



University of Sheffield
Department of Mechanical Engineering

Flow Feature Aligned Mesh Generation and Adaptation

M. J. Harris

Supervised by Prof. Ning Qin

April 2013

This thesis is submitted to University of Sheffield in partial fulfilment of the requirement for the degree of Doctor of Philosophy

Abstract

Methods which allow for construction of flow feature aligned meshes in two- and three-dimensions have been developed in this thesis to investigate their potential for improvements in the numerical solution relative to globally refining the mesh. Of particular interest in the work is the generation of high-quality quadrilateral and hexahedral elements aligned with the dominant flow features. The two-dimensional techniques are applied on unstructured quad-dominant meshes, whilst the three-dimensional problems involve embedding high-quality hex-dominant mesh blocks into a hybrid volume mesh to improve their ability to capture anisotropic flow features such as shock waves, trailing shear layers/wakes and wing tip vortices.

A method involving the medial axis has been studied to provide a geometric representation of two-dimensional flow features to allow feature-aligned meshes to be generated. Due to the flexibility of the approach, a range of complex features can be represented as simple geometric entities. These curves are embedded into the domain as virtual geometries to force alignment of unstructured quad-dominant surface mesh elements. The mesh locally mimics the attributes of a structured grid and provides high quality numerical solutions due to the alignment of the cell interfaces with the flow features.

To improve the capability of hybrid meshes to resolve anisotropic flow physics, a method involving the extrusion of quad-dominant surface meshes has been developed. Surface meshes are extruded in the direction of extracted flow features, yielding feature-aligned semi-structured hex-dominant mesh blocks which can be embedded into the hybrid volume mesh. The presence of feature-aligned hexahedra has been shown to greatly enhance the resolution of anisotropic flow features compared with both isotropic and anisotropic tetrahedral elements, due to a significant reduction in numerical diffusion. Furthermore, improvements in the numerical solution have been also been obtained in a more efficient manner than isotropically refining the hybrid mesh. The results indicate that the type, orientation and size of the elements are significant contributing factors in the resolution of the dominant flow features.

Acknowledgements

Firstly I would like to thank Professor Ning Qin for acting as my supervisor for the duration of this project. His advice, support and guidance have been invaluable during my progression through the period of this thesis, and I have come to learn much from our discussions on various CFD related topics.

During the project I frequently visited Aircraft Research Association (ARA) in Bedford, England for a series of short work placements. I would like to express my appreciation to all the members of the Computational Aerodynamics team who provided me with helpful suggestions and technical expertise. Specifically, I would like to thank David Martineau (now of ICON CFD) for serving as my initial industrial supervisor. I am grateful for his assistance (and patience!) in the early stages of the project as I overcame the steep learning curve involved with understanding and using the SOLAR source code. I feel I have developed a significant level of programming capability within a short space of time which would not have been possible without his initial guidance.

I would also like to thank Dr Craig Johnson of ARA for providing me with all the software I required at the beginning of the project and generally being another source of valuable technical expertise in both SOLAR and TAU. Other words of thanks are necessary for Dr Andrew Peace, Head of Computational Aerodynamics at ARA, who (with Craig) took the time to give several drafts of one of my papers a very thorough read. Their detailed feedback was greatly appreciated and I hope greatly increased the quality of the final paper.

Last but not least, I would like to take this opportunity to say thank you to all my family and friends for the constant support given to me as I progressed through this work. In particular I must say a massive thank you to my parents. Without them I would never have got to where I am today. For all the support and encouragement they have given me over the years (not just during the period of this thesis) I am eternally grateful. I hope that I have made them proud through my achievements to date and that this goes some way in repaying them for all they have done for me.

Table of Contents

Abstract	iii
Acknowledgments	iv
Table of Contents	v
Nomenclature	x
1 Introduction	1
1.1 Background	1
1.2 Mesh Classification	2
1.3 Generation of Flow Feature Aligned Meshes	4
1.4 Importance of Mesh Alignment	6
1.5 Aims and Objectives	8
1.6 Thesis Outline	9
2 Literature Review	12
2.1 Introduction	12
2.2 Mesh Adaptation	13
2.2.1 Classifying Mesh Adaptation Techniques	13
2.2.2 A Remark on Error Estimators	14
2.3 Mesh Adaptation: Methods for Feature Alignment	18
2.3.1 Structured Meshes	18
2.3.2 Unstructured Meshes	19
2.3.2.1 Triangular and Tetrahedral Meshes	19
2.3.2.2 Quadrilateral and Hexahedral Meshes	22
2.3.2.3 Cartesian Meshes	26
2.4 Mesh Insertion, Regeneration and Specialized Regeneration Techniques	28
2.5 Miscellaneous Techniques: Overset Grid ('Chimera') Approach	32
2.6 Summary	33
3 Governing Equations and Numerical Methods	36
3.1 Introduction	36

3.2 The Unsteady Navier-Stokes Equations	37
3.3 Preconditioning	41
3.4 Turbulence Modelling	42
3.4.1 Reynolds-Averaged Navier-Stokes (RANS) Simulations	43
3.4.2 Spalart-Allmaras Turbulence Model	44
3.5 Discretisation and Numerical Schemes	46
3.5.1 Spatial Discretisation	46
3.5.2 The Primary Grid	47
3.5.3 The Secondary Grid	50
3.5.4 Flux Discretisation	50
3.5.5 Higher-Order Reconstruction	50
3.5.6 Evaluation of the Gradient	52
3.5.7 Temporal Discretisation	54
3.6 Flow Feature Extraction	54
3.6.1 Shock Waves	55
3.6.1.1 Transonic Flows	55
3.6.1.2 Supersonic Flows	56
3.6.2 Wakes and Wing Tip Vortices	57
3.6.2.1 Steady Wakes	57
3.6.2.2 Wing Tip Vortices	58
3.7 Mesh Adaptation Schemes	59
3.7.1 Construction of Anisotropic Metric for Adaptation	59
3.7.2 Nodal Movement	62
3.7.3 Adaptation in TAU	63
4 Using the Medial Axis to Represent Complex Flow Features for Feature-	
Aligned Unstructured Quad-dominant Surface Mesh Generation	65
4.1 Introduction	65
4.2 Building the Medial Axis	68
4.2.1 Concave Hull Construction	68
4.2.2 Medial Axis Approximation	70
4.2.3 Constrained Delaunay Triangulations	71

4.2.4	Cleaning the Medial Axis	72
4.2.4.1	False Junction Nodes	72
4.2.4.2	Small Branch Removal	73
4.2.5	Splitting the Medial Axis	75
4.2.6	Recombining Medial Axis Branches	76
4.2.7	Fitting Polynomials	78
4.3	Controlling Mesh Spacing	80
4.4	Global Mesh Refinement	83
4.5	Identifying Multiple Regions of Points	85
4.6	Mesh Adaptation	86
4.6.1	Determining Local Mesh Spacing	86
4.7	Splitting Bad Quadrilateral Elements	89
4.8	Test Cases	90
4.8.1	Double Wedge Supersonic Inviscid Flow, $M_\infty = 3.0$	90
4.8.2	NACA0012 Aerofoil	93
4.8.2.1	Inviscid Transonic Flow, $M_\infty = 0.75$, $\alpha = 1.25^\circ$	93
4.8.2.2	Inviscid Transonic Flow, $M_\infty = 0.95$, $\alpha = 0.0^\circ$	96
4.8.2.3	Inviscid Supersonic Flow, $M_\infty = 1.2$, $\alpha = 10.0^\circ$	100
4.8.3	RAE2822 Aerofoil, Transonic Viscous Flow:	
	$M_\infty = 0.730$, $\alpha = 1.25^\circ$, $Re = 6.5 \times 10^6$	102
4.8.4	L1T2 Multi-Element Aerofoil, Subsonic Viscous Flow:	
	$M_\infty = 0.197$, $\alpha = 4.01^\circ$, $Re = 3.52 \times 10^6$	110
4.8.5	Wind Tunnel With Forward Facing Step, Inviscid Supersonic Flow:	
	$M_\infty = 3.0$	115
4.9	Conclusions	120
5	Feature-Aligned Hex-Dominant Mesh Block Insertion: Part I: Shock Waves	121
5.1	Introduction	121
5.2	Generating Shock-Aligned Surface Meshes	122
5.3	Volume Mesh Generation	124
5.3.1	Generating Shock-Aligned Mesh Blocks	125
5.3.1.1	Summary of Process	134
5.4	Test Cases	135

5.4.1 ONERA-M6 Wing, Viscous Transonic Flow:	
$M_\infty = 0.8395, \alpha = 3.06^\circ, Re = 11.72 \times 10^6$	135
5.4.2 DLR-F4, Viscous Transonic Flow:	
$M_\infty = 0.75, \alpha = 2.0^\circ, Re = 3.0 \times 10^6$	145
5.5 Conclusions	148
6 Feature-Aligned Hex-Dominant Mesh Block Insertion: Part II: Wing Tip	
Vortices and Wakes	149
6.1 Introduction	149
6.2 Surface Mesh Extrusion	150
6.3 Defining Source Surfaces	150
6.3.1 Wing Tip Vortices	150
6.3.2 Trailing Wakes	151
6.4 Determining the Extrusion Direction	152
6.5 Mesh Block Growth and Expansion	153
6.6 Merging of Mesh Block into Hybrid Volume Mesh	155
6.7 Test Cases	156
6.7.1 ONERA-M6 Wing, Viscous Transonic Flow:	
$M_\infty = 0.8395, \alpha = 3.06^\circ, Re = 11.72 \times 10^6$	156
6.7.1.1 Wing Tip Vortex Extrusion	156
6.7.1.2 Wing Tip Vortex and Wake Extrusion	159
6.7.2 DLR-F4, Viscous Transonic Flow:	
$M_\infty = 0.75, \alpha = 2.0, Re = 3.0 \times 10^6$	163
6.7.3 Delta Wing, Viscous Transonic Flow:	
$M_\infty = 0.75, \alpha = 15.0, Re = 2.0 \times 10^5$	167
6.8 Conclusions	174
7 Feature-Aligned Hex-Dominant Mesh Block Insertion: Part III: A	
Comparison with Anisotropic Adaptation Techniques	175
7.1 Introduction	175
7.2 Test Cases	175
7.2.1 Delta Wing, Viscous Transonic Flow:	

$M_\infty = 0.75, \alpha = 15.0, \text{Re} = 2.0 \times 10^5$	175
7.2.2 ONERA-M6 Wing, Viscous Transonic Flow:	
$M_\infty = 0.8395, \alpha = 3.06^\circ, \text{Re} = 11.72 \times 10^6$	178
7.2.3 DLR-F4, Viscous Transonic Flow:	
$M_\infty = 0.75, \alpha = 2.0, \text{Re} = 3.0 \times 10^6$	183
7.3 A General Comment on the Reliability of Results	187
7.4 A Discussion on Feature-Alignment and Adaptation Techniques	187
7.5 Conclusions	189
8 Conclusions	190
8.1 Achievements and Concluding Remarks	190
8.2 General Concluding Remarks	192
8.3 Suggestions for Future Work	192
List of Published/Submitted Papers	195
REFERENCES	196

Nomenclature

Latin Letters

Symbol	Quantity
A_i	Area of face i
dS	Surface boundary
c_p	Gas specific heat at constant pressure
c_v	Gas specific heat at constant volume
C_D	Drag coefficient
C_f	Skin friction coefficient
C_L	Lift coefficient
C_p	Surface pressure coefficient
D/Dt	Material derivative, $= \partial/\partial t + u_i \partial/\partial u_i$
e	Total energy per unit mass
E	Total energy
F	Inviscid flux vector
F_i	Inviscid flux vector on face i
G	Viscous flux vector
G_i	Viscous flux vector on face i
h	Total enthalpy per unit mass or thickness
h_{ij}	Hessian matrix
H	Total enthalpy
i	Cell or face notation
κ	Turbulent kinetic energy (per unit mass) or thermal conductivity
k_{ij}	Stiffness of the spring connecting nodes i, j
M_n	Normal Mach number
M_∞	Freestream Mach number
\mathbf{n}	Outward unit normal vector
\mathbf{n}_i	Unit normal vector of face i

p	Fluid static pressure
\bar{p}	Reynolds-averaged pressure
Pr	Prandtl number
Pr_t	Turbulent Prandtl number
q_i	Heat flux vector
q_{t_j}	Turbulent heat flux
Q	Vector of fluid primitive variables (p, u, v, w, ρ)
R	Residual or gas constant
Re	Reynolds number
s	Mesh spacing
S	Strain rate, Sutherland's temperature or vorticity
S_{ij}	Mean strain rate tensor,, $\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)$
$t, \Delta t$	Non dimensional time
$t^*, \Delta t^*$	Physical time
T	Fluid absolute temperature
u, v, w	Instantaneous velocity components of \vec{v} in Cartesian coordinates
\bar{u}	Mean velocity
u_{ij}	Velocity gradient of \vec{v} in i, j subscript form
U_∞	Freestream velocity
\vec{v}, \vec{V}	Fluid velocity vector
V	Bounded domain, or an arbitrary control volume
∂V	Surface boundary of domain V
W	Vector of fluid conservative variables ($\rho, \rho u, \rho v, \rho w, \rho E$)
x, y, z	Cartesian coordinates.
y^+	Dimensionless wall distance

Greek Letters

Symbol	Quantity
α	Angle of attack or radius of circles used to compute alpha shape
δ_{ij}	Kronecker delta
γ	Ratio of specific heats, for air $\gamma = 1.4$
Γ	Preconditioning matrix
Δ	Increment, vector differential operator (gradient of a scalar field, Divergence of a vector field)
κ	Thermal conductivity coefficient
λ	The second coefficient of viscosity, $\lambda = -2/3\mu$
λ_2	Lambda-2 criterion (for vortex extraction)
μ	Dynamic viscosity
μ_0	Reference dynamic viscosity of air
μ_t	Eddy or turbulent viscosity
ν	Kinematic viscosity, $\nu = \mu/\rho$
ν_t	Kinematic eddy viscosity
$\tilde{\nu}$	Modified kinematic eddy viscosity in Spalart-Allmaras turbulence model
ρ	Fluid density
$\bar{\rho}$	Reynolds-averaged density
ρ'	Density fluctuation by Reynolds-averaging
τ_{ij}	Reynolds stress tensor, $\tau_{ij} = -\overline{\rho u'_i u'_j}$
τ_w	Wall shear stress, $\tau_w = \mu(\partial u/\partial y)_{y=0}$
ω	Relaxation factor
ω_i	Vorticity, $i = x, y, z$
Ω_{ij}	Mean rotation rate tensor

CHAPTER I:

Introduction

1.1 Background

Mesh or grid generation can be considered an essential pre-requisite of any numerical simulation of physical field phenomena described by partial differential equations. The mesh provides a means to discretise the governing equations in space to allow an approximate numerical solution to be obtained. Over the past two decades, considerable effort has been devoted to the development of a variety of meshing techniques for computational fluid dynamics (CFD). The increasing maturity of the available techniques has provided a great level of flexibility for the discretisation of increasingly complex geometries. Despite this flexibility, for some techniques a number of deficiencies still persist and therefore offer scope for further improvement. Ultimately, the domain geometry and the physical problem being modelled determine the most suitable mesh generation technique. Furthermore, the quality and behaviour of the numerical solution can be greatly affected by the choice of mesh type.

Meshing can be considered as a bottleneck in the CFD process from two viewpoints. Firstly, as an essential stage of any numerical simulation process, failure to generate a valid mesh renders any simulation impossible. Secondly, for increasingly complex and realistic geometries, successful generation of an initial mesh may consume a vast period of time. This essentially delays obtaining a solution of the flow problem. Typically, for three-dimensional problems, it is likely that the mesh is composed of millions of elements; thus the required solution time to reach a converged state may also be considerably large. It is obvious that robust and flexible meshing techniques are required, not only to successfully generate valid meshes, but also to produce that mesh with a minimum amount of effort. From an industrial viewpoint this is extremely important. Not only does alleviating the bottleneck associated with meshing shorten the duration of a design cycle, but also allows the designer or engineer to better spend their knowledge and time modifying, assessing and improving designs in a shorter timeframe.

Robust meshing techniques are not the only requirement for successful CFD

simulations. Solution fidelity is closely linked to mesh quality and the appropriateness of a particular mesh for a given problem. Strong gradients in the solution can become a source of error that may contaminate the solution if the mesh local to these regions is not suitable [1, 2]. Global refinement of the mesh allows a reduction in the discretisation error but, particularly in the context of three-dimensional problems, such an approach will quickly exhaust computational resources. Modifying the mesh locally based on knowledge and experience alone is not sufficient: the flow features are part of the solution and often difficult to predict. Methods to automatically refine the mesh in the regions which contribute to the solution error have found extensive use in CFD to improve the solution in a cost-efficient manner. Consideration can also be given to the type of elements which are being employed to capture the flow physics. Before considering these aspects in more detail, a brief discussion on types of mesh generation techniques will be given.

1.2 Mesh Classification

The simplest form of mesh classification is based upon the connectivity of the mesh; i.e. whether it is structured or unstructured^[3, 4]. Structured meshes possess a regular connectivity that may be expressed in two- or three-dimensional arrays. Due to this regularity, the connectivity is implicit in the sense that neighbour information can be identified based on the storage of the mesh points in the computer memory. This allows for efficient mesh storage, which combined with the high quality of the actual mesh itself, contributes to efficient and accurate application of numerical schemes. Unfortunately, direct use of structured meshes is usually limited to simple geometries, restricting widespread application. For more complicated geometries, structured mesh generation may still be possible, but will usually be accompanied by the penalty of a lengthy period of user interaction in order to construct the mesh through domain decomposition, e.g. multi-block structured mesh methods.

Unstructured meshes are composed of an arbitrarily connected collection of elements (sometimes of multiple cell types – these meshes are often referred to as ‘hybrid’) which fill the domain space. This leads to a greater overhead in terms of mesh storage as the connectivity of the mesh must be explicitly stored to allow the flow solver access to the neighbourhood of a mesh element. However, the primary advantage of unstructured meshes

is their flexibility – modern algorithms can automatically produce valid meshes for highly complex geometries. In addition, unstructured meshes lend themselves to a variety of powerful solution-based adaptation algorithms that can improve the capability of unstructured elements to capture highly anisotropic flow physics and provide accurate and efficient solutions to common CFD problems.

Beyond the simple classification of mesh types described above, a more specific and useful classification can be carried out based on the actual methods used to generate the mesh. Mesh generation methods can be classified based on their intrinsic properties^[4] leading to the definition of several main categories. In the context of CFD, the classes of methods that most commonly find use include:

- Parameterisation or mapping methods: An inverse transformation maps a regular grid of points from a parametric space into the physical space, creating a structured mesh. The first of two main approaches is algebraic interpolation, where a mesh is generated through transfinite interpolation from discretised curves or surfaces. The second approach involves numerically solving a system of partial differential equations in order to fill the domain^[5-9].
- Domain decomposition methods: the global domain is split into smaller sub-domains. Two main approaches exist; the primary difference being the nature of the resulting mesh which covers the sub-domains.
 - Block-decomposition methods: These methods involve decomposition of the domain into smaller blocks. Within each block, a structured mesh generation technique (such as one of the mapping methods described previously) is applied. Manual decomposition of the domain can be an extremely time-consuming task which has led to a number of algorithms being developed to allow for automatic decomposition^[10-13].
 - Spatial-decomposition methods: each domain is progressively decomposed into a collection of disjoint mesh elements creating an approximation of the domain (i.e. the mesh will not conform to the boundaries). The techniques

applied to generate such a mesh are more specifically referred to as quadtree and octree methods^[14-21], and can automatically generate Cartesian type meshes (which may be considered as an unstructured type of mesh), triangular and tetrahedral meshes, and have also been used to create unstructured hexahedral meshes, with the latter most commonly appearing in the context of finite element applications for computational solid mechanics.

- Point insertion & element creation methods: After discretisation of boundary curves or surfaces, nodes or elements are gradually inserted into the empty domain until the entire space is filled with the mesh. Delaunay-based^[22-31] mesh generators insert points based on the Delaunay criterion, whilst advancing-front methods generate the mesh by advancing mesh points outwards from a discretised boundary^[32-37]. Application of these methods will yield unstructured meshes. These techniques are usually applied to generate simplicial elements. Some methods allow for conversion of triangular elements into quadrilateral elements^[122, 123, 128]. Other element insertion methods have been developed for quadrilateral or hexahedral-dominant meshes, although again, usually with finite element applications in mind^[38-43].

As mentioned previously, the complexity of the geometry and type of physical phenomena that is required to be modelled is likely to be a determining factor in not only the type of mesh, but also the method that is applied to generate it. The initial mesh also influences the potential types of adaptation schemes available. The second aspect, namely the type of physical phenomena required to be captured, is of particular relevance to the present project.

1.3 Generation of Flow Feature Aligned Meshes

From a mathematical point of view, one can consider the convergence of the approximate numerical solution to the exact solution of the system of partial differential equations as the mesh element size tends to zero. However, from an engineering point of view, this is not particularly meaningful due to limitations of computer resources. A compromise must therefore be made in terms of the mesh size (i.e. the total number of points or elements present) and the desired accuracy of the simulation. It would be expected

that coarser meshes, whilst computationally cheap, would provide less accurate results compared with a finer mesh due to a greater level of discretisation error. The sensitivity of the converged solution to the mesh element density is often referred to as mesh-dependence. As part of the CFD process, it is usually necessary to carry out mesh sensitivity studies in order to approach a grid independent solution.

As part of the solution of the partial differential equations describing the flow problem of interest, a number of flow features may exist. For a standard mesh¹ to accurately capture flow features such as shock waves, boundary layers, wakes and vortical flows, a large number of mesh points will be required. Flow features such as shock waves and shear layers are highly anisotropic in nature, that is, the gradient of the flow variables has a dominant direction which is usually normal to the feature in question. To accurately capture anisotropic flow features using a standard mesh without some form of adaptation will require excessive refinement of the mesh throughout the entire domain - this is a highly inefficient approach.

From a basic understanding of the flow field and general CFD experience, one may attempt to generate a more efficient grid *a priori* to resolve the flow features. This is possible for flow features such as boundary layers in high-Reynolds number flows which are captured near the wall. In order to resolve the high gradients present in the shear layers normal to the wall, one common approach is to construct a structured or prismatic mesh which is clustered towards the surface in the normal direction. Since the element spacing along the boundary in the streamwise direction will be several orders higher than the spacing in the normal direction, these cells are considered anisotropic, and are therefore ideal for the capture of the anisotropic flow physics.

Whilst this is possible for solution features such as boundary layers, other flow features are generally solutions of the problem and therefore are unknown *a priori*. The location and extent of shock waves and wakes can only be determined from the solution itself. Even for the near wall meshes used to capture the boundary layers, the extent or thickness of the boundary layer, along with the optimal distribution of points in that region

¹‘Standard’ or ‘initial’ mesh refers to a mesh which has been generated with no *a priori* knowledge of the location or extent of the flow features

cannot be known prior to the generation of a solution. To remedy this, a range of mesh generation and adaptation schemes have been developed over the past few decades to allow for automatic refinement of the mesh in these flow feature regions. However, simply applying refinement in the vicinity of the feature is not sufficient. Alignment of the mesh with strong features in the numerical solution can be considered as important an attribute of an adapted mesh as small spacing^[44].

1.4 Importance of Mesh Alignment

For compressible flow problems, the finite volume formulation allows the solution of the local Riemann problem to be constructed. When the governing equations are discretised across the mesh, the numerical fluxes are generally computed for each face of the control volume by using an approximate Riemann solver for the face. These fluxes are usually calculated in a reference frame that is aligned with the mesh. The approximate Riemann solver is applied in a quasi-one-dimensional fashion normal to the cell interface. This leads to a problem, particularly for unstructured meshes, where the physical features can be misrepresented and this degrades the potential for high accuracy, since the control volume faces are at arbitrary angles to the flow feature^[44-46]. Minimum error should occur when the normal of the control volume interface coincides with the normal to the flow feature, and the distance is small. A well-fitted and designed structured mesh can possess faces which are either tangent or normal to the flow features with small spacing, but construction of such a mesh can be labour-intensive. Unstructured meshing algorithms are usually automatic, but the aforementioned accuracy problems exist due to the lack of alignment of the mesh elements which must be addressed.

In an attempt to deal with these difficulties in unstructured meshes, one approach aims to improve the discretisation scheme by developing multi-dimensional approximate Riemann solvers. Such a solver would be insensitive to the orientation of the control volume faces. Mavriplis^[45] reports that whilst progress has been made in this area, it remains a difficult problem. Another approach involves the use of rotated Riemann solvers^[47, 48, 51]. Rather than apply the Riemann solver aligned with the grid, an upwinding angle is determined based on the physical features present, usually from the flow gradients. The finite differencing then occurs normal to the wave fronts to provide better resolution of discontinuities. However, van Leer^[49] suggests that such methods are not particularly robust

when used with higher order schemes. The use of higher-order schemes to reduce numerical diffusion on coarse non-aligned meshes is also an option.

An alternative approach, and the method that will be focused on in this thesis, is the application of mesh modification techniques which allow for alignment of the mesh with the flow features. The alignment can be achieved with two methods, either applied together or in isolation. The first method is to develop specialised mesh generation techniques which use different element types, such as the use of prismatic and / or hexahedral elements in the flow feature regions. The other approach is to adaptively refine an existing mesh based on the corresponding solution, such that the mesh becomes aligned with the flow features. Mesh adaptation requires computation of a suitable error estimate which in itself can affect the final solution, since it will drive the adaptation scheme to refine the mesh in different locations. Error estimation is the subject of extensive study^[50, 52-56]. Most CFD problems have no analytical solution which means determining the formulation of appropriate error estimates, and in turn their effect on the numerical solution offers a significant challenge. The relative benefits of using one error estimator over another are not considered in the present work.

Adaptation schemes are usually based on some anisotropic metric which seeks to refine and stretch the mesh in the physical space where necessary. The anisotropic elements are more suitable for the efficient capture of anisotropic flow physics. MacCormack (cited by McRae^[44]) noted that dispersion in a solution is reduced to a local minimum when the centre of a shock transition is located at the midpoint between two mesh lines. In fact, the schemes should be able to capture a shock wave within a one- or two-cell width, provided suitable alignment is present. Most adaptation schemes in the context of CFD have been developed for use with unstructured triangular and tetrahedral meshes. As a result of the adaptation, these elements become extremely skewed in the flow feature regions. Whilst providing improved resolution of the discontinuities and quite often improved accuracy, the skewed cells have been observed to cause some problems. Qin and Liu^[57] report the detrimental effect of the highly stretched triangular elements on the flow solver behaviour, preventing robust convergence and affecting solution accuracy. Mavriplis^[45, 46] discusses similar issues with stretched simplicial meshes, and that an optimal shape needs to be defined. In particular, spanwise grid stretching, widely used in CFD, may have effects on

overall solution accuracy. It could therefore be expected that highly stretched triangular elements may also have a negative impact. Babushka and Aziz^[58] show how the accuracy of a two-dimensional finite element degrades as the maximum angle of the element increases, suggesting that prevention of obtuse angles is necessary in an adaptive scheme. Marcum and Gaither^[101] also support this viewpoint: angles approaching 180° can reduce stability and convergence rates, leading to significantly increased CPU time. Even if constraints are imposed on the mesh generation or adaptation process, yielding nearly right-angled triangles in the flow feature regions with the shortest edge normal to the feature direction, the diagonal edge will remain non-aligned with the flow feature.

Quadrilateral, prismatic and hexahedral elements on the other hand may have their edges or faces directly aligned with the flow. This became the primary reason for the development of hybrid mesh technologies^[59-65] to at least provide more efficient and accurate resolution of boundary layers. Unfortunately, adaptation schemes applied to quadrilateral and hexahedral meshes are less flexible than those for triangular and tetrahedral meshes, and are usually restricted to nodal redistribution methods. The techniques available for different types of mesh are discussed in more detail in the literature review following the present chapter. The development of methods to allow for the use of quadrilateral and hexahedral elements in the flow feature regions forms the basis of this thesis.

1.5 Aims & Objectives

The present work can be considered as an extension to the structured block insertion method of Qin and Liu^[57]. In particular, issues with the approach will be addressed in terms of eliminating user-interaction and attempting to extend to three-dimensions for a range of flow features. Of primary interest is the generation of high-quality quadrilateral and hexahedral elements aligned with the flow features. With this in mind, the following are the main aims of the thesis:

- Surface meshing: generation of feature-aligned surface meshes comprising of quadrilateral elements
 - Develop an automatic and general geometric representation approach for

- complex flow features
 - Use the flow feature curves as virtual geometries to force alignment of the quadrilateral mesh elements
 - Implement adaptation schemes to further optimise the mesh based on the solution
- Volume meshing: improving the ability of hybrid meshes to capture anisotropic flow physics by embedding hex-dominant mesh blocks in the flow feature regions
 - Transonic shock waves:
 - Generate quadrilateral surface meshes aligned with transonic shock footprints
 - Develop a means to extrude quadrilateral surface meshes aligned with the shock wave to form aligned hexahedral mesh blocks
 - Trailing shear layers/wakes and wing tip vortices:
 - Development of a method to allow suitably located quadrilateral surface meshes to be extruded through the extent of the extracted shear layers and / or vortices to form a feature-aligned hexahedral mesh block
 - Implement adaptation schemes to further optimise the feature-aligned hexahedral mesh blocks based on the solution.
- Demonstrate the benefits of the presence of feature-aligned quadrilaterals and hexahedra on the numerical solution compared with triangular and tetrahedral elements.

1.6 Thesis Outline

Due to the nature of the work undertaken during this project, it became apparent that the most logical way to write the thesis was to combine the methodology with the results. The main reason for this is that the developed feature-alignment techniques which will be described require information from the CFD solution, and therefore the presentation of the methodology makes more sense in the context of actual flow solutions. Dividing the thesis in such a way also allows a clearer demonstration of the development and

progression of these feature-aligned mesh generation techniques over time.

Chapter I presents an introduction to mesh generation, with brief details on common mesh types and generation techniques. A discussion on the importance of aligning the mesh with flow features present in the solution is provided which forms the motivation for the work undertaken in this thesis. Aims and objectives for the project are defined and described.

Chapter II is the literature review and discusses classic published literature and state of the art within the field of mesh generation, particularly those techniques which allow meshes to be generated aligned with flow features. Also of interest are methods related to mesh adaptation, which allow the mesh to be manipulated based on the underlying solution.

Chapter III presents the basic governing equations for fluid dynamics, along with the discretisation and numerical schemes which have been employed to obtain approximate numerical solutions to the flow problems of interest. The flow feature extraction and solution-based adaptation techniques which are exploited in the present work are also discussed. The two main pieces of software used throughout the thesis are introduced: the mesh generation tool SOLAR and the flow solver TAU.

Chapter IV is the first combined methodology and results chapter. It focuses on two-dimensional flow features of varying complexity, including shock waves and wakes. The development of a new technique to automatically represent flow features of varying complexity as geometric entities to influence the surface mesh generation is presented. This is in contrast to other geometric representation methods already established which are limited to simple flow physics and / or require significant user interaction. The technique involves the use of the medial axis as means to compute a geometric representation of flow features and is applied on unstructured quad-dominant surface meshes.

Chapter V shifts the focus onto three-dimensional shock waves. Of primary interest are transonic shock waves, where the medial axis approach is used to generate feature curves which provide a representation of the shock footprint on the wing surface. After

generation of a shock-aligned quad-dominant surface mesh, a novel extrusion method is applied to allow for the generation of a high quality shock-aligned hex-dominant mesh block, which is applicable for viscous transonic problems. Due to the complexity of transonic shock waves and their proximity to the wing surface, such flow problems had not been considered before with other mesh block insertion approaches. Furthermore, the majority of three-dimensional adaptation schemes established in the literature are applicable to inviscid flows only, due to the difficulty in adapting the boundary layer mesh in viscous computations. The method is applied to several test cases and the feature-aligned mesh performance is compared with standard hybrid meshes and adaptive hybrid meshes.

Chapter VI describes a modified extrusion process to allow feature-aligned meshes to be generated for three-dimensional problems involving trailing shear layers and wing tip vortices. After embedding surfaces into the domain which can be meshed like any other surface of the CAD model, the surface meshes are extruded along the feature direction to provide feature-aligned hex-dominant mesh blocks. These mesh blocks are then embedded into the hybrid volume mesh to improve the resolution of the trailing shear layers and wing tip vortices.

Chapter VII is the final results chapter and revisits previous test cases which have undergone all the aforementioned feature-alignment techniques. The performance of these meshes is compared with hybrid meshes which have been adapted based on an anisotropic metric. The suitability of the embedded hex-dominant mesh blocks to be optimised based on the solution through application of a nodal-movement adaptation technique is also presented.

Chapter VIII presents a brief summary of the techniques developed during this project, and some general concluding remarks regarding the numerical solutions. Finally, recommendations for future work are suggested.

CHAPTER II

Literature Review

2.1 Introduction

Chapter I highlighted the motivation for the development of methods to achieve mesh alignment with flow features present in the solution. Specifically, the desire to align the mesh with flow features using anisotropic quadrilateral or hexahedral elements was described. The purpose of the present chapter therefore is to review and discuss current trends and previous work in the field of flow feature aligned mesh generation and adaptation; with attention particularly given to methods which allow for alignment of quadrilateral and hexahedral elements. General mesh generation techniques are not of particular interest here and only referred to where relevant, but thorough reviews on mesh generation technologies may be found in the references^[3, 4, 46, 66].

There are a wide range of different feature-aligned mesh generation and adaptation techniques established in the literature. A feasible method for a particular flow problem depends on the type of mesh used to obtain the initial solution. For the purposes of this discussion, the possible approaches have been classified into three main categories. The first of these groups involve general adaptation techniques: these methods locally or globally modify the mesh based on some error estimate and include operations such as refinement, coarsening, edge or face swapping, nodal redistribution and complete mesh regeneration. The second group include specialised mesh generation techniques, such as mesh block insertion, after which mesh regeneration around the block (in a local or global sense) yields a new mesh which is aligned with the flow features. The final group includes miscellaneous methods which involve automatic domain decomposition and the overset grid / chimera approach.

The order of the discussion is as follows: adaptation techniques will be considered first, followed by methods which require mesh regeneration (at local or global levels) and / or some form of mesh block insertion process; and finally other miscellaneous methods which are not so easily classified. Within each category, the methods will be assessed based

on their relative abilities to produce high quality feature-aligned meshes in a robust and efficient manner, whilst minimising user interaction, as well as their abilities to provide improvements in the numerical solution accuracy.

The final part of the review will provide a summary of the findings described in the review. Following this discussion, the relative merits and shortcomings for each method will have been determined, including areas which have received little attention in the literature. This allows for the identification of potential directions for the research undertaken in the present thesis.

2.2 Mesh Adaptation

Mesh adaptation methods provide a means to optimise the mesh based on the solution to reduce the numerical error in a computationally efficient way, with minimum effort from the user. All mesh adaptation methods, regardless of how they actually change the mesh, have one aspect in common: they all require the determination of some error estimate. If the mesh fails to meet some prescribed accuracy, the chosen mesh optimisation scheme will modify the mesh based on the error estimate in order to better capture the physics of the problem and thus reduce the solution error. The process iteratively continues until the numerical error has dropped to a satisfactory level.

2.2.1 Classifying Mesh Adaptation Techniques

A short description on the main classes of mesh adaptation methods is in order since they will be referred to in the subsequent discussion. The three main types are h -methods, r -methods and p -methods. Other operations which swap edges and faces may also be applied to in order to change mesh topology and improve geometric properties of the mesh elements based on some quality metric. These types of mesh adaptation techniques modify the mesh in local regions of the domain. However, other adaptation techniques are global in the sense that at each iteration step, the entire mesh will be regenerated. In some situations, combinations of these types of mesh refinement and modification techniques are combined into a single mesh optimisation strategy.

h -methods modify the mesh by changing its connectivity. Common strategies include simple subdivision of cells as well as insertion or removal of nodes to change the overall mesh topology. The refinement operation will increase the density of the mesh in high-error regions, such as in the vicinity of a shock wave, whilst a coarsening operation leads to a reduction of the density of the mesh where the flow variations are more uniform in nature. The algorithms which allow the refinement or coarsening to occur differ depending upon the type of mesh element and whether it is necessary for the mesh to remain conformal (i.e. no hanging nodes or faces).

The r -method allows deformation of the mesh whilst maintaining element connectivity by redistributing nodes to more optimum locations. Thus, the computational demands remain constant throughout the adaptive process. The nodal-redistribution method will increase or decrease the vertex density depending on the local behaviour of the flow. Care must be taken in the application of this method that the deformed elements are not only valid, but also of an acceptable quality.

The final class of method are p -methods. These are primarily of interest in Finite Element Modelling (FEM) for computational structural mechanics; however recent efforts have allowed application of the techniques to computational aerodynamics problems^[67, 68]. The method adaptively varies the degree of the polynomial across each element in the mesh, and therefore the order of the numerical approximation, whilst keeping the mesh size constant. p -methods are usually combined with a h -method to combine the attributes of both approaches. Due to the current lack of widespread application of these methods in the context of CFD, and the fact they do not involve mesh alignment with flow features, they will not be considered within this review.

2.2.2 A Remark on Error Estimators

Any discussion on mesh adaptation also requires consideration for the error estimate chosen to drive the adaptation scheme. This aspect will be considered briefly in the present section as some details may be referred to in the subsequent text. The error estimation techniques which are most commonly applied in CFD problems, and those considered in detail for the present review, fall under two main categories: solution-based and adjoint-based. Solution-based techniques may involve the use of reconstructed gradients to drive

the adaptation scheme, but such methods are generally considered inappropriate and can lead to unreliable results^[50, 52, 69-72]. Other solution-based techniques, such as curvature-based methods, use the Hessian of a flow variable to construct an anisotropic metric. These methods have found more widespread application and success for complex two- and three-dimensional flow problems on a variety of different types of mesh over the decades and will be considered in more detail in the subsequent discussion.

There are some distinct problems with the solution-based error estimators, particularly gradient-based, which one should be aware of. These methods focus on resolving discontinuities or strong gradients in the flowfield. Quite often however, features such as shock waves can be predicted to be in the incorrect location^[69, 70] due to numerical errors which are convected downstream from regions of the flow where the solution varies in a more continuous fashion: these regions are ignored by a feature-based error indicator. In turn, this leads to the mesh becoming overly refined in the wrong location and, despite the resolution of the shock wave improving, incorrect lift and drag quantities are obtained. As discussed by Dwight^[52] one of the most obvious examples of this occurring is for the fishtail shock wave from high transonic flow over the NACA0012 aerofoil. The location of the normal shock is highly sensitive to mesh resolution elsewhere in the flowfield, leading to inconsistencies between the shock location for a feature-based adapted mesh and a globally refined mesh.

The curvature-based method shares some of the aforementioned drawbacks of gradient-based methods in that it targets regions of the domain where discontinuities exist, with little regard to regions where the flow varies in a more continuous manner. However, it also takes into account the natural anisotropy of physical phenomena. Anisotropic adaptation is vital for the accuracy of many CFD simulations. The application of these methods allows for a reduction in the numerical dissipation in flow feature regions. When coupled with a suitable adaptation strategy, the mesh becomes aligned with the dominant flow features. This alignment allows for more effective application of approximate Riemann solvers in the flux computations, leading to a reduction in discretisation error.

The relatively recent advent of the adjoint solver has allowed a more rigorous error indicator to be developed based on the adjoint solution^[71-79]. Adjoint-based indicators have

highlighted severe shortcomings in feature-based approaches for the computation of engineering quantities of interest, such as lift and drag. The adjoint approach estimates the local contribution of each mesh element to the error, and therefore refines the mesh in regions which will influence the output functional. The regions of refinement do not necessarily correspond to the regions refined for feature-based methods. As an example, for an inviscid transonic flow over an aerofoil as presented in the paper by Dwight^[52], the gradient-based method refines in the shock and wake regions only. The adjoint-based error indicator also adapts the mesh to features such as acoustic waves above the aerofoil, as well as near the aerofoil surface, particularly near the leading edge. It is these regions of the domain which are influencing the outputs of interest.

The distinct differences between pure Hessian and adjoint-based error estimators are demonstrated by Venditti and Darmofal^[71, 72] for a range of two-dimensional inviscid and viscous flows. Loseille et al.^[73] provide a similar scheme in three-dimensions for inviscid flows featuring supersonic shock waves and wing tip vortices. Both techniques have formulated the adjoint error problem to include the natural anisotropy based on the Hessian of the solution, allowing for application of the adjoint error estimate for adapting to anisotropic features. This allows direct comparisons to be made between the two different types of error estimates. Both papers highlight the greater ability of the adjoint-based method to refine the mesh in the correct locations to provide improvements in the quantities of interest. In other words, the adjoint method defines an optimal distribution of the degrees of freedom for the specified target. On the other hand, Hessian-based adaptation provides non-optimal results with less appropriate distribution of mesh resolution for accurate evaluation of the desired functional. Both studies have shown a greater level of accuracy can be achieved with significantly less numbers of mesh nodes through use of the adjoint error estimate.

There are some drawbacks associated with adjoint methods however. The methods can be complex and relatively expensive to compute, since the error estimator requires a dual solution: both the original flow solution and its corresponding adjoint solution. The original flow solution must also be well converged. Fidkowski and Darmofal^[74, 75] suggest other areas where the adjoint approach may be lacking. One issue is that coarse initial meshes can sometimes be unsuitable for capturing the output of interest, leading to the

error-estimate being severely unreliable. This is problematic since an inaccurate error estimate on a coarse mesh may cause an automatic adaptation process to terminate prematurely before the functional has been sufficiently resolved.

The application of some adjoint-based methods^[76, 77] on Cartesian meshes can only be applied to construct adapted isotropic meshes for the Euler equations. The main benefit of using a Cartesian mesh is that the mesh does not need to conform to the geometry boundary, allowing for complex geometries to be automatically handled. Refinement and coarsening schemes are also relatively simple to implement with no restrictions on conformity. However such approaches lead to hanging nodes or faces which are not suitable for some flow solvers. Whilst the mesh is refined in regions which contribute to the error, the solution anisotropy cannot be taken into account, meaning the application of such techniques will be unsuitable for viscous simulations and the associated solution features which may be present. The development of Cartesian mesh generation with prismatic layers around the geometry to allow for viscous computation has become an active area of research. Relevant work in this field will be discussed in more detail later in the review.

Another aspect to consider in anisotropic metric construction for mesh adaptation is the norm in which the metric is computed. Alauzet et al.^[109] demonstrate and discuss the effects of error estimators in controlling the interpolation error in different \mathbf{L}^p norms. Depending on the value of p , the mesh adaptation will react to different variations in the solution. For example, in the \mathbf{L}^∞ norm (i.e. $p = \infty$), the adaptation reacts strongly to regions of steep gradient, such as across a shock wave, with little regard to regions where there are smaller variations in the solution. On the other hand, lower values of p , or ‘weaker’ norms, such as the \mathbf{L}^2 norm, the adaptation becomes more sensitive to smaller variations in the solution. It is shown for a range of analytical cases that metrics constructed in the weaker norms lead to a more rapid reduction in the solution error as the number of mesh vertices increases. As observed with the adjoint-based research previously described, this work suggests that it is not just the regions of steep gradient that contribute to the solution error.

It is clear from the preceding discussion that associated with each of the error estimators are a number of distinct advantages and disadvantages. The field of error estimation remains an active area of research, but it would appear from the related literature

that an adjoint approach, which combined with the Hessian anisotropic metric information, is starting to become the most efficient way to accurately compute engineering quantities of interest. Furthermore, metrics based on weaker norms (i.e. \mathbf{L}^p with $p = 1, 2$) are potentially a more appropriate choice than the more traditional \mathbf{L}^∞ norms, as previously discussed. However, one is unable to ignore the range of problems for which traditional error estimators based on purely on the Hessian in the \mathbf{L}^∞ norm, have been put to good use for aligning the mesh with dominant flow features.

For the purposes of the present thesis, the focus will not be on the relative benefits of one error estimator over another. Of more interest is the adaptation scheme which is coupled with the chosen error estimator. As previously stated, adaptation or feature-alignment schemes which involve quadrilateral and hexahedral meshes are of primary interest – avoiding the generation of highly skewed triangular or tetrahedral elements is of primary importance. The estimation of error and the chosen adaptation scheme can be considered as independent problems. Given then, some error estimator or indication of flow activity which allow subsequent manipulation of the mesh, the goal is therefore to determine how the *type* of element present in the flow feature regions can influence the numerical simulation.

2.3 Mesh Adaptation Methods for Feature-Alignment

As mentioned previously, the type of adaptation scheme available to the user depends strongly on the type of the underlying mesh used to compute the initial solution. In the following discussion, the most common types of grid employed in computational aerodynamic problems will be considered in turn. The applicable adaptation techniques for each will be reviewed in order to determine their respective abilities in providing feature-alignment within the mesh.

2.3.1 Structured Meshes

Adaptation on structured meshes is usually limited only to nodal redistribution methods, in order to maintain the topology of the quadrilateral or hexahedral cells. Both the connectivity of the mesh and the number of nodes remain unchanged throughout the adaptation process, leading to a constant demand on computational resources. It is a method

whereby the error is equi-distributed over all the *edges* of the mesh, i.e. it is directionally based and can produce highly stretched meshes in the vicinity of flow features, as discussed by Tam^[80]. The quadrilateral cell interfaces eventually become aligned with the flow features, providing excellent resolution of the features for a variety of problems in two- and three-dimensions^[81-86]. The more reasonable grid distribution allows for resolution of important viscous interactions such as shock-boundary layer interactions^[87]. Application of nodal redistribution methods have also been applied effectively for unsteady problems such as shock-induced combustion^[88].

The limitation and inflexibility of this form of adaptation becomes obvious for more complicated geometries and flow features due to the structured nature of the grid. Care must be taken that the redistribution does not distort the grid in such a way that the quadrilateral or hexahedral elements become invalid. Maintaining the topology of elements is quite often an important consideration in a scheme which implements nodal redistribution alone. The inability of nodal movement techniques on structured meshes to capture features such as vortices was discussed in some length by Ait-Ali-Yahia et al.^[85].

Another general drawback of structured meshes is that even for some two-dimensional geometries, structured mesh generation can be a time-consuming task, requiring manual decomposition of the domain into small sub-regions for individual meshing. The requirement for a high level of user interaction makes these techniques somewhat unattractive, despite the potential for a highly efficient grid. However, efforts have been made to attempt to reduce the user interaction involved in the process. For example, the medial object of a flow domain provides a means to construct a decomposition automatically^[10-13, 89] and has been the subject of extensive research. Other techniques include the buffer or zipper method^[90, 91] where two multi-block meshes are joined together using a small layer of unstructured elements to yield a conformal mesh.

2.3.2 Unstructured Meshes

2.3.2.1 Triangular and Tetrahedral meshes

The majority of mesh adaptation techniques for CFD have been implemented for unstructured triangular and tetrahedral meshes, regardless of the chosen error indicator. There are several reasons for this. Firstly, the algorithms which generate unstructured

meshes comprising of these types of elements can be generalised to allow automatic meshing of complex configurations. This somewhat alleviates the bottleneck associated with the mesh generation process, which is prevalent for the structured mesh generation methods previously discussed. Furthermore, the inherent flexibility of simplicial elements means that they lend themselves to a variety of adaptation strategies, including the aforementioned r - and h -methods, along with edge / face swapping operations, complete mesh regeneration or even a combination of all these strategies. The application of these methods for unstructured triangular and tetrahedral meshes in aerodynamic problems is therefore widespread^[92-118]. One drawback of unstructured meshes relative to structured grids is that they require larger overheads for computation time, and also for memory since the connectivity must be explicitly stored.

Provided that a suitable anisotropic error estimator is coupled with the adaptation strategy, the coupled process should construct a mesh which will allow for the resolution of anisotropic flow physics in an efficient manner. Habashi et al.^[92] proposed a scheme which combined refinement, coarsening and nodal movement based upon the previously discussed Hessian-based error estimator. Dompierre et al.^[93] used these fully automatic adaptation strategies effectively for a wide range of two-dimensional flow problems. Frey and Alauzet^[94] presented a method involving complete mesh regeneration after construction of an anisotropic metric map which allowed for information on element stretching and orientation to be used in the meshing process. This was successfully applied to three-dimensional test cases featuring shock waves, but was limited to inviscid flows since the technique is unsuitable for constructing boundary layer meshes.

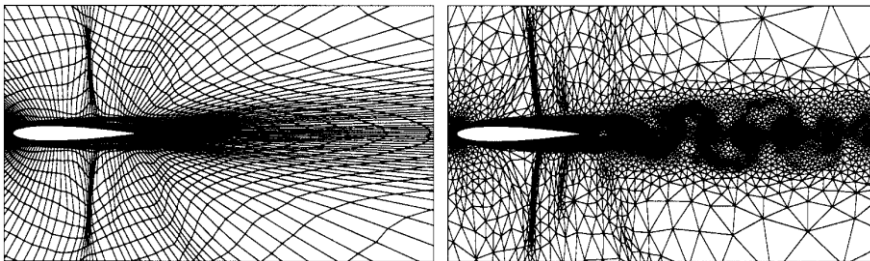


Figure 2.1: Comparison of structured vs. unstructured adaptation in capturing the von Karman street about a NACA 0012 airfoil^[85]

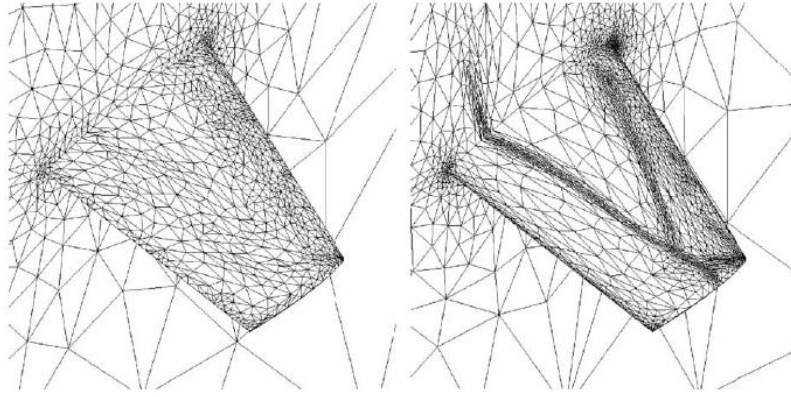


Figure 2.2: Anisotropic surface adaptation for inviscid transonic M6 case^[94]

Whilst the previously described methods work well in two-dimensions for a range of flows and for inviscid three-dimensional simulations, transonic viscous simulations for fully tetrahedral meshes are more problematic to address. This is pointed out by Loseille and Lohner^[102], who have developed a scheme using cavity-based operators to alleviate the problems associated with boundary layer creation from highly anisotropic surface meshes when using an iterative adaptive scheme.

Despite the inherent flexibility of unstructured triangular and tetrahedral meshes, there are a number of distinct drawbacks. One of the major problems with purely anisotropic adaptation on these types of mesh is that the elements can become extremely skewed as they are clustered towards the regions of high gradient. In two-dimensions, the flow feature will only be aligned with one of the edges of the triangular elements. Such stretched elements can be considered to be of poor quality in a geometric sense. Depending on whether there are any restrictions on the internal angles of the triangles, quite often only one of the triangular edges can be aligned with the flow feature. The resulting mesh can, in some cases, degrade the potential for high resolution and high-order reconstruction of the flow variables, as discussed in Chapter I. The numerical solution can then suffer from a number of issues which include oscillations, slow convergence and loss of accuracy. Such an effect has been presented by Qin and Liu^[57] where purely anisotropic adaptation on a triangular mesh has caused convergence problems and loss of accuracy despite being fully automatic.

2.3.2.2 Quadrilateral and Hexahedral Meshes

Unstructured quadrilateral meshes represent an alternative to triangular meshes, but in the context of CFD they have not received the same level of attention, with notable exceptions ^[119-124]. For surface meshing, the advancing front or paving algorithm^[43] can offer a similar level of flexibility as its triangular counterpart. Given a number of points, an unstructured quadrilateral mesh will contain significantly fewer elements than the triangular mesh, leading to the quadrilateral mesh being more computationally efficient. This fact also extends to the three-dimensional case, where a number of tetrahedra are required to fill the void which one hexahedral element would occupy. A wide range of hex-dominant meshing techniques have been developed with varying levels of robustness, although usually with finite element analysis for computational structural mechanics in mind^[38-40, 125-127, 132].

Perhaps one of the reasons unstructured quadrilateral meshes do not frequently appear in the context of CFD, is due to the lack of maturity of their associated adaptation techniques. For example in two dimensions, if a mesh is to remain conformal, certain restrictions must be placed on the quadrilateral element to maintain its topology and convexity. The relative lack of flexibility quadrilateral meshes possess due to these geometric restrictions has led to a greater level of application of triangular and tetrahedral adaptation methods for aerodynamic problems. The level of maturity for the triangular / tetrahedral methods is clear to see from the related literature. Any initial unstructured mesh suffers from a lack of alignment of the mesh elements with potential flow features; thereby degrading the mesh performance and highlighting the necessity for some form of adaptation scheme to be applied.

Another aspect to consider is that robust hexahedral meshing is only just starting to become possible for complex geometries of interest in aerodynamics. Now hexahedral meshing algorithms are maturing, this calls for a new breed of conformal methods to allow solution-based refinement of such meshes. The development of these techniques will allow the full benefits of using unstructured quadrilateral / hexahedral meshes to be realised: not only in terms of their greater level of efficiency, but their potential for the cell interfaces to be fully aligned with the flow features of a CFD solution.

There has been some research into adaptation methods applied on unstructured quadrilateral meshes. Zheng et al.^[124] present a cell-based anisotropic adaptive solution for the Euler equations for quadrilateral cells. Due to the anisotropic refinement, the resolved flow features become aligned with the quadrilateral mesh structure. The technique allows for efficient and accurate capture for a range of problems involving complex shock structures. Furthermore, the adaptation strategy also allows for coarsening so that mesh resolution is not wasted in regions where it is not needed. The quality of the quadrilateral elements is always maintained since the refinement and coarsening operations do not distort the mesh structure in the way nodal redistribution methods can.

Tchon et al.^[119] implement a different approach which allows for conformal refinement, based on a strategy referred to as pillowing or buffer insertion. According to an anisotropic control metric, regions for refinement are located and a scheme following that of Schneiders^[129], which uses layers of contiguous elements known as a ‘shrink set’. As the name implies, this small selection of elements is shrunk in order to generate a gap between the shrink set and the surrounding element set. The two elements sets are then reconnected, bridging this narrow gap, forming a new sheet of elements. The algorithm is explained in detail within the paper, but figure 2.3 provides a simple demonstration of the effect of the refinement in two-dimensions. The algorithm also naturally extends to three-dimensions.

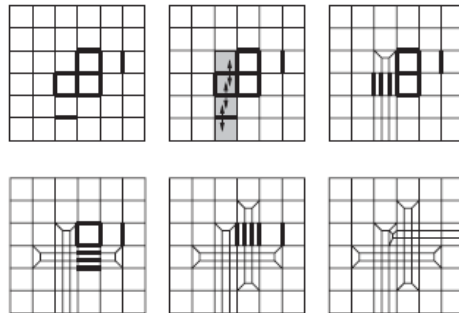


Figure 2.3: Refinement algorithm steps for the pillowing method^[119]

The technique is applied to an unsteady viscous laminar flow around a NACA0012 aerofoil, and adapts well to the shocks and trailing vortex street. Note that this is the case for which Ait-Ali-Yahia et al.^[85] demonstrated that nodal movement adaptation alone on structured grids failed to adapt to the vortices. However, the pillowing method has not

provided the same level of mesh resolution in the wake region as the unstructured triangular refinement methods described by Dompierre et al.^[93]. This could be due to the use of different formulations of the metric which drives the adaptation. Figure 2.4 shows an image of the adapted mesh from Tchou et al.^[119]. Whilst the algorithm has worked well to force alignment of the mesh with the flow features for this case, there are a few issues with the quality of the mesh elements, particularly in the vicinity of the shock waves.

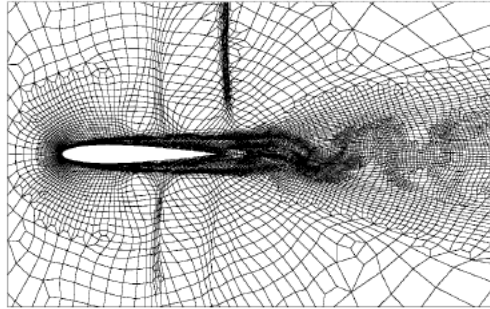


Figure 2.4: Pillowing method applied to an unsteady viscous laminar case after smoothing^[119]

A slightly different approach is considered by Borouchaki et al.^[122, 123], where the adaptive quadrilateral mesh is indirectly obtained by using a scheme which converts the triangular mesh to a quadrilateral one. The triangular mesh is adapted based upon an anisotropic metric until the lengths of the edges in the Riemannian space are almost unity. This mesh is then converted into a quadrilateral mesh by considering merging individual pairs of triangles, if the resulting quadrilateral quality will be sufficient. However, the method cannot guarantee the mesh is entirely made up of quadrilaterals, and can occasionally lead to a mixed mesh composing of both types of elements. The proportion of triangles is relatively small however. The technique is applied to a supersonic aerofoil case and the resulting unstructured quad-dominant mesh is aligned with the shock waves and wake with high-quality elements. As the majority of the elements in the new mesh are primarily quadrilaterals, this mesh is inherently more efficient than the equivalent fully triangular mesh.

Merkley et al.^[130] describe the concept of sheet insertion, upon which all hexahedral based refinement techniques, which include the aforementioned pillowing approach, depend on. A hexahedral sheet can be viewed as a dual to a layer of hexahedral elements,

and these can be inserted into an existing mesh. The new elements therefore will directly correlate with the shape of the sheet, as shown in figure 2.5. If a sheet is restricted to be refined along a particular direction influenced by its shape, then such an approach could prove useful in the context of feature-alignment. Indeed, this is pointed out by Merkley et al.^[130]. A mesh could be roughly aligned with the feature shape, such as a bow shock, using a simple r -method. This essentially changes the shape of the sheets, meaning sheet insertion along the relevant sheets allows refinement to occur along the shock wave.

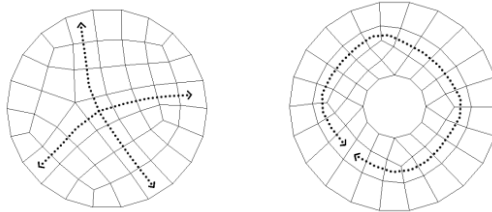


Figure 2.5: Rows of quadrilaterals and sheet pathways^[130]

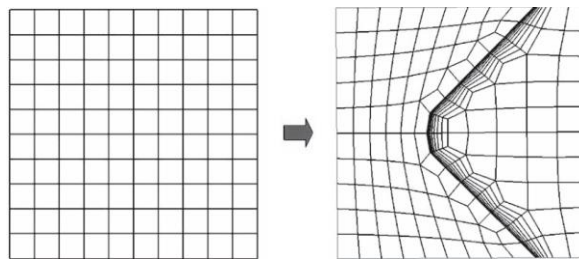


Figure 2.6: Application of sheet insertion for a bow shock type feature after approximate alignment^[130]

One problem with sheet insertion is that refinement will occur along the entire sheet length, and propagate towards the boundary. In an unstructured hexahedral mesh, the pathway for a sheet is unlikely to be regular, resulting in a rather unpredictable refinement path. Whilst the example presented in figure 2.6 would be suitable for this refinement to the boundaries, for other flow features, such as transonic shocks, it is not desirable to have this refinement propagating throughout the domain. One method which allows the sheet propagation to be terminated in a conformal manner is through the use of refinement templates.

Usually with quadrilateral refinement, non-conformity arises whereby local refinement introduces non-conformal nodes that lie on edges of neighbouring elements. The

majority of recent work^[133-136] is an extension of some of the best known work on the topic, albeit with finite element solids modelling in mind, by Schneiders et al.^[129]. Garimella^[131] describes a technique similar to Tchou et al.^[119] to refine unstructured quadrilateral meshes such that the resulting mesh maintains conformity throughout and prevent hanging nodes.

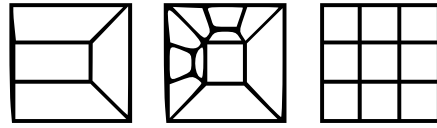


Figure 2.7: Examples of quadrilateral refinement templates^[131]

The refinement methods discussed in these papers are carried out utilising a set of defined templates (see figure 2.7 for some examples). Application of such refinement methods alone may be unsuitable in the context of feature-aligned adaptation. In order to achieve alignment of the cell interfaces, some form of smoothing will also be required. However, the unstructured methodology by Garimella^[131] may have potential in terms of increasing the element density in an isotropic manner (figure 2.8) necessary for the capture of vortex shedding for example. A range of three-dimensional refinement templates has also been defined for hexahedral meshes.

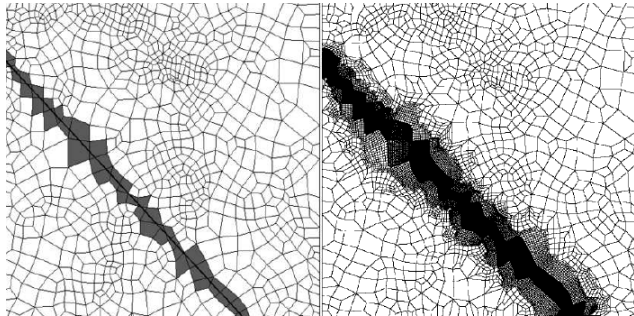


Figure 2.8: Isotropic refinement of an unstructured quadrilateral grid^[131]

2.3.2.3 Cartesian Meshes

Cartesian grids do not require body-fitting in contrast to structured and unstructured meshes. This means that the mesh is independent of the geometry discretisation, leading to generation methods which are fully automatic and robust. This allows meshes to be generated for complex three-dimensional configurations with relative ease^[137-142]. The

actual geometry is carved out from the interior of the mesh, leaving a set of irregular cells at the boundaries. However, the majority of the domain is made up of completely regular cells. This leads to highly efficient and accurate finite volume scheme application^[66]. Despite these benefits the majority of Cartesian-based mesh methods are only suitable for inviscid flows, since the cells cut at the boundaries are unsuitable for resolving viscous boundary layers.

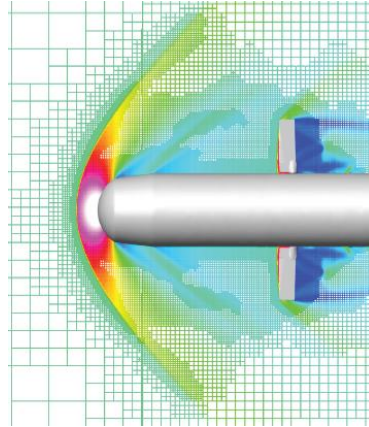


Figure 2.9: Example of isotropic Cartesian refinement in shock wave region^[76]

Combining such meshes with boundary-fitted prismatic cells is one method to allow Cartesian type meshes to be applicable for viscous flows, and allow predictions of the same quality with other types of boundary fitted meshes. Recent advancements in this technology have made this possible^[139, 140], with the development of the BOXER software. The ultimate goal of the research is to unify all aspects of the CFD process into a single piece of software, all implemented in parallel to allow for robust and automatic design analysis, particularly important for industrial design loops. As the references show, this approach has been applied to extremely complex geometries with a minimum of user effort, something other mesh generation methods would struggle to achieve. Furthermore, the resulting mesh is extremely efficient as it is essentially a hex-dominant mesh. One major drawback is that for three-dimensional problems where small geometric features are present, a high density of cells will be required leading to extremely large meshes being generated.

From a feature-alignment point of view, other than the prismatic cells in the boundary layer, anisotropic refinement may be difficult to achieve for other flow features such as shock waves. In fact, without some form of refinement template (as considered in

section 2.3.2.2) which will allow transition of the mesh size from smaller regions to the larger regions, any form of refinement whether that be isotropic or anisotropic will lead to hanging nodes or faces. Even anisotropic types of refinement still prevents full alignment of the cell interfaces with the flow feature, particularly if the feature possesses some form of curvature – the feature will continue to pass through the cells at an angle. As previously discussed, McRae^[44] suggests that both small spacing and alignment are important attributes to an effective adaptive scheme. However, an exception to other Cartesian methods is the work of Wintzer^[138] where the initial Cartesian mesh is aligned with the shock direction and therefore the subsequent refinement occurs aligned with the shock waves, displayed in figure 2.10. As the application of the work is for sonic boom prediction, this refinement allows for excellent resolution of the supersonic shock waves.

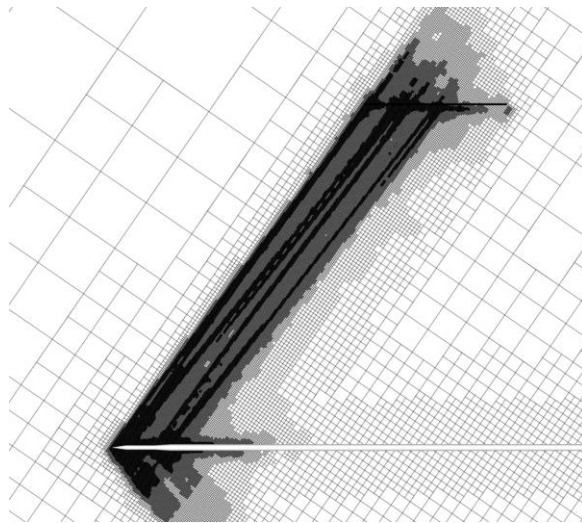


Figure 2.10: Aligned Cartesian mesh refined in shock wave region for sonic boom prediction^[138].

Whilst the previously described types of refinement have been shown to be effective for the capture of a range of flow phenomena^[76, 77, 141, 142], the lack of conformity is not suitable for some flow solvers.

2.4 Mesh Insertion, Regeneration and Specialised Generation Techniques

An alternative approach that has evolved in recent times is to represent solution features as geometric entities. After flow feature extraction using some feature detector, a geometric representation technique is applied to approximate the flow feature topology as a means to influence the mesh generation process^[57, 101, 143-146]. Since the flow feature

geometry is computed from an initial solution, these techniques usually require that the mesh is regenerated to obtain the feature-alignment. The solution may then continue on this new mesh.

The earliest form of geometric-representation techniques for aerospace applications was described by Marcum and Gaither^[101]. A pseudo-pattern recognition technique allowed for extracted regions to be reduced to groups of simple geometric entities. For example, shock waves and wakes are reduced to lines, whereas expansion or stagnation regions are reduced to points. The process allowed unstructured adaptive grids to be constructed automatically to capture the features of interest. Singular points are treated as adaptive sources which allow for localised isotropic refinement, which reflects the isotropic nature of the physics in those regions. The lines are embedded dual-sided boundaries which allow high-aspect ratio cells to be generated aligned with the anisotropic features. Whilst the technique can be applied to fairly complex two-dimensional flow features, such as cases where shock waves cross each other, the technique is applied to triangular meshes only. Extension to three-dimensions is also problematic. The pattern recognition technique may be applicable in three-dimensions, but generation of surfaces to represent flow features poses a significant challenge.

Ito et al.^[143] present another geometric representation process for steady-state problems. After using feature detectors, the medial axis of the flow feature is extracted using either a least-square approach or a Delaunay triangulation method. The method embeds the medial axis within the two-dimensional domain and removes a section of unstructured mesh around the flow feature. The advantage of this approach is that the entire domain does not need to be regenerated, only the void created by mesh removal. However, the approach is only suitable for simple flow features, requires user interaction to smooth medial axis and also only allows for isotropic refinement, which as previously discussed is unsuitable for anisotropic flow physics.

One problem with fully unstructured meshing is that for the inclusion of viscous computations, the boundary layer must be captured by extremely skewed triangular or tetrahedral elements. Such cells are generally not desirable in the boundary layer. This led to the development of hybrid meshes^[59-65], which feature structured or semi-structured

meshes near the boundary consisting of quadrilateral, prismatic or hexahedral elements. Triangular or tetrahedral elements are employed to cover the rest of the domain, with an interface region, sometimes referred to as the ‘buffer’ mesh comprising of pyramids and tetrahedra. These meshes can provide more efficient and accurate capture of near-wall physics, since the hybrid mesh can possess significantly less elements in the boundary layer, which are also not severely skewed.

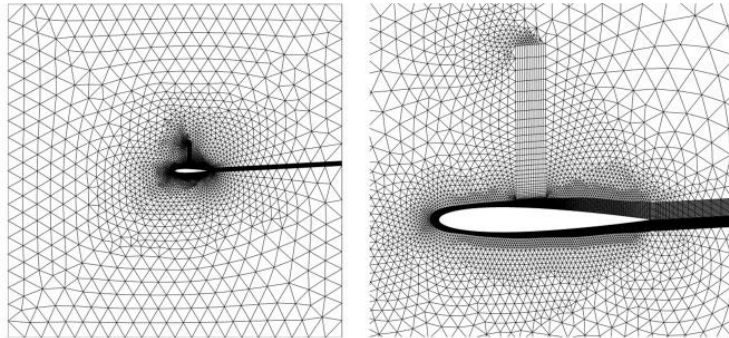


Figure 2.11: Structured block insertion method prior to nodal movement adaptation^[57]

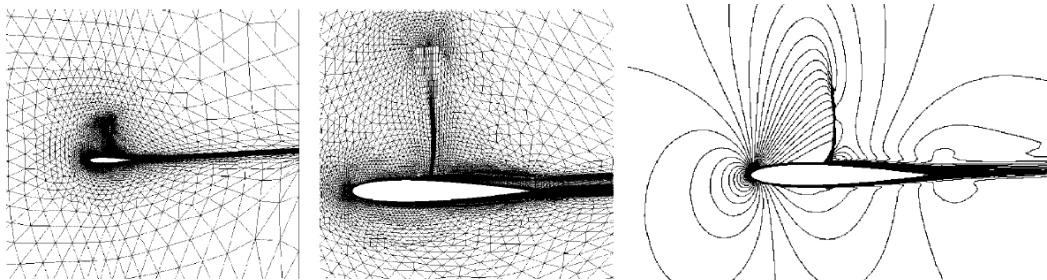


Figure 2.12: Structured block mesh after application of anisotropic adaptation method used for feature-alignment with shocks and trailing wakes^[57]

The term ‘hybrid’ mesh is therefore applied for those meshes which contain a mixture of elements and have seen application for a range of two- and three-dimensional viscous flows. Some methods have emerged which exploit the ability of the hybrid mesh to be comprised of different elements types in the context of feature-aligned meshing. Burgos^[147] presents a method which allows for the embedding of semi-structured mesh blocks aligned with the wake region downstream of aerofoil sections in turbo-machinery applications. Qin and Liu^[57] proposed a two-dimensional feature-aligned mesh generation and adaptation method by inserting structured blocks of mesh into the flow feature regions

(figure 2.11), using a geometric curve to guide the placement of the mesh block. The remainder of the domain is made up of triangular elements. The mesh can then be adapted using an anisotropic redistribution method, displayed in figure 2.12. This prevented the generation of highly skewed anisotropic triangular elements in the flow feature regions. Compared with standard anisotropic adaptation on triangular grids, the structured-block insertion method provided superior accuracy, convergence properties and resolution of the flow features. Unfortunately there were several issues with the approach. Most notably, the process is semi-automatic and remains significantly challenging to extend to three-dimensional problems.

Ito et al. and Shih et al.^[144-146] have developed and extended feature-alignment techniques for three-dimensional hybrid meshes. Surfaces are generated after feature detection and these surfaces allow construction of feature-aligned prismatic layers which provide improved solution of bow shocks for complex geometry (figure 2.13). Whilst some high-quality feature-aligned meshes have been constructed, the approach is limited to capture of bow shocks. Other shock structures must be resolved by isotropic refinement. The refinement approach is also applied for wing tip vortex capture in a viscous case, where there is significant improvement in the resolution of the vortex. However, the isotropic refinement in the wake region can be considered sub-optimal, where ideally an anisotropic form of refinement would be more appropriate. Consideration to three-dimensional shock waves within the transonic range has not been given by any of the authors who have adopted the geometric representation approach. This is most likely due to the complexity of the shock structure and its proximity to the body, leading to a need for modification of the surface mesh.

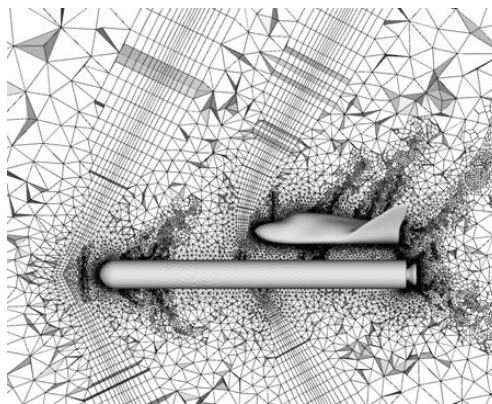


Figure 2.13: Prismatic layer mesh insertion into hybrid mesh using embedded surfaces^[146]

2.5 Miscellaneous Techniques: Overset Grid ‘Chimera’ Approach

A wide variety of complex fluid problems can be dealt with through the use of this approach. As previously eluded to, not all geometries will be able to be easily meshed with a single contiguous mesh. The use of structured overset grids allows simplification of the mesh generation process since the grid zones are allowed to overlap, rather than be aligned with their neighbours, as in multi-block structured mesh generation. However, the overset grid approach does require the domain connectivity to be determined so that adjacent overset grids can share information. Information is interpolated from solutions in the overlapping region of adjacent grids. This approach is claimed to be able to enjoy the advantages associated with unstructured meshes, whilst retaining the computational benefits of structured grid data^[66]. Craft et al.^[148] discuss the ease with which meshes can be generated aligned with the flow direction.

Since adjacent domains do not need to conform means that overset grids are able to be used in adaptive methods, such as the method described by Berger and Oliger^[149] and Berger and Jameson^[150]. Starting with a coarse mesh to resolve the uniform regions of the flow, shock and other steep gradient regions are adaptively resolved by overlaying a series of finer grids onto the coarse mesh. Other practical examples^[66] involve a grid generation approach for a flapped-wing configuration. A convenient set of topologies can be used to fill the farfield domain, in this case a Cartesian grid. Then, a hyperbolic approach which is ideal for near-body grids in generation of suitable boundary layer meshes is used around the flapped-wing geometry. These grids can be combined to cover the entire computational domain. Since each grid generated was defined individually, any changes to geometry will require only modification of those particular grid sections, relieving the need of remeshing an entire domain. This ultimately gives a reduction in the time required to plan and generate the grid.

The fact that only local regeneration is required and that the mesh blocks can therefore be generated around moving components mean that such methods could be applied for transient flow features, such as moving shock waves. This technique was investigated by Chawla and Banks^[151] who use the method to track features such as moving

shock waves and vortices. Figure 2.14 has been taken from work by Meaking^[152, 153] where the individual domain blocks can be observed for the near-wall physics and transonic shock.

Typical problems encountered however include the complexity of more than two grids overlapping, particularly three-dimensions, and the resulting complex interpolation of data between the overlapping zones may prevent accurate solutions. Indeed, Guerrero^[154] discusses that the main issue for the chimera approach is successful interpolation of the variable fluxes, particularly in terms of ensuring that conservation of the variables is maintained. However, good results have been achieved through the use of the chimera method. It is certainly a method worth considering in the context of feature-aligned mesh generation, but obviously is restricted for flow solvers designed with this kind of mesh in mind.

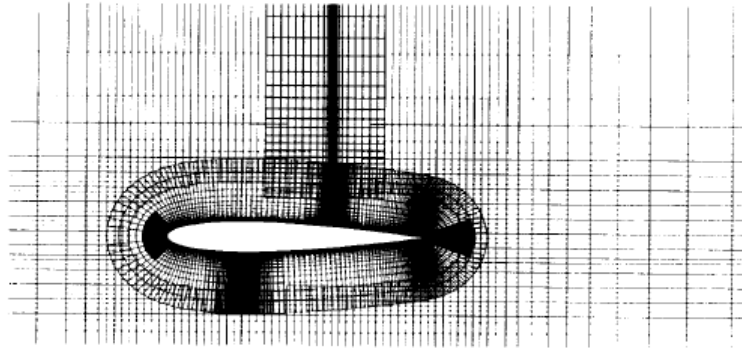


Figure 2.14: Example of the chimera approach applied to resolve a transonic shock wave^[153]

2.6 Summary

- The majority of mesh adaptation schemes in the context of computational aerodynamics have been developed for triangular and tetrahedral meshes due to the inherent flexibility of such meshes.
- Structured meshes are usually limited to r -methods (nodal redistribution) to maintain element connectivity. These methods work well for capturing shock waves and shear layers as full alignment of the cell interfaces can be achieved, but cease to be effective for flows involving vortices.

- Anisotropic adaptation methods on triangular and tetrahedral meshes lead to extremely skewed elements which can affect solution quality. Such elements are inappropriate to capture near wall shear layers efficiently, meaning anisotropic methods are usually restricted to inviscid flows.
- During the advent of CFD, unstructured quadrilateral and hexahedral meshes have not received the same level of attention as triangular / tetrahedral meshes. Whilst robust quadrilateral meshing algorithms have been developed, reliable and robust conformal hex-dominant meshing in three-dimensions has only recently become feasible. Further reasons for the lack of such meshes being applied in the context of CFD is that the associated adaptation schemes have not yet reached the same level of maturity as those developed for triangular and tetrahedral meshes.
- Cartesian mesh generation can be applied to automatically create meshes for complex geometry due to the removal for the requirement of body-fitted cells. This has restricted their application to inviscid flows for the majority of cases. However, recently the emergence of methods to connect the Cartesian mesh to a body-fitted prismatic layer has allowed for viscous problems to be considered. The resulting meshes are essentially hex-dominant. For conformal adaptation on such meshes, the use of refinement templates is necessary.
- To alleviate some the restrictions of tetrahedral meshes in terms of viscous simulations, hybrid meshes were developed which allow prismatic or hexahedral elements to be grown from the boundary surface. Some methods further exploit that hybrid meshes can be comprised of multiple element types by using methods such as geometric representation techniques to allow for the embedding of high quality regions of mesh aligned with flow features. Drawbacks of geometric representation methods include that user interaction is usually required at some stage in the process, and they are limited to two-dimensional or very simple three-dimensional flow features.
- Chimera approaches allow for complex geometries to be dealt with and offer efficiency improvements compared with unstructured meshes. In the context of feature-alignment, structured mesh blocks can be placed in regions of high error and aligned with flow features to improve resolution. Such techniques also offer the

advantage of being applicable to problems with moving geometry and unsteady flow features.

The considerations above have allowed for identification of areas where research is currently lacking in terms of feature-alignment techniques for CFD applications. As described in Chapter I, one aspect of this thesis is to address some of the drawbacks of the structured mesh block insertion method^[57]. In particular, development of a method to reduce, if not eliminate, user interaction is required. Application to more complex flow features would also be desirable. The limitations of geometric representation approaches have been described, and addressing these problems represents a potential challenge to be undertaken. Furthermore, development of mesh modification techniques for quad-dominant meshes such that they become viable alternatives to triangular meshes is required. This will allow the greater efficiency and potential accuracy of quadrilateral meshes to be fully realised in the context of CFD.

As previously discussed, for three-dimensional flows, hexahedral or prismatic cells are desirable to provide non-skewed alignment with flow features. Ito et al. and Shih et al.^[144-146] have already made some progress towards this goal by embedding prismatic elements into a hybrid volume mesh for supersonic bow shocks. However, bow shocks are relatively simple structures which are detached from the geometry surface. Representing complex features such as transonic shock waves which are attached to bodies offer a significant challenge. Wing tip vortices and steady trailing wakes also offer themselves to hex-dominant block insertion. Due to current limitations of the available software (discussed in Chapter III), hex-dominant meshing is not available. Therefore for the purposes of this thesis, methods will be developed which allow for feature-aligned hex-dominant mesh blocks to be inserted into the hybrid volume mesh for a range of flow features and geometries.

CHAPTER III

Governing Equations and Numerical Methods

3.1 Introduction

Computational Fluid Dynamics (CFD) is fast becoming one of the most important aspects of design processes within a range of industrial contexts from aerospace to automotive to energy and process engineering. CFD applies mathematical models of fluid flow phenomena in order to provide a numerical approximation to the governing equations of fluid motion. More specifically, CFD aims to accurately predict fluid flow behaviour, heat transfer and in some cases, chemical reactions, for increasingly realistic systems or configurations.

In the context of computational aerodynamics, the main goals in aircraft design typically include calculation of lift, drag and moment characteristics. Recent advancements in computational capabilities, numerical schemes and the mathematical models which describe the flow have allowed improved solutions to be obtained in a more efficient manner. However, a number of assumptions and simplifications persist in the models applied to solve the flow problems. This means that besides the most simple of flows, it can be difficult to achieve a high level of confidence in the final result. Despite finding increasing application in the aerodynamic design process, CFD is currently a tool which will, at best, only complement wind tunnel testing. It will be some time before CFD has matured to a stage where it will be able to replace physical experiments.

A typical CFD process usually starts with a physical description of the problem in physical space. A mesh is then generated, which is usually a division of the problem domain with a collection of non-overlapping elements. The governing fluid equations can be written in conservation and integral form, which allow both space and time to be discretised through application of the Finite Volume Method (FVM) across the mesh elements. After application of suitable boundary conditions, the resulting system of equations can be solved with numerical schemes to provide the approximate values of the flowfield variables.

The governing equations are the result of applying the fundamental laws of nature to a physical model. The fluid medium is assumed to be compressible (changes in density are not negligible) and Newtonian (stress is proportional to the time rate of strain). Another assumption made is that the fluid is a continuum, that is, the flow quantities vary continuously from one point to another. This allows the physical properties of the medium to be described mathematically as functions of space and time. The numerical methods employed throughout this project in order to provide approximate solutions to these governing equations, forms the basis of the present chapter.

3.2 The Unsteady Navier-Stokes Equations

For an arbitrary control volume V fixed in space with a differential surface boundary dS contained in a surface boundary ∂V with outer normal vector \mathbf{n} , the governing equations, known as the unsteady Navier-Stokes equations, can be written in integral and conservation form for a three-dimensional flow as

$$\frac{\partial}{\partial t} \iiint_V \mathbf{W} dV + \iint_{\partial V} [\mathbf{F} - \mathbf{G}] \cdot \mathbf{n} dS = 0 \quad (3.1)$$

where \mathbf{W} is the vector of conserved quantities and given as

$$\mathbf{W} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix}$$

and \mathbf{F} and \mathbf{G} are the inviscid and viscous flux vectors respectively

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_x \\ \mathbf{F}_y \\ \mathbf{F}_z \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_x \\ \mathbf{G}_y \\ \mathbf{G}_z \end{pmatrix}.$$

In the equations above ρ is the fluid density; u , v and w are the Cartesian components of the velocity vector \vec{V} in the x , y and z directions respectively, and E is the

total energy per unit mass. \mathbf{F}_x , \mathbf{F}_y and \mathbf{F}_z are the inviscid flux vector components in the three Cartesian directions, with \mathbf{G}_x , \mathbf{G}_y and \mathbf{G}_z representing the viscous flux vector components in the three Cartesian directions. The inviscid and viscous flux components can be decomposed in more detail as

$$\begin{aligned}
 \mathbf{F}_x &= \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uE + \rho u \end{Bmatrix} & \mathbf{G}_x &= \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{Bmatrix} \\
 \mathbf{F}_y &= \begin{Bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho vE + \rho v \end{Bmatrix} & \mathbf{G}_y &= \begin{Bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{Bmatrix} \\
 \mathbf{F}_z &= \begin{Bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho wE + \rho w \end{Bmatrix} & \mathbf{G}_z &= \begin{Bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{Bmatrix}.
 \end{aligned} \tag{3.2}$$

The inviscid flux \mathbf{F} may be further decomposed into convection and pressure terms as follows

$$\mathbf{F}_x = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uE + \rho u \end{Bmatrix} = \underbrace{\rho u \begin{Bmatrix} 1 \\ u \\ v \\ w \\ E \end{Bmatrix}}_{\text{convective terms}} + \underbrace{p \begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ u \end{Bmatrix}}_{\text{pressure terms}}$$

$$\mathbf{F}_y = \begin{Bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v w \\ \rho v E + \rho v \end{Bmatrix} = \underbrace{\rho v \begin{Bmatrix} 1 \\ u \\ v \\ w \\ E \end{Bmatrix}}_{\text{convective terms}} + \underbrace{p \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ v \end{Bmatrix}}_{\text{pressure terms}} \quad (3.3)$$

$$\mathbf{F}_z = \begin{Bmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w^2 + p \\ \rho w E + \rho w \end{Bmatrix} = \underbrace{\rho w \begin{Bmatrix} 1 \\ u \\ v \\ w \\ E \end{Bmatrix}}_{\text{convective terms}} + \underbrace{p \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ w \end{Bmatrix}}_{\text{pressure terms}}$$

where

- p static pressure;
- τ_{ij} viscous stress tensor;
- q_i heat transfer flux vector;
- E total energy.

In the above definitions, the subscripts $i, j = x, y, z$.

The viscous stress components τ_{ij} are given by

$$\begin{aligned} \tau_{xx} &= 2\mu \frac{\partial u}{\partial x} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\ \tau_{yy} &= 2\mu \frac{\partial v}{\partial y} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\ \tau_{zz} &= 2\mu \frac{\partial w}{\partial z} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\ \tau_{xy} &= \tau_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \tau_{xz} &= \tau_{zx} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ \tau_{yz} &= \tau_{zy} = \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \end{aligned} \quad (3.4)$$

where μ is dynamic viscosity, and is a function of temperature $\mu = \mu(T)$. The dynamic viscosity can be calculated using Sutherland's law which is based on kinetic theory of ideal gases:

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T_0 + S}{T + S} \quad (3.5)$$

where

μ_0 is the reference viscosity, with a value for air of $\mu_0 = 1.7894 \times 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}$;

T_0 is the reference temperature with a value of $T_0 = 288.15 \text{ K}$;

S is Sutherland's temperature with a value of $S = 110.4 \text{ K}$.

λ is the second coefficient of viscosity, given by the Stokes hypothesis:

$$\lambda = -\frac{2}{3}\mu. \quad (3.6)$$

The heat flux term q_i may be given from the thermal conductivity relation

$$q_i = -\kappa \frac{\partial T}{\partial x_i} \quad (3.7)$$

where κ is the thermal conductivity and can be expressed as a function of μ as

$$\kappa = \mu \frac{C_p}{\text{Pr}} \quad (3.8)$$

with C_p as the specific heat at constant pressure and Pr as the Prandtl number. The total energy E may be written as a function of the total enthalpy H and temperature T

$$E = H - \frac{p}{\rho} \quad (3.9)$$

where

$$H = h + \frac{|V|^2}{2} = C_p T + \frac{1}{2}(u^2 + v^2 + w^2)$$

so

$$E = C_p T + \frac{1}{2}(u^2 + v^2 + w^2) - \frac{p}{\rho}.$$

In order to close equation 3.1, the equation of state for an ideal gas, which gives the density of air as a function of the pressure and temperature $\rho = \rho(p, T)$, is required:

$$\rho = \frac{p}{RT} \quad (3.10)$$

where R is the gas constant with, for air, a value of $R = 287.05 \text{ J Kg}^{-1} \text{ K}^{-1}$.

3.3 Preconditioning

The performance of compressible codes which were originally implemented for transonic or supersonic problems will degrade as the Mach number of the flow field tends to zero. For these ‘density-based’ methods, the lower the freestream Mach number, the higher the potential for degradation of both solution accuracy and iterative convergence. It is necessary to allow flow solvers to deal with engineering problems that may involve low Mach number flows or locally compressible flows. Generally, compressible schemes without any form of modification will become impractical for Mach numbers lower than about 0.3. A typical example of such a configuration in aerodynamics is the subsonic flow over a multi-element aerofoil at high angle of attack. Whilst the Mach number of the oncoming flow is low, there exist regions in the vicinity of the aerofoil which exhibit significant compressibility effects.

Steady and transient compressible flows can be solved using time-marching schemes. However, for low speed flows, there is a large difference between the acoustic wave speed and the waves convected at fluid speed. The dominance of the convection terms in the system of equations renders the system numerically stiff leading to slow convergence^[155]. For explicit methods, the time-step must be extremely small to maintain the numerical stability of the system. In the case of implicit methods, the stiffness arises due to the wide variation in the order of magnitude of the system eigenvalues, leading to an ill-conditioned system, thus leading to a time consuming iterative process.

Preconditioning methods have been developed to alleviate the aforementioned problems associated with the numerical stiffness that appear in low Mach number flows. By pre-multiplying the time derivative term in equation 3.1 by a preconditioning matrix, the eigenvalues become rescaled to form a well-conditioned system. This allows the convergence to be accelerated towards a steady-state solution. However, this modification to the governing equation leads to it no longer being valid for transient problems. An implicit time-stepping method (dual-time formulation) allows the solution of time-dependent flows, where a fictitious pseudo-time τ is added, so that the governing equation takes the following form

$$\underbrace{\frac{\partial}{\partial t} \iiint_V \mathbf{W} dV}_{\text{physical time}} + \underbrace{\Gamma \frac{\partial}{\partial \tau} \iiint_V \mathbf{Q} dV}_{\text{pseudo time}} + \underbrace{\iint_{\partial V} [\mathbf{F} - \mathbf{G}] \cdot \mathbf{n} dS}_{\text{flux residual}} = 0 \quad (3.11)$$

where \mathbf{Q} is the vector of primitive variables and Γ represents the preconditioning matrix. The preconditioning is applied only to the pseudo-time term. During a physical time step, the pseudo time is advanced such that as $\tau \rightarrow \infty$, equation 3.1 is recovered, and the pseudo-time term vanishes. For steady-state problems, the physical time term vanishes, but the pseudo-time will remain to allow the solution to advance in the preconditioned artificial time τ .

3.4 Turbulence Modelling

The governing flow equations were presented in detail in section 3.2 which express the fundamental concepts of conservation of mass, momentum and energy. CFD is a means to simulate fluid flows by solving these equations after appropriate application of boundary conditions using numerical techniques. Solving the Navier-Stokes equations in their full form requires direct numerical simulation (DNS). Using such a method, no turbulence model will be employed and so resolution of all the spatial and temporal scales of turbulence is necessary. This soon becomes computationally unfeasible due to the excessive memory and solution time requirements, even for relatively simple flows. The high cost associated with the DNS method has led to methods such as Reynolds averaging to reduce the scales of motion through averaging of the momentum equation. The resulting set of equations is referred to as the Reynolds-Averaged Navier-Stokes (RANS) equations, which allow for use in practical engineering applications.

The averaging process results in additional terms known as Reynolds stresses, which represent the transport of momentum caused by turbulent fluctuations. The Reynolds stresses cannot be determined from first principles which is a consequence of turbulence being a property of the flow itself, rather than the fluid. The additional quantities have increased the number of unknowns in the system of equations. Since there are greater unknowns than equations, this leads to the ‘closure problem’. A range of turbulence models have therefore been developed to approximate the Reynolds stresses through the introduction of extra equations to close the system.

3.4.1 Reynolds-Averaged Navier-Stokes (RANS) Simulation

Reynolds averaging of the Navier-Stokes equations was introduced by Osborne Reynolds, which provides time-averaged equations describing an incompressible fluid. The averaging occurs by Reynolds decomposition where an instantaneous flow variable is decomposed into its time-averaged (indicated by a bar over the variable) and fluctuating quantities (indicated by a single prime), and essentially separates the turbulent fluctuations from the mean flow.

A different form of averaging is used for the compressible Navier-Stokes equations. Favre-averaging is applied in this case, which is a density-weighted averaging procedure. The instantaneous flow variable $f(x, t)$ is decomposed into a Favre-averaged term (indicated by a tilde symbol) and a Favre-averaged fluctuating term (indicated by a double prime ('')) as follows

$$f(x, t) = \tilde{f}(x, t) + f''(x, t). \quad (3.12)$$

The Favre-averaged part is given by

$$\tilde{f}(x, t) = \frac{1}{\bar{\rho}T} \int_t^{t+T} \rho(x, t) f(x, t) dt$$

where $\bar{\rho}$ is the Reynolds averaged density. The instantaneous primitive variables of the fluid flow may be decomposed as

$$\begin{aligned} u_i &= \tilde{u}_i + u_i'' \\ \rho &= \bar{\rho} + \rho' \\ p &= \bar{p} + p' \\ T &= \bar{T} + T'' \end{aligned}$$

and inserted into equation 3.1 without changing the form of the averaged variables. As a result, two extra terms are generated which leads to the closure problem described previously. The first term appears in the momentum equation and is referred to as the Reynolds stress tensor

$$\tau_{ij} = -\overline{\rho u_i'' u_j''} \quad (3.13)$$

and the second term is the turbulent heat flux vector which appears in the energy equation

$$q_{t_i} = c_p \overline{\rho u_i'' T''} \quad (3.14)$$

where q_{t_i} is the turbulent heat flux. The equation of state after averaging may be written as

$$\overline{\bar{p} + p'} = \overline{(\bar{\rho} + \rho')R(\tilde{T} + T'')} \quad (3.15)$$

so

$$\bar{p} = \bar{\rho}R\tilde{T}$$

In order to close this system of equations, a range of turbulence models have been developed so that the Reynolds stress, turbulent heat flux and mean flow variables can be resolved. Almost all turbulence models are based upon the Boussinesq eddy viscosity hypothesis, which allows the Reynolds stress tensor to be formulated as

$$-\overline{\rho u_i'' u_j''} = \mu_t \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (3.16)$$

where μ_t is the turbulent eddy viscosity and κ is the turbulent kinetic energy. To approximate the turbulent heat flux tensor q_{T_i} , Reynolds momentum heat transfer analogy is applied which yields

$$q_{T_i} = -c_p \frac{\mu_t}{Pr_t} \frac{\partial \tilde{T}}{\partial x_j} \quad (3.17)$$

where Pr_t is the turbulent Prandtl number.

3.4.2 Spalart-Allmaras Turbulence Model

In order to close the system of RANS equations, Spalart-Allmaras (S-A)^[156] proposed a one-equation turbulence model. It is one of the most successful and widely used eddy viscosity based model for external aerodynamic flow simulation. The model has good numerical stability and robustness compared with other turbulent models. It solves a single partial derivative transport equation to obtain the turbulent eddy viscosity, μ_t . The most popular form of the S-A model includes a wall destruction term which will reduce the eddy viscosity in both the log layer and laminar sub-layer. The equation is described as

$$\frac{D\tilde{\nu}}{Dt} = \underbrace{c_{b1}\tilde{S}\tilde{\nu}}_{\text{production term}} + \underbrace{\left(\frac{1}{\sigma} [\nabla \cdot ((\nu + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2] \right)}_{\text{diffusion term}} - \underbrace{c_{w1}f_w \left(\frac{\tilde{\nu}}{d} \right)^2}_{\text{destruction term}} \quad (3.18)$$

where d is the wall distance, ν is the molecular kinematic viscosity and $\tilde{\nu}$ is the modified kinematic eddy viscosity. The turbulent eddy viscosity μ_t is related to the kinematic eddy viscosity ν_t by

$$\mu_t = \bar{\rho} \nu_t, \quad \nu_t = f_{v1} \tilde{\nu}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}. \quad (3.19)$$

The definitions of the production and destruction terms depend on the version of the S-A model being used. During this project, the original S-A model was employed for all turbulent flows considered. The remaining constants and relationships for the model are as follows

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad (3.20)$$

where S refers to the vorticity which may be represented in terms of the mean rotation rate tensor Ω_{ij}

$$S = |\Omega| = \sqrt{2\Omega_{ij}\Omega_{ij}} \quad (3.21)$$

where

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{u}_j}{\partial x_i} \right). \quad (3.22)$$

The wall-blockage function f_w needed for formulation of the destruction term is

$$f_w = g \left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6} \quad (3.23)$$

with the limiter function g :

$$g = r + c_{w2}(r^6 - r) \quad (3.24)$$

and

$$r = \frac{\tilde{\nu}}{\tilde{S} \kappa^2 d^2} \quad (3.25)$$

where d is the near wall distance. The following empirical coefficients are used in the above equations:

$$\begin{aligned} \kappa &= 0.41 \\ c_{b1} &= 0.1355 \\ c_{b2} &= 0.622 \\ c_{w1} &= \frac{c_{b1}}{\kappa^2} + \frac{c_{b2} + 1}{\sigma} \\ c_{w2} &= 0.3 \\ c_{w3} &= 2.0 \\ c_{v1} &= 7.1 \end{aligned} \quad (3.26)$$

For the S-A model, the turbulent kinetic energy $-\frac{2}{3}\bar{\rho}k\delta_{ij}$ in the Boussinesq eddy viscosity hypothesis (equation 3.16) is ignored. Computation of the turbulent kinetic energy is performed in other turbulence models, such as the two-equation k - ω model. However, for the purposes of the present section, the closure of the RANS equations can be considered complete.

3.5 Discretisation and Numerical Schemes

The flow solver used for all the simulations in the present work is TAU developed by DLR^[157]. It implements a cell-vertex Finite Volume Method (FVM) to solve the RANS equations. A range of methods are available for computation of the fluxes depending on whether an upwind or central scheme has been selected. For an upwind scheme, the possible options for flux computations include Van Leer FVM, a number of variations of Advection Upstream Splitting Methods (AUSM) and also the Roe method. The central scheme is available with both scalar and matrix dissipation models.

For time accurate computations global time-stepping and dual time-stepping schemes are implemented. The global time-stepping scheme is based on the explicit multi-step Runge-Kutta method. Alternatively, a dual time formulation with first, second and third order backward difference formula can be used, where a higher order implies an increased overhead.

Particularly important for three-dimensional problems, TAU is linked with the Message Passing Interface (MPI) library which allows multiple processors to be used to solve the flow in parallel after partition of the mesh into a suitable number of domains.

3.5.1 Spatial Discretisation

The first step in the FVM is to divide the computational domain into a small number of elements which are often referred to as control volumes. The governing flow equations are discretised over each control volume using the FVM which allows them to be cast into algebraic form. The control volumes may be of arbitrary shape, but only set types of intersections between elements are allowed, with no overlapping.

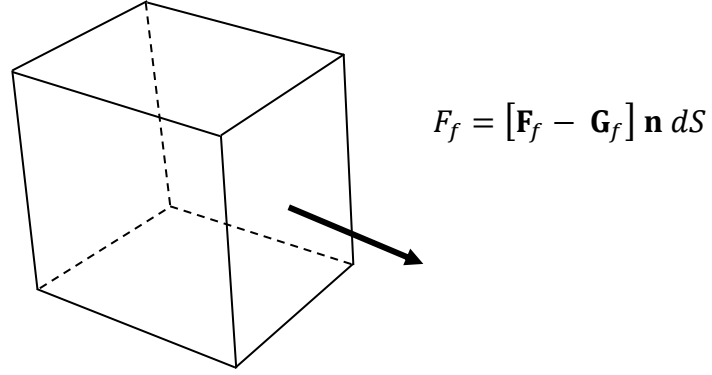


Figure 3.1: Numerical flux calculation F_f for a hexahedral cell face f

Equation 3.11 is an example of a conservation law, and these types of equations allow relations to be written which relate the volume integral to the surface fluxes of the control volume using the divergence theorem. Thus the governing equations can be integrated over each control volume and the values of the variables can be obtained. Equation 3.11 can be discretised using FVM to obtain

$$\frac{\partial(\mathbf{W}V)}{\partial t} + \Gamma V \frac{\partial \mathbf{Q}}{\partial \tau} = - \sum_{i=1}^{nf} [\mathbf{F}_i - \mathbf{G}_i] \mathbf{n}_i S_i \quad (3.27)$$

where the summation is a loop over the number of faces nf which belong to the cell. Figure 3.1 shows the flux calculation F_f for a single face f belonging to a hexahedral element where $\mathbf{n} dS$ represents the area normal vector.

3.5.2 The Primary Grid

The primary grid data (i.e. the mesh), is generated by the software SOLAR^[65], a program developed jointly by Aircraft Research Association (ARA), BAE Systems and Airbus. This industrial level meshing tool is designed and written using Object Oriented Programming (OOP) concepts with the C++ language.

For two-dimensional problems, an advancing front technique is employed to generate unstructured quad-dominant meshes. These meshes are primarily made up of unstructured quadrilateral elements, but a small proportion may be triangles. In the case of

a viscous simulation, an advancing layer approach is applied to grow highly anisotropic semi-structured quadrilateral cells from the discretisation of the bounding curves representing the geometry. The generated surface mesh is extruded by one cell in the span-wise direction to create a quasi-two-dimensional flow domain, which is necessary for the TAU pre-processing and solver codes.

For three-dimensional problems, a quad-dominant surface mesh is generated on the geometry surfaces. From this surface mesh, the boundary layer mesh is grown by advancing layers, yielding a nearfield mesh made up of hexahedral and prismatic elements. A buffer region consisting of pyramids is then generated around the prismatic nearfield mesh. This buffer region is necessary to allow a conformal interface between the prismatic nearfield and the tetrahedral farfield, which is created through the use of a Delaunay tetrahedra generation algorithm. The result is a fully conformal hybrid mesh which is suitable for the TAU flow solver.

The primary grid is used by the TAU pre-processor in order to generate all the necessary grid data for the flow solver, for example information about control volumes and their connectivity. Some of the pre-processing routines are too expensive to be carried out during an iterative calculation of the flowfield, which means the pre-processing module is run prior to the flow solver is started. Some of the routines performed by the pre-processor include checking that the surface and volume elements are orientated correctly, the generation of data structures to allow efficient access to the components of each element (i.e. points, edges and faces), and partitioning of the primary grid into domains for multi-processor solver runs.

The primary grid consists of unstructured polyhedral elements which can possess triangular and quadrilateral surfaces, which do not necessarily need to be planar. The grid must be conformal, that is, only certain element intersection cases are allowed in order to prevent hanging nodes or faces. The possible elements which are permitted to appear in the mesh are hexahedra, prisms, pyramids and tetrahedra. This flexibility is made possible by the fact that TAU uses an edge-based data structure, meaning that it can be applied on meshes of any type. Meshes made up of a mixture of different types of elements are often

referred to as hybrid grids. The philosophy behind this choice of types of elements can be summarised as follows:

- Hexahedra allow all three space directions to be resolved accurately to different length scales.
- Prismatic elements allow two directions to be resolved isotropically, with the third direction resolving in a different length scale.
- Tetrahedra allow only isotropic resolution, however they enable a wide range of adaptation schemes to be applied in order to capture anisotropic flow features.
- Pyramids allow a conformal bridge between the different elements.

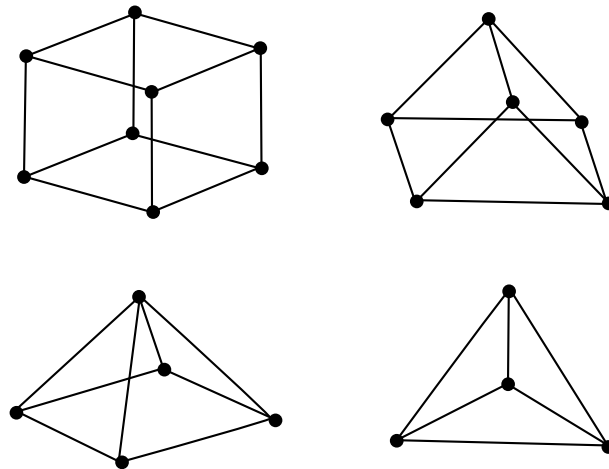


Figure 3.2: Types of elements permitted in the primary grid: Hexahedra, prisms, pyramids and tetrahedra

Related to the third point above, tetrahedra find frequent use in mesh generation due to the fact that the algorithms which produce such elements are flexible and easily automated. However, standard tetrahedral element generators will provide only isotropic elements, which are numerically diffusive in flow feature regions since none of the cell interfaces will be aligned with the flow feature, and this degrades the potential for high resolution when applying the approximate Riemann solvers. To remedy this, a wide range of anisotropic generation and adaptation algorithms for tetrahedral elements have been developed to allow for the resolution of steep variable gradients. However such adaptation schemes usually lead to highly skewed elements in the flow feature regions, which can in some cases affect the flow solver behaviour and performance.

The main benefit of using hybrid meshes is that they allow a mixture of elements to be present in the mesh. It is this property of a hybrid mesh that will be exploited in the development of feature-aligned meshes during the present project.

3.5.3 The Secondary Grid

The secondary grid is constructed from the primary grid and contains all the necessary data required for the flow solver, such as number of points, vertex information, control volume sizes, edge or face connectivity, boundary face information and so on. For the cell-vertex FVM, the flow variables are associated with the vertices of the elements of the primary grid. Therefore, the primary and secondary grids share the same points in physical space, but the secondary grid consists of control volumes surrounding each grid point.

3.5.4 Flux Discretisation

In TAU, the computation of both inviscid and viscous fluxes can be carried out with an upwind or central type scheme. The scheme and the various associated solution parameters were chosen depended on the performance of the solver for a given problem. For the majority of cases presented in this thesis, the inviscid flux solver chosen is the AUSMDV^[159] which is recognised as an improved version of the Advection Upstream Splitting Method (AUSM) as proposed by Liou and Steffen^[165]. The development of the AUSMDV came from the need to remove numerical dissipation of Van-Leer type flux vector splitting methods at discontinuities. It also eliminates the slight numerical overshoot which can appear immediately behind shock waves which are observed when using the original AUSM. For viscous flows, the AUSM has been slightly modified^[158] to improve the capability of the method to give accurate viscous flow prediction. The modifications guarantee accurate viscous shear layer resolution without the generation of spurious oscillations.

3.5.5 Higher-Order Reconstruction

If the solution is assumed to be piecewise constant across the control volume, only first order accuracy can be achieved for the inviscid and viscous flux calculations. To obtain second order accuracy, the solution can be assumed to vary in a piecewise linear fashion across the control volumes. A Taylor series expansion, based on the work by

Barth^[160] is applied for all the primitive variables in order to obtain the higher order reconstruction. The Taylor expansion is written from some local point to each of the neighbouring points, rather than metric terms such as the face normals or volumes of local control volumes as would be used in the Green-Gauss method.

Consider an arbitrary function φ which may be expressed by a Taylor expansion for a local point including the neighbouring points (in this case written for point \mathbf{P}_0 to \mathbf{P}_1)

$$\varphi_I = \varphi_0 + \frac{\partial \varphi_0}{\partial x} \Delta x + \frac{\partial \varphi_0}{\partial y} \Delta y + \frac{\partial \varphi_0}{\partial z} \Delta z + O(\Delta x^2) + \dots \quad (3.28)$$

A system of linear equations can then be written by considering the Taylor series expansion for each neighbouring point \mathbf{P}_i of the local source point \mathbf{P}_0 . For higher order reconstruction of the primitive variables, the values are extrapolated to the faces of the control volume from the points where the variables are stored leading to

$$\mathbf{Q}_F = \mathbf{Q}_P + \psi \nabla \mathbf{Q} \Delta \mathbf{r} \quad (3.29)$$

where \mathbf{Q}_F and \mathbf{Q}_P are the values on the face and points respectively, $\nabla \mathbf{Q}$ is the gradient and $\Delta \mathbf{r}$ is the displacement vector between a point of the grid and the face of the control volume. A limiter, ψ , as proposed by Venkatakrishnan^[161], reduces the scheme down to first order at discontinuities to avoid oscillations of the gradient. The only remaining unknowns are the gradients $\nabla \mathbf{Q}$ which is the subject of the following section.

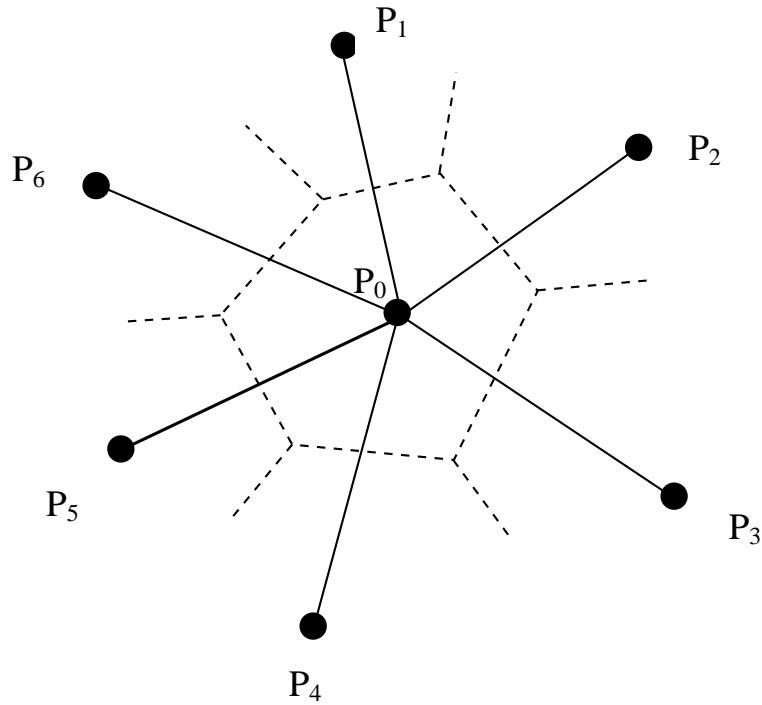


Figure 3.3: Surrounding points used for the least squares algorithm

3.5.6 Evaluation of the Gradient

For reconstruction of the gradient in both inviscid and viscous flux calculations, a least-squares method composing of a QR decomposition and Gram-Schmidt orthogonalisation is used, as first proposed by Anderson and Bonhaus^[162]. From figure 3.3 a system of linear equations for all the neighbouring points n can be constructed as follows

$$\begin{bmatrix} w_1 \Delta x_{01} & w_1 \Delta y_{01} & w_1 \Delta z_{01} \\ w_2 \Delta x_{02} & w_2 \Delta y_{02} & w_2 \Delta z_{02} \\ \vdots & \vdots & \vdots \\ w_n \Delta x_{0n} & w_n \Delta y_{0n} & w_n \Delta z_{0n} \end{bmatrix} \begin{pmatrix} \partial \varphi / \partial x \\ \partial \varphi / \partial y \\ \partial \varphi / \partial z \end{pmatrix} = \begin{pmatrix} w_1 (\varphi_1 - \varphi_0) \\ w_2 (\varphi_2 - \varphi_0) \\ \vdots \\ w_n (\varphi_n - \varphi_0) \end{pmatrix} \quad (3.30)$$

which can be written more concisely as

$$W[\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3] \nabla \varphi = Wb \quad (3.31)$$

Or

$$WA x = Wb \quad (3.32)$$

with

$$W = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}$$

The introduction of a weighting factor, which is related to the geometry, allows computation at very high cell aspect ratios. The weighting factor is defined as

$$w_i = \frac{1}{\sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}} \quad i = 0, 1, \dots, n. \quad (3.33)$$

After computation of the weights for each point in equation 3.33, the elements of the matrix A and vector b are multiplied by these values, as shown in equation 3.30 (the weighting matrix has been dropped from the subsequent equations). The solution of the matrix A requires a QR decomposition with a Gram-Schmidt orthogonalisation. Q is an orthogonal matrix $Q \in \mathbb{R}^{n \times m}$, and R is an upper triangular matrix $R \in \mathbb{R}^{m \times m}$. This solution process can be written as

$$\begin{aligned} A x &= b \quad \text{with } A \rightarrow A = QR \\ QR x &= b \\ Q^T QR x &= Q^T b \quad \text{with } Q^T Q = I, \quad I \in \mathbb{R}^{m \times m} \\ x &= R^{-1} Q^T b \end{aligned}$$

with the matrix entities defined as

$$A = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3], \quad Q = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \mathbf{q}_3], \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix}. \quad (3.34)$$

The entries of the matrix R can be computed with

$$\begin{aligned} r_{11} &= \sqrt{\sum_i^n (\Delta x_{0i})^2}, \quad r_{12} = \frac{1}{r_{11}} \sum_i^n (\Delta x \Delta y) \\ r_{13} &= \frac{1}{r_{11}} \sum_i^n (\Delta x \Delta z), \quad r_{22} = \sqrt{\sum_i^n (\Delta y)^2 - r_{12}^2} \end{aligned} \quad (3.35)$$

$$r_{23} = \frac{1}{r_{22}} \sum_i^n (\Delta y \Delta z) - \frac{r_{12}}{r_{11} r_{22}} \sum_i^n (\Delta x \Delta z)$$

$$r_{33} = \sqrt{\sum_i^n (\Delta z)^2 - (r_{23}^2 + r_{13}^2)}.$$

With the methodology described above, each component of the gradient can be computed. The reconstruction of the gradients with a least-square approach has shown to provide more accurate gradient capture and better robustness compared with the Green-Gauss theorem. Furthermore, it allows reconstruction of linear functions exactly on mixed grids.

3.5.7 Temporal Discretisation

For steady state flow problems, one of the main purposes of time stepping schemes is to accelerate the convergence to steady state. Since the flow problems that will be considered in this thesis will be steady this implies that the physical time derivative term in equation 3.11 vanishes as a time-independent solution exists. For the present work, equation 3.11 is marched towards steady state in the fictitious pseudo-time τ with a dual-time stepping scheme. A second order backward difference formula is employed to discretize the pseudo-time derivative. Both explicit and implicit schemes can be applied in order to solve the backward difference formula. TAU implements both explicit multi-stage Runge-Kutta scheme and an implicit Lower-Upper Symmetric-Gauss-Seidel (LUSGS) method^[164]. For the test cases in this thesis, the LUSGS method is chosen to provide the iterative solution.

3.6 Flow Feature Extraction

In order to extract flow features it is necessary to define criteria which allow identification of the regions in the flow domain where the flow features are present. The feature extraction is performed within the meshing software SOLAR after reading in the mesh and corresponding solution. Filters have been developed as part of this project within SOLAR which take the mesh vertices and flow information as input, performs some defined operation, and outputs the processed data set. Typically the processed data set will be smaller than the original data set, and can be piped into another filter for further processing. Sensible application of these filters can allow a gradual reduction of the total

number of data points such that a clean extraction of the flow feature(s) can be made. The extracted points are then used to create an approximate geometric representation of the flow feature topology. The flow features of interest in this work are shock waves, wakes and wing tip vortices.

3.6.1 Shock Waves

There are two approaches which have been followed in the present work for shock wave extraction depending on the flow regime.

3.6.1.1 Transonic Flows

For transonic flows in two- or three-dimensions, there are pockets of locally supersonic flow above and / or below the geometry surface. These regions of supersonic flow are terminated by a shock wave which is in close proximity to the surface. Based upon this knowledge of the physics, the approach to extract the shock wave starts with application of a filter which extracts these supersonic regions. In this simple extraction, the entire data set is processed with points possessing a Mach number greater than 1.0 output in the resulting data set. This reduces the data set down to a smaller region, reducing the memory overhead and speeding up the filtering process, but more importantly also reduces the risk of spurious points appearing in the data set allowing a clean extraction to be made.

To actually extract the shock waves from the supersonic regions, the method proposed by Lovely^[166] has been employed to identify their location and extent. Shock waves in a solution can be difficult to determine because they do not necessarily exist where the Mach number is unity. From the consideration of the geometry of a shock wave, it can be shown that the direction of the shock surface is aligned with the pressure gradient vector. The normal Mach number is defined as the Mach number in the direction of this vector. Since the normal Mach number has a value of at least one just before the shock, it can be tested for and used to extract regions where the shock front exists. The normal Mach number M_n is defined as

$$M_n = \frac{v \cdot \nabla p}{a |\nabla p|} \quad (3.36)$$

where points which satisfy $M_n = 1.0$ exist at the shock front.

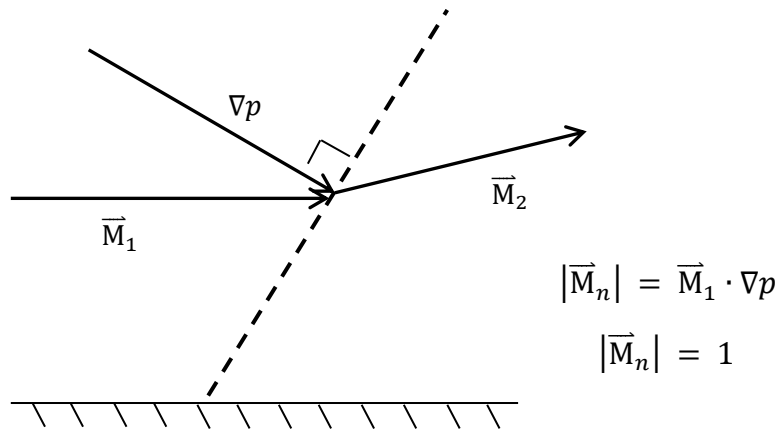


Figure 3.4: Geometry of a stationary shock wave and computation of the normal Mach number

3.6.1.2 Supersonic Flows

A different approach is required for supersonic flows since the previously described method is not suitable when the freestream Mach number is supersonic. A different sequence of filters is applied in order to extract the supersonic shock waves.

Shock waves lead to regions of high compression of the fluid. Therefore, if these compression regions can be identified and extracted, this should lead to an indication of shock waves present in the solution. To this end, an extraction technique described by Marcum and Gaither^[101] is employed to find these compression regions by considering the dot product of the normalised velocity vector with the density gradient

$$f = \left(\frac{\mathbf{v}}{|\mathbf{v}|} \right) \cdot \nabla \rho \quad (3.37)$$

where the scalar value f is computed for every vertex in the active data set.

Note that the extracted data set will also include any regions of compression that may be present at the leading edge of the geometry. Therefore, to extract only the shock

waves, the normal Mach number filter is applied on the reduced data set to obtain the points corresponding to the shock fronts only.

3.6.2 Wakes and Wing Tip Vortices

Boundary layers eventually separate from the body and trail downstream to form a wake. A loss of momentum appears in the wake region with a reduction in the velocity. The velocity profile changes with increasing distance downstream. The size and intensity of the trailing wake provide an indication of the profile drag.

Wing tip vortices appear as a three-dimensional effect of fluid travelling around wing tips. The physical mechanism for generating lift is due to the contrasting pressures on the upper and lower surfaces of the wing. The imbalance of the pressure distribution the fluid is not only responsible for the lift, but has another effect on the wing: the fluid has the tendency to curl around the tip as it is driven by the high pressure region towards the low pressure region on the upper wing surface. This causes the fluid to enter a circulatory motion and thus establish a vortex which trails downstream. The formation of these vortices creates the induced drag component of total drag.

3.6.2.1 Steady Wakes

For viscous computations it is necessary to extract trailing shear layers. There are two possible approaches which have been implemented and used for wake extraction in the current work which are applicable in both two- and three-dimensions. In two-dimensions, only steady wakes are considered at high Reynolds number which does not include unsteady wake physics such as vortex shedding. For three-dimensional flows, the physics of a finite wing mean that the trailing wake extraction will usually include the wing tip vortex embedded within the extracted data set. Other extraction techniques are required to extract the vortex core alone.

This first wake extraction approach simply considers the eddy viscosity variable calculated directly by the TAU flow solver. Any mesh points (beyond the trailing edge of the geometry) with a non-zero eddy viscosity actually give an approximate indication of the wake extent and orientation. Note that due to numerical dissipation on an original non-aligned unstructured mesh, the wake eventually disappears within a very short distance,

usually several chord lengths. However this approach usually gives enough information about the orientation of the wake for use in the feature-aligned meshing process.

Another method which can be used to extract wakes is to compute the magnitude of some flow property in all directions normal to the velocity vector

$$f = \left| \nabla \rho - \left(\left(\frac{\mathbf{v}}{|\mathbf{v}|} \right) \cdot \nabla \rho \right) \frac{\mathbf{v}}{|\mathbf{v}|} \right| \quad (3.38)$$

where the density gradient is again chosen as the suitable flow property. A lower threshold may be computed using a method described by Marcum and Gaither^[101].

3.6.2.2 Wing Tip Vortices

For the present work, the λ_2 criterion of Jeong and Hussain^[167] is used. The λ_2 criterion has been found to be successful for clean extraction of the vortex even when the farfield mesh is particularly coarse and not aligned with the vortex core. Application of this technique will extract only the vortex and not the shear layers emanating from the trailing edge of the wing.

Pressure tends to have a local minimum on an axis of swirling motion as the centrifugal force is balanced by the pressure force. Jeong and Hussain argue that simply taking the local pressure minimum is not sufficient for a general detection criterion, since well-defined pressure minima can exist in unsteady irrotational motion, which may not correspond to a vortex. It was found that this inconsistency arose from two main effects. The first is unsteady straining, which can create a pressure minimum without involving any form of swirling motion. The second is due to viscous effects, which can serve to eliminate pressure minima in vortical flows. This alternative definition therefore discards these effects in order to allow a better indication of the vortex presence. The specific derivation details can be found in the paper by Jeong and Hussain^[167], but a brief description is included here.

In order to locate the pressure minimum, the velocity gradient tensor (Jacobian) is split into its symmetric part, \mathbf{S} , the mean strain rate tensor, and its anti-symmetric part, $\mathbf{\Omega}$,

the vorticity tensor. Since unsteady irrotational straining and viscous effects are ignored, only $\mathbf{S}^2 + \mathbf{\Omega}^2$ is considered to determine the local pressure minima due to vortical motion. A vortex core is then defined as a connected region with two negative eigenvalues of $\mathbf{S}^2 + \mathbf{\Omega}^2$. Since this expression is symmetric, three real eigenvalues λ_1, λ_2 and λ_3 can be computed. If $\lambda_1 \geq \lambda_2 \geq \lambda_3$, the requirement that $\lambda_2 < 0$ will lead to the identification of the pressure minimum in a plane perpendicular to the vortex axis, and therefore the vortex core.

3.7 Mesh Adaptation Schemes

3.7.1 Construction of Anisotropic Metric for Adaptation

It remains to discuss the construction of the anisotropic metric which is used to drive the adaptation schemes employed in the current work. The anisotropic metric is based on finite element interpolation theory. The derivations of interpolation error estimates from basic principles are widely available in the literature where more in-depth discussions are available^[80, 92]. In summary, it can be shown from one-dimensional analysis that the interpolation error across linear finite volume cells is proportional to the product of the characteristic length of the mesh cell and the second derivative. For a given problem, a mesh may be considered optimised (not necessarily ‘optimal’) if it allows the error to be equally distributed across all the edges. This implies that the product should be constant, which will ensure that the mesh satisfies the equi-distribution principle^[80].

These ideas are readily extended to two- and three-dimensions. The second derivative term is now replaced by a symmetric Hessian matrix $\mathbf{H}(\mathbf{x})$. The error estimate represents the length of an edge in a Riemannian metric which can be deduced from the Hessian of some flow variable q . The tensor is made up of the following entries

$$h_{ij} = \frac{\partial^2 \hat{q}}{\partial x_i \partial x_j} \quad (3.39)$$

where \hat{q} is an approximation of the flow variable q .

The Hessian matrix can be diagonalised using the matrix of eigenvectors $\mathbf{R}(\mathbf{x})$ and the diagonal matrix of eigenvalues $\mathbf{\Lambda}(\mathbf{x})$

$$\mathbf{H}(\mathbf{x}) = \mathbf{R}(\mathbf{x})\mathbf{\Lambda}(\mathbf{x}) \mathbf{R}^T(\mathbf{x}). \quad (3.40)$$

The Hessian is also required to be symmetric and positive-definite by the length of a curve in the defined Riemannian metric. This is achieved by taking the absolute values of the eigenvalues as follows

$$\mathbf{M}(\mathbf{x}) = \mathbf{R}(\mathbf{x}) |\mathbf{\Lambda}(\mathbf{x})| \mathbf{R}^T(\mathbf{x}) = \mathbf{S}\mathbf{S}^T \quad (3.41)$$

where $\mathbf{M}(\mathbf{x})$ is the symmetric positive-definite matrix that defines the Riemannian metric, since the matrix is a function of the position vector \mathbf{x} , with a transformation $\mathbf{S} = \mathbf{R}\sqrt{|\mathbf{\Lambda}(\mathbf{x})|}$. The ultimate goal is to construct the mesh in this space such that all the elements are equilateral. A unit circle (sphere in three-dimensions) in the metric \mathbf{M} would be an ellipse (ellipsoid), rotated through an angle α , with its principal axes possessing lengths inversely proportional to the square roots of the eigenvalues. In three-dimensions the transformation will yield three separate eigenvalues. Figure 3.5 shows the stretching applied due to the transformation \mathbf{S} in two dimensions, where only two eigenvalues will be present.

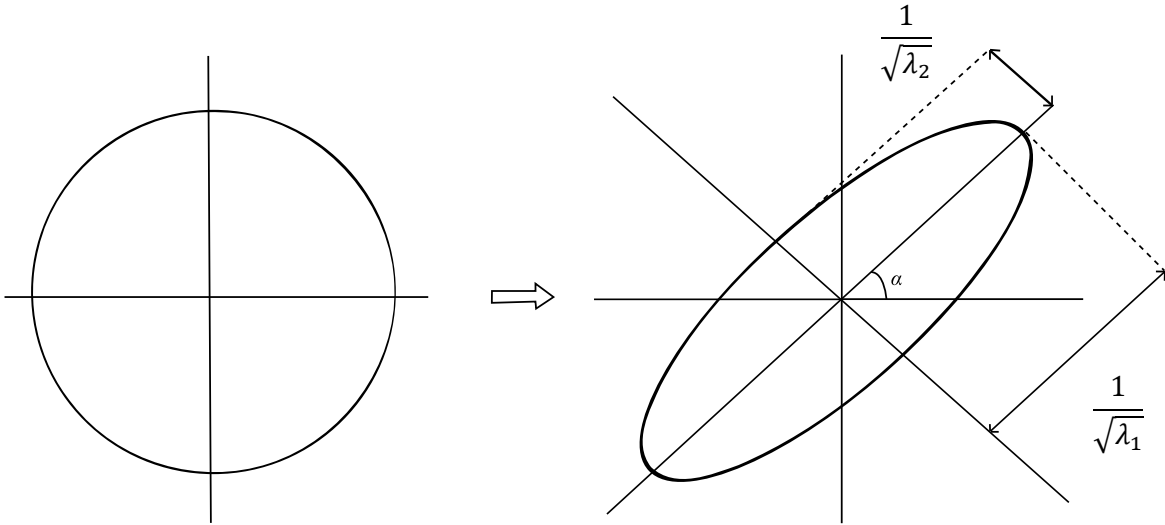


Figure 3.5: Mapping of a unit circle to an ellipsoid in the metric space where the two spacings act parallel and perpendicular to the axis rotated by an angle α

The modified Hessian $\mathbf{M}(\mathbf{x})$ is computed and stored on a background mesh which serves as the initial mesh for each adaptation loop. In the present work, the Hessian matrices are computed and stored for each node of the current mesh. The error estimates could also be defined on a much coarser background mesh, with the values for each mesh vertex of the current mesh interpolated from this background mesh. A decision was made to simply store the Hessian values at each mesh vertex to remove the need for the interpolation calculation. However, this of course leads to higher memory requirements with the requirement of the Hessian components being stored at every mesh vertex.

From elemental differential geometry, the length of a parametric curve d in the Riemannian space is given by

$$d = \int_0^1 \sqrt{s'(t)^T \mathbf{M}(t) s'(t)} dt \quad (3.42)$$

where $s'(t)$ is the derivative of the line s about t , with $t \in [0, 1]$. For an edge of an element in the mesh, this length becomes

$$d = \int_0^1 \sqrt{(\mathbf{x}_2 - \mathbf{x}_1)^T \mathbf{M}(t) (\mathbf{x}_2 - \mathbf{x}_1)} dt \quad (3.43)$$

with the value d representative of the error along the edge in the Riemannian metric, and calculated for each edge of an element in the mesh. The error estimate will have been equi-distributed when the error estimate is the same across all edges in the mesh, i.e. the length of the mesh edges in the transformed space are equal.

In order to compute d the method described by Castro-Diaz et al.^[95] is used. It is shown that the value can be calculated from

$$d = \frac{2}{3} \frac{l_0^2 + l_0 l_1 + l_1^2}{l_0 + l_1} \quad (3.44)$$

where

$$l_i = \sqrt{(\mathbf{x}_2 - \mathbf{x}_1)^T \mathbf{M}(x_i) (\mathbf{x}_2 - \mathbf{x}_1)}, \quad i = 0, 1. \quad (3.45)$$

For the present method, the entries of the Hessian matrix which are used to compute the Riemannian metric are computed by twice applying a least squares method to compute the gradient of the flow variable \hat{q} . Equation 3.43 can then be numerically evaluated for each edge in the mesh.

3.7.2 Nodal Movement

Using the error estimate described in section 3.7.1 as a guide, the nodal movement strategy aims to redistribute the nodes to increase the mesh density in the regions of highest error. This should provide a more equal distribution of error across the edges in the adapted mesh, and thus provide improvements to the accuracy of the numerical solution. The anisotropic metric that is defined through the error estimate allows the mesh cells to become stretched in the physical space (as the mesh sizing approaches unity in the transformed metric), to give high aspect ratio cells in the regions of the solution where the gradient of the flow variables is high.

The nodal movement algorithm makes use of a spring based analogy (see figure 3.6). The mesh is considered as a network of springs which possess a stiffness constant that is proportional to the edge-based error estimate. Interpreting the situation as an energy minimisation problem, the ideal position of the vertices may be calculated. This analogy also readily extends to three-dimensions.

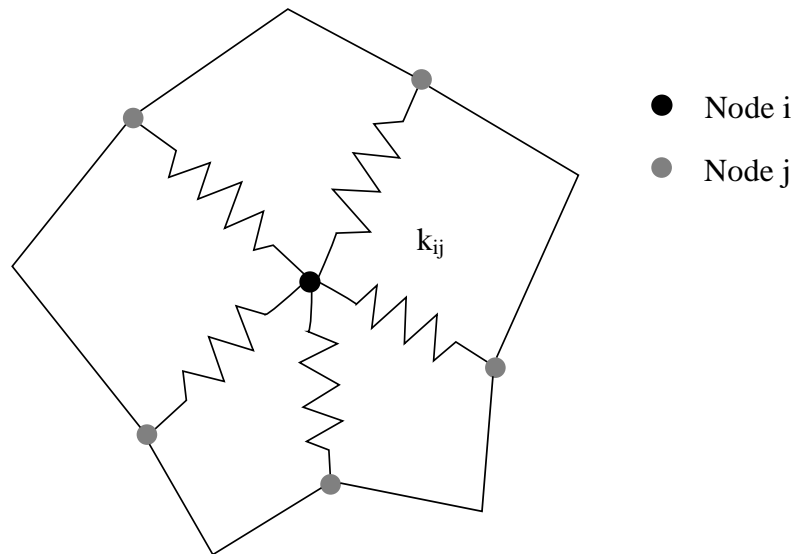


Figure 3.6: Spring analogy for a node in the mesh

The stiffness constant can be calculated by dividing equation 3.43 by the corresponding edge length in the Euclidean metric. This is computed for each edge connected to the current node i

$$k_{ij} = d / \sqrt{(\mathbf{x}_2 - \mathbf{x}_1)^T (\mathbf{x}_2 - \mathbf{x}_1)} \quad (3.46)$$

By considering the neighbouring vertices j of the current vertex i , the position of node i is updated according to

$$x_i^{m+1} = x_i^m + \omega \left[\frac{\sum_j (x_j^m - x_i^m) k_{ij}^m}{\sum_j k_j^m} \right] \quad (3.47)$$

where ω is the relaxation factor. More detailed implementation details can be given in the references^[85, 92, 93] where these methods were developed. The nodal movement adaptation is applied for the two-dimensional quadrilateral and triangular meshes considered in the thesis. A three-dimensional implementation is also investigated and applied within the hex-dominant mesh blocks to improve the mesh alignment with the flow physics.

3.7.3 Adaptation in TAU

The TAU flow solver incorporates an adaptation module which allows modification of any hybrid or tetrahedral mesh (the process will not work for two-dimensional grids). A brief description of the adaptation process in TAU is described in the present section.

With the aid of refinement sensor functions, the code determines which edges of the mesh are to be bisected, giving consideration to the desired dimensions of the resulting adapted mesh. For all edges the value of

$$I_e = \Delta V_e \|x_e\|_2^\alpha \quad (3.48)$$

is calculated, where I_e is an indicator value for the current edge e with $x_e = x_{p1} - x_{p2}$, α is an edge length scaling factor, which is set to 1, and

$$\Delta V_e = \max \left(c_{\varphi_i} \frac{\Delta \varphi_i}{(\Delta \varphi_i)_{\text{ref}}} \right), \quad \text{with } 0 \leq i < n \quad (3.49)$$

where n is the number of selected solution values for the sensor function, and c_{φ_i} is the user defined scaling value for variable φ . The maximum of all the user defined scaling values is determined by

$$(\Delta\varphi_i)_{\text{ref}} = \max((\Delta\varphi_i)_e), \forall e. \quad (3.50)$$

There are a number of different sensor functions available within the adaptation module; the sensor chosen for the present work is based on differences of the gradients of flow variables

$$\Delta\varphi_i = \left| \text{grad}(\varphi(x_{p1})) - \text{grad}(\varphi_i(x_{p2})) \right|. \quad (3.51)$$

This refinement scheme is applicable everywhere in the mesh, including the prismatic boundary layer. The schemes are used in conjunction with a separate anisotropic Hessian-based nodal movement and face and edge swapping techniques^[168-172], which are applied to the tetrahedral elements only.

CHAPTER IV

Using the Medial Axis to Represent Complex Flow Features for Feature-Aligned Unstructured Quad-Dominant Surface Mesh Generation

4.1 Introduction

Unstructured meshes generally perform quite poorly in resolving highly directional flow physics without some form of adaptation, due to the lack of alignment of the cells with any flow features present. As was identified in the literature review, the majority of mesh generation and adaptation techniques are tailored towards triangular and tetrahedral meshes, mainly due to their inherent flexibility of such elements. Development of adaptive methods for unstructured quadrilateral meshes (the default type of surface mesh generated by the meshing software SOLAR) have not received the same level of attention.

As the work is in a similar vein to the structured-block insertion method of Qin and Liu^[57], it became apparent that a method was required to force alignment of the quadrilateral elements with the flow features. This could not be achieved by simply applying a nodal movement technique, which is typically carried out for structured quadrilateral meshes. The unstructured nature of the quadrilateral cells and the requirement to maintain element topology places severe restrictions on the possible locations nodes can be moved to. The arbitrarily positioned quadrilaterals in the initial mesh are therefore problematic to align in an anisotropic fashion whilst maintaining element quality. Fortunately prior to this thesis, some fairly basic functionality existed within SOLAR to allow for curves to be embedded into the domain as a means to influence the surface mesh generation. In particular, it forced the local quadrilaterals to be aligned with the embedded curve. However, the technique was fairly limited in ability and thus offered itself for extension. A more sophisticated means to generate the curves was also required.

SOLAR allows arbitrary curves (referred to as ‘virtual geometry’) to be connected to any surface of the CAD model. The functionality was originally designed to allow curves to be connected to trailing edges of aerofoils so that during the advancing layer process, the

layers would be grown not only from the geometry, but also from the connected wake curve. This is ideal for cases where trailing wakes are required to be resolved, which are usually captured very poorly on standard unstructured quad-dominant meshes. The curve would be connected to the trailing edge and defined towards the farfield boundary. Furthermore, the curve could be easily orientated with the true direction of the wake, determined from the numerical solution.

Whilst this is acceptable for features such as wakes, a different approach is required for flow features such as shock waves. In this case, it is not necessary or desirable to have a refined mesh of the same order as the boundary layer or wake mesh. Therefore, modifications were made as part of the current project to allow the curves representing shock waves to be treated differently, so that the advancing layers were not grown from these curves.

In processing the input geometry prior to mesh generation, SOLAR recognises closed boundary loops made up by the geometry curves. This functionality was altered such that lines in isolation can also be identified. A further step was required to prevent the advancing layer mesh to be grown from the isolated curve. A list of curves is defined which specifies the curves from which the advancing layer mesh will be grown from. Any isolated curves are appended to an ignored zones list and will be ignored by the layer generator. Whilst the curves are now ignored, their presence is still acknowledged and respected by the advancing front algorithm. This results in the unstructured quad-dominant mesh being forced to be locally aligned with the curve(s).

Once the dominant features have been extracted from the flow solution, they may be represented by simple virtual geometries which are then appended to the standard physical geometry which describes the model. These geometries are virtual in the sense that whilst they are present in the domain for the purpose of influencing the mesh generation, they will not appear as a physical boundary in the final mesh.

However, there are a number of issues with adopting an approach involving geometric representation of the flow features. Even for two-dimensional problems, flow features such as shock waves can possess extremely complex structures which means

attempting to generate a geometric representation of the feature can pose a serious challenge. This means that geometric representation will either not be possible, or at best require some form of user-interaction. Such issues need to be addressed if geometric representation techniques will ever be considered as viable and attractive alternatives to other adaptation techniques already established which are usually fully automatic.

It quickly became apparent in the early stages of the present project that simply applying curve fitting techniques directly to extracted point cloud data was sufficient for simple flow features, but if complexities appeared, for example the bifurcation of a shock wave, such an approach was not appropriate. This provided motivation for a new and more general approach to the problem. The idea was to make use of the medial axis which has gained popularity in other contexts^[10-13, 173-177] due to its ability to provide useful information on shape and topology.

The main subject of the present chapter is therefore to describe the process which has been developed during this project to provide a means to represent complex two-dimensional flow features as geometries. After extraction of the flow feature as a set of points in space, the ‘concave hull’ is found which provides a hollow representation of the point set. It is necessary to point out that the *concave* hull of a group of points is distinctly different to the *convex* hull. The convex hull refers to the boundary of a minimal convex set which contains a finite set of points. In other words, the convex hull envelops the entire point set, and is unique. The convex boundary does not always fully reflect the topology or shape of the point cloud, as shown in figure 4.1. From figure 4.1(b) it can be observed that the convex hull edges connect only the exterior points, and cannot take into account the ‘void’ region in the centre of the data set. In contrast, the concave hull shown in figure 4.1(c) provides a better approximation of the shape of the point cloud, identifying both interior and exterior points. However, the concave hull is not unique, hence a set of possible concave hulls will exist for a given point cloud. More details on the construction of concave hulls and the α -shape are given later in the chapter.

The concave hull can be used to construct a constrained Delaunay triangulation which allows an approximation of the medial axis to be made, which essentially provides a description of the shape and topology of the extracted point cloud and hence the flow

feature. The word ‘approximate’ is used since the construction of the initial medial axis from an unstructured point cloud will be noisy, and also different concave hulls will give slightly different constrained Delaunay triangulations and resulting medial axes. Following extraction of the medial axis, it must then be smoothed in order to be of use in the mesh generation.

The present chapter will describe each part of the process in detail starting with concave hull construction, followed by building, cleaning and decomposing the medial axis, recombining medial branches and fitting smooth polynomials through each branch. The mechanism through which SOLAR controls mesh spacing is then described. The approach is then applied to a range of two-dimensional test cases to demonstrate its ability to generate high-quality feature-aligned surface meshes.

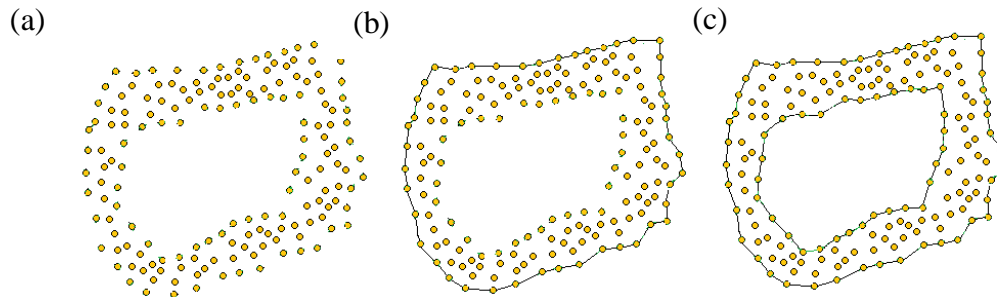


Figure 4.1: (a) Arbitrary point cloud, (b) convex hull and (c) a possible concave hull

4.2 Building the Medial Axis

4.2.1 Concave Hull Construction

After extraction of an initial two-dimensional point cloud from the solution using a feature extraction technique described in section 3.6, the concave hull may be computed using the α -shape. Edelsbrunner and Mücke^[178, 179] provided the notion and formal definition of the α -shape, which can be considered as a generalisation of the convex hull of a point set. The α -shape of a point set S defined in two- or three-dimensional space, with α defined as a real number, $0 < \alpha < \infty$, is a polytope which is neither necessarily convex nor connected. For $\alpha = \infty$, the α -shape is defined as the convex hull of S . In figure 4.1(b) the convex hull for a point set is given, and corresponds to an α -shape with $\alpha = \infty$. As α decreases, cavities start to appear in the α -shape. As $\alpha \rightarrow 0$, the α -shape degenerates to the original point set S .

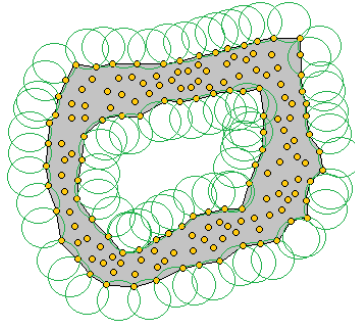


Figure 4.2: Identification of a possible point cloud through computation of the α -shape. The green circles represent the circular ‘eraser’

When α becomes small enough, a circle in two-dimensions with radius α can be present in space without enclosing any point in the set S . The shape of S is then carved out from the point cloud by this circular ‘eraser’ at all positions where it does not enclose any of the points in S . For suitable values of α , the original point set is reduced to a concave hull (see figure 4.2). This allows the points on the boundary of S to be extracted, leading to a hollow representation of the set. A range of values of α yields a family of different α -shapes for the same point cloud i.e. the concave hull is not unique. It is likely that different values of α will generate valid non-intersecting polygons to build the constrained triangulation.

A less formal and perhaps more intuitive description of the α -shape is provided by Edelsbrunner and Mücke^[178] which can aid the understanding of the α -shape. Imagine a mass of styrofoam which makes up the space \mathbb{R}^3 , containing the points which represent some other material, like rock. A sphere-shaped eraser (circular shaped in two-dimensions) then carves away all parts of the styrofoam block that can be reached without colliding into the pieces of rock. This allows regions within the interior of the point set to be carved away, and eventually an object which provides a description of the shape of the point set S , is constructed. The value of α represents the squared radius of the carving eraser. As $\alpha \rightarrow 0$, all of the styrofoam surrounding the rock pieces is carved away (since α is smaller than the distance between the pieces of rock and can move freely to carve away styrofoam without collision) and the α -shape degenerates to the original point set. On the other hand, as $\alpha \rightarrow \infty$, the eraser becomes too large and it can no longer be moved between points to carve away styrofoam on the interior (since α is larger than the distance between points, the

eraser cannot be moved freely without colliding into them), leading to the convex hull of the point set S .

In terms of the algorithm to compute the α -shape, a Delaunay triangulation is constructed for the entire point set. The α -shape is then defined as the union of all the triangles whose circumradius is less than the given value of α . After this operation, the facets which are referenced by a single triangle (i.e. an edge on the boundary of the shape will belong to only one triangle) are found. As output, the algorithm returns a list of ordered pairs of points which can be used to form edge segments which represent the boundary of the point set, if a suitable value of α has been chosen.

In the present work, in order to compute a suitable value of α , the average and maximum edge lengths are computed from the underlying mesh structure which makes up the point cloud. An initial guess for α is obtained by considering the average edge length and gradually increasing α in small increments, up until the maximum edge length, until a valid non-intersecting polygon is formed. It is likely that increasing α far beyond the maximum edge length will cause essential shape information to be lost. For each computed α -shape, a check is made to ensure it forms a non-intersecting polygon. If a suitable polygon has been constructed, the process will stop.

4.2.2 Medial Axis Approximation

The medial axis of a domain is defined as the locus of the centres of the circles of maximum radius that can be inscribed inside the domain^[180]. Based on this definition, the medial axis or the skeleton of a polygonal domain may be found from its boundary discretisation. A constrained Delaunay triangulation may be constructed where the only vertices present are the points of the edges making up the polygon boundary. The medial axis may be approximated by joining the circumcentres of all the triangles within this Delaunay triangulation. For the current application, the boundary of the polygon will not be smooth and therefore the medial axis will also be non-smooth. In particular, a variety of triangles with varying sizes and internal angles can appear in the triangulation which leads to inconsistent circumcentre location. This is the cause for the jagged appearance of the initial unprocessed medial axis. The medial axis requires to be smoothed in some way in order to allow it to be of use in the mesh generation step.

Properties of the underlying Delaunay triangulation which defines the medial axis approximation may be exploited in order to split the medial axis into its constituent branches. Each branch is then fitted with a smooth polynomial curve.

4.2.3 Constrained Delaunay Triangulations

Considering a constrained Delaunay triangulation, one may define three types of triangle which each possess properties that can be exploited in processing the medial axis. A similar notation as presented by Frey and George^[4] is used. The three types of triangles are simply classified based upon the number of boundary entities they possess. The three possible cases are (see figure 4.3):

- *Type-0* triangle: Three bounded edges (yellow). These are defined as triangles which have all three edges bounded by adjacent triangles. The circumcentre of this particular triangle is a junction node - it represents the start of three branches in the medial axis approximation.
- *Type-1* triangle: Two bounded edges (blue). These are defined as the triangles which have only two edges bounded by adjacent triangles, with the third edge part of the polygon boundary. Successive combinations of these types of triangles contribute to the main branches of the medial axis approximation.
- *Type-2* triangle: One bounded edge (red). These are defined as the triangles which have only one edge bounded by an adjacent triangle, with two edges part of the polygon boundary. The circumcentre of such a triangle is defined as a termination node – this indicates the end point of a medial axis branch.

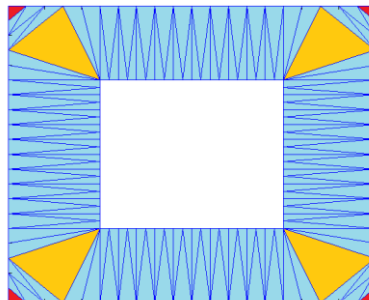


Figure 4.3: Types of triangle in a constrained Delaunay triangulation, *Type-0* (yellow), *Type-1* (blue) and *Type-2* (red)

The above definitions provide some properties which may be exploited to automatically process the medial axis. By itself, the approximation of the medial axis is not particularly of much use in the representation of flow features. However, using the definitions described above, the medial axis may be broken down into its corresponding branches to allow further processing. As the point cloud is extracted from an unstructured mesh, it is unlikely that the splitting process can be carried out straight away. Prior to the splitting of the medial axis, it must be cleaned to remove false junction and termination nodes which may have been generated in the constrained triangulation. Small branches which do not represent important shape detail are also removed.

The medial axis processing algorithm takes as input constrained triangulation data and circumcentre coordinates. Whilst only the circumcentres are necessary to construct the medial axis, the underlying triangulation is required to construct a suitable data structure to allow queries to be made about the number of boundary entities for each triangle; which in turn provides a means to clean the medial axis and then decompose it at the branch junctions.

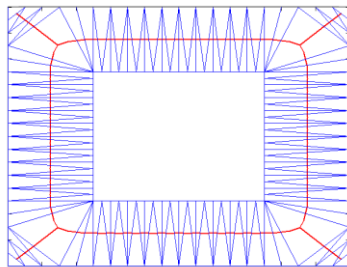


Figure 4.4: Example constrained triangulation and medial axis approximation for a simple shape with smooth boundaries

4.2.4 Cleaning the Medial Axis

4.2.4.1 False Junction Nodes

Since the point cloud is extracted directly from an unstructured mesh, the polygon boundary will be non-smooth leading to a chance of *Type-0* triangles being constructed which do not represent true junction nodes. These false triangles can be easily identified, since one of its adjacent neighbours will be of *Type-2*, highlighted as green triangles in figure 4.5. True junction nodes belong to *Type-0* triangles with three bounded edges which

are bounded by *Type-1* triangles. By considering the neighbouring triangles of each *Type-0* triangle, if a *Type-2* triangle is encountered, it is removed from the triangulation leading to a new *Type-1* triangle being defined. The triangulation and neighbour information is updated to reflect this change. This process is summarised in algorithm 4.1.

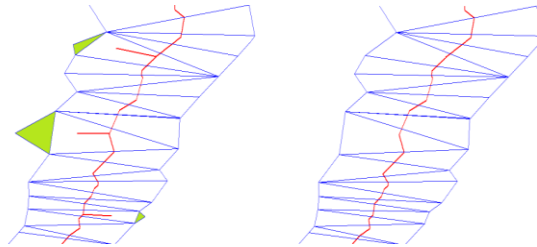


Figure 4.5: Cleaning the medial axis

4.2.4.2 Small Branch Removal

Other small branches may also occur in the medial axis. These usually appear in regions where the polygon boundary approximates a convex corner. At a convex corner, a *Type-0* triangle will always appear, indicating the junction of three branches as usual. One of these branches will converge towards the convex corner point. Figure 4.4 shows an example of this occurring for a smooth polygon. This shape features four sharp convex corners which lead to *Type-0* triangles appearing at the locations where the medial axis changes direction. For a non-smooth boundary, an example of which is displayed in figure 4.6, these triangles may also appear where the boundary approximates a convex corner. These small branches do not represent important shape detail and their inclusion in the polynomial fitting step is unnecessary.

Although the medial axis behaviour in specific regions of a shape has been identified, it can be difficult to determine whether the small branch converging into a convex corner requires to be removed for an irregular polygon. For a simple smooth shape it is clear which these branches are since convex corners are easier to identify. However for the current application, the formation of the medial axis depends entirely on the underlying point cloud and the resulting α -shape, which is in turn dependent upon the original mesh density. The randomness of the point cloud leads to a variety of triangles with varying internal angles being generated in the constrained Delaunay triangulation. The unpredictability and inconsistency of these angles for different problems means that

identifying convex corner regions (see figure 4.6) in the triangulation is extremely difficult. In the smooth square ring case displayed in figure 4.4, the termination triangles at convex corners can easily be seen to be 90° . This cannot be guaranteed to be true for a flow feature α -shape polygon, where the angle between the two unbounded edges of a termination triangle is not necessarily less than or equal to 90° . Therefore, even if the branch does appear to converge into a convex corner, an angle test is not guaranteed to work. Currently, a method to successfully determine such regions has yet to be implemented.

In order to remove the small branches, they can be recognised by their relatively low branch count. For the cases tested in the present work, this approach appeared to work satisfactorily. In the event that an important shape feature is missed however, two options have been implemented. Firstly, the ‘trimming’ value can be modified by the user to change the size of the branches removed. Secondly, the small branch removal process can be switched off entirely, which means every single branch of the medial axis decomposition is output. Whilst this may mean some user-interaction is then required to check the decomposition (or the user can let the process continue with all medial branches), the unwanted branches can be easily visualised and removed since they are simply lists of points.

Algorithm 4.1: Cleaning the Medial Axis (pseudo code)

```
function medialAxisClean()
for each triangle t
    if(type (t) == type0)
        for each neighbouring triangle nt
            if(getType(nt) != type1)
                deleteTriangle(nt)
                updateTriangulation()
            end if
        end for
    end if
end for
```

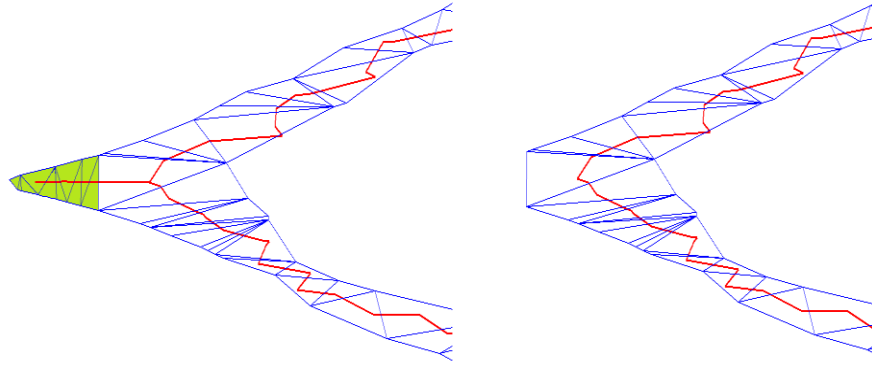


Figure 4.6: Removal of small branches from medial axis

4.2.5 Splitting the Medial Axis

The algorithm considers all *Type-0* triangles (the junction nodes) and will travel along each branch storing all the circumcentres of *Type-1* triangles, until either a termination node or another junction node is encountered. The algorithm keeps a check on circumcentres already considered to prevent duplication of branches, in the event of multiple junction nodes appearing in the medial axis which are connected to the same branch. The branch identification process is demonstrated in figure 4.7. The algorithm starts at triangle jt which represents a junction point (*Type-0* triangle), and is bounded by three neighbouring *Type-1* triangles ($nt01$, $nt02$ and $nt03$). The neighbouring triangles of $nt01$ are $nnt01$ and $nnt02$. The triangle index of jt is the same as $nnt01$, and since jt will have already been checked, this means $nnt01$ has already been checked. $nnt02$ is therefore added to the list of points for that particular branch. These checks are made while the number of boundary entities of a triangle is equal to two. The effect of this is to allow the algorithm to travel along each branch, storing all the unchecked circumcentres, until it encounters either a junction or termination triangle.

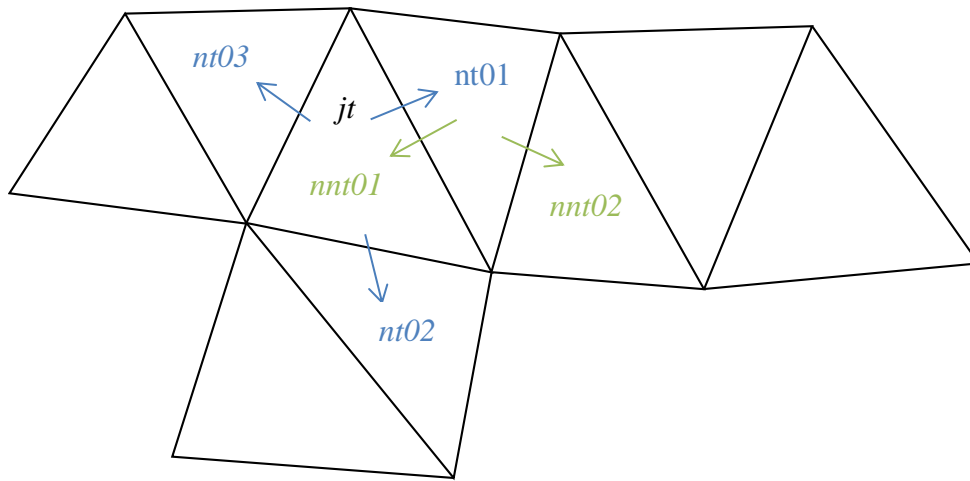


Figure 4.7: Neighbouring triangle query to allow travel across each medial branch (see Algorithm 4.2)

Medial branches may also be merged together if their recombination will lead to a higher quality curve being generated (i.e. it makes more sense to represent two or more branches of the medial axis as a single curve, rather than a number of piecewise curves). These aspects are considered in more detail in the following subsections.

4.2.6 Recombining Medial Axis Branches

It may be desirable in some cases to recombine adjacent branches of the medial axis, as this can allow smoother representation of the flow feature. The algorithm uses the identified junction points of the medial axis branches in order to perform the check.

The neighbouring junction or termination nodes of each junction node are found from the neighbour information constructed from the underlying triangulation. Vectors pointing from each junction node to its neighbours are then determined, as displayed in figure 4.8. A junction point is the start of three branches in the medial axis approximation, and therefore three vectors are calculated. These vectors give an indication of the direction the medial axis branches will emanate from each junction node. Next, for each junction node, the angles between each vector are computed. Three angles are therefore associated with each junction node.

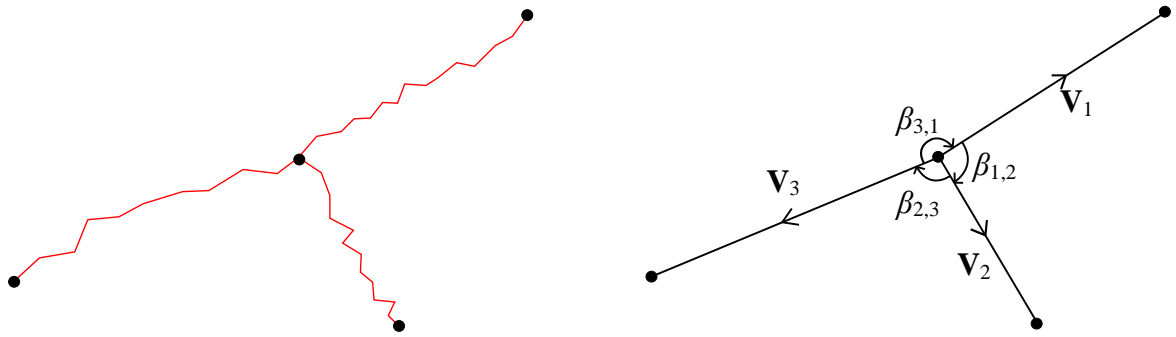


Figure 4.8: Branch recombination: comparison of vectors from junction points to surrounding junction / termination points. Angle $\beta_{3,1}$ falls within the desired range; branches belonging to vectors \mathbf{V}_1 and \mathbf{V}_3 are merged to form a single branch

Algorithm 4.2: Splitting the Medial Axis (pseudo code)

```

function medialAxisSplit()
for each junction triangle jt //type-0 triangles
  for each neighbouring triangle nt
    //travel along branch 2 bounded edges -> 2 neighbours (type-1 triangle)
    //while loop will terminate when type-0 or type-2 triangle encountered
    considered [ nt ] = true
    while ( numNeigh[ nt ] == 2 )
      get indices of neighbouring triangles nnt01, nnt02
      if ( considered[ nnt01 ] )
        /* triangle nnt01 has already been considered, so get circumcentre of
        nnt02 */
        add circumcentre of nnt02 to branch list
      else if ( considered[ nnt02 ] )
        add circumcentre of nnt01 to branch list
      end if
    end while
  end for
end for

```

As the angle between two branches approaches 180° , this increases the likelihood that the two branches are suitable candidates for recombination. A check is made to see which of the three angles is the largest, and that it is in the range $160 - 220^\circ$ (this range can be modified by the user if required). If the angle meets this criterion, then the two branches are combined as a single branch. To combine branches, the lists of circumcentre points obtained from the medial axis split are merged together. Generally if the angle is acute then curve recombination is not appropriate since the small angle represents an abrupt change in the direction of the medial axis.

It should be noted that there will only ever be a maximum of one merge per junction point. Also an individual branch can be merged to a maximum of two other branches, since the maximum number of junction nodes which bound an individual branch is two.

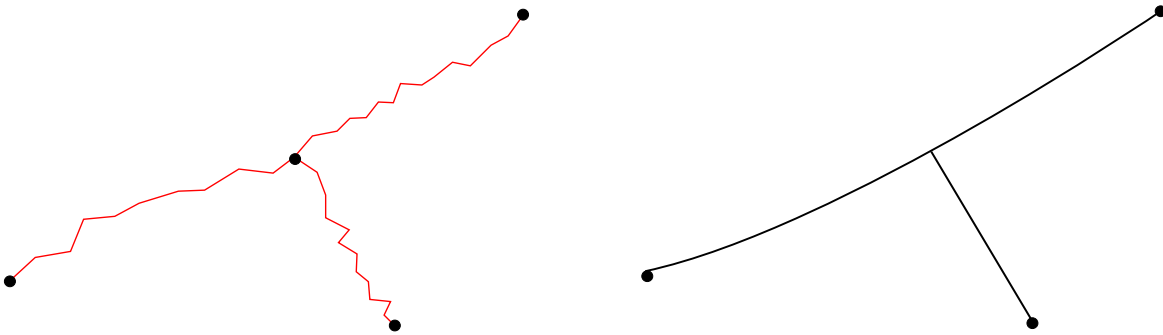


Figure 4.9: Smoothed medial axis after recombination of branches (see Figure 4.8) and polynomial fitting for each remaining branch

4.2.7 Fitting Polynomials

After the automatic decomposition of the medial axis into its constituent branches, smooth curves can be fitted through the resulting points. Polynomial coefficients for each branch can be calculated with a simple least squares approach. In order to fit suitable curves through the points, a goodness of fit value is calculated for each branch curve.

Initially each branch is fitted with a linear polynomial. If the R^2 value falls below a certain value, (for example, fitting a linear curve through a curved branch which would be better suited by a quadratic or cubic polynomial) then the order of the polynomial is increased and a new fit is made until the R^2 value meets the specified requirements. R^2

values close to 1.0 indicates a high quality polynomial fit through the medial axis branch. An example of the branch curve fitting for an arbitrary shape is shown in figure 4.9.

Algorithm 4.3: Recombining Medial Branches (pseudo code)

```
function recombineBranches ()
/* loop over junction points and compute vectors pointing to neighbour junction / termination points */
for each junction triangle jt
    for each neighbouring junction or termination triangle nt
        compute and store vectors from jt to nt
    end for
end for

/* compare angles: the index that vec1 is stored in corresponds to the same index as branch 1 and so on,
meaning a comparison between vec1 and vec2 is comparing the angle between branches 1 and 2 */
angle_1 = compareVectorAngles (vec1,vec2)
angle_2 = compareVectorAngles (vec2,vec3)
angle_3 = compareVectorAngles (vec1,vec3)
if( in_range (angle1) )
/* combine branches which correspond to vec1 and vec2 i.e. branch1 and branch2 */
    combinePointLists (branch1,branch2)
else if( in_range (angle2) )
    combinePointLists (branch2,branch3)
else if( in_range (angle3) )
    combinePointLists (branch1,branch3)
end if
```

A least-squares approach is used which allows the topology of the medial axis branch to be represented by a simple mathematical expression. For a set of data points (x_i, y_i) , the resulting expression is a linear combination of any m specified functions of x , where $m-1$ is the order of the polynomial. Generally, the form of the least squares model can be given by

$$y(x) = \sum_{k=0}^{m-1} a_k X_k(x) \tag{4.1}$$

where $X_0(x), \dots, X_{m-1}(x)$ are the basis functions. For the present work, a simple linear combination of increasing powers of x were used as the basis functions

$$y(x) = a_0 + a_1x + a_2x^2 + \dots + a_{m-1}x^{m-1}. \quad (4.2)$$

The least-squares problem is then to compute suitable values for a_k such that a polynomial is generated which best represents the given data set. The points across the resulting curve are used as the locations for the application of background sources, and this information is appended to the mesh generator input files prior to regeneration of the mesh.

4.3 Controlling Mesh Spacing

To control the mesh spacing in the domain, SOLAR uses the concept of background sources which are a set of basis functions that provide a continuous distribution of mesh spacing throughout the domain. Essentially, these functions allow control of the mesh spacing and in some cases the orientation of the elements at any location in the computational domain; whether that be on the surface of the geometry or in the field surrounding the geometry. There are three different types of sources available to be used in the generation of a mesh. These are point sources, line sources and triangle sources. Each type of source has a different effect on the formation of the mesh. Given a Cartesian background grid, the spacing parameters specified at each source is propagated to the nodes of this background grid. During the actual generation of the mesh, the spacing parameters are interpolated to determine the local mesh size. All the sources are taken into account during the mesh generation in order to determine the new element created. This is achieved through the interpolation of all the spacing parameters from the sources and the Cartesian background grid. An intuitive analogy of these mesh sources is that they act rather like heat sources conducting through a continuous medium.

Point sources are essentially piecewise linear radial basis functions, with a spacing value s , and two radii which define the source strength r_1 and r_2 with $r_1 < r_2$. The strength of a source controls its intensity – determining how far the effect of a source is felt in the field. Consider a point source defined at a point \mathbf{p} , then the spacing s at an arbitrary point \mathbf{x} is

$$s = \begin{cases} s_0, & r < r_1 \\ s_0 \left(1 + \frac{r-r_1}{r_2-r_1}\right), & r > r_1 \end{cases} \quad (4.3)$$

where $r = \|\mathbf{x} - \mathbf{p}\|$. The spacing remains constant for the distance r_1 and will start to decay linearly beyond this, and the rate of decay is such that the mesh spacing has doubled at the distance r_2 . Point sources can also be set to decay in an exponential fashion as follows:

$$s = \begin{cases} s_0, & r < r_1 \\ s_0 \left(b - 2 + b^{\frac{r-r_1}{r_2-r_1}}\right), & r > r_1 \end{cases} \quad (4.4)$$

where b is a user-defined exponent base. Figure 4.10 shows graphically how point sources typically work to influence the mesh spacing across a region in the domain with several different types of spacing philosophies.

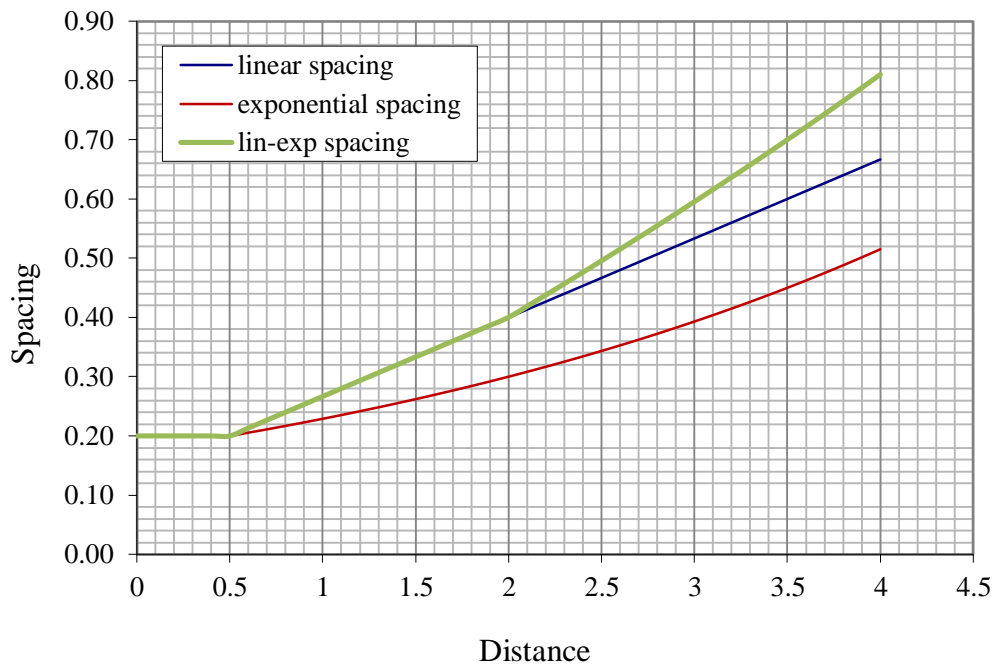


Figure 4.10: Mesh spacing decay for linear, exponential and linear-exponential spacing philosophies

Line sources are defined by two point sources. These sources are evaluated by projecting a point \mathbf{x} onto the line segment (i.e. the closest point in the line to point \mathbf{x}) joining the two positions of the two point sources, and then computing a point source at that point, constructed by linearly interpolating the source spacing and strength between the two segment end points (see figure 4.11). Spacing defined by triangle sources are evaluated in a similar manner, with the point \mathbf{x} projected to the triangle with the necessary quantities interpolated from the nodal values defining the triangle source.

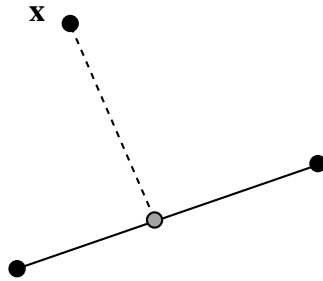


Figure 4.11: Line source with the interpolation of the spacing for a point \mathbf{x}

Line sources can also be defined as anisotropic, in which case a stretching value is calculated and the elements will become elongated along the spacing direction with the specified spacing value. The anisotropic spacing distribution requires a mechanism to make the evaluation of a source dependent not only on \mathbf{x} , but also on a direction \mathbf{d} . Anisotropic line sources are evaluated by applying this stretching to the usual spacing value. Let S represent the stretching value, with the angle α defined between a point \mathbf{x} and the direction of the source line \mathbf{d} . The stretching S is simply defined as the reciprocal of the sine of this angle α . Multiple sources are required to control the mesh density, but the final spacing at some point in the domain is computed as the minimum of all the sources which may influence that particular region.

The sources are also applied across the final flow feature curves in order to influence the spacing of the mesh in the flow feature regions. Using anisotropic line sources provides local regions of high-quality mesh which almost mimic the attributes of a structured mesh, despite still being defined as a globally unstructured mesh. In contrast with anisotropic adaptation on triangular meshes, the process avoids the generation of

highly skewed elements in the flow feature regions which can affect the performance of some flow solvers^[57].

4.4 Global Mesh Refinement

Mesh refinement studies are carried out in order to obtain grid independent solutions. Not only is it important for grid sensitivity studies to be carried out when performing a CFD calculation, but also turbulence models can often give varying shock locations compared with experimental observations. There is a degree of uncertainty with experimental data due to errors in measurement, and the simulation must usually use corrected values for the freestream conditions, for two-dimensional cases. It was therefore decided to compare the solution obtained from the feature-aligned adaptive solution with the grid independent solution of the particular mathematical model used to solve the flow, rather than with experimental data. This section briefly describes how the global mesh refinement on unstructured quad-dominant meshes is carried out using the background sources.

As described in section 4.3 sources possess a spacing value and two radii which represent the source strength and influence the distance over which the mesh spacing will decay. Figure 4.12 shows a typical set up of sources for a simple two-dimensional aerofoil, with an example of global refinement in figure 4.13. The lengths of the arrows are indicative of the distance across which the particular spacing value will exist. At the leading and trailing edges two point sources exist which influence the mesh spacing in these regions. The spacing is much smaller than those of the surrounding sources, which increases the density of the mesh points generated at the leading and trailing edges. A line source exists to influence the distribution of points on the aerofoil surface in the streamwise direction. Another source influences the density of points in the wake region, and finally a global point source influences the global point density. All these sources will affect the mesh density in an isotropic manner.

To modify all the sources for a given problem, a simple program was written to read in the spacing file and modify each spacing value by multiplying by a constant factor which is set by the user. The radii of influence for each source remain constant. This allows the

user to easily refine or coarsen the mesh without having to manually modify each source. The use of this spacing-modification routine is particularly necessary for three-dimensional problems where the spacing file may contain several hundred sources.

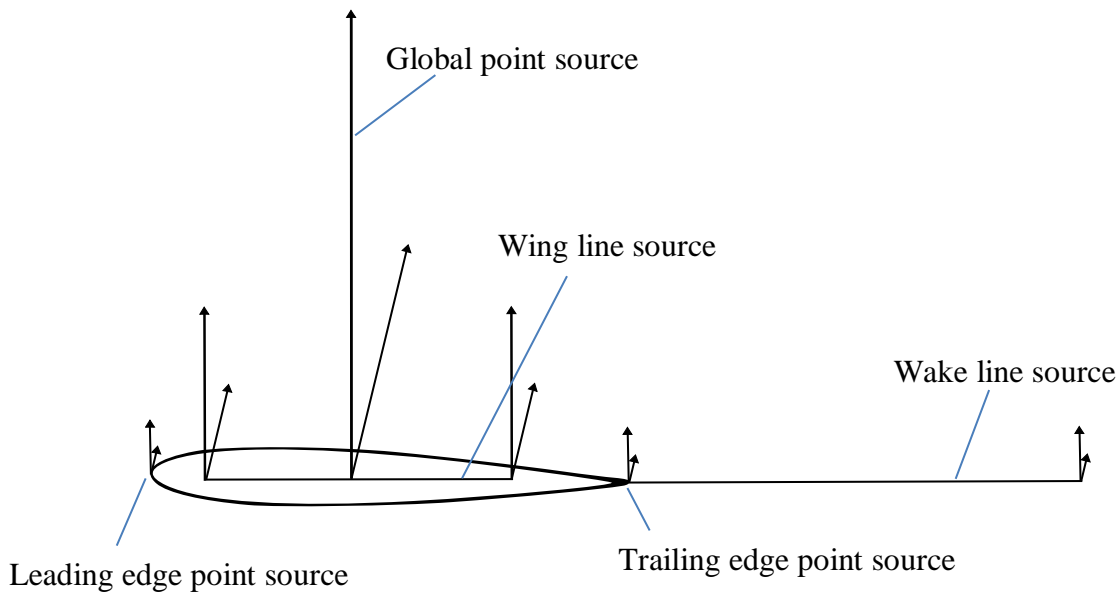


Figure 4.12: Example sourcing philosophy for simple aerofoil case

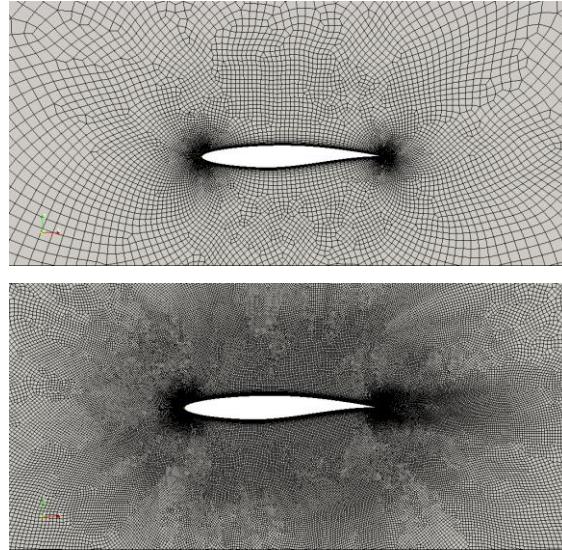


Figure 4.13: Two-dimensional surface global mesh refinement example

Boundary layers were generated such that valid y^+ values were obtained when solving on all meshes. Once a suitable first cell height had been tested and yielded suitable y^+ values, this first cell height was applied for all subsequent mesh generation. As

mentioned in Chapter III, the turbulence model of choice was the Spalart-Allmaras model for all viscous computations, requiring a y^+ value of approximately 1.0.

4.5 Identifying Multiple Regions of Points

Another important consideration, particularly in the development of an automatic process, is the separation of multiple flow features extracted with the same feature detector. For example, in a supersonic flow around an aerofoil, there are likely to be a number of shock waves present, including a strong bow shock as well as shock waves emanating from the trailing edge due to flow recompression. The filter which extracts these shock waves will only return a collection of points which belong to all of the shocks detected. There is no direct way of telling which point belongs to a particular shock wave.

One possible method to address this problem involves considering the connectivity of the underlying mesh to determine which points are physically connected to each other. A simple algorithm has been implemented such that these separate regions of points can be identified, and this allows each region to be dealt with separately in the feature representation step. Note that the developed algorithm presented here has been designed to work only in two-dimensions, but it may lend itself to extension to a three-dimensional version.

This advancing front style algorithm works by initially finding the faces in the mesh which are made up by the extracted points. Using this face format makes navigating the different regions less complicated and also has the added benefit of removing any spurious unconnected points which may exist. Starting with the face stored in the first element of the original face list, the bounding faces of this face are found. If any of these bounding faces are part of the original face list, they form a face front which allows propagation throughout a region by considering all the bounding faces of the faces in the current front. The considered bounding faces, provided they are in the original list, form the updated front and these faces are flagged as part of that particular region.

Each face that has been visited is removed from the original face list. The propagation of the front through a region continues until the front contains no faces. When

the front is empty, it is an indication that all the faces in that particular region have been checked, and there are no new faces for that region. If the original face list is not empty, the process restarts with the face index now stored in the first element of the original face list. The algorithm continues until the original face list is empty, which indicates all faces have been visited and there are no more regions to consider.

A diagrammatic display of the algorithm working on a simple region of faces is shown in figure 4.14. The blue region indicates the original collection of unvisited faces, and the current face front is coloured orange. A visited face is coloured green. It can be seen how the front propagates throughout the region of unvisited faces until all faces forming the extracted region have been checked.

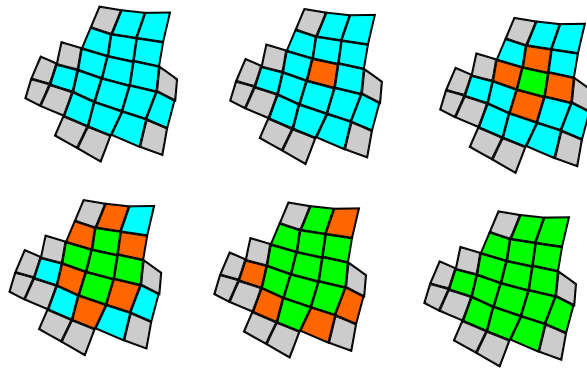


Figure 4.14: Process to identify a region of connected points in two-dimensions

The algorithm provides an effective way to automatically identify separate regions. It will detect every single region provided an initial front can be formed. Small regions of points which cannot form an initial front or possess a low face count are considered as noise and discarded.

4.6 Mesh Adaptation

4.6.1 Determining Local Mesh Spacing

Since the mesh is required to be regenerated after the feature curves have been created, it made sense to use the error estimate and attempt to determine a better mesh spacing in important regions around the geometry. In order to define a more optimal mesh for a given degree of accuracy, the approach described by Peraire and Peiro^[114] is adopted. Here it is desirable to generate a mesh where the root mean square error for each element is

distributed along the local eigendirections. From these considerations, it can be shown that the spacing values at each point are

$$\delta_1^2 |\lambda_1| = \delta_2^2 |\lambda_2| = C \quad (4.6)$$

where C is a user-specified positive constant. The nodal spacings in the eigendirections can be calculated from

$$\delta_i = \sqrt{\frac{C}{\lambda_i}}. \quad (4.7)$$

The value of therefore C influences the number of elements in the regenerated mesh, which is set to 1.0. In the present process, the mesh is regenerated only once, and from that point on the mesh node numbers are fixed. The nodal movement algorithm then seeks to redistribute those points to more appropriate locations in the domain to reduce the error.

For the existing point sources, as presented in figure 4.12, each source possesses a radius of influence which is set to r_I , a value which is determined heuristically. The spacing values are computed for every node which lies within this radius of influence. This yields, based on equation 4.7, two spacing values for each node. The average of the minimum spacing values δ_i for the group of nodes within a particular radius of influence is calculated, and this new constant spacing value is applied to that particular point source (each source in SOLAR may possess one spacing value only). The radius values remain unchanged throughout this process (i.e. identical to those used in the original mesh generation). All the existing point and line sources used are isotropic, and therefore will influence the local mesh spacing in an isotropic manner.

The process is applied only to the sources which affect the resolution of the mesh near the geometry, i.e. at leading and trailing edges, as well as the source which affects the streamwise distribution of points. Application to the sources which affect the resolution in a more global fashion is unsuitable as it can create excessively fine meshes, and therefore these sources remain unchanged. Whilst this approach was followed for the aerofoil cases presented, a distinct disadvantage quickly became apparent when using sources in this fashion - unfortunately the spacing mechanism in SOLAR was not particularly designed with the potential for linking with error estimates in mind. Furthermore, whilst mesh

resolution can be improved in local regions of the geometry, e.g. at the leading edge, due to large eigenvalues being computed (and hence small spacing according to equation 4.7) the number of points in those regions is not necessarily optimal. For anisotropic flow features, in its current implementation, the spacing cannot be applied along the direction of the eigenvectors to align the mesh; instead the flow feature curves are necessary to give the guidance of the mesh direction, with local spacing applied along the curves computed from equation 4.7. It is the author's opinion that it would be useful if the sourcing philosophy was modified in some way to allow the natural anisotropy of the solution to be taken into account. Furthermore, the development of refinement and coarsening strategies within SOLAR would also be desirable to compliment the sourcing approach.

For anisotropic flow features, the spacing values are computed for all the nodes which have been extracted based on the particular feature extraction technique. The spacing value is calculated as before. The sources applied across the feature curve were not present in the original mesh generation, and therefore suitable radius values are supplied based on the particular flow feature under consideration. These sources are also set to be anisotropic (see section 4.3) such that the mesh is locally aligned with the feature curves when the mesh is regenerated with anisotropic cells.

In some cases, the final shock location can be sensitive to the mesh resolution. In order to ensure the shock wave is still captured within the high quality region of mesh, the radius values are set such that the structured-like region of mesh exists over a fairly wide area. In the event that the shock position does slightly move away from the original extracted shock location, the refined region is large enough such that there is still a high quality local mesh which can then be adapted to the final shock location.

The success of an adaptation scheme comprising only of nodal-movement is strongly related to the topology of the original mesh. The initial feature-alignment step aims to give a better starting point for the node movement, by approximately aligning the mesh with the flow features. It essentially reduces the amount of work the adaptation scheme is required to do and improves element quality in the adapted mesh. A standard quad-dominant mesh could also be adapted, but the quality of the adapted cells is likely to be very poor. This is particularly true for unstructured quadrilateral meshes, where there are

greater restrictions on element quality than for triangular meshes (e.g. avoiding non-convex elements).

4.7 Splitting Bad Quadrilateral Elements

Due to the unstructured nature of the quadrilateral elements in the mesh, it is likely that bad elements are formed as no restrictions are placed upon the movement of the nodes during the adaptation. Typically these bad elements are ‘chevron’ quadrilaterals, i.e. non-convex. This type of element is unacceptable in the mesh and will lead the flow solver TAU to reject the mesh during an element quality check, meaning no further solution can be obtained. Rather than place restrictions on node movement to prevent these elements from forming, the implemented approach is to split these non-convex quadrilaterals into triangles. Usually only several quadrilateral splits occur during the entire adaptive process. However if a large number of splits occur, it is usually an indication that the relaxation factor ω (see section 3.7.2) has been given a value which is too large.

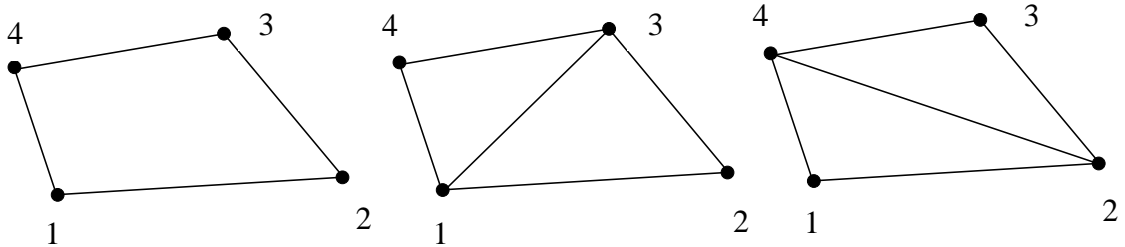


Figure 4.15: Splitting of quadrilateral elements

In order to determine if a quadrilateral is non-convex, four separate triangles can be defined by considering the two possible orientations of the diagonal. For a simplex k the surface area S can then be simply computed using the following expression

$$S_k = \frac{1}{2} \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix}. \quad (4.8)$$

If all four surface areas are positive then the quadrilateral is convex. If any of the area calculations returns a negative value, then the quadrilateral is non-convex. This quadrilateral is then split into two triangles by inserting an edge along the diagonal of the quadrilateral that resulted in the two triangles with positive surface areas. The mesh is then

updated with the new element face and the data structures (which are edge-based) required for the nodal movement adaptation are updated.

4.8 Test Cases

4.8.1 Double Wedge Supersonic Inviscid Flow, $M_\infty = 3.0$

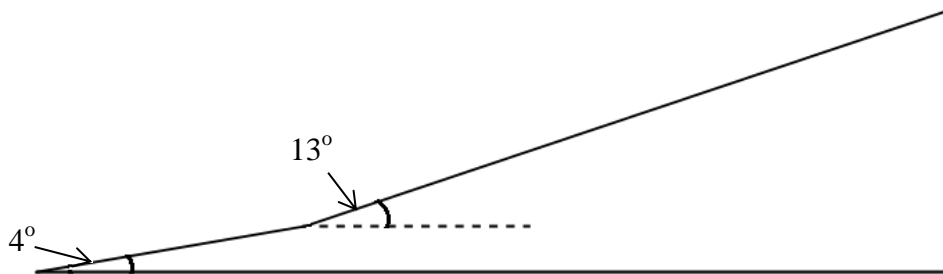


Figure 4.16: Schematic of the double wedge geometry

The geometry for this case is presented in figure 4.16 (not to scale). As the flow reaches the double wedge the reduction in the flow area causes the shock waves to appear, one for each part of the wedge. These two shocks then coalesce and the resulting final shock structure is quite complex. However, the medial axis process can efficiently and automatically deal with this complexity to allow the final shock curves to be obtained. After extraction of the structure using the normal Mach number computation, the medial axis process can begin. Figure 4.17 shows the initial mesh and pressure contour solution. The raw unprocessed medial axis and identification of junction and termination points is displayed in figure 4.18. After the medial axis process, the final smoothed feature curves are displayed in figure 4.19. In this case the medial axis process identifies three separate branches around a single junction point. The junction point appears where the shock waves have intersected. Three termination points are also identified. The angle between two of the medial branches emanating from the junction point falls within the desired range, and therefore the two medial branches (highlighted in figure 4.18) are automatically detected as candidates for recombination. The algorithm also determines that simple linear polynomials are sufficient to represent each branch based on the computed R^2 values.

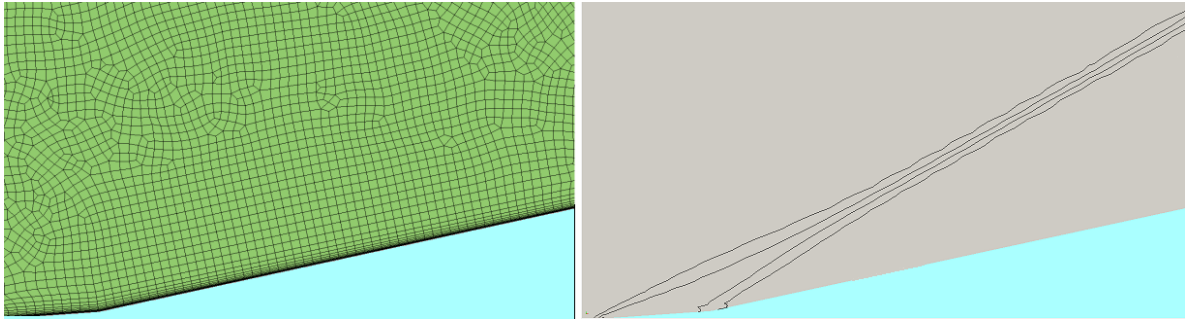


Figure 4.17: Initial mesh and corresponding pressure contours indicating shape of shock wave

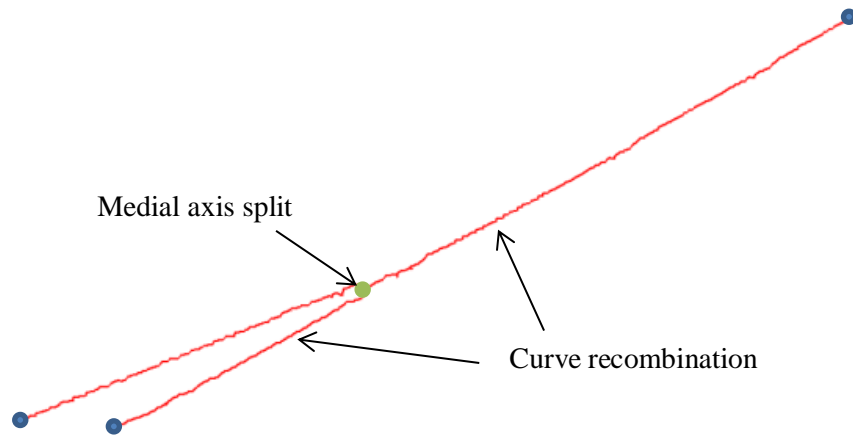


Figure 4.18: Extracted medial axis of shock wave with identification of junction (green) and termination (blue) points. The split at the junction point leads to three separate branches being defined, after which two can be recombined based on the angles between the branches to form a single branch

After application of anisotropic background sources across the smoothed curves, the mesh may be regenerated. The feature-aligned mesh is displayed in figure 4.20, which also includes a zoomed view of the mesh in the shock region to clearly show the high-quality anisotropic cells which may be generated using the current approach. Figure 4.21 shows the resolution of the shock structure using pressure contours. It can be observed that the feature-aligned mesh provides much sharper resolution of the shock wave compared with the original mesh.

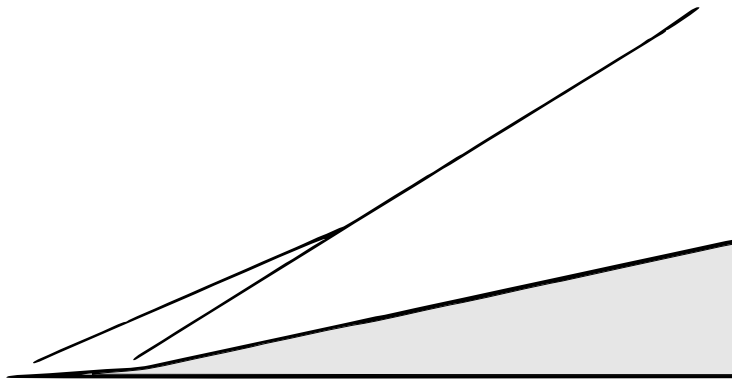


Figure 4.19: Processed medial axis curves appended to original geometry

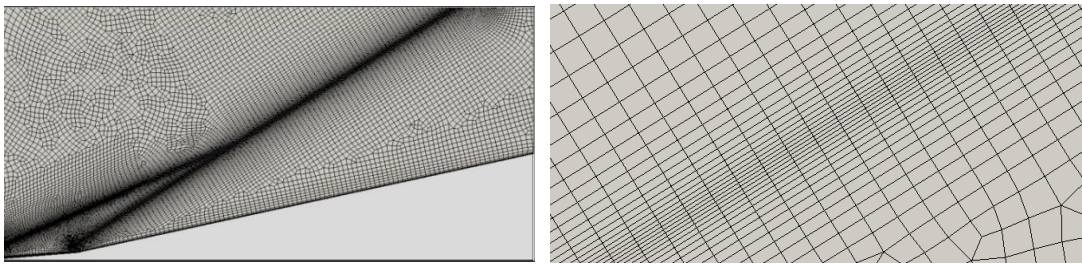


Figure 4.20: Feature-aligned mesh and zoom of high quality mesh in shock region

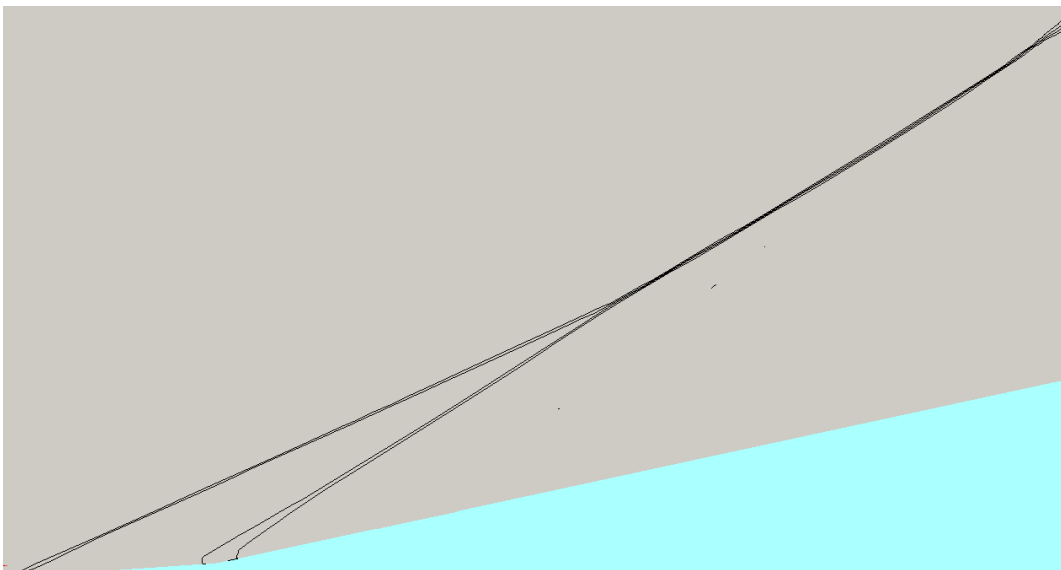


Figure 4.21: Pressure contours on feature-aligned mesh

4.8.2 NACA0012 Aerofoil

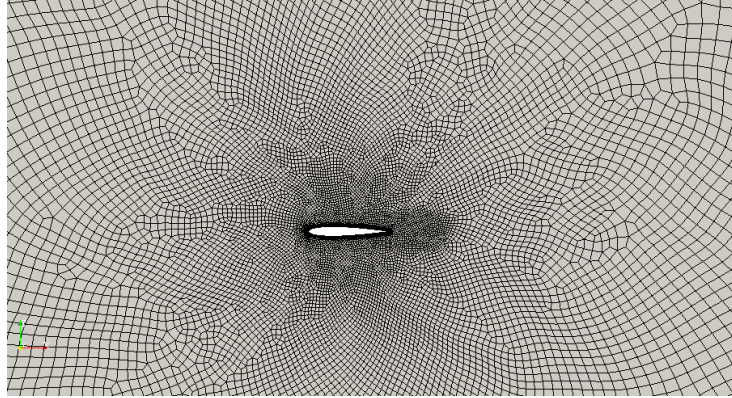


Figure 4.22: Original mesh used for the NACA0012 cases

Three separate flow conditions are simulated using the original mesh shown in figure 4.22 to obtain the initial solution. For the presented cases in this section, the only flow features of interest are shock waves.

4.8.2.1 Inviscid Transonic Flow: $M_\infty = 0.75$, $\alpha = 1.25^\circ$

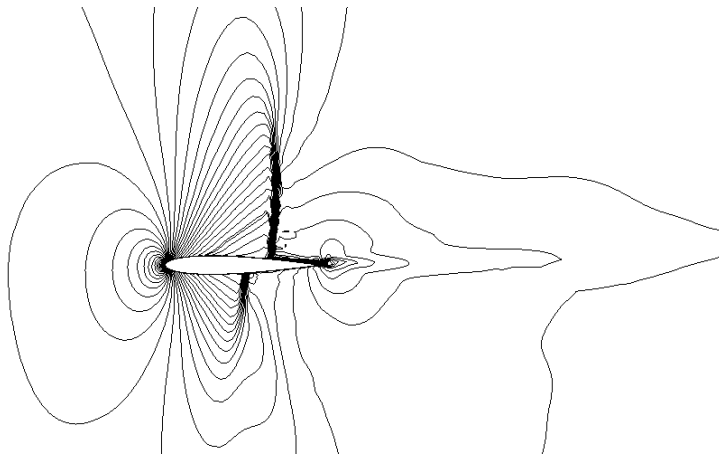


Figure 4.23: Original mesh Mach number contours

The flow features present in the solution for this case are two shock waves, one on both the upper and lower surfaces of the aerofoil. The original solution is displayed in figure 4.23. It can be seen that the shock waves are relatively simple in shape. Despite the simplicity of the flow features, this case does allow demonstration of the connectivity algorithm, described in section 4.5, since multiple shock waves exist in the domain. After

application of the normal Mach number filter, an array of points which satisfied the normal Mach criterion is returned. There is no way to directly determine from this list which points correspond to either shock wave. The connectivity filter automatically determines this information and groups the points according to which shock region they lie in, assigning each point a region ID. Figure 4.24 shows how the points have been grouped, the different colours representing a separate group of points.

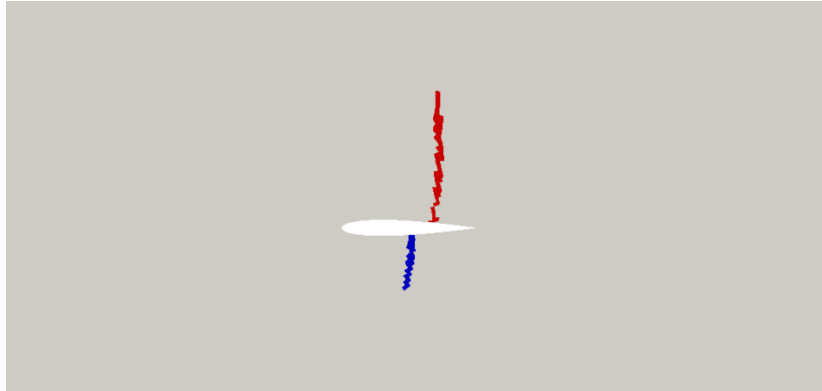


Figure 4.24: Extracted shock waves coloured by region ID

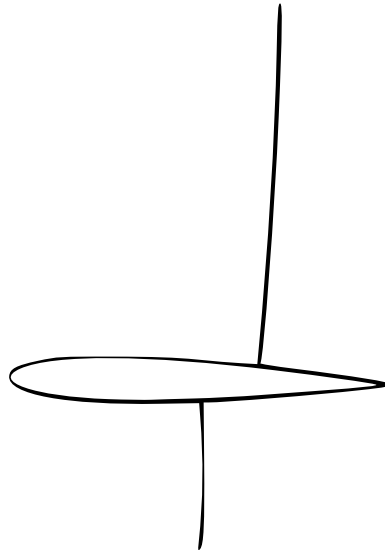


Figure 4.25: Shock curves embedded into domain

The medial axis approach is applied on each group of points in order to generate the shock curves. Due to the simplicity of the shock topology, single curves are generated for each shock wave which can then be embedded into the domain, as displayed in figure 4.25 (note that the shock curves are not connected to the original aerofoil geometry). Anisotropic background sources are applied across the shock curves, new spacing values are computed

for the existing sources and the surface mesh is regenerated to yield the initial feature-aligned mesh (figure 4.26(a)). The original solution can then be interpolated onto the new mesh, and the solution continues to evolve. Since the feature-alignment step can only guarantee approximate alignment with the shock wave, the nodal movement adaptation scheme is applied based on the Hessian of the Mach number in order to shift the mesh nodes towards the shock waves.

The resulting mesh is fully aligned with both shock waves in figure 4.26(b), leading to an improved resolution of the Mach number contours compared with the original mesh, displayed in figure 4.27 and 4.28. In particular, figure 4.28(a) shows that the shock is excessively smeared across many cells. The non-aligned cells increase the numerical diffusion in the shock region. In comparison, the feature-aligned adaptive solution in figure 4.28(b) provides much sharper shock resolution across the width of a couple of cells due to the near perfect alignment of the mesh cells.

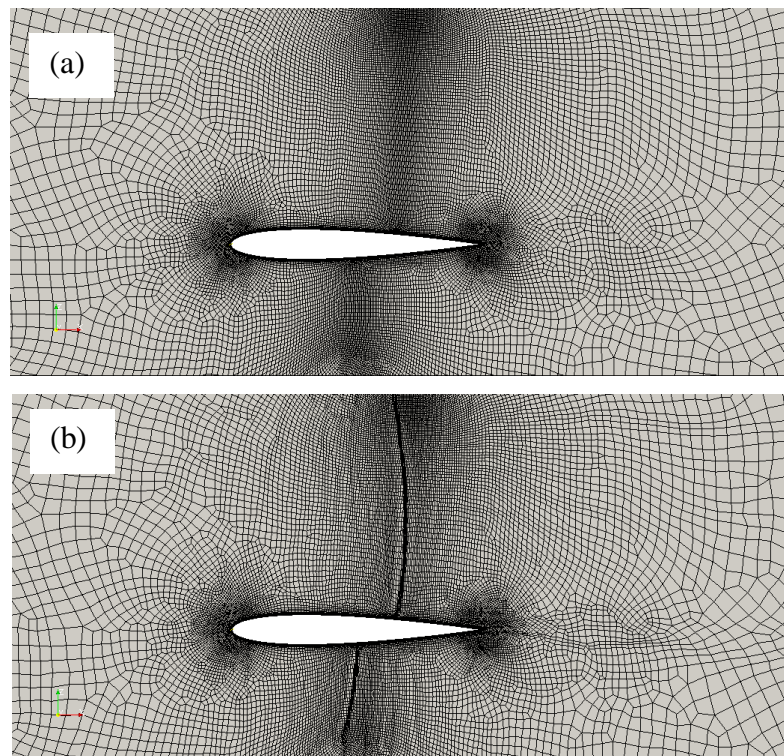


Figure 4.26: (a) Initial feature-aligned mesh and (b) final feature-aligned adaptive mesh

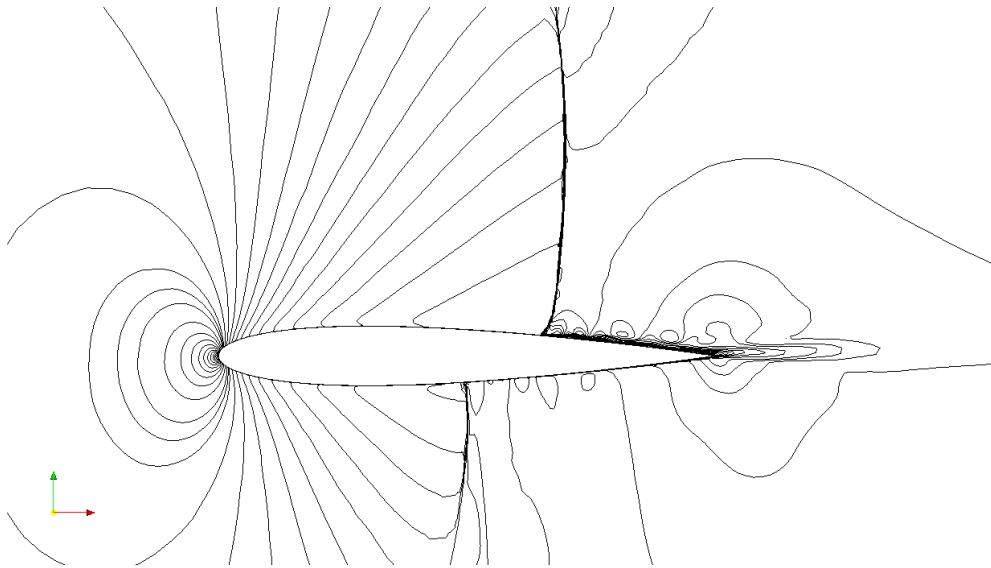


Figure 4.27: Mach number contours for feature-aligned adaptive mesh

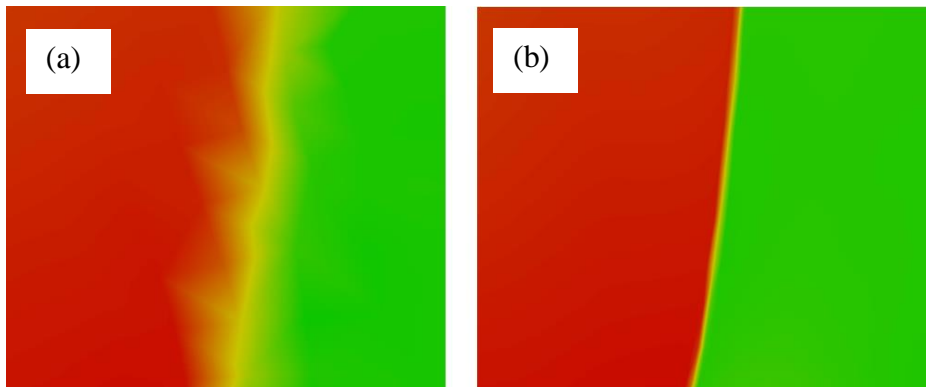


Figure 4.28: Filled Mach number contours - zoom of shock region for (a) original mesh and (b) feature-aligned adaptive mesh

4.8.2.2 Inviscid Transonic Flow: $M_\infty = 0.95$, $\alpha = 0^\circ$

The flow conditions for this case correspond to case AGARD03^[193]. This case presents more of a challenge for the geometric representation process due to the complex structure of the flow feature – this particular shock wave is often referred to as a ‘fishtail’ shock. The current approach deals with this complexity such that the shock curves can easily be extracted for the feature-alignment process. The connectivity filter is not required for this case since only one group of points is detected.

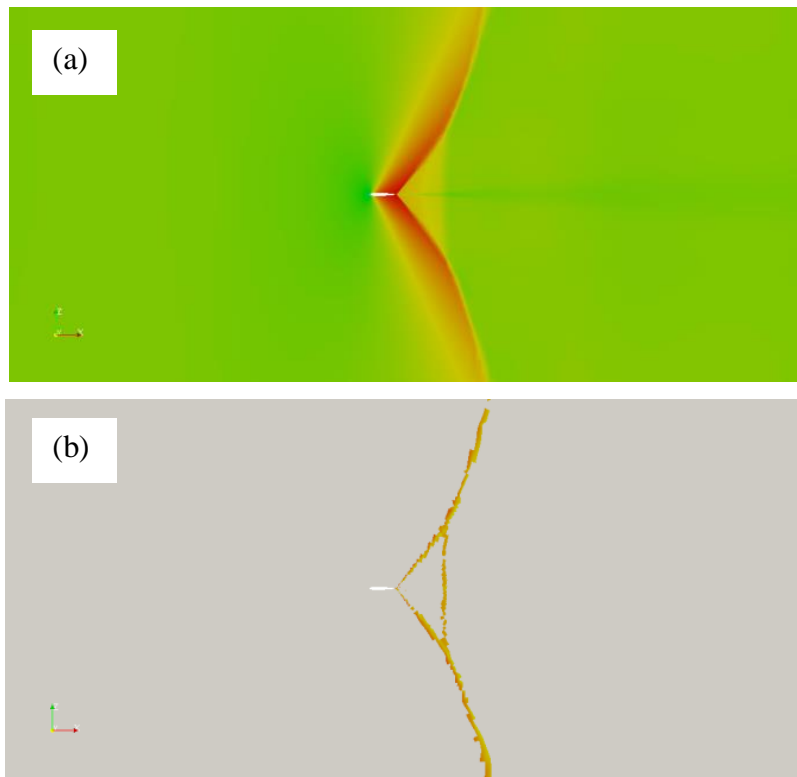


Figure 4.29: (a) Original Mach number solution and (b) extracted shock wave

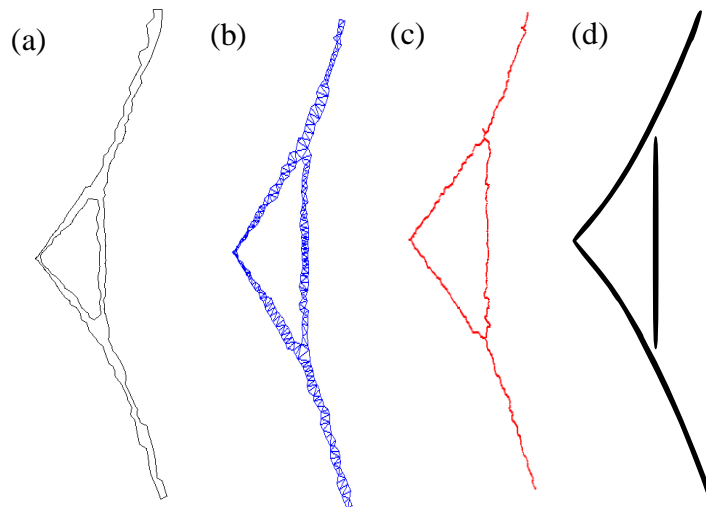


Figure 4.30: Medial axis process applied to the fishtail shock structure, (a) concave hull, (b) constrained Delaunay triangulation, (c) unprocessed medial axis and (d) final smoothed medial axis

The original solution and extracted fishtail structure is shown in figure 4.29. The concave hull polygon, constrained triangulation, unprocessed medial axis and final

smoothed curves after application of the medial axis process are displayed in figure 4.30. For this case, after the medial axis cleaning is complete, three junction nodes are identified with two termination points. Two pairs of these branches are identified as candidates for recombination after consideration of the angles between the medial branches, resulting in a total of three branches for the polynomial fitting. The medial axis processing algorithm starts with linear polynomials for all three curves but increases the polynomial order for two of the branches to improve the R^2 value of the fit. The resulting quadratic polynomials can be seen to fit the branches well, as indicated in figure 4.30(d). A linear fit is deemed sufficient by the algorithm for the normal shock which appears downstream of the aerofoil trailing edge.

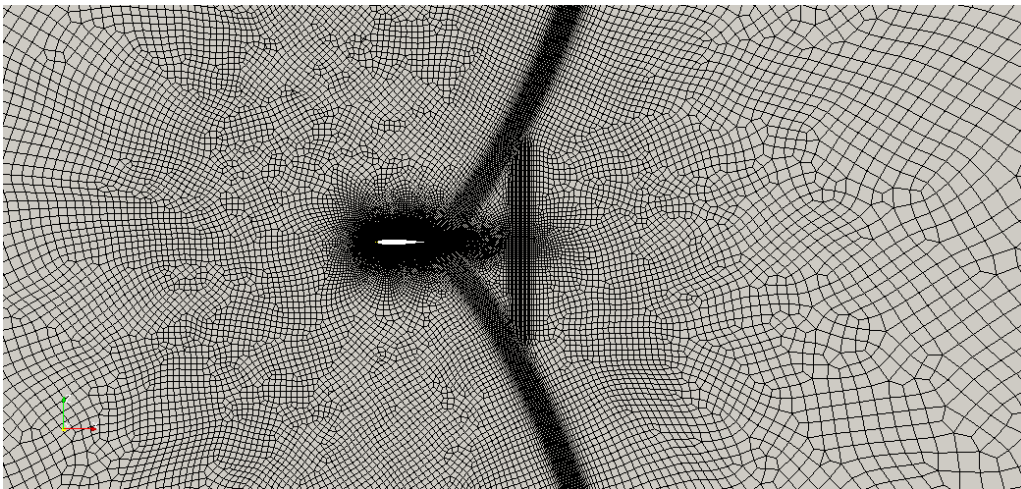


Figure 4.31: Initial feature-aligned mesh prior to nodal movement

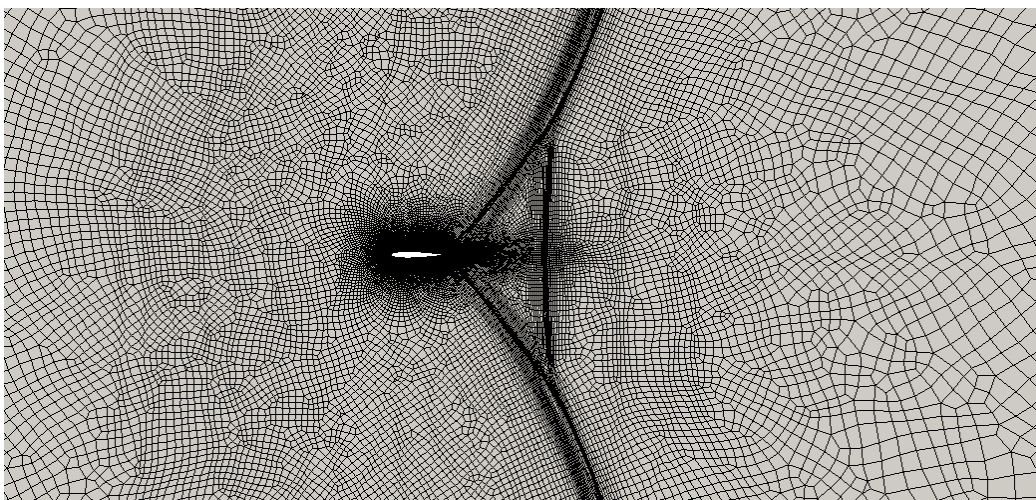


Figure 4.32: Final feature-aligned adaptive mesh

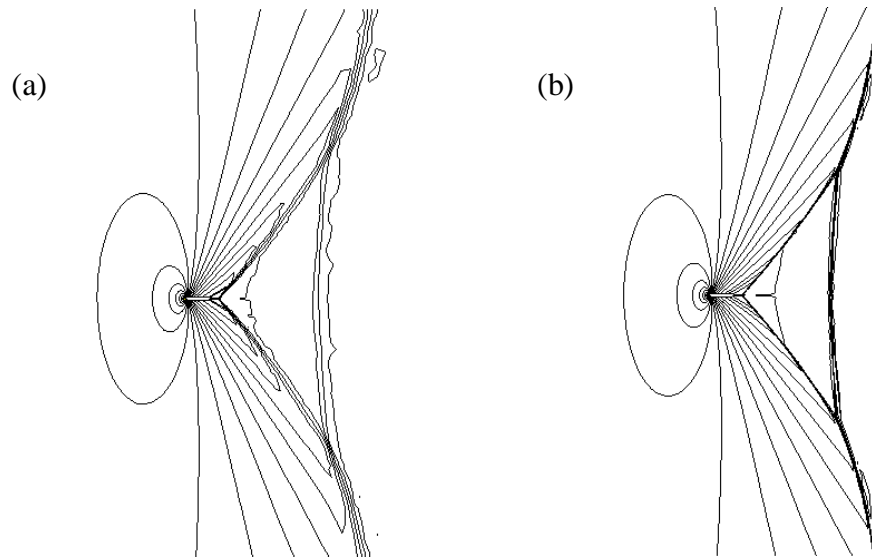


Figure 4.33: Mach number contours for (a) original and (b) feature-aligned adaptive meshes

The initial feature-aligned mesh is displayed in figure 4.31, with the final adaptive mesh in figure 4.32. The Hessian of the Mach number is once again used to adapt the mesh. A comparison of the Mach number contours for both original and feature-aligned adaptive mesh is displayed in figure 4.33. Once again, due to the full alignment of the cell interfaces with the shock wave, the resolution of the shock is much improved compared with the original mesh. A quantitative comparison of the mesh performances is given in Table 4.1. The values obtained from the feature-aligned adaptive mesh are in closer agreement with the results reported in the literature compared with the original unstructured quad-dominant mesh.

	Original Mesh	Feature-aligned adaptive mesh	Ait-Ali-Yahia et al.^[85]
Lift	0.012	0.0002	0.0001
Drag	0.1098	0.1093	0.1092

Table 4.1: Comparison of lift and drag coefficients for NACA0012 fishtail shock case

4.8.2.3 Inviscid Supersonic Flow: $M_\infty = 1.2$, $\alpha = 10^\circ$

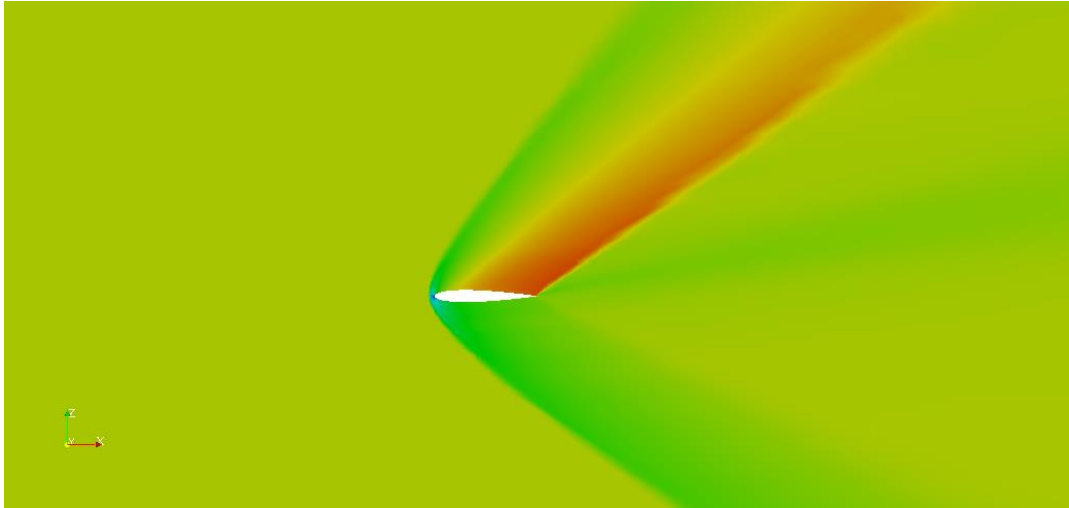


Figure 4.34: Original Mach number solution

Whilst the shock structures in this case are relatively simple, there are multiple shocks to be resolved. A strong bow shock which is detached from the leading edge exists with a weaker shock emanating from the trailing edge due to flow recompression. Numerical diffusion on the initial mesh causes the resolution of the shock waves to be quite poor. The aim for this case is to improve the resolution of the shock waves such that they are resolved to the domain boundaries. Figure 4.34 shows the original solution in terms of Mach number contours.

The shock structure is obtained which yields two main point clouds representing the two shock waves. The connectivity filter is then applied to group the separate regions of points so that the curve fitting process can occur. The original mesh is too coarse to accurately resolve the shocks far into the domain, so the curves are extrapolated towards the domain boundaries (figure 4.35). Associated with the extrapolation is a degree of uncertainty with the exact location of the shock waves. To ensure the shock waves remain captured within the high-quality portion of mesh, the radii of influence of the background sources applied along the curves are progressively increased so that the structured-like mesh is generated over a wider area. The region of mesh ahead of the bow shock is not required for a supersonic simulation and therefore the domain shape can be changed. This

reduces the initial solution domain and indicates an improvement in efficiency through application of the feature-alignment process.

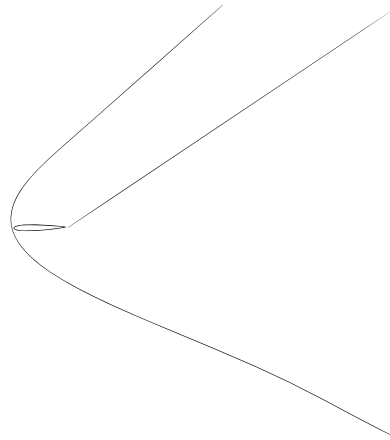


Figure 4.35: Supersonic shock wave geometry

To achieve full alignment of the cell interfaces with the shock wave, the nodal movement is applied again based on the Hessian of the Mach number. Figure 4.36(b) illustrates the final adaptive mesh achieved after interpolating the solution onto the initial feature-aligned mesh. The corresponding Mach number contours in figure 4.37 indicate a substantial improvement in the resolution of the bow and trailing edge shocks towards the domain boundaries. Such resolution is not possible with a standard quad-dominant mesh. Resolving shock waves towards the domain boundaries is important for some applications such as for the study of sonic boom ground signatures generated by aircraft flying at supersonic speeds.

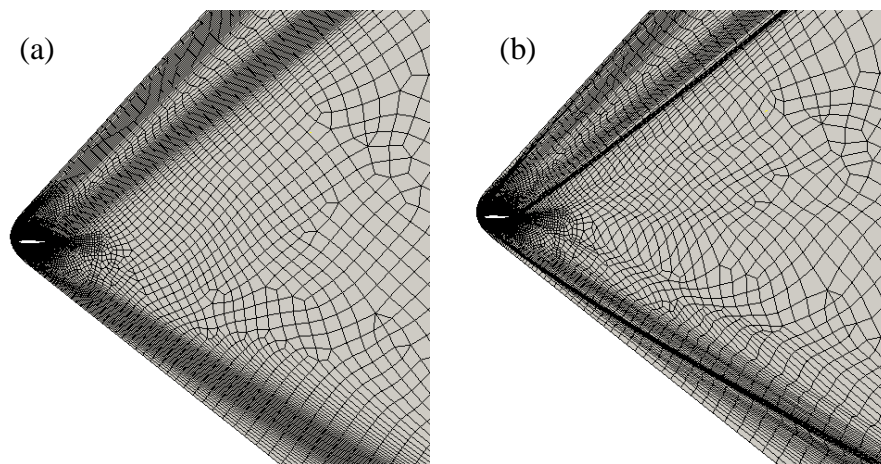


Figure 4.36: (a) Initial feature-aligned mesh and (b) final feature-aligned adaptive mesh

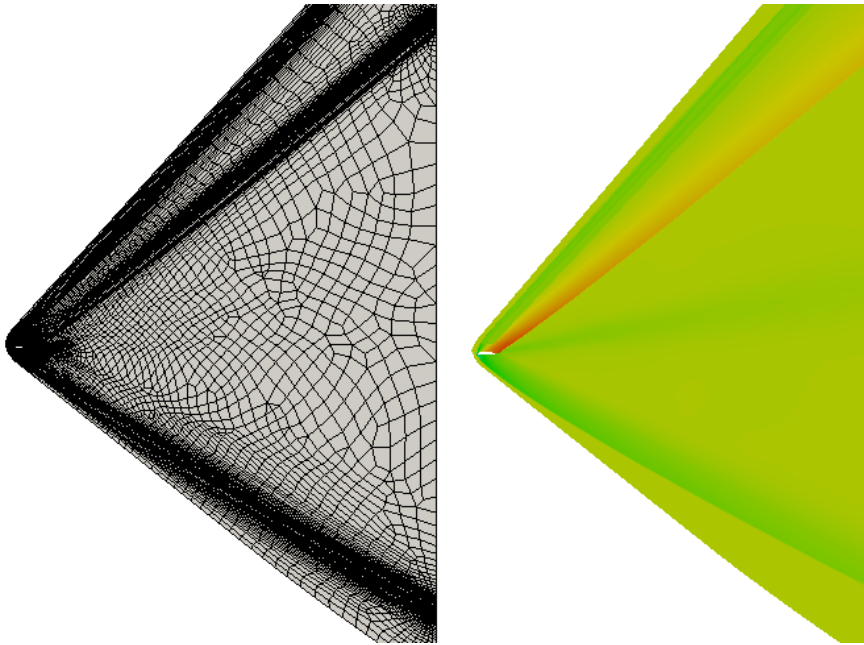


Figure 4.37: View of entire domain to show resolution of shock wave to farfield boundaries

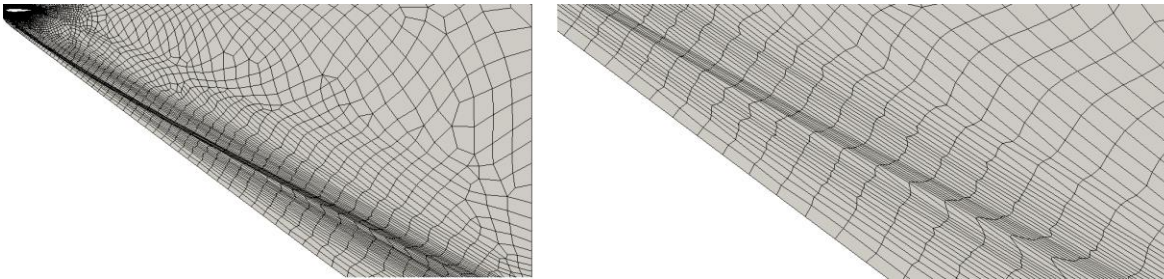


Figure 4.38: Zoom of mesh in lower portion of bow shock

Figure 4.38 shows a zoomed in view of a portion of high quality adapted mesh in the bow shock region, indicating effectiveness of the nodal movement scheme in shifting the points to where the shock wave actually exists in the solution.

4.8.3 RAE2822 Aerofoil Transonic Viscous Flow: $M_\infty = 0.730$, $\alpha = 2.79^\circ$, $Re = 6.5 \times 10^6$

The flow conditions for this problem correspond to case 9 in the AGARD report^[181], where the angle of attack used in the present computation is a corrected value. However, rather than compare the computational results with the reported wind tunnel data, the grid independent solution is obtained in order to observe the quality of the numerical solution of the feature-aligned adaptive mesh. This method provides a more appropriate

means to assess the solution quality. Different turbulence models can predict different shock positions which can often fail to match the experimental data. The matching of the turbulence model with experimental data is not an indicator of the ability of an adaptive mesh strategy to reduce the solution error in a numerical sense, since the assumptions and simplifications in the physical modelling can cause discrepancies with the physical data. Instead, the present approach provides an indication in the reduction of the numerical error through application of the adaptive feature-alignment process. The solution of the feature-aligned adaptive mesh is also compared with an anisotropically adapted triangular mesh in order to demonstrate the potential benefits in solution accuracy and efficiency using quadrilateral meshes rather than triangular meshes.

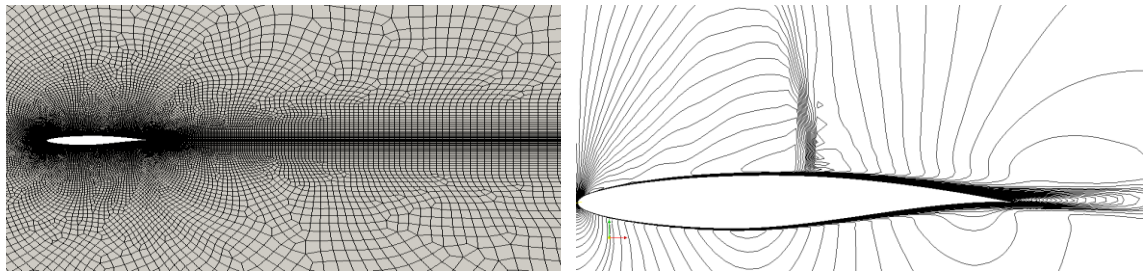


Figure 4.39: Initial RAE2822 unstructured quad-dominant mesh and Mach number contours

For this case the flow features are relatively simple, but since a viscous simulation is being considered, the feature alignment process is also required to represent the trailing wake. A transonic shock exists above the upper surface of the aerofoil which interacts with the boundary layer. The initial mesh and corresponding Mach number solution is shown in figure 4.39.

The shock wave and wake are separately extracted using the extraction filters described in Chapter III. The curves generated for these flow features are shown in figure 4.40. After application of background sources across the curves, the mesh is generated to yield the initial feature-aligned mesh, as displayed in figure 4.41. After interpolation of the original solution onto this mesh, the nodal-movement adaptation strategy can be employed to align the cell interfaces with the flow features.

The Mach number is chosen to compute the Hessian matrix components which define the anisotropic metric, as it is sensitive to both shock waves and shear layers, whereas for a variable such as pressure, the mesh will only adapt to the shock waves. Figure 4.42(a) shows the final feature-aligned adaptive quad-dominant mesh, where it can be observed that the mesh has adapted to the shock wave, thickening of the boundary layer towards the trailing edge, as well as the trailing shear layers which flow downstream.

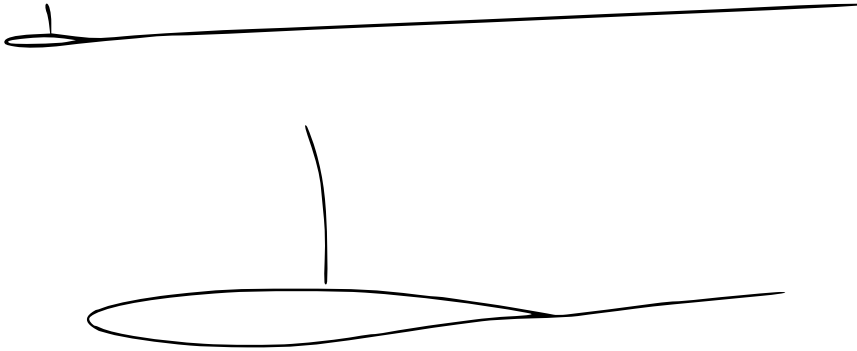


Figure 4.40: Flow feature geometry for extracted shock wave and wake

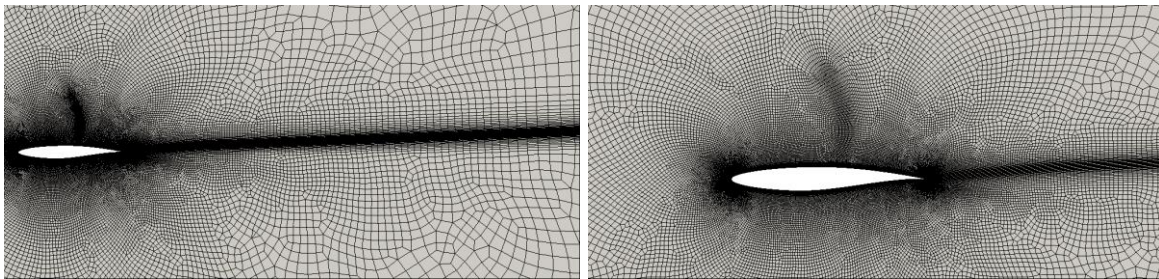


Figure 4.41: Initial feature-aligned quad-dominant mesh

In order to demonstrate that the current method can provide improved results compared with an unstructured anisotropic adaptation on a triangular mesh, the flow is calculated again using the latter method. This mesh is generated with a similar number of points as the feature-aligned quad-dominant mesh, but contains approximately double the number of cells. The feature-alignment step is not carried out on this mesh, but the mesh contains the same density of points as the original feature-aligned mesh around the geometry in both the normal and streamwise directions. The anisotropic adaptation on this mesh leads to much improved capture of the shock, boundary layer and wake, compared with the original quad-dominant mesh. The adapted triangular mesh has adapted in a

similar fashion to the quad-dominant mesh (figure 4.42(b)), which would be expected since the anisotropic error estimate indicates the same flow feature regions requiring adaptation.

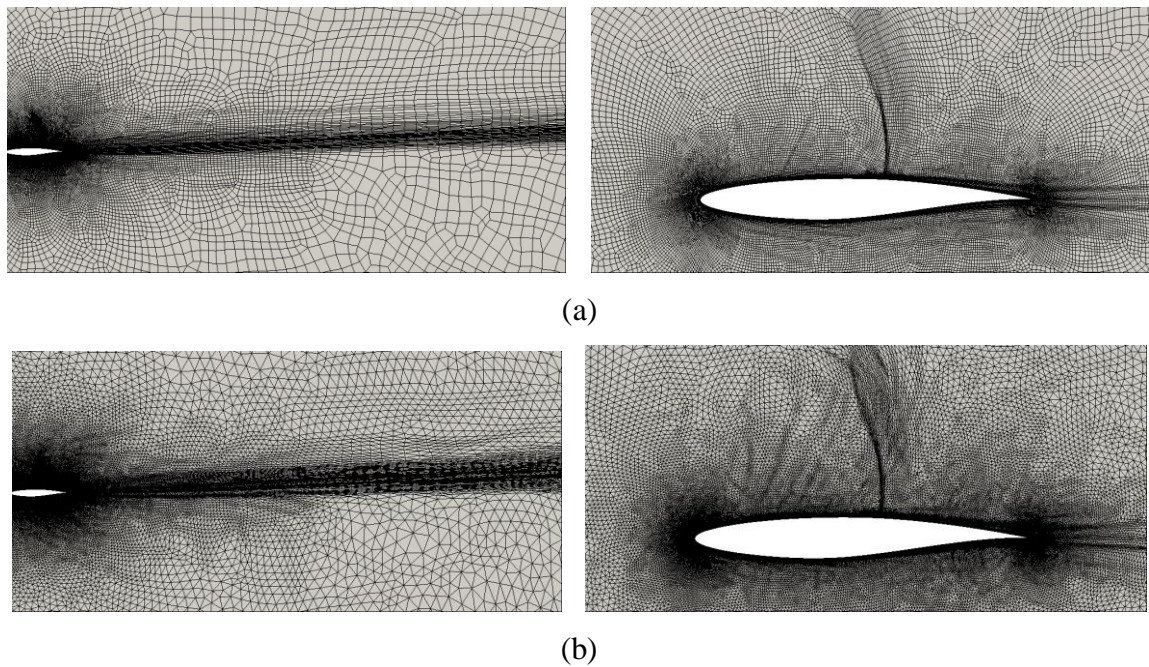


Figure 4.42: (a) Final feature-aligned adaptive quad-dominant mesh and (b) adaptive triangular mesh

The results in Table 4.2 show that the adapted triangular mesh gives slightly improved lift and drag compared with the original quad-dominant mesh, but does not quite yield the same level of accuracy as the feature-aligned quad-dominant mesh and the grid independent solution. This difference could be due to the highly skewed cells affecting the accuracy of the solution, and also that the solution usually failed to converge properly. The adaptive triangular mesh may actually perform better in combination with other unstructured mesh adaptation techniques such as refinement, coarsening and edge swapping. Due to time constraints, these methods were not implemented for the current study. However, even the inclusion of these techniques in the adaptive strategy would still lead to the highly skewed elements present in the flow feature regions. One of the edges of the triangular elements will also not be fully aligned with the shock, degrading the potential for the same quality of resolution observed on the adaptive quad-dominant mesh in figure 4.43(c), which is perfectly aligned with the shock wave.

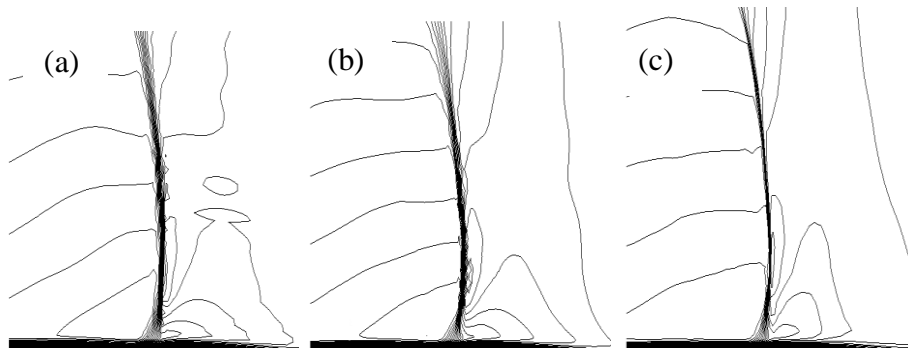


Figure 4.43: Mach number contours in shock region (a) refined mesh 3, (b) adaptive triangular mesh and (c) feature-aligned adaptive quad-dominant mesh

In Table 4.2, the drag value for the feature-aligned adaptive quad-dominant mesh is within a tenth of a single drag count to the grid independent solution. There is a slight under prediction of the lift coefficient, but significant improvement compared to the original and triangular meshes. This is also reflected in the pressure distributions in figure 4.45. The differing lift values appear to be a result of the shock position being predicted slightly upstream compared with the grid independent solution. The solution improvements of the feature-aligned adaptive mesh however, have been obtained at a fraction of the computational cost associated with generating an extremely fine mesh and corresponding solution. This indicates that aligning the cells with the dominant flow features using solution-based adaptation can lead to the solution improvements in a more efficient manner than using isotropic global refinement.

Case	Nodes	Cells	Lift	Drag
Original Mesh	119204	119300	0.76713	0.015677
Refined Mesh 1	305654	305885	0.78686	0.015971
Refined Mesh 2	489274	489671	0.80144	0.016233
Refined Mesh 3	700130	700647	0.80643	0.016387
Refined Mesh 4	924186	924840	0.80643	0.016385
Feature-aligned adaptive quad mesh	200602	200762	0.80093	0.016375
Adaptive triangular mesh	201196	401683	0.78511	0.015718

Table 4.2: Comparison of lift and drag coefficients

Adaptation on both types of grid can provide the high aspect ratio cells which are desirable for the accurate capture of flow features with strong gradients. However, only adaptation on the feature-aligned adaptive quad-dominant mesh can provide the low skewness of the cells in the most important regions of the mesh, which has appeared to provide a more robust convergence compared with the adapted triangular mesh. For comparison, the convergence histories are plotted against CPU time in figure 4.44.

For the adaptive triangular case the residual starts to oscillate around a constant value. The presence of excessively skewed elements in the flow feature regions, particularly in the boundary layer part of the mesh may be the cause for this convergence problem. It can be observed that the feature-aligned adaptive quad-dominant mesh converges much more robustly than both the adaptive triangular mesh and globally refined quad-dominant mesh. This implies that for an efficient process, both quadrilateral cells and adaptation are necessary to align the cells with the feature in question. The first spike in the residuals for the feature-aligned adaptive quad-dominant mesh occurs due to the solution interpolation onto the newly generated feature-aligned mesh. The adaptation for the triangular mesh occurs at a similar residual level, but since the triangular mesh contains a much larger number of cells it therefore takes longer to reach this point. The globally refined mesh presents slow convergence in terms of CPU time due to greater number of computations required.

A comparison of the shock resolutions for the globally refined mesh, triangular mesh and feature-aligned adaptive quad-dominant mesh is shown in figure 4.43. It can be observed that both quad-dominant and triangular adaptive meshes have given improved resolution of the shock wave. Both adaptive meshes also resolve the trailing shear layers towards the farfield boundaries, whereas the globally refined mesh fails to provide the same level of resolution. This is due to the numerically diffusive nature of the isotropic cells present in the wake.

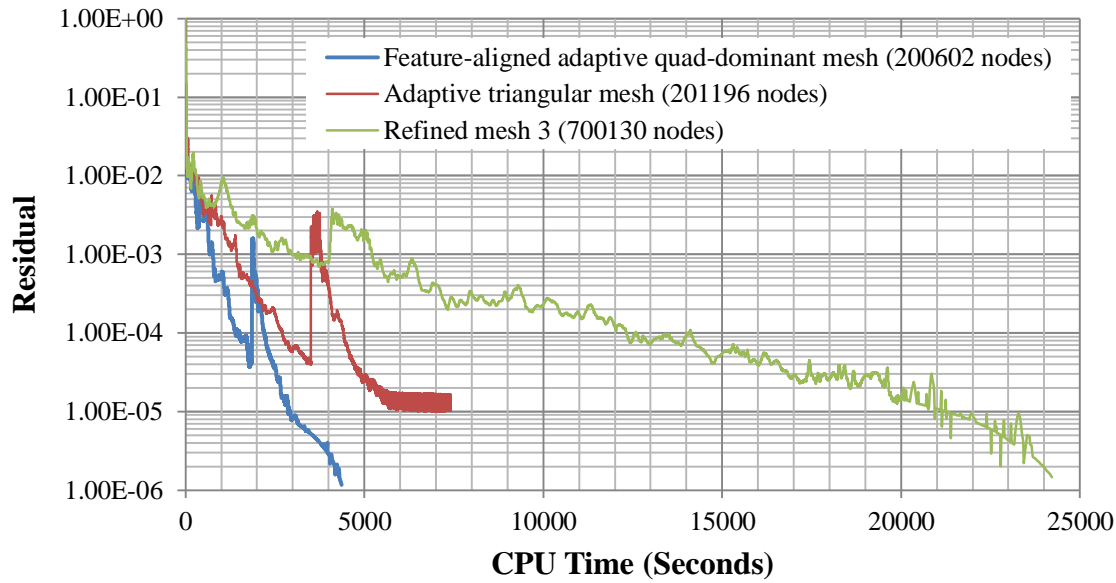


Figure 4.44: Convergence histories for Feature-aligned mesh, adaptive triangular mesh and isotropically refined mesh

Figures 4.45 and 4.46 show the surface pressure and skin friction distributions respectively for the original mesh, feature-aligned adaptive quad-dominant mesh, triangular mesh and isotropically refined meshes. The grid independent solution is indicated by the overlapping of the distributions in both plots. It can be observed that the feature-aligned adaptive quad-dominant mesh gives results for both surface pressure and skin friction close to a global refinement which generated approximately 3.5 times the number of nodes, indicating a significant computational saving. The skin friction plot obtained from the adaptive triangular mesh demonstrates some strange behaviour with a noisy distribution across the aerofoil surface. Again, this is most likely due to the highly skewed elements present in the boundary layer, and could be ultimately contributing to the loss of accuracy compared with the adaptive feature-aligned quad-dominant mesh which contains almost the same number of nodes.

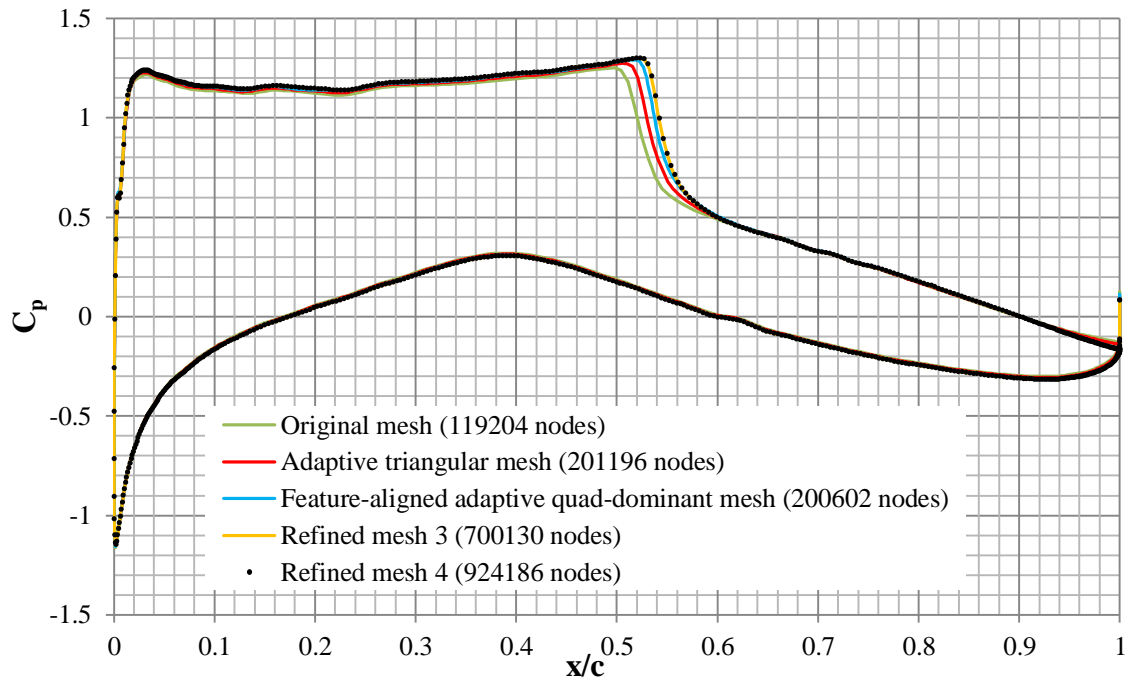


Figure 4.45: Surface pressure distributions

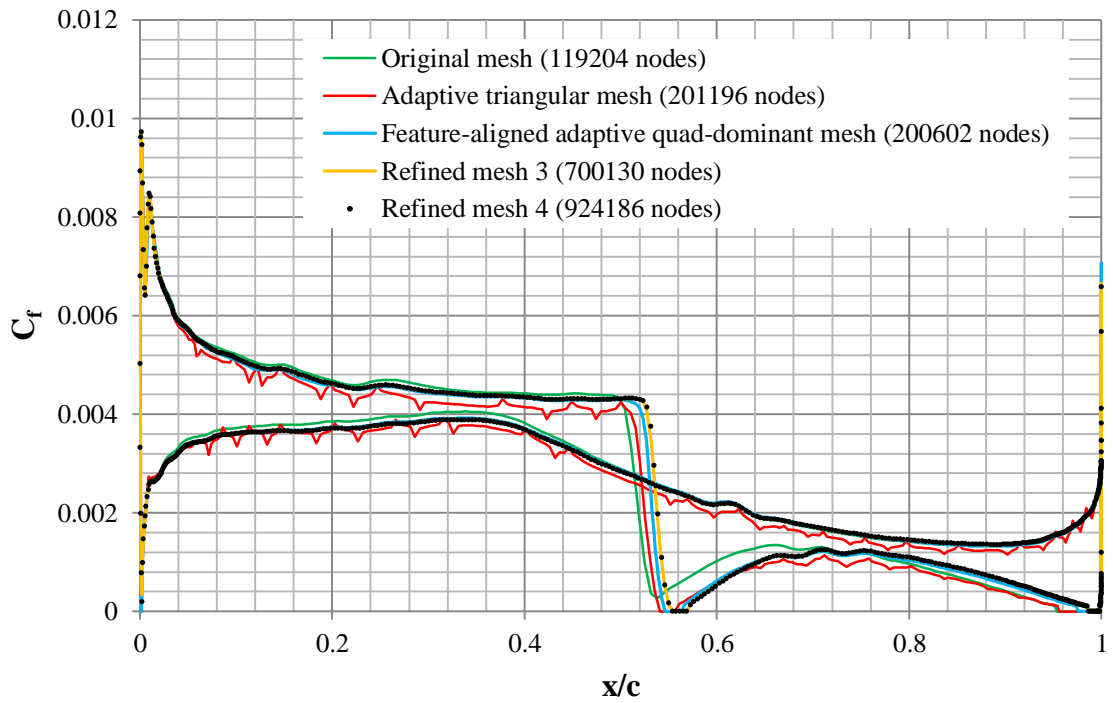


Figure 4.46: Skin friction distributions

4.8.4 L1T2 Multi-Element Aerofoil Subsonic Viscous Flow: $M_\infty = 0.197$, $\alpha = 4.01^\circ$, $Re = 3.52 \times 10^6$ [197]

This two-dimensional high lift configuration is a three-element aerofoil which consists of a main element, a slat forward to the main element with a deflection angle of 25° , and a Fowler flap aft of the main element which has a deflection angle set to 20° . This test case is dominated by viscous flow phenomena and the main flow physics of interest are strong interactions between the boundary layers and wakes of the three individual elements. The interaction between a wake and a boundary layer downstream can lead to a thickening of the boundary layer. It is therefore important that these features are adequately resolved. The original mesh is displayed in figure 4.47, with the initial Mach number contours in figure 4.51(a). For this particular case, there are two ways to capture the wakes using the feature-alignment approach since there will be wakes trailing from each component. The first method involves generating a region of aligned mesh which covers all three wakes, and the second method generates an aligned mesh for each individual wake.

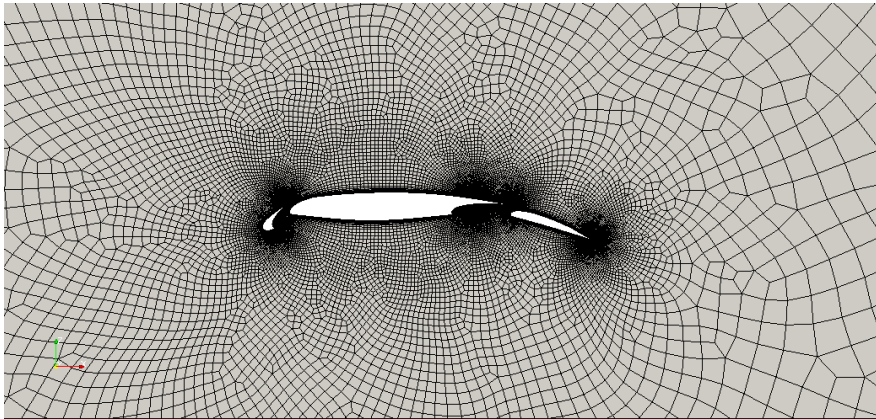


Figure 4.47: Initial unstructured quad-dominant mesh

For the first method, after the initial feature-aligned mesh has been generated, it can then be adapted to each individual wake using the nodal movement algorithm. The benefits of the first approach include that the process does not need to extract each wake individually, which can be a difficult task. Also the approach takes into account the fact that at some point downstream the wakes will eventually merge to form one wake. The second approach involves representing each wake individually. Unfortunately, in order to successfully generate the mesh using the second approach, some user interaction is required. The reason for this is that after extraction of the wakes it can be difficult to

identify individual wakes due to their proximity in the mesh – the connectivity algorithm will usually only detect one region of points, since the extracted points for each individual wake are connected at some location. This means that the curve representation process fails to produce three separate curves. A more suitable technique to generate the individual wake curves is to trace streamlines emanating from the trailing edge of each component. This tracing provided the curves necessary to generate the individual wake meshes. Therefore with this second approach, the medial axis method is not suitable. For comparison purposes, both approaches have been included to observe the effects on the numerical solution. Figures 4.48 and 4.49 show the wake geometry for the two approaches.



Figure 4.48: Embedded wake geometry for the single wake approach



Figure 4.49: Embedded wake geometries for multiple wake approach

Figure 4.50(a) shows the initial feature-aligned mesh and adapted feature-aligned mesh for the single wake approach, with the multiple wake approach shown in figure 4.50(b). The adaptation is performed using the Hessian of the Mach number once again due to its sensitivity to shear layers. It can be observed that the multiple wake approach allows for anisotropic refinement downstream of each trailing edge, whereas the single wake approach covers a more general wake region which can then be adapted. The wake region for the single approach thickens as the mesh adapts to the shear layers detected in the solution.

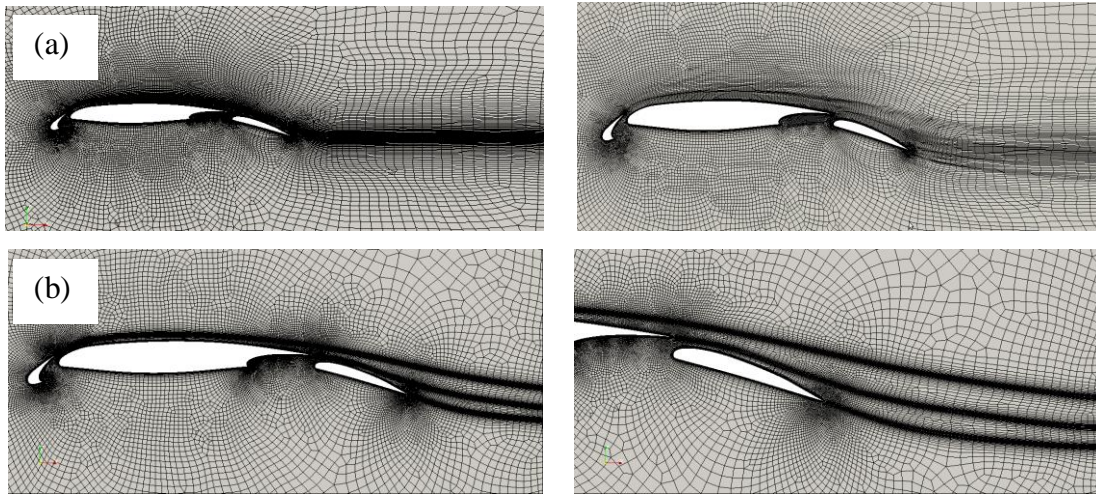


Figure 4.50 (a) Initial feature-aligned mesh for single wake approach and final adaptive mesh (b) feature-aligned mesh using multiple wake approach

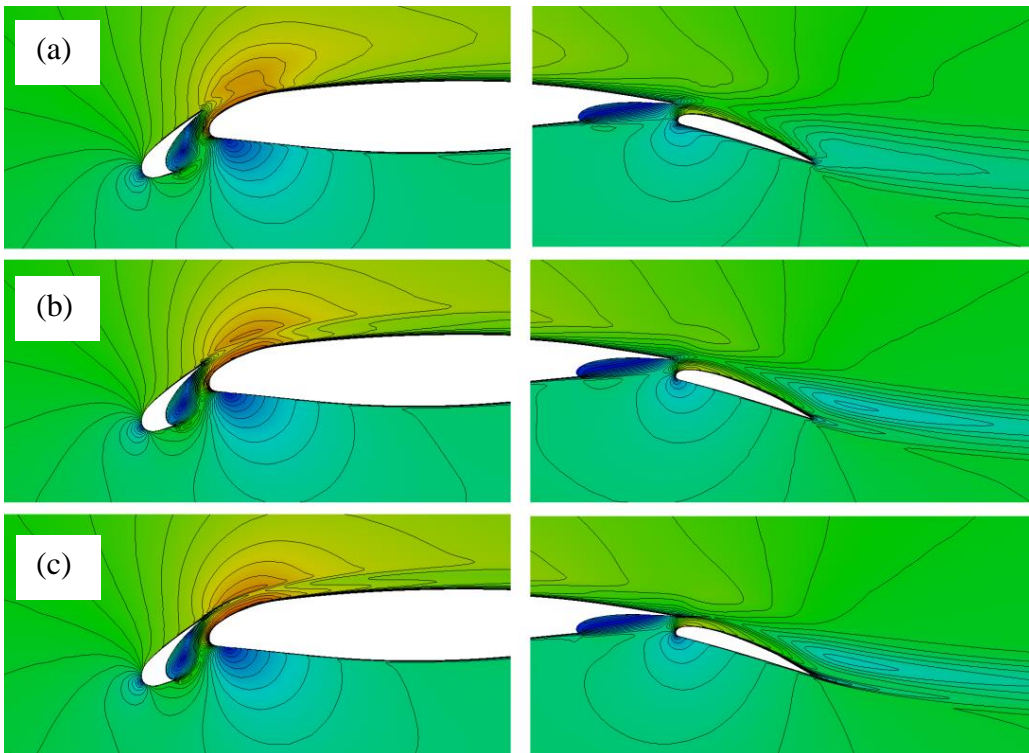


Figure 4.51: Mach number contours (a) original mesh, (b) single wake approach with adaptation and (c) multiple wake approach

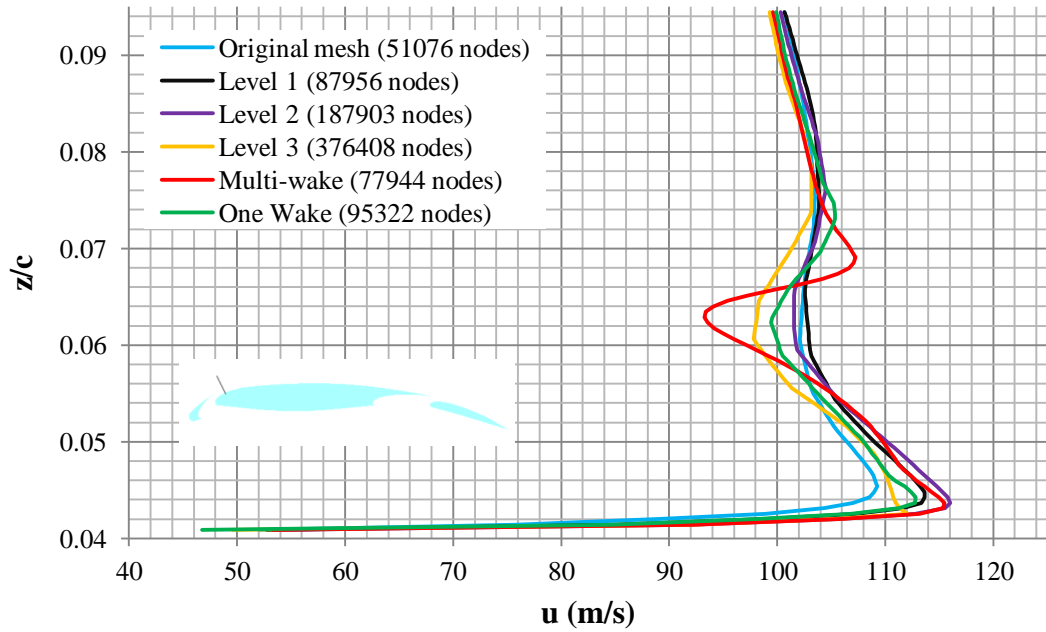


Figure 4.52: Comparison of velocity deficits at the first station for both feature-alignment approaches and a range of progressively refined meshes (cut line coordinates – point 1: 0.082, 0.029, point 2: 0.06, 0.072)

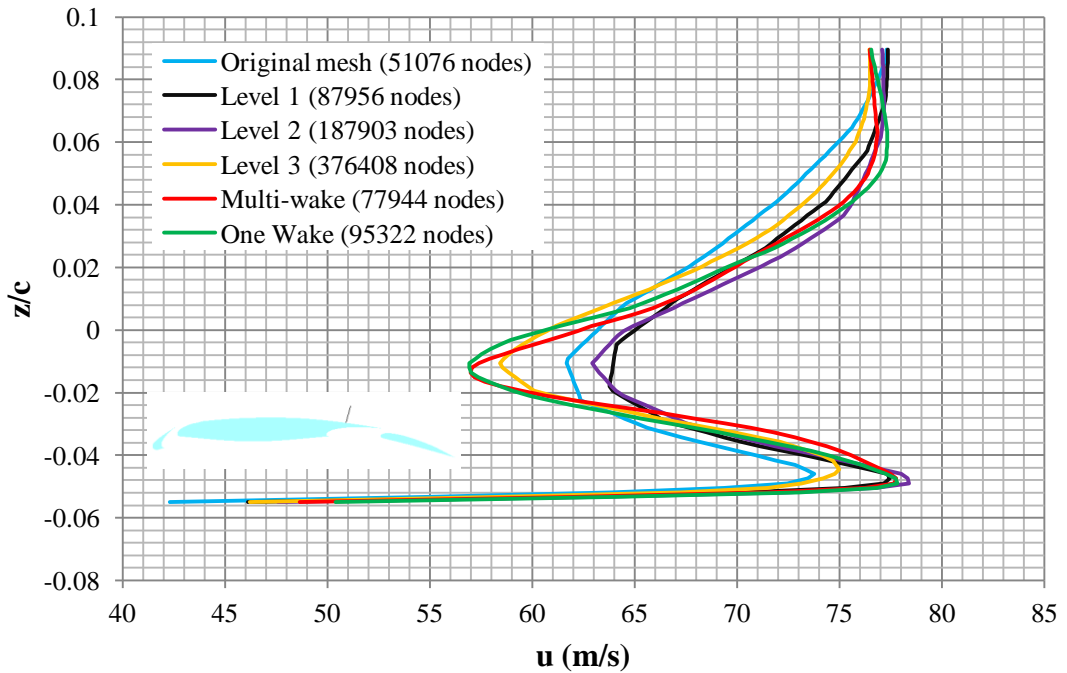


Figure 4.53: Comparison of velocity deficits at the second station for both feature-alignment approaches and a range of progressively refined meshes (cut line coordinates – point 1: 1.07, -0.04, point 2: 1.11, 0.068)

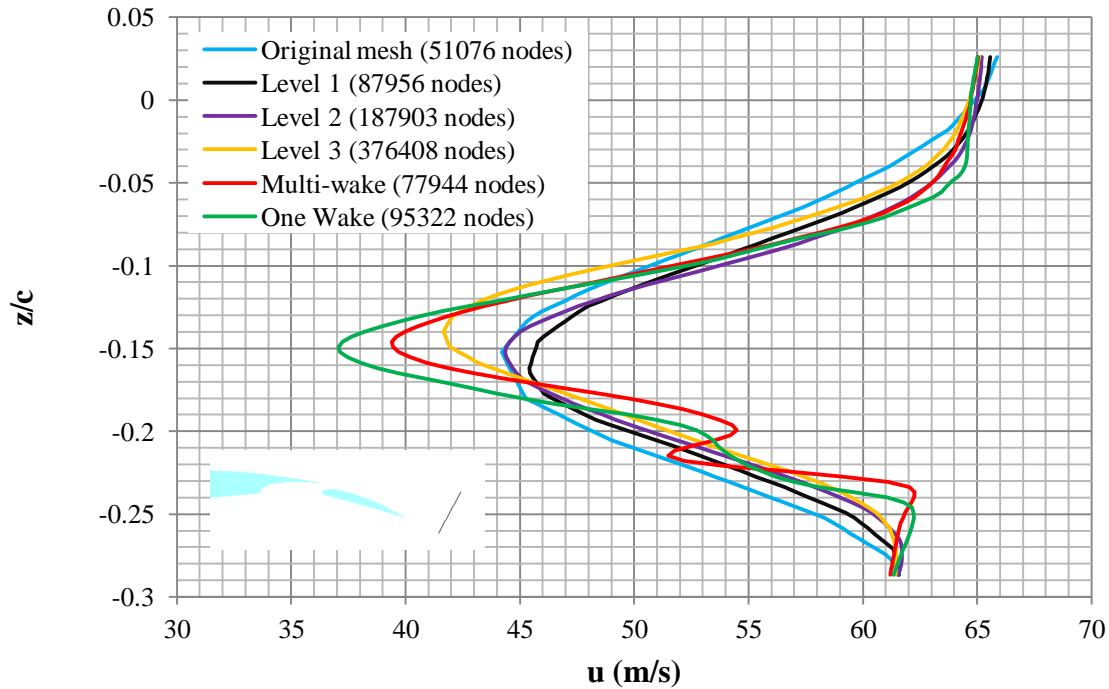


Figure 4.54: Comparison of velocity deficits the third station for both feature-alignment approaches and a range of progressively refined meshes (cut line coordinates – point 1: 1.45, -0.219 point 2: 1.48, 0.02)

Figure 4.51(a), (b) and (c) compare the resolution of the Mach number contours for the original mesh, single wake approach and multiple wake approach respectively. The contours indicate that both feature-alignment methods improve the resolution of the trailing shear layers downstream of each element. Both the single wake and multiple wake approaches have improved the resolution of the shear layers downstream of each component in comparison with the original mesh.

For a quantitative comparison of the mesh performance, figures 4.52, 4.53 and 4.54 show the x -component of velocity deficits at three locations for both feature-aligned meshes. A range of progressively isotropically refined meshes are also considered. The cuts are made across lines (the coordinates are given in the figure titles) defined within the program ParaView. Across the three locations, both single and multiple wake approaches predict slightly higher velocity deficits compared to the most refined isotropic mesh.

It is difficult to determine from the present results whether the single or multiple wake approach provides the most physically realistic results. A more extensive grid independence study is required, as only a small range of meshes were considered. It does appear that both single and multiple wake approaches predict large velocity deficits across the three stations compared with the isotropic meshes. This indicates that the anisotropic elements present are more effective at resolving this effect. However, which result is physically ‘correct’ is unknown at this stage. Whilst at station 2 (figure 4.53) both approaches appear to be in close agreement, at station 3 (figure 4.54) the multiple wake approach is resolving a second peak in the velocity deficit which is not evident with any of the other meshes. Comparison with experimental data would be necessary for a proper validation of such behaviour.

4.8.5 Wind Tunnel with Forward Facing Step, Inviscid Supersonic Flow, $M = 3.0$

This test case has become a bench mark for validation of computational and numerical schemes and was investigated in detail by Woodward and Colella^[182]. The problem involves a uniform Mach 3 flow through a wind tunnel containing a step. The original quad-dominant mesh is shown in figure 4.55.

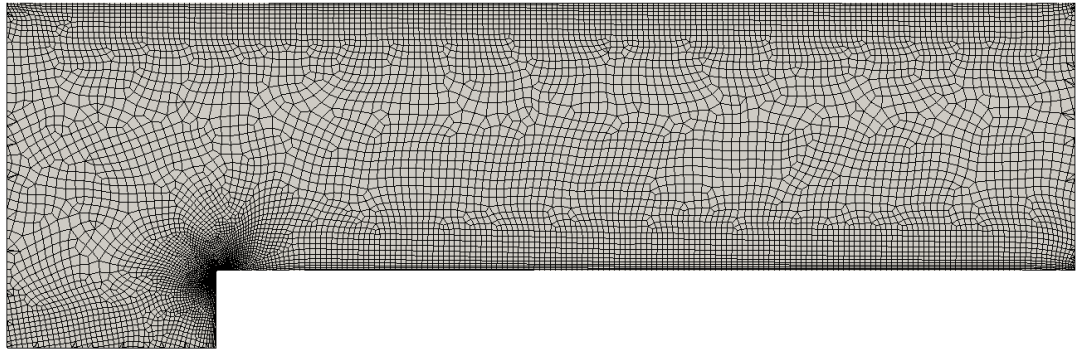


Figure 4.55: Initial unstructured quad-dominant mesh

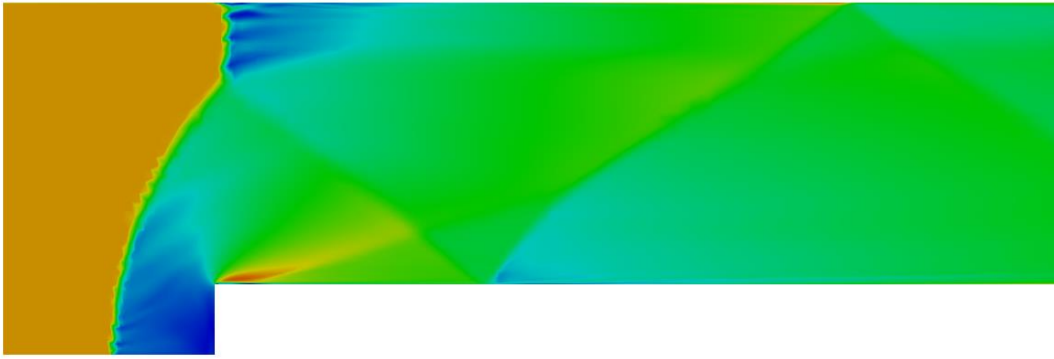


Figure 4.56: Original Mach number contours

This particular flowfield is interesting and presents a challenge for the medial axis process due to the way the shock wave develops around the step and then interacts and reflects off the walls of the tunnel. The flow structure changes throughout the time evolution of the flow. The solution at time $t = 4$ is shown on the original mesh in figure 4.57. Whilst the flow at this time is still unsteady, it has been chosen due to the complexity in the flow structure, which provides a challenge for the robustness of the medial axis approach due to the fact that the shock fronts cross each other. The steady state flow is established at $t = 12$ but the resulting feature is not as interesting or complex.

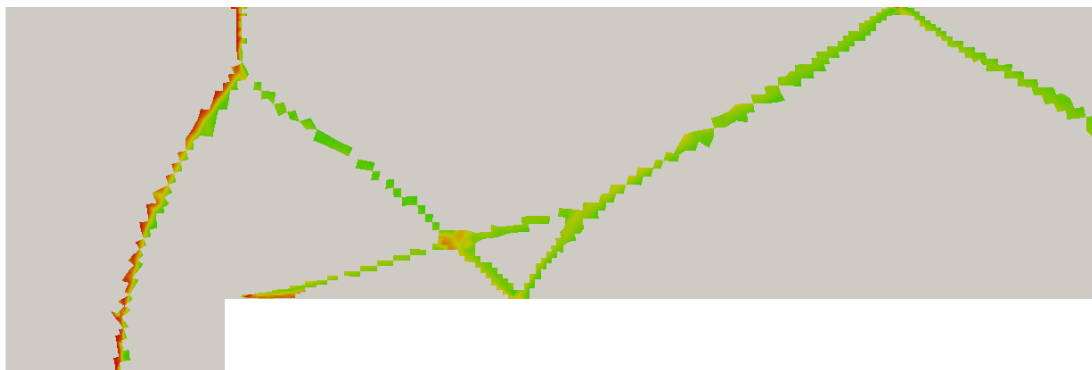


Figure 4.57: Extracted shock structure

The resulting point cloud after the shock structure extraction is shown in figure 4.57. This case actually highlights one of the issues with using the normal Mach number in that it can occasionally pick up structures which are not true shocks. Figure 4.57 shows that the extraction has picked up the expansion region around the corner. However, the

inclusion of this structure actually provides a more complex point cloud to test the robustness of the medial axis approach. The concave hull and unprocessed medial axis are displayed in figure 4.58 and 4.59 respectively. For the medial axis processing this case presents slightly more of a challenge compared with the previously presented external flows.

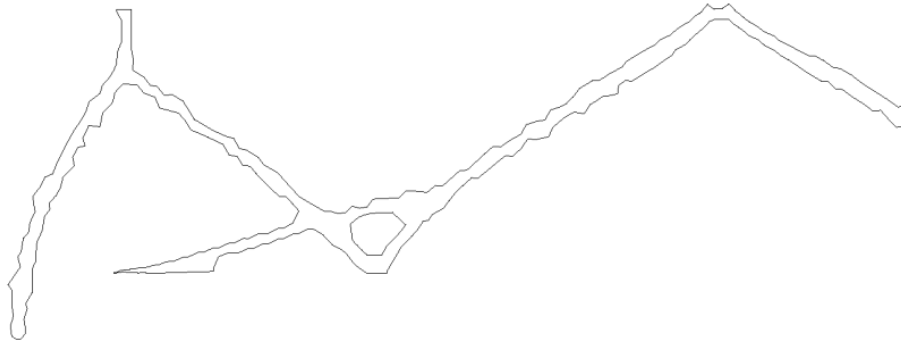


Figure 4.58: Concave hull of extracted shock structure

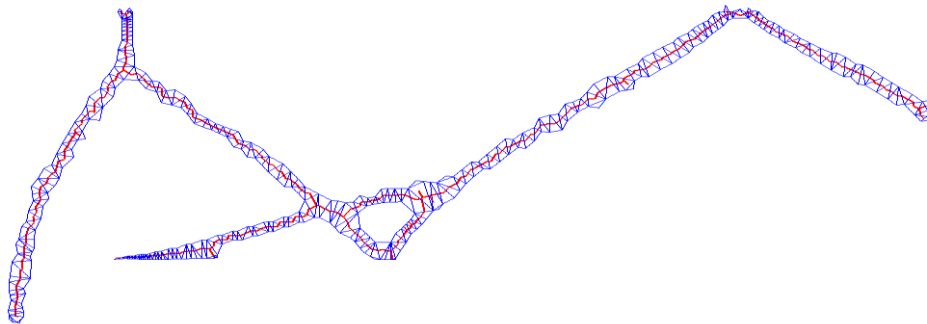


Figure 4.59: Constrained Delaunay triangulation with medial axis approximation

It can be seen from the original solution and the corresponding extraction that the shocks cross over and intersect at a point. A zoomed-in image of this region is shown in figure 4.60. This crossover initially appears represents a point where the medial axis branches out in four separate directions from a junction point. However, this is impossible since the medial axis approximation is based upon a triangulation where branches may only cross edges of the triangles.

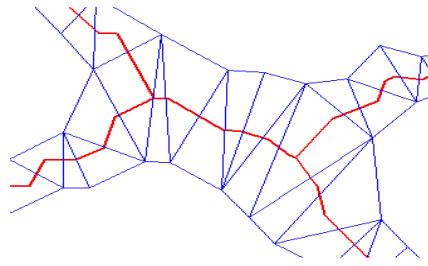


Figure 4.60: Zoom of junction points at cross over region

Figure 4.60 displays the type of scenario which occurs whenever a crossover appears in the extracted flow structure. It must be dealt with in a slightly different manner than a usual junction because, as the figure shows, this crossover point actually possesses two junction points in close proximity. The split of the medial axis allows the small bridging curve between the two points to be isolated and removed automatically. Such a scenario can easily be detected since the considered branch will be bounded by two junction points, and of small size. One other difficulty that appears in this case is that in the locations where the shock wave is reflected off the wall, the medial axis changes direction without a junction node appearing. This prevents a split from occurring using the previously described methods in Chapter IV. In an attempt to remedy this, after the polynomials are fitted, points of inflection are computed and the curve is then split at this location to indicate the change of direction. New polynomials are then fitted through the separated data.

Since the SOLAR surface mesh is likely to fail during meshing if multiple curves actually intersect, the first and last points in each curve list are deleted to prevent this from occurring. This can be observed in the smoothed geometry curves in figure 4.61 where small gaps exist between each individual curve (the size of the gap is exaggerated for the purposes of the diagram). For this particular test case, the curve recombination is switched off as it resulted in a better quality set of final curves. The final flow feature curves are then appended to the geometry file prior to surface mesh regeneration. Figure 4.62 displays the resulting feature-aligned mesh.

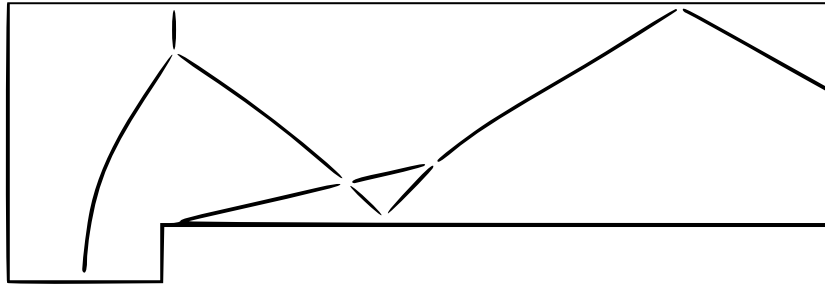


Figure 4.61: Processed medial axis curves

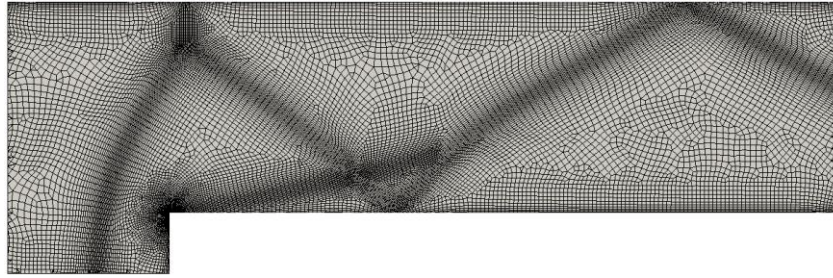
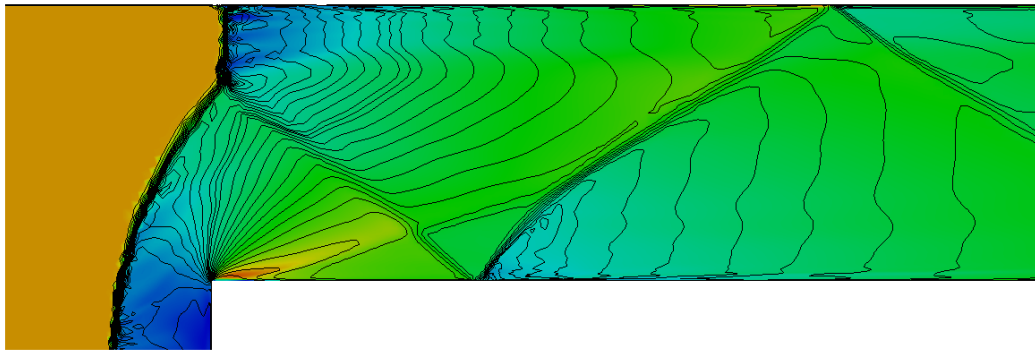
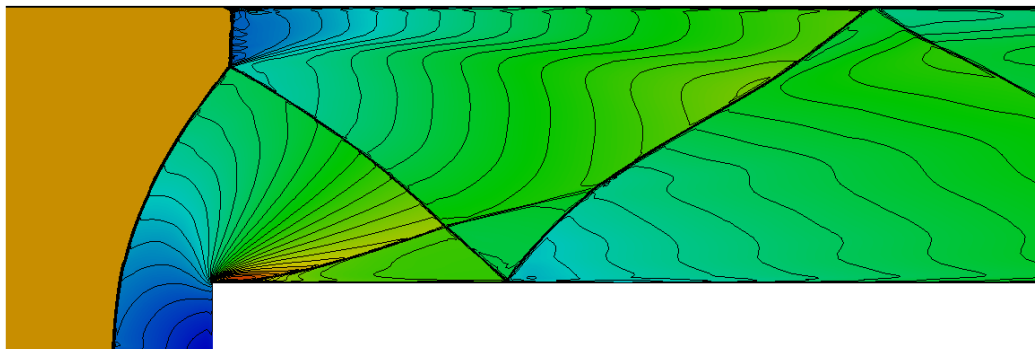


Figure 4.62: Feature-aligned mesh



(a)



(b)

Figure 4.63: (a) Original Mach number contours and (b) Feature-aligned adaptive Mach number contours

The medial axis process has allowed for a feature-aligned mesh to be generated which has improved the resolution of the flow structure considerably compared with the initial mesh, as observed in figure 4.63. It should be noted that generally this process of feature-alignment through mesh regeneration is not particularly suitable for unsteady problems, since the medial axis and mesh would be required to be regenerated at each time step. However, the case does provide a demonstration that the medial axis process can be used for considerably complex flow structures which occur for internal flows. It has also highlighted some limitations of the existing approach which were not evident for the previously presented external flows, leading to a number of general improvements in the medial axis process to allow different levels of complexity to be dealt with.

4.9 Conclusions

A method to represent complex flow features as geometric entities using the medial axis has been described in this chapter. The feature curves are used to guide the location of background sources which influence mesh spacing and orientation of the mesh elements during the generation of unstructured quad-dominant meshes. The resulting feature-aligned meshes possess high-quality local regions of mesh which mimic the attributes of a structured mesh. This feature-aligned mesh can serve as a suitable starting point for a nodal redistribution scheme, which fully aligns the cell interfaces with the flow features based on the solution. The final meshes are aligned with the flow features to give high resolution in the feature normal direction.

The process has been demonstrated for a range of two-dimensional test cases with multiple types of flow features of varying complexity. As an extension of the structured block insertion method^[57], the present work aimed to address the issues regarding the reduction and removal of user interaction for the feature-alignment step and has made significant steps towards achieving this; as well as extending the range of application to more complicated cases. Improved solution accuracy is obtainable by the increased suitability of the mesh for the schemes which capture and resolve the flow features. Mesh efficiency is also improved through the use of high-quality non-skewed anisotropic triangular elements applied in the vicinity of highly directional flow features.

CHAPTER V

Feature-Aligned Hex-Dominant Mesh Block Insertion: Part I: Shock Waves

5.1 Introduction

At transonic flow conditions, shock waves can possess extremely complex structures and in close proximity to the object surface. This means that any adaptation scheme must be applied on both surface and volume meshes. In the context of unstructured meshes, the anisotropic adaptation for a fully tetrahedral mesh will lead to highly skewed surface and volume elements. In hybrid meshes, different regions of elements must be identified in order to apply a suitable adaptation scheme. For either type of unstructured mesh, part or all of the shock wave will be resolved by anisotropic tetrahedra. Techniques which have been established in the literature for embedding blocks of prismatic elements have been applied to supersonic shock waves only. This is primarily due to a supersonic shock wave such as a bow shock around a rounded leading edge, lends itself more readily to geometric representation. Furthermore, the bow shock is also not attached to the geometry. The complexity of transonic shock waves has meant development and application of feature-aligned mesh generation methods has not been considered before - standard adaptation schemes are usually implemented instead, but even they may be problematic for adaptive surface meshing in viscous computations.

The present chapter describes a method to allow for a hex-dominant mesh block to be embedded into the hybrid volume mesh which is aligned with complex transonic shock waves. The generation of shock aligned surface meshes is first considered, which makes use of the medial axis functionality described in the previous chapter. A method to decompose the extracted three-dimensional point cloud is described which allows growth information to be calculated for each part of the shock. An extrusion process which grows selected surface mesh faces in the shock wave region to grow beyond the nearfield mesh, aligned with each portion of the shock wave is then presented. The approach is applied to a couple of test cases to demonstrate its effects on the numerical solution.

5.2 Generating Shock-Aligned Surface Meshes

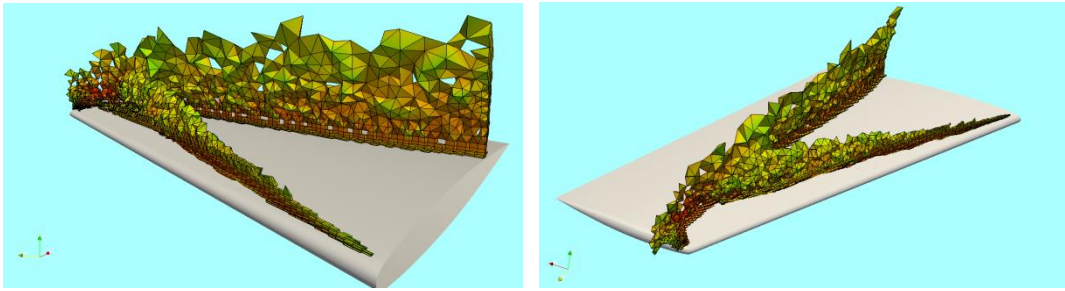


Figure 5.1: Extracted shock wave point cloud for ONERA M6 wing

The main subject of the present chapter is generation of shock-aligned mesh blocks which are embedded into a hybrid volume mesh. This section will focus on a particular shock wave, namely the ‘lambda’ shock which manifests itself during transonic flow around an M6 wing. The actual flow results are discussed in section 5.4.1, but the main shock structure obtained from this test case is presented and discussed here to allow for an introduction to the shock-aligned mesh insertion technique developed as part of this thesis.

Figure 5.1 displays the extracted shock structure. The main lambda shape is evident, with a weaker shock towards the leading edge emanating from the main shock. There are also some complexities in the structure at the wing tip, where the shock wave bifurcates. The figure also displays that the extracted shock point cloud is not at a constant height, and the mesh block growth process will attempt to take this into account.

As discussed previously, SOLAR generates unstructured quad-dominant surface meshes for three-dimensional problems. The process described in Chapter IV which allowed for curves to be embedded into the surface also applies for non-planar surfaces. This means that provided suitable curves can be generated, the unstructured quad-dominant surface mesh can be aligned with the shock wave footprint. For transonic problems, the first task is to therefore represent the shock footprint as a series of curves to allow the surface mesh alignment to occur.

The medial axis process can be applied to generate these curves, as shown in figures 5.2 and 5.3. After extraction of a three-dimensional shock point cloud, it is transformed to a two-dimensional problem by setting the z -coordinate values to zero. This is tantamount to considering the footprint of the shock wave. After creation of the smoothed shock curves,

they are projected onto the upper wing surface to obtain the correct z -coordinates. Anisotropic background sources are then applied along the shock curves, just as in the two-dimensional cases, and the mesh is regenerated. The result is a quad-dominant surface mesh aligned with the shock curves with high-quality structured-like elements.

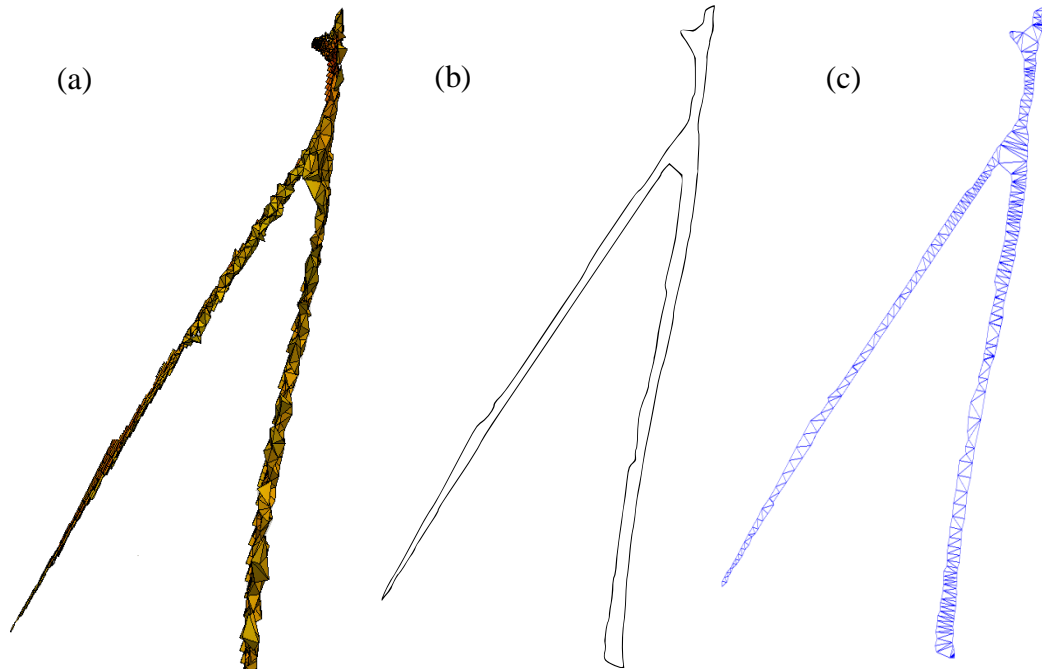


Figure 5.2: (a) Extracted shock footprint, (b) concave hull and (c) constrained Delaunay triangulation

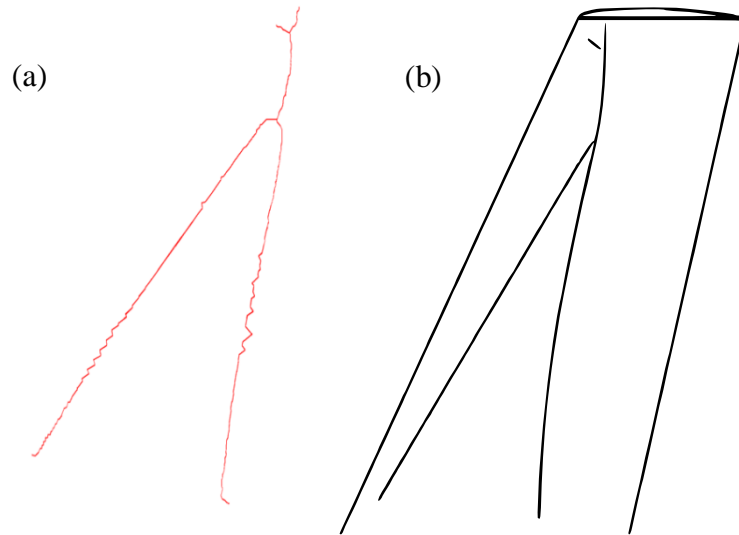


Figure 5.3: (a) Extracted medial axis and (b) final smoothed medial axis curves projected onto wing surface

After generation of a shock-aligned surface mesh (figure 5.4), a standard SOLAR run will maintain this alignment through the nearfield region only. It became apparent that it would be useful if the advancing layer mesh was allowed to continue to grow beyond the nearfield mesh, aligned with the shock wave. This would avoid the generation of isotropic tetrahedra in the shock region which are generally unsuitable for resolution of anisotropic flow features.

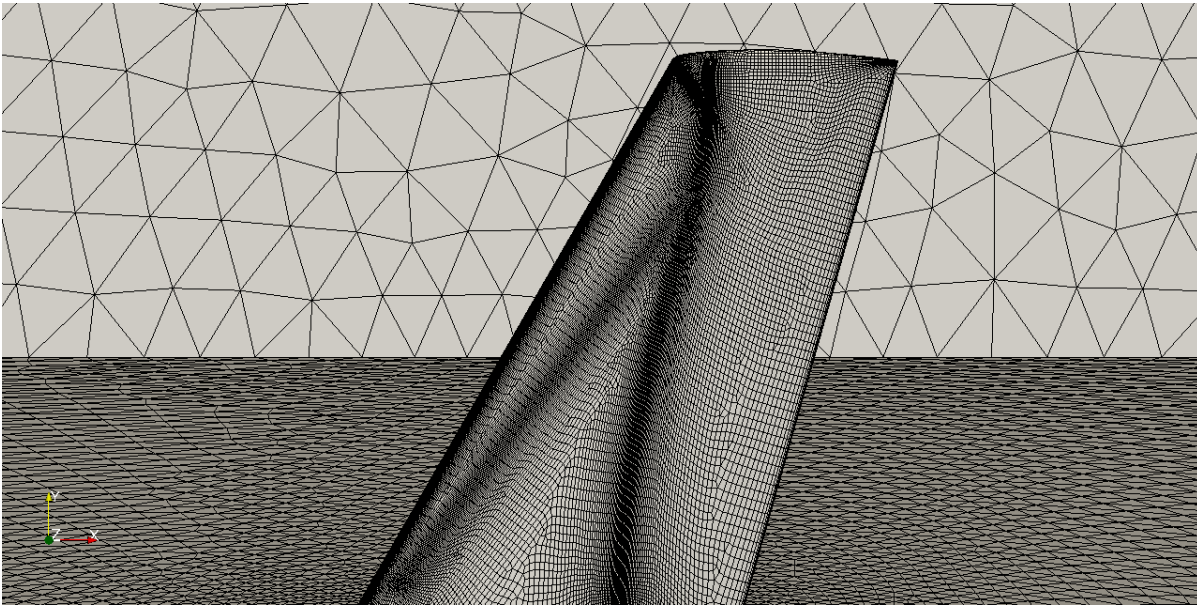


Figure 5.4: Surface mesh aligned with shock footprint

5.3 Volume Mesh Generation

SOLAR generates hybrid volume meshes which are composed of a mixture of elements. The types of element allowed in the mesh are hexahedra, prisms, pyramids and tetrahedra. After generation of an unstructured quad-dominant surface mesh, advancing layers are grown yielding a nearfield mesh comprising of hexahedral and prismatic elements, which are suitable for the resolution of viscous layers. A buffer mesh is then generated around the nearfield, made up of pyramids and tetrahedra in order to provide a conformal interface for the farfield tetrahedra created using a Delaunay point insertion method. The farfield tetrahedra are isotropic in nature, that is, they are only suitable for capturing regions of the flow where the solution behaves in a uniform manner.

The present section describes the methods developed during the project which allow extrusion of unstructured-quad dominant meshes, leading to the generation of hex-

dominant meshes aligned with transonic shock waves. The resulting mesh blocks should provide higher quality resolution of the shock wave and thus allow improvements in the numerical solution to be obtained.

5.3.1 Generation of Shock-Aligned Mesh Blocks

During the project two approaches were implemented for the generation of shock-aligned mesh blocks for transonic problems. The block growth process is essentially the same for each, but the method to identify faces to extrude from is distinctly different. The first approach involved extrusion of a target surface mesh (i.e. the upper wing surface mesh). The implementation of such an approach was much simpler to carry out, but carried two distinct disadvantages. The extrusion of the surface mesh led to an extremely high number of mesh cells being generated, particularly for a refined surface mesh. This method also led to the potential for a large number of poor quality elements in the buffer region of the mesh block, after extrusion of faces near the leading and trailing edges. More importantly, the block can only grow aligned with one part of the shock wave and therefore the alignment of the block with other parts of the shock wave was not particularly good. This method was tested, and presented by the present author^[194] but eventually discarded in favour of the second technique. A slice of the volume mesh after application of the original approach is displayed in figure 5.5.

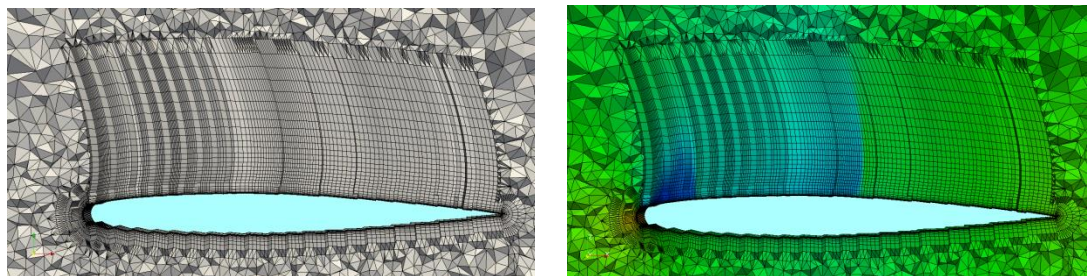


Figure 5.5: Original technique involving extrusion of the entire surface mesh^[194]

The second approach exploits the presence of the medial axis curves representing the shock footprint in the CAD model to create the mesh block. The resulting mesh block matches the shape of the shock wave. The medial axis curves allow the surface mesh faces in the vicinity of the shock footprint to be identified, as well as providing a means to decompose the extracted three-dimensional shock point cloud. This allows portions of the

mesh to grow aligned with each section of the shock and to varying heights. The following section will describe this particular process in more detail.

Stage 1: Decomposition of Shock Wave

A transonic shock wave possesses a complex structure which can prove problematic for the insertion of feature-aligned hex-dominant mesh blocks. Different shock branches may have dissimilar heights and curvatures (figure 5.1) making the alignment for all parts of the shock wave by extruding the entire surface mesh impossible. The medial axis curves can be exploited to allow a decomposition of the shock wave to occur in both a two- and three-dimensional sense.

For each medial curve a simple polygon enclosing the curve is constructed, as displayed in figure 5.7(a). Then, by considering the x and y coordinates of the shock points, a ray casting algorithm can be used to identify which points exist in each polygon. The ray casting algorithm is a simple test which computes the number of times a ray, passing from the exterior of a polygon to the point under consideration, intersects an edge of the polygon. If the number of intersections is odd, then the point exists inside the polygon. For an even number of intersections, the point lies outside the polygon. This process is displayed in figure 5.6, where the ray has intersected the polygon a total of five times, and therefore must exist within the polygon.

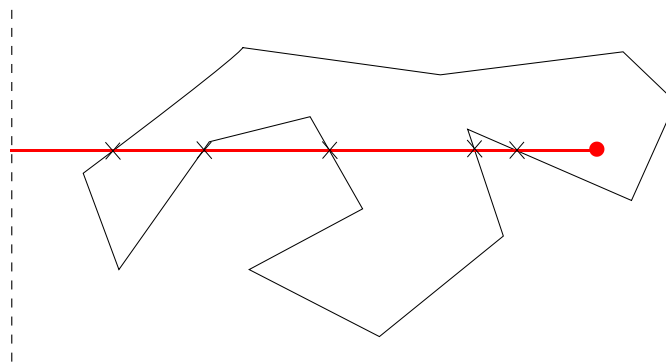


Figure 5.6: Ray casting algorithm – a ray emanating from the exterior of the polygon to the test point intersects the polygon an odd number of times, indicating that the point is on the interior

Whilst the decomposition algorithm is implemented in two-dimensions for the shock wave footprint points, it also allows for decomposition of the shock wave in a three-

dimensional sense (since the point indices remain the same – only the z -coordinate was set to zero to transform the point cloud into two-dimensions). In figure 5.7(b), each shock portion is associated a label ‘A’, ‘B’ and ‘C’. With this data, information such as estimated shock heights for each part of the shock wave can be identified, which will assist in the generation of the mesh block. In the regions where polygons intersect, points may appear in more than one polygon after application of the ray casting algorithm. For simplicity, in the current implementation, these points are simply discarded. Figure 5.7(a) shows an example of the polygons constructed and a result of the decomposition algorithm, where the points are coloured based on the polygon they exist in.

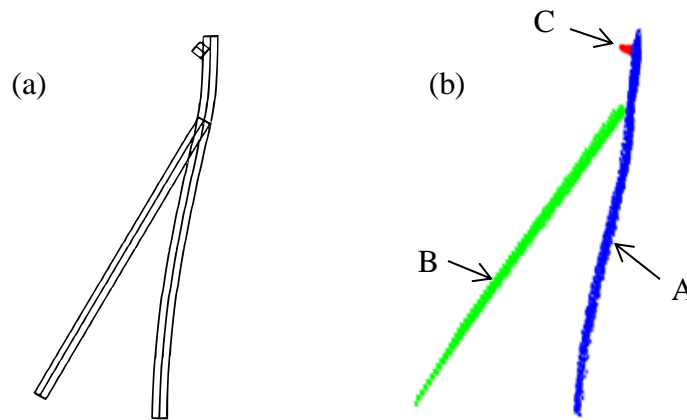


Figure 5.7: (a) Construction of polygons around each medial curve and (b) resulting decomposed point sets

Stage 2: Compute growth information for each portion of the shock

After decomposing the shock, information about each portion can be collected. Of initial interest is the height of the shock portion, which can be found by searching for the point with the highest z -coordinate. If the height of the shock portion is within the estimated height the nearfield mesh will grow, then the associated medial curve is discarded for the extrusion process (the shock will already be approximately aligned within the nearfield due to the surface mesh alignment). Note that for the shock wave displayed in figure 5.2, the shock bifurcation at the wing tip (shock portion ‘C’ in figure 5.7(b)) has been discarded since this portion of the shock curve is below the estimated final nearfield height. For each remaining portion, a target number of layers are computed based on the portion height. To ensure the shock wave is completely enclosed by the mesh block, the computed maximum z -coordinate is slightly increased by 10%.

Referring to figures 5.1 and 5.7, the maximum height of shock portion B is some fraction of shock portion A. Therefore, if the user requests that the maximum number of layers is 20, the set of faces associated with medial curve A will grow 20 layers, whereas the set of faces associated with medial curve B will grow some proportion of this number, corresponding to the calculated ratio of the heights. For example, if the maximum height of shock portion B is computed to be 60% of shock portion A, then shock portion B will grow 12 layers. Since all shock portions grow at the same time, the height of a particular layer will be the same throughout the entire mesh block.

For each remaining curve ('A' and 'B' – recall that 'C' has been discarded) the midpoint is calculated and a surface is constructed at this position, perpendicular to the curve as shown in figure 5.8. The three-dimensional shock points associated with that particular medial curve are projected onto these virtual surfaces if they are in the vicinity of the surface. The aim of this process is to simplify the problem and allow approximate growth vectors to be calculated for each shock portion. Figure 5.8(b) shows a typical result of the projection. For the projected points, a least-square fit is made to obtain a polynomial expression describing the curvature of the shock at the midpoint of the medial curve. This polynomial can then be differentiated at discrete points (the number of points corresponds to the number of layers that are required to be grown) to allow computation of the tangent vectors. This provides the directions that each portion of the mesh block will grow.

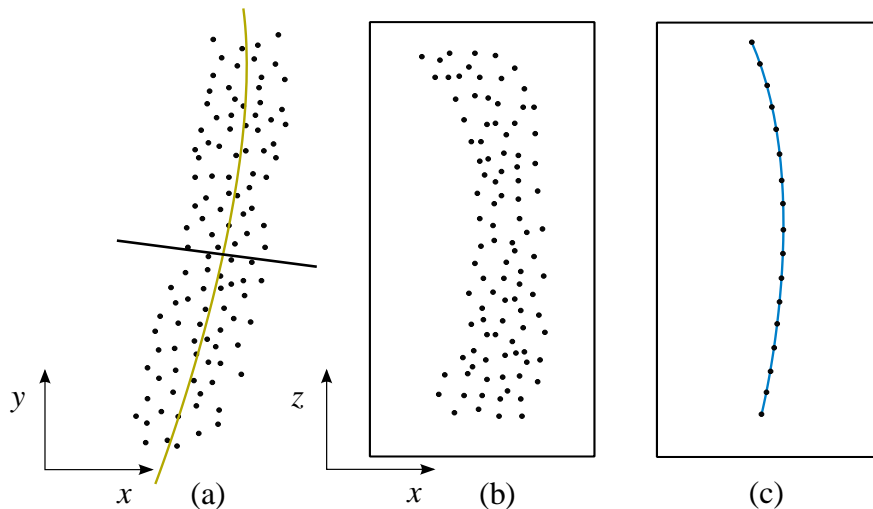


Figure 5.8: (a) Projection of nearby shock points onto surface defined at medial curve midpoint, (b) projected points on surface and (c) polynomial fit through projected shock points

Stage 3: Identification of the faces of the surface mesh that will be extruded

The medial axis curves prove useful not only for generating the alignment of the surface mesh in the first instance, but also providing a means to automatically identify the surface mesh faces which will form the base of the mesh block. Furthermore, due to the almost structured-like nature of the quadrilateral elements in the vicinity of the curves, the resulting mesh block surface follows the shape of the medial axis curves.

The method to identify a suitable set of faces for the extrusion is as follows. The surface mesh generator will force some of the nodes to lie precisely along the shock curves. These nodes can be identified, which then allow the mesh faces attached to these nodes to be found. The resulting rows of faces (one row either side of the curve) are then expanded into the surface mesh, row by row, until a suitable thickness of surface mesh has been obtained (the number of expansion layers is configurable by the user and influences the block thickness). This process is carried out for each medial curve, resulting in a set of faces associated with each curve. After obtaining each individual set, the union and intersection of these sets are found. The intersection set contains faces belonging to multiple medial curves. For faces which appear in the intersection set (figure 5.9(c)), they are removed from the set which belongs to the shortest medial curve. After this operation, the number of faces in the union set must be equal to the number of faces in each of the individual sets.

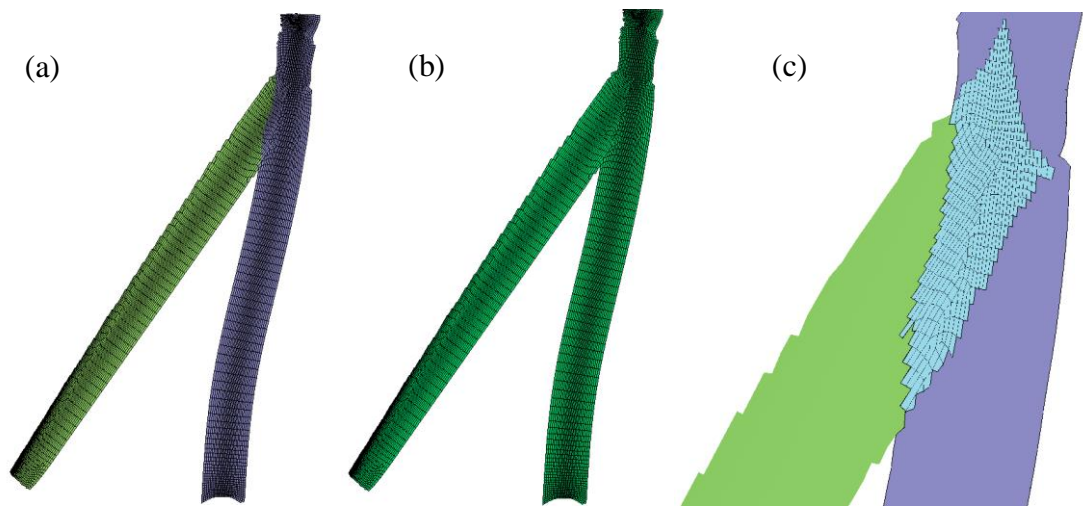


Figure 5.9: Face set identification from the medial axis shock footprint curves (a) individual face sets, (b) union of the faces and (c) intersection set of the faces

Whilst the medial curves allow for the quadrilateral surface mesh to exhibit some structured-like qualities, the mesh is still inherently unstructured. This means that the exterior boundary of the selected mesh faces may be irregular in some locations. This eventually will manifest itself as an irregular mesh block boundary. Another problem is that the local mesh in the vicinity of two medial curves in close proximity is usually of poor quality, as indicated in figure 5.10. Preventing these elements from appearing in the surface mesh is difficult to control, and is again due to the inherently unstructured nature of the mesh. However, the resulting set of faces still approximately follows the shape of the underlying medial curves i.e. allows for a lambda-shaped set of faces to be identified.

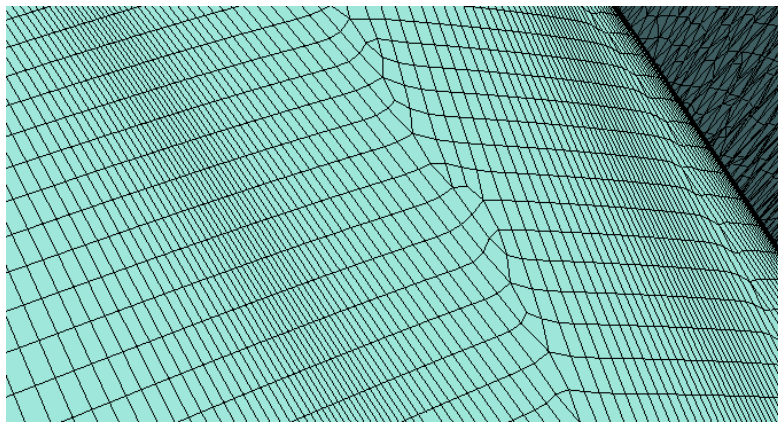


Figure 5.10: Poor quality quadrilaterals generated at the intersection region

Stage 4: Nearfield mesh block growth

The node marching directions and layer heights are computed for each node of the faces identified in Stage 3. It is important that the layer growth is consistent with the standard advancing layer algorithm in SOLAR to ensure nearfield mesh quality later on. The new node locations are calculated and a layer of cells is added to the mesh block.

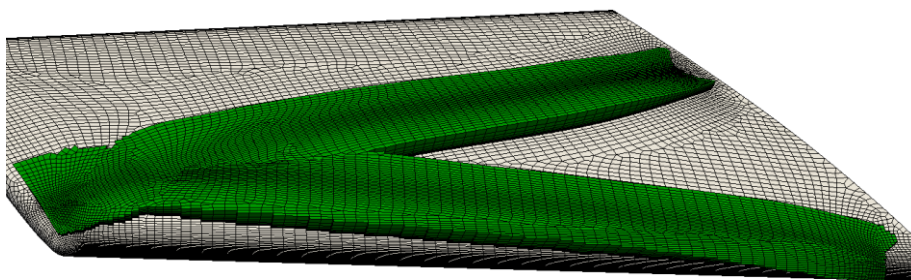


Figure 5.11: Nearfield growth for selected surface mesh faces

Stage 5: Modification of the growth direction

After the required number of nearfield layers has been grown, the growth direction and layer height is changed to the computed values obtained from stage 2 to allow the mesh block to continue growing in order to cover the extent of the shock wave. The cell indices of the elements generated in this stage are recorded for use in the buffer mesh creation in the next step.

During the loop over the targeted faces (i.e. the union set), the process will check which individual set the face belongs to (portion ‘A’ or portion ‘B’), and apply the corresponding growth vectors that are associated with that particular shock portion to the nodes of that face. If the current layer number is below the target number of layers computed for that particular set of faces, the layer will grow, otherwise it will stop. This allows the mesh portions to grow to different heights, as shown in figure 5.12.

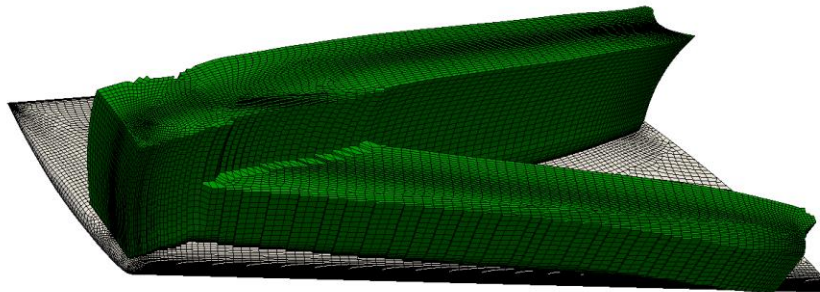


Figure 5.12: Mesh block growth beyond the nearfield to enclose the shock wave

Stage 6: Generation of the buffer mesh

The cells which were found in the previous step are considered in turn. If a cell is found to be on the exterior the mesh block (i.e. one or more of the cell’s faces will not have a neighbouring cell) then it is flagged as a buffer cell. These cells are split by inserting a new point at the centroid of the cell and connecting this point to the existing cell vertices. For a hexahedral element, this leads to the creation of six new pyramids elements per cell considered. For the triangular prism cells (recall that the surface mesh is quad-dominant – there is a chance triangular elements may be present), the resulting split leads to two tetrahedra and four pyramids. Another cell splitting function will replace each boundary pyramid with two tetrahedral cells by splitting along the diagonal of the quadrilateral

boundary face. This provides the conformal interface required for the surrounding farfield tetrahedral mesh.

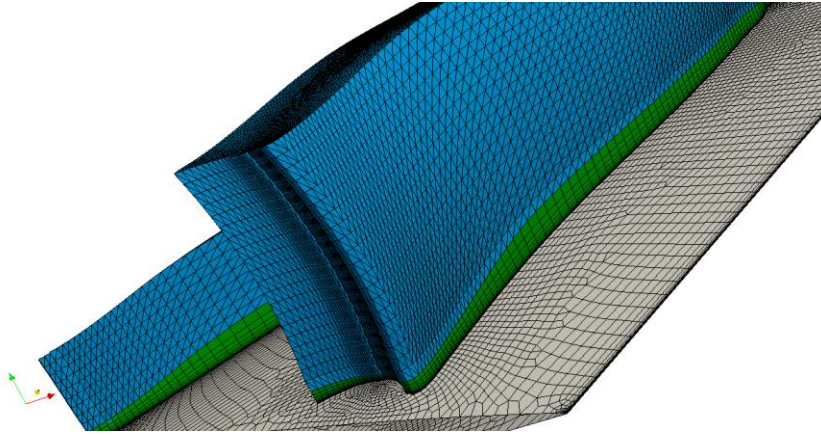


Figure 5.13: Generation of the buffer mesh for outer layer of cells above the nearfield mesh

Mesh conformity is guaranteed since the boundary cells will always be hexahedra or prisms and their boundary faces will always be quadrilateral (i.e. the prisms will never have their triangular faces on the boundary) after construction of the mesh block. Once the splitting operation is complete, the external shell of the mesh block comprises of two distinct sections: the nearfield (quadrilateral faces - green) and the buffer mesh (triangular faces - blue) – see figure 5.13.

Stage 7: Growth of the nearfield from the remaining surface mesh faces

The remaining surface mesh of the model is constrained to grow along the quadrilateral faces of the extruded mesh block. The layers continue to grow until a triangular face (i.e. the start of the buffer mesh) is encountered. Essentially, as the mesh is grown the nodes ‘snap’ to the pre-existing quadrilateral layers, shown in figure 5.14. This is the reason for ensuring the nearfield section of the mesh block was grown consistently with the standard advancing layer algorithm in order to preserve mesh layer quality. The process continues to grow layers until an exterior triangular face is encountered, i.e. the start of the buffer mesh, essentially concealing all the boundary quadrilateral faces. After constructing a buffer mesh for the new nearfield, the outer shell of the whole mesh block is now entirely made up of triangular faces (figure 5.15).

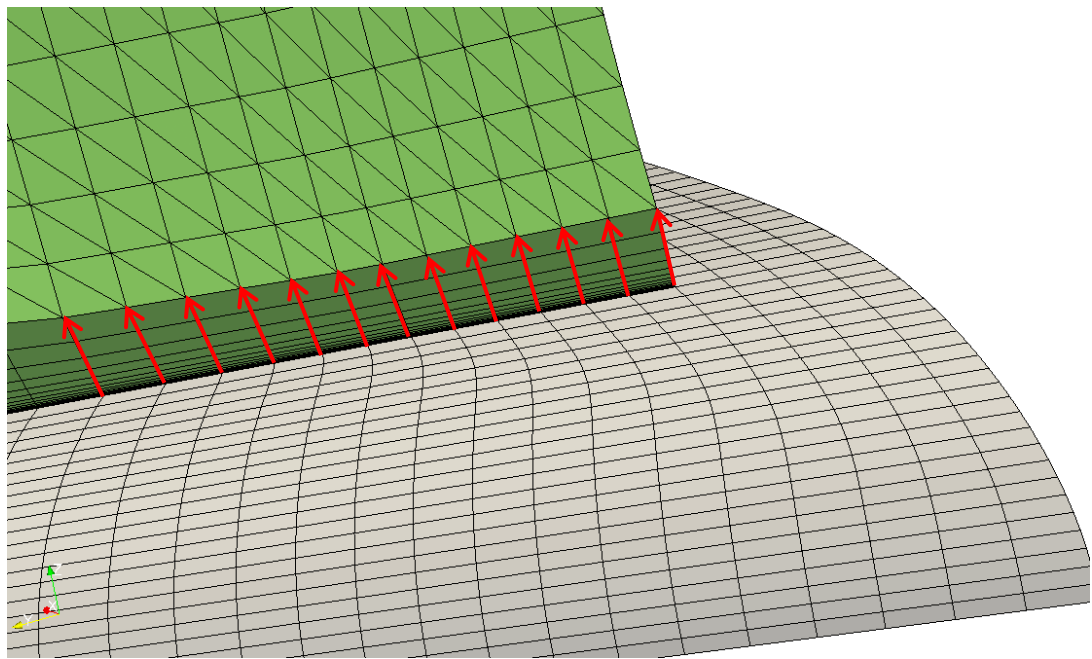


Figure 5.14: Constrained nearfield growth from remaining faces

Stage 8: Generation of the farfield mesh.

The global triangulated surface mesh is updated by adding the external triangular faces of the mesh block shell. This new global surface mesh provides the necessary input for the farfield meshing process which generates Delaunay tetrahedra with a point insertion method. After the tetrahedra have been constructed, a boundary recovery process must occur. Delaunay point insertion methods start with a tessellation of the surfaces, and it is required that the internal tetrahedra respect the original tessellation. In three-dimensions, there is no guarantee that the surface triangulation will be satisfied. A boundary recovery process must occur to perform a number of edge and face operations to recover the surface triangulation, such that it appears in the final volume mesh. If the boundary recovery is successful, the mesh block is then merged into the final global volume mesh. The result is a fully conformal hybrid volume mesh suitable for the TAU flow solver. A typical result of the process is displayed in figure 5.16.

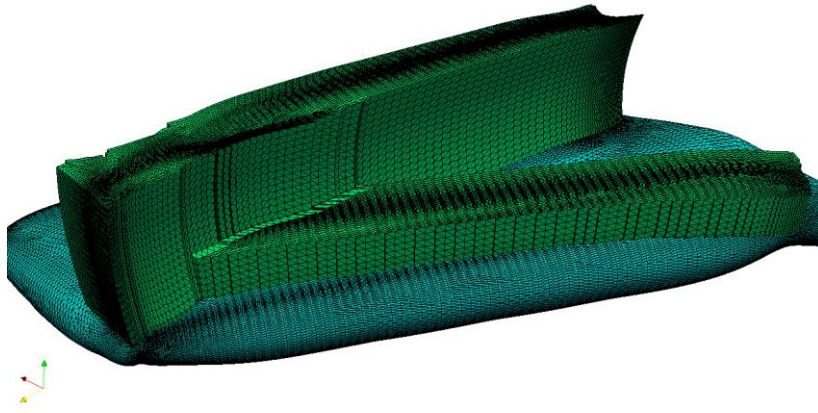


Figure 5.15: Triangulated outer shell of lambda-shaped mesh block merged into nearfield shell around wing

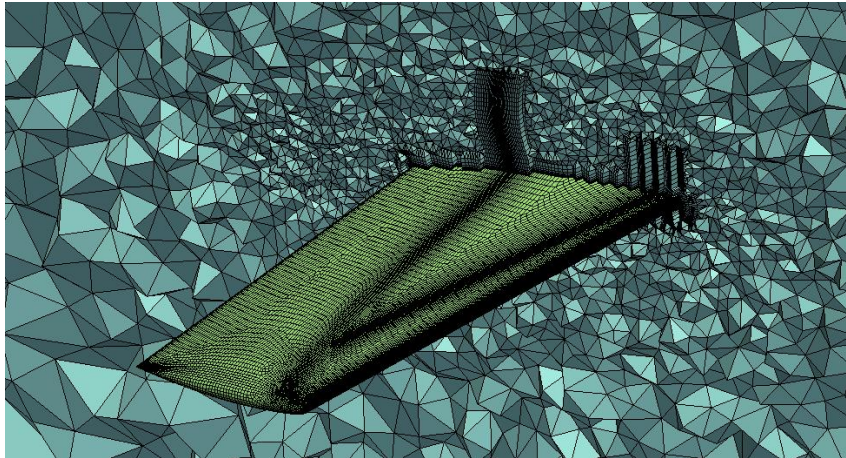


Figure 5.16: Slice of final hybrid volume mesh including the shock aligned mesh block

5.3.1.1 Summary of Process

- Generate surface mesh aligned with shock footprint
- Construct polygons for each medial curve and apply ray casting algorithm to decompose shock wave point cloud
- Compute growth information for each shock wave portion
- Determine sets of surface mesh faces attached to each medial curve
- Loop over the target faces and grow the nearfield portion of the mesh block
- Change the growth direction: identify the set the current face belongs to and grow face nodes in the direction associated with that face set. Record cell indices.
- Create buffer mesh for cells flagged in the previous step

- Grow nearfield from remaining surface mesh faces, the mesh is constrained to grow along the quadrilateral faces of the extruded mesh block
- Update the global surface mesh and generate the farfield tetrahedral mesh

5.4 Test Cases

5.4.1 ONERA-M6 Wing: 3D Viscous Transonic Flow, $M_\infty = 0.8395$, $\alpha = 3.06^\circ$, $Re = 11.72 \times 10^6$ [183]

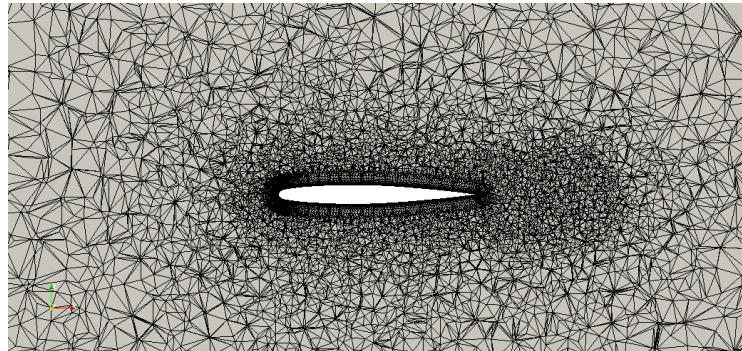


Figure 5.17: Slice of the original hybrid mesh at wing mid-span

A slice of the original hybrid mesh is displayed in figure 5.17, with pressure coefficient contours at four different spanwise stations across the wing in figure 5.18. The application of the medial axis process and hex-dominant mesh block insertion for this test case have been presented in the previous section (figures 5.6 to 5.16). The medial axis approach is employed to initially align the surface mesh with the shock footprint. The extrusion functionality is then applied to generate a high-quality hex-dominant mesh block through extrusion of selected surface mesh faces attached to the medial curves.

The shock wave footprint in this case is quite complex, particularly at the wing tip. After treatment of the shock footprint as a two-dimensional problem, the medial axis curves are generated and then projected onto the upper wing surface. The intermediate steps of the medial axis process are displayed in figures 5.2 and 5.3: the extracted point cloud, concave hull polygon, constrained Delaunay triangulation, unprocessed medial axis and final projected curves. It can be observed that the approach has captured all the complexities of this particular shock footprint. The resulting shock-aligned surface mesh is shown in figure 5.4. The surface mesh can be seen to be made up of high-quality structured-like quadrilaterals which are aligned with the feature curves that had been previously generated.

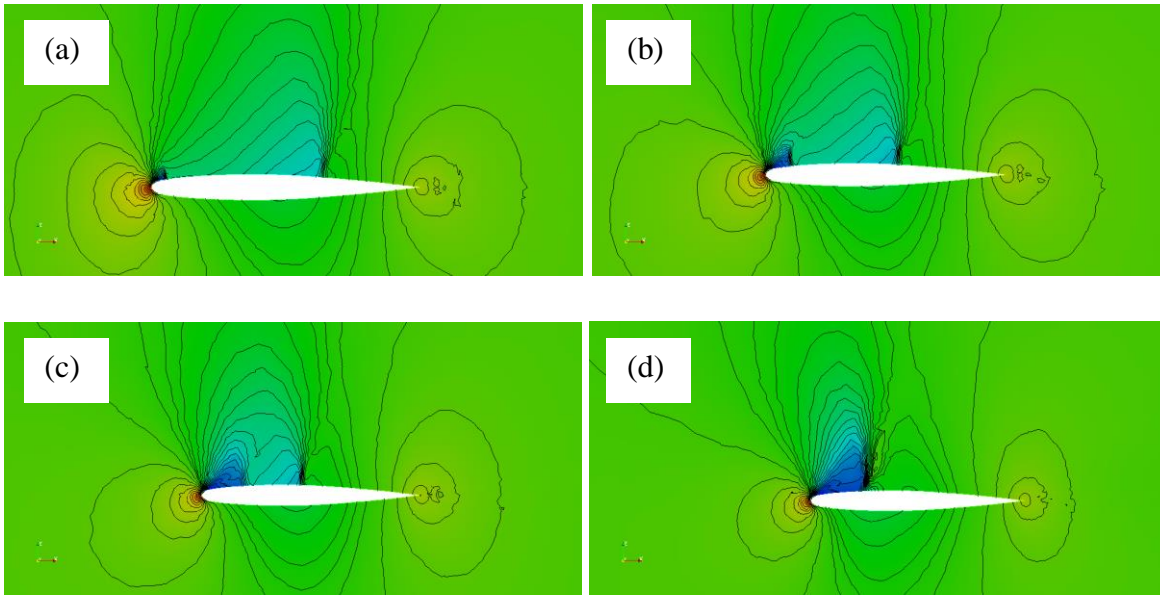


Figure 5.18: Pressure coefficient contours for original mesh at four spanwise stations across wing (a) $y/b = 0.2$, (b) $y/b = 0.5$, (c) $y/b = 0.8$ and (d) $y/b = 1.1$

After the shock decomposition process, only two of the three shock portions are deemed to be above the estimated boundary layer height. The bifurcation of the shock at the wing tip is therefore not considered when constructing the face sets. The surface mesh alignment will be maintained throughout the standard nearfield mesh to allow for alignment with the part of the shock wave closest to the wing surface.

The process provides good alignment of the cells with the shock wave in the final mesh. However the current implementation will still only provide approximate alignment with the shock wave, in a similar fashion to the two-dimensional initial feature-aligned meshes presented in Chapter IV. A three-dimensional implementation of the nodal movement algorithm could be used within the mesh block in order to provide full alignment of the shock wave with the cell interfaces. However, during the present study it was found to be very difficult to successfully adapt within the mesh block without distorting the buffer mesh of pyramids. Furthermore, adapting the nearfield layers particularly those closest to the wing surface also caused problems. This was because some nodes were displaced by distances greater than their proximity to neighbouring nodes, causing inverted elements to appear in the mesh.

To assess the quality of the solution of the feature-aligned meshes, a study of drag coefficients is performed for a range of standard (i.e. no feature-alignment) hybrid meshes

with different levels of global mesh refinement. Experimental drag coefficient data is not available for this particular test case. Due to computational resources (the version of SOLAR was only available on a local workstation) the maximum number of elements was set to six million. The data from these progressively refined meshes are used for the purpose of assessing the performance of the feature-aligned mesh. For further comparison, the original mesh is adapted using an anisotropic metric and refined in the nearfield mesh region using the TAU adaptation module. The mesh is refined until the number of elements reached a similar number as that for the feature-aligned mesh. As the present feature-alignment method only refines the mesh in the shock region, in order to attempt to make a fair comparison, the Hessian of the pressure variable is used to define the anisotropic metric. This is because the adaptation will only be sensitive to the presence of the shock wave and will not adapt to the shear layers in the trailing wake.

Figure 5.19 and Table 5.1 displays data collected from the drag coefficient study. One can observe the convergence of the drag coefficient as the number of elements increases towards the upper limit. Note that whilst this grid independence study has given a converged drag coefficient value for the purposes of comparison, which is assumed as the ‘exact’ solution for the set of partial differential equations describing this particular flow, whether the value is ‘correct’ in the physical sense is unknown due to the lack of experimental data. It should also be noted that further isotropic refinement of the mesh may in turn lead to further variations in the drag coefficient, but of course such simulations will become increasingly expensive to compute. The data presented in figure 5.19 and Table 5.1 can be used however to verify that the presence of the shock-aligned mesh block has a favourable impact on the numerical solution, relative to the isotropic refinement.

On the chart four individual meshes have been highlighted. The most refined mesh containing around 5.9 million elements is considered as the grid-independent drag coefficient for these flow conditions. The base mesh for feature-alignment, comprising of approximately 0.9 million elements, is the mesh upon which the feature-extraction, medial axis and extrusion processes are applied. This is the coarsest mesh considered for the present study. Other than the addition of background sources along the shock wave curves to cluster and force alignment of the surface mesh elements, the background spacing files for the base mesh and feature-aligned mesh are identical. This means that the spacing of both meshes in regions such as at the leading edge or in the wake will be the same.

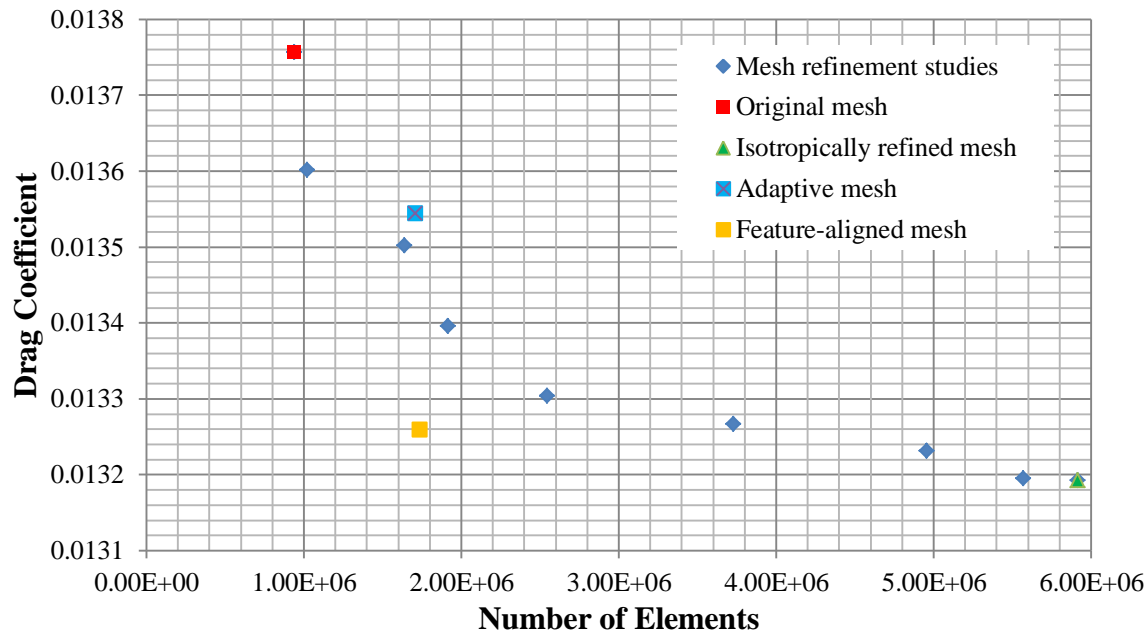


Figure 5.19: Drag coefficient study

The generation of the hex-dominant mesh block aligned with the shock wave has introduced approximately an extra 0.8 million elements into the mesh, leading to a total of 1.74 million elements for the feature-aligned mesh. The resulting final drag has dropped by approximately 5.0 drag counts, matching a drag coefficient which was obtained with an isotropically refined mesh of 3.7 million elements. These results indicate that the feature-alignment process has enhanced the ability of the hybrid mesh to capture and resolve the shock wave, leading to improved accuracy in a more efficient manner. The adapted mesh has also provided an improved drag coefficient, with a decrease of approximately 2.0 drag counts. Improvement on the solution in terms of the drag coefficient is not of the same magnitude as the feature-aligned mesh. This could be due to the highly stretched elements in the shock region degrading the potential for accurate capture of the shock wave and thus substantial improvement in the final drag coefficient.

Mesh	Number of Elements (x 10⁶)	Drag Coefficient	Lift Coefficient
Original mesh	0.94	0.013757	0.2032
Refined mesh 1	1.02	0.013602	0.2035
Refined mesh 2	1.64	0.013502	0.2035
Refined mesh 3	1.91	0.013395	0.2034
Refined mesh 4	2.55	0.013304	0.2031
Refined mesh 5	3.73	0.013267	0.2033
Refined mesh 6	4.95	0.013231	0.2031
Refined mesh 7	5.57	0.013195	0.2035
Isotropically refined mesh	5.91	0.013193	0.2034
Feature-aligned mesh	1.74	0.013259	0.2044
Adaptive mesh	1.71	0.013544	0.2048

Table 5.1: Comparison of drag and lift coefficients

Further improvements in the drag coefficient may be possible through adaptation of the feature-aligned mesh. The feature-aligned mesh has only modified the original mesh in the shock wave region, and since it has been observed that the resolution of the mesh at the leading and trailing edges can substantially affect the final result, intuitively one might expect that subsequent refinement of the feature-aligned mesh in these regions could further improve the solution.

Unfortunately, for the feature-aligned meshes, the TAU refinement techniques for prismatic and hexahedral cells are unable to be employed. This is because the refinement algorithms were not designed to work on these new types of mesh which SOLAR can now generate. For example, TAU identifies different regions of elements such that the appropriate refinement or coarsening template, based on the type of element, is applied. Currently TAU identifies prism and hexahedra piles adjacent to body surfaces which represent the nearfield mesh, but will fail to recognise the mesh block which has grown beyond the nearfield to cover the shock wave, or the mesh blocks present in the wake / vortex region. Therefore the TAU adaptation module is currently not applicable within the feature-aligned mesh blocks. The nodal movement scheme implemented in SOLAR during

the period of this thesis is therefore applied within the hex-dominant mesh blocks (with care to avoid distorting the buffer mesh). Unfortunately, the present nodal movement scheme is unsuitable for the viscous transonic shock cases, due to the difficulty to adapt the nearfield mesh in three-dimensions. The proximity of adjacent nodes in the hexahedra piles means it can be difficult to maintain the validity of the elements when the nodes are displaced. Furthermore, adapting the surface mesh and the volume mesh separately is difficult for hybrid meshes. However, the nodal movement scheme was successfully implemented and tested for the feature-aligned hexahedral mesh blocks for wakes and wing tip vortices, presented in Chapter VII. The tetrahedral adaptation schemes implemented in TAU are only valid for the surrounding farfield tetrahedra, but could also be applied on the feature-aligned meshes outside of the hex-dominant mesh blocks if desired.

Figure 5.20 compares the shock resolution at four spanwise stations for the feature-aligned mesh. The feature-aligned mesh provides better resolution of the shock, compared with the original mesh (figure 5.18), due to the nature of the high-quality cells within the mesh block. The resolution of the weaker shock wave near the trailing edge is also significantly improved. The adaptive mesh also improves the shock resolution compared with the original mesh, as shown in figure 5.21. Figure 5.22 displays the adaptation on the surface of the wing. Above the nearfield portion of mesh, the highly stretched tetrahedra can be observed, which have failed to provide the same level of resolution of the shock compared with the feature-alignment approach.

Finally, a comparison of the convergence histories for the feature-aligned mesh, adaptive mesh and the refined mesh that gave the closest drag coefficient to the feature-aligned mesh (i.e. refined mesh 5) are shown in figure 5.23. The small spike in the feature-aligned and adaptive mesh residuals occur when restarting the solution after interpolation of the original solution onto the new mesh. Both the feature-aligned and adaptive meshes converge considerably faster than the refined isotropic mesh. There is not a considerable difference between the convergence rates for the feature-aligned and adaptive meshes. The adaptive mesh residual can be seen to temporarily oscillate during the adaptive cycles, due to the interpolation step associated with every cycle. After the adaptation is complete, the residual converges at a slightly slower rate than the feature-aligned mesh. However, the skewed elements of the adaptive mesh in this case appear to have no detrimental effect on the convergence as was observed in Chapter IV for the two-dimensional triangular adaptive

case. It must be pointed out that the nearfield mesh for the present case is fully hexahedral / prismatic, whereas the two-dimensional mesh was completely triangular. The convergence behaviour of a fully anisotropic adaptive tetrahedral mesh therefore remains to be seen, although it is unlikely that a three-dimensional viscous simulation would be run using a fully tetrahedral mesh. The use of such a mesh would be more appropriate for an inviscid case where near-wall viscous effects do not need to be resolved.

It is necessary to make a comment about the relation between mesh resolution, pressure distributions and the lift coefficient. In the construction of progressively refined meshes it became apparent that the resolution of the mesh, particularly at the trailing and leading edges, had a significant impact on the converged shock location, hence the pressure distribution and the computed lift coefficient. The resolution of the original mesh also affects the ability of the mesh to capture all of the shock complexities. This is an important consideration for geometric representation approaches, which require a fairly well resolved flow feature on the initial mesh.

The current process seeks to refine the mesh in the shock wave regions only. The resolution of the mesh everywhere else in the domain remains the same as the original mesh. Therefore, the initial mesh resolution must be sufficient to predict an accurate shock structure and location. If this is not the case, the refinement may lead to the shock location changing and the refined region being in the incorrect location. Whilst the extrusion process itself is automatic, it is not currently aimed at being an iterative process due to the fact that the entire mesh must be regenerated and this can be time consuming.

Over the range of meshes presented in figure 5.19, the variation of lift coefficient is particularly small and also shows no convergence as the mesh size decreases. These values are presented for each mesh in Table 5.1. The feature-aligned mesh and adaptive mesh provide slightly higher lift coefficients (although the actual difference is about 0.5% which could be considered negligible) compared with any of the isotropically meshes, although the reasons for this small difference are unclear. The only differences between the adaptive and feature-aligned mesh compared with the other meshes are the number and type of elements in the shock wave region.

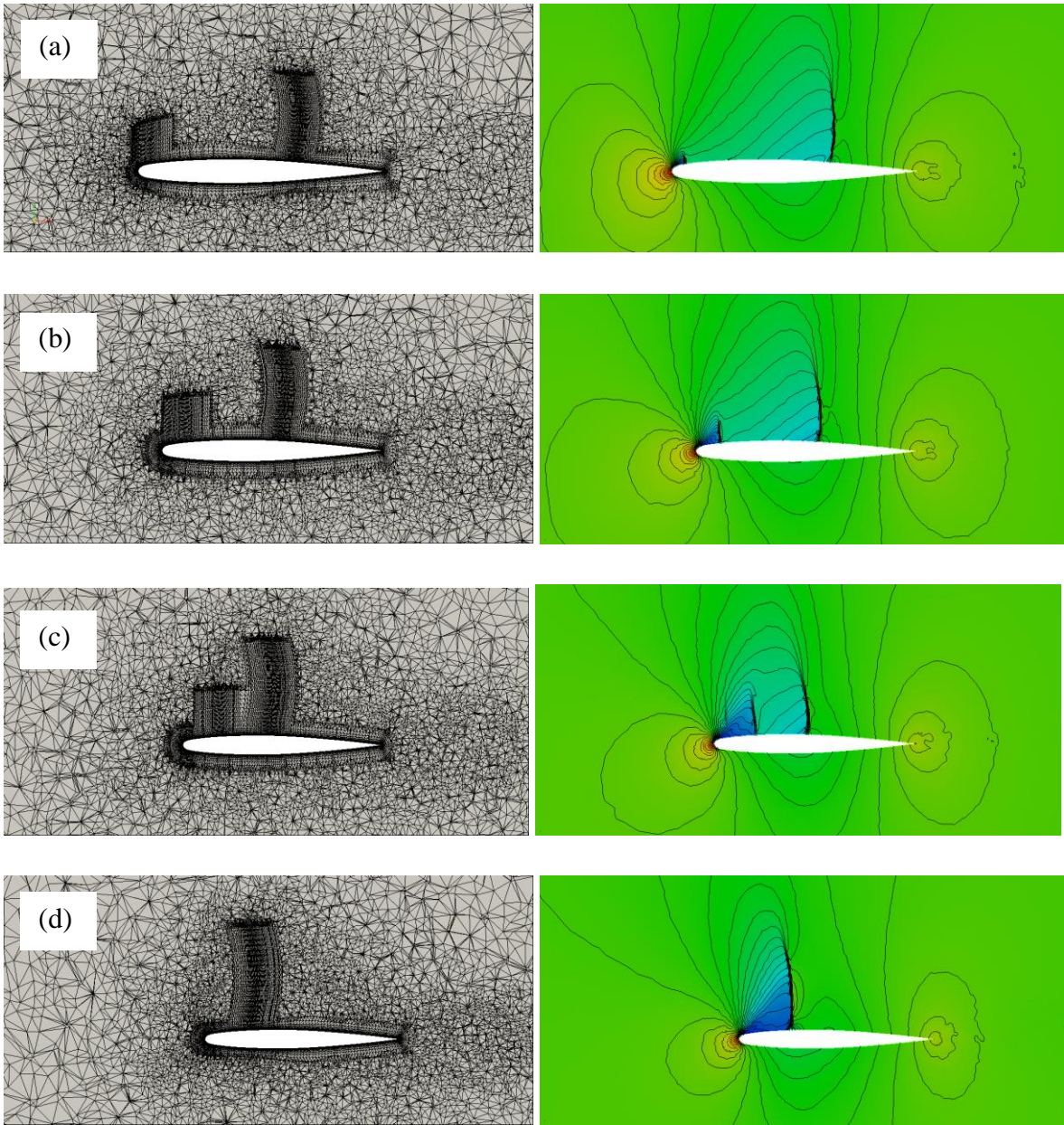


Figure 5.20: Pressure coefficient contours for feature-aligned mesh at four spanwise stations across wing (a) $y/b = 0.2$, (b) $y/b = 0.5$, (c) $y/b = 0.8$ and (d) $y/b = 1.1$

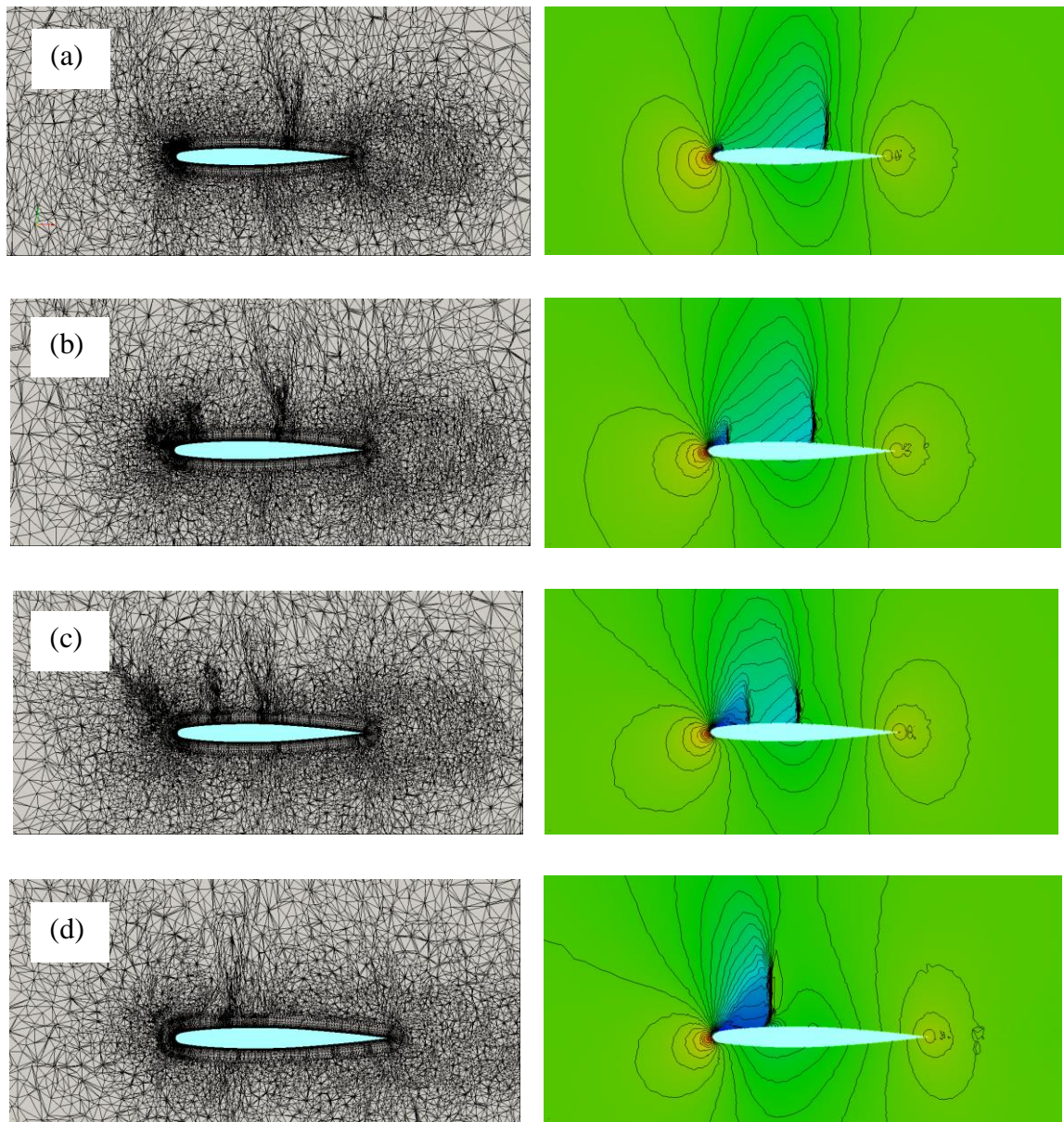


Figure 5.21: Pressure coefficient contours for adaptive mesh at four spanwise stations across wing (a) $y/b = 0.2$, (b) $y/b = 0.5$, (c) $y/b = 0.8$ and (d) $y/b = 1.1$

Whilst the pressure distributions from the coarsest mesh to the finest isotropically mesh are essentially the same, the resolution of the shock waves away from the nearfield is particularly poor, even for the most refined isotropic mesh, due to numerical diffusion smearing the shock. This may explain the range of values observed for the drag coefficient, since the presence of shock waves contributes to the overall drag. How well the numerical solution resolves this feature can therefore be expected to have an effect on the final value. Isotropic refinement will eventually increase the number of points in the shock region to

improve the shock resolution, but this approach is computationally expensive. The focus of the present approach is therefore on reducing the error at the shock through the use of more suitable elements, thus ultimately improving the resolution of the shock and final drag coefficient in a numerically efficient manner.

Drag prediction is an important area of computational aerodynamics. The range in the drag coefficient from the coarsest mesh to the finest mesh is almost 6.0 drag counts (where one drag count is equal to 10^{-4}). A drag decrease can be equated to decreasing the aircraft weight that is required to carry a specified payload over the required distance. Minor changes in the drag can therefore be extremely important: a single drag count can in some cases be the equivalent to the weight of a couple of passengers^[184]. In order to accurately assess designs, the computational methods employed to predict the drag must be reliable. The application of the shock-aligned mesh block insertion method has indicated that improvements in the predicted drag coefficient can be obtained at a fraction of the number of elements associated with isotropic refinement.

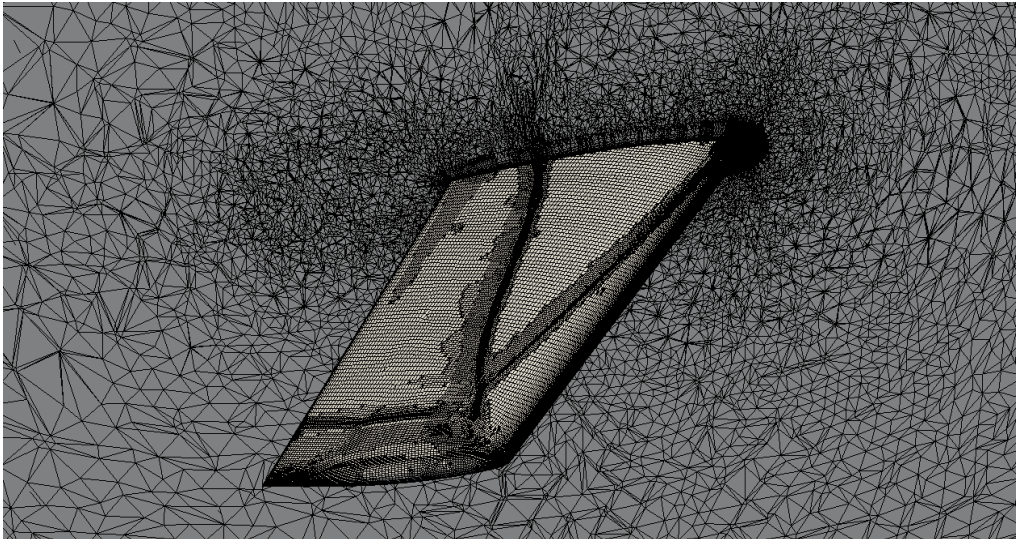


Figure 5.22: Adaptation of the nearfield mesh in the shock region

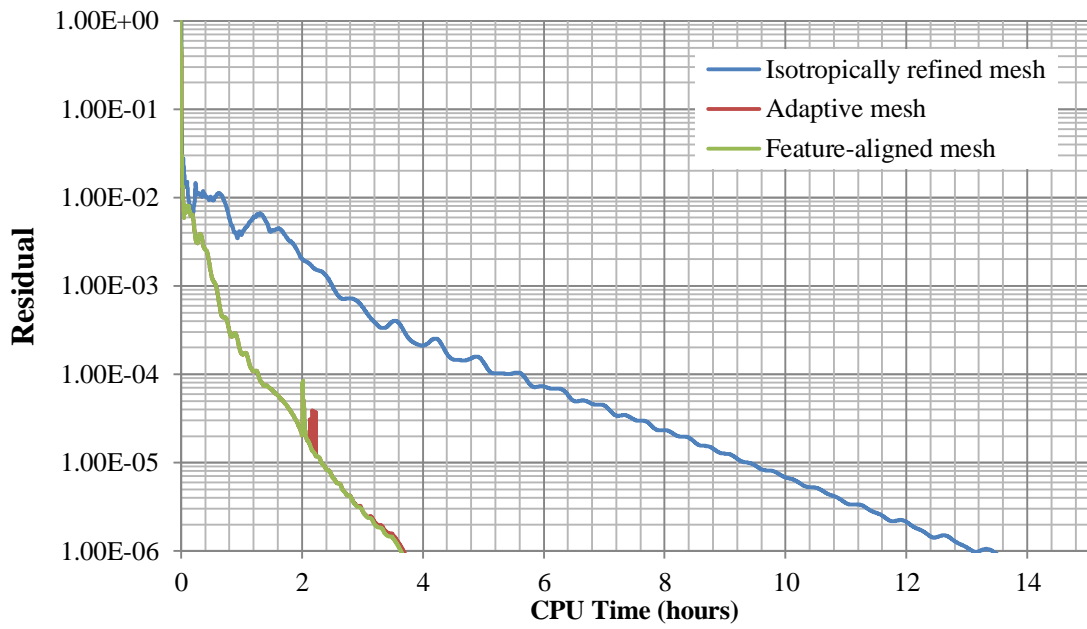


Figure 5.23: Convergence history comparison for feature-aligned mesh, adaptive mesh and isotropically refined mesh

5.4.2 DLR-F4, Transonic Viscous Flow: $M_\infty = 0.75$, $\alpha = 2.0^\circ$, $Re = 3.0 \times 10^6$

Figure 5.24 shows the extracted shock structure for this case. It is also lambda-shaped, but does not include the bifurcation at the wing tip. The points corresponding to the branching shock are at a lower height than the points within the main shock wave, which the mesh block growth process attempts to take into account after decomposition of the extracted shock wave point cloud.

Due to the setup of the geometry for this case, the application of the surface mesh alignment process and extrusion process is not as trivial as it was for the M6 wing case presented previously. The topology of the shock wave is simpler as there is no bifurcation of the shock wave at the wing tip. However, the generation of the surface mesh becomes more complicated as the upper wing surface is made up of multiple zones which are meshed independently, instead of a single zone as with the M6 wing. This means that the medial curves had to be manually split where they passed from one surface to another – they cannot intersect or cross the common curve that the two surfaces share at their intersection. The two zones must also be meshed individually. In order to allow the

extrusion process to be applied across the entire wing surface, the two upper wing surface zones are first merged together to form a single surface mesh, the result of which is displayed in figure 5.25.



Figure 5.24: Extracted shock structure

Whilst generation of a feature-aligned volume mesh for this case appears to have been successful, there were problems in generating a mesh that TAU will accept. The mesh initially appears to be suitable for TAU input, and as indicated by figures 5.26 and 5.27 it can be visualised by post processing software. However, during the TAU element validity and quality checks which are carried out prior to the solution, the mesh is rejected due to a number of invalid elements. The reason for the invalid elements occurring in the mesh has yet to be determined. Due to time constraints unfortunately this issue was not able to be resolved within the timeframe of the PhD and writing of this thesis.

As the mesh was successfully output by SOLAR, the size of the mesh can be reported. The original, non-aligned mesh possessed 2.4 million elements. After application of the feature-alignment process, the mesh block was made up of around 1.4 million elements in total. This includes all the hexahedra and prisms, as well as the pyramids and tetrahedra within the buffer mesh. After generation of the surrounding nearfield mesh and generation of the farfield tetrahedra, the final mesh contained approximately 3.9 million elements. Whilst no solution was obtainable, it is envisaged that similar effects as those

observed for the M6 wing case, certainly in terms of shock wave resolution. The effects on aerodynamic quantities such as the drag coefficient remain to be seen.

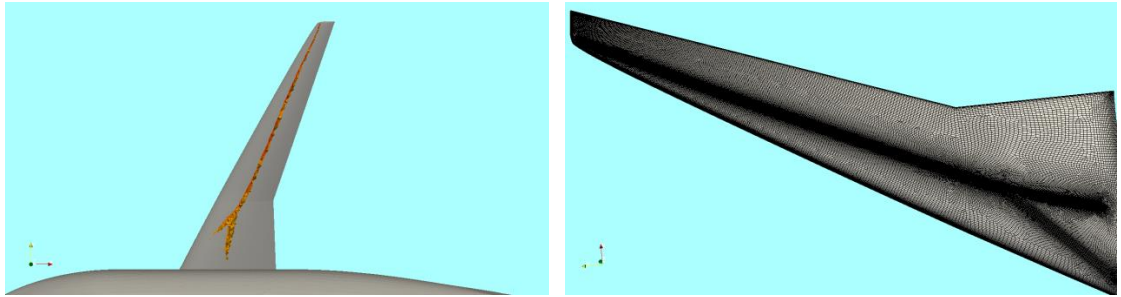


Figure 5.25: View of extracted shock structure and merged upper-wing surface meshes aligned with shock footprint

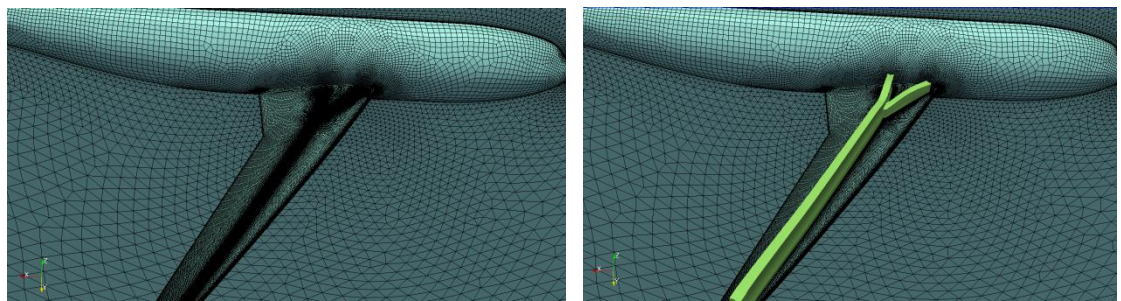


Figure 5.26: Shock aligned surface mesh and embedded shock-aligned mesh block

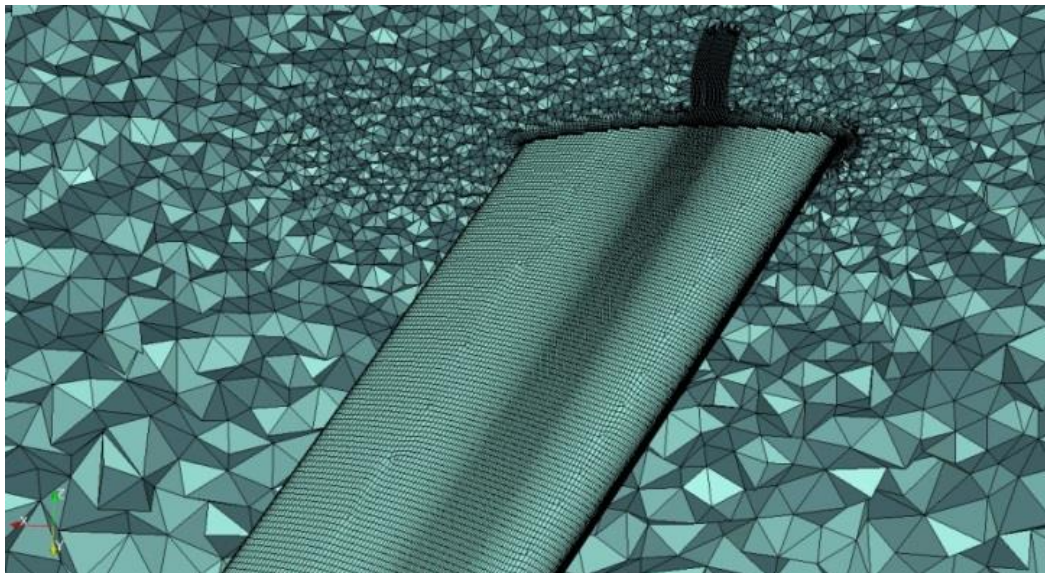


Figure 5.27: Surface aligned mesh and shock-aligned mesh block embedded in hybrid volume mesh

5.5 Conclusions

A method to grow shock-aligned hex-dominant mesh blocks which can be embedded in a conformal manner into a hybrid volume mesh has been presented for transonic shock waves. The method initially involves aligning the surface mesh using the medial axis curves representing the shock footprint. The medial curves allow construction of polygons which are used to decompose the extracted shock data into portions. For each portion of the shock wave, heights and growth curves can be estimated. After identifying sets of surface mesh faces attached to each medial curve, the faces are extruded beyond the nearfield mesh to be grown aligned with the shock wave using the previously computed growth information. The result is a mesh which includes semi-structured regions of hexahedral elements aligned with the complex transonic shock structure.

Compared with global isotropic refinement of the mesh, the inclusion of the shock-aligned mesh block has been shown to provide improvements in shock resolution and drag prediction with significantly lower number of elements. This is due to the increased suitability of the mesh for the approximate Riemann solvers applied across the cell interfaces. An indication of the improved efficiency can be observed from the comparison of the convergence histories, where the feature-aligned mesh converges more rapidly than the isotropically refined mesh. The method also provided improved results compared with the anisotropic adaptive hybrid mesh, relative to the most refined isotropic case.

CHAPTER VI:

Feature-Aligned Hex-Dominant Mesh Block Insertion: Part II: Wing Tip Vortices and Wakes

6.1 Introduction

Predicting the behaviour and lifetime of wing tip vortex flows offers a significant challenge in computational aerodynamics and has been the subject of a number of studies^[73, 186-170, 195, 196]. Wing tip vortices can persist for long periods of time and can present safety concerns for following aircraft. It is therefore important that such features can be effectively modelled and resolved by simulations. Successful resolution of the vortices can also allow assessment of different wing tip devices to be effectively made. From a numerical point of view, the capture and resolution of a vortex is challenging from two perspectives – capturing the boundary layer over the wing which eventually rolls up to form the vortex, and then resolving the vortex downstream^[186]. Whilst advancements have been made to allow for improved vortex preservation in CFD solutions, it can be considered to be an on-going challenge since artificial diffusion can be difficult to prevent, particularly at large distances downstream of the vortex source^[73]. The accurate capture of wing tip vortices will require a suitable mesh and turbulence model in combination. Grid size, cell aspect ratio and skewness of the mesh are contributing factors in vorticity capturing^[187, 188]. Therefore, aligning the mesh with the vortices as much as possible is necessary.

In the context of standard hybrid meshes, the prismatic or hexahedral elements in the nearfield can provide excellent resolution of the boundary layer and therefore provide suitable initial conditions for the vortex development. Whilst the high density regions of isotropic tetrahedra close to the surface will continue predicting the development of the vortex, as the tetrahedra grow in size, the rate of numerical diffusion also increases. This causes the trailing vortex to quickly disappear from the solution. The lack of alignment of the unstructured cells with the shear layers also contributes to this diffusive behaviour. It is therefore necessary to apply adaptation or specialised mesh generation techniques to improve the capability of the mesh to capture the vortices present and resolve them deep into the farfield.

The present chapter describes a method to embed semi-structured hex-dominant mesh blocks into the domain through extrusion of quad-dominant surface meshes. These surfaces are embedded at suitable locations within the domain and the extrusion direction can be determined from the extracted data in the original solution. The approach is applied on the two transonic cases presented in the previous chapter and finally a transonic Delta wing case at a high angle of attack. Note that the methodology is the main subject of this chapter, with some preliminary results. The performance of the feature-aligned meshes with isotropically refined meshes and anisotropically adapted meshes is presented and discussed in Chapter VII.

6.2 Surface Mesh Extrusion

The extrusion process for wing tip vortices and wakes is similar in some respects to the previously described approach applied to shock waves. After definition of suitable surfaces at the desired locations, the resulting quad-dominant surface meshes can be extruded along the vortex or wake path. Unfortunately, whilst the extrusion process in itself is an automatic process, the user must initially define suitable locations for the surfaces which serve as the source of the extrusion. Since some user-interaction is required to set up the geometry for a SOLAR-TAU process anyway, a little more time invested in the construction of these surfaces (which once constructed for a particular geometry can simply be copied for other similar problems), can yield significant improvements in the resolution of the wing tip vortices and wakes due to the high quality semi-structured hex-dominant mesh block present. After generation of the source surface mesh, the direction and expansion of the extrusion is determined automatically using extracted data from the solution.

6.3 Defining Source Surfaces

Suitable surfaces need to be defined prior to the extrusion taking place. The way the surfaces are set up depends on the flow feature under consideration.

6.3.1 Wing Tip Vortices

For the wing tip vortex case, a simple circular surface is created and embedded close to the wing tip as shown in figure 6.1 (note that the surface can be an arbitrary shape

but a circular form is the most appropriate in this instance). The surface is treated like any other by the surface mesh generator, and as such the feature curve functionality which has been used extensively for surface mesh alignment can be applied again. In this case, the feature line is simply a concentric circle which is connected to the source surface. The effect of this is to create rows of high quality quadrilateral elements in the radial direction.

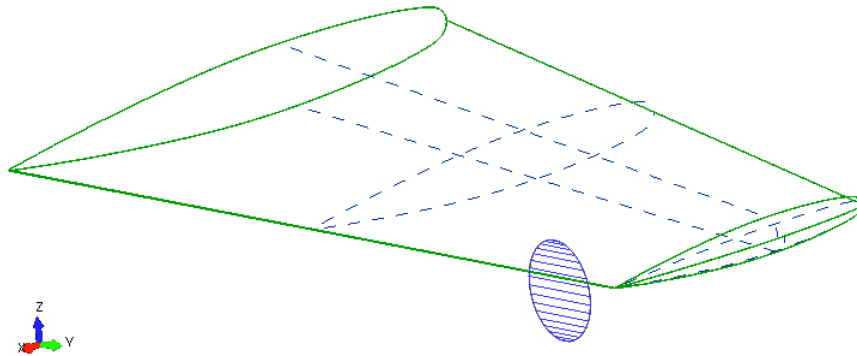


Figure 6.1: Typical source surface setup for capture of wing tip vortex

6.3.2 Trailing Wakes

For simple wing geometries, a typical source surface could be as simple as a rectangle which covers the length of the trailing edge, as shown in figure 6.2. The surface is extended a small distance beyond the wing tip to also ensure capture of the vortex when it forms. For more complex wing geometries with multiple trailing edge surfaces, the definition of two or more surfaces may be required. Any number of surfaces can be connected, and despite being meshed separately they will be extruded as a single zone. Once again, the feature curve functionality can be exploited to improve the quality of the source surface mesh to allow semi-structured quadrilateral elements to be generated. For wakes, a curve aligned with the longest edges of the surfaces, along with appropriate choice of spacing values for the background sources can yield a surface mesh which almost completely resembles a structured quadrilateral mesh.

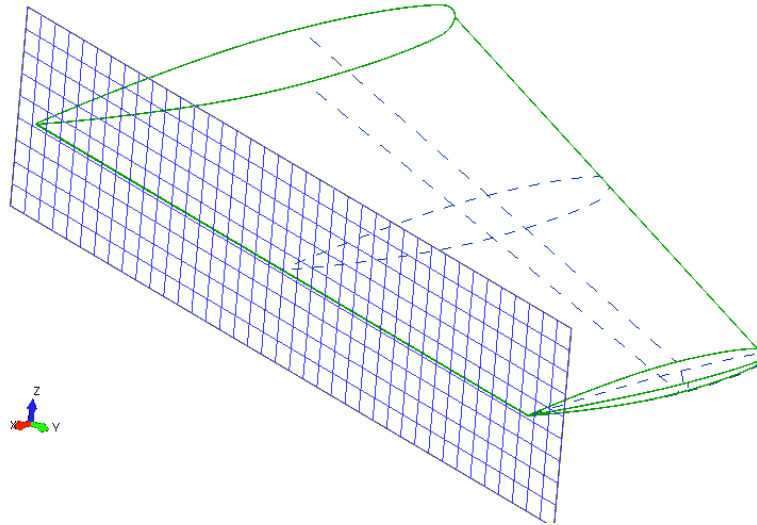


Figure 6.2: Example source surface setup for capture of trailing wake for simple wing geometry

6.4 Determining the Extrusion Direction

After a solution has been obtained with the original mesh and the source surfaces have been defined, the flow features of interest may be extracted using the techniques described in Chapter III. The direction of the extrusion is simply defined by a three-dimensional growth vector. As the extrusion always occurs downstream (i.e. along the x -axis) the x component of the vector is always 1. Therefore all that remains is to compute the other two components. This is achieved by projecting the extracted data set firstly onto the xz -plane followed by the xy -plane to form two-dimensional point sets. Straight lines are fitted through the extracted data. In the xz -plane, the minimum and maximum y -coordinate values of the line are found, whilst in the xy -plane, the minimum and maximum z -coordinates are found. It is trivial to then determine the other two components of the extrusion vector using this information. This process is displayed in figure 6.3.

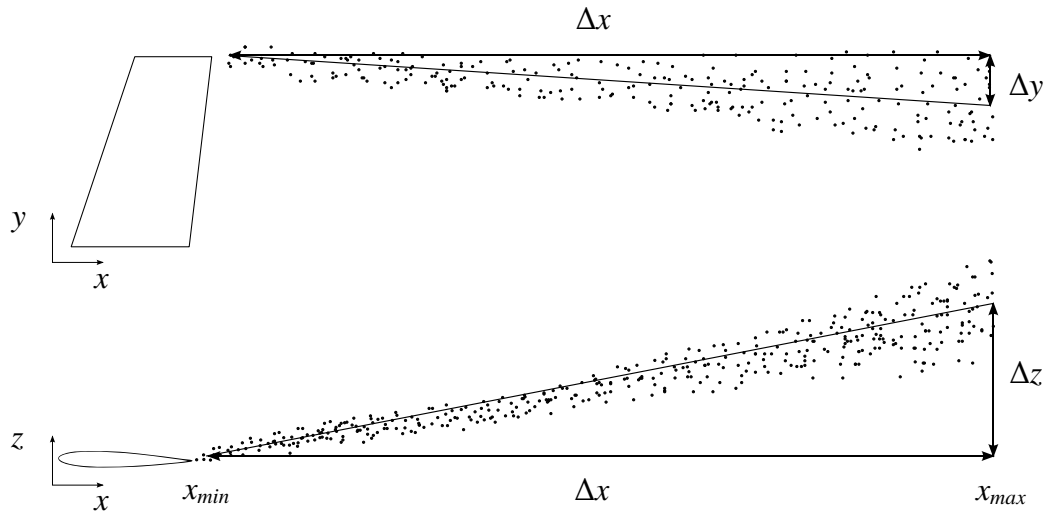


Figure 6.3: Determining the extrusion direction in the planes

For wake block extrusion only one component is required to be computed since the extrusion must occur downstream and parallel with the symmetry plane. This is in contrast with the vortex case where the extrusion does not need to be aligned with the symmetry plane. Whilst it is the user's responsibility to ensure the source surface has been set up in the appropriate location, the extrusion process will automatically warn the user if the block intersects any existing geometric entity. When this occurs, the extrusion process will cease adding more layers, but the volume meshing will continue.

6.5 Mesh Block Growth and Expansion

The user chooses a suitable number of layers for each of the mesh blocks. The layer size is constant. An expansion factor can also be defined which is applied to each component of the growth vector to allow the block to expand in that direction. Since a separate factor can be applied to each component of the growth vector, the block can expand at different rates in different directions. For all extrusions, the extrusion direction is downstream i.e. along the x -axis. However, the type of surface used (circular for wing tip vortex or rectangular for wake *and* wing tip vortex), will affect the directions the mesh block is allowed to expand.

For wing tip vortices (circular surfaces), the expansion factor can be applied in the y and z directions. Expanding the block excessively in one direction only can be detrimental to final mesh block quality. Whilst different expansion factors can be applied to each component of the growth vector, once an expansion factor has been calculated, it remains constant for each component. The expansion functionality is useful for capturing wing tip vortices which grow in size with increasing distance downstream as displayed in figure 6.5. This allows the entire vortex to be captured throughout the mesh block.

For rectangular wake surfaces, which are embedded to capture both the trailing shear layers and the wing tip vortex (see figure 6.2), the source surface is constrained to expand along the z -axis only. This is due to the fact that expansion in the y direction may cause the mesh block to intersect existing geometric entities, such as the symmetry plane or fuselage. Such a constraint is not necessary for the vortex extrusion as the vortex mesh is unlikely to collide with other entities.

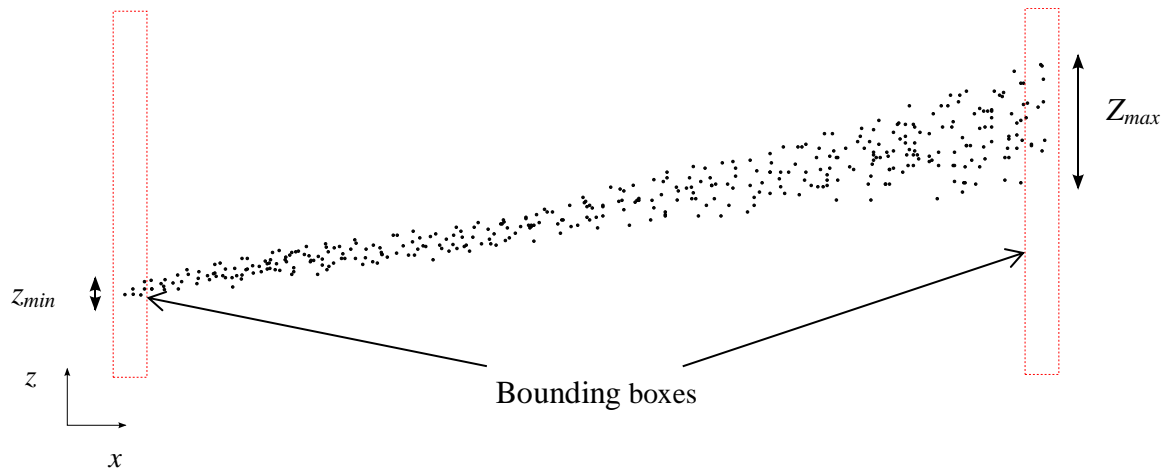


Figure 6.4: Determining the expansion factor

The expansion rate is determined from the extracted data set by considering distances between points at the extremities of the data set. Using information from the previous step, i.e. the identification of the extrusion direction, a group of points are identified at the minimum and maximum x locations. To identify these points, bounding boxes are placed in the vicinity of these identified locations, as displayed in figure 6.4. The bounding box functionality is a class in SOLAR that includes a test to see if a point exists within the box. In terms of the box dimensions, the x -coordinate is some Δx from the x_{min} (see figure 6.3) and the y and z -coordinates of the box are bound by the farfield surfaces of

the domain. Each bounding box can therefore be imagined as a slice of the domain, with a small region of points enclosed within.

After identification of the bounding box points, the maximum distance between points in the z -direction are estimated by simply comparing the coordinates of pairs of points. The ratio of these values obtained from the two bounding boxes, i.e. z_{max} / z_{min} is used to estimate an expansion factor. This operation provides an approximate expansion in one direction only, i.e. along the z -axis. However, as previously described, to maintain the quality of the mesh block it must expand at the same rate in the y direction also. The expansion is therefore applied in both directions.

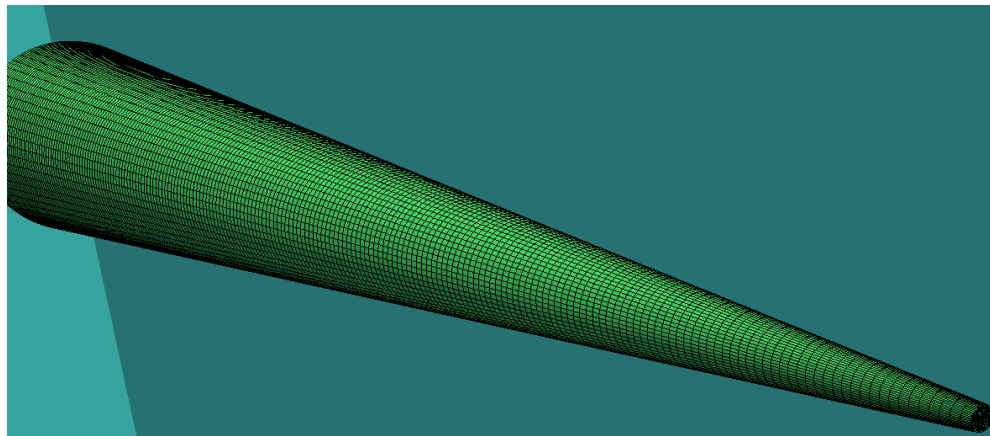


Figure 6.5: Expansion of mesh block with increasing downstream distance

The current method is fairly simplistic in that the extrusion and expansion can only occur along linear directions. However, the methods described above worked satisfactorily for the cases tested. As an alternative approach, the user may also provide extrusion and expansion data in the SOLAR control file. The extruded mesh can also be written out independently of the rest of the mesh for visualisation purposes.

6.6 Merging the Mesh Block into the Hybrid Volume Mesh

As with the shock wave extrusion process described in the previous chapter, after generation of the hex-dominant mesh block, a buffer mesh is required to allow for a conformal interface with the surrounding tetrahedra. This is done by adding a layer of pyramids to the exterior of the mesh block, followed by tetrahedra which connect the apexes of the pyramids and form a fully triangulated outer surface. Once the farfield has

been generated, the boundary recovery process must occur in order to force the tetrahedra to conform to the surface tessellation so that it appears in the final volume mesh.

6.7 Test Cases

6.7.1 M6 Viscous Transonic Flow: $M = 0.8395$, $\alpha = 3.06^\circ$, $Re = 11.72 \times 10^6$ ^[183]

6.7.1.1 Wing Tip Vortex Extrusion

The initial mesh provides extremely poor resolution of the trailing shear layers and vortex when visualising slices of different flow variables at various downstream locations - an example is shown in figure 6.6 for eddy viscosity. Within a short distance downstream of the trailing edge of the wing, the shear layers have almost completely disappeared. This numerical diffusion has prevented the wing tip vortex from properly forming and being resolved, so it is difficult to visualise. These problems occur despite a high quality nearfield mesh which exists to resolve the viscous layers attached to the wing surface. Figure 6.6 shows that the shear layers are well resolved near the wing surface, but the transition from hexahedra to tetrahedra, which quickly grow in size, enhances the effect of numerical diffusion. However, despite the relatively large tetrahedra in the farfield, the λ_2 criterion has been found to work effectively in extracting the vortex region to allow extrusion information to be computed. Figure 6.7 displays the result of the vortex extraction operation on the original mesh. Due to the size of the tetrahedra in the farfield, this extraction process may not provide a ‘true’ indication of the vortex core, but the data provided is sufficient to allow the embedded surface mesh extrusion to occur.

The source surface shape chosen for wing tip vortices is circular, as displayed in figure 6.8. As previously mentioned, the surface itself could be any shape, but a circular form is the most appropriate. The surface is located a small distance away from the wing tip at the trailing edge. Since SOLAR treats this surface like any other surface, the previously described approaches of embedding curves (described in Chapter IV) within the surface to influence the formation of the surface mesh can be employed. Whilst figure 6.9(a) shows a standard unstructured quad-dominant surface mesh, the inclusion of an embedded circle can allow for greater control of the mesh formation in the radial direction. This method is employed for all circular surface meshes to ensure a high quality source surface mesh for extrusion.

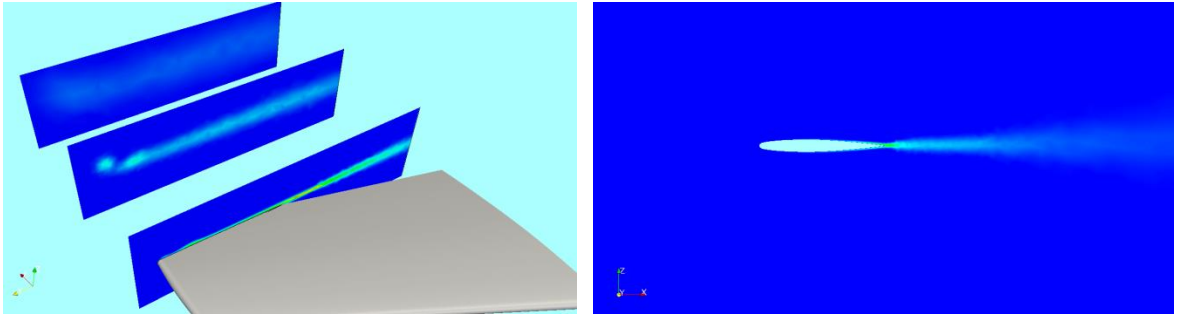


Figure 6.6: Resolution of the trailing shear layers for the original mesh – eddy viscosity contours

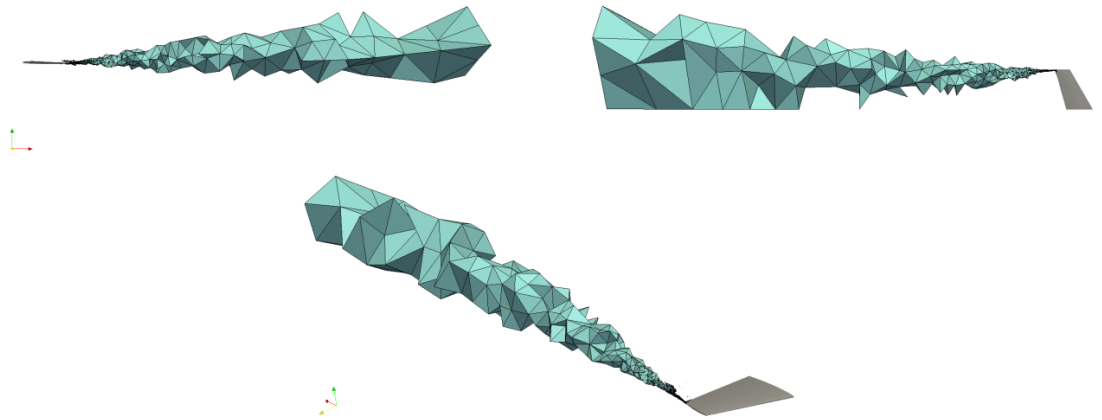


Figure 6.7: Extraction of the trailing vortex using the λ_2 criterion

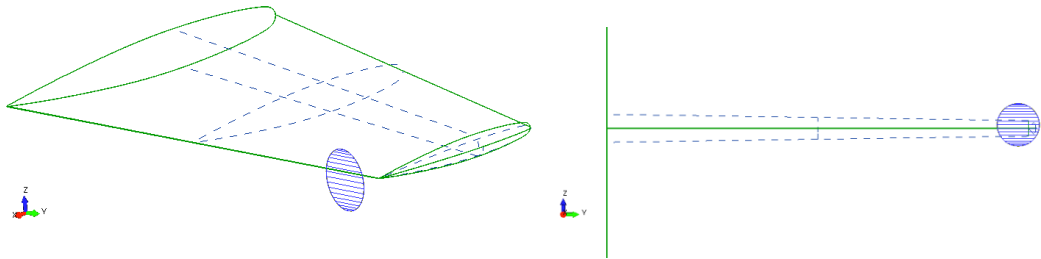


Figure 6.8: Circular source surface insertion at the wing tip

The extrusion expands with increasing downstream distance to account for the increase in the size of the vortex. Figure 6.10 shows the tube of hex-dominant mesh embedded into the hybrid volume mesh. The resulting improvement in the resolution of the

vortex and its preservation downstream is displayed in figure 6.11. It can be observed that the method has worked well in capturing the roll up of the shear layers at the tip to form the initial vortex structure and eventually the formation and preservation of the vortex core. Unfortunately, the majority of the wake region is still resolved by isotropic tetrahedra. The presence of these elements causes the shear layers across the rest of the wing to quickly dissipate. To remedy this, the source surface used for extrusion can simply be changed to a rectangular surface embedded across the length of the wing trailing edge. This is considered in the next sub-section.

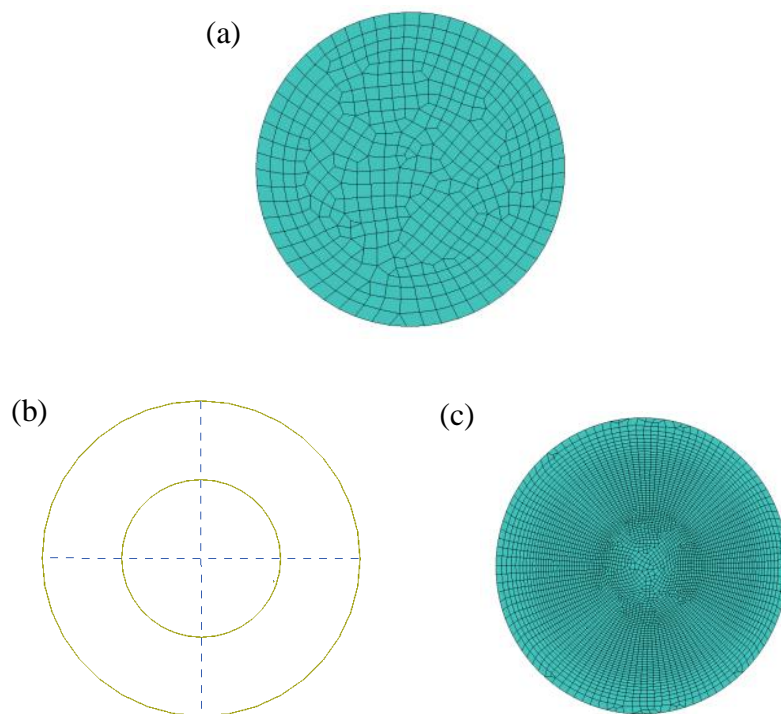


Figure 6.9: (a) Standard quad-dominant source surface mesh, (b) mesh control curve embedded in source surface, (c) resulting surface mesh

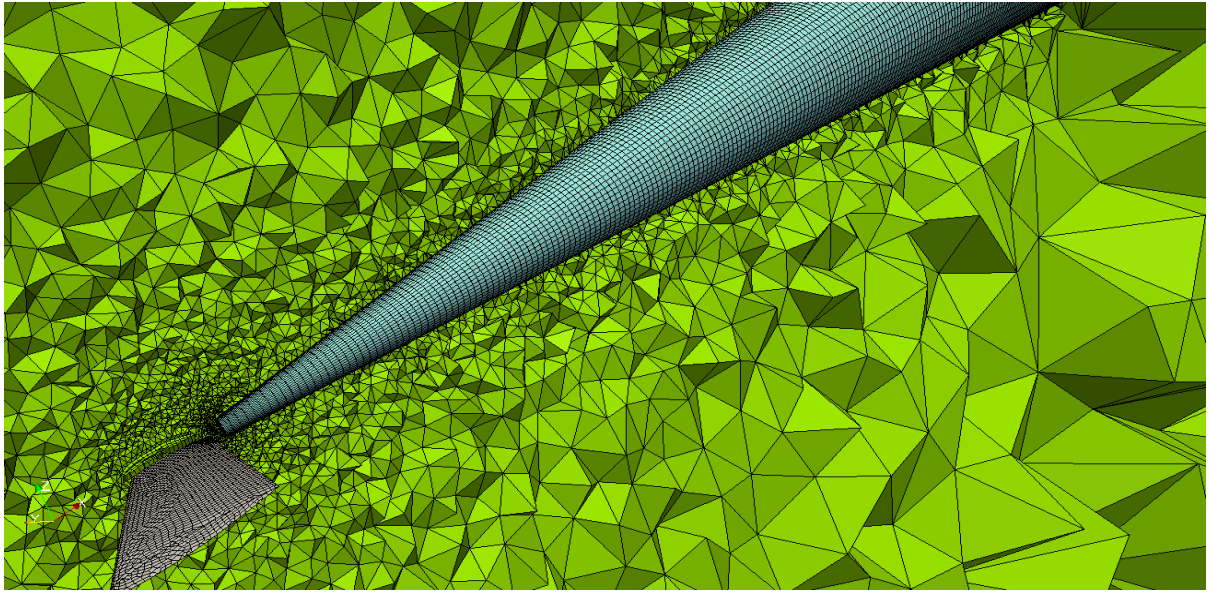


Figure 6.10: Embedded block of hex-dominant mesh for wing tip vortex

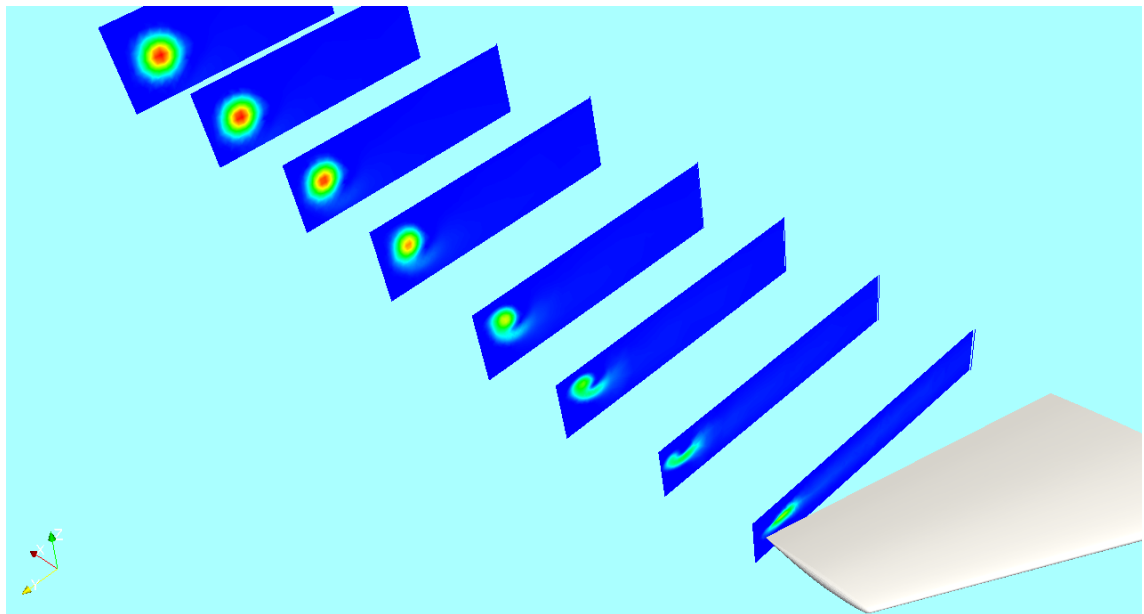


Figure 6.11: Resolution of wing tip vortex on feature-aligned mesh – eddy viscosity contours

6.7.1.2 Wing Tip Vortex and Wake Extrusion

The wake in this case is extracted by considering points in the mesh with an eddy viscosity greater than a small non-zero value, 10^{-4} for example. As a result, the extracted data set shown in figure 6.13 corresponds to the shear layers and possibly part of the wing tip vortex. In a similar fashion to the surface insertion for the wing tip vortex case, a

rectangular surface is now embedded across the length of the trailing edge region, as shown in figure 6.14, where the surface overlaps the wing tip. This is necessary so that the hex-dominant block also eventually covers the wing tip vortex region downstream of the wing.

Extruding two separate surfaces to cover the extent of the wake and vortex separately is more difficult to implement with the current approach. The main reason for this is that extrusion of the two surfaces separately can lead to eventual intersection of the mesh blocks. The approach adopted for the present study was initially to merge both surfaces and extrude the resulting surface mesh. However, a problem became apparent if the surfaces were of different shapes, as displayed in figure 6.12. This set up was originally intended for capture of both wake and wing tip vortex. The problem is that since the rectangular surfaces are constrained to expand in the z -direction only to avoid collision with other geometric entities such as the symmetry plane or fuselage, the circular surface must also expand in this direction. The resulting circular portion of the mesh block is then excessively stretched in the z -direction, resulting in poor quality elements. As the subsequent examples show, use of suitably located rectangular surfaces alone for the capture of both wake and wing tip vortices has appeared to improve the resolution of both features significantly.

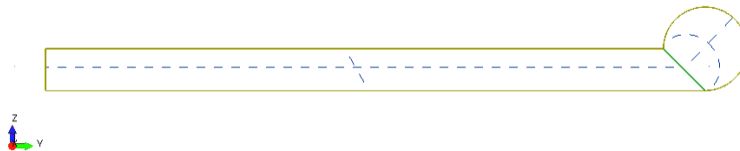


Figure 6.12: Combined rectangular and circular surfaces – expansion of the circular surface in the z -direction only leads to an excessively stretched mesh block and poor mesh quality

To allow for high quality formation of the quad-dominant surface mesh, a curve can be embedded into the surface in a similar fashion to previously presented cases. After application of suitable anisotropic sources, a refined region within the source surface mesh can be obtained, as displayed in figure 6.15. This refinement will be maintained throughout the extrusion to allow for capture of the trailing shear layers throughout the entire trailing edge region of the wing.

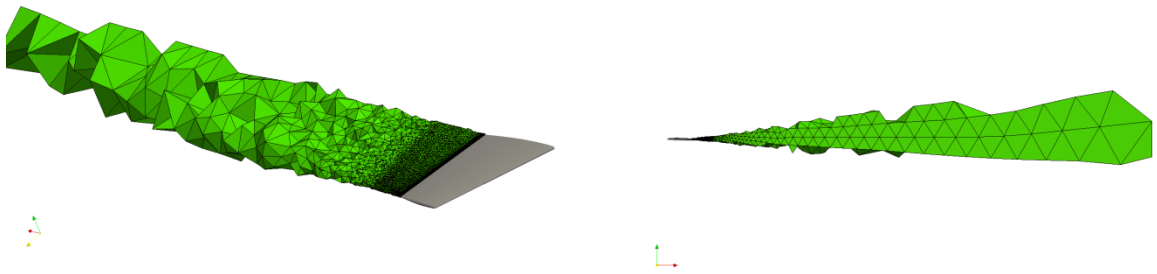


Figure 6.13: Extracted wake points from original mesh

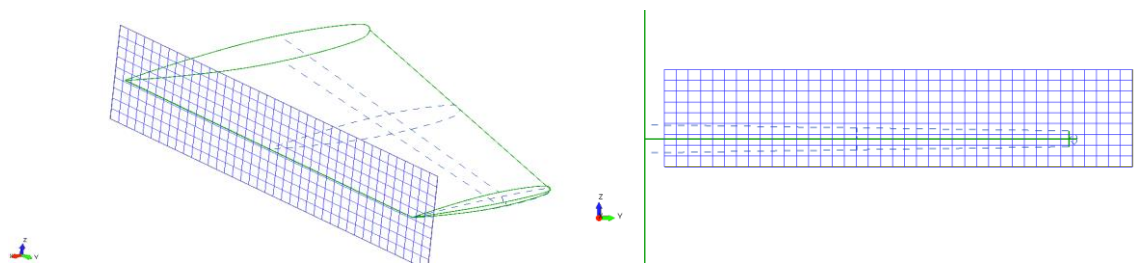


Figure 6.14: Source surface insertion along the trailing edge for capture of wing tip vortex and wake

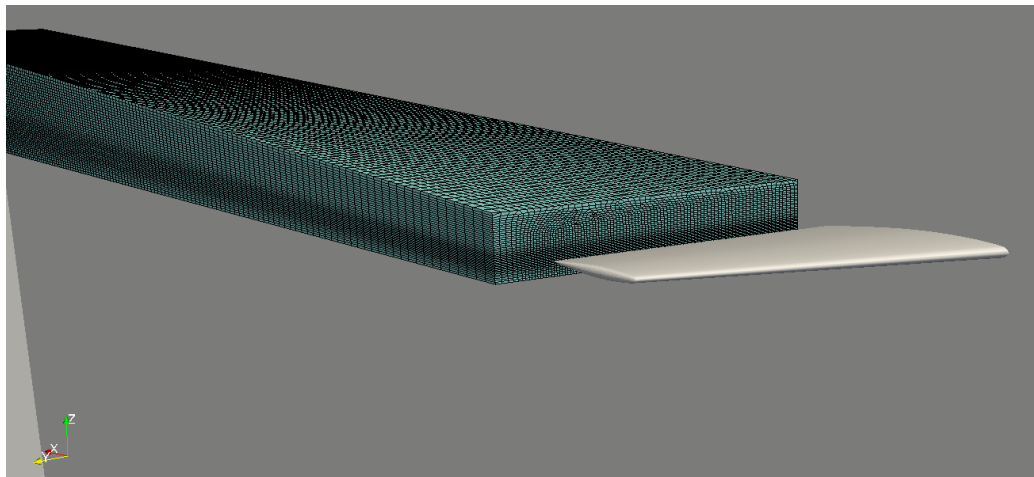


Figure 6.15: Extruded hex-dominant wake block prior to generation of the farfield mesh

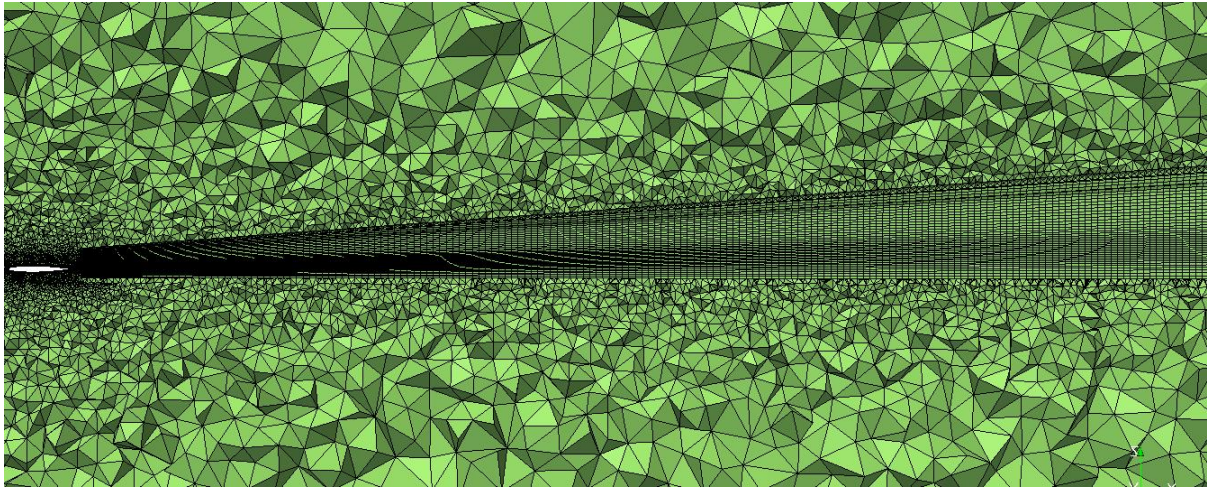


Figure 6.16: Slice of the hex-dominant mesh block embedded in the hybrid volume mesh for the ONERA-M6 wing case

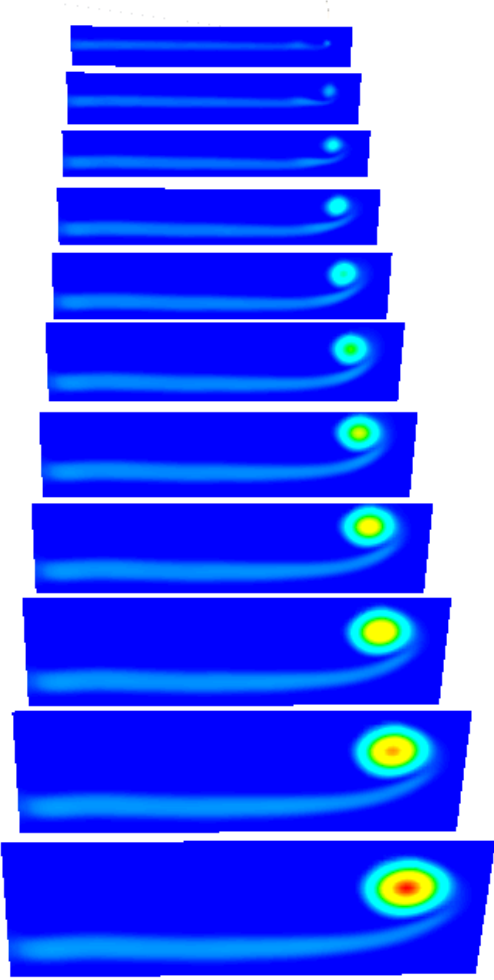


Figure 6.17: Preservation of the wake and wing tip vortex downstream using the feature-aligned mesh – eddy viscosity contours

The resulting feature-aligned mesh is displayed in figure 6.16, and the refined region is clearly visible, as well as the expansion of the mesh block downstream to allow for capture of the vortex as it grows. Figure 6.17 displays the preservation and resolution of the entire trailing shear layer rolling over to form the wing tip vortex using the feature-aligned mesh. Finally, the eddy viscosity contours of the vortex structure approximately fifteen chord lengths downstream of the wing trailing edge can be observed in figure 6.18. The quality of the feature resolution is clear. Such resolution would be impossible on standard hybrid meshes at such a distance downstream of the vortex formation.

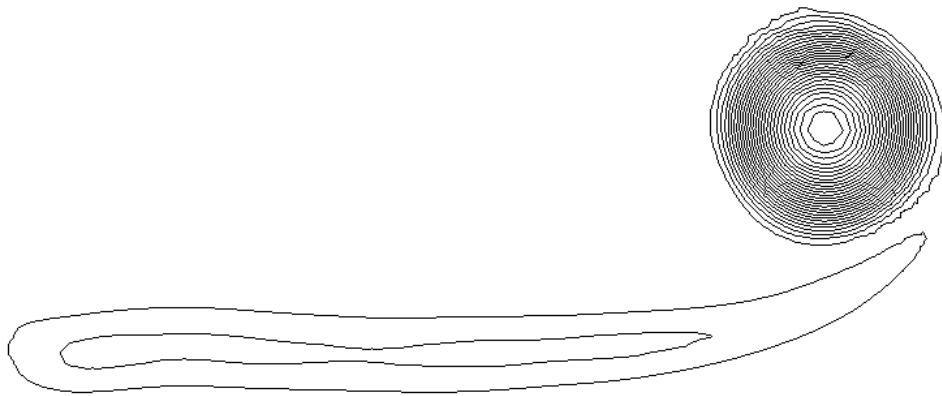


Figure 6.18: Eddy viscosity contour at approximately fifteen chord lengths downstream of trailing edge

6.7.2 DLR-F4, Transonic Viscous Flow: $M = 0.75$, $Re = 3.0 \times 10^6$, $\alpha = 2.0^\circ$

For the present test case the extrusion process for the wing tip vortex alone is not performed. The reason for this is that the effect of the insertion of a wing tip vortex mesh by itself has already been presented in section 6.7.1, where it was observed that the vortex resolution was considerably improved, but the trailing wake rapidly disappeared from the solution. Instead, a rectangular source surface is embedded to capture the entire free shear layer region and wing tip vortex. The extracted data set is displayed in figure 6.19. Since the wing geometry is different in this case, the setup of the surfaces is modified and requires two surfaces. The orientation of the two surfaces is aligned with the wing trailing edge as displayed in figure 6.20. After generation of the high-quality semi-structured quadrilateral source surface mesh, shown in figure 6.21, the extrusion can occur through the

extracted wake. The quad-dominant surface mesh can be observed to be generally semi-structured, which aids with the alignment of the mesh block with the trailing shear layers. Figure 6.22 shows the hex-dominant mesh block prior to generation of the volume mesh.

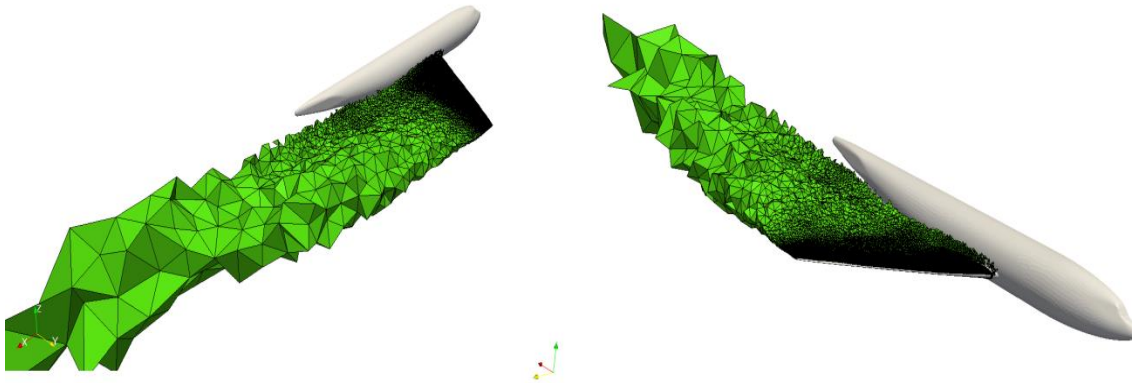


Figure 6.19: Extracted wake points from the original mesh

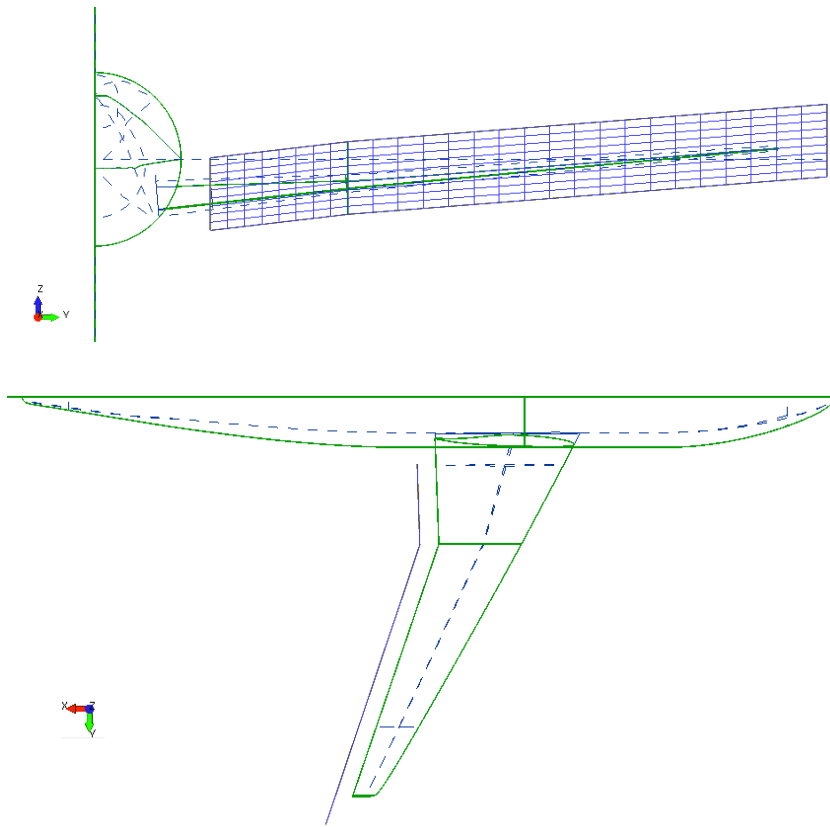


Figure 6.20: Source surface insertion at wing trailing edge

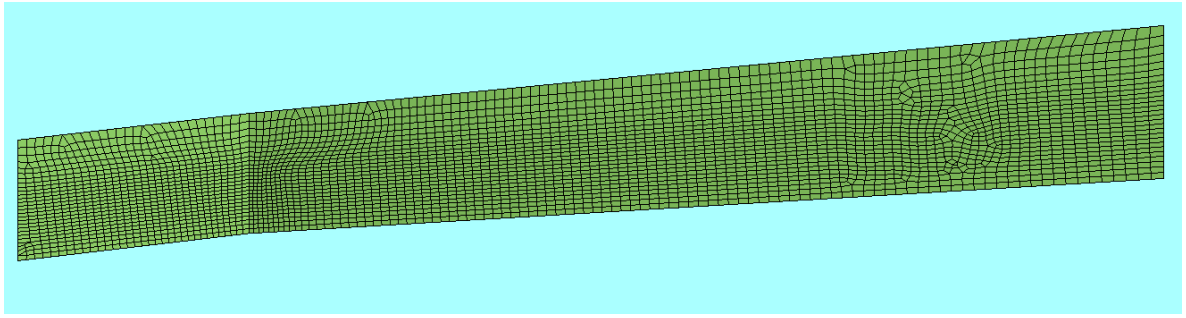


Figure 6.21: Source quad-dominant surface mesh

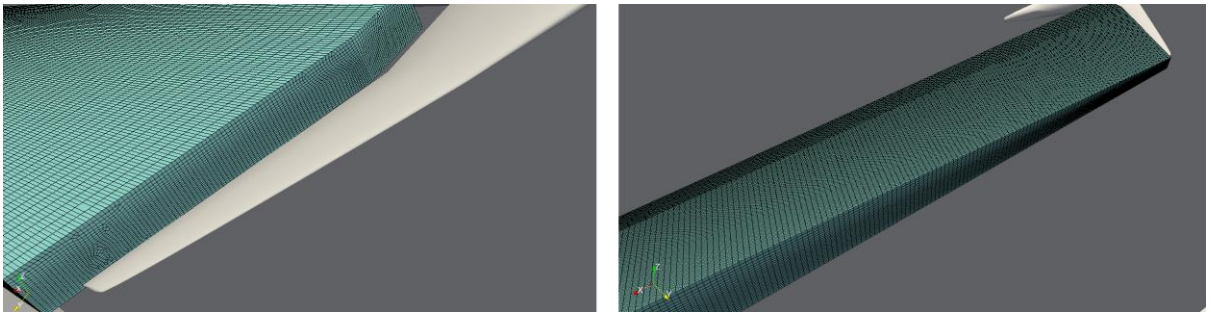


Figure 6.22: Extruded hex-dominant wake block prior to generation of the farfield mesh

A slice of the final hybrid volume mesh is displayed in figure 6.23. The presence of the mesh block can be observed to extend far downstream which will allow for preservation of the shear layers for a considerable distance into the domain, towards the farfield boundary.

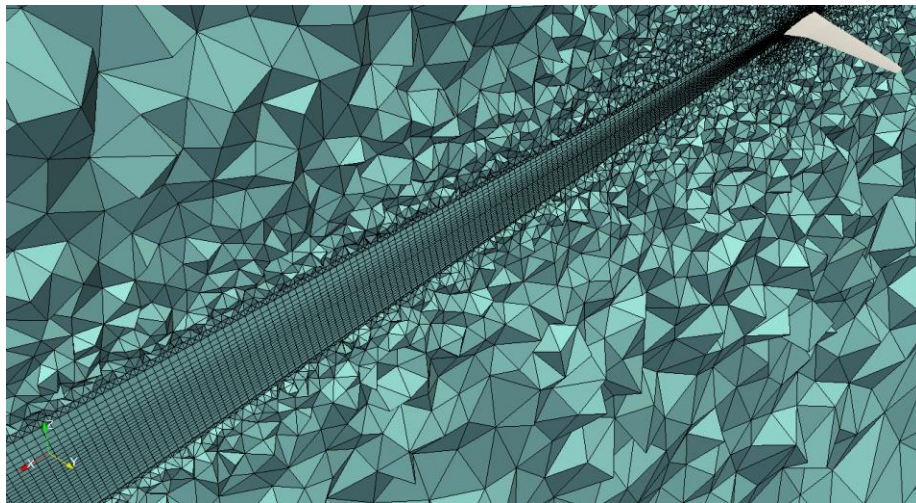


Figure 6.23: Slice of the hex-dominant mesh block embedded in the hybrid volume mesh for the DLR-F4 case

Figure 6.24 shows slices of the trailing shear layers at various downstream locations. The structured mesh block has allowed the formation of the wing tip vortex to be captured, as well as preservation of the wake downstream of the trailing edge. In a similar manner to the M6 case, the feature-aligned mesh is again allowing both evolution and preservation of the entire shear layer, in particular the vortex core, which increases in intensity with distance downstream. This is also indicated by the spanwise slice displayed in figure 6.25. The eddy viscosity contours at a slice location approximately fifteen chord lengths downstream of the trailing edge is presented in figure 6.26, further indicating the level of shear layer preservation achievable by using the feature-aligned mesh. Such resolution of the flow features at this distance downstream of the trailing edge using a standard hybrid mesh would not be possible.

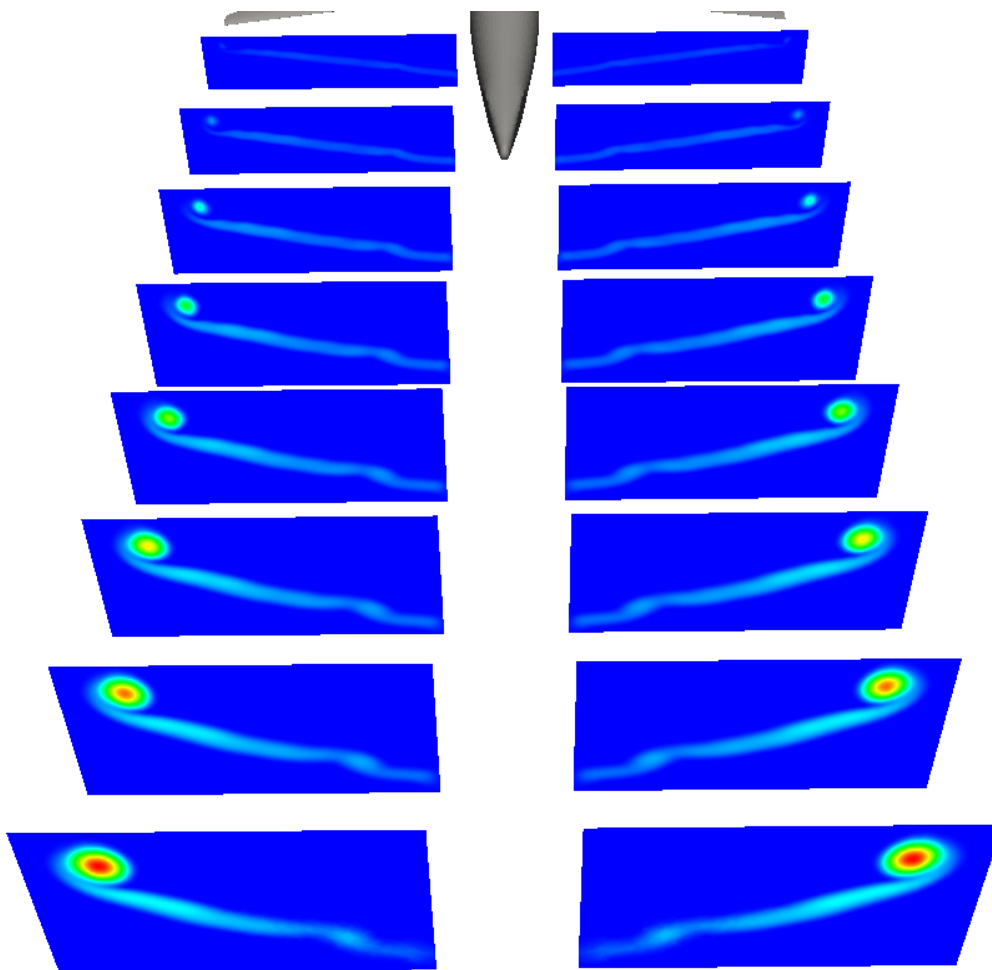


Figure 6.24: Preservation of the wake and wing tip vortex downstream using the feature-aligned mesh –eddy viscosity contours

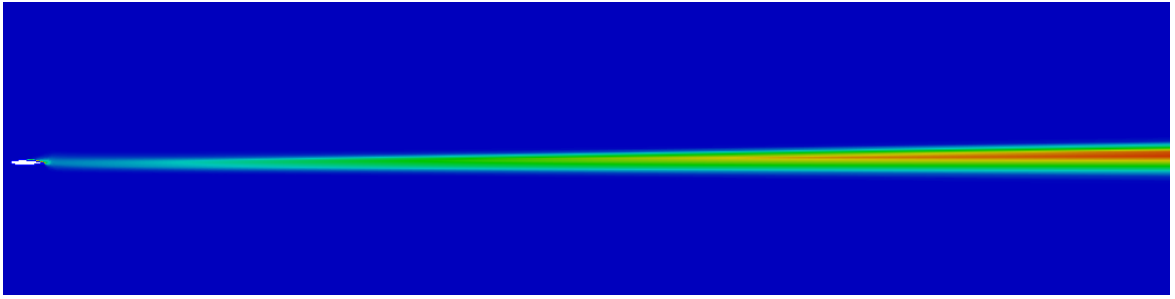


Figure 6.25: Spanwise slice of eddy viscosity contours indicating the preservation of the wake using the feature-aligned mesh

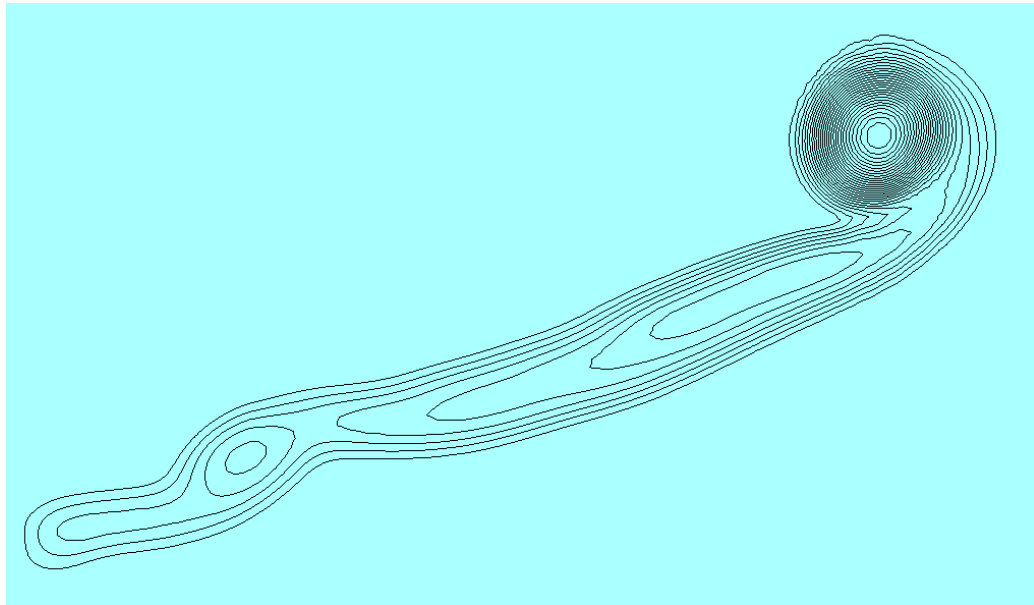


Figure 6.26: Eddy viscosity contour at approximately fifteen chord lengths downstream of trailing edge

6.7.3 Delta Wing, Transonic Viscous Flow, $M = 0.75$, $\alpha = 15.0^\circ$, $Re = 2 \times 10^5$

The concept of the delta wing has been exploited extensively in the design of supersonic aircraft. At moderate to high angles of attack the formation of leading edge vortices occur, which can provide a beneficial effect in terms of lift generation. The flow over a delta wing depends primarily on the geometry and the flow conditions. At low values of α the flow remains attached to the wing, however as α increases from moderate to high values, three-dimensional boundary layer separation takes place as the flow curls around the leading edge. This separation in the flow causes a large primary vortex to be

formed and trail downstream. A secondary vortex is also formed underneath the primary vortex, which rotates in the opposite sense, as displayed in figure 6.27.

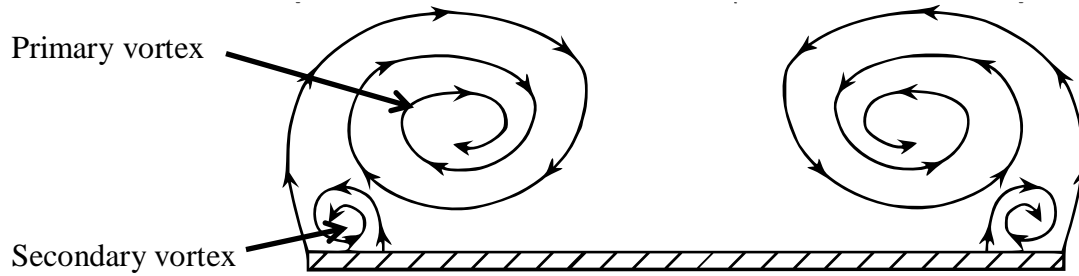


Figure 6.27: Schematic showing formation of primary and secondary vortices around delta wing leading edge (flow direction out of page)

For a certain range of angles, these leading edge vortices are steady, symmetric and regions of high energy, high-vorticity flow which induce regions of low static pressure. The low pressure regions form on the wing below the vortices which give rise to increases in lift even at high angles of attack which would cause stall for conventional wing planforms. However, as α is increased to very high values, the vortices continue to be shed, but the flow system is now accompanied by a distinct loss of stability, leading to asymmetry and unsteadiness. In some cases the vortex may ‘burst’ above the wing surface which can have undesirable effects on the wing performance due to the sudden increase in pressure.

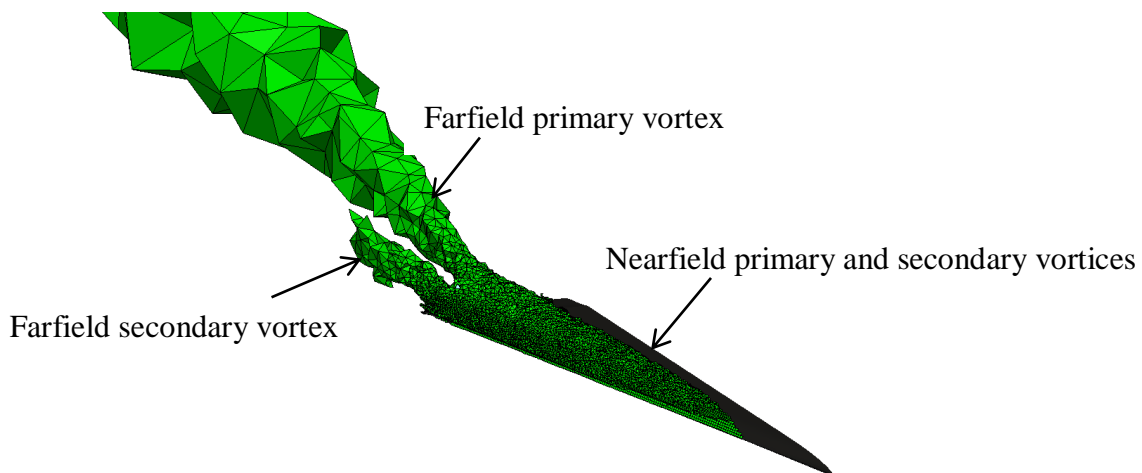


Figure 6.28: Vortex extraction and identification of vortex structures

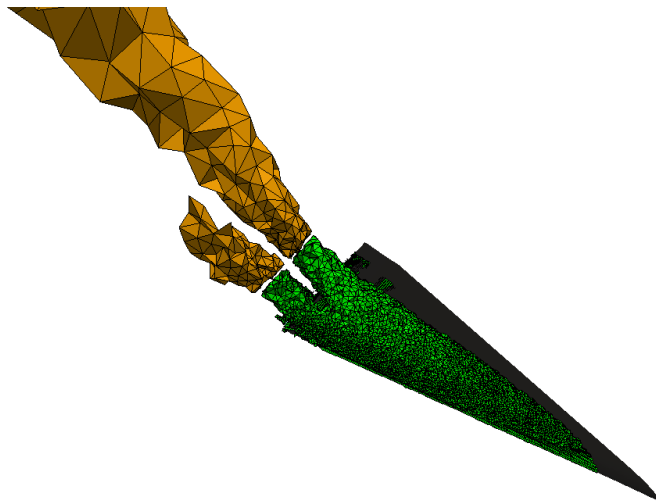


Figure 6.29: Grouping of the extracted vortex structure point clouds into nearfield (green) and farfield (orange) portions

Of interest in the present work is a case where the flow is stable and symmetric. The geometry is defined analytically^[192] where a delta wing with a slightly rounded edge is chosen. This means that the vortices will start to develop a small distance downstream of the apex of the leading edge. The extrusion process is applied to generate hex-dominant mesh blocks in the vortex region both above the nearfield mesh around the wing, as well downstream of the trailing edge. As observed in the other test cases, the standard hybrid mesh cannot resolve the vortices far downstream due to the increased numerical diffusion by the isotropic tetrahedra in the wake region. The poor resolution is shown in figure 6.30.

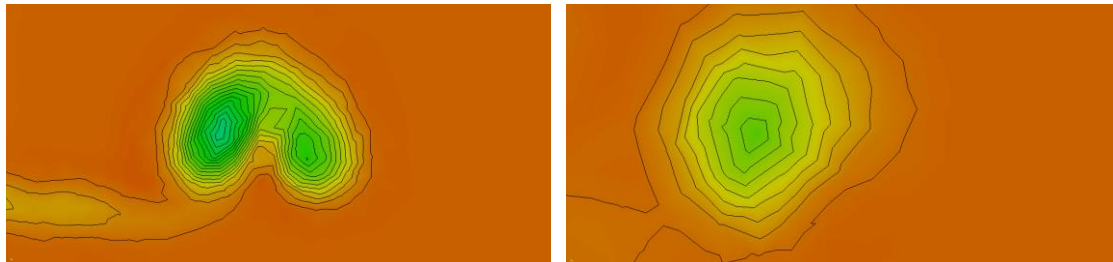


Figure 6.30: Total pressure contours for initial mesh at 120% and 160% chord

The delta wing is somewhat of a special case. At these flow conditions, there are multiple vortex structures to be considered in the extrusion process. The difficulty in applying the extrusion is that one of the vortex structures exists in close proximity to the

geometry itself; meaning that care is required to ensure the extrusion will not intersect the nearfield mesh.

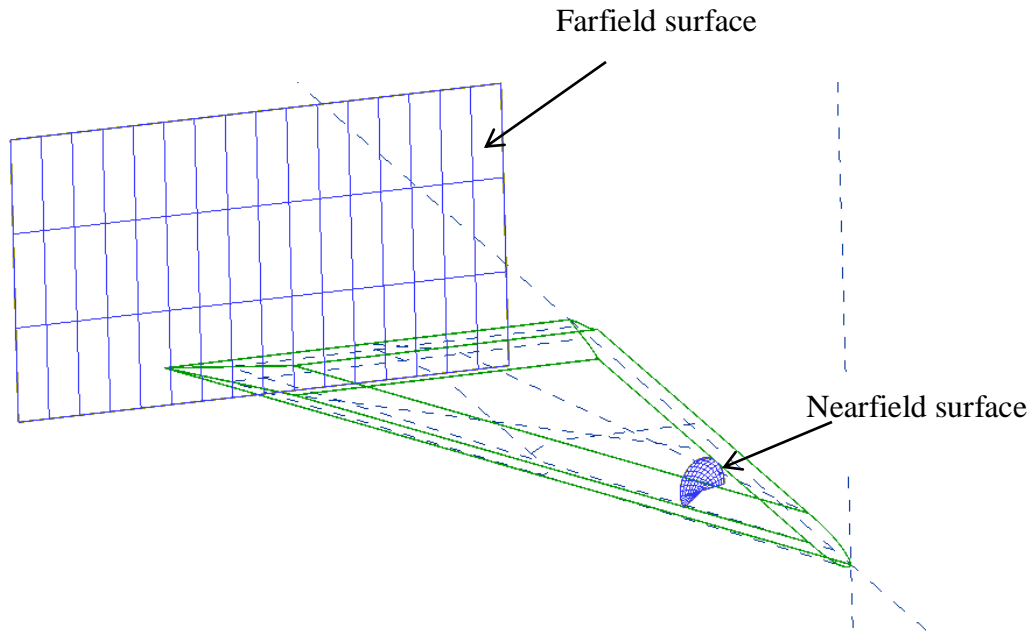


Figure 6.31: Source surfaces insertion for primary and secondary vortices along leading edge and the wake / trailing vortex downstream of trailing edge

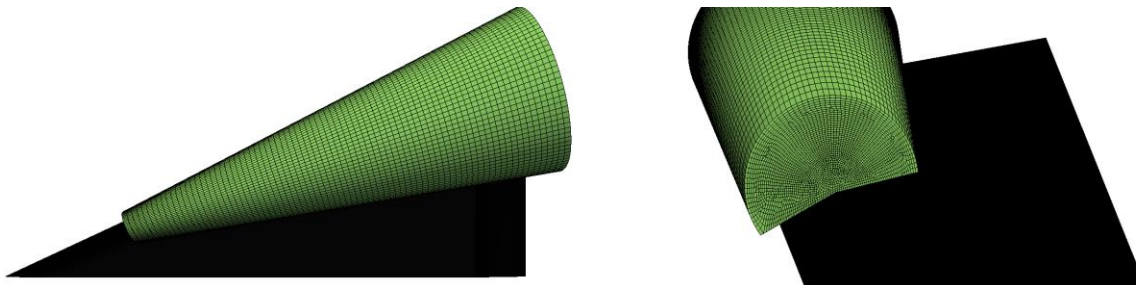


Figure 6.32: Extrusion of nearfield mesh block to capture primary and secondary vortices around leading edge of delta wing

Figure 6.28 shows the extracted vortex structures, obtained through application of the λ_2 criterion. Despite the poorly resolved vortex shown in figure 6.30, the λ_2 criterion has again worked well in capturing the vortex in the farfield mesh. One can observe the development of the primary vortex around the leading edge, which begins to develop around 20% chord due to the rounded leading edge of this particular delta wing geometry. The secondary vortex is also present, but for the majority of the leading edge is

indistinguishable and is extracted with the primary vortex point cloud. Beyond the trailing edge, the primary vortex suddenly changes direction and the secondary vortex can be seen to persist for a short distance. Since the current extrusion functionality may only occur in linear directions, the primary vortex structure must be considered in separate portions – one above the nearfield, which will cover the nearfield primary and secondary vortices and another which will capture the vortices in the farfield. The partition of the datasets is displayed in figure 6.29.

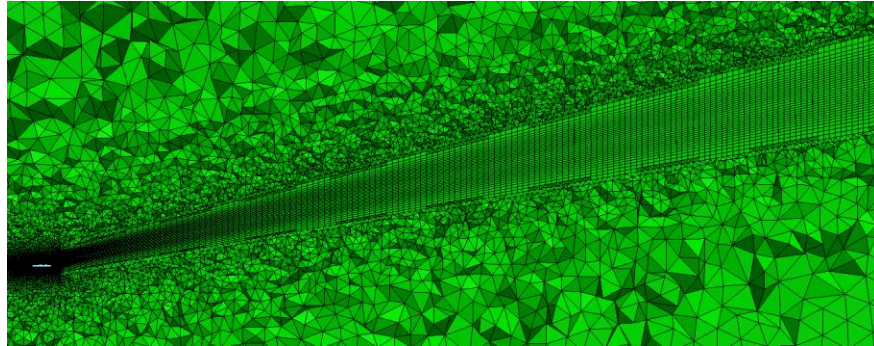


Figure 6.33: Slice of the farfield hex-dominant mesh block embedded in the hybrid volume mesh for the delta wing case

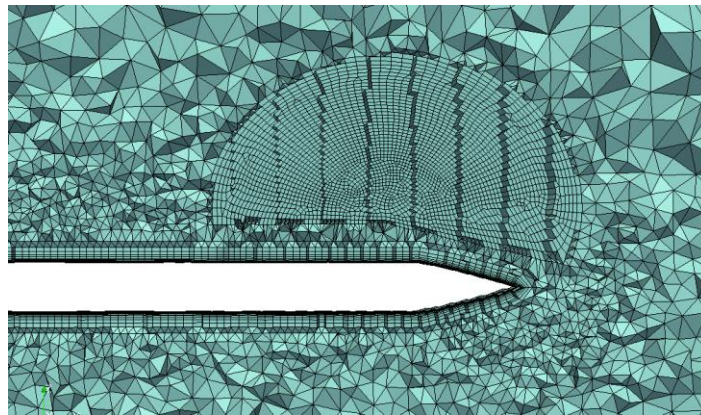


Figure 6.34: Slice of the nearfield hex-dominant mesh block embedded in the hybrid volume mesh for the delta wing case

Figure 6.31 shows the set up for the source surfaces. The source surfaces for the nearfield vortices are semi-circles and inserted a small distance above the expected height of the nearfield mesh. However, SOLAR implements a process to automatically pull back the advancing layers of the nearfield based on their proximity to other mesh or geometric entities. This automatically prevents collision of the layers with the mesh blocks, but may

in some cases affect nearfield mesh quality. The farfield mesh block is rectangular in shape such that the wake emanating from the trailing edge can also be captured. Figures 6.32, 6.33 and 6.34 show the resulting mesh blocks. The nearfield extrusion expands in the y and z directions to allow the vortices to be captured within the mesh block as they develop around the trailing edge.

Figure 6.35 compares slices of total pressure for the original mesh and the feature-aligned mesh to emphasise the improvement in the vortex capture on the feature-aligned mesh. In particular, the secondary vortex at 120% chord is clearly visible before it starts to rotate around the primary vortex. The feature-aligned mesh is still able to resolve the secondary vortex on the periphery of the primary vortex at 160% chord. Further downstream, whilst the secondary vortex has almost dissipated, the hex-dominant mesh block allows preservation of the primary vortex deep into the farfield.

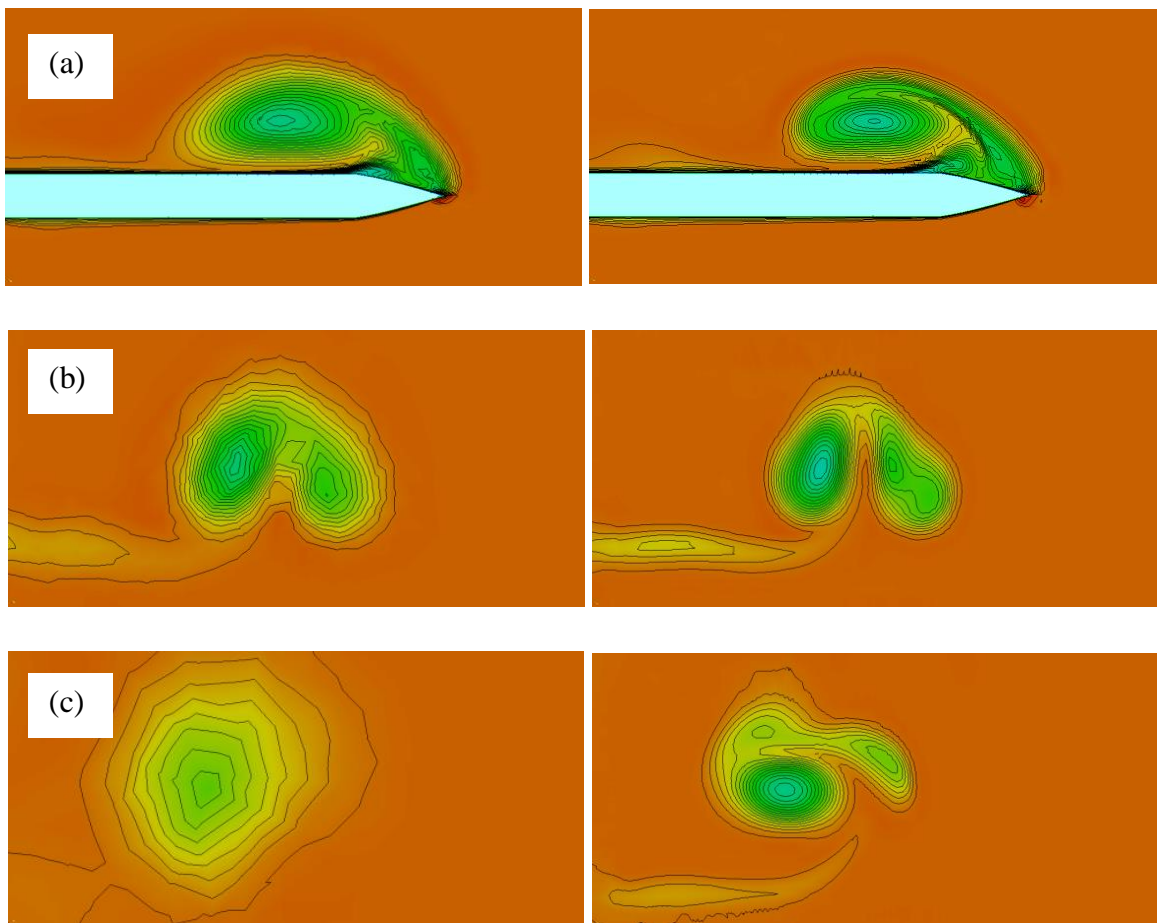


Figure 6.35: Comparison of total pressure contours for original mesh and feature-aligned mesh at various downstream locations (a) 80% chord, (b) 120% chord and (c) 160% chord

Figure 6.36 provides further indication of the vortex development and preservation in the form of streamlines coloured by eddy viscosity. The streamlines change colour with increasing distance downstream due to the formation of the vortex core. For the original mesh, the streamlines do not exhibit this behaviour, and do not follow any form of spiral pattern beyond the nearfield because the vortex core is never established. It can be observed that the vortex does start to develop around the delta wing on the original mesh due to high quality hexahedral elements in the nearfield, which resolve the separation of the near-wall viscous layers round the leading edge. However, once the mesh type changes to tetrahedra the flow feature formation ceases quite abruptly. This is again due to the dissipation of the vortex from the solution across the isotropic tetrahedra downstream of the trailing edge.

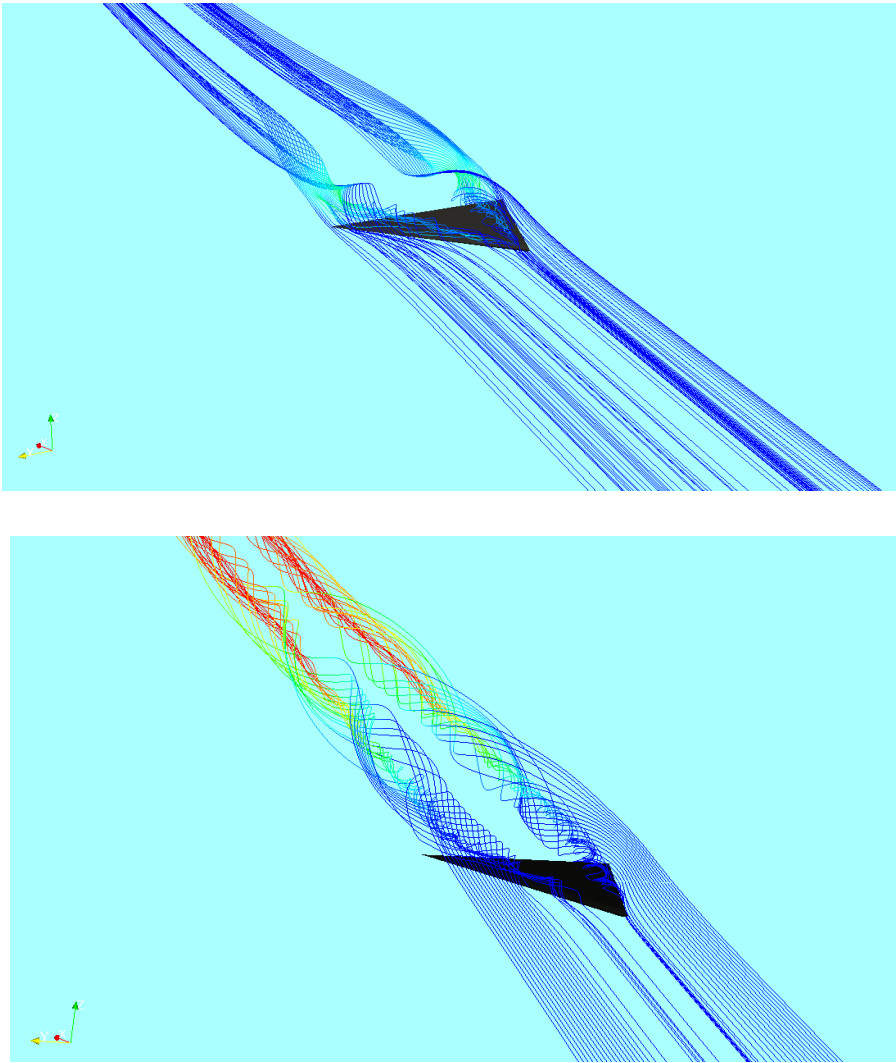


Figure 6.36: Streamlines coloured by eddy viscosity for (a) original mesh and (b) feature-aligned mesh

6.8 Conclusions

A method to embed hex-dominant mesh blocks aligned with wakes and wing tip vortices into hybrid meshes, through extrusion of quad-dominant surfaces meshes, has been presented. The process computes the extrusion direction from the extracted data set and allows for semi-structured hex-dominant mesh blocks to be embedded into the domain. The feature-aligned meshes allow for both evolution and preservation of the free shear layers for a great distance downstream of their origin. Resolution of the vortex core is particularly improved using the feature-aligned mesh. This is again due to the increased suitability of the semi-structured hexahedral mesh elements for the resolution of anisotropic flow physics. Standard hybrid meshes which possess isotropic regions of tetrahedra in the wake lead to high levels of numerical dissipation, particularly when the spacing of the elements increases in the farfield. However, the ability of the tetrahedra to resolve the wake and vortices can be improved through the use of some suitable adaptation scheme. The performance of anisotropically adapted tetrahedra in resolving these features is considered in detail in the next chapter.

CHAPTER VII

Feature-Aligned Hex-Dominant Mesh Block Insertion: Part III: A Comparison with Anisotropic Adaptation Techniques

7.1 Introduction

The current chapter revisits previously presented test cases for which the feature-alignment processes were applied for different flow features in turn. These techniques are combined so the hex-dominant mesh blocks for each flow feature exist in the same mesh. The performance of the feature-aligned meshes is then compared with hybrid meshes which have undergone anisotropic adaptation techniques as a means to improve the flow feature resolution, as briefly demonstrated in Chapter VI for the shock wave case. This comparison will allow the relative benefits of certain types of elements in the flow feature regions to be determined in both a qualitative and quantitative sense.

7.2 Test Cases

7.2.1 Delta Wing, Transonic Viscous Flow, $M = 0.75$, $\alpha = 15.0^\circ$, $Re = 2 \times 10^5$

After definition of an anisotropic metric using the Hessian of the Mach number, the adaptation schemes described in Chapter III can be applied on the tetrahedral part of the hybrid mesh. This allows the mesh to adapt to the shear layers to improve the capability of the original mesh to resolve them. The original mesh used for the purposes of adaptation consists of approximately 3.0 million elements. It was carefully constructed based on the initial mesh presented in the previous chapter, with addition of sources in the wake region only to increase the number of elements generated in this region. Note that including the tetrahedral optimisation algorithms which perform edge and face swapping, along with nodal movement, the mesh is also refined. The final adaptive mesh contains approximately 4.0 million elements.

Figure 7.1 shows a spanwise slice of the final adapted mesh where it can be observed that the mesh has adapted to the trailing wake of the delta wing in an anisotropic fashion. Figure 7.1 also shows two streamwise slices at 80% and 120% chord. The adaptive

tetrahedra become a means to observe the flow behaviour itself, with the tetrahedra following the shape of the both primary and secondary vortices..

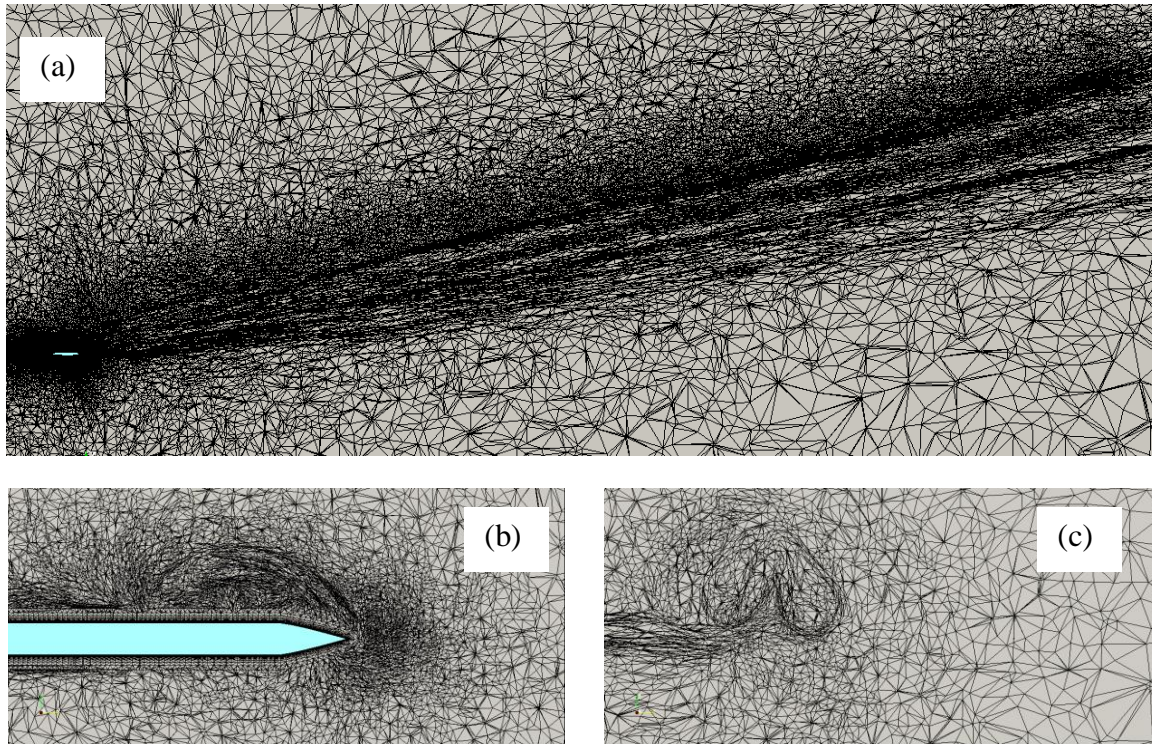


Figure 7.1: Anisotropic adaptive mesh (a) spanwise wake adaptation and downstream adaptation at (b) 80% chord and (c) 120% chord

The resolution of the flow features are indicated with total pressure contours. Figure 7.2 compares the adaptive and feature-aligned meshes at various streamwise locations. Whilst the adaptive mesh does significantly improve the vortex resolution compared with the original mesh, the feature-aligned mesh still provides a smoother representation of the total pressure contours, even at 160% chord. It is also interesting to observe the spanwise eddy viscosity contours in figure 7.3. The magnitude of the eddy viscosity for both approaches is included. Application of the adaptive schemes has allowed the adaptive mesh to resolve the vortex development which is indicated by the raise in eddy viscosity in the vortex core. However, the eddy viscosity magnitude for the feature-alignment approach is much higher which indicates that the vortex development, and hence the intensity of the resolved vortex, is significantly increased within the hex-dominant mesh block compared with the anisotropic tetrahedra. This has occurred despite the greater number of elements present in the adaptive mesh. Whilst the tetrahedra are aligned with the flow, there is likely to still be a level of numerical diffusion present which is preventing the same level of

intensity within the vortex being predicted. This is likely due to some faces of the tetrahedra still not being aligned with the features and thus degrading the potential for high quality resolution. Note that again, it is difficult to determine from the present study which of these solutions are ‘correct’ without other data to validate or verify the results.

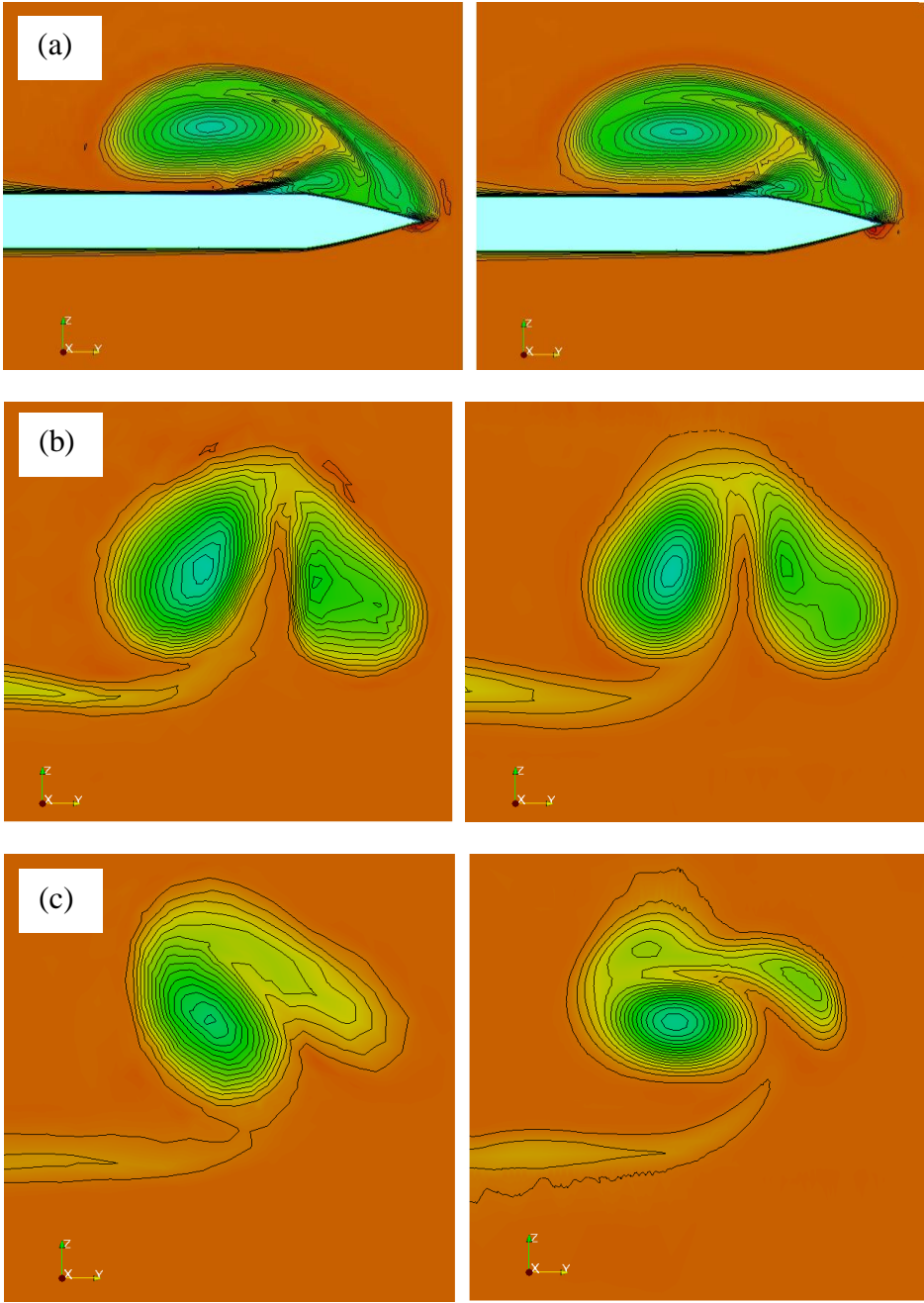


Figure 7.2: Comparison of total pressure contours for adaptive mesh and feature-aligned mesh at various downstream locations (a) 80% chord, (b) 120% chord and (c) 160% chord

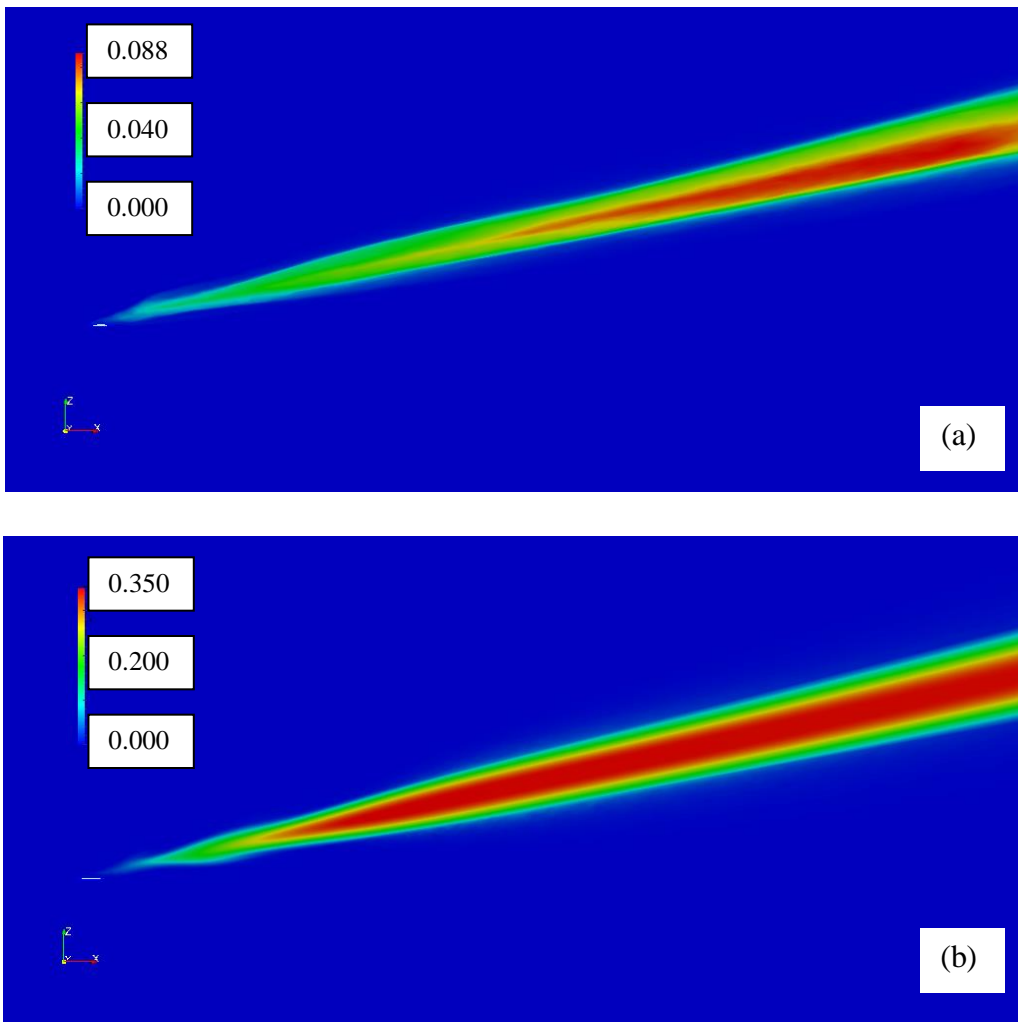


Figure 7.3: Comparison of spanwise eddy viscosity contours (a) anisotropic adaptive mesh and (b) feature-aligned mesh

7.2.2 M6 Viscous Transonic Flow, $M = 0.8395$, $Re = 11.72 \times 10^6$, $\alpha = 3.06^\circ$ [183]

The feature-alignment process is applied for the wake / wing tip vortex region through extrusion of a rectangular surface on the original hybrid mesh, which consisted of approximately 1.86 million elements. The feature-aligned mesh consists of 3.4 million elements. For comparison purposes, a standard hybrid mesh is generated with application of sources throughout the wake region to get a similar number of elements. This mesh, referred to as the isotropically refined mesh, will also undergo the anisotropic adaptation (edge/face swapping and nodal movement only) to assess the effect of anisotropic tetrahedra on the wake resolution. These meshes contain approximately 3.4 million elements. For the feature-aligned mesh, an anisotropic nodal movement algorithm, described in section 3.7.2 is applied within the hexahedral mesh block to deform the mesh

to the flow features. The feature-aligned mesh is not adapted in the region outside of the mesh block. It is necessary to point out that the resolution of the surface mesh and boundary layers around the wing are the same for each case. The difference between the three meshes will therefore be the type of element appearing in the wake region – isotropic tetrahedra (original), anisotropic tetrahedra (adaptive) and finally anisotropic hexahedra (feature-aligned).

The anisotropic adaptation based on the Hessian of the Mach number is applied once again to adapt the isotropically refined mesh to the flow features. Figures 7.4 and 7.5 display the effect of the adaptation on the mesh in the shear layer regions and the generation of highly anisotropic tetrahedra. The mesh has clustered in the trailing wake region and is also adapting to the rolling-over of the free shear layers to form the wing tip vortex.

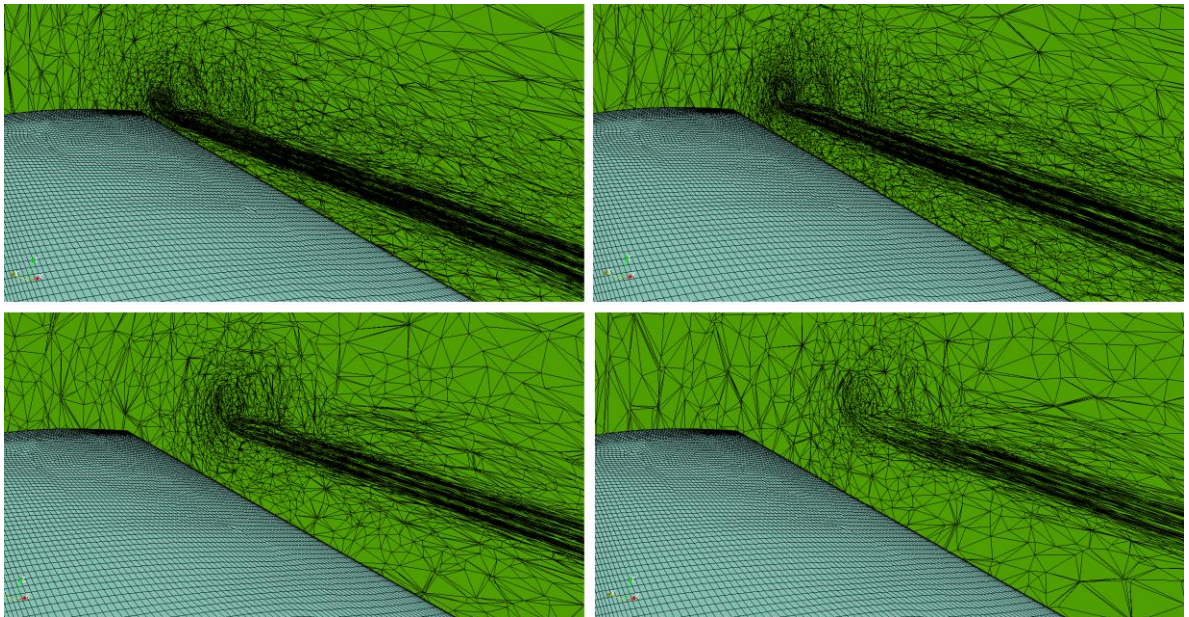


Figure 7.4: Various streamwise slices of anisotropic adapted mesh near the trailing edge

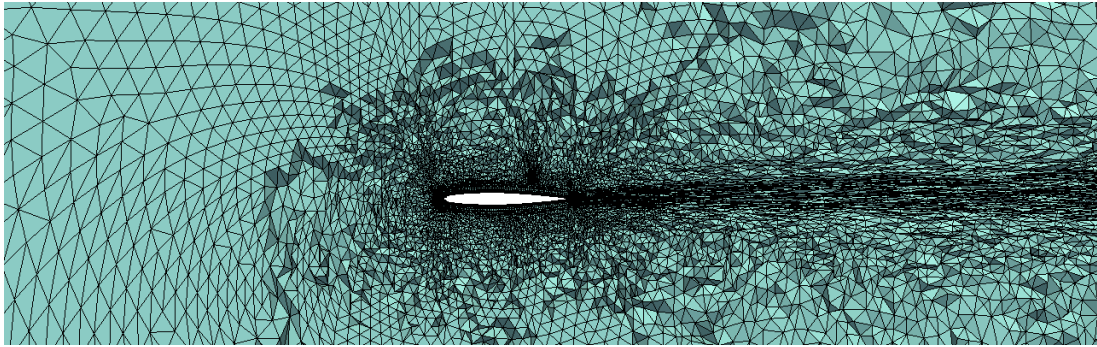


Figure 7.5: Spanwise slice of anisotropic adapted mesh

Figure 7.6 shows various slices of the adaptive feature-aligned mesh and the anisotropic adaptive mesh. The convergence of the adaptive scheme for the feature-aligned mesh is fairly rapid, since the mesh points are already approximately aligned with the flow, leading to a reduced amount of work necessary to adapt the mesh (convergence in this case refers to the positions of the nodes no longer changing due to the scheme having already identified the optimum position for that particular adaptation cycle). Furthermore, the nodal movement scheme is applied locally within the hex-dominant mesh block only, not across the entire mesh. As previously discussed, it is essential that during local adaptation of the mesh block that the buffer region of pyramids which provide the conformal interface between the hexahedral cells and the tetrahedra are not modified, as distortion of these elements can be detrimental to the overall mesh validity. For the anisotropic adaptive mesh, a greater level of work is required on the mesh since none of the initial tetrahedra are aligned with the features, and they also require use of a wider range of adaptation schemes (edge/face swapping, edge collapsing as well as nodal movement) to provide alignment, which can be time consuming to perform on a large mesh to convergence.

Figure 7.7 shows the anisotropic adaptive mesh has improved the resolution of the trailing shear layers and vortex in terms of eddy viscosity contours compared with the original hybrid mesh. This is due to the formation of anisotropic tetrahedra in the free shear layer regions. However, even though a reduction in numerical diffusion is present across these anisotropic tetrahedra, the mesh still does not predict as high eddy viscosity values as the adaptive feature-aligned mesh. In particular, the vortex core is very strongly resolved within the hexahedral mesh block.

Mesh	Number of Elements ($\times 10^6$)	C_D	C_L
Original	1.86	0.01345	0.2033
Isotropically refined	3.4	0.01344	0.2033
Feature-aligned	3.4	0.01360	0.2044
Adaptive	3.4	0.01342	0.2039

Table 7.1: Comparison of lift and drag coefficients for the ONERA M6 case

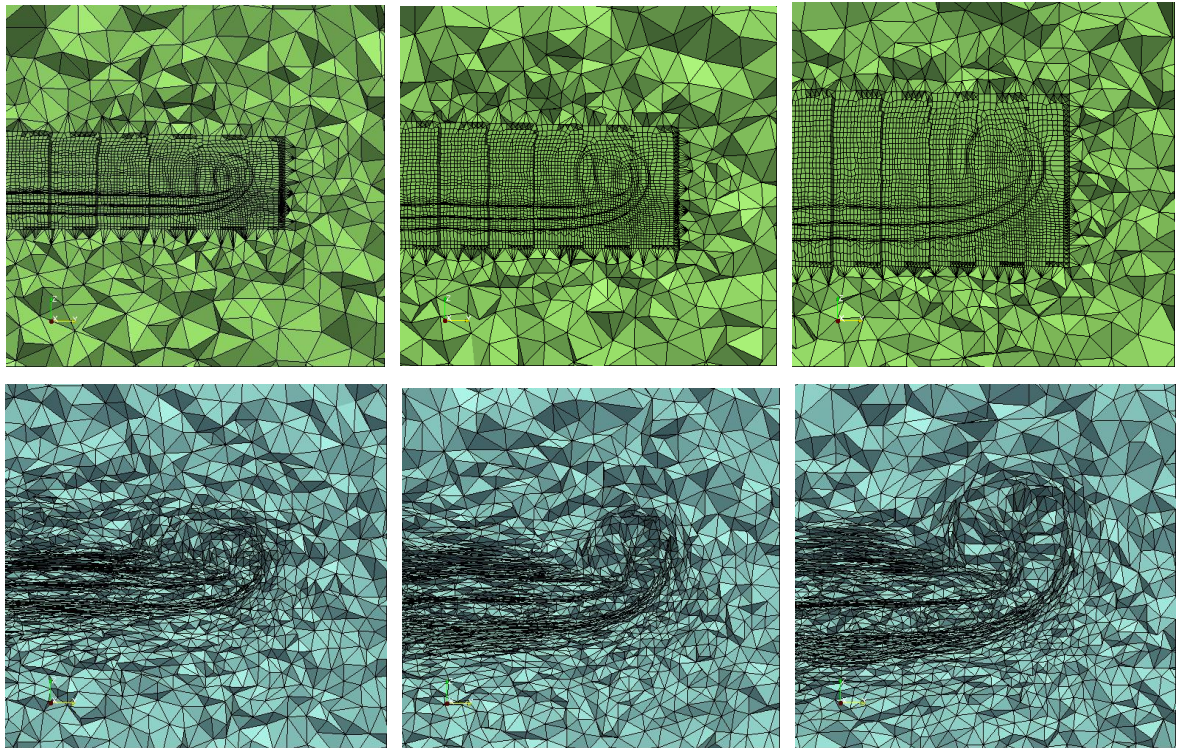


Figure 7.6: Comparison of mesh at multiple downstream wake locations for the feature-aligned adaptive mesh and anisotropic adaptive mesh at two, four and eight chord lengths downstream

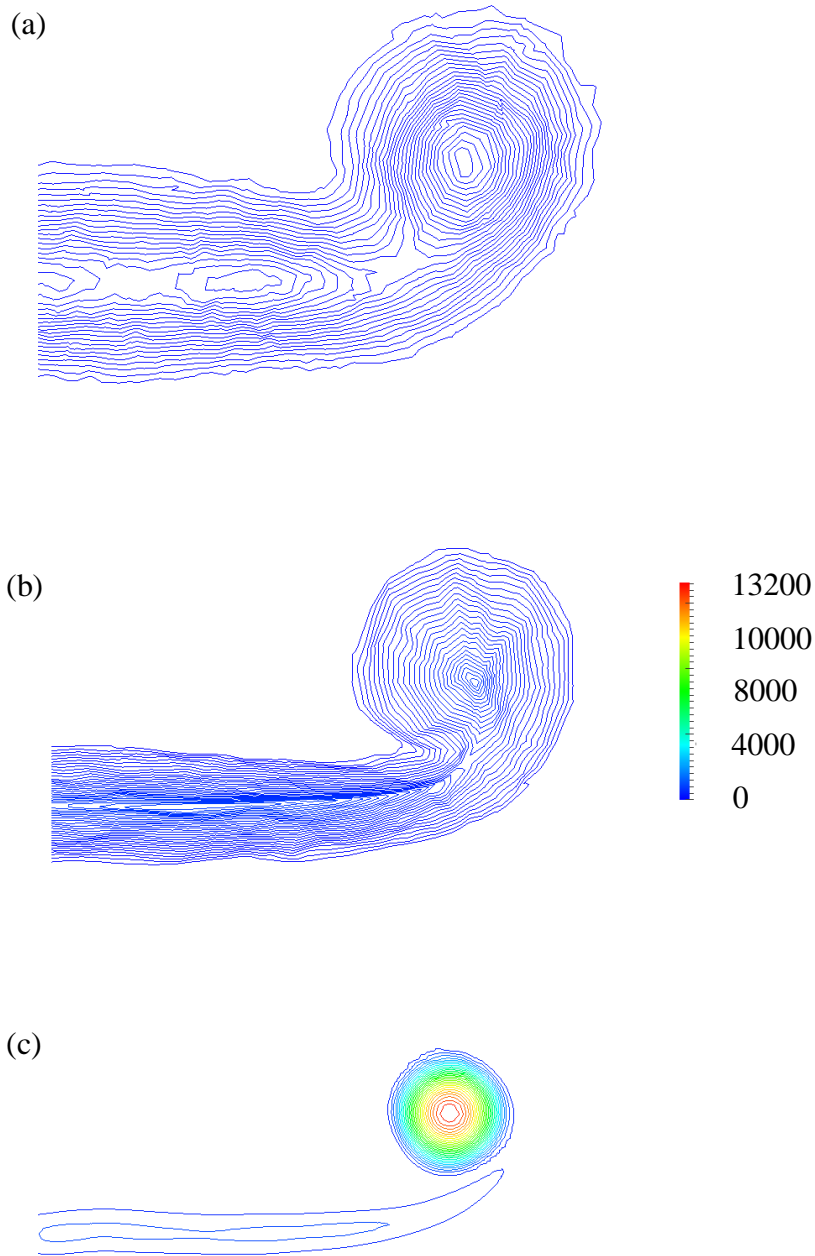


Figure 7.7: Eddy viscosity ratio contours ten chord lengths downstream for (a) original mesh, (b) adaptive mesh and (c) adaptive feature-aligned mesh

Table 7.1 displays the lift and drag coefficients for each of the meshes. Whilst the lift and drag show marginal differences between original, isotropically refined and adaptive meshes, the feature-aligned mesh predicts slightly different results. The feature-aligned

mesh drag values are also inconsistent with the shock-aligned mesh presented in Chapter V, and the reasons for this are unclear. Since the ‘correct’ solution is unknown, only an estimation of a grid independent solution was made, it is difficult to draw conclusions on the values presented in this section. It may be possible that the refinement study presented in Chapter V would continue to display variations in the drag coefficient if the mesh was refined further. For three-dimensional problems, grid refinement studies can quickly exhaust the available computational resources, which meant a limit had to be placed on the mesh size for the studies in this thesis.

7.2.3 DLR-F4, Transonic Viscous Flow: $M = 0.75$, $Re = 3.0 \times 10^6$, $\alpha = 2.0^\circ$

The final case considers the feature-aligned mesh for the wake and wing tip vortex region only. For comparison purposes, the meshes have again been carefully constructed such that a similar number of elements exist in the wake region, as with the M6 case presented previously. The total number of elements for each mesh is approximately 3.5 million.

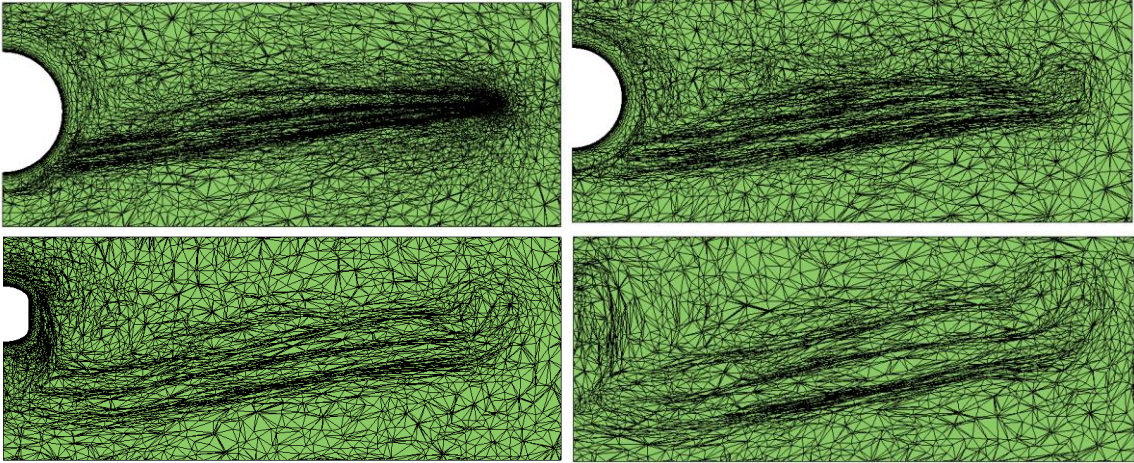


Figure 7.8: Various downstream slices of the anisotropic adaptive mesh

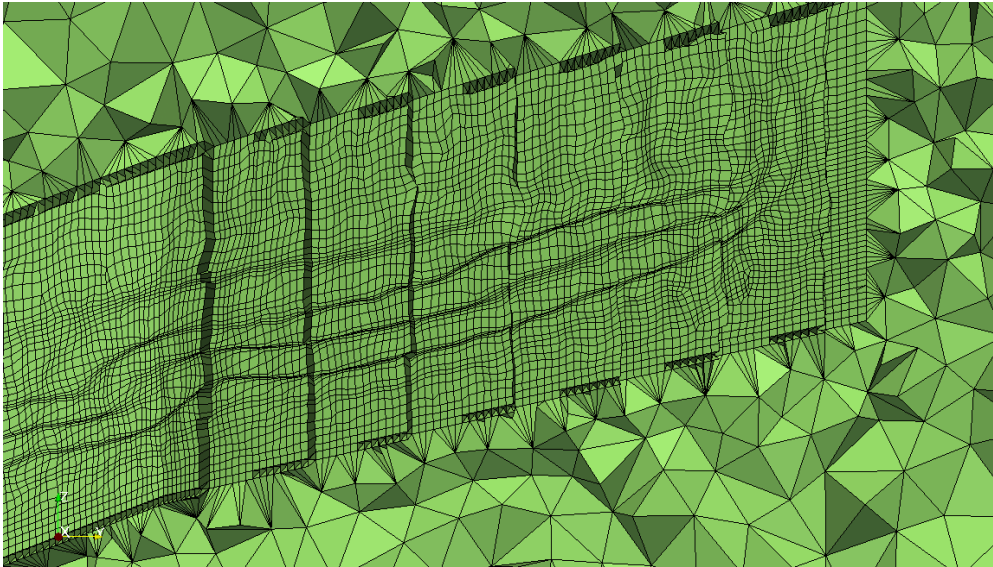


Figure 7.9: Slice of the adaptive feature-aligned mesh

Figure 7.8 displays a slice of the anisotropic adaptive mesh at various downstream locations. As expected, the mesh adapts to the shear layers and vortex, including the shear layers trailing from the fuselage. Figure 7.9 shows a slice of the adaptive feature-aligned mesh. As with the previous case, the mesh block deforms to the shear layers and the vortex downstream of the wing (the mesh downstream of the fuselage remains unchanged since the nodal movement is applied within the hexahedral mesh blocks only – the tetrahedral modification techniques are not applied on the feature-aligned meshes). The convergence of the adaptive scheme within the mesh block is once again fairly rapid, as with the M6 case, since the mesh cells are already approximately aligned with the flow feature and only several cycles are necessary. On the other hand, convergence of the anisotropic adaptation algorithms on the tetrahedra for the adaptive mesh is a time consuming process, and a large number of adaptive cycles are required to yield the final adaptive mesh.

The magnitude of the eddy viscosity ratio contours for each mesh at fifteen chord lengths downstream is shown in figure 7.10. Whilst the adaptive mesh does improve the resolution of the shear layers and increases the eddy viscosity ratio values in this region, the resolution of the vortex core is only marginally improved. This could be an issue with the present error estimator's ability to identify the vortex region. Other error estimators, or even different flow variables used to construct the present error estimator, may lead to different effects. Generally, the adaptive feature-aligned mesh displays significantly greater values of eddy viscosity, primarily due to the much improved resolution of the vortex core.

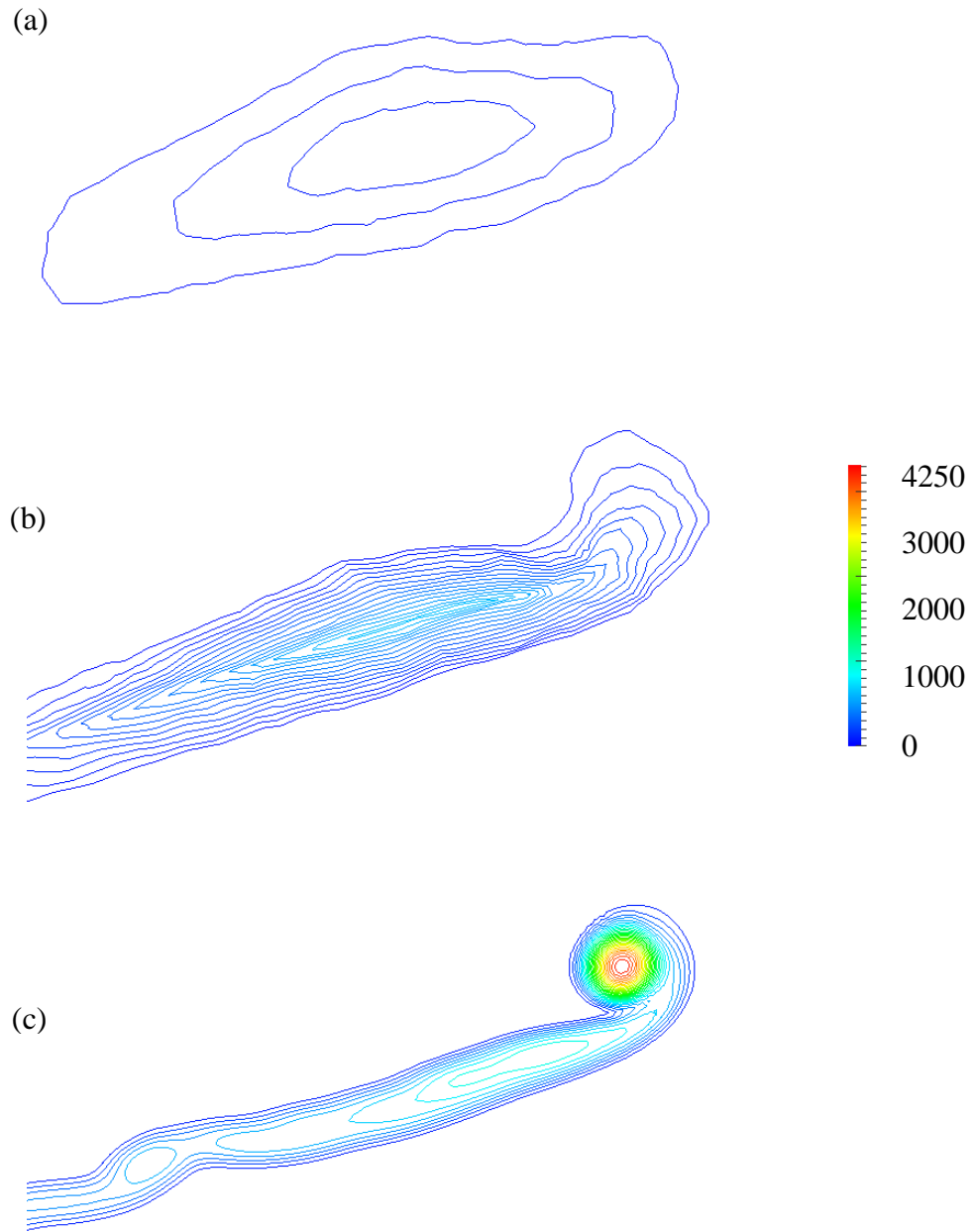


Figure 7.10: Eddy viscosity ratio contours fifteen chord lengths downstream for (a) original mesh, (b) adaptive mesh and (c) adaptive feature-aligned mesh

Mesh	Number of Elements ($\times 10^6$)	C_D	C_L
Original	2.40	0.02865	0.4171
Isotropically refined	3.50	0.02927	0.4206
Feature-aligned	3.50	0.03519	0.4538
Adaptive	3.50	0.02869	0.4177

Table 7.2: Comparison of lift and drag coefficients for the DLR-F4 case

Table 7.2 compares the lift and drag coefficients for the range of meshes considered for the study. As with the M6 case, it can be observed that the lift and drag are in close agreement for all the meshes apart from the feature-aligned mesh. The coefficients obtained from the feature-aligned solution are substantially different to the other meshes. Again, the surface and nearfield meshes are the same for each mesh considered. It is difficult to determine which of the results presented in Table 7.2 is correct without comparison to experimental data or grid independence studies. The results appear to indicate that whilst the presence feature-aligned mesh block can substantially improve the resolution of the flow features in a qualitative sense, they also have an effect on the final lift and drag coefficients. The reasons for this remain unclear.

The general trend that can be observed across all the cases presented in this chapter is that the isotropically refined mesh generally provides very poor resolution of the wake and wing tip vortex. The adaptive mesh improves the resolution somewhat, due to alignment of the tetrahedra with the feature. It provides good resolution of the shear layers within a few chord lengths downstream of their formation. However, there still appears to be significant dissipation across the mesh that is preventing resolution of the vortex core at further distances downstream. One possible reason for this is that whilst the tetrahedra are aligned, the size of the elements tend to increase further downstream, as indicated by figure 7.8. For the feature-aligned mesh, the hexahedral mesh elements are constantly spaced, since the embedded surface mesh is grown with a constant layer size. This has led to very high eddy viscosity values predicted within the wake region. The studies suggest therefore that element type and orientation, as well as element spacing will have a great effect on the resolution of these features and their preservation in the CFD solution.

7.3 A General Comment on the Reliability of Results

The author feels it is necessary to point out that whilst the feature-aligned hexahedral mesh blocks have significantly improved the resolution of the flow features, which has been indicated by preservation of solution quantities like eddy viscosity for vast distances downstream, the solution given by these particular meshes is not necessarily 'correct'. Whilst the meshes used for the results in this chapter were carefully constructed so that similar numbers of elements in the flow feature regions were compared, the main conclusion that can be drawn is that the presence of the hexahedral mesh blocks improves the flow feature resolution, compared with the isotropic or anisotropic tetrahedra. However, without extensive validation with real experimental data, or further verification through grid independence studies or studies on other types of mesh (for example, comparing with a structured multi-block mesh), the results presented in this chapter should not currently be assumed to be reliable by the reader. This could be one aspect of future work to further demonstrate that the present feature-alignment methods are worthwhile additions to hybrid mesh generation technology.

7.4 A Discussion on Feature-Alignment for Wakes and Vortices

In an industrial context, it is likely that the designs under consideration are more complex than those presented in this thesis. For example, a wing-fuselage configuration could be modified to include flap-track-fairings, pylons, nacelles, winglets and so on. The source surface used for the extrusion need not change. The hex-dominant mesh block will be present to capture the complex wake flow structures that manifest themselves. Whilst in the current implementation there is a narrow gap of isotropic tetrahedra between the geometry trailing edge and the hex-dominant mesh block, this appears to have minimal effect on capturing the development of the flow structures downstream. However, generation of wake blocks attached to the nearfield mesh would be desirable in the future since it would negate the requirement for a source surface to be embedded in the domain. In terms of current capability in SOLAR such a method would require extensive modification of the nearfield growth functionality. Tests were made during the period of this thesis in an attempt to connect these wake blocks to the nearfield, in a similar vain to the method

presented for embedding shock-aligned mesh blocks, but the complexity of nearfield growth at trailing edge junctions caused the attempts to fail.

SOLAR generally provides good resolution of boundary layers and flow structures in the nearfield region due to the high-quality hexahedral and prismatic elements present. However, resolving the viscous flow phenomena in the wake of the geometry sufficiently, where the element type suddenly changes to isotropic tetrahedra is not possible. The most straight-forward way to improve the resolution of the wake in a standard SOLAR mesh is to add sources in the wake region as a means to increase the density of the mesh in an isotropic fashion. However, even local sources can drastically increase the number of elements present in the entire mesh, thus placing strains on computational resources with no guarantee of worthwhile improvements in the feature resolution.

The placement of the mesh blocks at suitable locations improves the capability of the mesh to capture the wake and vortex structures in an efficient manner. These flow features are resolved far downstream due to the reduction in numerical diffusion that one would usually observe in a tetrahedral farfield. The embedded hex-dominant mesh blocks are semi-structured and may provide similar resolution as an actual structured mesh – although a direct comparison has yet to be made. However, the present method inherits several distinct advantages of unstructured meshes which are worth reiterating. Firstly, unstructured mesh generation techniques are flexible and automatic, allowing for generation of meshes for complex geometries. Secondly, it exploits the fact that hybrid meshes can be comprised of a mixture of elements.

The other benefit of the inserted mesh blocks is the ability of the mesh to capture the flow physics without applying many cycles of adaptation (as demonstrated in Chapter VI) which can be expensive for large meshes and possibly unattractive in an industrial context (although the suitability of the hex-dominant mesh blocks to undergo a nodal movement scheme has been presented in this thesis). Many adaptation schemes (such as the TAU adaptation module) are unable to be run in parallel, meaning that for parallel jobs, application of the adaptation scheme requires substantial extra work during the solution. For these problems, the original mesh is partitioned to allow the solution to be run in parallel. When the adaptation is required to take place, the solution and mesh files must be ‘gathered’ together so that the adaptation metrics can be constructed and applied on the

unified mesh. The original solution file must then be interpolated onto the new mesh, followed by a re-partition to allow the solution to continue in parallel. Whilst the extrusion method does require regeneration of the entire mesh after determination of an extrusion direction, for low angles of attack where the orientation of the wake will be essentially tangential to the downstream direction, the extrusion direction can easily be estimated prior to the solution. This means that the mesh would not need to be regenerated, and requires partitioning only once prior to the start of the simulation.

Finally, it is worth pointing out that whilst the present flow solutions have been obtained using a RANS solver, these meshes are not restricted to such models. It would be interesting to observe the potential improvement in the performance of DES or LES solvers to resolve the flow structures in wakes downstream of complex configurations.

7.5 Conclusions

The present chapter compared two methods for improving the capability of hybrid meshes to resolve anisotropic flow physics. Both techniques significantly improved the mesh performance compared with that of standard or isotropically refined hybrid meshes. In particular, the adaptive schemes have allowed the mesh to become aligned with features in the solution such as the roll-over of the shear layers during the formation of the wing tip vortex. Whilst improved resolution of the flow features has been obtained using the adaptive mesh, the results have shown that the resolution is not at the same level as the adaptive feature-alignment approach. This has been indicated by considering eddy viscosity magnitude and eddy viscosity ratios in the wake regions. The adaptive feature-aligned mesh appears to capture the evolution of the trailing shear layers and vortex, and also allows for the preservation of these flow structures at a great distance downstream. This is due to a smaller amount of numerical diffusion across the semi-structured hex-dominant mesh block compared with the anisotropic tetrahedra.

8 Conclusions

8.1 Achievements and Concluding Remarks

The main target of this project was to develop mesh generation and adaptation methods which allow for the use of high-quality quadrilateral or hexahedral elements aligned with flow features in the CFD solution. Such elements can be aligned with highly anisotropic flow physics more effectively than their triangular and tetrahedral counterparts. As hex-dominant meshing software was not available for the project, a decision was made to exploit one of the primary advantages of hybrid meshes: their ability to include a mixture of different elements. This ultimately allowed for the development of methods to embed blocks of hex-dominant mesh into the domain, aligned with the flow features, in an attempt to improve the capability of hybrid meshes to resolve the flow physics.

For the generation of high-quality unstructured quad-dominant surface meshes, a novel method involving the medial axis as a means to construct a geometric representation of complex flow structures has been developed. The technique allows for a smooth estimation of the flow feature topology as a series of curves, which are treated as virtual geometries and after embedding into the geometric domain, can influence the formation of the two-dimensional or three-dimensional surface mesh. Due to the flexibility of the approach, a range of complex features can be considered with a minimum of user interaction. After mesh regeneration, the quadrilateral elements are aligned in an anisotropic fashion with the feature curves. The locally aligned mesh actually mimics the attributes one would observe in a typical structured mesh. Due to this alignment, the mesh readily lends itself to a simple anisotropic nodal redistribution technique to further optimise the mesh based on the solution, allowing the cell interfaces to be fully aligned with the feature. This represents the ideal scenario for the application of the approximate Riemann solvers in the finite volume formulation. The approach has been applied on a wide variety of test cases in two-dimensions. The resulting high-quality regions of mesh have been shown to significantly improve the resolution of the flow features and solution accuracy (relative to a globally refined mesh) compared with the standard unstructured quad-dominant mesh. Performance of the approach has been compared to that of an anisotropic adaptive triangular mesh. The results have indicated that improved solution accuracy and

efficiency can be obtained through the use of the developed method applied on the unstructured quadrilateral mesh.

The medial axis approach also lent itself for application to creating high-quality quad-dominant surface meshes aligned with shock footprints in three-dimensional transonic problems. This allowed for the development of a process which became the next major achievement in the present project. A new method which allows for embedding of semi-structured hex-dominant mesh blocks, by extruding regions of surface mesh aligned with transonic shock wave regions, has been developed. The medial curves allow for decomposition of the extracted shock wave which in turn allow for growth heights and directions to be computed for each portion of the shock wave. After identification of a suitable set of surface mesh faces, the hex-dominant mesh block grows from the surface beyond the nearfield mesh aligned with each portion of the shock wave. The resulting mesh block is then embedded into the hybrid volume mesh and drastically improves the capability of the mesh to resolve the transonic shock across the entire span of the wing. Comparisons of the mesh performance with isotropically refined hybrid meshes and other adaptive techniques were made. The shock-aligned mesh performance has appeared to be much improved compared with an adaptive mesh and isotropically refined meshes, in terms of feature-resolution and accuracy for drag prediction.

The final main achievement in the project was to apply a similar surface mesh extrusion idea to improve hybrid mesh ability to capture wing tip vortices and wakes. A source surface is defined at the desired location by the user. This surface is treated like any other surface in the domain, and therefore a high-quality semi-structured quadrilateral mesh can be generated. After extraction of the relevant data from the solution, an extrusion direction and expansion factor is computed for the surface to allow for growth of the quadrilateral surface mesh through the detected flow feature. The resulting semi-structured hex-dominant mesh block can then be embedded into the hybrid volume mesh. For the presented test cases, the improvement of the resolution of trailing shear layers and in particular the wing tip vortex, compared with standard hybrid meshes, was substantial. This is due to the structured-like nature of the mesh that now exists downstream of the trailing edge of the geometry. Preservation of the trailing free shear layers further into the farfield was possible for each of the presented test cases. A comparison with anisotropic adaptation

schemes was carried out, to compare the semi-structured hexahedral cells with highly stretched tetrahedra. Whilst the anisotropic tetrahedra also significantly improved the hybrid mesh capability to resolve the features, they did not provide the same level of resolution as the hexahedra present in the feature-aligned meshes. The results suggest that element type, orientation and spacing influence the quality of shear layer and vortex resolution.

Generally, whilst the presence of the feature-aligned mesh blocks appear to have had favourable impacts on the numerical solution, these improvements have been observed relative to an isotropically refined mesh, under the assumption that as mesh size decreases, the set of partial differential equations that are being solved will converge to an ‘exact’ value. It is worth reiterating that which of these results are ‘correct’, particularly those presented in Chapter VII, is difficult to determine without further verification and / or validation.

8.2 General Concluding Remarks

- Isotropic elements of any type are generally unsuitable for resolving anisotropic flow features.
- Quadrilateral meshes can be more effectively aligned with flow features than anisotropic triangular elements, avoiding highly skewed elements.
- Isotropic tetrahedra generally provide poor resolution of anisotropic flow features, but application of anisotropic adaptation can improve the mesh performance.
- Feature-aligned hexahedral mesh blocks can substantially improve flow feature resolution, particularly wakes and wing tip vortices
- Whilst the presence of the mesh blocks appear to have a favourable impact on the numerical solution, further verification and / or validation of the results is required.

8.3 Suggestions for Future Work

In terms of the medial axis approach presented in Chapter IV, there is potential for extending the process into three-dimensions, but such an approach presents a significant challenge. Three-dimensional α -shapes can be computed for arbitrary point clouds and constrained tetrahedralization of this data set should be possible. The centres of the

circumspheres will then approximate the medial object in three-dimensions. The main difficulty however is that, as with the two-dimensional case, the boundary of the α -shape will be non-smooth and irregular, leading to a non-smooth medial object. Processing the medial object becomes distinctly less trivial in three-dimensions since the medial object will possess surfaces which intersect at curves, rather than curves intersecting at points.

Currently the process requires regeneration of the entire mesh. A more suitable approach would involve only local regeneration of the mesh. Using the medial curves to form surfaces which allow intersecting tetrahedra to then be identified to form a void region is one potential idea. The mesh block can then be grown into this void and the gap region remeshed. However, since the surface mesh must be regenerated to be aligned with the shock wave, the advancing layers must also be completely regenerated. Application of this technique in combination with other general adaptation methods could also prove useful and extend the applicability and effectiveness of the developed approach.

Currently for the extrusion techniques presented in Chapter VI a level of user interaction is required to add the surfaces into the geometry. One means to remove this requirement is to modify the nearfield growth functionality such that the wake mesh block can be connected to the nearfield mesh. This would prevent the need for the addition of the embedded surface into the geometric domain in the first instance. The extrusion process for the wakes and wing tip vortices also requires regeneration of the entire hybrid mesh. Extending the process to allow for local regeneration around the mesh blocks would also be useful to carry out in order to improve the efficiency of the method. Insertion of mesh blocks into hybrid meshes has been considered by Ebeida et al.^[191] Developing a means to allow for extrusion along more complex vortex paths rather than a linear direction would also be desirable. More complicated wing configurations including winglets or nacelle and pylons, flaps and so on could also be considered using the present methods. The source surface(s) used for the extrusion would not necessarily need to change, but still ultimately provide a quality resolution of the complex wake flow structures. Comparison of the present methods with the performance of multi-block structured meshes may also be interesting. Finally, the feature-aligned meshes are not restricted to RANS solvers. The performance of DES or LES solvers are likely to be much improved through application on

the feature-aligned meshes compared with standard hybrid meshes. This aspect is certainly worthy of investigation.

List of Published/Submitted Papers

1. Harris, M. J., Qin, N. and Martineau, D. G., *Using the medial axis to represent flow features for feature-aligned unstructured quad-dominant mesh generation*, submitted to Computers and Fluids, submitted October 2012 (accepted pending corrections).
2. Harris, M. J. and Qin, N., *Using the medial axis to represent complex flow structures for flow feature aligned mesh generation*, In *Proceedings, 51st AIAA Aerospace Sciences Meeting*, AIAA Paper 2013-0150, 7th-10th January 2013, Grapevine, Dallas / Ft. Worth, Texas, 2013.
3. Harris, M. J. and Qin, N., *Using the medial axis to represent complex flow structures for flow feature aligned mesh generation*, submitted to AIAA Journal, July 2013.

REFERENCES

1. Bonfigliodi, A., Paciorri, R. and Di Mascio, A., *The role of mesh generation, adaptation and refinement on the computation of flows featuring strong shocks*, 2012, Modelling and Simulation in Engineering.
2. Carpenter, M. and Casper, J., *The accuracy of shock capturing in two spatial dimensions*, 1999, AIAA Journal **37**(9): pp. 1072-1079.
3. Owen, S. J., *A Survey of Unstructured Mesh Generation Technology*, Proceedings, 7th International Meshing Roundtable, pp.239-267, 1998.
4. Frey, P. J. and George, P. L., *Mesh generation: application to finite elements*, 2nd Ed., Hermes Science Publishing, Oxford, 2000.
5. Thompson, J. F., *Numerical grid generation*, 1982, Applied Mathematics and Computation, pp. 10-11.
6. Eiseman, P. R. and Erlebacher, G., *Grid generation for the solution of partial differential equations*, Chapter 12, State of the art surveys on computational mechanics, Eds. Noor, A. K. and Oden, J. T., 1989.
7. Thompson, J. E., Warsi, Z. U. A. and Wayne Mastin, C., *Numerical Grid Generation*, Ed. Joe F. Thompson, 1997.
8. Steger, J. L. and Sorenson, R. L., *Use of hyperbolic partial differential equations to generate body fitted coordinates*, Proceedings, NASA Langley Workshop on Numerical Grid Generation Techniques, 1980.
9. Baker, T. J., *Developments and trends in three-dimensional mesh generation*, 1989, Applied Numerical Mathematics, **5**: pp. 275-304.
10. Tam, T. K. H. and Armstrong, C. G., *2D finite element mesh generation by medial axis subdivision*, 1991, Advances in Engineering Software, **13**: pp. 313-324.
11. Ang, P. Y. and Armstrong, C. G., *Adaptive Curvature-Sensitive Meshing of the Medial Axis*, Proceedings, 10th International Meshing Roundtable, pp.155-165, 2001.
12. Sampl, P., *Semi-Structured Mesh Generation Based on Medial Axis*, Proceedings, 9th International Meshing Roundtable, pp.21-32, 2000.
13. Gould, J., Martineau, D. and Fairy, R., *Automated Two-Dimensional Multi-block Meshing using the Medial Object*, Proceedings, 20th International Meshing

- Roundtable, pp. 437-452, 2012.
14. Frey, P. J. and Marechal, L., *Fast Adaptive Quadtree Mesh Generation*, Proceedings, 7th International Meshing Roundtable, pp. 211-224, 1998.
 15. McMorris, H and Kallinderis, Y., *Octree-advancing front method for generation of unstructured surface and volume meshes*, 1997, AIAA Journal **35**(6): pp. 976-984.
 16. Yerry, M. A. and Shephard, M. S., *A modified quadtree approach to finite element mesh generation*, 1983, IEEE Computer Graphics and Applications **3**(1): pp. 39-46.
 17. Schneiders, R., Schindler, R. and Weiler, F., *Octree-based generation of hexahedral element meshes*, Proceedings, 5th International Meshing Roundtable, 1996, pp. 205-216.
 18. Ito, Y., Shih, A. M. and Soni, B. K., *Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement techniques*, 2008, International Journal for Numerical Methods in Engineering, **77**(13): 1809-1833.
 19. Schneiders, R., Schindler, R. and Weiler, F., *Octree-based Generation of Hexahedral Element Meshes*, Proceedings, 5th International Meshing Roundtable, pp. 205-216, 1996.
 20. Schneiders, R., *An Algorithm for the Generation of Hexahedral Element Meshes based on an Octree Technique*, Proceedings, 6th International Meshing Roundtable, pp. 183-194, 1997.
 21. Shepherd, M. S. and Georges, M. K., *Automatic three-dimensional mesh generation by the finite octree technique*, 1991, International Journal of Numerical Methods in Engineering, **32**: pp. 709-749.
 22. Delaunay, B., *Sur la sphère vide*, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, **7**:793–800, 1934.
 23. Borouchaki, H., Hecht, F., Saltel, E. and George, P. L., *Reasonably Efficient Delaunay Based Mesh Generator in 3 Dimensions*, Proceedings, 4th International Meshing Roundtable, pp. 3-14, 1995.
 24. Rebay, S. *Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm*, Journal of Computational Physics, **106**: pp. 125-138.
 25. George, P. L., *Tet Meshing: Construction, Optimization, And Adaptation*, Proceedings, 8th International Meshing Roundtable, pp. 133-141, 1999.
 26. Cavendish, J. C., Field, D. A. and Frey, W. H., *An approach to automatic three-*

- dimensional finite element mesh generation*, 1985, International Journal for Numerical Methods in Engineering, **21**: pp. 329-347.
27. Baker, T. J., *Three-dimensional mesh generation by triangulation of arbitrary point sets*, 1986, Proceedings, AIAA 8th Computational Fluid Dynamics Conference, Honolulu, HI, **87**: pp.1124.
 28. Bowyer, A., *Computing Dirichlet tessellations*, 1981, Computer Journal **24**(2): pp. 162-167.
 29. Watson, D. F., *Computing the n-dimensional Delaunay Tessellation with applications to Voronoi polytopes*, 1981, Computer Journal, **24**(2): pp. 167-172.
 30. Green, P. and Sibson, R., *Computing Dirichlet tessellations in the plane*, 1978, Computer Journal, **21**, pp. 168-173.
 31. Lo, S. H., *A new mesh generation scheme for arbitrary planar domains*, 1985, International Journal for Numerical Methods in Engineering, **21**: pp, 1403-1426.
 32. Frey, P., Borouchaki, H. and George, P., *Delaunay Tetrahedralization using an Advancing-Front Approach*, Proceedings, 5th International Meshing Roundtable, pp. 31-46, 1996.
 33. Fleischmann, P. and Selberherr, S., *Three-Dimensional Delaunay Mesh Generation Using a Modified Advancing Front Approach*, Proceedings, 6th International Meshing Roundtable, pp. 267-278, 1997.
 34. Ito, Y., Shih, A. M. and Soni, B. K., *Reliable Isotropic Tetrahedral Mesh Generation Based on an Advancing Front Method*, Proceedings, 13th International Meshing Roundtable, pp. 95-106, 2004.
 35. Lohner, R. and Parikh, P., *Three-dimensional grid generation by the advancing front method*, 1988, International Journal for Numerical Methods in Fluids, **8**: 1135-1149.
 36. George, P. L. and Seveno, E., *The advancing-front mesh generation method revisited*, 1994, International Journal for Numerical Methods in Engineering, **37**: pp. 3605-3619.
 37. Lohner, R., *Extensions and improvements of the advancing-front grid generation technique*, 1996, Communications in Numerical Methods in Engineering, **12**: pp. 683-702.
 38. Miyoshi, K. and Blacker, T., *Hexahedral Mesh Generation Using Multi-Axis Cooper Algorithm*, Proceedings, 9th International Meshing Roundtable, pp. 89-97, 2000.

39. Ledoux, F. and Weill, J., *An Extension of the Reliable Whisker Weaving Algorithm*, Proceedings, 16th International Meshing Roundtable, pp. 215-232, 2007.
40. Thompson, D. S. and Soni, B. K., *Generation of Quad and Hex Dominant Semistructured Meshes Using An Advancing Layer Scheme*, Proceedings, 8th International Meshing Roundtable, pp. 171-178, 1999.
41. Lo, S. H., *Generating quadrilateral elements on plane and over curved surfaces*, 1989, Computers and Structures, **31**(3): pp. 431-426.
42. Lee, C. K. and Lo, S. H., *A new scheme for the generation of a graded quadrilateral mesh*, 1994, Computers and Structures, **52**: pp. 847-857.
43. Blacker, T. D. and Stephenson, M. B., *Paving: A new approach to automated quadrilateral mesh generation*, 1991, International Journal for Numerical Methods in Engineering, **32**(4): pp. 811-847.
44. McRae, D. S., *r-refinement grid adaptation algorithms and issues*, 2000, Computer Methods in Applied Mechanics and Engineering, **189**: pp. 1161-1182.
45. Mavriplis, D. J., *Unstructured mesh discretizations and solvers for computational aerodynamics*, 18th AIAA Computational Fluid Dynamics Conference, Miami, FL., 2007.
46. Mavriplis, D. J., *Unstructured Grid Techniques*, 1997, Annual Review of Fluid Mechanics, **29**: pp. 473-514.
47. Levy, D. W., Powell, K. G. and Van Leer, B., *Use of a rotated Riemann solver for the two-dimensional Euler equations*, 1993, Journal of Computational Physics, **106**: pp. 201-214.
48. Nishikawa, H. and Kitamura, K., *Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers*, 2008, Journal of Computational Physics, **227**: pp. 2560-2581.
49. Van Leer, B., *Upwind and high-resolution methods for compressible flow: from donor cell to residual-distribution schemes*, 2006, Communications in Computational Physics, **1**(2): pp. 192-206.
50. Roy, C. J., *Strategies for Driving Mesh Adaptation in CFD*, Proceedings, 47th AIAA Aerospace Sciences Meeting, 2009, Orlando, FL, AIAA-2009-1302.
51. Ren, Y., *A robust shock-capturing scheme based on rotated Riemann solvers*, 2003, Computers and Fluids, **32**: pp. 1379-1403.

52. Dwight, R. P., *Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation*, Journal of Computational Physics, 2008, **227**(5): pp. 2845-2863.
53. Picasso, M., Alauzet, F., Borouchaki, H. and George, P. L., *A numerical study of some Hessian recovery techniques on isotropic and anisotropic meshes*, 2011, SIAM Journal on Scientific Computing, **33**: pp. 1058-1076.
54. Lipnikov, K. and Vassilevski, Y., *Analysis of Hessian Recovery Methods for Generating Adaptive Meshes*, Proceedings, 15th International Meshing Roundtable, pp. 163-172, 2006.
55. Almeida, R. C., Feijoo, R. A., Galeao, A. C., Padra, C. and Silva, R. S., *Adaptive finite element computational fluid dynamics using an anisotropic error estimator*, 2000, Computer Methods in Applied Mechanics and Engineering, **182**: pp. 379-400.
56. Lipnikov, K. and Vassilevski, Y., *On discrete boundaries and solution accuracy in anisotropic adaptive meshing*, Proceedings, 14th International Meshing Roundtable, 2005.
57. Qin, N and Liu, X., *Flow feature aligned grid adaptation*, International Journal for Numerical Methods in Engineering, 2006, **67**: pp. 787-814
58. Babushka, I. and Aziz, A. K., *On the angle condition in the finite-element method*, 1976, SIAM Journal on Numerical Analysis, **13**(2): pp. 214-226.
59. Weatherill, N. P., *A strategy for the use of hybrid structured-unstructured meshes in CFD*, in : KW Morton and MJ. Baines, eds., Numerical Methods for Fluid Dynamics (Oxford University Press, Oxford, 1988.
60. Kallinderis, Y., Khawaja, A and McMorris, H., *Hybrid prismatic / tetrahedral grid generation for complex geometries*, 1995, AIAA Paper 1995-0211.
61. Khawaja, A., McMorris, H and Kallinderis, Y., *Hybrid grids for viscous flows around complex 3D geometries including multiple bodies*, 12th AIAA CFD Conference, AIAA Paper 1995-1685.
62. Khawaja, A., Kallinderis, Y. and McMorris, H., *Adaptive Hybrid Grids for Diverse Industrial Applications*, Proceedings, 7th International Meshing Roundtable, pp. 167-184, 1998.
63. Gloth, O. and Vilsmeier, R., *Level Sets as Input for Hybrid Mesh Generation*, Proceedings, 9th International Meshing Roundtable, pp. 137-146, 2000.
64. Garimella, R. V. and Shephard M., *Boundary Layer Meshing for Viscous Flows in*

- Complex Domains*, Proceedings, 7th International Meshing Roundtable, pp. 107-118, 1998.
65. Leatham, M., Stokes, S., Shaw, J. A., Cooper, J., Appa, J. and Blayloc, T. A., *Automatic mesh generation for rapid response Navier-Stokes calculations*, AIAA Paper 2000-2247, 2000.
 66. Handbook of Grid Generation, edited by Thompson, J. F., Soni, B. K. and Weatherill, N. P. 1999, CRC Press LLC.
 67. Cagnone, J. S., Vermaire, B. C. and Nadarajah S. K., *A polynomial adaptive LCP scheme for viscous compressible flows*, in Proceedings, 51st AIAA Aerospace Sciences Meeting, 2013, AIAA-2013-1000.
 68. Shi, L. and Wang, Z. J., *Adjoint-based error estimation and hp-adaptation for the high-order CPR method*, in Proceedings, 51st AIAA Aerospace Sciences Meeting, 2013, AIAA-2013-0999.
 69. Park, M. A., *Three-dimensional turbulent RANS adjoint-based error correction*, 2003, AIAA Paper 2003-3849.
 70. Park, M. A., *Adjoint-based, three-dimensional error prediction and grid adaptation*, 2004, AIAA Journal **42**(9): pp. 1854-1862.
 71. Venditti, D. A. and Darmofal, D. L., *Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows*, 2003, Journal of Computational Physics, **187**(1): pp. 22-46.
 72. Venditti, D. A. and Darmofal, D. L., *Grid adaptation for functional outputs: application to two-dimensional inviscid flows*, 2002, Journal of Computational Physics, **176**(1): pp. 40-69.
 73. Loseille, A., Dervieux, A. and Alauzet, F., *A 3D goal-oriented anisotropic mesh adaptation applied to inviscid flows in aeronautics*, 48th AIAA Aerospace Sciences Meeting, 2010, AIAA 2010-1067.
 74. Fidkowski, K. J. and Darmofal, D. L., *Review of output-based error estimation and mesh adaptation in computational fluid dynamics*, 2011, AIAA Journal, **49**(4): pp. 673-694.
 75. Fidkowski, K. J. and Darmofal, D. L., *Output-based error estimation and mesh adaptation in computational fluid dynamics: overview and recent results*, In Proceedings, 47th AIAA Aerospace Sciences Meeting, 2009, AIAA 2009-1303.

76. Nemec, M. and Aftosmis, M. J., *Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes*, AIAA Paper 2007-4187, Proceedings, 18th AIAA Computational Fluid Dynamics Conference, 2007.
77. Nemec, M., Aftosmis, M. J. and Wintzer, M., *Adjoint-based adaptive mesh refinement for complex geometries*, AIAA Paper 2008-0725, Proceedings, 46th AIAA Aerospace Sciences Meeting, 2008.
78. Yi Li, L., Allaneau, Y. and Jameson, A., *Continuous adjoint approach for adaptive mesh refinement*, AIAA Paper 2011-3982, Proceedings, 20th AIAA Computational Fluid Dynamics Conference, 2011.
79. Ozcer, I. A. and Kandil, O. A., *FUN3D / OptiGRID coupling for unstructured grid adaptation for sonic boom problems*, AIAA Paper 2008-61, Proceedings, 46th AIAA Aerospace Sciences Meeting, 2008.
80. Tam, A., *An anisotropic adaptive method for the solution of 3D inviscid and viscous compressible flows*, PhD Thesis, Concordia University Montreal, Quebec, Canada, 1998.
81. Ait-Ali-Yahia, D., Habashi, W. and Tam, A., *A directionally adaptive methodology using an edge-based error estimate on quadrilateral grids*, International Journal for Numerical Methods in Fluids, 1996, **23**(7): pp. 673-690.
82. Tam, A., Ait-Ali-Yahia, D., Robichaud, M. P., Moore, M., Kozel, V. and Habashi, W. G., *Anisotropic mesh adaptation for 3D flows on structured and unstructured grids*, 2000, Computer Methods in Applied Mechanics and Engineering, **189**(4): pp. 1205-1230.
83. Tam, A. and Robichaud, M., *A 3D adaptive anisotropic method for external and internal flows*, AIAA Paper 1998-0771, 1998.
84. Bono, G. and Awruch, A. M., *An adaptive mesh strategy for high compressible flows based on nodal re-allocation*, 2008, Journal of the Brazilian Society of Mechanical Sciences and Engineering, **30**(3): pp. 189-196.
85. Ait-Ali-Yahia, D., Baruzzi, G., Habashi, W., Fortin, M., Dompierre, J. and Vallet, M., *Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part II: structured grids*, International Journal for Numerical Methods in Fluids, 2002, **39**(8): pp. 657-673.
86. Tishkin, V. and Bogomolov, K. *Adaptive High-Aspect Ratio Structured 2-D Grids*, Proceedings, 9th International Meshing Roundtable, pp. 193-204, 2000.

87. Qin, N. and Zhu, Y., *Grid adaptation for shock / turbulent boundary layer interaction*, 1998, AIAA Journal **37**(9): pp. 1129-1131.
88. Yuan, L. and Tang, T., *Resolving the shock-induced combustion by an adaptive mesh redistribution method*, Journal of Computational Physics, 2007, **224**(2): pp. 587-600.
89. Guoy, D. and Erickson, J., *Automatic Blocking Scheme for Structured Meshing in 2D Multiphase Flow Simulation*, Proceedings, 13th International Meshing Roundtable, pp. 121-132, 2004.
90. Qin, N., Carnie, G., Moigne, A.L., Liu, X. and Shahpar, S. *Buffer layer method for linking two non-matching multi-block structured grids*, 2009, AIAA-2009-1361.
91. Qin, N., Wang, Y. and Carnie, G., *Zipper Layer Method*, Proceedings of the 20th International Meshing Roundtable, October 2011.
92. Habashi, W. G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M. and Vallet, M., *Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles*, 2000, International Journal for Numerical Methods in Fluids, **32**(6): pp. 725-744.
93. Dompierre, J., Vallet, M., Bourgault, Y., Fortin, M. and Habashi, W., *Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part III: Unstructured meshes*, International Journal for Numerical Methods in Fluids, 2002, **39**(8): pp. 675-702.
94. Frey, P. and Alauzet, F. *Anisotropic mesh adaptation for CFD computations*, Computer Methods in Applied Mechanics and Engineering, 2005, **194**: pp. 5068-5082.
95. Castro-Diaz, M. J., Hecht, F., Mohammadi, B. and Pironneau, O., *Anisotropic unstructured mesh adaptation for flow simulations*, 1997, International Journal for Numerical Methods in Fluids, **25**(4): pp. 475-491.
96. Fortin, M., Vallet, G., Dompierre, Y., Bourgault, Y. and Habashi, W., *Anisotropic mesh adaptation: theory, validation and applications*, 1996, In Proceedings of ECCOMAS CFD, John Wiley & Sons, 1996.
97. Marcum, D. L. and Weatherill, N. P., *A procedure for efficient generation of solution adapted unstructured grids*, 1995, Computer Methods in Applied Mechanics and Engineering, **127**: 259-268.

98. Marcum, D. L., *Adaptive unstructured grid generation for viscous flow applications*, 1995, AIAA Journal, **34**(11): pp. 2240-2243.
99. Marcum, D. L. and Weatherill, N. P., *Aerospace applications of solution adaptive finite element analysis*, 1995, Computer Aided Geometric Design, **12**(7): pp. 709-731.
100. Baker, T. J., *Mesh adaptation strategies for problems in fluid dynamics*, 1997, Finite Elements in Analysis and Design, **25**: pp243-273.
101. Marcum, D. L. and Gaither, K. P., *Solution adaptive unstructured grid generation using pseudo-pattern recognition techniques*, Proceedings, 13th Computational Fluid Dynamics Conference, 1997, AIAA Paper 1997-1860.
102. Loseille, A. and Lohner, R., *Cavity-based operators for mesh adaptation*, AIAA 2013-0152, Proceedings, 51st AIAA Aerospace Sciences Meeting, 2013.
103. Bibb, K. L., Gnoffo, P. A., Park, M. A. and Jones, W. T., *Parallel, gradient-based anisotropic mesh adaptation for re-entry vehicle configurations*, AIAA Paper 2006-3579, 9th AIAA.ASME Joint Thermophysics and Heat Transfer Conference, June 5-8, 2006, San Francisco, CA.
104. Yano, M. and Darmofal, D. L., *An optimization framework for anisotropic simplex mesh adaptation: application to aerodynamic flows*, 2012, Journal of Computational Physics, **231**(22): pp. 7626-7649.
105. Peraire, J. and Morgan, K., *Unstructured mesh generation including directional refinement for aerodynamic flow simulation*, 1997, Finite Elements in Analysis and Design, **25**(3-4), pp. 343-355.
106. Marcum D. L. and Weatherill N. P., *Unstructured grid generation using iterative point insertion and local reconnection*, AIAA Journal **33**(9): pp. 1619-1625, 1995.
107. Frey, Pascal J., *About Surface Remeshing*, Proceedings, 9th International Meshing Roundtable, Sandia National Laboratories, pp. 123-136, October 2000.
108. Loseille, A. and Lohner, R., *On 3D Anisotropic Local Remeshing for Surface, Volume and Boundary Layers*, Proceedings, 18th International Meshing Roundtable, pp. 611-630, 2009.
109. Alauzet, F., Loseille, A., Dervieux, A. and Frey, P., *Multi-Dimensional Continuous Metric for Mesh Adaptation*, Proceedings, 15th International Meshing Roundtable, pp. 191-214, 2006.
110. Borouchaki, H., Hecht, F. and Frey, P., *Mesh Gradation Control*, Proceedings, 6th

- International Meshing Roundtable, Sandia National Laboratories, pp. 131-141, October 1997.
111. Castro-Diaz, M.J., Hecht, F. and Mohammadi, B., *New Progress in Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations*, Proceedings, 4th International Meshing Roundtable, pp. 73-85, 1995.
 112. Frey, P. and Alauzet, F., *Anisotropic mesh adaptation for transient flows simulations*, Proceedings, 12th International Meshing Roundtable, pp. 335-348, 2003.
 113. Loseille, A. and Alauzet, F., *Optimal 3D Highly Anisotropic Mesh Adaptation Based on the Continuous Mesh Framework*, Proceedings, 18th International Meshing Roundtable, pp.575-594, 2009.
 114. Peraire, J., Peiro, J. and Morgan, K., *Adaptive remeshing for three-dimensional compressible flow computations*, 1992, Journal of Computational Physics, **103**(2): pp. 269-285.
 115. Mavriplis, D. J. *Adaptive mesh generation for viscous flows using Delaunay triangulation*, 1990, Journal of Computational Physics, **90**(2): pp. 271-291.
 116. Peraire, J., Vahdati, M., Morgan, K. and Zienkiewicz, O. C., *Adaptive remeshing for compressible flow computations*, 1987, Journal of Computational Physics, **72**(2): pp. 449-466.
 117. Yamahara, T., Nakahashi, K. and Kim, H., *Adaptive mesh refinement using viscous adjoint method for multi-element airfoil computations*, AIAA Paper 2008-416, Proceedings, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.
 118. Yamazaki, W., Matsushima, K and Nakahashi, K., *Drag decomposition-based adaptive mesh refinement*, 2007, Journal of Aircraft, **44**(6): pp. 1896-1905.
 119. Tchon, K., Dompierre, J. and Camarero, R., *Automated refinement of conformal quadrilateral and hexahedral meshes*, 2004, International Journal for Numerical Methods in Engineering, **54**(12): pp. 1539-1562.
 120. Tchon, K., Dompierre, J. and Camarero, R., *Conformal refinement of all-quadrilateral and all-hexahedral meshes according to an anisotropic metric*, In Proceedings, 11th International Meshing Roundtable, 2002.

121. Tchou, K. and Camarero, R., *Quad-dominant mesh adaptation using specialized simplicial optimization*, Proceedings, 15th International Meshing Roundtable, 2006, pp. 21-38.
122. Borouchaki, H. and Frey, P. J., *Adaptive triangular-quadrilateral mesh generation*, 1998, International Journal for Numerical Methods in Engineering, **41**: pp.915-934.
123. Borouchaki, H., Frey, P. and George, P. L., *Unstructured Triangular-Quadrilateral Mesh Generation. Application to Surface Meshing*, Proceedings, 5th International Meshing Roundtable, pp. 229-242, 1996.
124. Zheng, H. W., Qin, N., Nicolleau, F., Shu, C., *Quadrilateral cell-based anisotropic adaptive solution for the Euler equations*, 2011, Communications in Computational Physics, **9**(1): pp. 68-88.
125. Staten, M. L., Owen, S. J. and Blacker, T. D., *Unconstrained Paving & Plastering: A New Idea for All Hexahedral Mesh Generation*, Proceedings, 14th International Meshing Roundtable, pp. 399-416, 2005.
126. Staten, M. L., Kerr, R. A., Owen, S. J. and Blacker, T. D., *Unconstrained Paving and Plastering: Progress Update*, Proceedings, 15th International Meshing Roundtable, pp. 469-486, 2006.
127. Schneiders, R., *Automatic Generation of Hexahedral Finite Element Meshes*, Proceedings, 4th International Meshing Roundtable, pp.103-114, 1995.
128. Johnston, B. P., Sullivan Jr., J. M. and Kwasnik, A., *Automatic conversion of triangular finite element meshes to quadrilateral elements*, 1991, International Journal for Numerical Methods in Engineering, **31**: pp. 67-84.
129. Schneiders, R., *Refining quadrilateral and hexahedral meshes*, 5th International Conference on Numerical Grid Generation in Computational Fluid Simulations, Mississippi State University, pp. 679-688, 2006.
130. Merkley, K., Ernst, C., Shepherd, J. F. and Borden, M. J., *Methods and applications of generalised sheet insertion for hexahedral meshing*, Proceedings, 16th International Meshing Roundtable, pp. 233-250, 2007.
131. Garimella, R., *Conformal refinement of unstructured quadrilateral meshes*, Proceedings, 18th International Meshing Roundtable, pp. 31-44, 2009.
132. Blacker, T., *Meeting the Challenge for Automated Conformal Hexahedral Meshing*, Proceedings, 9th International Meshing Roundtable, pp. 11-19, 2000.

133. Anderson, B., Benzley, S, and Owen, S. J., *Automatic All Quadrilateral Mesh Adaption through Refinement and Coarsening*, Proceedings, 18th International Meshing Roundtable, pp. 557-574, 2009
134. Woodbury, A. C., Shepherd, J. F., Statenand, M. L. and Benzley, S. E., *Localized Coarsening of Conforming All-Hexahedral Meshes*, Proceedings, 17th International Meshing Roundtable, pp. 603-618, 2008.
135. Borden, M., Benzley, S., Mitchell, S. A., White, D. R. and Meyers, R., *The Cleave and Fill Tool: An All-Hexahedral Refinement Algorithm for Swept Meshes*, Proceedings, 9th International Meshing Roundtable, pp. 69-76, 2000.
136. Harris, N. J., Benzley, S. E. and Owen, S. J., *Conformal Refinement of All-Hexahedral Element Meshes Based on Multiple Twist Plane Insertion*, Proceedings, 13th International Meshing Roundtable, pp. 157-168, 2004.
137. Aftosmis, M. J., Berger, M. J., Melton J. E., Robust and efficient Cartesian mesh generation for component-based geometry, 1997, AIAA Paper 1997-0196.
138. Wintzer, M., Nemece, M. and Aftosmis, M. J., *Adjoint-based adaptive mesh refinement for sonic boom prediction*, AIAA Paper 2008-6593, 26th AIAA Applied Aerodynamics Conference, 2008.
139. Dawes, W. N., Harvey S. A., Fellows, S., Favaretto, C. F. and Vellivelli, A., *Viscous layer meshes from level sets on cartesian meshes*, in Proceedings, 45th AIAA Aerospace Sciences Meeting, 2007, Reno, NV, AIAA-2007-0555.
140. Dawes, W. N., Kellar, W. P. and Harvey, S. A., *Using level sets as the basis for a scalable, parallel geometry engine and mesh generation system*, 2009, AIAA Paper AIAA-2009-0372.
141. Keats, W. A. and Lien, F. S., *Two-dimensional anisotropic Cartesian mesh adaptation for the compressible Euler equations*, 2004, International Journal for Numerical Methods in Fluids, **46**(11): pp. 1099-1125.
142. Ham, F. E., Lien, F. S. and Strong, A. B., *A Cartesian grid method with transient anisotropic adaptation*, 2002, Journal of Computational Physics **179**(2): pp. 469-494.

143. Ito, Y., Shih, A. M., Koomullil, R. P. and Soni, B. K., *A solution-based adaptive redistribution method for unstructured meshes*, Proceedings, 15th International Meshing Roundtable, pp. 147-161, 2006.
144. Ito, Y., Shih, A. M., Koomullil, R., Kasmai, N., Jankun-Kelly, M. and Thompson, D., *Solution adaptive mesh generation using feature-aligned emedded surface meshes*, 2009, AIAA Journal, **47**(8): pp. 1879-1888.
145. Ito, Y. Shih, A. M. and Soni, B. K., *Hybrid mesh generation with embedded surface meshes using a multiple marching approach*, International Journal for Numerical Methods in Fluids, 2011, **67**(1): pp. 1-7.
146. Shih, A. M., Ito, Y., Koomullil, R., Kasmai, T., Jankun-Kelly, M., Thompson, D. and Brewer, W., *Solution adaptive mesh generation using feature-aligned embedded surface meshes*, in Proceedings, 45th AIAA Aerospace Sciences Meeting, 2007, AIAA 2007-0558.
147. Burgos, M. A., Corral, R., Fernández-Castañeda, J. and López, C., *Rapid Meshing of Turbomachinery Rows Using Semi-Unstructured Conformal Grids*, Proceedings, 15th International Meshing Roundtable, pp.93-108, 2006.
148. Craft T., Iacovides, H. and Skillen, A., *A new overset grid algorithm applied to the simulation of flows involving complex geometries*, 15th International Conference on Fluid Flow Technologies, Budapest, Hungary, 2012.
149. Berger M. J., and Oliger, J., *Adaptive mesh refinement for hyperbolic partial differential equations*, 1984, Journal of Computational Physics, **53**(3): pp. 484-512.
150. Berger, M. J., and Jameson, A., *Automatic adaptive grid refinement for the Euler equations*, 1985, AIAA Journal, **23**(4): pp. 561-568.
151. Chawla, K. and Banks, D., *Tracking flow features using overset grids*, In *Proceedings, 31st AIAA Aerospace Sciences Meeting*, 1993, AIAA-1993-0197.
152. Meakin, R. L., *Grid related issues for static and dynamic geometry problems using systems of overset structured grids*, 1995, Technical Report, NASA Ames Research Center.
153. Meakin, R. L., *On the spatial and temporal accuracy of overset grid methods for moving body problems*, 1994, AIAA Paper 1994-1925.
154. Guerrero J., *Overset composite grids for the simulation of flows in complex moving geometries*, 2006, DICAT, University of Genoa, Italy.

155. Choi, Y.H. and Merkle, C.L., *The Application of Preconditioning to Viscous Flows*, Journal of Computational Physics, 1993, **105**: pp. 207-223.
156. Spalart, P.R.; Allmaras, S.R., *A One-Equation Turbulence Model for Aerodynamic Flows*, 1992, AIAA Paper 1992-0439.
157. Gerhold, T., Friedrich, O., Evans, J. and Galle, M., *Calculation of complex three-dimensional configurations employing the DLR Tau-code*, AIAA Paper 1997-0167, Proceedings, 35th Aerospace Sciences Meeting, 1997.
158. Kroll, N.; Radespiel, R., *An improved flux vector split discretisation scheme for viscous flows*, DLR-FB 93-53, 1993.
159. Wada, Y. and Liou, M., *A flux splitting scheme with high-resolution and robustness for discontinuities*, AIAA Paper 1194-0083, 1994.
160. Barth, T., *A 3D upwind Euler solver for unstructured meshes*, AIAA Paper 1991-1548, 1991.
161. Venkatakrishnan, V., *On the accuracy of limiters and convergence to steady state solutions*, AIAA Paper 1993-0880, 1993.
162. Anderson, W. and Bonhaus, L., *An implicit upwind algorithm for computing turbulent flows on unstructured grids*, 1994, Computers and Fluids, **23**: pp. 1-21.
163. Holmes, D., Connell, S., *Solution of the 2D Navier Stokes equations on unstructured adaptive grids*, AIAA Paper 1989-1932, 1989.
164. Yoon, S. and Jameson, A., *Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations*, 1988, AIAA Journal **26**(9): pp. 1025-1026.
165. Liou, M. and Steffen, C. J., *A new flux splitting scheme*, 1993, Journal of Computational Physics, **107**: pp. 23-29.
166. Lovely, D. and Haines, R., *Shock detection from computational fluid dynamics results*, AIAA Paper 1999-3285, 14th AIAA Computational Fluid Dynamics Conference, 1999.
167. Jeong, J. and Hussain, F., *On the identification of a vortex*, 1995, Journal of Fluid Mechanics, **285**: pp. 69-94.
168. Canann, S. A., Tristano, J. R. and Staten, M. L., *An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes*, Proceedings, 7th International Meshing Roundtable, pp. 479-494, 1998.

169. Freitag, L. A. and Ollivier-Gooch, C., *A comparison of tetrahedral mesh improvement techniques*, Proceedings, 5th International Meshing Roundtable, pp. 87-100, 1996.
170. Freitag, L. A. and Ollivier-Gooch, C., *Tetrahedral mesh improvement using swapping and smoothing*, 1997, International Journal for Numerical Methods in Engineering **40**: pp. 3979-4002.
171. Buscaglia, G. C. and Dari, E. A., Anisotropic mesh optimization and its application in adaptivity, 1997, International Journal for Numerical Methods in Engineering **40**: pp. 4119-4136.
172. Pain, C. C., Umpieby, A. P., de Oliveira, C. R. E. and Goddard, A. J. H., *Tetraherdral mesh optimization and adaptivity for steady-state and transient finite element calculations*, 2001, Computer Methods in Applied Mechanics and Engineering **190**: pp. 3771-3796.
173. Dey, K. T. and Zhao, W., *Approximating the Medial Axis from the Voroni Diagram with a Convergence Guarantee*, 2003, Algorithmica, 38(1): pp. 179-200.
174. Xia, H. and Tucker, P. G., *Distance Solutions for Medial Axis Transform*, Proceedings, 18th International Meshing Roundtable, pp. 247-265, 2009.
175. Sheehy, D. J., Armstrong, C. G. and Robinson, D. J., *Shape description by medial surface construction*, 1996, IEEE Transactions on Visualization and Computer Graphics, **2**(1): pp. 62-72.
176. Tchou, K., Khachan, M., Guibault, F. and Camarero, R., *Constructing anisotropic geometric metrics using octrees and skeletons*, Proceedings, 12th International Meshing Roundtable, pp. 293-304, 2003.
177. Sheffer, A., Etzion, M. Rappoport, A. and Bercovier, M., *Hexahedral Mesh Generation using the Embedded Voronoi Graph*, Proceedings, 7th International Meshing Roundtable, pp.347-364, 1998.
178. Edelsbrunner, H. and Mucke, E., *Three-dimensional alpha shapes*, 1994, ACM Transactions on Graphics, **13**(1): pp. 43-72.
179. Edelsbrunner, H., *Weighted alpha shapes*, Technical Report UIUC-CS-R-92-1760, Computer Science Department, University of Illinois, Urbana, Illinois, 1992.
180. Blum, H. A., *A transformation for extracting new descriptors or shape*, 1967, Models for the Perception of Speech and Visual Form, pp. 362-380.

181. Cook, P., McDonald, M. and Firmin, M., *Aerofoil RAE2822 - pressure distributions and boundary layer and wake measurements*, AGARD Report AR138, 1979.
182. Woodward P., Colella P. *The numerical simulation of two-dimensional fluid flow with strong shocks*, 1984, Journal of Computational Physics,. **54**: pp. 115-173.
183. Schmitt, V. and Charpin, F., *Pressure distributions on the ONERA-M6-Wing at transonic Mach numbers*, Experimental Database for Computer Program Assessment, Report of the Fluid Dynamics Panel Working Group 04, AGARD AR138, 1979.
184. Strang, W. J. and McKinlay, R. M., *Concorde in Service*, Aeronautical Journal, February 1979.
185. Joubarne, E. and Guibault, F., *3D metric-based anisotropic mesh adaptation for vortex capture*, 2011, Mathematics and Computers in Simulation, **82**: pp. 163-180.
186. Craft, T. J., Gerasimov, A. V., Launder, B. E. and Robinson, C. M. E., *A computational study of the nearfield generation and decay of wing tip vortices*, 2006, International Journal of Heat and Fluid Flow, **27**(4): pp. 684-695.
187. Wake, B. E. and Choi, D., *Investigation of High-Order Upwinded Differencing for Vortex Convection*, 1996, AIAA Journal, **34**(2): pp. 332-337.
188. Allen, C. B., *Parallel simulation of lifting rotor wakes in forward flight*, 2007, International Journal of Numerical Analysis and Modelling, **4**(1): pp. 1-15.
189. Azlin, M. A., Mat Taib, C. F., Kasolang, S., and Muhammad, F. H., *CFD analysis of winglets at low subsonic flow*, Proceedings of the World Congress on Engineering, Vol. 1, WCE 2011, London U.K., July 6-8, 2011.
190. Krieger, M. and Wimmer, P., *Dissipation of vortices in CFD simulations*, 2003, Proceedings in Applied Mathematics and Mechanics, **3**: pp. 344-345.
191. Ebeida, Mohamed S., Mestreau, E., Zhang, Y. and Dey, S., *Mesh Insertion of Hybrid Meshes*, Proceedings, 18th International Meshing Roundtable, pp. 337-357, 2009.
192. Chu, J. and Luckring, J. M., *Experimental Surface Pressure Data Obtained on 65-Degree Delta Wing Across Reynolds Number and Mach Number Ranges, Volume 4- Large-Radius Leading Edge*, 1996, NASA Technical Memorandum 4645.
193. Jones, D. J., *Test cases for inviscid flow field methods: reference test cases and contributions*, Technical Report AR-211, AGARD, NATO, 1985.

194. Harris, M. J. and Qin, N. *Using the Medial Axis to Represent Complex Flow Structures for Flow Feature Aligned Mesh Generation*, In *Proceedings, 51st AIAA Aerospace Sciences Meeting*, AIAA Paper 2013-0150, 7th-10th January 2013, Grapevine, Dallas / Ft. Worth, Texas, 2013.
195. Gerz, T. and Holzapfel, F., *Wing-Tip Vortices, Turbulence, and the Distribution of Emissions*, 1999, AIAA Journal, **37**(10): pp. 1270-1276.
196. Revell, A., Iaccarino, G. and Wu, X., *Advanced RANS modeling of wingtip vortex flows*, Center for Turbulence Research, Proceedings of the Summer Program, pp. 73-85, 2006.
197. Moir, I. R. M., *Measurements on a Two-Dimensional Aerofoil with High Lift Devices*, AGARD AR 303, Vol. II, pp. A2.1-A2.12, 1994.