge connecting nodes l and m
$\mathbf{F}$	Vector of fluxes
$\mathbf{F}_n$	Normal flux vector
$\mathbf{g}$	Vector of boundary data
$\mathbf{n}$	Outward pointing normal
$\mathbf{Q}$	Vector of characteristic variables
$\mathbf{t}_{lm}$	Edge tangent vector pointing from node l to node m
$\mathbf{U}^*$	Vector of HLLC state variables
$\mathbf{U}$	Vector of conservative variables
$\mathcal{A}$	Surface Area
$\mathcal{V}$	Volume
$\overline{\mathbf{F}}$	Edge face-flux
$\tilde{R}$	Dimensional specific gas constant
a	Speed of sound
A_n	Normal flux jacobian
C_p	Dimensional constant pressure specific heat co-efficient
C_v	Dimensional constant volume specific heat co-efficient
E	Total specific energy
I	Identity matrix
p	Pressure
Pr	Prandtl number
q	Heat flux

$q_{L/R}$	Normal velocity used to calculate the HLLC flux and the subscripts L/R refer to the left and right states
r	Ratio used in van Albada flux limiter
R_{Roe}	Roe average ratio of left and right state densities
Re	Reynolds number
S_M	Middle wave speed
$S_{L/R}$	Left and right wave speeds
T	Temperature
u	Velocity
X	Matrix of right eigen-vectors
x	Cartesian coordinate
x_0, y_0	Vortex centre start coordinate
AR	Aspect Ratio
B	Boundary operator matrix $= diag[-1, 0...0, 1]$
c	Chord length
D	Matrix derivative operator
H	Specific Enthalpy
h	Finite difference mesh spacing
L	Left face state
M	Mach number
P	A positive definite matrix.
p	Order of accuracy
Q	Discretisation matrix
R	Right face state
S	Reference area of the NASA CRM
t	Time

Superscripts

$\hat{}$	Quantity in curvilinear space
$\tilde{}$	Dimensional value
$Inter$	Interface boundary component
$Wall$	Physical wall component
I	Inviscid component
T	Transpose
V	Viscous component

Subscripts

∞	Free-stream value
ana	Analytical solution
num	Numerical solution
i,j,k	Indices used to indicate coordinate direction or position in vector
ref	Reference value for the NASA CRM
Roe	Roe averaged variable

Abbreviations

AoA	Angle of Attack
CFD	Computational Fluid Dynamics
CRM	Common-Research-Model
DG	Discontinuous Galerkin
FDM	Finite Difference Method
FVM	Finite Volume Method
GCI	Grid Convergence Index
SAT	Simultaneous-Approximation-Terms
SBP	Summation-by-Parts
SymPy	Python Symbolic Mathematics Package

List of Figures

List of Tables

List of Code

Chapter 1

Introduction

1.1 Background

The airfoil sections of an aircraft's wing produce a pressure difference between the top and bottom surface. This pressure differential is the mechanism through which a wing produces lift. In the case of finite length wings, this drives air to wrap around the wing tip resulting in strong vortical motion. The so called trailing vortices may then persist for hundreds of wingspans downstream before decaying. As a result, they pose significant danger to trailing aircraft during in-trail wake encounters [1]. Should a trailing aircraft fly parallel to the axis of the vortex, then the swirl velocities induce a rolling moment on it. In the event that this moment cannot be countered through trimming, the handling of the trailing aircraft may become unstable. This has given rise to numerous investigations with the aim of quantifying the factors that influence the structure and persistence of the vortices. In particular, the advancements in computing power and Computational Fluid Dynamics (CFD) technology show promise in providing high fidelity solutions of large portions of an aircraft wake. This has practical significance for the commercial air transport industry which is expected to grow by 6% yearly [2, 3], for years 2015-2035.

Early analysis of the trailing vortices show that there are four parts of wake behaviour that need to be understood: vortex formation from flow leaving the trailing edge, vortex interaction, vortex decay mechanisms and atmospheric effects on the vortices [4]. The limited computing power up to the latter part of the 20^{th} century favoured the development of analytical methods to quantify these phenomena. The roll up of the trailing edge flow into the vortices is generally agreed to happen rapidly and the effect of viscosity to be small [4, 5, 6, 7]. This consensus stems from how well the

Betz method [8] predicts the structure of the rolled up vortices through the application of conservation principles for an inviscid fluid. Keeping with an inviscid fluid, Crow [9] used vortex filaments to show that a pair of vortices will develop instabilities in the form of long wave sinusoidal oscillations that amplify until the vortices touch each other. The touching, or pinching, of the vortices causes the formation of vortex rings which greatly reduces their strength. This analysis aligns with the stochastic collapse theory of vortex decay which suggests that the strength of the vortices remains fairly constant until a disturbance is introduced that deforms their path [5]. Contrary to this view is the predictable decay theory which suggests that the kinetic energy of the vortices is gradually decreased due to dissipation mechanisms such as viscous effects [5]. Greene's decay model [10] is an example of this: the vortex strength "spills" into the free stream due to viscous shear effects.

The aforementioned analytical methods provide insight into the general behaviour of the vortices. However, they typically make use of assumptions to simplify the analysis *e.g.* the load distribution of the lifting body in question. This confines the analysis to specific geometries if data collected from flight tests or numerical approximations is not provided as input. The route of numerical approximations has been favoured over flight tests due to the improvements in computational power and the expense of performing flight tests. Lower fidelity numerical approximations based on potential flow theory, *viz.* Vortex Filament and Vortex Lattice Methods [11] (VFM and VLM respectively), have been used to model vortex interaction and formation flight configurations [12, 13]. These methods capture the general trends predicted by the analytical models such as the Crow instability [14]. However, they over-estimate the induced drag savings and the lift and moment coefficients for trailing wings in a formation flight configuration [15, 6]. Rossow [6] suggests the over-estimate stems from the underlying assumption of irrotational flow.

The use of CFD removes these assumptions by solving the full Navier-Stokes equations but at significantly greater computational expense than the lower fidelity methods. The development of massively parallel codes used on high performance computing clusters combats this problem. As a result, there have been successful 3D CFD calculations of realistic commercial transport aircraft ranging from the subsonic to transonic flow regimes [16, 17]. An example of this is the study by Ning and Kroo [17] as it analyses compressibility effects on the NASA Common-Research-Model (CRM), a transonic commercial transport model, in formation flight. The study considered two CRMs flying at cruise conditions in formation separated by 20 wingspans in the stream-wise direction. The problem was broken down into three separate parts: resolving the flow around the lead aircraft; propagating the vortex shed from the lead aircraft downstream to the trailing aircraft via an analytical method and resolving the flow around the trailing aircraft. The flow around the lead aircraft was solved using AERO, NASAs 3D finite volume (2^{nd} order spatially accurate) inviscid compressible flow solver. The output of this simulation was used as the input to an Augmented Betz method [17] which propagated the vortex downstream by inferring the rolled up vorticity distribution in the wake at the beginning of the trailing aircraft's domain. This distribution was then used as the inflow boundary condition for the trailing aircraft's domain where AERO was used again to solve for the flow.

The above methodology, by using the Augmented Betz method, avoids the use of the 2^{nd} order CFD scheme to propagate the vortex over large distances. In doing so, two important considerations were tackled *viz.* accurate propagation of the vortex over large distances and dramatically reducing computational expense. The high levels of numerical dissipation resulting from 2^{nd} order CFD schemes will distort the vortex [18]. Mesh refinement is therefore necessary along the entire path of the vortex. If this were done over 20 wingspans, the computational cost would be raised to unacceptable levels.

The shortcoming of the above use of the Augmented Betz method is however that it reduces the vortex modelling to a 2D problem by neglecting the variations in axial velocity. This limits the applicability of the method. In a multiple vortex wake, where there may be movement of the vortices, due to vortex interaction, the axial velocity may need to be accounted for. This is because the variations in axial velocity plays a role in the stability of the trailing vortices [19]. Similarly for a transient simulation in which an atmospheric gust causes the onset of Crow instability [20], the subsequent sinusoidal motion of the vortices produces velocity components in all three coordinate directions.

An alternative to the cited work is to perform a full 3D simulation over the entire flow domain. As already mentioned, the exclusive use of 2^{nd} order schemes will be prohibitively expensive. This problem may be alleviated through the use of higher order spatial discretisation [21]. In numerical experiments performed by Wake and Choi [22], the use of a 5^{th} order scheme increased the time per iteration by 10% over a 2^{nd} order scheme and it took 1.5 times more iterations to reach the convergence criteria for a given mesh. However, the 5^{th} order scheme achieved a factor of 10 reduction in the number of mesh points over a 3^{rd} order scheme whilst still accurately preserving the vortex structure. This leads to an estimated order of magnitude reduction in the required CPU hours to perform the simulation.

The use of the Finite Difference Method (FDM) mentioned above necessitates the use of a structured mesh. That is to say that the connectivity between the nodal points are to be 6 (in 3D) at interior vertices. An example of this is a Cartesian grid. The generation of a single structured block around complex geometries, such as the CRM, is a daunting task. Two methods are used to overcome this *viz.* chimera and multi-block meshes. In the former, the domain and geometry are meshed with overlapping structured grids. It is then necessary to interpolate information between them. This process in itself is non-trivial and conservation as well as stability issues arise. In multi-block methods, the domain is broken up into numerous smaller structured blocks that do not overlap. Instead, information is passed between the blocks through stable and conservative interface conditions (see Chapter 3). Despite the use of either of these methods, the generation of structured meshes around complex geometries still remains a challenge.

Unstructured mesh methods such as the Finite Volume Method (FVM) and Discontinuous Galerkin (DG) method reduce the complexity of creating a single mesh around a complex geometry. However, FVM schemes widely in use are only 2^{nd} order accurate. DG methods, on the other hand, may be of arbitrary order although there is little evidence to show that this outweighs the increased cost of complexity if used as 6^{th} order across the entire domain. Ultimately, the choice between the two remains a contentious issue.

The strengths of both the FDM and the chosen unstructured mesh method may however be leveraged by creating a hybrid solver. In doing so, the unstructured method may be used in the vicinity of the CRM thereby ensuring rapid meshing. The development and propagation of the trailing vortices may then be done using higher order FDM. The structured mesh provides greater cell and node efficiency than the unstructured version whilst the high order of accuracy discretisation minimises the numerical diffusion. The coupling of the two methods is achievable through the use of Summation-by-Parts (SBP) operators and weak imposition of boundary conditions through Simultaneous-Approximation-Terms (SAT) [23]. Note that the interface procedures that arise from this have been used to develop

2D hybrid solvers for the advection-diffusion equation as well as the time dependent compressible Navier-Stokes equations [24, 25, 26, 27, 28, 29]. This work is the first to apply these concepts to capturing the vortices under transonic conditions using 3D hybrid meshes.

1.2 Project Objective

The focus of this work is to develop a 3D hybrid finite-volume-finite-difference solver that is capable of investigating the trailing vortices of an aircraft. The *Elemental*® finite volume and 6^{th} order *Essense* finite difference codes will be used for this purpose. The development begins by modifying the application of boundary conditions in *Elemental*® so that it is done using SAT conditions. This is followed by implementing the interface procedures in the form of a coupling library that facilitates interoperability between the two coding languages and for distributed memory parallel architectures. It is at this stage that the implementation of the SBP-SAT framework in *Elemental*® as well as the coupling library is tested. This is achieved using two analytical test cases *viz.* the propagating isentropic vortex and a manufactured solution for the compressible Navier-Stokes equations. The ability of the developed hybrid scheme to propagate trailing vortices is then assessed via application to the CRM under transonic cruise conditions. Both wing and horizontal stabiliser (tail) vortices are propagated over a distance of 3 and 1 chord lengths respectively.

Chapter 2

Problem Formulation

The aim of this project is to develop a hybrid finite-volume-finite-difference solver capable of investigating aircraft trailing vortices that are developed from aircraft at transonic flow conditions. This chapter details the governing equations resolved for this purpose as well as the relevant physical boundary conditions.

2.1 Governing Equations

The non-dimensionalised Navier-Stokes equations are employed to describe the flow physics. Non-dimensional variables are related to their dimensional counterparts, denoted by $\tilde{\bullet}$, via the relations

$$\rho = \frac{\tilde{\rho}}{\tilde{\rho}_\infty} \quad p = \frac{\tilde{p}}{\tilde{\rho}_\infty \tilde{a}_\infty^2} \quad T = \frac{\tilde{T}}{\tilde{T}_\infty} \quad E = \frac{\tilde{E}}{\tilde{a}_\infty^2} \quad u_i = \frac{\tilde{u}_i}{\tilde{a}_\infty} \quad \mu = \frac{\tilde{\mu}}{\tilde{\mu}_\infty} \tag{2.1}$$

where ∞ denotes free-stream conditions, a is the speed of sound, ρ the density, p the pressure, T the temperature, E the total specific energy, u_i the Cartesian velocity components (u_1, u_2, u_3) and μ the dynamic viscosity coefficient. The spatial and temporal domains are non-dimensionalised according to

$$x_i = \frac{\tilde{x}_i}{\tilde{c}_{ref}} \quad t = \frac{\tilde{t}\tilde{a}_\infty}{\tilde{c}_{ref}} \tag{2.2}$$

where x_i refers to the Cartesian coordinate directions, t is time and c_{ref} is the airfoil reference chord length. By collecting the conservative variables into the vector $\mathbf{U} = (\rho, \rho u_1, \rho u_2, \rho u_3, \rho E)^T$, the non-dimensional Navier-Stokes equations follow in conservative form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = 0 \tag{2.3}$$

Using the Mach number, M, and Reynolds number, Re, to define the constant ε =M/Re, the flux vector $\mathbf{F}_i$ follows as

$$\mathbf{F}_i = \mathbf{F}_i^I - \varepsilon \mathbf{F}_i^V \tag{2.4}$$

where the inviscid flux vector $\mathbf{F}_i^I$ and the viscous flux vector $\mathbf{F}_i^V$ are given by

$$\mathbf{F}_i^I = \begin{pmatrix} \rho u_i \\ \rho u_1 u_i + p\delta_{1i} \\ \rho u_2 u_i + p\delta_{2i} \\ \rho u_3 u_i + p\delta_{3i} \\ u_i(\rho E + p) \end{pmatrix} \quad \mathbf{F}_i^V = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ u_j \tau_{ij} + q_i \end{pmatrix} \tag{2.5}$$

In the above, the shear tensor for a Newtonian fluid, τ, and heat flux, q_i, are defined as

$$\tau_{ij} = \lambda \frac{\partial u_k}{\partial x_k}\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad , \quad i,j,k = 1,2,3 \tag{2.6}$$

$$q_i = \frac{1}{Pr(\gamma - 1)} \frac{\partial T}{\partial x_i} \tag{2.7}$$

where the second viscosity coefficient is $\lambda = \frac{-2}{3}\mu$, $\gamma = \tilde{C}_p/\tilde{C}_v$ the constant ratio of specific heats and $\tilde{C}_p, \tilde{C}_v$ are specific heats at constant pressure and volume respectively. It is noted that in this study the gas is assumed to be calorically perfect, the viscosity coefficient to be constant non-dimensional $\mu = 1$ and Pr=0.72.

2.2 Constitutive Equations

The Ideal Gas Law is used to close the system of equations as it relates the temperature, density and pressure in dimensional form as

$$\tilde{p} = \tilde{\rho}\tilde{R}\tilde{T} \tag{2.8}$$

where $\tilde{R} = \tilde{C}_p - \tilde{C}_v$ is the specific gas constant. Using isentropic relations, the dimensional speed of sound is written as

$$\tilde{a} = \sqrt{\gamma \tilde{R}\tilde{T}} \tag{2.9}$$

Using Equation (2.1) and Equation (2.9), the non-dimensional Ideal Gas Law follows

$$\rho T = \gamma p \tag{2.10}$$

The non-dimensional equation of state and speed of sound are then calculated as

$$p = (\gamma - 1)\rho(E - \frac{1}{2}u_i u_i) \quad a = \sqrt{T} \tag{2.11}$$

Using the relations in Equation (2.1) the far-field values of the following variables are found to be

$$\rho_\infty = 1 \quad p_\infty = \frac{1}{\gamma} \quad T_\infty = 1 \quad |u_\infty| = M \quad E_\infty = \frac{1}{\gamma(\gamma-1)} + \frac{1}{2}M^2 \tag{2.12}$$

The values in Equation (2.12) are used to initialise the flow domain.

2.3 Curvilinear Grid Transformation

In the context of the finite difference code, *Essense* , the physical domain (x_1, x_2, x_3) is mapped to the computational domain (ξ_1, ξ_2, ξ_3) through the use of a curvilinear grid transformation. To this end, the coordinate transformation $x_i = x_i(\xi_1, \xi_2, \xi_3)$ is introduced such that $0 \leq \xi_i \leq 1$. This results in the Jacobian being

$$\mathbf{J}_{ij} = \frac{\partial x_i}{\partial \xi_j} \tag{2.13}$$

Defining $J = det(\mathbf{J})$ and applying the transformation to the Navier-Stokes equations results in

$$(J\mathbf{U})_t + \frac{\partial \hat{\mathbf{F}}_i}{\partial \xi_i} = 0 \tag{2.14}$$

$$\hat{\mathbf{F}}_i = J\frac{\partial \xi_i}{\partial x_j}\left(\mathbf{F}_j^I - \varepsilon \mathbf{F}_j^V\right) \tag{2.15}$$

where the symbol $\hat{\bullet}$ represents a transformed quantity and $()_t$ denotes derivative with respect to time.

2.4 Physical Boundary Conditions

The governing equations alone are not enough to provide a unique flow solution. Therefore providing the physical boundary conditions is required. Three boundary types were used in this study *viz.* adiabatic no-slip, symmetry plane and characteristic inflow/outflow. The first is applicable to solid walls such as the surface of an aircraft. The no-slip part of the condition implies that there is to be no velocity at the surface and the adiabatic part implies that there is to be no heat transfer normal to the surface. This is expressed mathematically in continuum form as

$$\mathbf{u} = 0 \quad \frac{\partial T}{\partial \mathbf{n}} = 0 \tag{2.16}$$

where $\mathbf{n}$ is the outward pointing normal.

The second condition, symmetry plane, is used to reduce the size of the domain and therefore computational cost. It is similar to the adiabatic no-slip condition except that only the normal component of velocity is removed which is expressed in continuum form as

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \frac{\partial T}{\partial \mathbf{n}} = 0 \tag{2.17}$$

where the nomenclature is as previously defined.

The particular form of the characteristic inflow/outflow condition used in this study is developed in [30] and complements the implementation of the discrete SAT framework (detailed in Chapter 3). To facilitate the illustration of these conditions, the flux Jacobian A_n is introduced as

$$A_n = \frac{\partial \mathbf{F}_n^I}{\partial \mathbf{U}} \tag{2.18}$$

where the inviscid normal flux vector $\mathbf{F}_n^I = \mathbf{F}_i^I \mathbf{n}_i$. This is then rotated into characteristic space through the use of the transformation

$$\Lambda = X^{-1} A_n X \quad \mathbf{Q} = X^{-1} \mathbf{U} \tag{2.19}$$

where Λ is the diagonal matrix containing the eigen-values

$$\Lambda = \text{diag}[u_n, u_n, u_n, u_n - a, u_n + a],$$

X is the matrix with the corresponding right eigen-vectors and $\mathbf{Q}$ is the vector of characteristic variables. The incoming and outgoing characteristics are identified by splitting Λ into two matrices

$$\Lambda = \Lambda^+ + \Lambda^- \tag{2.20}$$

where Λ^+ contains only the positive eigen-values and Λ^- only the negative. In this way, the incoming characteristics are identified as they correspond to

Λ^-. The incoming characteristics are prescribed at the far-field boundaries using the condition

$$(\Lambda^- + \Lambda^{'})\mathbf{Q} - \varepsilon X^{-1}\mathbf{F}_n^V = \mathbf{g} \tag{2.21}$$

where $\Lambda^{'}$ is a matrix with $\Lambda^{'}_{4,5} = -(u_n - a)$ and the rest of the entries are 0. Similar to $\mathbf{F}_n^I$, $\mathbf{F}_n^V$ is the normal viscous flux vector. Lastly, the vector $\mathbf{g}$ contains the resulting boundary condition to be applied.

2.5 Interface Boundary Conditions

As mentioned in Section 1, the FVM is to be used in the vicinity of the aircraft and the FDM downstream of it. This results in two numerical domains that are joined together at an artificial boundary: the finite volume domain is on the one side with flow variables of $\mathbf{U}$ and the finite difference on the other with flow variables of $\mathbf{V}$. Interface conditions are required to accurately couple the two methods at this boundary in a stable manner. There are two conditions that are to be enforced.

$$\begin{aligned} \text{Condition 1}: \quad & \mathbf{U}_{interface} = \mathbf{V}_{interface} \\ \text{Condition 2}: \quad & \mathbf{F}_i^V(\mathbf{U})_{interface} = \mathbf{F}_i^V(\mathbf{V})_{interface} \end{aligned} \tag{2.22}$$

These conditions enforce continuity in the flow variables and viscous flux at the interface.

Chapter 3

Numerical Discretisation

The discretisation methods that will be described briefly in this chapter are the FDM, as employed by *Essense* , and the FVM as employed by *Elemental*® . Common to both of these methods is the manner in which the boundary conditions are applied *viz.* the use of SAT. This scheme allows for natural interface conditions between the two methods which satisfy certain conservation and stability conditions. These as well as the temporal discretisation are discussed next.

3.1 *Essense*

Essense is built upon what is commonly referred to as the SBP-SAT framework. In short this refers to the spatial discretisation operators used and the manner in which boundary conditions are enforced. The spatial operators are so named as they represent the discrete analogue of integration-by-parts. The use of these operators in conjunction with weak imposition of boundary conditions (through the use of SAT) have been proven to be time stable through the use of energy estimates on finite domains (see [31, 23, 32]). This section details the 2^{nd}, 4^{th} and 6^{th} order accurate SBP operators used in *Essense* .

3.1.1 1st Derivative

The SBP property of the 1^{st} derivative operator is easily demonstrated using a 1-D finite domain $x \in [x_l, x_r]$ on which the smooth scalar function $\phi(x,t)$ is defined. The value of this function is held at discrete points as shown in Figure 3.1 and stored in the vector $\boldsymbol{\phi} = [\phi_0, \phi_1, ..., \phi_n]^T$.

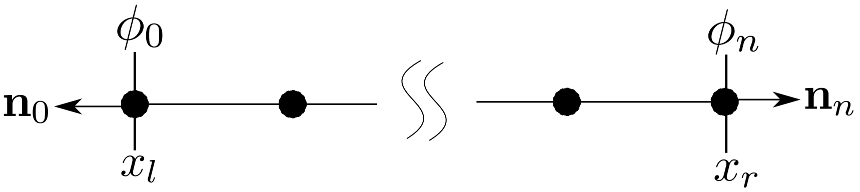


Figure 3.1: Schematic of the continuous solution ϕ that has been discretised as $\boldsymbol{\phi}$ on a finite 1-D domain.

With reference to Figure 3.1, $\mathbf{n}$ is the outward pointing unit normal and for the 1-D problem $|\mathbf{n}| = 1$. The approximation of the 1^{st} spatial derivative ϕ_x at each point is stored in the vector $\boldsymbol{\phi_x}$ and is calculated through the use of the 1^{st} derivative operator D_x

$$\boldsymbol{\phi_x} = D_x\boldsymbol{\phi} = P^{-1}Q_x\boldsymbol{\phi} \tag{3.1}$$

where P is a positive symmetric definite matrix, Q_x a skew symmetric matrix operator with the property

$$Q_x + (Q_x)^T = B = diag[-1, 0...0, 1] \tag{3.2}$$

It is seen from Figure 3.1 that the operator B is a matrix that defines the values in $\boldsymbol{\phi}$ that are on the boundary according to the outward pointing normal $\mathbf{n}$. Multiplying Equation (3.1) by $\boldsymbol{\phi}^T P$ and using Equation (3.2) results in

$$\begin{aligned}\boldsymbol{\phi}^T P\left(P^{-1}Q_x\right)\boldsymbol{\phi} &= \boldsymbol{\phi}^T B\boldsymbol{\phi} - \boldsymbol{\phi}^T\left(Q_x\right)^T\boldsymbol{\phi}\\ &= \boldsymbol{\phi}^T B\boldsymbol{\phi} - \boldsymbol{\phi}^T\left(Q_x\right)^T P^{-1}P\boldsymbol{\phi}\\ &= \boldsymbol{\phi}^T B\boldsymbol{\phi} - \boldsymbol{\phi}_x^T P\boldsymbol{\phi}\end{aligned} \tag{3.3}$$

The analogy to integration-by-parts of Equation (3.3) can be seen by considering the energy estimate (multiply by the solution and integrate over the domain) of the continuous function ϕ

$$\int_{x_l}^{x^r} \phi\phi_x \; dx = \phi\phi|_{x_l}^{x_r} - \int_{x_l}^{x^r} \phi_x\phi \; dx \tag{3.4}$$

Through comparing Equation (3.3) and Equation (3.4), it is seen that the matrix P forms a weighted norm. This is referred to as a P-norm and is convenient for assessing the orders of convergence presented in Chapter 5.

3.1.2 2nd Derivative

The description of the 2^{nd} derivative will use the same domain x and scalar function $\phi(x,t)$ as described above. One manner in which the second derivative operator may be achieved is to apply the first derivative operator, D_x, twice.

$$\begin{aligned} \boldsymbol{\phi_{xx}} &= D_x D_x \boldsymbol{\phi} \\ &= \left(P^{-1}Q_x\right)\left(P^{-1}Q_x\right)\boldsymbol{\phi} \end{aligned} \tag{3.5}$$

Multiplying Equation (3.5) by $\boldsymbol{\phi}^T P$ and applying Equation (3.2) results in

$$\begin{aligned} \boldsymbol{\phi}^T P D_{xx}\boldsymbol{\phi} &= \boldsymbol{\phi} P \left(P^{-1}Q_x\right)\left(P^{-1}Q_x\right)\boldsymbol{\phi} \\ &= \boldsymbol{\phi}\left(B - (Q_x)^T\right)\left(P^{-1}Q_x\right)\boldsymbol{\phi} \\ &= \boldsymbol{\phi}^T B \boldsymbol{\phi_x} - \left(\boldsymbol{\phi_x}\right)^T P \boldsymbol{\phi_x} \end{aligned} \tag{3.6}$$

Again, the analogy to integration-by-parts may be seen as

$$\int_{x_l}^{x_r} \phi\phi_x \; dx = \phi\phi_x|_{x_l}^{x_r} - \int_{x_l}^{x_r} \phi_x\phi_x \; dx \tag{3.7}$$

3.1.3 Higher Order

An attractive feature of finite difference operators is that they can be constructed to have an arbitrary order of accuracy. This is achieved through the use of weighted Taylor expansions at discrete points in the temporal or spatial domain. However, this does not guarantee that the resultant operator has the SBP properties detailed above. Modifications, which are briefly detailed in [23], are therefore made to the stencil near boundaries.

The modifications result in a lower order of accuracy near boundaries as compared to the internal part of the domain. For the operators used in *Essense* , this results in the boundaries having an order of accuracy (normal to the boundary) of p and the internal of the domain having 2p. Together this results in a global order of accuracy of p+1. This is illustrated with the standard 2^{nd} order central differencing operator for the first derivative. This operator, described in Equation (3.8), is known to have 1^{st} order boundary accuracy as well as 2^{nd} order internal and global accuracy.

$$Q = \frac{1}{2}\begin{bmatrix} -1 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & -1 & 0 & 1 \\ & & & & -1 & 1 \end{bmatrix}, \quad P = h\begin{bmatrix} \frac{1}{2} & & & & & \\ & 1 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & 1 & \\ & & & & & \frac{1}{2} \end{bmatrix} \tag{3.8}$$

where h is the mesh spacing and P is a diagonal matrix. The SBP property does not require for P to be diagonal and increased performance in terms of accuracy may be achieved by using so-called block-norms [23]. However, the use of curvilinear grids by *Essense* means that stability is only guaranteed for diagonal P norms.

Standard P norms exist for operators with an internal accuracy of 2^{nd}, 4^{th} and 6^{th} order. The form of these operators has implications for interfacing *Essense* and *Elemental*® (see Chapter 4) and are shown below.

$$\begin{aligned} P_{2nd} &= h \cdot diag\left(\frac{1}{2}, 1, 1...,\right) \\ P_{4th} &= h \cdot diag\left(\frac{17}{48}, \frac{59}{48}, \frac{43}{48}, \frac{49}{48}, 1, 1, ...,\right) \\ P_{6th} &= h \cdot diag\left(\frac{13649}{43200}, \frac{12013}{8640}, \frac{2711}{4320}, \frac{5359}{4320}, \frac{7877}{8640}, \frac{43801}{43200}, 1, 1, ...\right) \end{aligned} \tag{3.9}$$

where the subscript indicates the internal order of accuracy.

3.1.4 Extension to Navier-Stokes

Following the notation described in [30], the governing equation is discretised on an $n_{\xi_1} \times n_{\xi_2} \times n_{\xi_3}$ curvilinear grid as $\mathbf{U} = (U^{1111}, U^{2111}, ..., U^{n_{\xi_1}, n_{\xi_2}, n_{\xi_3}, 5})$. The superscripts represent the position in the ξ_1, ξ_2, ξ_3 direction followed by the variable. The 1-D spatial derivative operator, D, may be extended to the Navier-Stokes equation through the use of the Kronecker product $\otimes$

$$\begin{aligned} D_{\xi_1} &= (I_{\xi_3} \otimes I_{\xi_2} \otimes D \otimes I_5) \\ D_{\xi_2} &= (I_{\xi_3} \otimes D \otimes I_{\xi_1} \otimes I_5) \\ D_{\xi_3} &= (D \otimes I_{\xi_2} \otimes I_{\xi_1} \otimes I_5) \end{aligned} \tag{3.10}$$

where the matrices I_{ξ_i} are identity matrices of size $n_{\xi_i} \times n_{\xi_i}$ and I_5 is the 5×5 identity matrix. By defining $\mathbf{JU} = (J(\xi_1^1, \xi_2^1, \xi_3^1)U^{1111}, J(\xi_1^2, \xi_2^1, \xi_3^1)U^{2111}...)$, a semi-discrete form of Equation (2.15) is written as

$$(\mathbf{JU})_t + D_{\xi_i}\left(\hat{\mathbf{F}}^{\mathbf{I}}_{\xi_\mathbf{i}} - \varepsilon \hat{\mathbf{F}}^{\mathbf{V}}_{\xi_i}\right) = 0 \tag{3.11}$$

where $\hat{\mathbf{F}}^{\mathbf{I},\mathbf{V}}_{\xi_\mathbf{i}}$ are the flux vectors with components $\left(\hat{F}^{I,V1111}_{\xi_i}, \hat{F}^{I,V2111}_{\xi_i}, ...\right)$.

3.2 *Elemental®*

Elemental® is a multi-physics code that employs the vertex-centred edge-based finite volume discretisation method. By employing edge-based data structures, computational expense and memory usage is lowered. Furthermore, the inherent modular nature allows the undertaking of large numerical simulations on high performance distributed memory computing platforms. The following sections detail the finite volume method and edge-based data structures.

3.2.1 Edge-Based-Vertex-Centred Finite Volume Method

Consider an arbitrary control volume, $\mathcal{V}$, which is bounded by the surface $\mathcal{A}$. For this application, it is assumed that $\mathcal{V}$ is stationary and does not change over time. The governing equations are then cast into weak form by integrating over $\mathcal{V}$ and the Divergence Theorem applied to yield

$$\int\limits_{\mathcal{V}} \frac{\partial \mathbf{U}}{\partial t} d\mathcal{V} + \oint\limits_{\mathcal{A}} \mathbf{F}_i \mathbf{n}_i d\mathcal{A} = 0 \tag{3.12}$$

where $\mathbf{n}$ is the outward pointing normal associated to the surface segment $d\mathcal{A}$. Since the control volume is constant with time in this work, the volume integral and temporal derivative of the first term in the above equation are swapped.

$$\frac{d}{dt}\int\limits_{\mathcal{V}} \mathbf{U} d\mathcal{V} + \oint\limits_{\mathcal{A}} \mathbf{F}_i \mathbf{n}_i d\mathcal{A} = 0 \tag{3.13}$$

In the context of a mesh, these control volumes are created around the nodes at which the flow variables are stored. The bounding surfaces for the control volumes are constructed by joining the element centroids to the edge mid-points in what is called the dual-mesh. This process, performed on the 2D mesh in Figure 3.2, results in 2 faces bisecting the edge Υ_{lm} and 2 boundary faces attached to node l.

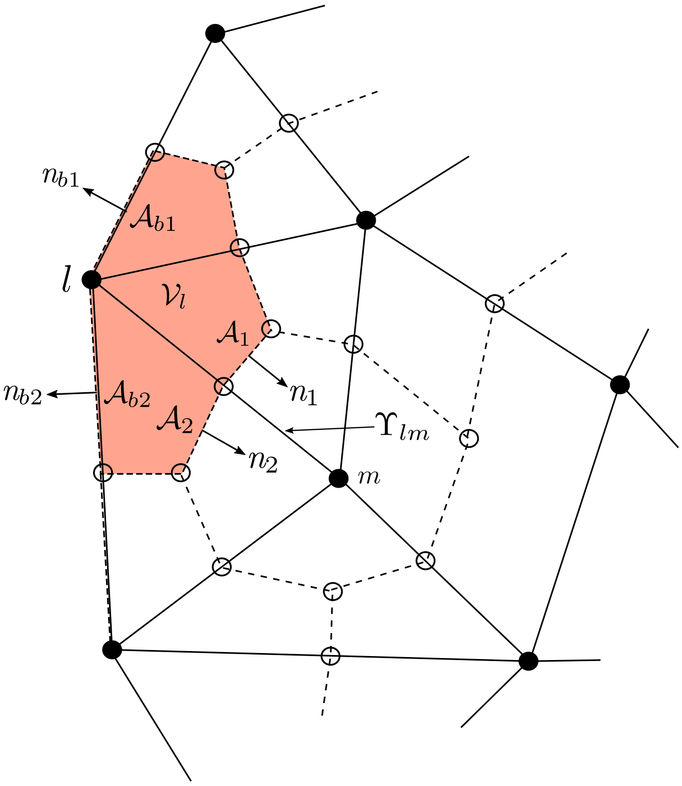


Figure 3.2: Schematic of the construction of the dual-mesh (dashed lines) from the mesh (solid lines).

Tthe flux term is discretised in an edge-wise manner by summing over the areas, $\mathcal{A}$, and outward pointing unit normals, $\mathbf{n}$, according to Equation (3.14) to produce the edge-coefficient

$$\mathbf{C}_{lm} = \mathcal{A}_1\mathbf{n}_1 + \mathcal{A}_2\mathbf{n}_2 \tag{3.14}$$

The flux term is then approximated as the sum of the flux through all the surrounding faces. Using node l as an example, the semi-discrete form of Equation (3.13) is written as

$$\frac{d\mathbf{U}}{dt}\mathcal{V}_l + \sum_{\Upsilon \in \mathcal{V}_l} \overline{\mathbf{F}}_i \mathbf{C}_i + \sum_{\mathcal{A}_b \in \mathcal{V}_l} \overline{\mathbf{F}}_i \mathbf{n}_i \mathcal{A}_b = 0 \tag{3.15}$$

It is assumed that the face-flux $\overline{\mathbf{F}}$ is constant along the face in question and is evaluated at the edge mid-point. While the nodal value of $\mathbf{F}$ is used at boundary faces $\mathcal{A}_b$, an interpolated value is used at edge faces. Since the inviscid and viscous flux possess different mathematical properties, two different methods are used to interpolate the edge-face values. In the case of the inviscid flux, *Elemental*® employs a 2^{nd} order MUSCL scheme [33, 34] with Van Albada flux limiter to interpolate left and right states of an edge-face. The HLLC upwind scheme [35, 36] is then used to evaluate the inviscid edge-face-flux. To construct the viscous flux, a compact stencil [37] is used to evaluate the viscous stresses at the edge-face.

3.2.2 Inviscid Edge-Face-Flux

A shock structure may lead to discontinuities in the flow variables. This may be captured by interpolating a left and right state of an edge-face using the MUSCL scheme with the 2^{nd} order Van Albada flux limiter. Using Υ_{lm} as an example, the edge tangent vector is defined

$$\boldsymbol{t}_{lm} = \boldsymbol{x}_m - \boldsymbol{x}_l \tag{3.16}$$

The left (L) and right (R) states for the flow variable $\mathbf{U}_i$ are then calculated according to

$$\begin{aligned} \mathbf{U}_{L:i} &= \mathbf{U}_{l:i} + \frac{1}{2}\Psi(r^L_{lm:i})\left(2\left(\frac{\partial \mathbf{U}_{l:i}}{\partial x_j}\right)\mathbf{t}_{lm:j} - \Delta\right) \\ \mathbf{U}_{R:i} &= \mathbf{U}_{m:i} - \frac{1}{2}\Psi(r^R_{lm:i})\left(2\left(\frac{\partial \mathbf{U}_{m:i}}{\partial x_j}\right)\mathbf{t}_{lm:j} - \Delta\right) \end{aligned} \tag{3.17}$$

where Ψ is the van Albada flux limiter

$$\Psi(r) = \frac{r(r+1)}{r^2+1} \tag{3.18}$$

and

$$\begin{aligned} \Delta &= \mathbf{U}_{m:i} - \mathbf{U}_{l:i} \\ r^L_{lm:i} &= \frac{1}{\Delta}\left(2\left(\frac{\partial \mathbf{U}_{l:i}}{\partial x_j}\right)\mathbf{t}_{lm:j} - \Delta\right) \\ r^R_{lm:i} &= \frac{1}{\Delta}\left(2\left(\frac{\partial \mathbf{U}_{m:i}}{\partial x_j}\right)\mathbf{t}_{lm:j} - \Delta\right) \end{aligned} \tag{3.19}$$

The HLLC flux [36] is then constructed from the states according to

$$\overline{\mathbf{F}}^{\mathbf{I}}_{lm} = \begin{cases} \mathbf{F}^{\mathbf{I}}(\mathbf{U_L}) & \text{if } S_L > 0 \\ \mathbf{F}^{\mathbf{I}}(\mathbf{U^*_L}) & \text{if } S_L \leq 0 < S_M \\ \mathbf{F}^{\mathbf{I}}(\mathbf{U^*_R}) & \text{if } S_M \leq 0 \leq S_R \\ \mathbf{F}^{\mathbf{I}}(\mathbf{U_R}) & \text{if } S_R < 0 \end{cases}$$

where

$$\mathbf{F^I}(\mathbf{U^*_{L/R}})_i = \begin{pmatrix} \rho^*_{L/R} S_M \\ (\rho u_1)^*_{L/R} S_M + p^* \delta_{1i} \\ (\rho u_2)^*_{L/R} S_M + p^* \delta_{2i} \\ (\rho u_3)^*_{L/R} S_M + p^* \delta_{3i} \\ S_M \left((\rho E)^*_{L/R} + p^* \right) \end{pmatrix} \tag{3.20}$$

and

$$\begin{aligned} \mathbf{U}^*_{L/R} &= \begin{pmatrix} \rho^* \\ (\rho u_1)^* \\ (\rho u_2)^* \\ (\rho u_3)^* \\ (\rho E)^* \end{pmatrix}_{L/R} \\ &= \frac{1}{S_{L/R} - S_M} \begin{pmatrix} (S_{L/R} - q_{L/R}) \rho_{L/R} \\ (S_{L/R} - q_{L/R})(\rho u_1)_{L/R} + (p^* - p_{L/R}) \delta_{1i} \\ (S_{L/R} - q_{L/R})(\rho u_2)_{L/R} + (p^* - p_{L/R}) \delta_{2i} \\ (S_{L/R} - q_{L/R})(\rho u_3)_{L/R} + (p^* - p_{L/R}) \delta_{3i} \\ (S_{L/R} - q_{L/R})(\rho E)_{L/R} - p_{L/R} q_{L/R} + p^* S_M \end{pmatrix} \\ p^* &= p_L + \rho_L (q_L - S_L)(q_L - S_M) = p_R + \rho_R (q_R - S_R)(q_R - S_M) \end{aligned} \tag{3.21}$$

In the above $q_{L/R} = (u_i n_i)_{L/R}$ is the normal velocity and the subscripts L/R refer to the left and right states. The wave speeds $S_{L/R}$ and S_M are calculated as

$$\begin{aligned} S_M &= \frac{\rho_R q_R (S_R - q_R) - \rho_L q_L (S_L - q_L) + p_L - p_R}{\rho_R (S_R - q_R) - \rho_L (S_L - q_L)} \\ S_L &= min \left[q_L - a_L, (q_L)_{Roe} - a_{Roe} \right] \\ S_R &= max \left[q_R + a_R, (q_R)_{Roe} + a_{Roe} \right] \end{aligned} \tag{3.22}$$

The *Roe* subscript in Equation (3.22) refers to the Roe-Averaged [38] variables which are calculated as

$$\begin{aligned} R_{roe} &= \sqrt{\frac{\rho_R}{\rho_L}} \\ (u_i)_{Roe} &= \frac{R_{roe}(u_i)_R + (u_i)_L}{R+1} \\ H_{Roe} &= \frac{R_{roe}H_R + H_L}{R_{roe}+1} \\ a_{Roe} &= \sqrt{(\gamma-1)\left(H_{Roe} - \frac{1}{2}(u_i)_{Roe}(u_i)_{Roe}\right)} \end{aligned} \tag{3.23}$$

where the specific enthalpy is $H = E + \frac{p}{\rho}$.

3.2.3 Viscous Flux Discretisation

The viscous stresses at the edge-face are interpolated using a compact stencil [37]. This is done to prevent odd-even decoupling and recover the standard 2^{nd} derivative finite difference operator on structured Cartesian grids. The stencil requires that the gradients at each node is calculated a priori. Using node l as an example, this is done according to

$$\left(\frac{\partial \mathbf{U}_i}{\partial x_j}\right)_l \approx \frac{1}{\mathcal{V}_l}\left[\sum_{\Upsilon\in\mathcal{V}_l} \frac{1}{2}\left(\mathbf{U}_{l:i} + \mathbf{U}_{m:i}\right)\mathbf{C}_{lm:j} + \sum_{\mathcal{A}_b\in\mathcal{V}_l} \mathbf{U}_{l:i}\mathcal{A}_b\mathbf{n}_{b:j}\right] \tag{3.24}$$

The edge-face stress is then calculated as

$$\left(\frac{\partial \mathbf{U}_i}{\partial x_j}\right)_{lm} \approx \frac{\mathbf{U}_{m:i} - \mathbf{U}_{l:i}}{|\mathbf{t}_{lm}|}\mathbf{t}_{lm:j} + \frac{1}{2}\left(\left(\frac{\partial \mathbf{U}_i}{\partial x_j}\right)_l + \left(\frac{\partial \mathbf{U}_i}{\partial x_j}\right)_m\right)_{normal} \tag{3.25}$$

where $()_{normal}$ refers to the component of Equation (3.24) in the normal direction to the edge.

3.3 Simultaneous-Approximation-Terms

As mentioned in Chapter 2, the flow description is incomplete without suitable boundary conditions. These conditions are introduced into the numerical scheme through the use of SAT which are described next. In particular, the stability properties of the resultant numerical scheme is demonstrated. This is followed by discussing the physical and interface boundary conditions as implemented into *Elemental*® .

3.3.1 Stability

The application of boundary conditions in computational mechanics is known to be closely linked to the stability of the resulting numerical scheme. One manner in which they are applied is to strongly inject the desired values at boundary nodes. This means overwriting the numerically calculated solution value with a pre-determined value. A common example of this is the no-slip condition where the boundary node velocity is hard set to be that of the solid surface. Through energy estimates on finite domains, it is shown that this may introduce spurious numerical growth [39].

Alternatively, the boundary conditions may be applied weakly through penalty terms commonly referred to as SAT. In doing so, the numerical instabilities are combated to ensure stability in a manner that is to be defined. To this end, the 1D advection-diffusion equation is introduced to serve as a model for the Navier-Stokes equations and reduce narrative complexity. The problem is defined as

$$\begin{aligned} u_t + a u_x &= \epsilon u_{xx} && 0 \leq x \leq 1 \\ \alpha u + \epsilon u_x &= g_0 && \text{at } x = 0 \\ \beta u - \epsilon u_x &= g_1 && \text{at } x = 1 \\ u(x,0) &= f(x) \end{aligned} \tag{3.26}$$

The stability analysis performed in Appendix A reveals that the energy estimate for the continuous problem is

$$\frac{d}{dt}||u||^2 \leq -\frac{g_1^2}{-a+2\beta} - \frac{g_0^2}{a+2\alpha} \tag{3.27}$$

It is seen from Equation (3.27) that the growth of the solution is bounded by the boundary data g_0 and g_1. Equation (3.26) is discretised similarly to Figure 3.1 into the vector $\mathbf{u}$ and the boundary conditions are applied using SAT. In doing so and referring to Appendix A, the energy estimate of the discrete system is found to be:

$$\frac{d}{dt}(\mathbf{u}^T P \mathbf{u}) \leq -\frac{g_1^2}{-a+2\beta} - \frac{g_0^2}{a+2\alpha} \tag{3.28}$$

Through comparison of Equation (3.27) and Equation (3.28), it is seen that the discrete energy estimate matches that of the continuous. Therefore the numerical growth is bounded by the boundary data and the scheme is said to be stable. In the context of the Navier-Stokes equations, this stability is proven for the linearised frozen co-efficient system and numerical experiments show that they are applicable to the non-linear system [30, 40]. The form of the SAT penalty terms for the boundary conditions described in Chapter 2 as well as a description of their implementation into *Elemental*® follows next.

3.3.2 Implementation into *Elemental*®

The derivation of the SAT terms for the boundary conditions described in Section 2.4 are well documented in [30, 40, 29, 24]. There are four SAT terms that are used to apply the boundary conditions: the inviscid, viscous, wall and interface terms. The characteristic and symmetry boundary conditions are the simplest as they use only the inviscid and viscous terms. In addition to these, the interface condition uses the interface term. Finally, for implementation purposes the adiabatic no-slip condition is applied in three parts: the no-penetration ($u_n = 0$), zero velocity and zero temperature gradient conditions. It therefore employs the inviscid, viscous and wall terms. The aforementioned terms may be added to the right-hand-side of Equation (3.15) at boundary nodes as follows

$$\sum_{\mathcal{A}_b \in \mathcal{V}} \left(\sigma^I \mathbf{S}^I + \mathbf{S}^V + \sigma^{Wall} \mathbf{S}^{Wall} + \sigma^{Inter} \mathbf{S}^{Inter} \right) \mathcal{A}_b \tag{3.29}$$

where the penalty parameters σ^I, σ^{Wall} and σ^{Inter} are used to switch on or off terms according to which boundary condition is being applied. The specific values for these penalties are derived from stability considerations and are summarised in Table 3.1.

Table 3.1: Penalty parameters for the adiabatic no-slip, interface and characteristic boundary conditions.

	Characteristic	Adiabatic No-Slip	Symmetry	Interface
σ^I	1	2	2	1
σ^{Wall}	0	$\leq -\frac{1}{4}$	0	0
σ^{Inter}	0	0	0	$\leq -\frac{1}{4}$
σ	1	0	0	0

Using previously defined nomenclature, the inviscid term has the form

$$\mathbf{S}^I = X_{Roe} \left(\Lambda^-_{Roe} + \sigma \Lambda'_{Roe} \right) X^{-1}_{Roe} \left(\mathbf{U} - \mathbf{U}_{target} \right) \tag{3.30}$$

The Roe subscript indicates that the respective terms should be constructed using Roe-averaged values of $\mathbf{U}$ (left state) and the target values $\mathbf{U}_{target}$

(right state). $\mathbf{U}_{target}$ is constructed from either the nodal value $\mathbf{U}$ or user prescribed data. The former is used to apply the no-penetration condition necessary for the symmetry and adiabatic no-slip conditions (the zero velocity and temparature gradient conditions are discussed further on): $\mathbf{U}_{target}$ is a copy of $\mathbf{U}$ except the component of normal velocity is removed. It is important to note that $u_n = 0$ is used to construct X_{Roe} and Λ^{-}_{Roe}. The latter form of $\mathbf{U}_{target}$ is used in the case of the characteristic condition where the far-field values are used as well as the interface condition where the values of the neighbouring mesh's co-located nodes are used. In these cases, the use of $\sigma^I = 1$ gives the scheme an upwind characteristic in that it introduces a dissipation term into the energy estimate.

The viscous term has two forms; one for the symmetry, adiabatic no-slip and characteristic boundary conditions and the other for the interface boundary. The former has the form

$$\mathbf{S}^V = -\varepsilon \left(\mathbf{F}^V_i - \mathbf{F}^V_{i,target} \right) \mathbf{n}_i \tag{3.31}$$

Similar to the inviscid term, the target fluxes are constructed either from nodal values or prescribed data. The adiabatic no-slip and symmetry conditions use a copy of $\mathbf{F}^V$ except that $\frac{\partial T}{\partial n}$ is removed. For the characteristic boundary conditions, a zero vector is used. This is done as it is assumed that no viscous fluxes are present at the far-field. In the case of an outflow boundary, this approximation does cause small reflections into the domain as viscous fluxes may be present. Nonetheless, the flow phenomenon are still allowed to exit smoothly [30].

The wall term has the form

$$\mathbf{S}^{Wall} = \varepsilon \max\left(\frac{\gamma}{Pr}, \frac{5}{3} \right) \frac{\mu}{\rho_{Roe}} \left(\mathbf{U} - \mathbf{U}_{target} \right) \frac{\mathcal{A}_b}{\mathcal{V}} \tag{3.32}$$

As is seen in Table 3.1, this term is only used for the adiabatic no-slip condition. The target values are a copy of $\mathbf{U}$ except that the velocity contribution is removed. This term works in conjunction with the inviscid term. As the viscosity tends to 0, then so does $\mathbf{S}^{Wall}$ leaving only the Euler slip condition.

The interface term is introduced by first referring to Figure 3.3a. Two meshes are coupled together along an interface boundary. With respect to the finite volume mesh, node a is chosen such that it has the smallest ratio of $P_{FVM} = \mathcal{V}/\left(\mathcal{A}_1 + \mathcal{A}_2\right)$ from all the nodes on the interface boundary. Should the 6^{th} order finite difference operator be used, then in the case of

Figure 3.3a $P_{FDM} = \frac{13649}{43200}\Delta x$. Using the constant $Z = P_{FVM} + P_{FDM}$, the interface term then has the form

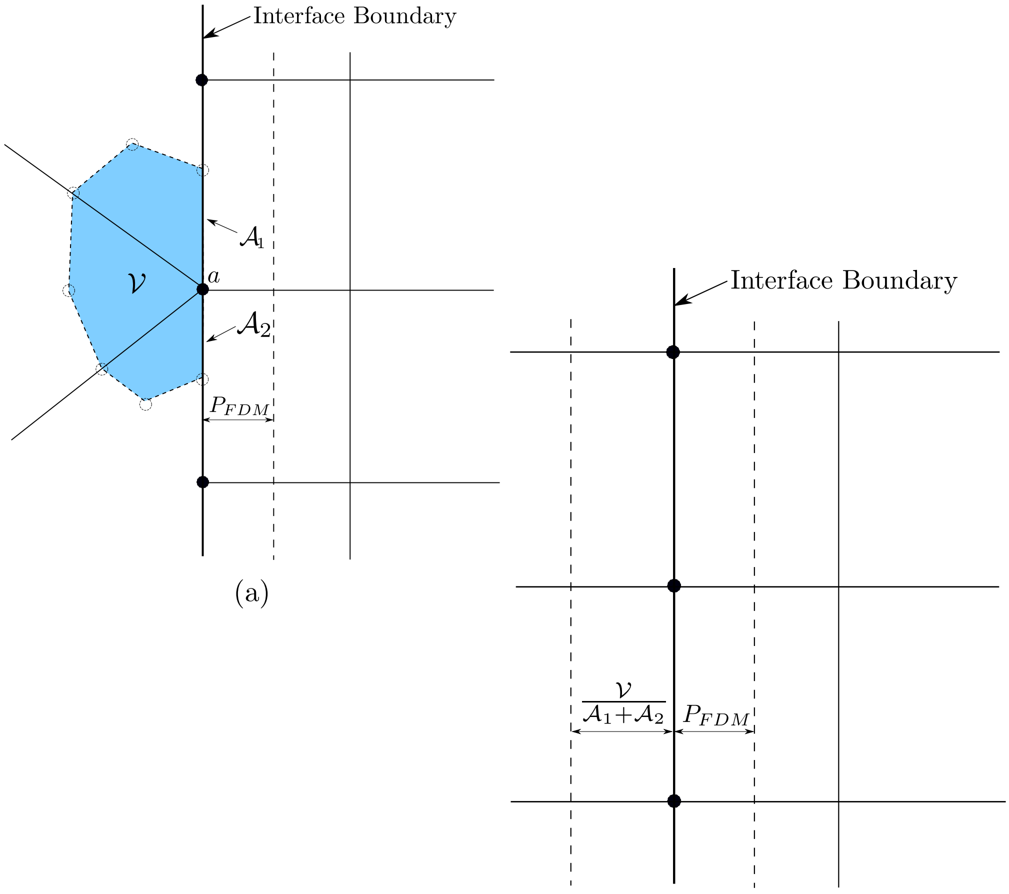
$$\mathbf{S}^{Inter} = \varepsilon \max\left(\frac{\gamma}{Pr}, \frac{5}{3}\right) \frac{\mu}{\rho_{Roe}} \frac{\mathbf{U} - \mathbf{U}_{target}}{Z} \tag{3.33}$$

The target values for each node is filled with the value received from the co-located node. The effect of using P_{FVM} is to interpret the finite volume mesh as a finite difference one as shown in Figure 3.3b.

The second form of the viscous SAT that is used at interface boundaries has the form

$$\mathbf{S}^{V} = -\varepsilon \frac{P_{target}}{Z} \left(\mathbf{F}_i^V \mathbf{n}_i + \mathbf{F}_{i,target}^V \, \mathbf{n}_{i,target}\right) \tag{3.34}$$

The target values for P, the viscous fluxes and unit normal are again taken from the neighbouring mesh. It is important to note that $\mathbf{n}$ and $\mathbf{n}_{target}$ are outward pointing normals for their respective volumes *i.e.* $\mathbf{n} = -\mathbf{n}_{target}$.



(a)

(b)

Figure 3.3: Schematic of co-located nodes on an interface boundary. (a) Dual-cell with the smallest ratio of $\mathcal{V}/(\mathcal{A}_1 + \mathcal{A}_2)$. (b) The finite volume mesh interpreted as a finite difference mesh.

3.4 Temporal Discretisation

The classic 4^{th} order 4-stage Runge-Kutta method is used for time advancement. This temporal discretisation is separate to that of the spatial and hence may be applied to Equation (3.11) and Equation (3.15). To this end, the temporal term is discretised and the equations expressed as follows

$$\frac{\Delta \mathbf{U}^j}{\Delta t^j} = \boldsymbol{w}\left(\mathbf{U}^j\right) \tag{3.35}$$

where all the spatial terms have been taken to the right-hand-side and evaluated as $\boldsymbol{w}$ at the current time-step j. The Runge-Kutta stages are then evaluated as

$$\begin{aligned} \mathbf{U}_0 &= \mathbf{U}^j \\ \mathbf{U}_k &= \mathbf{U}^j - R_k \Delta t^j \boldsymbol{w}\left(\mathbf{U}_{k-1}\right) \\ \mathbf{U}^{j+1} &= \mathbf{U}^j - \frac{1}{6}\Delta t^j \left(\boldsymbol{w}\left(\mathbf{U}_0\right) + 2\boldsymbol{w}\left(\mathbf{U}_1\right) + 2\boldsymbol{w}\left(\mathbf{U}_2\right) + \boldsymbol{w}\left(\mathbf{U}_3\right)\right) \end{aligned} \tag{3.36}$$

where the subscript $k = 1, 2, 3$ is the Runge-Kutta stage which has the stage co-efficient $R_k = 0.5, 0.5, 1$ respectively. The time-step Δt is calculated as the minimum time-step across all nodes in the domain. In the case that there is an interface boundary, the minimum of each domain is chosen. According to Blazek [41], the optimum time-step is as follows

$$\begin{aligned} \Delta t^j &= \mathrm{CFL}\frac{\mathcal{V}}{\sum\limits_{i=1}^{3}\left(\lambda_i^{I,j} + 4\lambda_i^{V,j}\right)} \\ \lambda_i^I &= \left(|u_i| + a\right)\Delta x_i \\ \lambda_i^V &= max\left(\frac{4}{3\rho}, \frac{\gamma}{\rho}\right)\frac{\mu}{Pr}\frac{\Delta x_i^2}{\mathcal{V}} \end{aligned} \tag{3.37}$$

where Δx_i is the effective mesh spacing in the i^{th} direction and CFL is the Courant-Friedrichs-Lewy number. In the case that a steady state simulation is being performed, the above solution procedure is still used. The simulation is considered converged when the L_2-norm of the right-hand-side at every node is below a predetermined convergence tolerance. In passing it is noted that the speed of convergence may be substantially improved via the use of local-time-stepping as well as implicit relaxation methods. This is however earmarked for follow-on work.

3.5 Parallel Computing

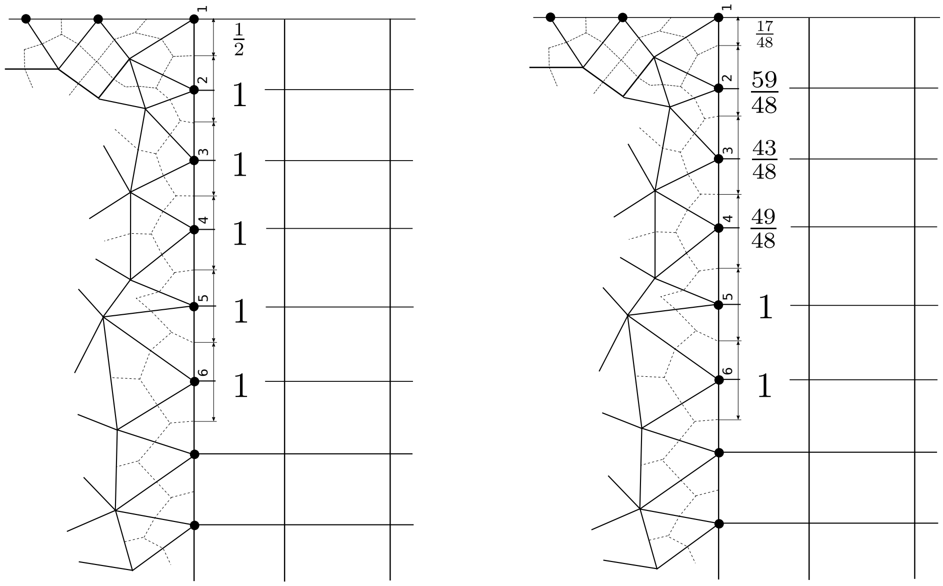
The Runge-Kutta scheme is a matrix-free solution method which makes it suitable for use on distributed memory computing platforms. Each code decomposes their respective domains into multiple sub-domains and the communication between them is done asynchronously. This level of communication is separate to the communication between *Elemental*® and *Essense* , and is performed within the appropriate communicator.

For the purpose of domain decomposition, *Elemental*® employs the METIS [42] library. A weighted decomposition is performed on the mesh connectivity graph so that each sub-domain has a similar number of edges and nodes. In the case of *Essense* , a load balancing algorithm is used to decompose the domain equally amongst the available computational cores.

Chapter 4

Interface Procedures

This chapter details two considerations necessary to realise the coupling of *Elemental*® to *Essense* . The first is that of conservation and stability: this requires modifying the boundary faces of interface nodes for the FVM in the vicinity of a physical boundary (Figure 4.1). The other is the coupling implementation that supports the simultaneous execution of the two codes.



2^{nd} Order

4^{th} Order

Figure 4.1: Schematic showing the modification of the finite volume method boundary faces in the vicinity of a physical boundary. Left: No change is necessary for 2^{nd} Order. Right: Boundary faces are modified according to the 4^{th} order finite difference P norm.

4.1 Conservation and Stability

The SAT interface terms presented in the previous chapter were derived on the assumption that boundary faces shared by the co-located nodes are of the same magnitude. This is necessary to enforce conservation and guarantee linear stability. As is seen from Equation (3.9) and Figure 4.1, this requirement is automatically fulfilled when coupling *Elemental*® to 2^{nd} order *Essense* . However, since the P norms for higher than 2^{nd} order are different, the boundary faces for the finite volume method are modified as shown.

4.2 Coupling Implementation

Following the philosophy presented in[29], *Elemental*® and *Essense* are run in a Multiple-Program-Multiple-Data session. To this end, this work involved the development of an interface library in *Elemental*® to facilitate the communication between the two codes. This allows them to run mostly independently of each other thus avoiding the need to develop a single hybrid code. To this end, two modifications to each code are made. The first of these is splitting of the common world communicator and giving each code a unique identifier. The second comprises the development of the interface library to allow for the exchange of data at appropriate times.

4.2.1 Splitting the Communicator

The common world communicator is created when the MPI environment is initialised by calling MPI_Init(). This global communicator is then split into sub-communicators according to the unique id keys that are assigned to each code through command-line arguments (Listing 4.1). This affords one the ability to make multiple *Essense* or *Elemental*® communicators and have them communicate with each other.

```
mpirun -np 32 Elemental -id 0 : \
       -np 32 Essense -id 1 : \
       -np 32 Elemental -id 2
```

Listing 4.1: Code listing illustrating the use of command-line arguments to pass unique id keys to *Elemental*® and *Essense* .

4.2.2 Interface Library

The interface library is responsible for linking interface nodes together by establishing a communication pattern, enforcing time synchronisation and the exchange of information. The use of each functionality during the solution process is illustrated in Figure 4.2.

Linking Interface Nodes

The building of the communication pattern begins with finding the co-located nodes. This is done by extracting and comparing the nodal coordinates from each code. Should coordinates match to within a prescribed tolerance then the two nodes are matched together and the association stored for future use *i.e.* the search only has to be performed once. To this end, an Alternating Digital Tree library (available in *Elemental*®) [43] is employed which results in a $\log_2(N)$ search where N is the number of nodes that are to be searched.

Data Exchange

The conservative variables being solved for (density, momentum and energy) and viscous fluxes are to be extracted from each code and transferred to the other. Once each code has sent and received its respective data, the data can be sent to the correct nodes through the use of the stored node association.

Time synchronisation

Each code will perform its respective time step size calculation which will be sent to the interface library. Upon receiving the time step size, the library will compare the two time step sizes and choose the smallest one. This will then be communicated and each code will adjust its time-step size accordingly.

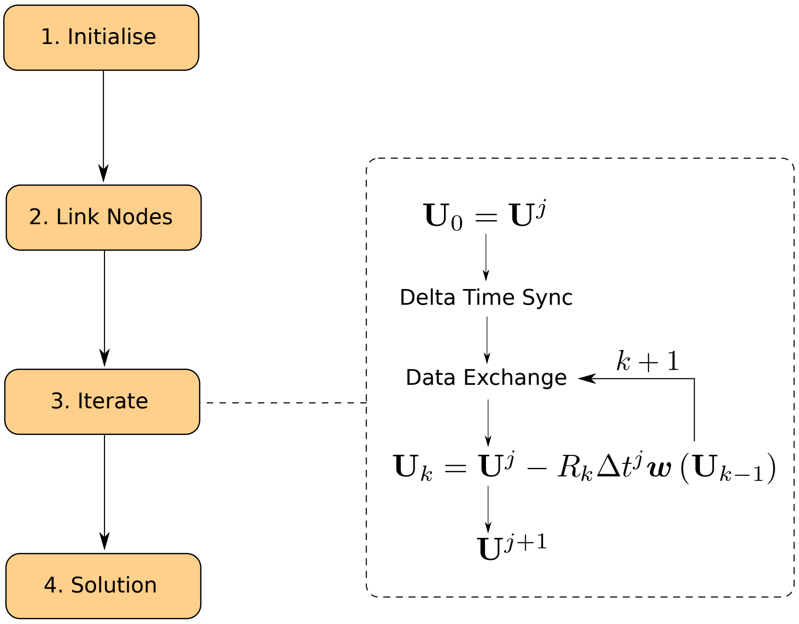


Figure 4.2: Flow chart illustrating the solution procedure of the hybrid *Elemental®* - *Essense* scheme.

Each of the above mentioned instructions are collective and blocking. Furthermore, their implementation supports the interoperability between C++ (for *Elemental®*) and Fortran (for *Essense*) [44].

Chapter 5

Numerical Results and Evaluation

The inviscid and viscous components of the interface implementation are assessed using the isentropic vortex and a 3-D manufactured solution respectively. By comparing these analytical solutions to the numerical result, the overall order of accuracy of the scheme may be compared to the expected value. This serves to evaluate the interface implementation in addition to the SAT inflow and outflow boundary conditions. To this end, 3 different coupling combinations of *Essense* and *Elemental*® are considered: each code coupled to itself and the hybrid *Elemental*® - *Essense* solver. Thereafter, the applicability of the hybrid solver to investigating trailing vortices is demonstrated by considering the NASA CRM under transonic flow conditions.

5.1 Isentropic Vortex

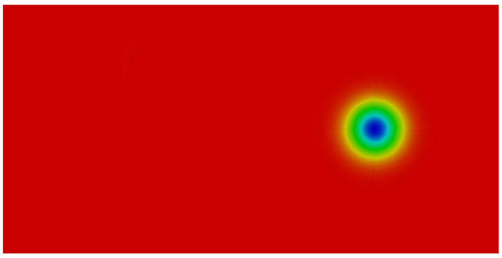
The first test-case involves prescribing an isentropic vortex as an initial condition in a 2D inviscid flow field and allowing it to propagate. The isentropic vortex is an analytical solution to the Euler equations which in non-dimensional form reads:

$$\begin{aligned} f &= 1 - (x_1 - x_0 - Mt)^2 - (x_2 - y_0)^2 \\ T &= 1 + (1 - \gamma)\frac{k^2 e^f}{8\pi^2} \\ u_1 &= M + k(y_0 - x_2)e^{(f/2\pi)} \\ u_2 &= (x_1 - x_0 - Mt)ke^{(f/2\pi)} \\ \rho &= T^{(\gamma-1)} \end{aligned} \tag{5.1}$$

where k is the vortex strength, x_0, y_0 are the starting coordinates of the vortex centre and f is the distance that the centre has moved with respect to the starting coordinate. The values used in this study are $x_0 = y_0 = 5$ and M$= \kappa = 0.5$. The spatial domain, $x_1 \in [0, 20]$ and $x_2 \in [0, 10]$, is discretised on two separate meshes with an interface at $x_1 = 10$. These are glued together via the developed interface conditions. An NxN Cartesian grid is used on both domains to negate the effect of mesh quality on the finite volume method. The meshes are refined according to $N = 50, 100, 150, 200$ and each simulation run such that the vortex crosses the interface ($t = 10$) and reaches the centre of the second domain ($t = 20$). These two instances in time are shown in Figure 5.1.

(a) t = 0s

(b) t = 10s



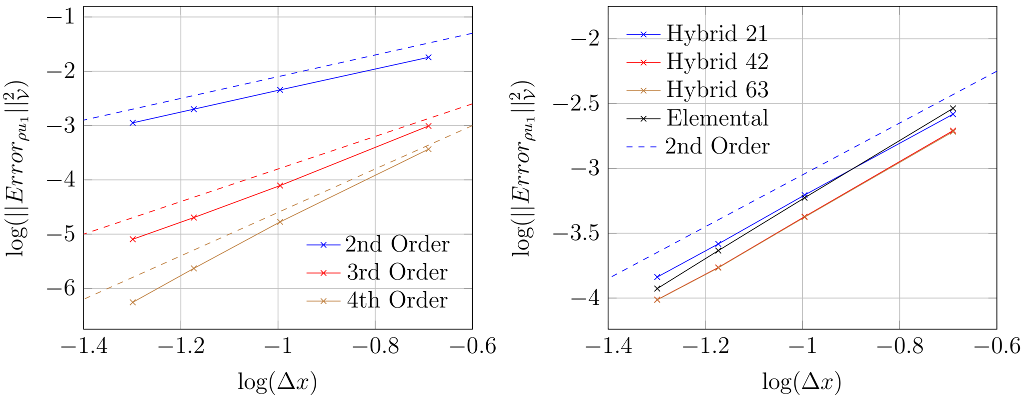
(c) t = 20s

Figure 5.1: Illustration of the isentropic vortex pressure field being solved for using 6^{th} order *Essense* on the left and *Elemental*® on the right. (a) $t = 0$ with the vortex centre at $x_1 = 5$. (b) $t = 10$ with the vortex centre on the interface boundary at $x_1 = 10$. (c) $t = 20$ with the vortex centre at $x_1 = 20$.

At t=20, the P-norm of the error in each of the conservative variables is evaluated according to

$$||Error||_{\mathcal{V}}^2 = \left(\sum_{i=1}^{\text{Total Nodes}} \mathcal{V}_i \left(\phi_{i,num} - \phi_{i,ana}\right)^2 \right)^{\frac{1}{2}} \tag{5.2}$$

where i is a node in the mesh and *ana* and *num* refer to the analytical and numerical solution respectively. The error in ρu_1 is plotted as a function of the mesh spacing in Figure 5.2 and is representative of the other variables. In doing so, the order of accuracy may be determined from the gradient of the resultant line. In the case of the hybrid solver, the coupling is tested using various order-of-accuracy operators in *Essense* on the left domain. These are distinguished in Figure 5.2 as Hybrid21, Hybrid42 and Hybrid63 which correspond to the 2^{nd}, 4^{th} and 6^{th} order operators respectively (the second digit refers to the order-of-accuracy normal to a boundary). Furthermore, a constant delta time step that corresponds to the finest mesh is chosen and used for all cases.



(a) (b)

Figure 5.2: Graphs showing the orders of accuracy for the 2D inviscid isentropic vortex problem as achieved using various scheme combinations. Solid lines indicate numerical results whilst dashed lines indicate the theoretical order of accuracy. (a) *Essense* is coupled to itself. (b) *Elemental*® is coupled to itself as well as the hybrid solver.

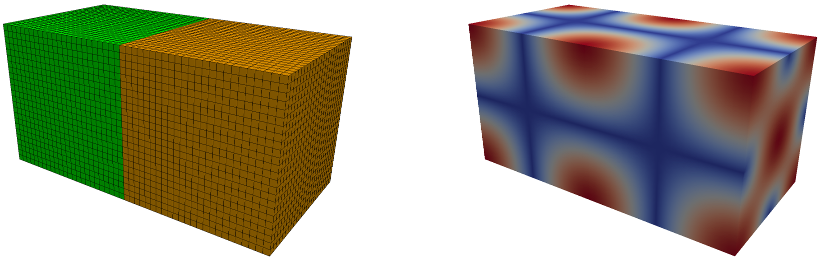
As mentioned in Chapter 3, an operator with a boundary order of accuracy of p will achieve a global order of p+1. It may be seen from Figure 5.2 that this is maintained for the various scheme combinations even though the vortex flows through the interface boundary. In fact Figure 5.2a, shows that the 4^{th} and 6^{th} order *Essense* operators, when coupled to themselves, achieve formal $p + 1$ accuracy. The 2^{nd} order *Essense* and *Elemental*® operators coupled to themselves both achieve 2^{nd} order. However, the magnitude of their errors are different. This may be attributed to the difference in discretisation operators *viz.* 2^{nd} order central difference with no artificial dissipation for *Essense* *vs.* flux limited MUSCL upwinding for *Elemental*® . Turning to the hybrid scheme, it may be seen from Figure 5.2b that regardless of the operator employed by *Essense* , 2^{nd} order accuracy is achieved. This is expected as comparison of Figure 5.2a and Figure 5.2b shows that the error from *Elemental*® overshadows that of the higher order operators. Thus the overall order is limited to that of *Elemental*® .

5.2 Manufactured Solution

The viscous interface implementation is evaluated by adding a 3^{rd} dimension to the previous 2D case. A resulting mesh with an interface is depicted in Figure 5.3a. Instead of the isentropic vortex equations, analytically proposed solutions to the Navier-Stokes equations are used. For the purposes of this study Re=100, M=0.5 and the following solutions are employed

$$\begin{aligned}
\rho &= 1 + e^{-0.1t}\left(cos(2\pi x_1)sin(2\pi x_2)sin(2\pi x_3)\right)\\
u_1 &= e^{-0.1t}\left(cos(2\pi x_1)sin(2\pi x_2)sin(2\pi x_3)\right)\\
u_2 &= e^{-0.1t}\left(sin(2\pi x_1)sin(2\pi x_2)cos(2\pi x_3)\right)\\
u_3 &= e^{-0.1t}\left(cos(2\pi x_1)sin(2\pi x_2)cos(2\pi x_3)\right)\\
T &= 3 + e^{-0.1t}\left(cos(2\pi x_1) + sin(2\pi x_2) + sin(2\pi x_3)\right)
\end{aligned} \tag{5.3}$$

where the magnitude of the resultant velocity fluctuation is shown in Figure 5.3b. The resultant source terms are given in Appendix B. An $N \times N \times N$ Cartesian grid is again used on both domains.



(a) (b)

Figure 5.3: Illustration of (a) the spatial domain, $x_1 \in [0, 20]$ and $x_{2,3} \in [0, 10]$, being discretised on two separate meshes with an interface at $x_1 = 10$. (b) the magnitude of the velocity flow field for the 3D manufactured solution test case.

The meshes are refined according to $N = 25, 35, 45, 55$ and each simulation run until $t = 1.8$ at which time the error in each of the conservative variables is evaluated according to Equation (5.2). As before, a constant time-step corresponding to the finest mesh level is chosen.

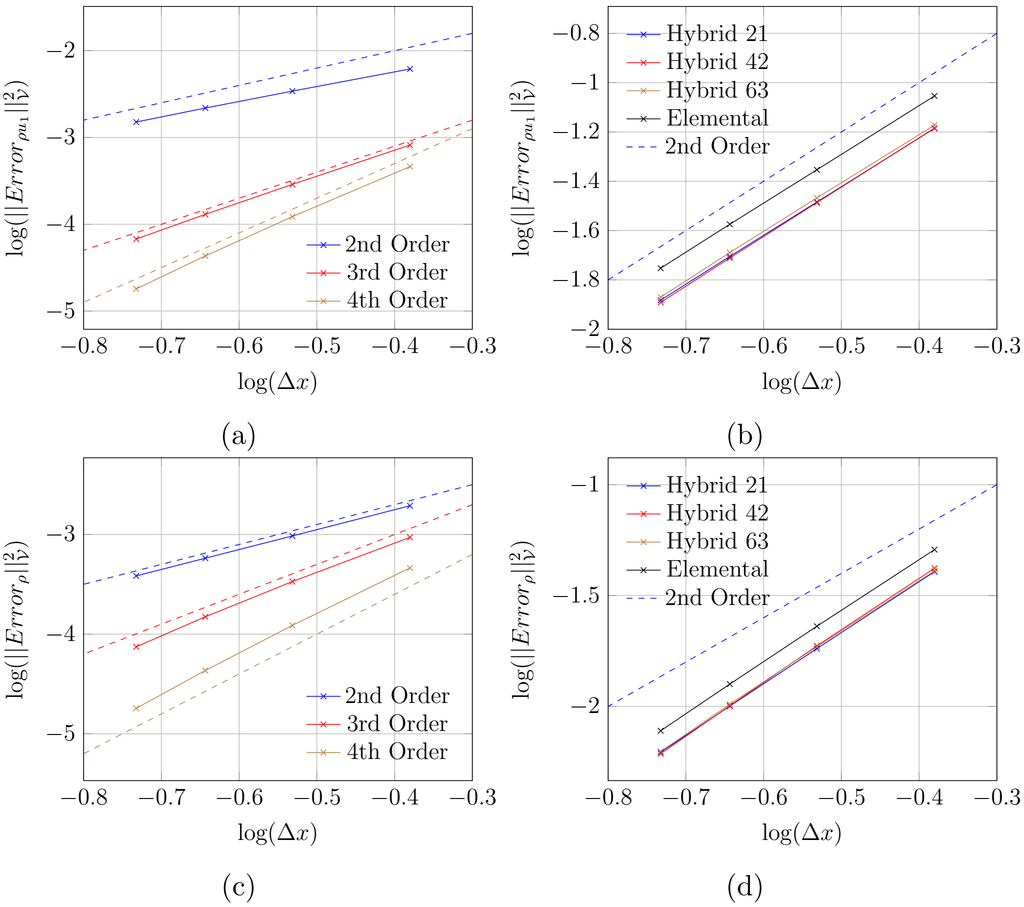


Figure 5.4: Graphs showing the orders of accuracy for the 3D manufactured solution problem as achieved using various scheme combinations. Solid lines indicate numerical results whilst dashed lines indicate the theoretical order of accuracy. (a) Error in ρu_1 where *Essense* is coupled to itself. (b) Error in ρu_1 where *Elemental*® is coupled to itself as well as the hybrid solver. (c) Error in ρ where *Essense* is coupled to itself. (d) Error in ρ where *Elemental*® is coupled to itself as well as the hybrid solver.

Similar trends to the isentropic vortex convergence study are presented in Figure 5.4 with the exceptions of the 2^{nd} order *Essense* and *Elemental*® schemes coupled to themselves. With regards to *Essense* , nominal 2^{nd} order convergence is achieved in the ρu_1 variable whilst for the others, such as ρ in Figure 5.4c, 2^{nd} order is achieved. For *Elemental*® , better than 2^{nd} order is achieved in some variables such as ρ in Figure 5.4d. There is almost an order of magnitude difference in the error between *Elemental*® and 2^{nd} order *Essense* . This may be due to *Essense* using a 1st order approximation of 2^{nd} derivatives normal to the boundary whilst *Elemental*® employs a 0^{th} order approximation. Considering that the *Essense* higher order schemes coupled to themselves achieve their expected orders and that they are more sensitive to incorrect implementation, it strongly suggests that the viscous SAT terms are implemented correctly.

The convergence studies performed above highlight a critical factor for application purposes. Although the global accuracy of the hybrid scheme is 2^{nd} order, the magnitude of discretisation error is far lower in regions where the higher order FD operators are used. As such, the FV mesh needs to be sufficiently refined in the vicinity of the hybrid interface to reduce the magnitude of the 2^{nd} order error. If this is not done, then this error will be propagated through the higher order domain.

5.3 NASA CRM

The CRM [45] industrially relevant geometry represents a modern wide-body aircraft that is designed to have a cruise speed of M=0.85 at a nominal lift coefficient of $C_L = 0.5$. Figure 5.5 depicts the particular geometry used in this study *viz.* the NASA CRM wing-body-tail configuration with a horizontal tail incidence setting of 0°.

Table 5.1: Table summarising the geometric reference quantities of the NASA CRM.

$\tilde{S}_{ref}$	$\tilde{c}_{ref}$	$\widetilde{Span}$	$\tilde{x}_{1ref}$	$\tilde{x}_{2ref}$	$\tilde{x}_{3ref}$	AR
383.69 m^2	6.99 m	58.76 m	33.68 m	11.90 m	4.52 m	9

Table 5.1 above summarises the dimensional (denoted by $\tilde{\bullet}$) reference quantities used when calculating, for example, the lift coefficient. The subscript ref refers to a reference quantity, S_{ref} the wing plan-form area based on the Wimpress area, c_{ref} the mean aerodynamic chord length, $x_{i_{ref}}$ the reference coordinate and AR is the wing aspect ratio. The Reynolds number used in this study is Re$= 5 \times 10^6$ which corresponds to experiments conducted in the NASA Ames transonic wind tunnel [45].

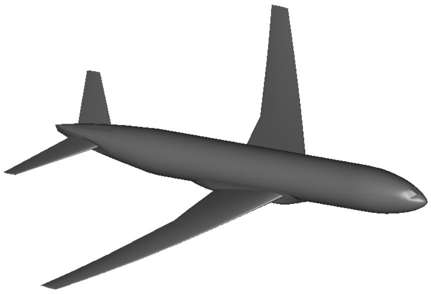


Figure 5.5: NASA Common-Research-Model wing-body-tail configuration with a horizontal tail incidence setting of 0°.

The flow domain of interest is non-dimensionalised such that $c_{ref} = 1$ and it extends 100 c_{ref} away from the geometry in all directions as depicted in Figure 5.6. In the interest of reducing computational expense, a symmetry plane was used. The resultant domain was split into 3 blocks *viz.* FV and 2 FD blocks. The latter extended circa 3 c_{ref} downstream of the wing-tip in order to form and propagate the trailing vortices. An adiabatic no-slip condition was applied on the aircraft surface and symmetry and far field conditions applied to the other boundaries; all of which were applied through the SBP-SAT framework.

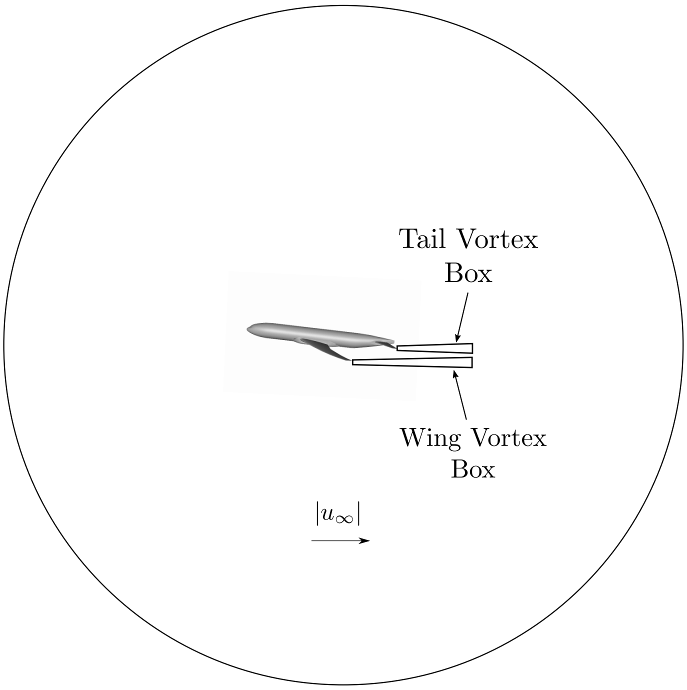


Figure 5.6: Figure illustrating the addition of the wing and tail refinement boxes to capture the development of the vortices. The refinement boxes extend from their respective tips up to 3 c_{ref} behind the wing-tip.

5.3.1 Mesh Development

It is known from the Betz method [8] that the span-wise loading on the aircraft drives the development of the vortices. It is therefore important that the *Elemental*® mesh be sufficiently fine to accurately resolve the load distribution. Furthermore, the 2^{nd} order nature of the FVM will numerically dissipate the vortices in the absence of a suitably fine mesh and erroneous data will be propagated by the FD operators. This led to the vortex mesh blocks being placed directly behind the wing and tail tips and the small regions in-between refined.

In light of the above, three refinement studies were performed to ensure that the solution is in the asymptotic convergence regime. Due to the co-located nature of the interface treatment and the fact that the solution flows from the FV mesh to the FD meshes, the refinement studies were done using *Elemental*® in isolation. The first study, using no vortex boxes, verifies the implementation of the no-slip boundary condition by refining the surface mesh of the CRM. In doing so, confidence in the solution is given and the convergence of the lift-coefficient may be checked. The second introduced a single vortex block behind the wing-tip. The L_2-norm of the up-wash along a line sensor that passes through the vortex was then used as the convergence metric. Lastly, digitised data from Ning and Kroo [17] was used as an inflow condition to determine the spacing required by the FVM to propagate an already formed vortex along the vortex box.

The meshes for the above refinement studies were generated using the Pointwise commercial software. Delauney triangulation was used to generate the aircraft's surface mesh as well as the volume mesh. For studies with vortex boxes, a hybrid element mesh was created. The vortex boxes consisted of hexahedral elements surrounded by a pyramid layer to allow for the transition to tetrahedral elements (see Figure 5.22). In all the aforementioned studies, the steady state solution was solved for and was considered converged once the residuals had dropped by 5 orders of magnitude.

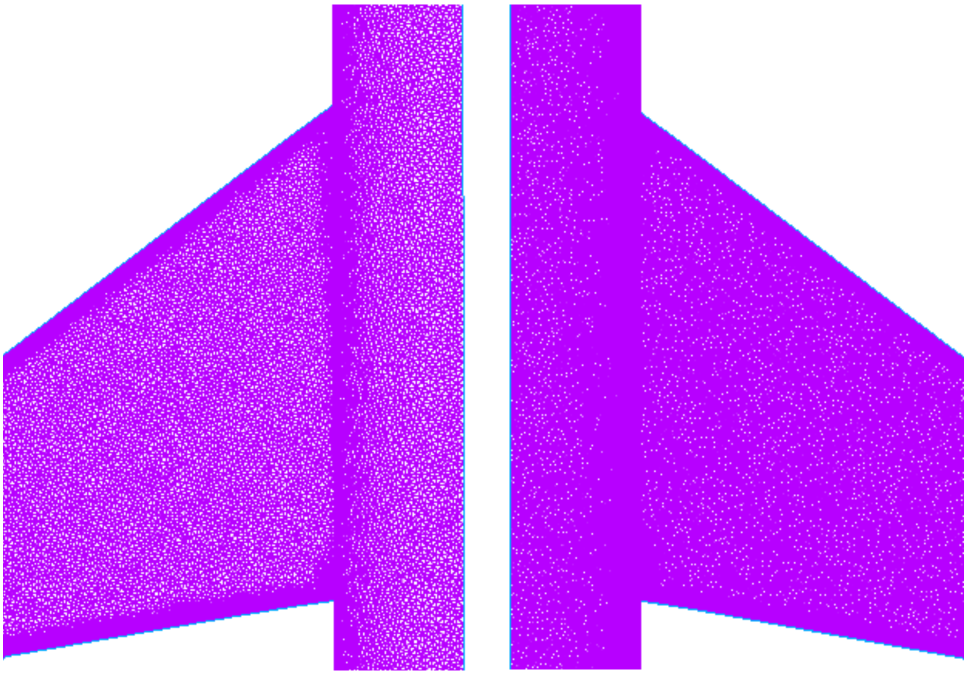
Normal Velocity and Lift Coefficient

The implementation of the adiabatic no-slip condition was verified using four meshes with increasing node density on the surface of the CRM. These are referred to as Tiny, Coarse, Medium and Fine and their properties are summarised in the table below.

Table 5.2: Table summarising the properties of the meshes used in the normal velocity and lift co-efficient refinement study.

Mesh Properties	**Tiny**	**Coarse**	**Medium**	**Fine**
Max edge length on wing (mm)	343	22	17	12
Total nodes (millions)	0.95	1.76	4.05	9.00

As mentioned in Chapter 2, the development of the vortices is mainly an inviscid process. Based on this and in the absence of a turbulence model, an under resolved viscous simulation was undertaken. This means that the boundary layer was not resolved. In doing so, the SAT no-slip condition is expected to behave like a slip condition [46, 47]. Unlike strong injection, the velocity on the surface of the CRM is not set to be 0. Instead, it is approximated and the convergence thereof is determined by the order of the discretisation. In the case of *Elemental*® , 2^{nd} order is expected.



(a) (b)

Figure 5.7: Figure illustrating the coarse and medium meshes used to verify the convergence of the normal velocity to the CRM wing surface. (a) Coarse level. (b) Medium level.

As described in Chapter 3, the no-slip condition is applied in two parts *viz.* the no penetration and zero velocity conditions. The degree to which the zero velocity condition is satisfied is a function of the normal wall spacing on the surface of the CRM. The smaller the spacing, the closer to zero the velocity is. Due to there being an absence of a boundary layer, the convergence of the no-penetration condition was only considered. This was done by calculating an average normal velocity from a sample of 50 nodes on the upper surface of the main wing. As mentioned, the convergence of this functional should be second order. This is seen to be the case from a Richardson Extrapolation plot in Figure 5.8b. For a 2^{nd} order scheme, the plot should appear as a straight line as the mesh spacing tends to 0. Note that for the purposes of this case, the representative spacing was chosen to be the normal wall spacing. Due to the unstructured nature of the mesh, the maximum edge length on the upper surface of the wing was used as a proxy for the normal spacing.

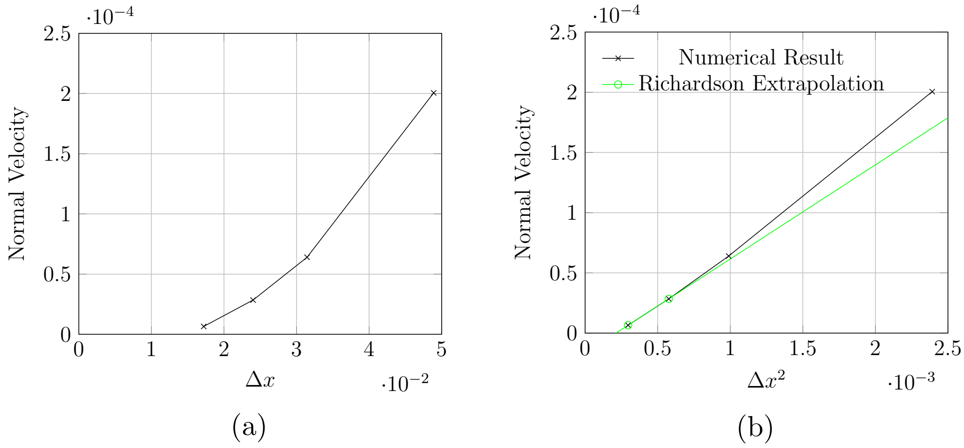
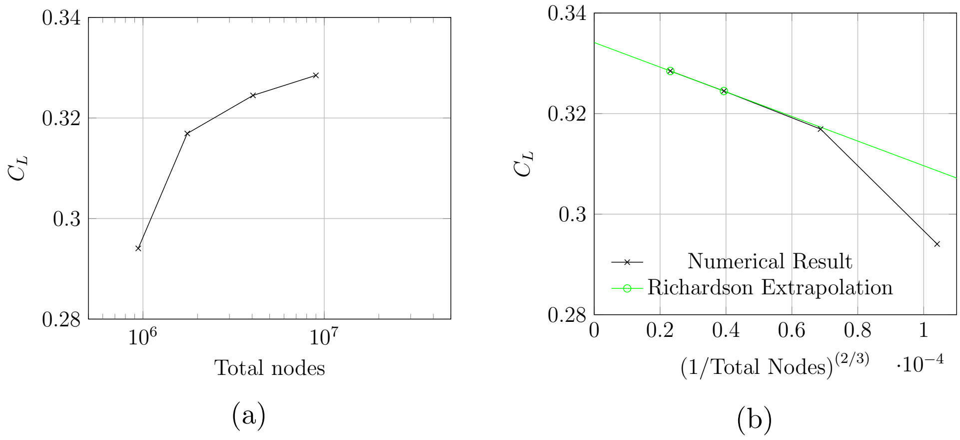


Figure 5.8: Graphs showing the result of the convergence study for the normal velocity on the surface of the CRM wing. (a) The average normal velocity on the wing *vs.* the maximum edge length on the CRM wing. (b) The Richardson Extrapolation of the average normal velocity on the wing. The extrapolation is calculated by fitting a straight line through the two finest data points.

The significance of the above 2^{nd} order convergence of the boundary normal velocity is twofold. Firstly, the boundary condition has been implemented correctly. Secondly, the asymptotic convergence regime for the inviscid solution of the lift coefficient has also been reached. This is clearly the case from Figure 5.9 where C_L is plotted against the total number of nodes in the mesh. Since C_L is a global functional, a representative spacing for the entire mesh is required to produce the Richardson Extrapolation. Due to the unstructured nature of the mesh, this is calculated according to

$$\Delta x = \left(\frac{1}{Total\ Nodes} \right)^{1/3} \tag{5.4}$$

where Δx is now the representative mesh spacing. As seen in Figure 5.9b, all but the coarsest mesh fall on the Richardson Extrapolation line. This again indicates 2^{nd} order convergence in C_L.



(a) (b)

Figure 5.9: Graph illustrating the convergence of the lift coefficient for the CRM at an AoA= 0°, M=0.85 and Re= 5×10^6. (a) Lift coefficient *vs.* the total number of nodes. (b) The Richardson Extrapolation of the lift coefficient. The extrapolation is calculated by fitting a straight line through the two finest data points.

The Grid Convergence Index (GCI) [48] gives an error band for solutions obtained from a pair of meshes. The refinement ratio between the two meshes is defined as

$$r = \frac{\Delta x_2}{\Delta x_1} \tag{5.5}$$

where the subscript 1 refers to a finer mesh and 2 to the coarser counterpart. The GCI for the finer mesh may then be defined as

$$GCI = F_s \left| \frac{(f_2 - f_1)/f_1}{1 - r^2} \right| \tag{5.6}$$

where $F_s = 1.25$ is the relaxation factor. Thus the meshes in the refinement study are paired as Tiny-Coarse, Coarse-Medium and Medium-Fine. The resultant GCIs for the lift coefficient are calculated and summarised in the table below.

Table 5.3: Table summarising the GCI of the lift co-efficient for the tiny, coarse, medium and fine meshes.

Tiny	**Coarse**	**Medium**	**Fine**
-	6.28 %	3.12 %	1.60 %

Based on Table 5.3 and Table 5.2, the Coarse surface mesh was deemed to provide a balance between computational expense and engineering accuracy.

Up-wash

The mesh spacing that will adequately capture the development of the vortices by the FVM was determined by placing a single vortex box of refinement is behind the main wing-tip as shown in Figure 5.10. As before, four meshes were created. These meshes make use of the Coarse level surface mesh deemed adequate in Section 5.3.1 in conjunction with increasing refinement of the wing vortex box.

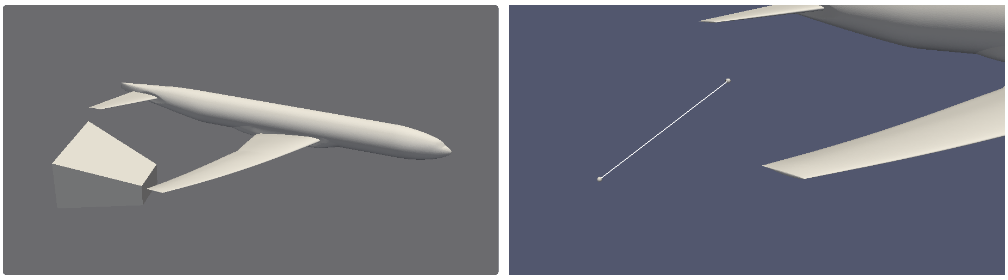
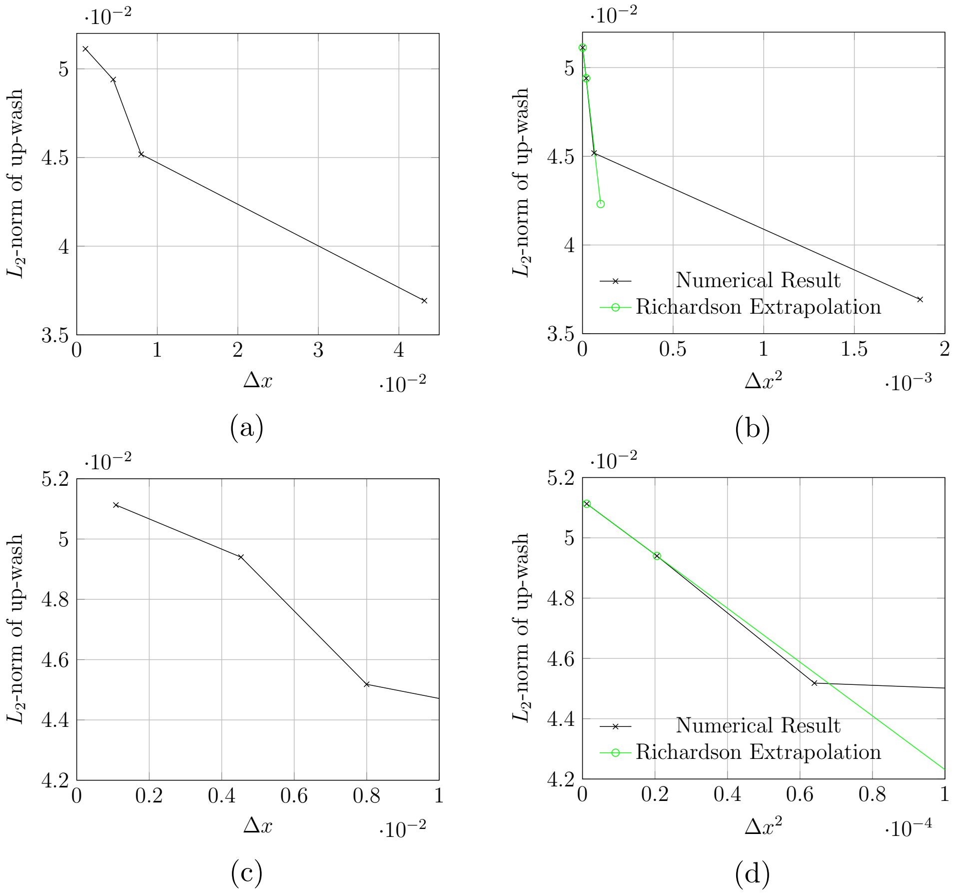

(a) (b)

Figure 5.10: Figures illustrating (a) the FD box of refinement placed behind the wing-tip. (b) the line sensor along which the L_2-norm of the up-wash is calculated.

As seen in Figure 5.10a, the wing box does not extend the full 3 c_{ref} as only the forming of the vortex as it enters the box is of concern. The spacing on the face closest to the wing-tip is refined as shown in Table 5.4 and the aspect ratio set to be 1.7 as was deemed to be appropriate by Ning and Kroo [17]. A line sensor was placed 1 c_{ref} behind the wing-tip that passes through the vortex centre as shown in Figure 5.10b. The L_2-norm of the up-wash is then computed along the line and plotted against the spacing as summarised in Table 5.4 and plotted in Figure 5.11.

Table 5.4: Table summarising the mesh quantities used in the up-wash refinement study.

Mesh Properties	**Tiny**	**Coarse**	**Medium**	**Fine**
Δx (mm)	302	56	32	8
GCI of L_2 error of up-wash (%)	-	14.8	4.5	0.3



(a) (b)

(c) (d)

Figure 5.11: Graph illustrating the convergence of the up-wash 1 c_{ref} downstream of the CRM wing-tip at an AoA= 0°, M= 0.85 and Re= 5×10^6. (a) Wing-tip up-wash *vs.* mesh spacing on the wing box. (b) The Richardson Extrapolation of the wing-tip up-wash. (c) Zoom plot of the wing-tip up-wash *vs.* mesh spacing. (d) Zoom plot of the Richardson Extrapolation.

The Tiny mesh is far from the asymptotic regime as shown in Figure 5.11a. As a result, aggressive refinement was undertaken and the asymptotic regime has been reached for the other 3 meshes as illustrated in Figure 5.11c. Furthermore, this convergence is nominally 2^{nd} order as shown in Figure 5.11d. The GCIs were again calculated as summarised in Table 5.4 and the up-wash profiles along the line sensor plotted in Figure 5.12. Based solely on the GCI, the Medium mesh would have been chosen. However, inspection of Figure 5.12 reveals that it does not capture the peak up-wash as well as the Fine mesh. Since the purpose is to investigate the trailing vortices, the Fine mesh spacing (with a GCI of 0.3%) was therefore selected for the application study.

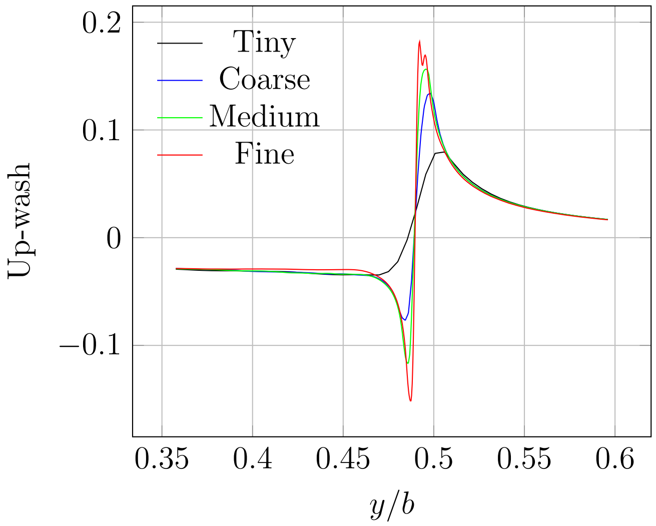


Figure 5.12: Graph of the up-wash profiles along the line sensor placed 1 c_{ref} behind the wing-tip. AoA= 0°, M=0.85 and Re= 5×10^6.

Vortex Propagation

In this project, the placement of the boxes that are to capture the trailing vortices occurred neither with a priori knowledge of the vortex position nor automatic mesh refinement. As such, it remained a concern that the core of the vortex would remain with in the refinement box with the rest spilling outside of it. Thus it was deemed informative to know the mesh spacing required by the FVM to propagate an already formed vortex downstream. For this purpose, *Elemental*® was used to propagate digitised up-wash data, from Ning and Kroo [17] for the NASA CRM, 1 wing-span (or 8.4 c_{ref}) downstream.

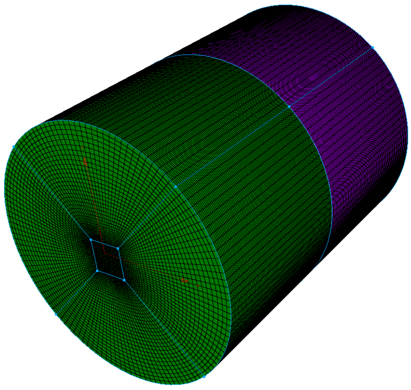


Figure 5.13: Figure illustrating the 1 wingspan (or 8.4 c_{ref}) long mesh used to determine the mesh spacing needed by *Elemental*® to propagate the vortex.

The domain takes the shape of a cylinder that is 1 wingspan in length and meshed with a structured H-Grid typology as shown in Figure 5.13. The cross-sectional spacing on the square part is refined according to Table 5.5 and the aspect ratio again set to be 1.7. The digitised data is used as an inflow condition whilst an outflow condition is used at the downstream face. A line sensor was placed at the midway point (where the green and purple surfaces meet in Figure 5.13) along which the percentage error in the L_2-norm of the up-wash solution was computed. This is plotted against the spacing from Table 5.5 in the figure below.

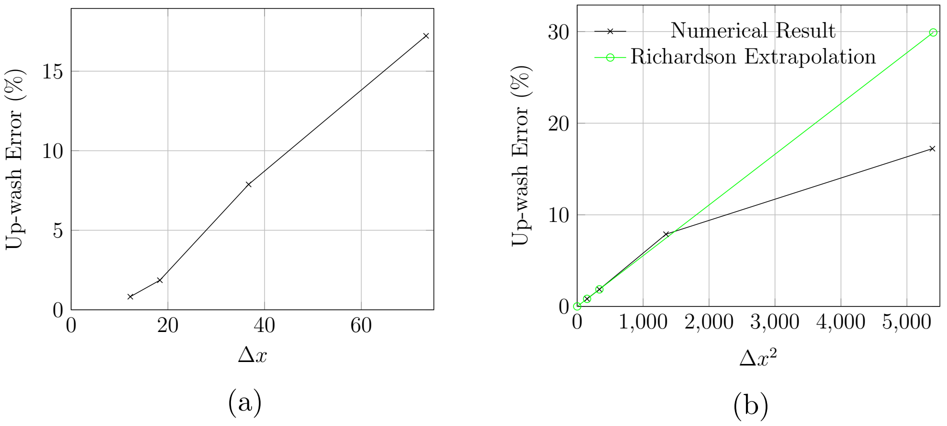


Figure 5.14: Graph illustrating the convergence of the percentage error in L_2-norm of the up-wash for the digitised vortex that has been propagated half a wing span. (a) Percentage error in up-wash *vs.* representative mesh spacing. (b) The Richardson Extrapolation of the percentage error in up-wash.

Table 5.5: Table summarising the mesh quantities used in the vortex propagation refinement study.

Mesh Properties	**Tiny**	**Coarse**	**Medium**	**Fine**
Δx (mm)	734	367	183	122
GCI of up-wash error (%)	-	9.92	2.33	1.04

Similar trends to the previous up-wash refinement study are seen in that the Tiny mesh is quite far from the asymptotic regime. The remaining three mesh levels are in the asymptotic region and the order of convergence is again seen to be 2. The resultant GCIs are summarised in Table 5.5 and the Medium mesh spacing deemed a balance between computational expense and engineering accuracy.

5.3.2 Hybrid Simulation

Due to the computational expense experienced during the refinement studies, an inviscid simulation was performed for the hybrid simulation. The mesh was created using the Coarse surface mesh from Section 5.3.1. The two vortex boxes were inserted and the Fine spacing from Section 5.3.1 was used on the faces closest to the wing and tail. The spacing on the downstream faces of the vortex boxes were set to the Medium spacing from Section 5.3.1. In addition, *Essense* was set to use its 6^{th} order operators and used to solve for the vortex boxes whilst *Elemental*® solved for the rest of the domain. The geometry was rotated until the design $C_L = 0.5$ was reached for M=0.85. As before, a steady state solution was solved for and considered converged when the residuals had dropped by 5 orders of magnitude.

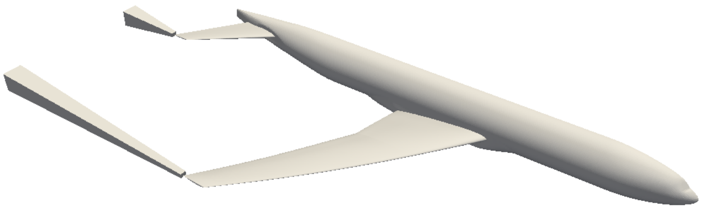
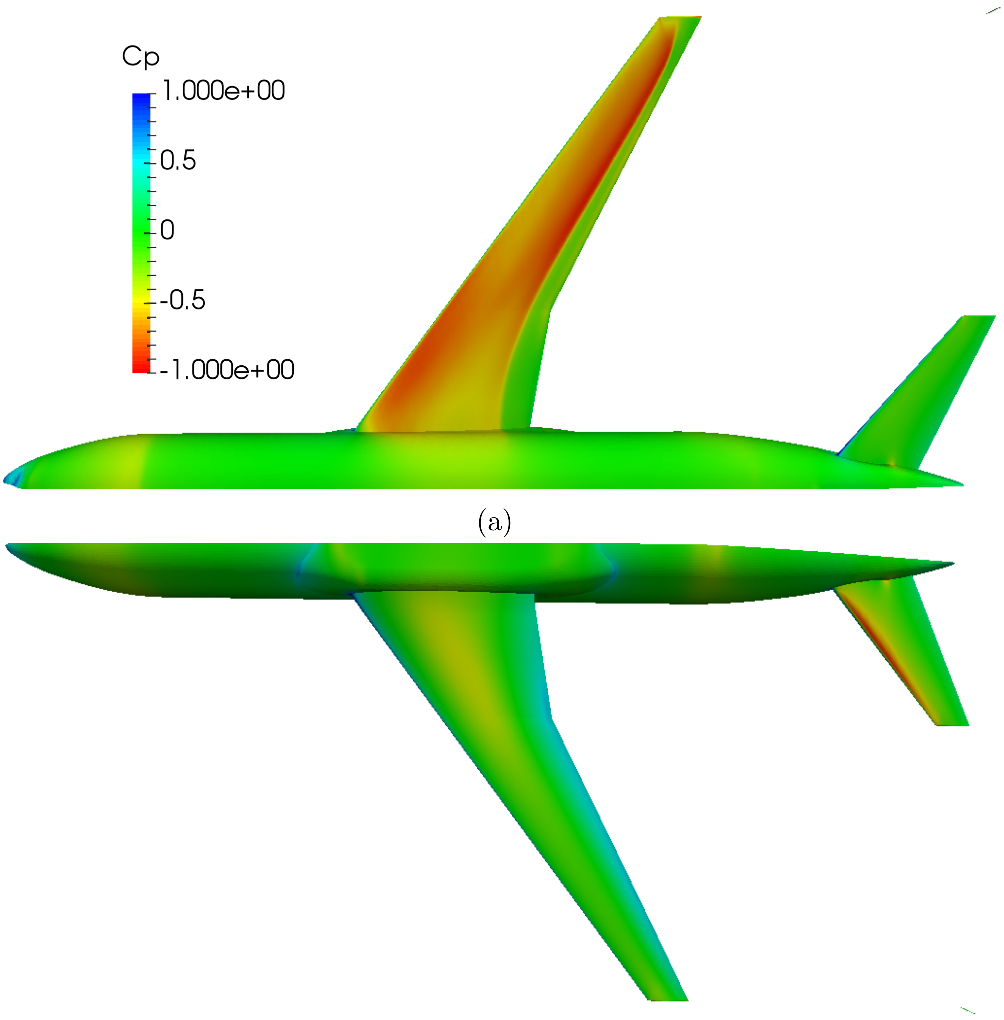


Figure 5.15: Figure illustrating the positioning of the wing and tail refinement boxes used for the hybrid simulation.

The pressure coefficient, C_p, is calculated according to

$$C_p = \frac{P - P_\infty}{0.5\rho_\infty \left|u_\infty\right|^2} \tag{5.7}$$

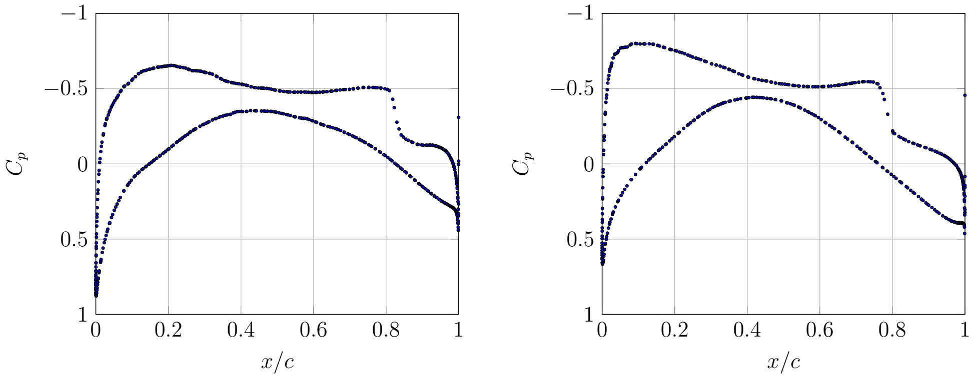
and the resultant distribution on the surface of the CRM is presented in Figure 5.16. There is a strong standing shock on the upper surface of the main wing which sits at circa 80% of the local chord length as seen from the C_p plots in Figure 5.17. Due to flow wrap around, it moves forward to about 70% of the local chord near the wing-tip. Shocks are also visible around the collar where the nose of the aircraft joins the fuselage and on the lower surface of the tail.



(a)

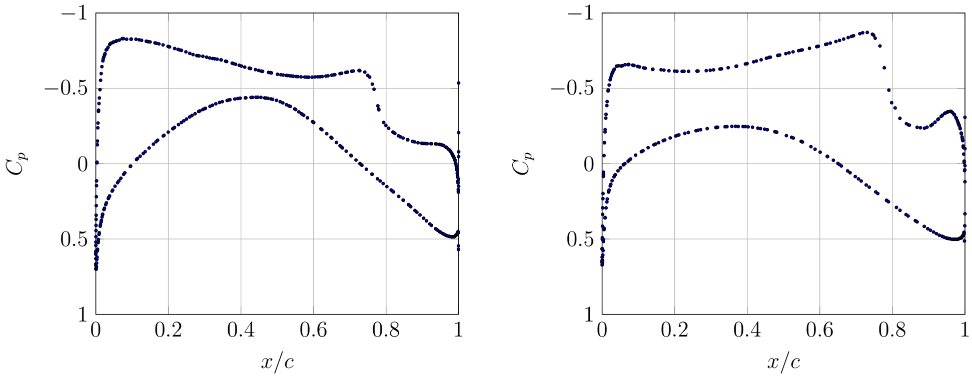
(b)

Figure 5.16: Figure of the C_p distribution on the CRM for M=0.85 and $C_L = 0.5$. (a) Upper surface. (b) Bottom surface.



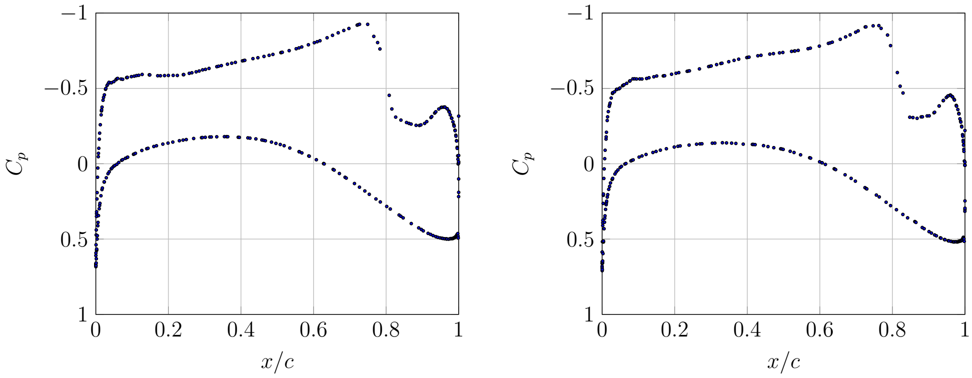
(a) 11.5% Span

(b) 20.1% Span



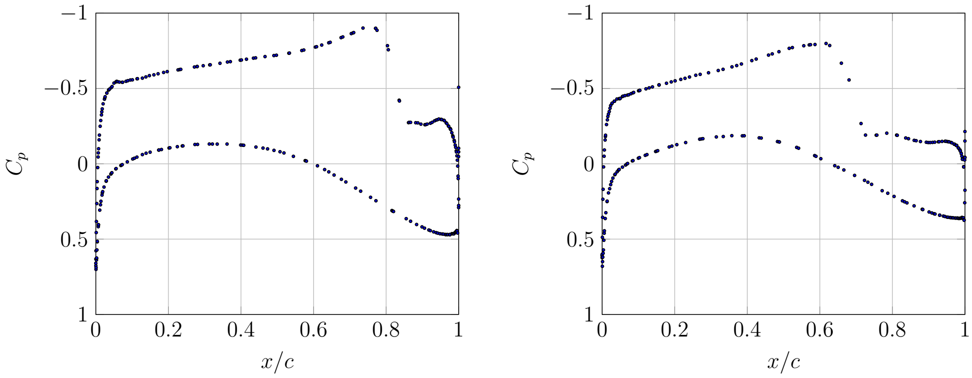
(c) 28.3% Span

(d) 50.2% Span



(e) 60.3% Span

(f) 72.7% Span



(g) 84.6% Span

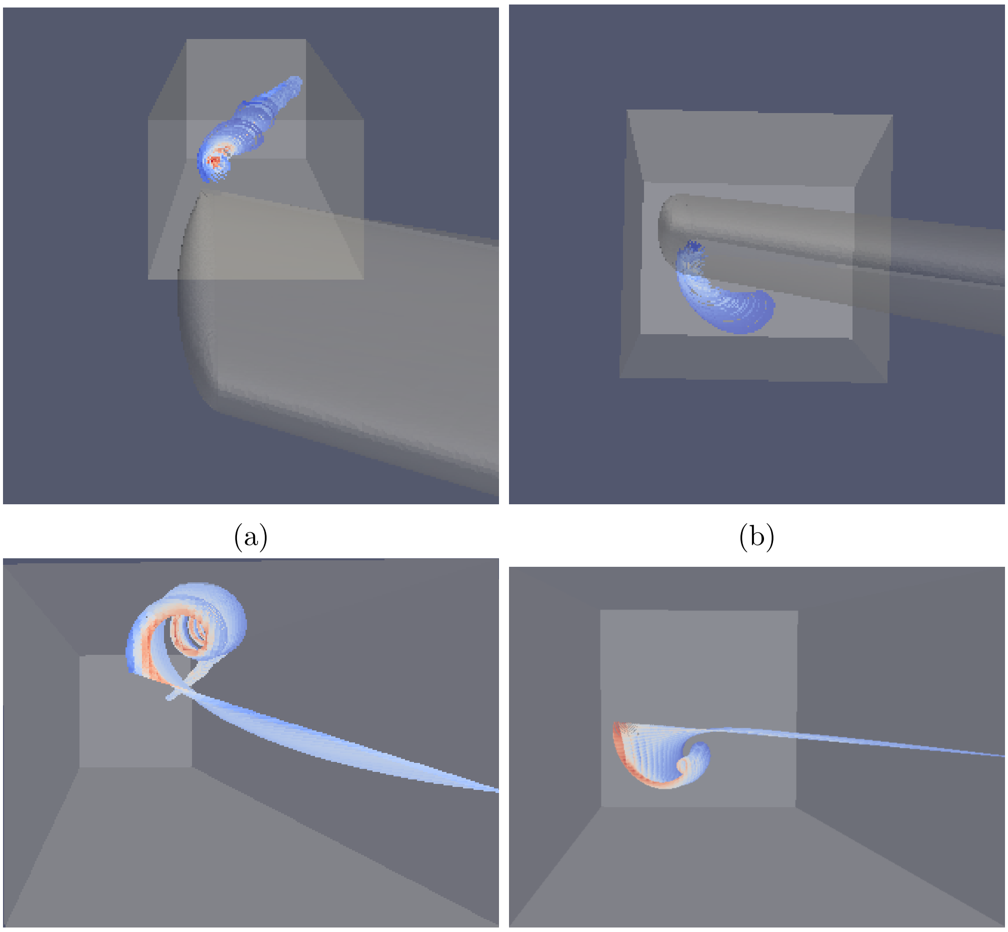
(h) 95.0% Span

Figure 5.17: Graphs plotting C_p as a function of the local chord length at different span-wise stations along the wing for M=0.85 and $C_L = 0.5$.

The position of the shocks on the wing and tail indicate that the wing is producing an upward force whilst the tail produces a downward force. This is also indicated by the counter-rotation of the resultant vortices as shown in Figure 5.18. More precisely, the x_1-vorticity is defined as the x_1 component of the curl of the velocity field

$$\omega_1 = (\nabla \times \mathbf{u}) \cdot \mathbf{x_1} \tag{5.8}$$

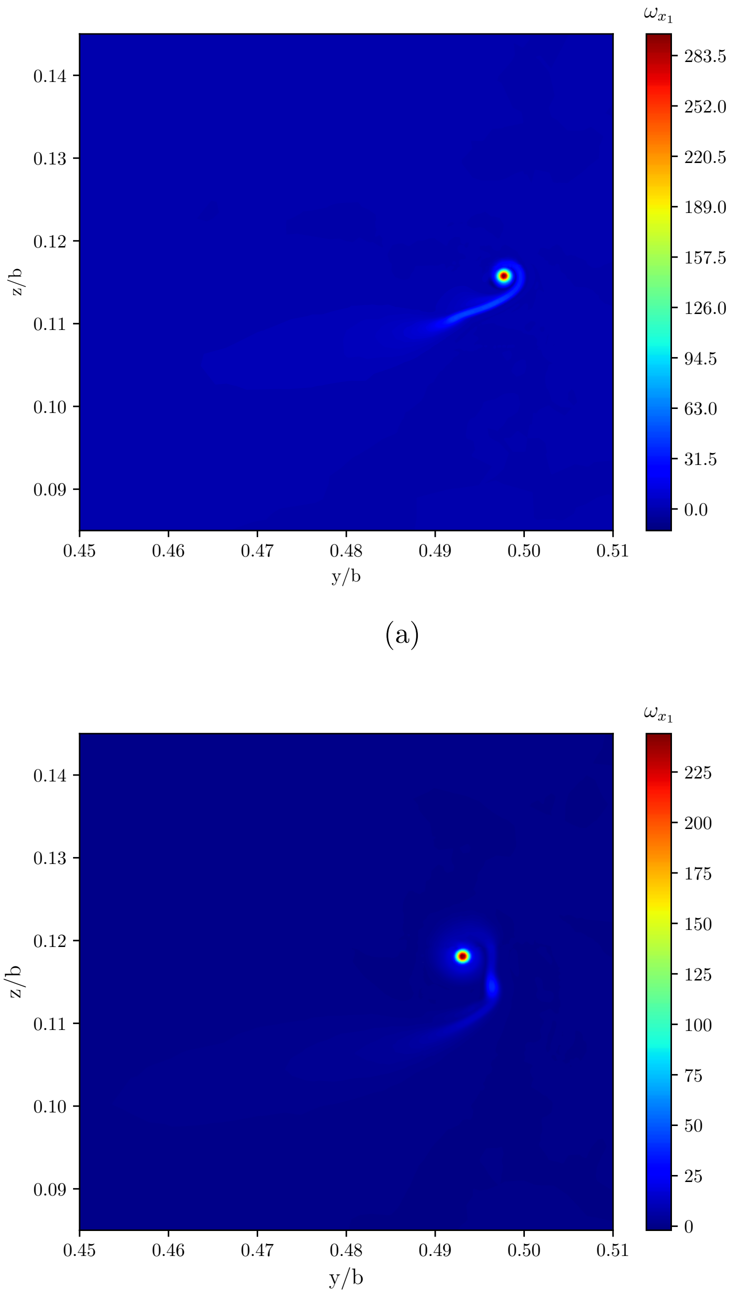
where $\mathbf{x_1}$ is a unit vector in the positive x_1 direction (direction of flow). Then the vortex shed from the wing has positive x_1-vorticity whilst that of the tail is negative. The counter-rotation results in the wing's vortex spiralling upwards and in-board as it is propagated downstream and the tail's vortex spiralling downwards and in-board as shown by the vortex streamlines (coloured with vorticity) in Figure 5.18.


(a) (b)

(c) (d)

Figure 5.18: Figures illustrating the vortex streamlines (colour depicts vorticity). (a) Wing vortex. (b) Tail vortex. (c) Roll-up of the vortex sheet being shed from the wing. (d) Roll-up of the vortex sheet being shed from the tail.

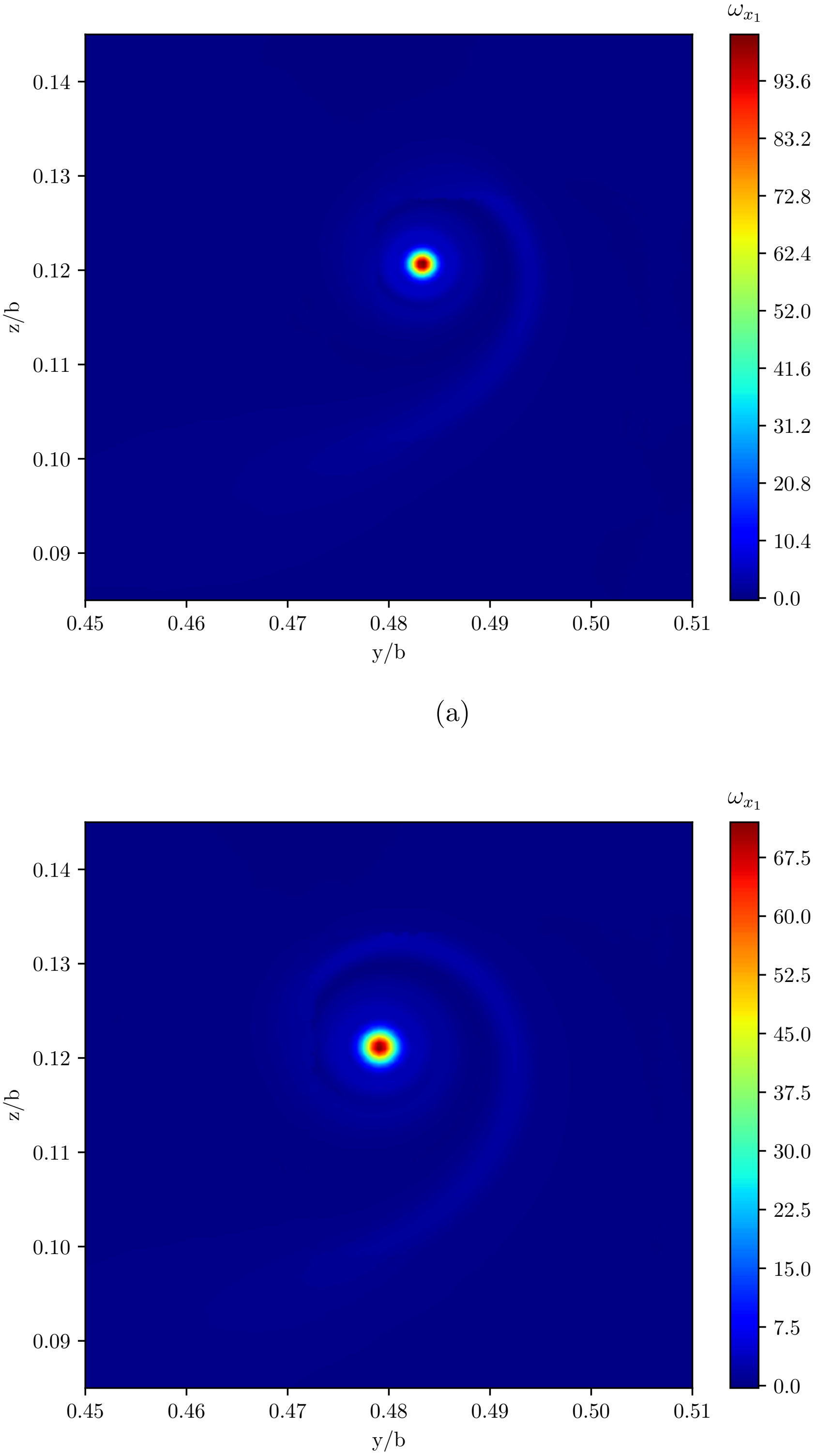
The development of the vortices up to 3 c_{ref} downstream of the wing-tip is more clearly seen in the cross-sections of the x_1-vorticity that are shown in Figure 5.19 to Figure 5.21. In the case of the wing, Figure 5.19b shows that a secondary vorticity peak is developed due to the vortex sheet rolling up on itself. This peak orbits the centre of the vortex before exiting the bounds of the wing's vortex box.



(a)

(b)

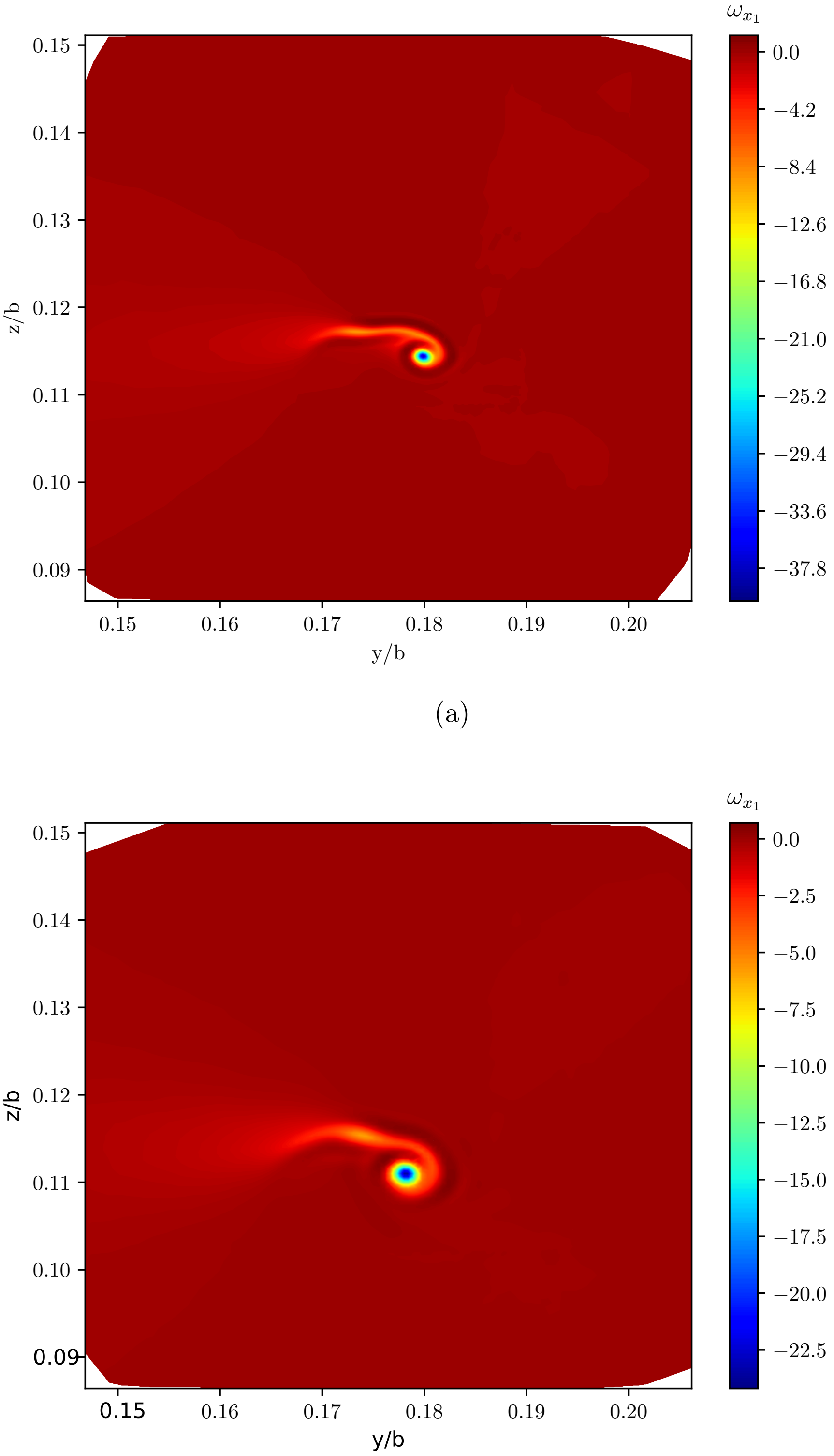
Figure 5.19: Contour plots of the non-dimensional x_1-vorticity downstream of the wing-tip showing development of second suction peak. (a) 0.15 c_{ref} downstream of the wing-tip (b) 0.55 c_{ref} downstream of the wing-tip.



(a)

(b)

Figure 5.20: Contour plots of the non-dimensional x_1-vorticity up to 3 reference chords downstream of the wing-tip. (a) 2.15 c_{ref} downstream of the wing-tip. (b) 3.15 c_{ref} downstream of the wing-tip.



(a)

(b)

Figure 5.21: Contour plot of x_1-vorticity downstream of the tail-tip. (a) 0.5 c_{ref} downstream of the tail. (b) 1 c_{ref} downstream of the tail.

The movement of the wing vortex brings it in close vicinity to the bounds of the vortex box as shown by Figure 5.22. The depicted $x_2 - x_3$

plane slices of the mesh show the vortex close to the boundary of the wing vortex box (circa 0.3 $_{ref}$ downstream of the wing-tip). The vortex sheet first exits the box at the top boundary and then enters again in a smooth and relatively undisturbed manner. This is also apparent from the next image which shows the vortex intersecting the hybrid mesh interface in a smooth manner. This occurs despite the mesh topology rapidly changing from tetrahedral to hex elements which demonstrates the efficacy of the proposed hybrid mesh interface methodology.

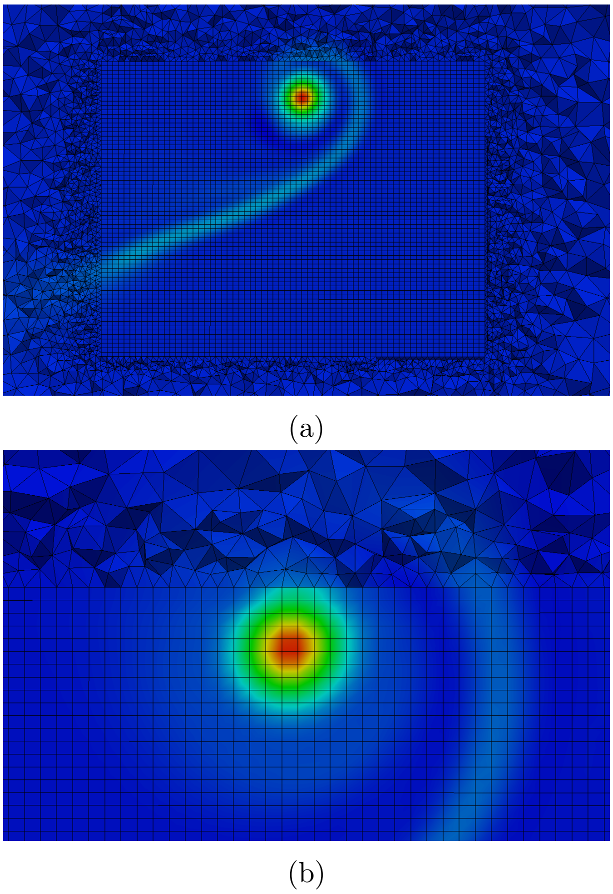
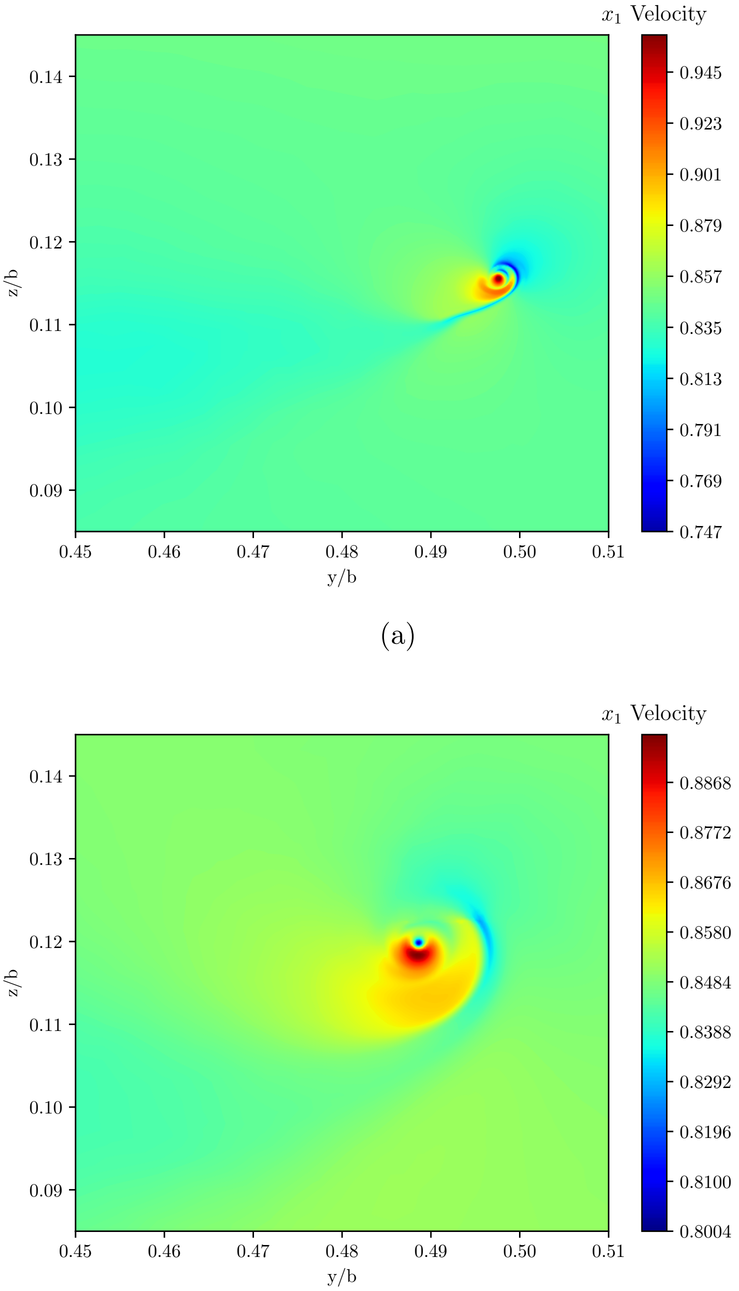


Figure 5.22: Contour plots of x_1-vorticity close behind the wing-tip with the hybrid mesh overlayed on top. (a) 0.15 c_{ref} downstream of the wing-tip. (b) 0.3 c_{ref} downstream of the wing-tip.

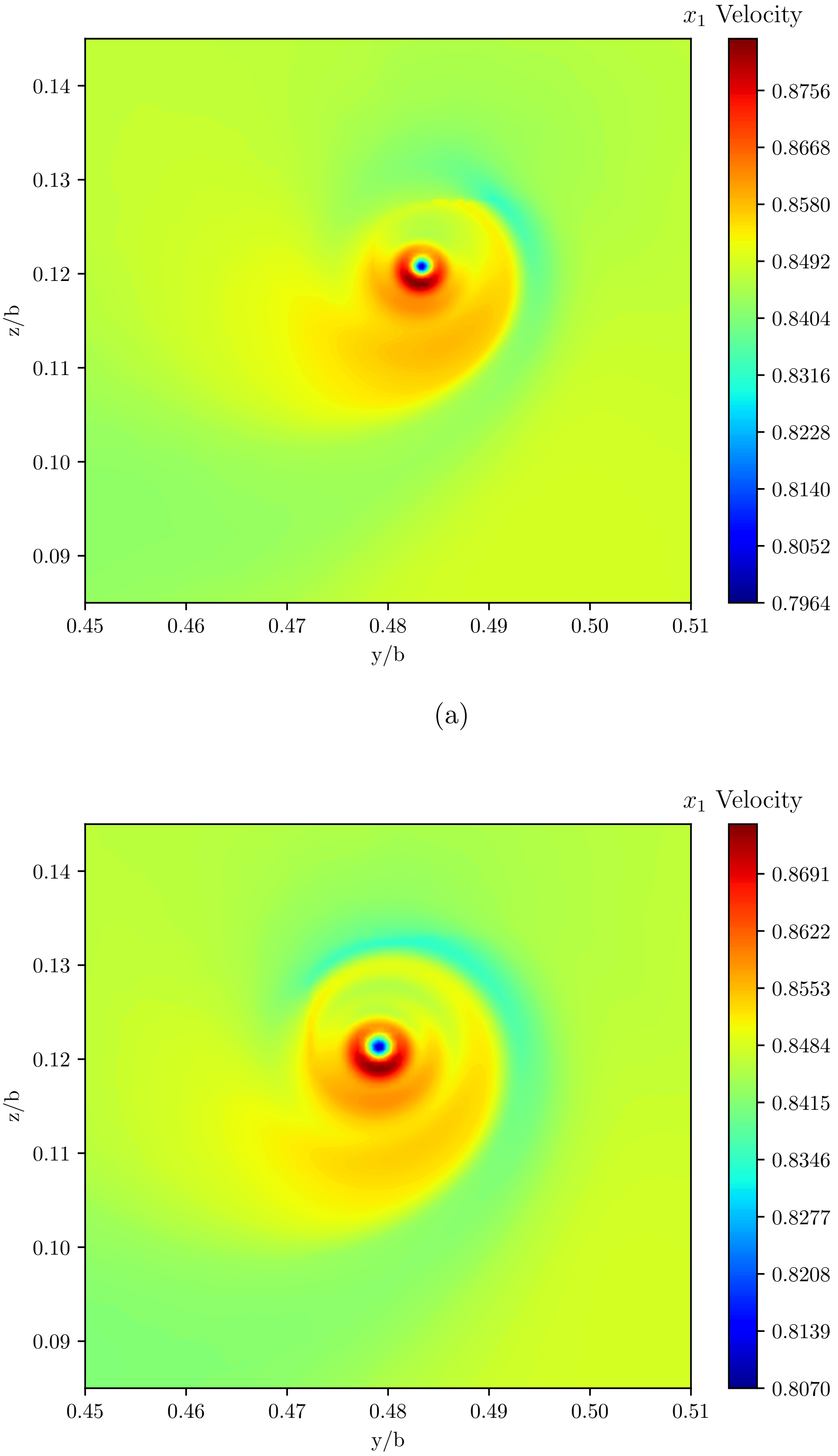
In Chapter 1, it is mentioned that the analytical methods assume no variation in vortex axial velocity. This is however not the case in practice, as shown in the cross-sections of the x_1-velocity for the wing and tail vortices in Figure 5.23 to Figure 5.25. In both cases, there is a large reduction in the axial velocity at the vortex centre compared to that of the free-stream as it leaves the trailing edge. For the tail vortex, there is a gradual recovery back to the free-stream velocity as one moves away from the vortex centre. However, for the wing there is a region where the flow is larger than that of the free-stream before returning to the free-stream value. By the time the vortices near the end of the boxes (3 c_{ref} behind the wing-tip for wing vortex and 1 c_{ref} behind the tail-tip for the tail vortex), the axial flow at their respective centres is circa 6% lower than that of the free-stream. However, the gradients in axial velocity remains high, particularly in near the vortex core. Also noteworthy is the smoothness of the flow field, which again underlines the efficacy of the developed interface condition.



(a)

(b)

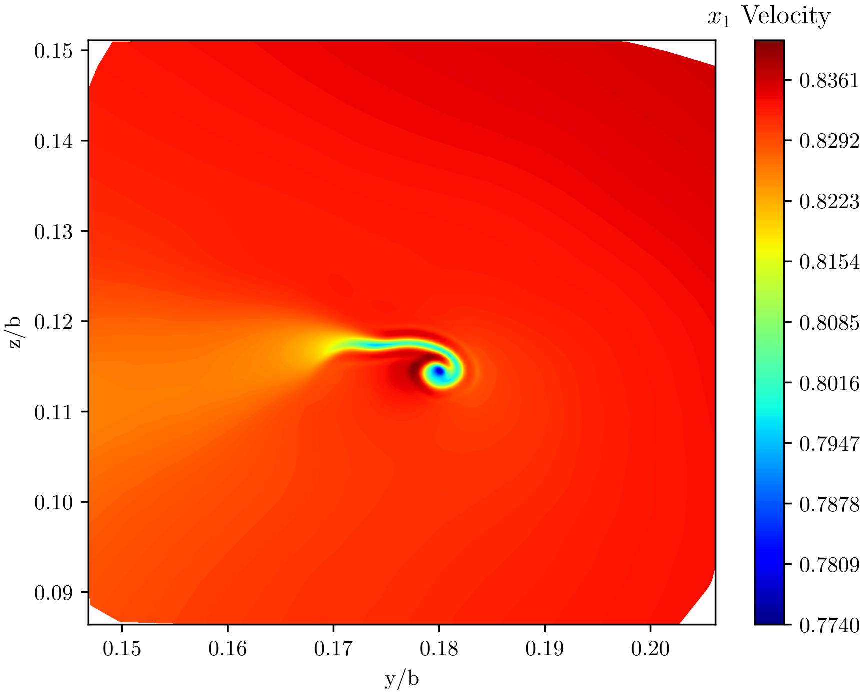
Figure 5.23: Contour plots of the non-dimensional x_1-velocity downstream of the wing-tip showing development of second suction peak. (a) 0.15 c_{ref} downstream of the wing-tip (b) 1.15 c_{ref} behind the wing-tip.



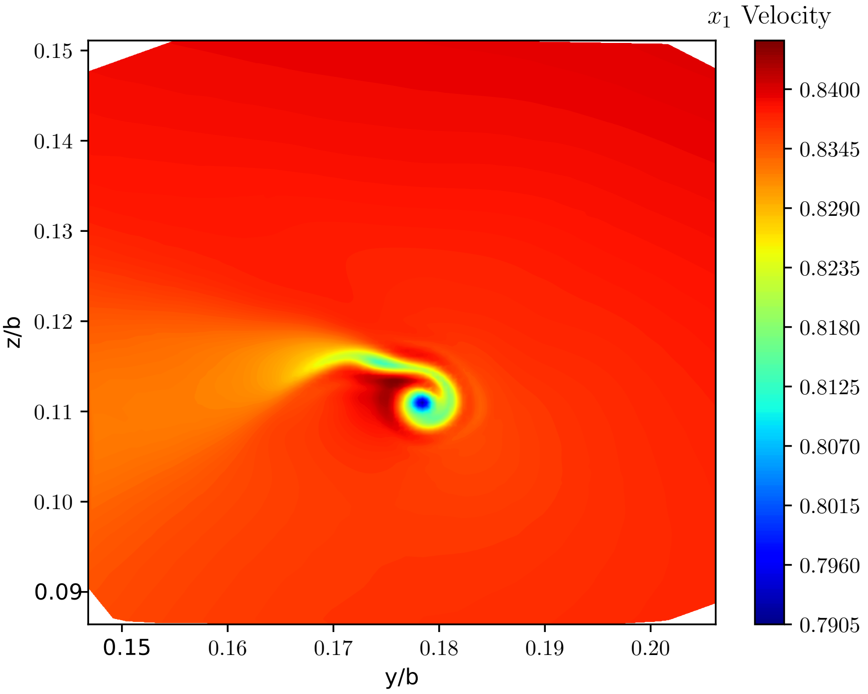
(a)

(b)

Figure 5.24: Contour plots of the non-dimensional x_1-velocity up to 3 reference chords downstream of the wing-tip. (a) 2.15 c_{ref} behind the wing-tip. (b) 3.15 c_{ref} behind the wing-tip.



(a)



(b)

Figure 5.25: Contour plot of x_1-velocity downstream of the tail-tip. (a) 0.5 c_{ref} downstream of the tail. (b) 1 c_{ref} downstream of the tail.

The above simulation neglected the effects of viscosity. This resulted in the shock shifting towards the trailing edge due to the lack of boundary-layer-shock interaction. Since the load distribution dictates the formation of the vortices, differences in vortex strength and position may be expected for real viscous flows. However, the actual roll-up of the vortex sheet is mainly an inviscid process [4].

Computational Performance

Performing the above simulation required a large amount of CPU hours. This was attributed to two things: large meshes and restrictive time step size due to small element sizes in the unstructured mesh. The former is due to the fact that the 2^{nd} order nature of the FVM requires small mesh spacings to resolve the flow around the trailing edges and that the interfaces are required to have co-located nodes for the current SBP-SAT implementation. This coupled with the close proximity of the baffles to the trailing edges forces the spacings on the baffle near the trailing edge to be small. This mitigates the potential node count savings expected from using the higher order FDM.

The latter coupled with the use of the minimum global time step size, as mentioned in Chapter 4, contributed to the high CPU hours required to reach the convergence criteria. The continued use when considering larger stream-wise distances or boundary layers would make the explicit solution procedure restrictive. This may be somewhat mitigated using local time-stepping which at the time of writing was successfully tested on 2D gust-airfoil interactions. However, implicit solution methods may reduce the computational expense further especially when boundary layers are being considered.

Chapter 6

Concluding Remarks and Future Work

This project has developed a hybrid finite-volume-finite-difference CFD solver. For the first time, such a solver has been applied to investigating the 3D formation of aircraft trailing vortices under transonic flow conditions using hybrid meshes. Instead of creating a single hybrid code, the SBP-SAT framework was used to couple the finite volume code *Elemental*® to the finite difference code *Essense* . This was done by first implementing SAT boundary conditions into *Elemental*® and the development of a special parallel interface library. The library facilitates the identification of co-located nodes, data communication and time synchronisation between the two codes. The resultant hybrid solver was then validated against two analytical solutions: the propagating isentropic vortex and 3D manufactured solution to the Navier-Stokes equations. In doing so, the interface implementation was shown not to impact upon the expected order of accuracy even when higher order FD operators were coupled to the FVM.

The solver successfully captured the formation of the wing and tail vortices shed from the NASA CRM geometry under transonic cruise conditions. Mesh refinment studies were performed to ensure that the solutions for the flow around the aircraft and development of the vortices were in the asymptotic region. The vortices shed from the wing and tail were propagated from their respective tips up to 3 c_{ref} behind the wing-tip. Over this distance, the movement of the wing vortex caused it to partially intersect a hybrid-interface boundary. Despite this, the solution remained smooth which demonstrated the success of the SBP-SAT interface treatment. In contrast to analytical methods, large variations in axial velocity along the vortex core were visible for both the wing and tail. At 3 c_{ref} behind the wing-tip, both vortices had a 6 % deficit in the axial velocity of the core when compared to the free-stream. Such a high fidelity result remains widely unreported for trailing vortices shed from an industrially relevant wing-fuselage geometry under transonic flow conditions.

6.1 Recommendations for Future Work

In light of the results presented in Chapter 5, the following recommendations are made for future work:

1. Use of SBP preserving interpolation operators to allow for non-colocated nodes on interface boundaries.
2. In conjunction with the above, automated mesh adaptation to capture the vortices.
3. Implicit solution methods to speed up solve time.
4. Extension of the domain in the stream-wise direction as shown in Figure 6.1 to model actual formation flight.

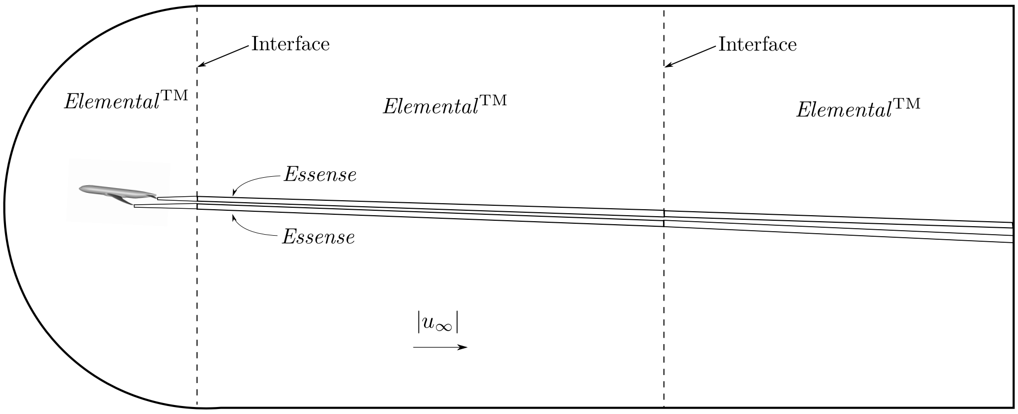


Figure 6.1: Schematic illustrating how the domain of interest may be extended in the stream-wise direction using interface boundaries to model formation flight.

References

[1] V. J. Rossow, Inviscid Modeling of Aircraft Trailing Vortices. In *NASA Symposium on Wake Vortex Minimization.* pp. 9–59, 1976.

[2] Airbus, Global Market Forecast 2015, 27, 2015.

[3] Boeing, 2015. Current Market Outlook 2015-2034.

[4] C. duP. Donaldson and A. J. Bilanin, Vortex Wakes of Conventional Aircraft. Technical report, Advisory Group for Aerospace Research and Development, Princeton, 1975.

[5] P. R. Spalart, Airplane Trailing Vortices. *Annual Review of Fluid Mechanics*, 30, 107–138, 1998.

[6] V. J. Rossow, Lift-generated vortex wakes of subsonic transport aircraft. *Progress in Aerospace Sciences*, 35(6), 507–660, 1999.

[7] T. Gerz, F. Holzäpfel, and D. Darracq, Commercial aircraft wake vortices. *Progress in Aerospace Sciences*, 38, 181–208, 2002.

[8] A. Betz, Behaviour of Vortex Systems. *NACA-TM-713*, 1933.

[9] S. Crow, Stability theory for a pair of trailing vortices. *AIAA Journal*, 8(12), 2172–2179, 1970.

[10] G. C. Greene, An approximate model of vortex decay in the atmosphere. *Journal of Aircraft*, 23(7), 566–573, 1986.

[11] J. D. Anderson Jr, 2010. *Fundamentals of aerodynamics.* Tata McGraw-Hill Education.

[12] W. Tipping Woods, *Wing Trailing Vortex Paths in Formation Flight.* Master's thesis, University of Cape Town, 2015.

[13] S. A. Ning, T. C. Flanzer, and I. M. Kroo, Aerodynamic Performance of Extended Formation Flight. *Journal of Aircraft*, 48(3), 855–865, 2011.

[14] J. D. Crouch, Instability and transient growth for two trailing-vortex pairs. *Journal of Fluid Mechanics*, 350, 311–330, 1997.

[15] W. B. Blake and D. R. Gingras, Comparison of predicted and measured formation flight interference effects. In *AIAA Atmospheric Flight Mechanics Conference*. DTIC Document, Montreal: American Institute of Aeronautics and Astronautics, 2001.

[16] T. Gerz and F. Holzaring, Wing-Tip Vortices, Turbulence, and the Distribution of Emissions. *AIAA Journal*, 37(10), 1270–1276, 1999.

[17] S. A. Ning and I. Kroo, Compressibility Effects of Extended Formation Flight. *29th AIAA Applied Aerodynamics Conference*, 2011.

[18] M. M. Rai, Navier-Stokes Simulations of Blade-Vortex Interaction Using High-Order Accurate Upwind Schemes. In J. C. Hardin and M. Y. Hussaini (Eds.), *Computational Aeroacoustics*. Springer, New York, NY, pp. 417–430, 1993.

[19] A. W. Moore and P. G. Saffman, Axial flow in laminar trailing vortices. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 333, 491–508, 1973.

[20] S. C. Crow and E. R. Bate, Lifespan of trailing vortices in a turbulent atmosphere. *Journal of Aircraft*, 13(7), 476–482, 1976.

[21] R. Sevilla, O. Hassan, and K. Morgan, An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows. *Computer Methods in Applied Mechanics and Engineering*, 253, 15–27, 2013.

[22] B. Wake and D. Choi, Investigation of high-order upwinded differencing for vortex convection. *AIAA journal*, 34(2), 2–7, 1996.

[23] D. C. Del Rey Fernández, J. E. Hicken, and D. W. Zingg, Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Computers & Fluids*, 95, 171–196, 2014.

[24] J. Nordström, J. Gong, E. van der Weide, et al., A stable and conservative high order multi-block method for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 228(24), 9020–9035, 2009.

[25] J. Gong and J. Nordström, Interface procedures for finite difference approximations of the advection-diffusion equation. *Journal of Computational and Applied Mathematics*, 236(5), 602–620, 2011.

[26] J. Nordström and J. Gong, A stable and efficient hybrid method for aeroacoustic sound generation and propagation. *Comptes Rendus Mécanique*, 333(9), 713–718, 2005.

[27] J. Nordström and J. Gong, A stable hybrid method for hyperbolic problems. *Journal of Computational Physics*, 212(2), 436–453, 2006.

[28] J. Gong and J. Nordström, A stable and efficient hybrid scheme for viscous problems in complex geometries. *Journal of Computational Physics*, 226(2), 1291–1309, 2007.

[29] J. Nordström, F. Ham, M. Shoeybi, et al., A hybrid method for unsteady inviscid fluid flow. *Computers & Fluids*, 38(4), 875–882, 2009.

[30] M. Svärd, M. H. Carpenter, and J. Nordström, A stable high-order finite difference scheme for the compressible Navier-Stokes equations, far-field boundary conditions. *Journal of Computational Physics*, 225(1), 1020–1038, 2007.

[31] M. H. Carpenter, J. Nordstrom, and D. Gottlieb, A stable and conservative interface treatment of arbitrary spatial accuracy. *Journal of Computational Physics*, 148, 341–365, 1999.

[32] M. Svärd and J. Nordström, Review of summation-by-parts schemes for initial-boundary-value problems. *Journal of Computational Physics*, 268, 17–38, 2014.

[33] Y. Zhao and A. Forhad, A general method for simulation of fluid flows with moving and compliant boundaries on unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 192(39-40), 4439–4466, 2003.

[34] A. G. B. Mowat, A. G. Malan, L. H. van Zyl, et al., Hybrid Finite-Volume Reduced-Order Model Method for Nonlinear Aeroelastic Modeling. *Journal of Aircraft*, 51(6), 1805–1812, 2014.

[35] E. F. Toro, M. Spruce, and W. Speares, Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1), 25–34, 1994.

[36] P. Batten, M. A. Leschziner, and U. Goldberg, Average-State Jacobians and Implicit Methods for Compressible Viscous and Turbulent Flows. *Journal of Computational Physics*, 137(1), 38–78, 1997.

[37] R. W. Lewis and A. G. Malan, Continuum thermodynamic modeling of drying capillary particulate materials via an edge-based algorithm. *Computer methods in applied mechanics and engineering*, 194(18), 2043–2057, 2005.

[38] P. Roe, Approximate Riemann Solvers, Parameter Vectors and Difference Schemes. *Journal of Computational Physics*, 43(2), 357–372, 1981.

[39] J. Nordström, K. Forsberg, C. Adamsson, et al., Finite volume methods, unstructured meshes and strict stability for hyperbolic problems. *Applied Numerical Mathematics*, 45(4), 453–473, 2003.

[40] M. Svärd and J. Nordström, A stable high-order finite difference scheme for the compressible Navier-Stokes equations No-slip wall boundary conditions. *Journal of Computational Physics*, 227(10), 4805–4824, 2008.

[41] J. Blazek, Computational Fluid Dynamics: Principles and Applications. *Computational Fluid Dynamics: Principles and Applications*, 2001.

[42] G. Karypis and V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1), 359–392, 1998.

[43] J. Bonet and J. Peraire, An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems. *International Journal for Numerical Methods in Engineering*, 31(1), 1–17, 1991.

[44] B. Einarsson, R. Hanson, and T. Hopkins, Standardized mixed language programming for Fortran and C. *ACM SIGPLAN Fortran Forum*, 1–16, 2009.

[45] J. Vassberg, M. Dehaan, M. Rivers, et al., Development of a Common Research Model for Applied CFD Validation Studies. *26th AIAA Applied Aerodynamics Conference*, 2008.

[46] Q. Abbas and J. Nordström, Weak Versus Strong No-Slip Boundary Conditions for the Navier-Stokes Equations. *Engineering Applications of Computational Fluid Mechanics*, 1(4), 29–38, 2010.

[47] P. Eliasson, S. Eriksson, and J. Nordström, The influence of weak and strong solid wall boundary conditions on the convergence to steady-state of the navier-stokes equations. *AIAA Paper*, 3551, 2009, 2009.

[48] P. J. Roache, Quantification of Uncertainty in Computational Fluid Dynamics. *Annual Review of Fluid Mechanics*, 29(1), 123–160, 1997.

[49] R. Ulerich, K. C. Estacio-Hiroms, N. Malaya, et al., A Transient Manufactured Solution for the Compressible Navier-Stokes Equations With a Power Law Viscosity. *10th World Congress on Computational Mechanics*, 1–16, 2012.

Appendix A

Stability of the Advection Diffusion Equation

A.1 Continuous Problem

The problem in the continuous setting is

$$\begin{aligned} u_t + au_x &= \epsilon u_{xx} && 0 \leq x \leq 1 \\ \alpha u + \beta u_x &= g_0 && \text{at } x = 0 \\ \psi u + \theta u_x &= g_1 && \text{at } x = 1 \\ u(x,0) &= f(x) \end{aligned} \tag{A.1}$$

Conditions for α, β, ψ and θ will be derived such that the boundary conditions given in Equation (A.1) lead to a well-posed problem.

A.2 Stability

An energy estimate was obtained by multiplying by u and integrating over the domain.

$$\begin{aligned} \int_x uu_t dx &= -a\int_x uu_x dx + \epsilon \int_x uu_{xx} dx \\ \frac{1}{2}\int_x \left(u^2\right)_t dx &= -\frac{a}{2}\int_x \left(u^2\right)_x dx + \epsilon\left(uu_x|_0^1 - \int_x u_x u_x dx\right) \\ \frac{d}{dt}\int_x u^2 dx &= \left[-au^2 + 2\epsilon uu_x\right]_0^1 - 2\epsilon\int_x \left(u_x\right)^2 dx \end{aligned} \tag{A.2}$$

It was assumed that $a > 0$ and $\epsilon > 0$ and then the boundary conditions in Equation (A.1) were inserted into Equation (A.2) which yielded

$$\begin{aligned}\frac{d}{dt}||u||^2 &= \left[-au^2 + 2\epsilon u u_x + 2\psi u^2 + 2\theta u u_x - 2ug_1\right]_1 \\ &\quad + \left[au^2 - 2\epsilon u u_x + 2\alpha u^2 + 2\beta u u_x - 2ug_0\right]_0 - 2\epsilon \int_x (u_x)^2\, dx \\ &= \left[(-a + 2\psi)u^2 + 2(\theta + \epsilon)u u_x - 2ug_1\right]_1 \\ &\quad + \left[(a + 2\alpha)u^2 + 2(\beta - \epsilon)u u_x - 2ug_0\right]_0 - 2\epsilon \int_x (u_x)^2\, dx \end{aligned} \tag{A.3}$$

In Equation (A.3) $||u||^2 = \int_x u^2 dx$. Let $\beta = \epsilon$ and $\theta = -\epsilon$ then the energy estimate becomes

$$\begin{aligned}\frac{d}{dt}||u||^2 &= \left[\frac{1}{-a + 2\psi}\left((-a + 2\psi)u - g_1\right)^2 - \frac{g_1^2}{-a + 2\psi}\right]_1 \\ &\quad + \left[\frac{1}{a + 2\alpha}\left((a + 2\alpha)u - g_0\right)^2 - \frac{g_0^2}{a + 2\alpha}\right]_0 \\ &\quad - 2\epsilon \int_x (u_x)^2\, dx \end{aligned} \tag{A.4}$$

In order for Equation (A.4) to be bounded then $a+2\alpha \leq 0$ and $-a+2\psi \leq 0$. This results in $\alpha \leq -\frac{a}{2}$ and $\psi \leq \frac{a}{2}$. The proposed boundary conditions are then written as

$$\begin{aligned} \alpha u + \epsilon u_x &= g_0 \qquad \text{at } x = 0 \\ \psi u - \epsilon u_x &= g_1 \qquad \text{at } x = 1 \end{aligned} \tag{A.5}$$

The above boundary conditions result in an energy estimate of

$$\frac{d}{dt}||u||^2 \leq -\frac{g_1^2}{-a + 2\psi} - \frac{g_0^2}{a + 2\alpha} \tag{A.6}$$

A.3 Discretisation

Let there be N+1 nodes in the domain x. The semi-discrete version of Equation (A.1) with SAT terms is then

$$\begin{aligned}\mathbf{u}_t + aP^{-1}Q\mathbf{u} =& \epsilon(P^{-1}Q)(P^{-1}Q)\mathbf{u} + \alpha_0 P^{-1}(\alpha \mathbf{u}_0 + \epsilon(D\mathbf{u})_0 - g_0)\mathbf{e}_0 \\ &+ \alpha_1 P^{-1}(\psi \mathbf{u}_N - \epsilon(D\mathbf{u})_N - g_1)\mathbf{e}_1 \end{aligned} \tag{A.7}$$

u is a column vector that holds the discrete values of u at the $N+1$ nodes. $\mathbf{e}_0$ is a column vector that chooses the node at $x = 0$, and $\mathbf{e}_1$; the node at

$x = 1$. P is an $(N+1)$x$(N+1)$ diagonal matrix that holds the volume, V_i, of each node. Q is the $(N+1)$x$(N+1)$ discretisation matrix and is an SBP operator i.e. $Q + Q^T = B = diag(-1\ \ 0 \ldots 1)$.

A.4 Energy Estimate of Discretised Equations

The SBP property that $Q + Q^T = B$ was used to calculate the energy estimate of the discrete system as

$$\begin{aligned}
\frac{d}{dt}(\mathbf{u}^T P \mathbf{u}) =& \mathbf{u}^T P \mathbf{u}_t + \mathbf{u}_t^T P \mathbf{u} \\
=& -a\mathbf{u}^T B \mathbf{u} + 2\epsilon \mathbf{u}^T B (P^{-1}Q)\mathbf{u} - 2\epsilon \mathbf{u}^T (P^{-1}Q)^T P (P^{-1}Q)\mathbf{u} \\
& + 2\alpha_0 \mathbf{u}_0 (\alpha \mathbf{u}_0 + \epsilon (D\mathbf{u})_0 - g_0) + 2\alpha_1 \mathbf{u}_N (\psi \mathbf{u} - \epsilon D\mathbf{u})_N - g_1) \\
=& (a + 2\alpha_0 \alpha)\mathbf{u}_0^2 + (-a + 2\alpha_1 \psi)\mathbf{u}_N^2 + 2\epsilon(-1 + \alpha_0)\mathbf{u}_0 \mathbf{u}_{x_0} \\
& + 2\epsilon(1 - \alpha_1)\mathbf{u}_N \mathbf{u}_{x_N} - 2\alpha_0 \mathbf{u}_0 g_0 - 2\alpha_1 \mathbf{u}_N g_1 - 2\epsilon \sum_{i=0}^{N} \mathbf{u}_{x_i} V_i \mathbf{u}_{x_i}
\end{aligned} \tag{A.8}$$

If $\alpha_0 = \alpha_1 = 1$ then Equation (A.8) becomes

$$\begin{aligned}
\frac{d}{dt}(\mathbf{u}^T P \mathbf{u}) =& (a + 2\alpha)\mathbf{u}_0^2 - 2\mathbf{u}_0 g_0 + (-a + 2\psi)\mathbf{u}_N^2 - 2\mathbf{u}_N g_1 - 2\epsilon \sum_{i=0}^{N} \mathbf{u}_{x_i} V_i \mathbf{u}_{x_i} \\
=& \frac{1}{-a + 2\psi}\left((-a + 2\psi)\mathbf{u}_N - g_1\right)^2 - \frac{g_1^2}{-a + 2\psi} \\
& + \frac{1}{a + 2\alpha}\left((a + 2\alpha)\mathbf{u}_0 - g_0\right)^2 - \frac{g_0^2}{a + 2\alpha} - 2\epsilon \sum_{i=0}^{N} \mathbf{u}_{x_i} V_i \mathbf{u}_{x_i}
\end{aligned} \tag{A.9}$$

It is seen from Equation (A.9) that the discrete energy estimate is equivalent to the continuous estimate in Equation (A.4). The boundary conditions proposed in Equation (A.5) along with $\alpha_0 = \alpha_1 = 1$ will result in a stable numerical scheme.

Appendix B

Manufactured Solutions Source Terms

The full expression resulting from inserting the analytical solutions into the governing equations are tremendous and seemingly unmanageable. There are many terms to calculate which gives rise to equally as many opportunities for algebraic mistakes to creep in. As such Ulerich et al [49] developed an automated way to calculate the forcing functions. The use of a symbolic math package such as SymPy provides an effective way to calculate the derivatives needed to evaluate the forcing functions. The code generation functions in SymPy write implementation files for the required derivatives. These are then imported and used to build the forcing function exactly as it is written in its symbolic form. The symbolic form of the inviscid source terms are

$$
\begin{aligned}
(F_1)_{x_1} &= \begin{bmatrix} \rho_{x_1} u + \rho u_{x_1} \\ \rho_{x_1} u^2 + 2\rho u u_{x_1} + p_{x_1} - (\tau_{x_1 x_1})_{x_1} \\ \rho_{x_1} v u + \rho(v_{x_1} u + v u_{x_1}) - (\tau_{x_2 x_1})_{x_1} \\ \rho_{x_1} w u + \rho(w_{x_1} u + w u_{x_1}) - (\tau_{x_3 x_1})_{x_1} \\ \rho_{x_1} H + \rho H_{x_1} - u(\tau_{x_1 x_1})_{x_1} - v_{x_1}\tau_{x_2 x_1} - v(\tau_{x_2 x_1})_{x_1} - w_{x_1}\tau_{x_3 x_1} - w(\tau_{x_3 x_1})_{x_1} + q_{x_1} \end{bmatrix} \\
(F_2)_{x_2} &= \begin{bmatrix} \rho_{x_2} v + \rho v_{x_2} \\ \rho_{x_2} u v + \rho(u_{x_2} v + u v_{x_2}) - (\tau_{xy})_{x_2} \\ \rho_{x_2} v^2 + 2\rho v v_{x_2} + p_{x_2} - (\tau_{x_2 x_2})_{x_2} \\ \rho_{x_2} w v + \rho(w_{x_2} v + w v_{x_2}) - (\tau_{zy})_{x_2} \\ \rho_{x_2} H + \rho H_{x_2} - u(\tau_{xy})_{x_2} - v_{x_2}\tau_{x_2 x_2} - v(\tau_{x_2 x_2})_{x_2} - w_{x_2}\tau_{zy} - w(\tau_{zy})_{x_2} + q_{x_2} \end{bmatrix} \\
(F_3)_{x_3} &= \begin{bmatrix} \rho_{x_3} w + \rho w_{x_3} \\ \rho_{x_3} u w + \rho(u_{x_3} w + u w_{x_3}) - (\tau_{x_1 x_3})_{x_3} \\ \rho_{x_1} v w + \rho(v_{x_3} w + v w_{x_3}) - (\tau_{x_2 x_3})_{x_3} \\ \rho_{x_1} w^2 + 2\rho w w_{x_3} + p_{x_3} - (\tau_{x_3 x_3})_{x_3} \\ \rho_{x_3} H + \rho H_{x_3} - u(\tau_{x_1 x_3})_{x_3} - v_{x_3}\tau_{x_2 x_3} - v(\tau_{x_2 x_3})_{x_3} - w_{x_1}\tau_{x_3 x_3} - w(\tau_{x_3 x_3})_{x_3} + q_{x_3} \end{bmatrix}
\end{aligned}
\tag{B.1}
$$

The viscous terms, assuming constant viscosity, are

$$
\begin{aligned}
(\tau_{x_1 x_1})_{x_1} &= 2\mu u_{x_1 x_1} + \lambda(u_{x_1 x_1} + v_{x_2 x_1} + w_{x_3 x_1}) \\
(\tau_{x_2 x_1})_{x_1} &= \mu(u_{x_2 x_1} + v_{x_1 x_1}) \\
(\tau_{x_3 x_1})_{x_1} &= \mu(w_{x_1 x_1} + u_{x_3 x_1}) \\
(\tau_{xy})_{x_2} &= \mu(u_{x_2 x_2} + v_{xy}) \\
(\tau_{x_2 x_2})_{x_2} &= 2\mu v_{x_2 x_2} + \lambda(u_{xy} + v_{x_2 x_2} + w_{zy}) \\
(\tau_{zy})_{x_2} &= \mu(v_{zy} + w_{x_2 x_2}) \\
(\tau_{x_1 x_3})_{x_3} &= \mu(w_{x_1 x_3} + u_{x_3 x_3}) \\
(\tau_{x_2 x_3})_{x_3} &= \mu(v_{x_3 x_3} + w_{x_2 x_3}) \\
(\tau_{x_3 x_3})_{x_3} &= 2\mu w_{x_3} + \lambda(u_{x_1 x_3} + v_{x_2 x_3} + w_{x_3 x_3}) \\
q_{x_1} &= kT_{x_1} \\
q_{x_2} &= kT_{x_2} \\
q_{x_3} &= kT_{x_3}
\end{aligned}
\tag{B.2}
$$