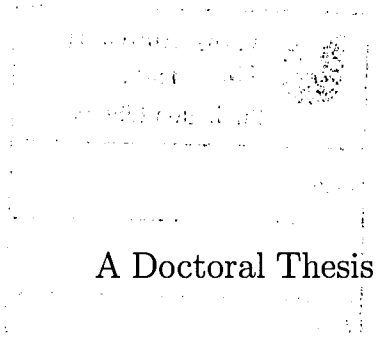
A mesh transparent numerical method for large-eddy simulation of compressible turbulent flows.

A Mesh Transparent Numerical Method for Large-Eddy Simulation of Compressible Turbulent Flows

by

Indi Himawan Tristanto, B.Eng.(Hons)



A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of
Doctor of Philosophy of Loughborough University
18 September 2004

A Mesh Transparent Numerical Method for Large-Eddy Simulation of Compressible Turbulent Flows

by

Indi Himawan Tristanto

Abstract

A Large Eddy-Simulation code, based on a mesh transparent algorithm, for hybrid unstructured meshes is presented to deal with complex geometries that are often found in engineering flow problems. While tetrahedral elements are very effective in dealing with complex geometry, excessive numerical diffusion often affects results. Thus, prismatic or hexahedral elements are preferable in regions where turbulence structures are important. A second order reconstruction methodology is used since an investigation of a higher order method based upon Lele's compact scheme has shown this to be impractical on general unstructured meshes. The convective fluxes are treated with the Roe scheme that has been modified by introducing a variable scaling to the dissipation matrix to obtain a nearly second order accurate centred scheme in statistically smooth flow, whilst retaining the high resolution TVD behaviour across a shock discontinuity. The code has been parallelised using MPI to ensure portability.

The base numerical scheme has been validated for steady flow computations over complex geometries using inviscid and RANS forms of the governing equations. The extension of the numerical scheme to unsteady turbulent flows and the complete LES code have been validated for the interaction of a shock with a laminar mixing layer, a Mach 0.9 turbulent round jet and a fully developed turbulent pipe flow. The mixing layer and round jet computations indicate that, for similar mesh resolution of the shear layer, the present code exhibits results comparable to previously published work using a higher order scheme on a structured mesh. The unstructured meshes have a significantly smaller total number of nodes since tetrahedral elements are used to fill to the far field region. The pipe flow results show that the present code is capable of producing the correct flow features. Finally, the code has been applied to the LES computation of the impingement of a highly under-expanded jet that produces plate shock oscillation. Comparison with other workers' experiments indicates good qualitative agreement for the major features of the flow. However, in this preliminary computation the computed frequency is somewhat lower than that of experimental measurements.

Keywords: Computational Method, Unstructured Mesh, Finite-Volume, Compressible Flow, LES, RANS, Parallel Computing

The Fragrance of the Rose (Words)

The disciples were absorbed in a discussion of Lao-tzu's dictum:
Those who know do not say;
Those who say do not know.
When the master entered,
They asked him what the words meant.
Said the master, "Which of you knows the fragrance of a rose?"
All of them indicated that they knew.
Then he said, "put it into words."
All of them were silent.

Anthony deMello, One Minute Wisdom

To my wife, Ira
and our daughter, Amadea

Acknowledgement

First and foremost, I would like to express my thanks to Dr. Gary Page for having presented me with the opportunity to carry out this project. I would also like to express a token of my appreciation for his supervision, discussion and encouragement throughout the duration of the study and I hope this is a reflection of the knowledge he has confered to me.

My postgraduate studies would not have been possible without generous financial support by the Department of Aeronautical and Automotive Enginnering.

I would like to thank Prof. James McGuirk who has advised on a simple second order approach for the convective scheme and also provided additional grant for the completion of the research. Many thanks also to: Mr Lance Portas who has taught me to use LaTeX 2_ε, Matlab and provided some discussion on Fortran Programming, Drs. Tang Gefeng, Yang Zhiyin and Jiang Dachun whose discussion has provide insights to turbulent flow simulation, also the METIS team of the University of Minnesota, Department of Computer Science team who have made the partitioning library publicly available.

Finally, I would like to express my gratitude to my parents for their unconditional support throughout my years in education. This thesis would not be possible without them.

Contents

List of Figures

Nomenclature

The symbols used in the present thesis are based on what is commonly found in the literature. Whenever possible, all scalar quantities are printed as normal typeface characters whereas the tensor (including vector) quantities are printed as bold typeface characters. Consequently, elements of a tensor are written as scalars.

Latin Characters

a	speed of sound
C_{DES}	constant for DES model
C_S	constant for Smagorinsky model
C_W	constant for WALE model
C_{sth}	constant for Sutherland viscous law, defined by 2.14
c_{b1}	a constant in Spalart-Allmaras model
c_{b2}	a constant in Spalart-Allmaras model
c_{v1}	a constant in Spalart-Allmaras model
c_{w1}	a constant in Spalart-Allmaras model
c_{w2}	a constant in Spalart-Allmaras model
c_{w3}	a constant in Spalart-Allmaras model
c_p	specific heat at constant pressure
c_v	specific heat at constant volume
c	aerofoil chord length
D	nozzle jet diameter
d	nearest distance to a solid wall
E	specific total internal energy of the fluid, defined by 2.8
F	convective flux vector, defined by 2.3
f	specific external body force, $\mathbf{f} = f_x\mathbf{i} + f_y\mathbf{j} + f_z\mathbf{k}$

$fv1$	viscosity function in Spalart-Allmaras model, defined by A.13
$fv2$	viscosity function in Spalart-Allmaras model, defined by A.19
fw	wall function in Spalart-Allmaras model, defined by A.20
$\mathbf{G}$	diffusive flux vector, defined by 2.4
g	a parameter in Spalart-Allmaras model, defined by A.21
$\mathbf{i}$	Cartesian unit vector along x-direction
$\mathbf{j}$	Cartesian unit vector along x-direction
j	imaginary component of a complex number, $j = \sqrt{-1}$
$\mathbf{k}$	Cartesian unit vector along x-direction
k	heat conductivity
K	an adjustable constant for MUSCL extrapolation
$\mathcal{K}$	an adjustable constant to tune Venkatakrishnan limiter
L	characteristic mesh spacing
M	Mach number
N	total number of samples for ensemble averaging
$\mathbf{n}$	normal vector, $\mathbf{n} = n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k}$, $\mathbf{n} = S\hat{\mathbf{n}}$
$\hat{\mathbf{n}}$	unit normal vector, $\hat{\mathbf{n}} = \hat{n}_x\mathbf{i} + \hat{n}_y\mathbf{j} + \hat{n}_z\mathbf{k}$
Pr	Prandtl number
Pr_t	turbulent Prandtl number
p	pressure
$\dot{\mathbf{q}}$	heat flux vector
$\mathbf{Q}$	conservative state vector, defined by 2.2
Re_D	Reynolds number based on nozzle jet diameter
R	nozzle exit radius
	pipe radius
$\mathbf{r}$	Cartesian position vector, $\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$
$\hat{\mathbf{r}}$	unit cartesian position vector, $\hat{\mathbf{r}} = \hat{r}_x\mathbf{i} + \hat{r}_y\mathbf{j} + \hat{r}_z\mathbf{k}$, $\mathbf{r} = \lvert\mathbf{r}\rvert\hat{\mathbf{n}}$
$\mathfrak{R}$	specific gas constant, $\mathfrak{R} = c_p - c_v$
$\mathbf{S}$	source term vector
S	surface area of the computational control volume
	component of velocity strain
S_{sth}	constant for Suther land viscous law, defined by 2.14
$\mathcal{S}^d$	component traceless symmetric part of velocity gradient tensor
T	temperature
	time period
t	time

Un	velocity component along unit normal direction, also: contravariant velocity, $Un = \mathbf{u} \cdot \hat{\mathbf{n}}$
$\mathbf{u}$	velocity vector, $\mathbf{u} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}$
$\hat{\mathbf{u}}$	unit velocity vector
u	Cartesian velocity component along x-direction, also u_1
u_τ	friction velocity on a solid wall
v	Cartesian velocity component along y-direction, also u_2
V	volume of the computational control volume
w	Cartesian velocity component along z-direction, also u_3
x	a spatial direction in Cartesian Coordinate Sytem, also: x_1
y	a spatial direction in Cartesian Coordinate Sytem, also: x_2
z	a spatial direction in Cartesian Coordinate Sytem, also: x_3

Greek Characters

ϵ	a small number to avoid division by zero
ε	low cut-off value for the numerical dissipation switch, eq. 4.19
Γ	computational control volume
$\partial\Gamma$	surface of the computational control volume
γ	ratio of specific heats, $\gamma = c_p/c_v$
Δ	specific filter size
δ_{ij}	Krönecker delta function, $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$
ζ	a local coordinate variable
η	a local coordinate variable
θ	turbulent heat conduction
κ	von Kármán constant
λ	eigen value of convective flux matrix
μ	molecular dynamic viscosity
μ_t	turbulent eddy viscosity based on Boussinesq hypothesis
o	a Venkatatkrishnan constant; scaled with the local cell size
ρ	density of the fluid
σ	turbulent Prandtl number in Spalart-Allmaras model
τ	Cartesian component of the shear stress
τ_w	shear stress on a solid wall

Φ	numerical dissipation switch, defined by eq. 4.19
φ	a scalar variable
χ	turbulent to molecular viscosity ratio,defined by eq.A.14
ψ	gradient limiter
Ω	magnitude of vorticity
$\tilde{\Omega}$	modified vorticity in Spalart-Allmaras model, defined by eq. A.18

Subscripts

x	Cartersian x component of a vector, $\partial/\partial x$
y	Cartersian y component of a vector, $\partial/\partial y$
z	Cartersian z component of a vector, $\partial/\partial z$
i	a Cartesian direction index
i	a nodal index
j	a Cartesian index
j	a nodal index
k	a Cartesian index
k	a nodal index
∞	evaluated at free stream state
1/2	evaluated at half centre line velocity state
wall	evaluated at wall state

Operators

$\bar{\cdot}$	RANS and LES filtered quantities
$\tilde{\cdot}$	RANS and LES Favre filtered quantities
∇	gradient operator, $\nabla = \partial/\partial x\mathbf{i} + \partial/\partial y\mathbf{j} + \partial/\partial z\mathbf{k}$
$G(\cdot)$	convolution function for filtering

Acronyms

AGARD	Advisory Group for Aerospace Research and Development
AIAA	American Institute for Aeronautics and Astronautics
CERFACS	Centre Europeen de Recherche et de Formation Avancée en Calcul Scientifique
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
DES	Detached Eddy Simulation
DLR	Deutsche Forschungsanstalt für Luft- und Raumfahrt
DRA	Defence Research Agency, currently: Qinetiq
LES	Large-Eddy Simulation
MILES	Monotonically Integrated Large-Eddy Simulation
MPI	Message Passing Interface
MUSCL	Monotone Upstream-Centred Schemes for Conservation Laws
NASA	National Aeronautics and Space Administration
ONERA	Office National d'Etudes et de Recherches Aerospatiales
QDNS	Quasi Direct Numerical Simulation
RANS	Reynolds-Averaged Navier-Stokes Equations
SGS	Sub-Grid Scale
TVD	Total Variation Diminishing
STOVL	Short Take-Off and/or Vertical Landing
WALE	Wall-Adapting Local Eddy-Viscosity

Chapter 1

Introduction

1.1 Motivation

Turbulence is a natural phenomenon that occurs in many practical flows of aeronautical engineering interest. Thus its prediction through a simulation method always plays an important role in aerodynamic design. While Computational Fluid Dynamics (CFD) has been established as an important tool in flow simulation, its success in dealing with turbulent flows has largely been dependent on turbulence simulation techniques. These simulation techniques can be classified into two approaches (see McComb's review [76] as an example), those that decompose the flow into a mean steady flow and a fluctuating turbulence component and those that try to resolve numerically the intermittent vortical structures in the fluctuating flow as much as possible. For ease of argument, the second approach is called the unsteady turbulence simulation. Reynolds-Averaged Navier-Stokes (RANS) turbulent simulation uses the first approach by introducing either an ensemble averaging or a time averaging to obtain a steady mean flow. The influence of the turbulence – in the form of normal and shear Reynolds stresses – on the mean flow is obtained from a semi-empirical turbulence model that has been built using statistical methods such as correlations, to take into account the physics of turbulence structures. The second approach includes Direct Numerical Simulation (DNS) that does not require any model and Large Eddy Simulation (LES) that is often performed with only a very simple turbulence model [34, 35]. This approach relies heavily on the non-linear numerical schemes to simulate the time-varying nonlinear interaction among the turbulence scales. In some techniques, known either as monotonically Integrated LES (MILES) or embedded LES, the turbulence model is even provided solely by the

artificial dissipation of the numerics [16, 28].

The experience in the past decades, mainly from the late 1980s onward, has shown that a CFD solution with a RANS turbulence model is reasonably accurate in predicting skin friction and pressure distribution of an attached flow over a complex geometry such as an aircraft in cruise configuration, but it suffers quite badly in separated flows [69, 114]. The situation is generally worse for flows where the prediction of particular turbulence properties such as the root-mean square (rms) of individual fluctuations or Reynolds stresses are the main concern. Even a simple prototype problem such as a round free jet flow demonstrates this limitation very clearly [7]. This is because most RANS models assume that the turbulence eddies are isotropic [69], in spite of the fact that the large scale turbulence eddies are anisotropic and not linearly proportional to the Reynolds stresses. Furthermore, the anisotropy and non-linearity exhibited by the larger eddies are problem dependent. These limitations prove to be unsatisfactory for many modern design issues. Having been established as the most cost efficient tool for flow prediction, CFD is expected to be able to cope with these flows, in which the turbulent structures play an important role. Thus, the implementation of the unsteady turbulence simulation in practical CFD is a matter of necessity.

An example of a flow problem where complex geometry as well as a complex flowfield with significant separation are involved is the Short Take-Off and Vertical Landing (STOVL) aircraft in ground effect. Figure 1.1 shows a typical scenario for a contemporary STOVL aircraft in vertical flight close to the ground with highly under-expanded supersonic jet from the rear engine nozzle impinging on the ground. Among other things, the jet impingement raises acoustic related problems ranging from airframe fatigue to ground crew health issues. When the nozzle is very close to the ground, the supersonic jet impingement becomes unsteady, which make it difficult for RANS to give an accurate prediction. Henderson et al. show that the ground impingement distance affects the jet acoustics [51]. Even when the impingement is steady, Parneix et al. have shown that unless the RANS turbulence model takes into account Reynolds stress anisotropy, the computational result will never be accurate [88]. In ref. [88] Parneix employed Durbin's v2f model [31] which involves as many as five partial differential equations rather than the one or two that are typically found in a simple RANS model. Another issue depicted in Fig. 1.1 is a possible engine surge due to the re-ingestion of the hot ground sheet (jet) flow caused by the ground vortex that is formed by interaction between the ground jet and the incoming headwind [86]. The prediction of hot gas ingestion using RANS has also met with limited success, mainly because of the RANS limitation in predicting the

separation that marks the beginning of the ground vortex [69].

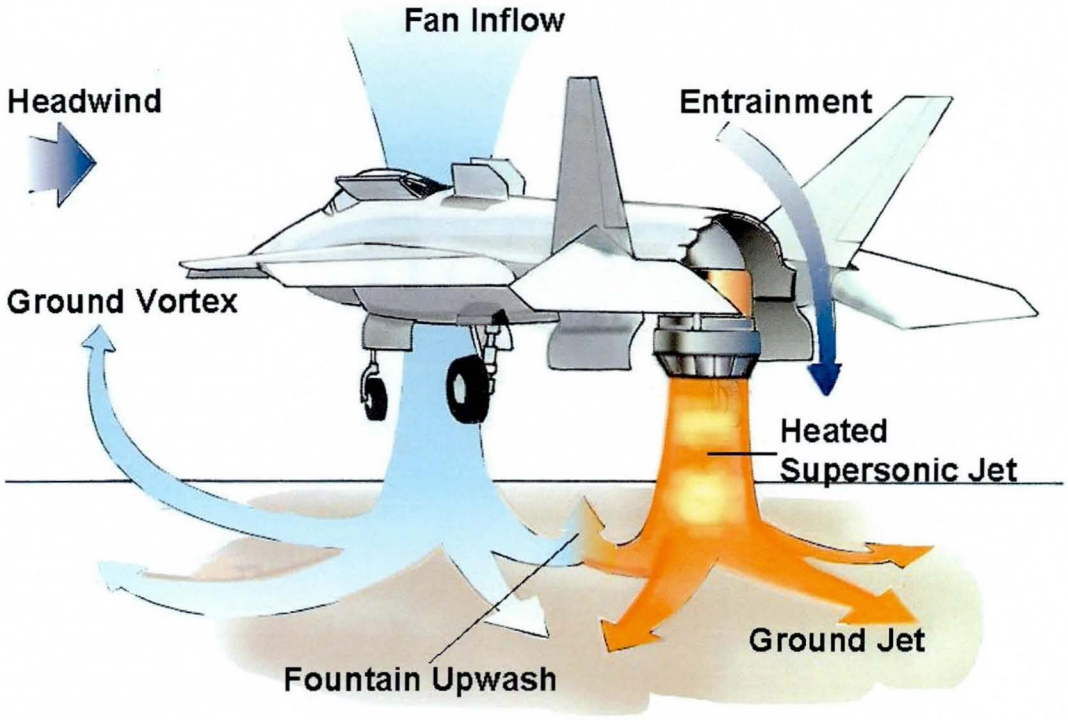


Figure 1.1: Complex flow around a STOVL aircraft in ground effect.

Another example of a practical engineering problem where accurate turbulent flow information is needed is turbulent mixing in a combustion chamber of an aero-engine, which has very complex geometry. Eventually such a mixing problem determines the quantity of pollutants emitted by an engine or noise generated by a jet, which is important in assessing the environmental impact of human flight.

In Direct Numerical Simulation (DNS) one resolves all of the turbulent eddies that exist in the flow. Thus, DNS is essentially free from any empirical model. To capture the complete eddy spectrum, the mesh resolution must be able to resolve the smallest eddies. For turbulent flow through a pipe of diameter D, this means that the number of cells in the computational mesh is proportional to $Re_D^{9/4}$, where Re_D is the Reynolds number based on pipe diameter. This mesh requirement means that DNS is too expensive for any practical flow. It is generally accepted that DNS is currently only useful for the investigation of the physics of turbulence [117].

Large-Eddy Simulation (LES) can be seen as a compromise between RANS and DNS. Here, the Navier-Stokes equations are filtered spatially to separate the large and small eddies. The large eddies, which are flow dependent, are directly resolved while the small ones, which are considered universal are modelled. Thus, the large scale turbulent unsteadiness is retained by the method, as in DNS, whereas the small scale relies more on empiricism like RANS. LES is often seen to be attractive due to its ability to give a more accurate result than RANS for a wider range of turbulence flow at a fraction of the cost of DNS. Hence LES could be very useful for flow prediction of STOVL aircraft in ground effect as well as the turbulent mixing in a combustion chamber.

Theoretically, the spatial filter size, Δ, in LES does not have to be dependent on the mesh spacing size h, other than $\Delta \geq h$ since filtering can only be done on the resolved flow-field which obviously depends on the cell size [36]. However, the two are often closely linked in practice [134]. In this case the length scale of the modelled eddies must be smaller than h. Hence the model is widely known as a sub-grid scale (SGS) model. This close relation between cell size and resolved turbulent scale means that LES requires a large number of cells, especially when a solid wall is involved. Furthermore, the unsteady nature of turbulent fluctuations needs a long time integration to gather the statistics. It must be emphasised that for wall bounded problems LES is still considerably more expensive than RANS.

1.2 Aim

The aim of the present research is to develop a practical tool to perform Large-Eddy Simulation for compressible turbulent flows involving complex geometries such as are typically found in industrial aeronautical engineering problems. Nevertheless, the resulting LES code is not aimed to replace RANS simulations especially on attached flow problems, as this still provides the most effective and efficient computational method for predicting skin friction and pressure distribution as typically required by engineers.

1.3 Large Eddy Simulation for Engineering Flows

It is generally acknowledged that the application of LES to a turbulent flow was pioneered by Smagorinsky who computed an atmospheric flow problem in 1963 [112]. LES application to internal flows of engineering interest followed shortly thereafter:

Deardroff's computation of a channel flow problem in 1970 [25], and Schumann's work in turbulent flow through an annulus which served as a very simplified model of flow in a modern reactor fuel rod assembly [103]. However, only recent developments in computer technology have made LES a more realistic tool for predicting practical engineering flows.

Early LES works employed spectral methods to obtain high spatial resolution on a relatively coarse mesh [34]. Unfortunately, this efficient method needs a uniform structured rectilinear mesh, which makes it awkward for complex geometries of practical engineering problems. Furthermore, difficulties in defining boundary conditions, means that most spectral method LES computations were carried out for problems with periodic boundary conditions [34]. During the 1970s Deardroff [25] and Schumann [103] pioneered the application of Finite Volume techniques, which had previously proved to be versatile for steady flow CFD [49] to address this limitation. Apart from its geometric flexibility, the finite volume discretisation can be seen as an implicit filtering of the LES equation [134].

Together with finite volume discretisation on structured meshes, the second order central difference scheme is widely used in current finite volume LES solvers. (See review by Rodi et al. [96], for instance.) Recently, a study by Ghosal has indicated that the dispersion error of the second order central spatial discretisation may mask the sub-grid scale model [45]. This gave rise to the suggestion by Lele that higher order compact schemes with spectral like resolution [67] may offer superior performance (see refs. [14, 113, 128, 129] for example). It must be noted that higher order discretisation is known to be unable to conserve both the momentum and energy at once [80].

1.3.1 Computational Mesh

CFD performs mathematical operations on a set of discrete points that are connected to one another to form a mesh or grid that covers the whole computational domain. The node connectivity, which is referred to as the mesh structure or simply structure, is important as calculation on each node requires information from its neighbours. This structure plays an important role in mesh topology, which affects the discretisation of the governing equation of the fluid flow for a given problem geometry. Without going into detail, there are two types of structure: structured and unstructured meshes.

Structured meshes – as the name implies – arrange the nodes along a coordinate system so that these can be addressed by regular indexing, for instance node i is always placed between $i-1$ and $i+1$. The regular indexing makes each node directly accessible. When implemented in a CFD code this results in low memory usage and good CPU cache efficiency. This regular arrangement may be easily linked by Taylor Series expansion from which the finite difference and finite volume techniques have been derived. On the other hand such regularity leads to a fixed number of neighbours for each node: two in one dimension, four in two dimensions and six in three dimensions. This non-local nature restricts the flexibility of the mesh in discretising a flow domain with complex geometry.

In general, there are two types of structured mesh, namely a rectilinear or curvilinear mesh. A rectilinear mesh is obtained when the coordinate system is defined by a set of orthogonal straight lines, usually along Cartesian axis. Due to its simple construction, rectilinear meshes are found in many early CFD calculation such as that of Harlow and Welch [49]. As well as bringing simplicity to the mesh, the orthogonality supports flow decomposition along two or three linearly independent directions, which minimises discretisation errors due to cross derivatives. This means that a rectilinear mesh is particularly suitable for spectral methods as well as finite difference and finite volume. However, such an advantage quickly diminishes when a problem with a non-rectangular flow domain is encountered. Curvilinear meshes addresses the geometric limitation of a rectilinear mesh by having a coordinate system that is defined along a boundary curve, i.e. a boundary fitted mesh. Unlike a rectilinear mesh that is always orthogonal, such a mesh requires smoothing techniques to achieve near orthogonality, i.e. avoiding skewed and distorted cells as much as possible [125]. To handle flow domains with more complex geometry, the domain is divided into several topologically hexahedral blocks whose connection to one another is far more flexible than that of cells in a structured mesh. Each block is then discretised using curvilinear structured meshes. As demonstrated by Fig. 1.2, a multiblock curvilinear mesh makes structured discretisation on a relatively complex geometry easier than its single block counterpart. Despite the greater flexibility, multi block mesh generation is still time-consuming and the cells in the interface regions between blocks can be highly skewed and distorted.

Unstructured meshes are specifically designed to discretise complex flow domains by having nodes whose number of neighbours is not fixed. Hence the mesh has local properties as each cell can be placed independently. The local nature of the mesh means

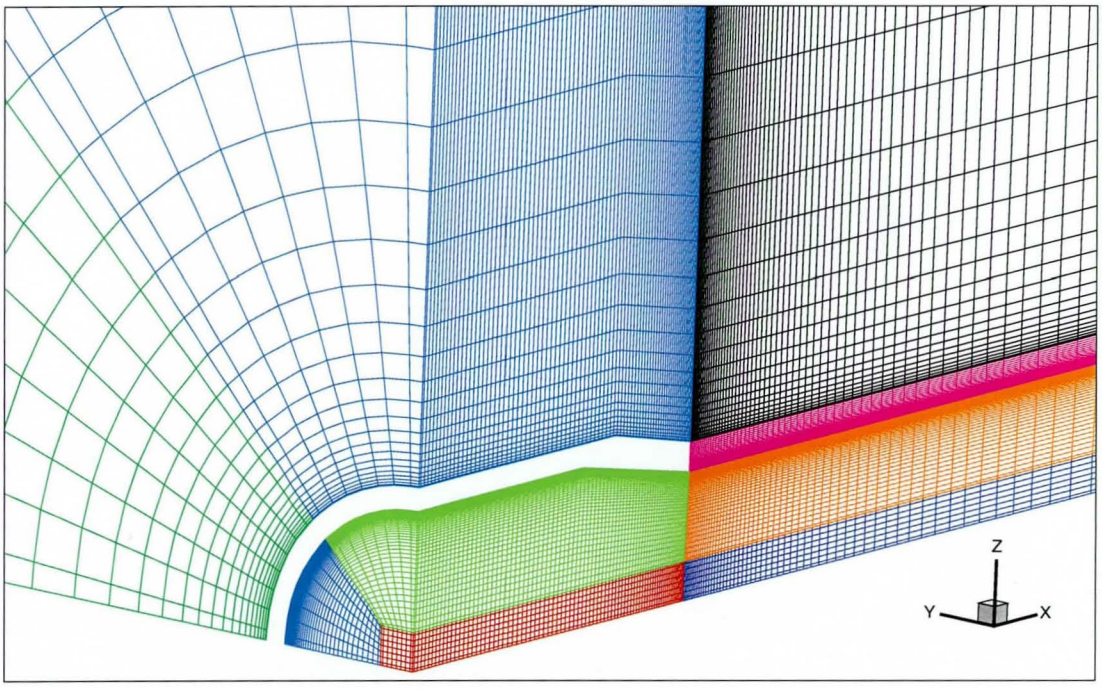


Figure 1.2: Multi block mesh of a round jet with nozzle. Different colours are used to show different blocks.

that a complex domain can be discretised with relative ease, which makes automatic mesh generation possible. Even when the geometry is simple, an unstructured mesh allows more efficient node clustering. Theoretically any cell shape can be used to discretise a problem geometry using an unstructured mesh. In practice, geometric simplices have proved to be the most widely used since these element types can easily tessellate any complex arbitrary shapes. Thus, triangles are often used for the surfaces of a domain boundary and tetrahedra for the domain's volume. The less flexible prismatic cell, on the other hand, is more suitable for regions with high velocity gradients such as the viscous region near a solid wall boundary and a shear layer [101]. When mixed element types are employed in a flow solution algorithm (see Fig. 1.3), it is desirable to avoid lists of element type – this results in a mesh-transparent algorithm.

An unstructured mesh requires an explicit list of node to node connectivity to enable indirect memory addressing since it lacks any regular indexing pattern that provides implicit connectivity information in a structured mesh. As the neighbouring node number is not generally held in a memory address that is close, these are not normally cached at the same time resulting in longer CPU time to collect the data [71]. Although multi-block curvilinear meshes employ the same technique for block to block communications,

there are normally only very few blocks in a computational domain in comparison to the number of cells in each block. However, as computer speed grows, computation cost becomes much cheaper than the more labour intensive pre-processing.

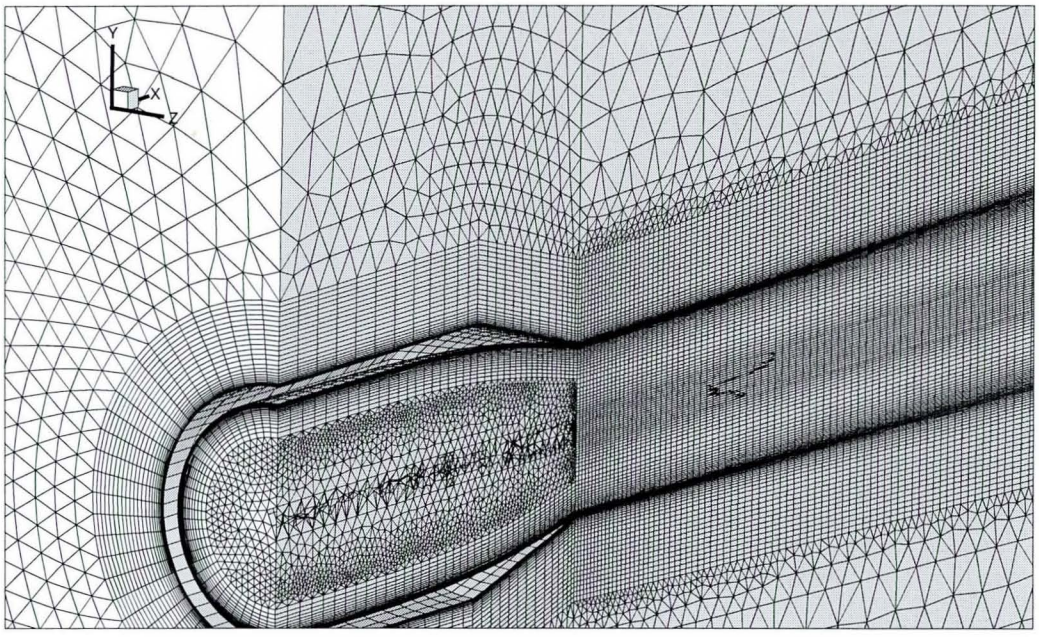


Figure 1.3: Unstructured mesh of a round jet with a nozzle. The irregular triangular shapes inside the nozzle are tetrahedra that have been sliced by the centre plane to show this image

1.3.2 LES on Unstructured Meshes

One of the most important aspects of implementing LES on more realistic engineering flow problems is the ability to handle complex geometries. As in RANS CFD, this can be addressed by having either a structured multi-block curvilinear mesh or an unstructured mesh. The work of Tang et al. [122] is an example of a successful application of multiblock LES on a lean premixed prevapourised combustor, which also highlight the difficulties in generating a high quality structured multiblock curvilinear mesh. Despite being computationally more expensive, the unstructured mesh is the preferred choice in this thesis since it offers a higher degree of flexibility that allows automatic mesh generation. The following discusses some of the issues of using an unstructured mesh for LES.

Numerical Dissipation is an important issue in a numerical schemes for LES because it can interfere with the sub-grid scale model which is designed to simulate turbulence dissipation by small eddies. The dissipation can originate from the leading term of the truncation error that is associated with upwind discretisation. In the worst scenario, Beaudan and Moin [12] reported that the numerical dissipation can mask the sub-grid scale model contribution. Hence centred numerical schemes, whose truncation error does not behave as numerical dissipation, are usually preferred. Unfortunately there are some instances where upwind schemes are necessary, such as compressible flows with shock waves. Here, the dissipation is required to damp out non-physical fluctuations around the shock that sometimes cause the the solution to diverge [1]. For this reason LES of compressible jets [14, 26, 128] have employed a filter that is designed to remove such oscillations (see Lele [67] and Yee et al. [138]).

State of the art unstructured finite volume solvers have been proved to be successful in RANS computations [71]. Flux evaluation along the direction normal to a face such as used in the finite volume technique creates artificial cross dispersion when a flow feature, such as shock, is not aligned to the face, irrespective of whether a centred or upwind scheme is employed. A properly designed structured mesh on a simple geometry can avoid this cross dispersion. On the other hand, polygonal cells of an unstructured mesh do not generally allow such alignment, unless a specific cell shape – cuboid – has been employed.

The effect of the mesh misalignment in an LES computation is illustrated by comparing computations of vortex shedding behind a cylinder using the 5^{th} order upwind convective scheme reported by Beaudan and Moin [12] with that of Mittal and Moin [78]. The former employed an O-mesh topology whereas the later employed a C -mesh topology. Whilst the C topology mesh was aligned with the flow in the wake region, the O topology was not. Beaudan and Moin reported that the numerical error was so large that the wake virtually became independent of Reynolds number. On the other hand, Mittal and Moin only noticed that the cut-off wave number of the energy spectrum had been reduced by the numerical error whereas the averaged Reynolds stresses in the wake were not seriously affected. Simons and Pletcher [109] carried out a similar comparison for homogeneous turbulence decay using two uniform meshes, consisting of hexahedral and tetrahedral cells, respectively. Unfortunately this investigation has not been useful since different numerical schemes were applied to each mesh. Whilst a centred convective numerical scheme was employed for the first mesh, an upwind scheme was employed on the other. As a result, the dissipation that came from the cell geometry could not

be analysed in isolation from the upwinding dissipation.

The simplest strategy that can be applied to deal with the inherent numerical dissipation in the numerical schemes is by introducing a scaling factor. If this is constant throughout the flow-field, as used by Bui [17] and Cammari and Salvetti [18], the scaling compromises the scheme's ability to dealing with shocks found in high speed flows. To address this shortcoming Ducros et al. [29] proposed a variable scaling factor, whose value depends on the velocity divergence and the vorticity to retain numerical dissipation near a shock wave for stability whilst reducing it in regions of shear. This method is much simpler than the filtering technique that was suggested by Yee et al. [138]. It should be noted that neither the low speed computation of Jansen et al. [58], nor the high speed computation of Knight et al. [62] reported any special treatment to control the amount of numerical dissipation in their schemes. Hence the inaccuracies due to excessive dissipation reported in those works may be alleviated by minimising the numerical dissipation.

In a RANS turbulent flow computation using unstructured meshes for a three dimensional problem, prismatic cells are often employed. Such a strategy has lead to the development of a mesh-transparent algorithm capable of handling virtually any cell geometry [50, 79, 102]. For the remainder of this thesis, such a mesh will be called a hybrid mesh. There is no reason why this strategy cannot be implemented in LES applications. Such implementation is very attractive as the numerical dissipation due to cell geometry can be reduced in areas where high shear due to turbulence exists by implementing prismatic cells in this region. In the meantime tetrahedral cells can be applied in other flow regions to ensure that a complex problem geometry can be handled correctly. It must be noticed that rather than trying to introduce complicated functions to minimise tetrahedral dissipation, this strategy simply opts to employ prismatic cells which are inherently less dissipative. The strategy also offers potential savings on the number of cells as more flexible mesh coarsening towards the far-field boundary can be achieved without sacrificing resolution in the region of interest. Despite this clear advantages, reference to this methodology in the literature is very rare. The author is only aware of two LES computation on hybrid meshes, namely turbulent flow in a round pipe [30, 84] and a combustion problem [105]. Both of these work have been performed at CERFACS.

The Fluctuation Splitting technique has been developed since the mid 1990s through a reinterpretation of the Roe Flux Splitting scheme [97], the aim being to overcome the

shortcomings of an unstructured finite volume solver by offering lower operating count and reduced numerical diffusion [108]. The first is achieved by avoiding face flux reconstruction on the cell faces that characterise the finite volume approach. The later is achieved through a truly multidimensional operator as opposed to locally one dimensional numerical dissipation of the convective scheme as typically done in finite volume methods to provide stability. Even though this technique shows marked improvement in accuracy due to lower numerical diffusion over unstructured finite volume method in linear problems, the same cannot be said for a system of non-linear equations such as the Navier-Stokes equations [136]. Unlike the finite volume technique that guarantees conservation of property through flux balancing, Fluctuation Splitting behaves like a Finite element method that may not conserve the flow properties [136]. There is no known LES application of this technique. As a number of researchers (e.g. [63, 80]) stipulate the conservative property as an important ingredient for LES, fluctuation splitting may not be suitable for LES.

Higher Order Schemes are generally a desirable property in CFD as they usually lead to smaller error than lower order schemes [45]. For ease of discussion, any discretisation scheme that is higher than second order is called a Higher Order Scheme.

In finite difference and finite volume algorithms a high order scheme is achieved by increasing the size of the stencil. Lele introduced a numerical scheme based on Padé (Hermitian) interpolation to address this problem [67]. In a structured mesh, such a method has enabled 4^{th} order spatial discretisation to be achieved by employing a stencil with two cells rather than four as needed for traditional schemes. Moreover, the Lele scheme also exhibits better resolution than the traditional discretisation scheme of the same order. Such a compact formulation can be particularly useful for unstructured meshes since a large stencil requires a complicated connectivity data structure. In spite of this, its implementation on unstructured meshes is rarely mentioned in the literature. The author is only aware of a preliminary study of Zingg and Lomax [144].

In the finite element method a higher order scheme is achieved by introducing a higher order trial function, i.e. a non-linear element is used. One method that has been gaining popularity in CFD is the Spectral Element technique (Discontinuous Galerkin) [21], which is essentially a finite volume based finite element algorithm that allows any order of spatial discretisation to be achieved by using a single element stencil. It is currently seen as one of the possible viable alternative methods to the traditional finite volume and finite element approach for LES on unstructured grids [60]. The

non linear element allows for a coarser mesh than typically used in CFD. Bassi and Rebay [11] demonstrated that the Discontinuous Galerkin method can give accurate results with a very coarse mesh.

Ideally the implementation of a high-order scheme for LES is very desirable. Unfortunately, the cost of its implementation has proved to be quite expensive. For an unstructured mesh algorithm, the large stencil translates to more complex data structures that lead to longer execution time, whereas a compact stencil such as the Discontinuous Galerkin requires a larger number of operation count that makes it more expensive than a second order finite volume unstructured solver. Although the Discontinuous Galerkin method allows for a very coarse mesh, its implementation for LES would be limited by the resolution of the smallest resolvable turbulent eddies. Hence the saving that should result from a smaller number of cells may not materialise. Even for structured mesh LES, whose higher order extension is quite straight forward, one rarely implements numerical schemes that are higher than 4^{th} order accuracy. The author believes that a properly designed sub-grid scale (SGS) model should take care of the truncation error. Such a pragmatic approach is taken here for two reasons. Firstly, to keep the operation count low in response to the LES requirement for long time integration over a large number of cells so that computation of practical engineering flow problems are still within reach. Secondly, there is no evidence that LES computation using second order finite volume method leads to erroneous results. Furthermore, the second order scheme is less suceptible to numerical oscillations that often plague the higher order scheme.

1.3.3 Sub-Grid Scale Model

On the sub-grid scale modelling front, eddy viscosity models based on the Bousinesq hypothesis remains the most widely used. The simplest form is the standard Smagorinsky model that is very similar to the algebraic Prandtl mixing length RANS model. The SGS viscosity is calculated from the square of a length scale and velocity strain. The length scale is found from a characteristic filter length and a model constant, C_s, that is assumed to be uniform throughout the flow-field. Even though such an assumption has lead to a robust model since a positive value for C_s guarantees the model providing dissipation to the numerical scheme, it is generally not correct. To rectify this Germano et al. [44] introduced a dynamic procedure to calculate C_s by employing a local test filter. Thus C_s varies from point to point in a turbulent flow-field. Jansen [58] and

Vasilyev [131] and his co-workers have developed a dynamic procedure for an unstructured mesh. While the dynamic procedure has improved the Smagorinsky SGS model significantly it increases the operation count dramatically as well as making the computation less robust [96] as C_s can become negative. Although this can be interpreted as back-scattering which actually happens to a small amount in the physics of turbulent flow, the negative values sometimes become too large. This is not only physically incorrect as it represents large amount of back-scattering but also causes the solution to diverge. Rodi et al. [96] have noted that there are two ways commonly used to remedy this problem. One is to introduce a filter to clip the negative value. The other is to perform some averaging along the direction where the flow is expected to be homogeneous such as utilised originally by Germano et al. [44]. However, the second remedy is only applicable to certain type of flow such as a channel flow or wing with infinitely long span (i.e. statistically 2D problems), which do not represent typical engineering flow problems.

The Scale Similarity model (mixed scale model) which was pioneered by Bardina et. al. [6] is another approach to SGS modelling that is based on the interaction around the cut-off frequency between the large and small scale eddies. This model allows energy back scatter but it does not normally produce enough dissipation and is therefore not robust [133]. Vreman et al. [133] showed that a hybrid of Scale Similarity and standard Smagorinsky model provides a good compromise between computational cost and accuracy. Rodi et al. [96] also mentioned this hybrid approach as a third way to remedy problems with the dynamic procedure. The deficiency with the Bardina model seems to have been addressed by later models such as that of Ta Phuoc Loc and Sagaut [99] and the structure function model of Métais and Lesieur [77], which is based on the interaction of kinetic energy at the cut-off frequency.

A rather different approach to eddy viscosity SGS modelling is to employ the turbulent kinetic energy of the small eddies rather than resolved velocity strain to calculate the SGS viscosity. Just as in RANS, the kinetic energy evolution is calculated using a partial differential equation that is formed by modifying the Navier-Stokes equation [76]. The earliest model based on this method was developed by Yoshizawa and Horiuti [142], Horiuti [54] and Yoshizawa [140]. Recently, a dynamic procedure version of this model has been employed by Siniša and Davidson [110].

SGS models based on the transport equation of the Reynolds stresses have also been suggested by Fureby et al. [41] and Carati and Wray [20]. Computationally this model

is very expensive since one has to calculate a set of coupled partial differential equations just like in RANS based on the Reynolds Stress model. Unlike the previous models, this type of model does not assume the SGS viscosity to be isotropic. The underlying motive is that the isotropy assumption for an existing SGS model can be violated by poor mesh resolution or even an anisotropic grid such as often seen in LES calculations. Fureby and Grinstein [39] and Drikakis [28] have shown that similar properties are also exhibited by embedded LES (also called MILES – Monotonically Integrated Large-Eddy Simulation), which was pioneered by Boris et al. [16]. Numerical dissipation, which is inherent in the convective numerical scheme, acts as a non linear sub-grid scale model in embedded LES rather than an explicit SGS model. It must be noted that MILES is still considered controversial as Garnier et al. [43] found that none of the upwind schemes they investigated actually mimic the behaviour of the Smagorinsky model in a number of homogeneous isotropic turbulent flows. However, such a discrepancy might be acceptable since rigorous mathematical analysis performed by Fureby and Grinstein [39] demonstrated that, unlike embedded LES, the existing eddy viscosity SGS model such as the standard Smagorinsky actually lacks terms associated with high frequency dissipation that arise in mathematical analysis of the LES governing equations. Since this issue is beyond the scope of the present work, it is not discussed in this thesis any further.

1.3.4 Near wall turbulence

It has been mentioned briefly earlier that most practical engineering flow problems – particularly in aerospace applications – involve the presence of a solid wall. Besides preventing the rapid growth of large scale turbulent eddies, the wall also reduces the turbulent intensities in comparison to levels found in free shear flows. Turbulence is generated by instability in the very thin laminar sub layer near the wall, where short vortex filaments, called 'splats' or 'streaks', are lifted up from the wall and then burst. To resolve a streak, a very fine mesh, often not much coarser than DNS, is needed. Although the additional nodes in itself is not a serious problem for present day high speed parallel computers, the small cell size in such a fine mesh dictates a smaller time step which increases the number of time steps by several orders of magnitude. Thus, wall-resolved LES is often called Quasi-DNS (QDNS) [114]. For some internal flows such as a combustor, the near-wall turbulence is relatively insignificant in comparison to that of free stream turbulence. Thus, the need for QDNS can be avoided. However for most external flows, such as flow over an aircraft wing, near wall turbulence is very

important in predicting transition, separation, and ultimately skin friction. Spalart argues in his rather controversial paper that for for this reason alone wall-resolved LES over aircraft components will not be practicable until 2045 [114]. Similar conclusions can be drawn from the LESFOIL project, which intended to perform LES over a high lift aerofoil near stall [23, 68].

The standard Smagorinsky model that relies on a constant value of C_s is found to give the most inaccurate representation of a solid wall, since the finite value of velocity strain on the wall surface would result in a finite value of SGS viscosity there that should be zero. Van Driest damping is often introduced to ensure that C_s and hence the SGS viscosity is zero on the wall. However, such treatment still requires a fine mesh to resolve the streaks. Furthermore, the adoption of an eddy viscosity model does not lead to correct near wall velocity behaviour [69, 84]. Near wall modelling has often been cited as a more practical way to resolve the near wall problem, even though it means more reliance on empiricism. The simplest method is to employ a log-law wall function [96], which is often used in RANS. Hence the first node from the wall can be placed quite far away as resolution of wall streaks becomes irrelevant. At first glance both of the two methods, i.e. Vand Driest Damping and log law wall function, seem to be quite straight forward as they only requires distance to the nearest wall in term of wall units, y^+. This is true for structured meshes as y^+ can be estimated fairly accurately along a coordinate direction. However, y^+ computation in unstructured meshes is rather cumbersome as it requires a dedicated data structure to pair each interior node to a node on the solid wall [48]. Nicoud and Ducros have addressed this problem by introducing WALE [84], an SGS model that includes a term to give the correct turbulence behaviour near a wall without explicitly computing y^+. WALE has also been shown to give correct near wall behaviour in ref. [84].

Another alternative that has been proposed is to use RANS, which is reasonably accurate for attached flow with mild separation, for the near wall region and LES in the outer turbulence region [5, 19, 24, 117, 123]. Although some RANS models require very close normal distance to the wall for the first point to resolve the laminar sub layer, they do not require the closely spaced nodes in the remaining directions that is needed by LES to resolve the streak. Hence a coarser mesh is possible by employing hybrid RANS/LES. Temmermann and Leschziner [123] have demonstrated that the RANS model actually responds relatively well to the LES fluctuation. Bagget [5] has demonstrated that in fact streaks are formed in the RANS region even though their sizes and spacing are not physically correct.

There are two approaches in a hybrid RANS/LES strategy that can be found in the literature. The first one is the zonal approach where the the computational mesh is divided into separate RANS and LES zones. Normally the interface is defined beforehand in an area where RANS and LES are expected to produce fairly similar turbulence quantities. [24, 123]. The second approach is to devise a single model that switches from RANS to LES as soon as the nodes position reaches a certain distance from the solid wall. Thus, a relatively smooth transition between RANS and LES is guaranteed [19, 123, 99] by the second approach, whereas the first approach sometimes produces a kink on the zone interface [24]. Among several model that belongs to the second approach, the Detached Eddy Simulation (DES) scheme of Spalart et al. [117] seems to be the most mature at the time of writing [81, 118]. DES relies on a single model – usually the one equation Spalart-Allmaras model [116] although a two equation model of Menter $SST - k - \omega$ has also been used [119] – this behaves either as a RANS turbulence model or an LES sub-grid scale model, depending on the cell size and its distance from the nearest wall. No explicit switching function is employed, this is provided by grid clustering [115] instead.

1.4 Objectives

Given the aim stated in section 1.2 and the summary of previous works outlined in section 1.3, the objectives of the present research are:

1. To assess the viability of implementing higher order convective schemes, especially compact Padé scheme on an unstructured mesh to minimise interference between the numerics and SGS model.

2. Development of a mesh-transparent numerical algorithm as a prediction tool suitable for three dimensional turbulent flow problems found in practical engineering application using RANS and LES on hybrid unstructured mesh.

3. Development of a portable parallel CFD code to take advantage of the current generation of parallel machines based on a distributed memory architecture.

4. To demonstrate the capability of the code written for objectives 2 and 3 in performing LES on compressible turbulent flows.

1.5 Outline of Thesis

The rest of the thesis is organised in the following chapters.

The governing equations for the filtered Navier-Stokes equations and the turbulence models are presented in chapter 2.

A preliminary study on the feasibility of the Compact Padé scheme for an unstructured mesh is presented in chapter 3. The study is focused on mathematical analysis using Fourier transform as well as numerical experiments on the linear Burger's equation.

The algorithm of the numerical scheme for the mesh transparent edge-based unstructured flow solver is described in chapter 4. The chapter also discuses parallelisation issues of the present code using the MPI library.

Chapter 5 is dedicated to verification of the basic steady solver. Two test cases are presented. The first is an inviscid flow over a generic wing-body configuration to assess the basic numerical scheme. The second is turbulent flow over an ONERA M6 wing designed to demonstrate the viscous part of the solver as well as addressing the Kutta condition problem that arises in inviscid flow problems.

Unsteady flow verification is presented in chapter 6. A two dimensional supersonic mixing layer is employed to assess the Navier-Stokes solver in dealing with unsteady flow without having to include a sub-grid scale model. A more thorough verification is done by performing an LES computation of a Mach 0.9 round free jet. The wall bounded flow performance is also assessed through a low Mach number fully developed turbulent pipe flow case.

Chapter 7 is dedicated to demonstration of the present method for a practical engineering flow, namely a preliminary LES of unsteady impingement of an under expanded jet.

Chapter 8 presents the conclusions and some recommended further works.

Chapter 2

Mathematical Model

The governing equations of fluid flow essentially consist of three conservation laws: the conservation of mass (continuity), the conservation of momentum as described by Newton's 2^{nd} law, and the conservation of energy. By applying the three conservation laws on a fluid continuum, one arrives at a set of equations usually called the Navier-Stokes equations. The continuum assumption has to satisfy one condition, namely, the molecular scale of the fluid should be far smaller than any practical length scale to measure it.

2.1 Navier-Stokes Equations

The integral form of the Navier-Stokes equations for a stationary control volume Γ enclosed by surface $\partial\Gamma$ can be written in vector form as follows:

$$\frac{\partial}{\partial t}\iiint_{\Gamma}\mathbf{Q}dV + \iint_{\partial\Gamma}\mathbf{F}(\mathbf{Q})\cdot\hat{\mathbf{n}}dS - \iint_{\partial\Gamma}\mathbf{G}(\mathbf{Q})\cdot\hat{\mathbf{n}}dS = \iiint_{\Gamma}\mathbf{S}dV \tag{2.1}$$

where $\hat{\mathbf{n}}$ is the outward pointing unit vector orthogonal to $\partial\Gamma$ whose Cartesian components are $\hat{n}_x$, $\hat{n}_y$ and $\hat{n}_z$, and t is time. The state vector $\mathbf{Q}$, convective flux vector $\mathbf{F}$ and diffusive flux vector $\mathbf{G}$ are defined by the following matrices.

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{2.2}$$

$$\mathbf{F}\cdot\hat{\mathbf{n}} = U_n \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho(E+p) \end{pmatrix} + p \begin{pmatrix} 0 \\ \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \\ 0 \end{pmatrix} \tag{2.3}$$

$$\mathbf{G}\cdot\hat{\mathbf{n}} = \begin{pmatrix} 0 \\ \hat{n}_x\tau_{xx} + \hat{n}_y\tau_{yx} + \hat{n}_z\tau_{zx} \\ \hat{n}_x\tau_{xy} + \hat{n}_y\tau_{yy} + \hat{n}_z\tau_{zy} \\ \hat{n}_x\tau_{xz} + \hat{n}_y\tau_{yz} + \hat{n}_z\tau_{zz} \\ \hat{n}_x b_x + \hat{n}_y b_y + \hat{n}_z b_z \end{pmatrix} \tag{2.4}$$

where ρ is the fluid density; u, v, w are velocity components in Cartesian coordinates x, y, z directions, respectively; p is pressure; E is the specific total internal energy. The contravariant velocity, U_n is defined as

$$U_n = u\hat{n}_x + v\hat{n}_y + w\hat{n}_z \tag{2.5}$$

This contravariant velocity component plays an important role in flux computation across the control volume that will be discussed in chapter 4.

In the absence of external forces and chemical reaction, the source term vector $\mathbf{S}$ is defined as

$$\mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.6}$$

When a resultant of external body forces per unit mass $\mathbf{f}$ acts on the flow in a non-conservative system, the source term in the momentum equation must be equal to $\rho\mathbf{f}$ while the source term in the energy equation describes the external work done by $\mathbf{f}$, hence the source term vector becomes

$$\mathbf{S} = \begin{pmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho f_z \\ \rho(f_x u + f_y v + f_z w) \end{pmatrix} \tag{2.7}$$

where f_x, f_y, and f_z are the Cartesian components of $\mathbf{f}$.

To provide a simple closure to the Navier-Stokes equation, the fluid is assumed to behave as a calorically perfect gas. Thus the total internal energy per unit volume can be defined as :

$$\rho E = \underbrace{\rho c_v T}_{internal\ \ energy} + \underbrace{\frac{\rho(u^2+v^2+w^2)}{2}}_{kinetic\ \ energy} \tag{2.8}$$

where T is the temperature. This assumption yields the well known ideal gas equation, namely $p = \rho \Re T$, where $\Re = c_p - c_v$; c_p is the specific heat capacity at constant pressure, and c_v is the specific heat capacity at constant volume. The two specific heat capacities are related by $\frac{c_p}{c_v} = \gamma$. Hence the total internal energy can be recast as a function of p and velocity components u, v, and w as follows

$$\rho E = \frac{p}{\gamma - 1} + \frac{\rho(u^2+v^2+w^2)}{2} \tag{2.9}$$

Components of b_{x_i} in the last element of vector $\mathbf{G}$ are defined using tensor notation with the summation convention implied as:

$$b_{x_i} = u_j \tau_{x_i x_j} - \dot{q}_{x_i} \tag{2.10}$$

where $i, j = 1, 2, 3$ indicate individual Cartesian coordinate directions. The heat transfer in the Cartesian coordinate system $\dot{q}_{x_i}$ follows Fourier law of conduction as:

$$\dot{q}_{x_i} = -k \frac{\partial T}{\partial x_i} \tag{2.11}$$

The heat conductivity, k can be modelled as

$$k = \frac{\mu c_p}{Pr} \tag{2.12}$$

where Pr is the non-dimensional Prandtl number.

For Newtonian fluids, such as air, the viscous stress components $\tau_{x_i x_j}$ in the diffusive flux $\mathbf{G}$ are defined as follows, using tensor notation with the summation convention implied.

$$\tau_{x_i x_j} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \delta_{ij} \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \tag{2.13}$$

The symbol δ_{ij} is the Krönecker delta function, which returns unity value when $i = j$ and 0 otherwise. The molecular viscosity, μ is a function of temperature that follows the Sutherland Law

$$\mu = \frac{C_{sth} T^{3/2}}{T + S_{sth}} \tag{2.14}$$

where C_{sth} and S_{sth} are the Sutherland constants.

Throughout the rest of the thesis, air is assumed to be the working fluid. Thus, the empirical values of the constants are given by table 2.1

constant	value	unit
γ	1.4	-
$\Re$	287	J/(kg K)
Pr	0.72	-
C_{sth}	1.458E-6	N.s/(m^{2}K$^{1/2}$)
S_{sth}	110.4	K

Table 2.1: Empirical constants for air.

2.1.1 Non-dimensionalisation

In some computations, it is more convenient to work in non-dimensional terms. For instance, in aerofoil applications the physical dimensions that come from wind tunnel measurements are typically normalised by the chord length. Thus, computational results that have been carried out in non-dimensional terms can be compared directly.

Non-dimensionalisation can be carried out easily by defining a set of reference quantities to scale the continuity, momentum and energy equations by $(\rho_{ref}u_{ref})^{-1}$, $\left(\rho_{ref}u_{ref}^2\right)^{-1}$, and $(\rho_{ref}u_{ref}\Re_{ref}T_{ref})^{-1}$, respectively. The reference quantities are $x_{ref}, \rho_{ref}, u_{ref}, T_{ref}, \Re_{ref}$ for length, density speed, temperature and gas constant, respectively. The dimensionless variables are then:

$$x_i^* = \frac{x_i}{x_{ref}} \qquad t^* = \frac{t}{x_{ref}/u_{ref}} \qquad \rho^* = \frac{\rho}{\rho_{ref}} \qquad u_i^* = \frac{u_i}{u_{ref}} \qquad p^* = \frac{p}{\rho_{ref}u_{ref}^2}$$
$$T^* = \frac{T}{T_{ref}} \qquad \Re^* = \frac{\Re}{\Re_{ref}} \qquad \mu^* = \frac{\mu}{\rho_{ref}u_{ref}x_{ref}}$$

It must be noted that strictly speaking, $\Re_{ref}$ can be derived from reference temperature and pressure, which in turn are functions of reference density and speed. Here, $\Re_{ref}$ is defined explicitly so that an arbitrary value for T_{ref} can be chosen while maintaining the simple scaling factor for the energy equation. Otherwise, a more complex scale based on $(T - T_0)/(T_1 - T_0)$ as non-dimensional temperature that involves two reference values, T_0 and T_1, must be used.

For the works in this thesis, the free stream density and sound speed are used as reference values. Consequently, the non-dimensional free stream velocity is the free stream Mach number, the non-dimensional free stream density is unity and the non-dimensional free stream pressure is $1/\gamma$. Since the non-dimensional free stream temperature is set to unity, $\Re_{ref}$ must be defined in such a way that $\Re^* = \frac{1}{\gamma}$ for consistency.

For the rest of the thesis, non dimensional variables are printed without the star (*) to simplify the notation. Such simplification does not cause any difficulties since the dimensional equation can be recovered by simply setting all of the reference variables to unity.

2.2 Filtered Navier-Stokes Equations

In laminar flow, the application of the Navier-Stokes equations is fairly straightforward. This is not the case for turbulent flows as the governing equations must deal with the intermittent coherent structures of the turbulence. As mentioned in the previous chapter, all of the turbulence eddies are resolved numerically in DNS. Hence, the Navier-Stokes equations can also be applied directly for DNS of turbulent flows. Unlike DNS however, the LES methodology relies upon filtering and modelling. For each flow variable, the following decomposition of filtered and modelled variables is used.

$$\varphi = \overline{\varphi} + \varphi' \tag{2.15}$$

where $\overline{\varphi}$ is the filtered and φ' is the modelled quantity.

LES employs a filtering operation to separate larger eddies to be solved numerically and smaller eddies to be modelled. Consider that the field φ is a function of coordinate position ξ in the domain $(-\infty, +\infty)$. The filtering operation is defined by

$$\bar{\varphi}(\xi) = \frac{1}{\Delta} \int_{-\infty}^{+\infty} G\left(\frac{\xi - \eta}{\Delta}\right) \varphi(\eta) d\eta \tag{2.16}$$

where Δ is the filter width, and η is the coordinate position relative to the filter frame of reference. These coordinate position can be spatial coordinates if the filtering operation is done in space or wave numbers for frequency domain operation. G is a filter function. The simplest form of G returns unity when η is within the filter width and 0 otherwise. This behaviour is exhibited by a Fourier cut-off filter in the frequency domain and a top hat filter in the spatial domain. Thus the former is mostly used in conjunction with the spectral method while the later is used mainly for finite difference and finite volume

based LES. Furthermore, it is easy to see that top hat filtering can be done implicitly during the finite volume discretisation [134] by specifying the computational cell as the filter width. Spatial filtering is used at the present work.

Inspection of the state vector **Q** shows that many of its elements are a product of density and other flow-field variables. Thus the filtered variables will have a rather tedious form [14]. As a common practice Favre averaging (density weighted averaging) is often use to simplify them. Favre averaging is defined as:

$$\widetilde{\varphi} = \frac{\overline{\rho\varphi}}{\overline{\rho}} \tag{2.17}$$

Thus the variable decomposition becomes

$$\varphi = \widetilde{\varphi} + \varphi'' \tag{2.18}$$

The notation defined by eq. 2.18 will be used for the following discussions.

Using the above filtering and Favre averaging technique, the convective term in the momentum equation becomes

$$\overline{\rho u_i u_j} = \overline{\rho}\widetilde{u}_i\widetilde{u}_j + \overline{\rho}\widetilde{u_i u_j''} + \overline{\rho}\widetilde{u_i'' u_j} + \overline{\rho}\widetilde{u_i'' u_j''} \tag{2.19}$$

The second and third terms on the right hand side are correlations between filtered and unfiltered components of the velocity field. In LES, these terms are usually called "cross term", which are not Galilean invariant. Since these are normally considered to be quite small in comparison to the last term, this is often neglected [89]. However, for the present work, the last three terms are simply formulated as $\overline{\rho u_i u_j} - \overline{\rho}\widetilde{u}_i\widetilde{u}_j$, which are subsequently modelled by a sub-grid scale model. Thus the filtered convective terms becomes

$$\overline{\rho u_i u_j} = \overline{\rho}\widetilde{u}_i\widetilde{u}_j + (\overline{\rho u_i u_j} - \overline{\rho}\widetilde{u}_i\widetilde{u}_j) \tag{2.20}$$

Using the above filtering and Favre averaging technique, the convective term in the energy equation is treated in manner as indicated in appendix A.

$$\overline{\rho E u_j} = \overline{\rho}\widetilde{E}\widetilde{u}_j + \overline{\theta} + \frac{1}{2}\mathcal{J}_j - \mathcal{D}_j \tag{2.21}$$

where $\nabla\overline{\theta}$ is the SGS heat flux, $\nabla\mathcal{J}$ is the SGS turbulent diffusion and $\nabla\mathcal{D}$ is the SGS viscous diffusion. These quantities are defined as follows

$$\overline{\theta} = c_v\overline{\rho}\left(\widetilde{Tu_j} - \widetilde{T}\widetilde{u}_j\right) + \overline{pu_j} - \overline{p}\widetilde{u}_j \tag{2.22}$$

$$\mathcal{J}_j = \overline{\rho}\left(\widetilde{u_j u_k u_k} - \widetilde{u}_j \widetilde{u_k u_k}\right) \tag{2.23}$$

$$\mathcal{D}_j = \overline{\tau_{ij} u_j} - \widetilde{\tau}_{ij}\widetilde{u}_j \tag{2.24}$$

It must be noticed that the cross terms between the high and low frequencies have not been neglected yet at this stage.

Martin et. al. [75], showed that the SGS viscous diffusion is very small. Thus the term is neglected at the present work. Following Knight et al. [62], the SGS turbulent diffusion is approximated as

$$\mathcal{J}_j \approx \left(\widetilde{\overline{\rho} u_j u_k} - \overline{\rho}\widetilde{u}_j\widetilde{u}_k\right)\widetilde{u}_j \tag{2.25}$$

which has been shown to be reasonably accurate [75].

2.2.1 Governing LES Equations

The filtering and decomposition of (2.15, 2.18) that has been described above modifies the **Q**, **F** and **G** terms (and also **S** when the system of equation is not conservative) in the Navier-Stokes equation as follows:

$$\mathbf{Q} = \begin{pmatrix} \overline{\rho} \\ \widetilde{\overline{\rho} u} \\ \widetilde{\overline{\rho} v} \\ \widetilde{\overline{\rho} w} \\ \widetilde{\overline{\rho} E} \end{pmatrix} \tag{2.26}$$

$$\mathbf{F}\cdot\hat{\mathbf{n}} = \widetilde{U}_n \begin{pmatrix} \overline{\rho} \\ \widetilde{\overline{\rho} u} \\ \widetilde{\overline{\rho} v} \\ \widetilde{\overline{\rho} w} \\ \widetilde{\overline{\rho} E} + \overline{p} \end{pmatrix} + \overline{p} \begin{pmatrix} 0 \\ \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \\ 0 \end{pmatrix} \tag{2.27}$$

$$\mathbf{G}\cdot\hat{\mathbf{n}} = \begin{pmatrix} 0 \\ \hat{n}_x(\overline{\tau}_{xx} + \overline{\sigma}_{xx}) + \hat{n}_y(\overline{\tau}_{yx} + \overline{\sigma}_{yx}) + \hat{n}_z(\overline{\tau}_{zx} + \overline{\sigma}_{zx}) \\ \hat{n}_x(\overline{\tau}_{xy} + \overline{\sigma}_{xy}) + \hat{n}_y(\overline{\tau}_{yy} + \overline{\sigma}_{yy}) + \hat{n}_z(\overline{\tau}_{zy} + \overline{\sigma}_{zy}) \\ \hat{n}_x(\overline{\tau}_{xz} + \overline{\sigma}_{xz}) + \hat{n}_y(\overline{\tau}_{yz} + \overline{\sigma}_{yz}) + \hat{n}_z(\overline{\tau}_{zz} + \overline{\sigma}_{zz}) \\ \hat{n}_x bt_x + \hat{n}_y bt_y + \hat{n}_z bt_z \end{pmatrix} \tag{2.28}$$

where the contravariant velocity, $\widetilde{U}_n$ is defined as:

$$\widetilde{U}_n = \widetilde{u}\hat{n}_x + \widetilde{v}\hat{n}_y + \widetilde{w}\hat{n}_z \tag{2.29}$$

and bt_{x_i} are defined using tensor notation with the summation convention implied as:

$$bt_{x_i} = \widetilde{u}_j(\widetilde{\tau}_{x_i x_j} - \overline{\sigma}_{x_i x_j}) - \overline{\dot{q}_{x_i}} + \overline{\theta_{x_i}} \tag{2.30}$$

where $\overline{\theta}$ and $\overline{\sigma}$ are the two modelled quantities. The former has been defined earlier while the later is defined as

$$\overline{\sigma}_{x_i x_j} = -(\widetilde{\overline{\rho} u_i u_j} - \overline{\rho}\widetilde{u}_i\widetilde{u}_j) \tag{2.31}$$

which is $\mathcal{J}_j = \overline{\sigma}_{x_i x_j}\widetilde{u}_j$

2.2.2 Eddy Viscosity Hypothesis for SGS term

For the present work, the LES SGS models are based on the eddy-viscosity hypothesis of Boussinesq. Hence it is assumed that the filtered kinematic viscosity $\overline{\nu}$ can be mimicked by an eddy viscosity ν_t. Using the analogy that kinematic viscosity and molecular viscosity μ is related to $\mu = \rho\nu$, the eddy viscosity can be related to the "turbulence viscosity" as $\mu_t = \rho\nu_t$. Thus the $\overline{\sigma}_{x_i x_j}$ term is modelled as "turbulence viscosity" μ_t multiplied by velocity strain as for the filtered counterpart.

$$\overline{\sigma}_{x_i x_j} = \overline{\rho}\mu_t\left(\frac{\partial\widetilde{u}_i}{\partial x_j} + \frac{\partial\widetilde{u}_j}{\partial x_i}\right) - \overline{\rho}\frac{2}{3}\mu_t\frac{\partial\widetilde{u}_k}{\partial x_k}\delta_{ij} - \frac{2}{3}\widetilde{\overline{\rho}u_k''u_k''}\delta_{ij} \tag{2.32}$$

As shown by Warsi [135], the last term on the right hand side involving the turbulent kinetic energy, $\widetilde{u_k''u_k''}$, is required to give the correct trace of the Reynolds stress, which takes into account compressibility effects on the turbulence field. Consequently, this term is not significant in low Mach number flows where no appreciable change of density has been observed.

For low Mach number LES where the flow is practically incompressible, Bui [17] demonstrated a good agreement with a DNS flow-field when the turbulent kinetic energy term was neglected. Unlike the Bui computation [17], the present research is aimed at compressible flow in transonic and supersonic region. Thus, following Martin et al. [75] this term is retained, which leads to the following expression in LES computation of the stress terms in 2.28:

$$\overline{\tau}_{x_i x_j} + \overline{\sigma}_{x_i x_j} = \overline{\rho}(\mu + \mu_t)\left(\frac{\partial\widetilde{u}_i}{\partial x_j} + \frac{\partial\widetilde{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial\widetilde{u}_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\widetilde{\overline{\rho}u_k''u_k''}\delta_{ij} \tag{2.33}$$

Similarly, $\overline{\theta}$ is modelled as "turbulence conduction"

$$\overline{\theta}_{x_i} = \frac{\mu_t c_p}{Pr_t}\frac{\partial T}{\partial x_i} \tag{2.34}$$

where the turbulent Prandtl Number Pr_t is fixed at 0.9 for the present work. Hence the conduction term in 2.30 becomes:

$$\overline{\dot{q}_{x_i}} + \overline{\theta}_{x_i} = \left(\frac{\mu_t c_p}{Pr_t} + \frac{\mu c_p}{Pr} \right) \frac{\partial \overline{T}}{\partial x_i} \tag{2.35}$$

To simplify notation, the filtered variables for the rest of the thesis are written without tilde ($\tilde{\cdot}$) and over-bar ($\bar{\cdot}$).

2.3 Sub-Grid Scale Model

Three sub-grid scale (SGS) models are considered, namely the standard Smagorinsky model [36], the WALE model [84] and the DES version of the S-A turbulence model [117]. The standard Smagorinsky model is the simplest sub-grid scale model that is commonly used in LES. Its accuracy is often cited to be inferior to the dynamic variation of the Smagorinsky model (e.g Refs. [44, 133]) but its lower computational cost is judged to be important for the present work. The WALE model has been designed by Nicoud and Ducros [84] to improve the standard Smagorinsky model without incurring significant additional cost. At the time of writing, DES is still seen as rather controversial within the LES community. The model is considered here simply because it offers a built-in near wall model [85] as well as a straightforward extension from the RANS S-A model, which can be easily implemented on an unstructured solver.

2.3.1 Standard Smagorinsky model

The standard Smagorinsky model is derived from Prandtl mixing layer theory. In this model, the sub-grid scale turbulence dissipation is simply proportional to velocity strain rate and a filter length scale, which in a finite volume implementation is simply a characteristic cell dimension. In many ways, it is very similar to an algebraic turbulence model of RANS. The standard Smagorinsky model, with summation convention implied, is defined as follows

$$\mu_t = C_s^2 \rho \triangle^2 \sqrt{2 S_{ij} S_{ij}} \tag{2.36}$$

and following Martin et al. [75], who implemented the Yoshizawa model [139], the SGS kinetic energy in eq. 2.32 is modelled as:

$$\widetilde{\rho u_k'' u_k''} = C_I^2 \rho \triangle^2 \sqrt{2 S_{kk} S_{kk}} \tag{2.37}$$

where strain rate is defined as

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.38}$$

It is obvious that this model is limited by having a single value of Smagorinsky constant C_s throughout the flow domain. This can be an important issue for wall bounded flows. On the wall surface, the dissipation must be zero. Since the strain rate is not zero on the wall, this implies that C_s should be zero. To rectify this a simple damping function based upon Van Driest damping is often introduced by replacing C_s^2 in (2.36) by C. An example of such modification is presented by DeBonis and Scott [26] as follows:

$$C = C_s^2\left[1.0 - e^{\left(\frac{d^+}{25.0}\right)}\right] \tag{2.39}$$

where d^+ is the distance from nearest wall in wall units, defined as

$$d^+ = \frac{\rho u_\tau d}{\mu} \tag{2.40}$$

Friction velocity u_τ is defined as a function of wall shear stress τ_w as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{2.41}$$

However, such a damping function is not implemented for the present work since the calculation of d^+ would require an awkward data structure to store the corresponding wall shear stress for each vertex.

As there is no agreement in the literature on the value of C_s and C_I that is applicable for general flow conditions, they will be defined according to the acceptable values for a particular flow problem.

The dynamic procedure that was introduced by Germano et. al. [44] tried to address this problem by calculating C_s through a test filtering. Hence its value becomes a function of space and time. This method has become one of the most popular practices in LES [96]. The implementation of the dynamic procedure is felt unnecessary for the present work because of problems related to robustness of the computation as C_s may become negative. One way to cure this problem is by taking a spatial averaging along a direction that is statistically uniform (e.g. the span-wise direction of a channel). Whilst such a direction can be found easily in simple problems it is not generally available in practical three dimensional engineering problems that typically need an unstructured mesh.

2.3.2 WALE model

In ref. [84] Nicoud and Ducros argued that the standard Smagorinsky model suffered from two problems. Firstly the strain rate disagrees with the kinematic and dynamic properties of turbulence. Instead, these properties should be better represented by both strain rate – the symmetric part of the velocity gradient – and vorticity – the anti-symmetric part of velocity gradient. Hence WALE is derived from the velocity gradient tensor as follows.

$$\mathcal{S}_{ij}^{d} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_k}\frac{\partial u_k}{\partial x_j} + \frac{\partial u_j}{\partial x_k}\frac{\partial u_k}{\partial x_i}\right) - \frac{1}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\frac{\partial u_k}{\partial x_k} \tag{2.42}$$

where the summation convention is implied. Secondly, the near wall behaviour of the standard Smagorinsky model with Van Driest damping is in general analytically incorrect. The WALE model is defined as

$$\mu_t = C_w^2 \rho \triangle^2 \frac{\overline{OP1}}{\overline{OP2}} \tag{2.43}$$

$$\overline{OP1} = \left(\mathcal{S}_{ij}^{d}\mathcal{S}_{ij}^{d}\right)^{3/2} \tag{2.44}$$

$$\overline{OP2} = \left(S_{ij}S_{ij}\right)^{5/2} + \left(\mathcal{S}_{ij}^{d}\mathcal{S}_{ij}^{d}\right)^{5/4} \tag{2.45}$$

where C_w is a model constant. This model can easily be related to a standard Smagorinsky model by defining C_w as follows

$$C_w^2 = C_s^2 \frac{\sqrt{2}\left(S_{ij}S_{ij}\right)^{3/2}}{S_{ij}S_{ij}\overline{OP1}/\overline{OP2}} \tag{2.46}$$

WALE is especially attractive to an unstructured mesh solver as it does not requires the computation of d^+. The Yoshizawa model [139] is also used for the compressibility correction as defined in equation 2.37.

2.3.3 Modified S-A model for Detached-Eddy Simulation

It was argued by Spalart et al. [117] that the dissipation provided by the Spalart-Allmaras model[1] is proportional to the nearest wall distance d that appears in the destruction of the source term. The dissipation of the model can be made proportional to filter size as in the Smagorinsky SGS model by replacing the d in equations A.16 and A.22 with $\widetilde{d}$ which is a function of filter size.

$$\widetilde{d} = min(C_{DES}\triangle, d) \tag{2.47}$$

[1]The full description of the S-A model is given in section A.4.

where C_{DES} is 0.65, as suggested by Spalart *et. al.*

In this modified form, the S-A model can be considered as a one equation sub-grid scale model similar to that of Yoshizawa [139, 141]. It has been shown by Shur et al. [106] that DES mimics the standard Smagorinsky based large eddy simulation for decaying homogeneous flow. It must be emphasised that the compressibility correction used in conjunction with the other two models is not needed by the DES.

Nikitin et al. [85] has shown DES can be seen as a form of LES with a built-in wall function, which is provided by the RANS part of the model. Unlike the mainstream research on wall modelling for LES whose aim is to obtain a near wall cell that lies in the logarithmic region of a boundary layer while maintaining low aspect ratio cells, DES retains the characteristics of a low-Reynolds-number turbulence model based RANS simulation that employs a very high aspect ratio cell that is capable of resolving the laminar sub-layer.

Chapter 3

Feasibility study of Padé Compact Scheme on Unstructured Grids

3.1 Introduction

The resolved component of an LES computation requires a numerical simulation for the convection of the intermittent turbulent vortical structures. Physically, these structures would continually be created, interact with the main flow and each other to be stretched, distorted, broken up to smaller scales and finally dissipated [76]. The dissipation is mainly carried out at the small scale level, i.e. it is performed by the SGS model. Consequently, numerical schemes for LES should ideally be free of dissipation in order to avoid any interference with the SGS term. Ghosal argued that a higher order scheme is necessary to achieve this [45]. In accordance with that study, there has been a growing trend in the LES community to implement fourth or sixth order accurate schemes [14, 26, 128], even though there is no definitive proof that the second order schemes predominant in the current generation of LES codes are unsuitable for such simulation [96]. As a short hand, in the rest of this chapter the term higher order scheme describes any spatial scheme that is higher than second order accurate. It is quite obvious that upwind schemes, whose leading truncation error term has a dissipative property, are deemed to be unsuitable for LES [12].

This chapter investigates the feasibility of implementing a high order scheme based

upon the Lele Padé compact scheme, henceforth simply called the compact scheme, in an unstructured solver. In one dimension, the scheme has been proved to have spectral method like resolution [67]. Whilst the three dimensional implementation in a structured mesh solver can be seen simply as an extension of the one dimensional formulation in three linearly independent directions [113], an unstructured mesh formulation is inherently multidimensional. Therefore, it is of particular interest to assess the practicality of implementing the compact higher order scheme on an unstructured mesh based solver. It must be noted that to this end the only work on a higher order LES code for unstructured mesh seems to the Spectral element method [60] that was essentially a Discontinuous Galerkin method.

3.2 Linear Convective Term in one dimension

To simplify the arguments, the linear wave equation (also known as linear Burgers equation) is used as a model in this section rather than the more complex system of equations that govern fluid flows as described in chapter 2. In one dimension, this equation reads

$$\frac{\partial U}{\partial t} = -c\frac{\partial U}{\partial x} \tag{3.1}$$

where c is the wave convection speed

The conventional finite differencing technique employs an explicit formulation that is derived from the Taylor expansion of the dependent variables, U, at a stencil of neighbouring nodes. A second order accurate estimate of the first derivative at node i is obtained by utilising Taylor expansion from node $\mathrm{i}+1$ to $\mathrm{i}-1$ as follows:

$$\left[\frac{\partial U}{\partial x}\right]_{\mathrm{i}} = \frac{U_{\mathrm{i}+1} - U_{\mathrm{i}-1}}{2\Delta x} + \mathcal{O}\left(\Delta x\right)^2 \tag{3.2}$$

An explicit fourth order accurate estimate of the first derivative can be obtained by expanding the stencil to include nodes $\mathrm{i}-2$ and $\mathrm{i}+2$.

$$\left[\frac{\partial U}{\partial x}\right]_{\mathrm{i}} = \frac{-U_{\mathrm{i}+2} + 8U_{\mathrm{i}+1} - 8U_{\mathrm{i}-1} + U_{\mathrm{i}-2}}{12\Delta x} + \mathcal{O}\left(\Delta x\right)^4 \tag{3.3}$$

Alternatively, it is possible to write the Taylor expansion of the first derivative on the same uniform mesh as follows

$$\left[\frac{\partial U}{\partial x}\right]_{\mathrm{i}+1} = \left[\frac{\partial U}{\partial x}\right]_{\mathrm{i}} + \Delta x\left[\frac{\partial^2 U}{\partial x^2}\right]_{\mathrm{i}} + \frac{\Delta x^2}{2!}\left[\frac{\partial^3 U}{\partial x^3}\right]_{\mathrm{i}} + \frac{\Delta x^3}{3!}\left[\frac{\partial^4 U}{\partial x^4}\right]_{\mathrm{i}} + \mathcal{O}\left(\Delta x\right)^4 \tag{3.4}$$

$$\left[\frac{\partial U}{\partial x}\right]_{i-1} = \left[\frac{\partial U}{\partial x}\right]_i - \Delta x \left[\frac{\partial^2 U}{\partial x^2}\right]_i + \frac{\Delta x^2}{2!}\left[\frac{\partial^3 U}{\partial x^3}\right]_i - \frac{\Delta x^3}{3!}\left[\frac{\partial^4 U}{\partial x^4}\right]_i + \mathcal{O}(\Delta x)^4 \quad (3.5)$$

Upon scaling eq. 3.4 and 3.5 by α and adding them to eq. 3.2 one obtains an implicit scheme with fourth order accuracy for the first derivative.

$$\alpha\left[\frac{\partial U}{\partial x}\right]_{i-1} + (1-2\alpha)\left[\frac{\partial U}{\partial x}\right]_i + \alpha\left[\frac{\partial U}{\partial x}\right]_{i+1} = \frac{U_{i+1} - U_{i-1}}{2\Delta x} + \beta\Delta x^2\left[\frac{\partial^3 U}{\partial x^3}\right]_i + \mathcal{O}(\Delta x)^4 \quad (3.6)$$

where $\beta = \alpha - 1/3!$ It can be seen clearly that setting α to $1/6$ (hence $\beta = 0$) leads to a fourth order accurate implicit discretisation with a three point stencil rather than the five that was used by the explicit scheme of eq. 3.3.

In ref. [67] Lele has employed this hermitian (Padé) methodology to devise a family of spatial derivative schemes that are widely known as compact schemes, which can be written using a set of parameters as follows:

$$\alpha_0[\delta_x U]_{i-1} + [\delta_x U]_i + \alpha_0[\delta_x U]_{i+1} = a\frac{U_{i+1} - U_{i-1}}{2\Delta x} + b\frac{U_{i+2} - U_{i-2}}{4\Delta x} + c\frac{U_{i+3} - U_{i-3}}{6\Delta x} \quad (3.7)$$

Notice δ_x signifies that a numerical approximation of $\partial/\partial x$ has been used, hence there is no need to write the small error term. Here the fourth order accurate scheme reads

$$\frac{1}{4}[\delta_x U]_{i-1} + [\delta_x U]_i + \frac{1}{4}[\delta_x U]_{i+1} = \frac{3}{2}\frac{U_{i+1} - U_{i-1}}{2\Delta x} \quad (3.8)$$

It must be noted that the saving that comes from having a smaller stencil is partly offset by the increase of of computational effort to solve the implicit formulation.

Fourier analysis has become an important tool to analyse partial differential equations, both in analytical and discrete forms. To simplify the current analysis, only the internal node discretisation is discussed in this thesis. Using a Fourier transform, U can be expressed as a complex number as follows

$$U(x,t) = u(t)e^{jKx} \quad (3.9)$$

where K is the wave number and $j = \sqrt{-1}$. In this format the first derivative can be written simply as

$$\frac{\partial U}{\partial x} = jKu(t)e^{jKx} = jKU \quad (3.10)$$

Thus the linear wave equation can be written as

$$\frac{\partial U}{\partial t} = -jcKU \quad (3.11)$$

In discrete form eq. 3.9 can be expressed as follows for node x_0 and its neighbour, $x_0 + \Delta x$

$$U_i = U(x_i, t) = u(t)e^{jKx_0} \tag{3.12}$$

$$U_{i+1} = U(x_{i+1}, t) = u(t)e^{jK(x_0+\Delta x)} \tag{3.13}$$

Fourier transform of the second order accurate approximation of the first derivative can be obtained by inserting eq. 3.12 and 3.13 to 3.2.

$$\delta_x U_i = \frac{u(t)}{2\Delta x}\left(e^{jK(x_0+\Delta x)} - e^{jK(x_0-\Delta x)}\right) \tag{3.14}$$

Thus the discrete linear wave equation simply becomes:

$$\delta_t U_i = -\frac{cu(t)e^{jKx_0}}{K\Delta x}\frac{jK}{2}\left(e^{jK\Delta x} - e^{-jK\Delta x}\right) = -jcKU_i\frac{\sin(K\Delta x)}{K\Delta x} = -jc^*KU \tag{3.15}$$

where c^* is the discrete wave speed. The Euler formula [64] has been used to connect the exponential and trigonometric forms of the complex number.

Comparison between the discrete second order first derivative of eq. 3.15 and its analytical form as defined by eq. 3.11 yields the following transfer function that relates the analytical wave speed to its numerical counterpart.

$$\frac{c^*}{c} = \frac{\sin(K\Delta x)}{K\Delta x} \tag{3.16}$$

This function reflects how the discretisation error varies with the wave number. Similarly, the discretisation error for the explicit fourth order scheme can be analysed by performing the Fourier transform on eq. 3.3 and yields the following result

$$\frac{c^*}{c} = \frac{8\sin(K\Delta x) - \sin(2K\Delta x)}{K\Delta x} \tag{3.17}$$

whereas the explicit sixth order scheme yields

$$\frac{c^*}{c} = \frac{45\sin(K\Delta x) - 9\sin(2K\Delta x) + \sin(3K\Delta x)}{K\Delta x} \tag{3.18}$$

To carry out Fourier analysis on the fourth order accurate compact scheme, one starts by performing the Fourier transform on eq. 3.8

$$\delta_x U_i\left(e^{-jK\Delta x} + 4 + e^{jK\Delta x}\right) = -\frac{3u(t)}{\Delta x}\left(e^{jK(x_0+\Delta x)} - e^{-jK(x_0-\Delta x)}\right) \tag{3.19}$$

Upon rearranging and manipulating eq. 3.19 using the Euler formula, the following expression for the discrete first order derivative is obtained

$$\delta_x U_i = -\frac{3U_i\sin(K\Delta x)}{\Delta x\left[2 + \cos(K\Delta x)\right]} \tag{3.20}$$

Thus, the discrete linear wave equation can be expressed as

$$\delta_t U_i = -jKU_i c \frac{3\sin(K\Delta x)}{K\Delta x\,[2+\cos(K\Delta x)]} \tag{3.21}$$

which yields the following discrete wave speed function

$$\frac{c^*}{c} = \frac{3\sin(K\Delta x)}{K\Delta x\,[2+\cos(K\Delta x)]} \tag{3.22}$$

For completeness, Fourier transform of the first order upwind scheme reads

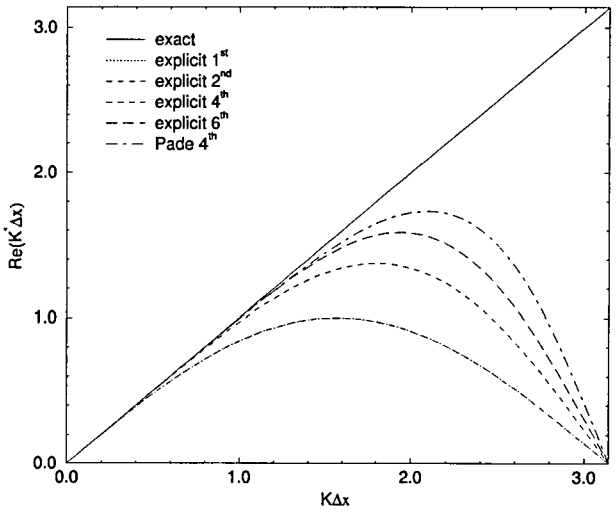
$$\delta_t U_i = -c\frac{u(t)e^{jKx_0}\left(1-e^{jK\Delta x}\right)}{\Delta x} = -jcKU_i\frac{1-\cos(K\Delta x)+j\sin(K\Delta x)}{K\Delta x} \tag{3.23}$$

which yields the following transfer function

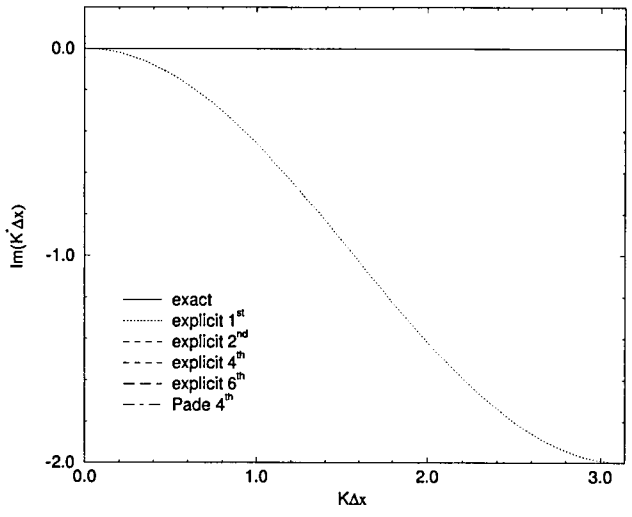
$$\frac{c^*}{c} = \frac{\sin(K\Delta x)-j\,[1-\cos(K\Delta x)]}{K\Delta x} \tag{3.24}$$

Even though the first order upwind scheme has a similar dispersion error to the second order scheme, it has an additional dissipation error from the leading error term that shows up as an imaginary component in the Fourier analysis. It must be mentioned that the Fourier transform of the discrete equation on a non uniform mesh always has the imaginary component that causes dissipative error.

Figure 3.1 shows the resolution of numerical wave speed as a function of wave number. Clearly the second order accurate scheme can only cope with low wave numbers and quickly suffers from dispersion error as the wave number increases whereas the explicit fourth order scheme is capable of dealing with larger wave numbers. It means that as the order of accuracy gets higher, the mesh can resolve a lot more information from the flow. In other words, a higher order scheme on a coarser mesh would produce similar accuracy to a lower order scheme on a finer mesh. It is also quite interesting to note that the Lele compact scheme yields lower dispersion error than its explicit scheme counterpart for the same order of accuracy. The Lele compact scheme is popular for this reason. Figure 3.1 also demonstrates that all of the centred schemes in uniform mesh does not have dissipation error. The figure also illustrates that the first order upwind scheme has identical dispersion error to that of second order centred scheme. But unlike the centred scheme, the first order upwind scheme has significant dissipation error.



(a) Dispersion:real component



(b) Dissipation:imaginary component

Figure 3.1: Modified wave number for convective term in 1 dimension

3.3 Convective term in two dimensions

The one dimensional analysis that has been shown previously can be easily extended to two dimensions, where the linear Burgers equation simply becomes

$$\frac{\partial U}{\partial t} = -c\cos\theta\frac{\partial U}{\partial x} - c\sin\theta\frac{\partial U}{\partial y} \tag{3.25}$$

where c is once again the wave convection speed and θ is the direction of the convection relative to the x axis. As in one dimension, U can be expressed as a complex number as follows

$$U(x,y,t) = u(t)e^{jK(x\cos\theta+y\sin\theta)} \tag{3.26}$$

In this format first derivatives can be written simply as

$$\frac{\partial U}{\partial x} = jK\cos\theta u(t)e^{jK(x\cos\theta+y\sin\theta)} = jKU\cos\theta \tag{3.27}$$

$$\frac{\partial U}{\partial y} = jK\sin\theta u(t)e^{K(x\cos\theta+y\sin\theta)} = jKU\sin\theta \tag{3.28}$$

Thus the linear wave equation can be written as

$$\frac{\partial U}{\partial t} = -jcKU\left(\cos^2\theta + \sin^2\theta\right) = -jcKU \tag{3.29}$$

Unlike the one dimensional analysis that can only have one type of discretising mesh, in two dimensions there can be an infinite number of ways to do this. Despite this, the discretisation can always be cast along two linearly independent directions, i.e. the orthogonal x and y directions. By simple algebraic manipulation, the wave speed transfer function can be expressed generally as follows:

$$\frac{c^*}{c} = \frac{\delta_x U_0\cos\theta + \delta_y U_0\sin\theta}{jKU_0} \tag{3.30}$$

where U_0 is the value at the node of interest, which for the rectangular mesh would be recast as $U_{I,J}$. The transfer function returns unity value when there is no error. In this section the analysis is performed only on a uniform rectangular mesh and a uniform equilateral triangular mesh.

3.3.1 Uniform rectangular mesh

Fig. 3.2 shows a uniform rectangular mesh which can be regarded as a prototype of a structured mesh with uniform spacing L.

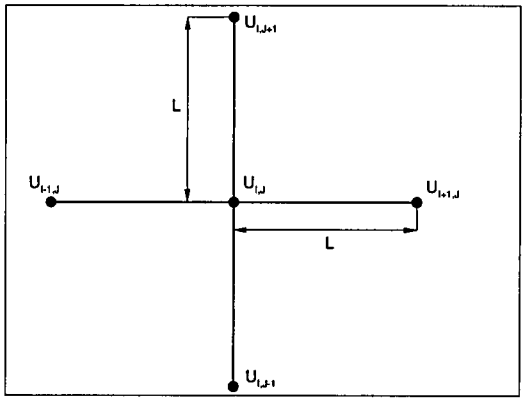


Figure 3.2: Five point structured stencil

For this mesh it can be shown that the Fourier Transform of the explicit second order accurate discretisation reads

$$\delta_x U_{\mathrm{I,J}} = jU_{\mathrm{I,J}} \sin(KL\cos\theta) \tag{3.31}$$

$$\delta_y U_{\mathrm{I,J}} = jU_{\mathrm{I,J}} \sin(KL\sin\theta) \tag{3.32}$$

whereas the Fourier Transform of the fourth order Padé scheme returns

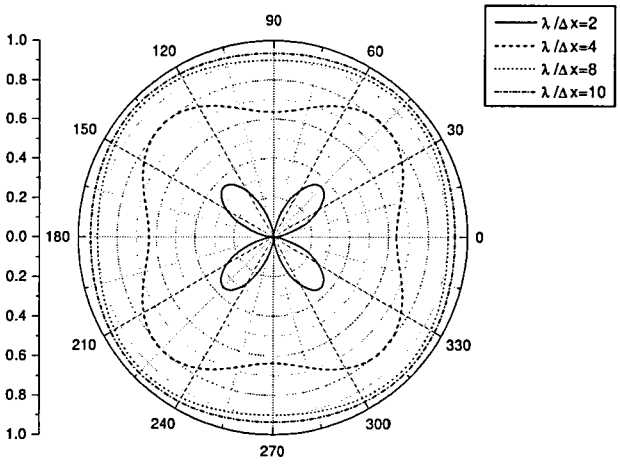
$$\delta_x U_{\mathrm{I,J}} = \frac{jU_{\mathrm{I,J}} 3\sin(KL\cos\theta)}{2+\cos(KL\cos\theta)} \tag{3.33}$$

$$\delta_y U_{\mathrm{I,J}} = \frac{jU_{\mathrm{I,J}} 3\sin(KL\sin\theta)}{2+\cos(KL\sin\theta)} \tag{3.34}$$

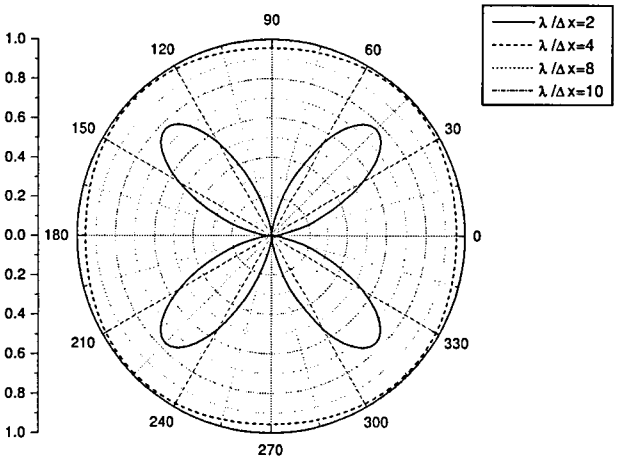
Figure 3.3 compares the polar plot of the transfer functions between the two discretisation methods. The difference between the analytical convective speed c and its numerical counterpart c^* is shown as deviation from a circle of unity radius. The fourth order compact scheme clearly demonstrates its ability in resolving higher wave numbers than its second order explicit counterpart. At very high wave number, two unit cell Δx per wavelength, both are noticeably directional with $\theta = 45\,\mathrm{deg}$ being the preferential direction.

3.3.2 A numerical experiment for a uniform rectangular mesh

It must be said that the superior performance in dealing with linear convection problems does not always extend to non-linear problems such as the Navier-Stokes equations. For this reason the non-linear property is assessed here.



(a) Second order explicit scheme



(b) Fourth order compact scheme

Figure 3.3: Polar plot of modified wave number for convective term in two dimension using five point structured stencil.

To demonstrate how the order of accuracy of spatial discretisation of the convective terms in the flow equations affects the flow structures, a simple numerical experiment was performed. To isolate the effect of the numerical scheme for the convective term, a single flow structure – namely a hypothetical vortex – was convected from one end of the computational domain to the other in an inviscid uniform flow. Figure 3.4 illustrates the basic set up of the experiment, notice that the observation line slices the domain at the position where the resultant velocity is more or less maximum.

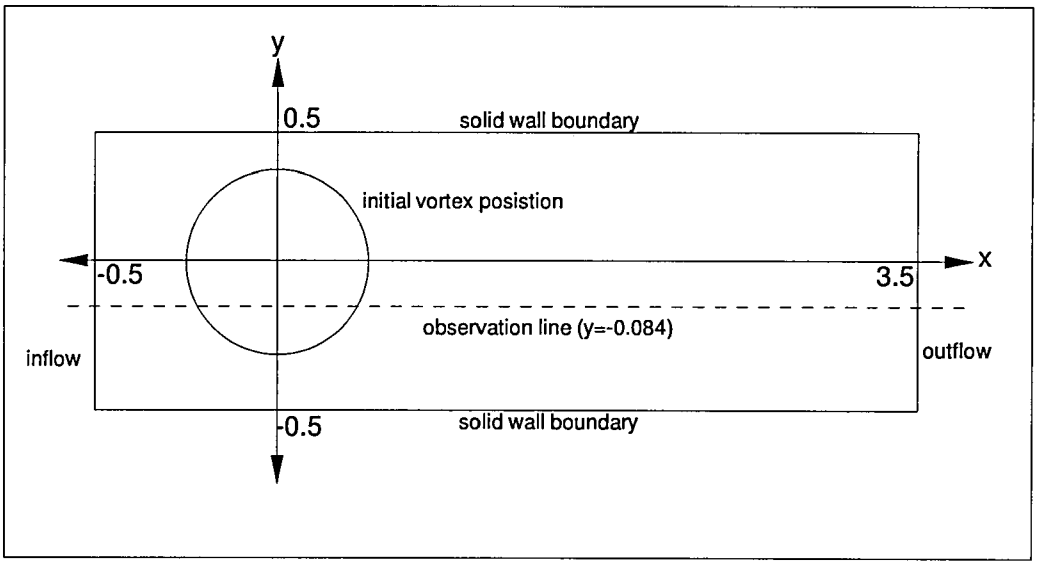


Figure 3.4: Computational set up for vortex convection

A uniform flow-field with 10 units velocity along the x-axis, U_∞ was used. The vortex was introduced by superimposing the following perturbation.

$$u = U_\infty - \frac{by}{R^2} e^{\left(-\frac{r^2}{2}\right)} \tag{3.35}$$

$$v = \frac{bx}{R^2} e^{\left(-\frac{r^2}{2}\right)} \tag{3.36}$$

$$T = T_o - \frac{\rho b^2 r^2}{2R^2 \Re\gamma} e^{(-r^2)} \tag{3.37}$$

$$\rho = \rho_o \left(\frac{T}{T_o}\right)^{\frac{1}{\gamma-1}} \tag{3.38}$$

$$p = p_o \left(\frac{T}{T_o}\right)^{\gamma} \tag{3.39}$$

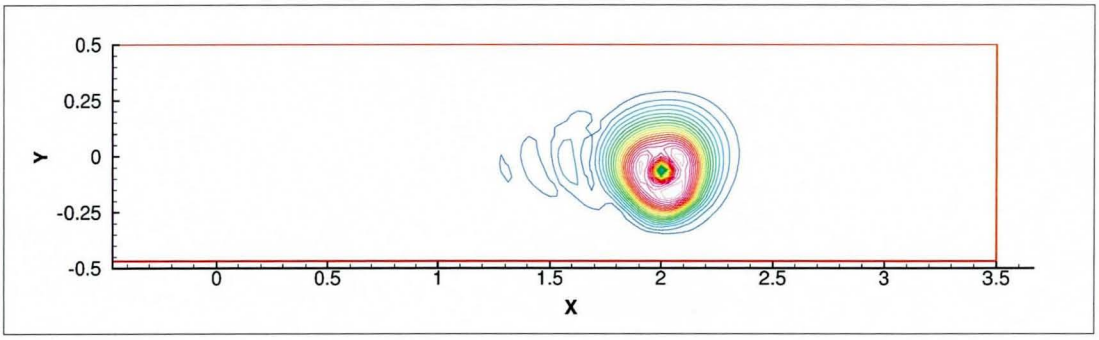
$$r = \sqrt{\frac{x^2 + y^2}{R^2}} \tag{3.40}$$

where the vortex radius,R and strength, b, are 0.1 and 2.0 units, respectively. The fluid properties were assumed to be 1.4 for γ, and 10.0 units for $\Re$. The stagnation pressure, p_o, temperature, T_o, and density, ρ_o were 10000, 100 and 100 units, respectively. It must be emphasised that such a vortex is purely hypothetical as the rotation did not introduce any pressure difference perpendicular to the convection (i.e. a radially symmetric pressure disturbance is applied for the vortical structure). This particular set up was introduced to ensure that the structure moves mainly along a single direction, namely the x axis. A similar test has also been used by Smirnov et al. [113]

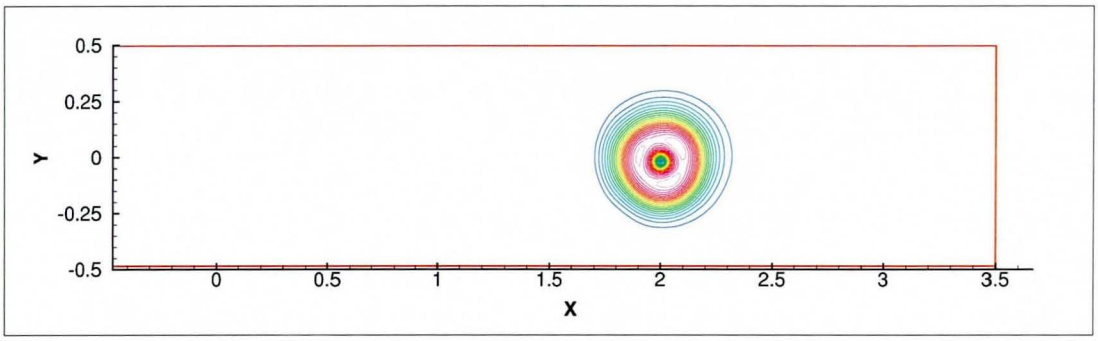
Two uniform structured meshes were employed for this study: one served as a baseline mesh while the other was used for a refinement study. The baseline mesh was 128×32 while the refined one was 256×64. Both were designed to be fine enough in resolving the vortex size to ensure that each cell only dealt with low wave numbers.

For this experiment a second order accurate LES code [137], whose viscous terms have been disabled, has been employed as a reference. A simple structured code with low storage third order accurate Runge-Kutta time stepping has been written for the higher order convective scheme [126]. In all codes the time stepping was set at 1.0e-4 units.

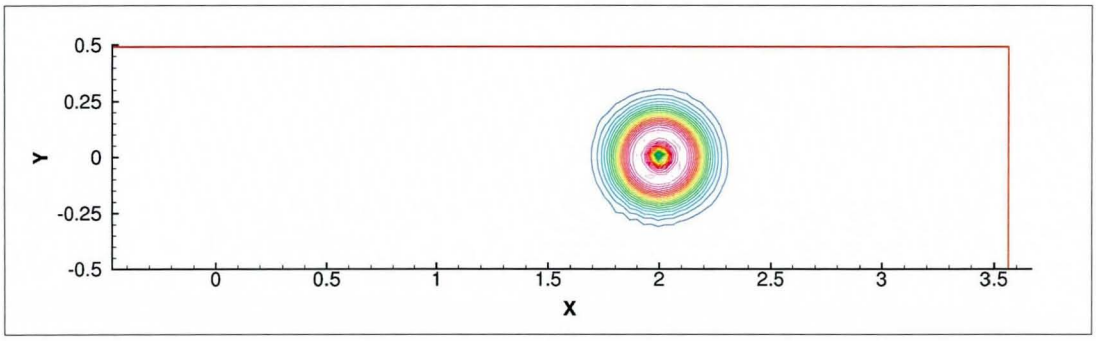
Figures 3.5 and 3.6 show the vortex evolution indicated by the two numerical schemes. The fourth order compact scheme gives very small dispersion error so that the vortex distortion is virtually non existent within the time integration interval. It can be said that the high resolution property that has been seen in the linear analysis does extend to the non linear problem of flow computation in a uniform Cartesian mesh. The second order scheme clearly exhibits dispersion error that leads to phase lag of the high wave number components. Parts of the original vortex lag behind the rest of the structure being detached into ripples that follow the structure as it moves. The apparent decay of the peak of the instantaneous u-velocity indicates the gradual movement of the vortex along the y axis. Although this sideways motion is expected to occur due to the speed differential between the top and bottom part of the vortex, Smirnov et al. [113] did not report any significant sideways movement. As the fourth order scheme hardly shows the deviation, it suggest that the sideways movement observed in the second order computation is mainly caused by numerical error, possibly the pressure-velocity coupling employed by the modified LES code. By employing twice as many nodes along the x and y directions, i.e. quadrupling the number of nodes, the second order results show more



(a) Second order explicit scheme; baseline mesh



(b) Second order explicit scheme; fine mesh



(c) Fourth order compact scheme; baseline mesh

Figure 3.5: Snap-shot of instantaneous perturbation speed at t=0.2 unit time. Consistent colour codes are used for the contour plots

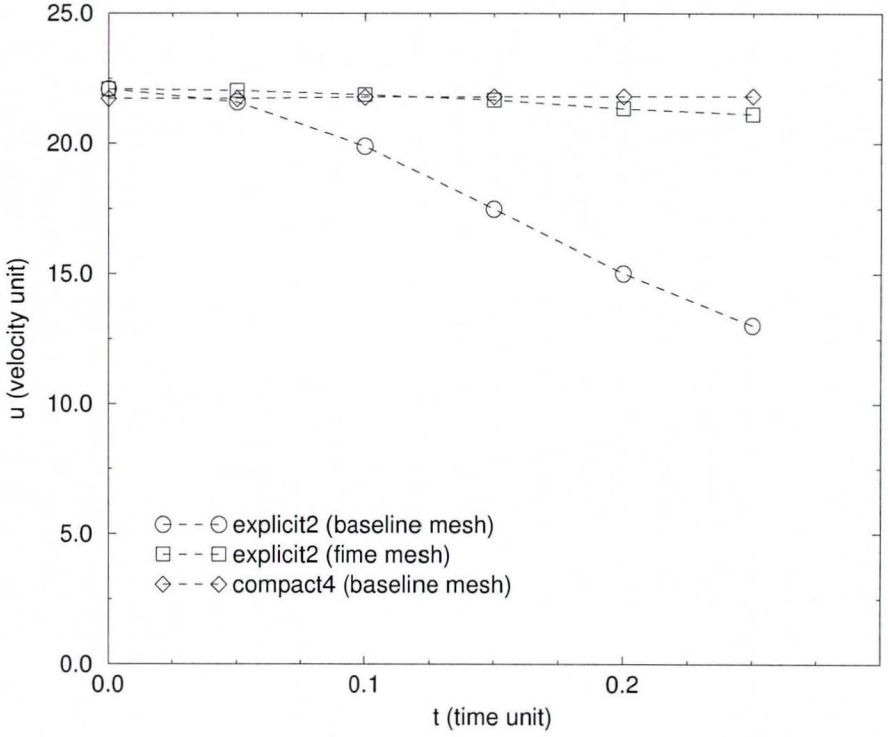


Figure 3.6: Evolution of peak velocity at the observation line. The explicit2 scheme clearly indicate that the vortex is shifted along y axis.

comparable accuracy to that of the fourth order compact scheme. In this finer mesh, the sideways motion is much smaller than in the coarser mesh computation. Notice that the modified LES code (for the second order explicit scheme) employs staggered mesh, hence the velocity is taken at slightly different location than the co-located code of the compact scheme, Hence the discrepancy at the initial condition.

For completeness, a result from the upwind bias scheme with the third order accurate MUSCL interpolation method [130] is also presented here. This scheme is employed in conjunction with Roe FDS scheme, and is described fully in section 4.4.1. Here, the upwinding effect is minimised by scaling down the Roe Dissipation to a quarter of its value and deactivating the limiter from the MUSCL interpolation. Figure 3.7(a) suggest that the sideways motion in this coupled solver is much smaller than that of second order modified LES code. Figure 3.7(b) shows that the rate of vortex strength decay during the computation. The plot suggests the kinetic energy is simply dissipated from the domain. In other word the scheme is not energy conserving. Such a behaviour is expected as a third order upwind scheme can be recast as fourth order scheme with numerical dissipation. Thus scaling down the numerical dissipation is indeed necessary to reduce the problem.

3.3.3 Uniform equilateral triangular mesh

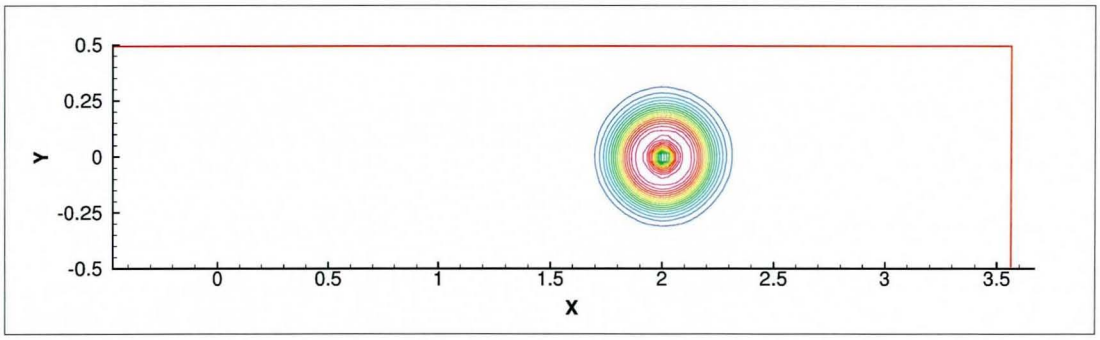
As before, the theoretical study for the triangular mesh is centred on linear convection. Figure 3.8 describes the uniform mesh that is used for this analysis. It is clear from the diagram that the one dimensional treatment can not be extended directly, hence, the second order accurate spatial derivative is calculated using the Green-Gauss theorem [64] that has been widely used for unstructured mesh solvers, (e.g. ref. [50]). It must be noted that in one dimension, the Green-Gauss theorem reverts to explicit second order finite difference formula. Thus,

$$\nabla U_0 = \frac{\sum_{\mathrm{i}} (U_0 + U_{\mathrm{i}}) \cdot \hat{n}_{0\mathrm{i}} s_{0\mathrm{i}}}{2\Gamma_0} \tag{3.41}$$

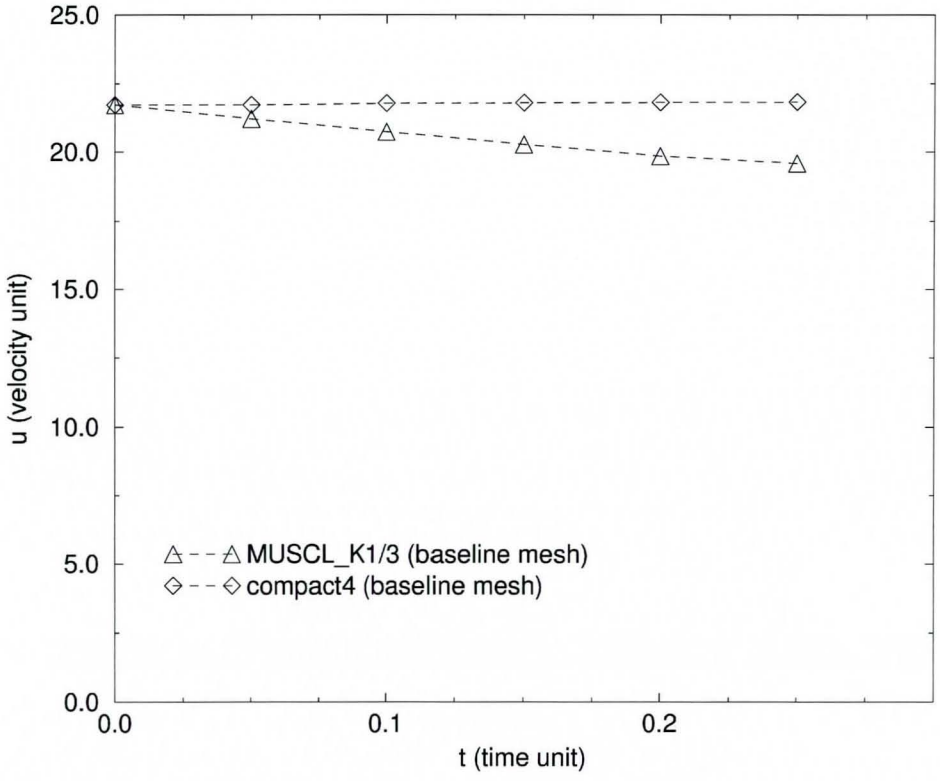
where Γ_0 is the area of the two dimensional median dual control volume, $\hat{n}_{0\mathrm{i}}$ is the unit vector of the control volume surface face with magnitude $s_{0\mathrm{i}}$. For the particular mesh shown in fig. 3.8 the resulting spatial derivatives are

$$\delta_x U_0 = \frac{U_3 - U_6}{3L} + \frac{U_2 + U_4 - U_1 - U_5}{6L} \tag{3.42}$$

$$\delta_y U_0 = \frac{\sqrt{3}\,(U_4 + U_5 - U_1 - U_2)}{6L} \tag{3.43}$$



(a) Snap-shot of instantaneous perturbation speed at t=0.2 unit time; Consistent colour code with the centred scheme contour plot are used.



(b) Decay of peak u-velocity along the observation line

Figure 3.7: Vortex evolution obtained from a scaled upwind-bias scheme

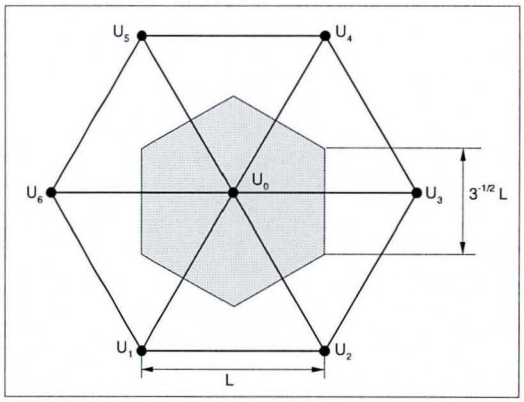


Figure 3.8: Equilateral triangular stencil; shaded area is the median dual.

Upon performing the Fourier transform and rewriting the exponential terms into sine and cosine functions, the following expressions are obtained.

$$\delta_x U_0 = \frac{jU_0}{3L}\left[2\sin\left(KL\cos\theta\right) + 2\sin\left(\frac{KL\cos\theta}{2}\right)\cos\left(\frac{\sqrt{3}KL\sin\theta}{2}\right)\right] \tag{3.44}$$

$$\delta_y U_0 = \frac{j\sqrt{3}U_0}{3L}\left[2\sin\left(\frac{KL\sin\theta}{2}\right)\cos\left(\frac{\sqrt{3}KL\cos\theta}{2}\right)\right] \tag{3.45}$$

Analogous to eq. 3.8, the fourth order compact scheme can be obtained by adding the derivative of the neighbouring nodes to the left hand side the Green-Gauss equation, thus

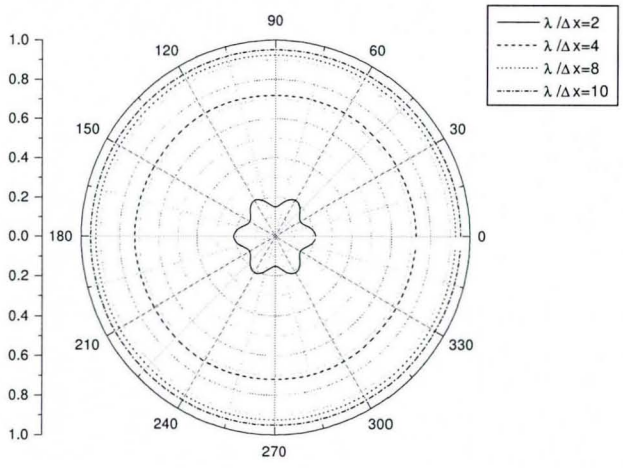
$$6\delta_x U_0 + \sum_{j=1}^{6}\delta_x U_j = \frac{4\left(U_3 - U_6\right)}{L} + \frac{2\left(U_2 + U_4 - U_1 - U_5\right)}{L} \tag{3.46}$$

$$6\delta_y U_0 + \sum_{j=1}^{6}\delta_y U_j = \frac{2\sqrt{3}\left(U_4 + U_5 - U_1 - U_2\right)}{L} \tag{3.47}$$

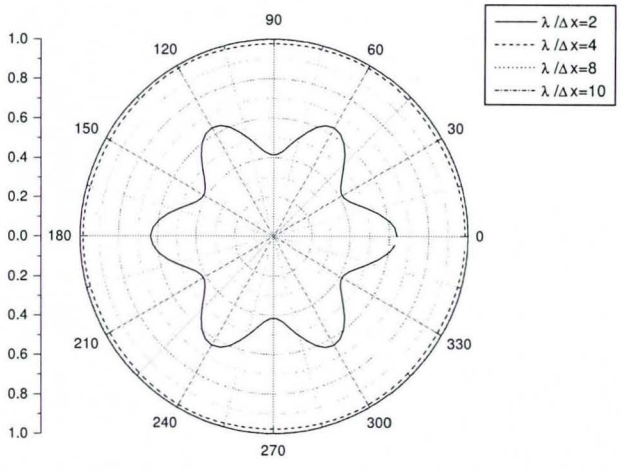
Upon performing Fourier Transform and rewriting the exponential terms into sine and cosine functions, the following expressions are obtained.

$$\delta_x U_0 = \frac{\left[\delta_x U_0\right]_{order2}}{4\cos\left(\frac{KL\cos\theta}{2}\right)\cos\left(\frac{\sqrt{3}KL\sin\theta}{2}\right)2\cos\left(KL\cos\theta\right) + 6} \tag{3.48}$$

$$\delta_x U_0 = \frac{\left[\delta_y U_0\right]_{order2}}{4\cos\left(\frac{KL\cos\theta}{2}\right)\cos\left(\frac{\sqrt{3}KL\sin\theta}{2}\right)2\cos\left(KL\cos\theta\right) + 6} \tag{3.49}$$



(a) second order explicit scheme



(b) fourth order compact scheme

Figure 3.9: Polar plot of modified wave number for convective term in two dimension using equilateral triangular stencil

where $[\delta_x U_0]_{order2}$ and $[\delta_y U_0]_{order2}$ are the explicit second order accurate approximation as given by eq. 3.44 and 3.45, respectively. This analysis is similar to that of Zingg and Lomax [144].

Figure 3.9 compares the polar plot of the transfer function between the two discretisation methods on the uniform equilateral triangular mesh. Once again both are noticeably directional, for two cells per wavelength in particular, which shows periodic pattern at 60 deg interval. The compact scheme also exhibits better resolution for 4 and 8 cells per wavelength.

3.3.4 General unstructured mesh

It must be noted that up to this point both the Fourier Analysis and numerical experiments involving a system of non-linear equation have been performed on uniform meshes. However, the real life applications of CFD are hardly ever carried out on such meshes. Since the unstructured mesh solver that is the main objective of this work usually has to deal with non-uniform cells, an analysis on a general unstructured mesh must be conducted.

Zingg and Lomax [144] suggested that the compact formulation for a cluster of m vertices surrounding an arbitrary vertex 0 can be generalised into a family, very much like the Lele scheme [67], by introducing a parameter β as follows:

$$(1-\beta)\nabla U_0 + \frac{\beta}{m}\sum_{i=1}^{m}\nabla U_i = \frac{\sum_i (U_0 + U_i)\cdot \hat{\mathbf{n}}_{0i} s_{0i}}{2\Gamma_0} \tag{3.50}$$

or alternatively

$$(1-\beta)\nabla U_0 + \frac{\beta}{m}\sum_{i=1}^{m}\nabla U_i = \frac{\sum_i (U_0 + U_i)\cdot \mathbf{n}_{0i}}{2\Gamma_0} \tag{3.51}$$

where the normal vector is used rather than the unit normal and face area. There is no reason why eq. 3.50 cannot be modified so that the right hand side expression is obtained from the Least squares method rather than the Green-Gauss theorem. However, in this section only the Green-Gauss gradient reconstruction is employed as it leads to a simpler analysis.

At a glance such a formulation seems to be promising as it suggests that the higher order accuracy using the compact scheme that has been applied successfully in curvilinear structured meshes with mild stretching [113] could be implemented on unstructured

meshes as well. However, the one dimensional analysis implies that such a compact scheme is actually quite sensitive to the mesh smoothness. The analysis on the uniform mesh also suggests that the neighbouring nodes should be clustered in a radially symmetric way to make sure that the imaginary component of the transfer function is zero. While smoothness is normally not a problem for a good quality unstructured mesh, the symmetry is rarely satisfied.

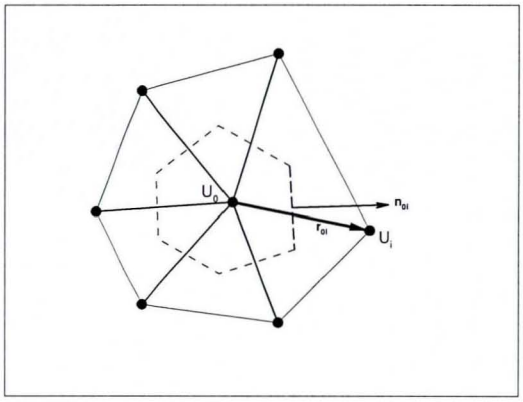


Figure 3.10: A cluster of vertices surrounding an arbitrary vertex 0

To illustrate the problem, consider fig. 3.10 that shows a typical stencil encountered in a computation on an unstructured mesh. From eq. 3.13 and 3.26 the discrete Fourier transform for a flow quantity U at neighbouring vertex i is defined as

$$U_{\mathrm{i}} = U_0 e^{jK(\mathbf{r}_{0\mathrm{i}_x}\cos\theta + \mathbf{r}_{0\mathrm{i}_y}\sin\theta)}$$

To simplify the notation, a complex number $\phi_{0\mathrm{i}} = K(\mathbf{r}_{0\mathrm{i}_x}\cos\theta + \mathbf{r}_{0\mathrm{i}_y}\sin\theta)$ is introduced so that the discrete Fourier transform becomes

$$U_{\mathrm{i}} = U_0 e^{j\phi_{0\mathrm{i}}} \tag{3.52}$$

and

$$\nabla U_{\mathrm{i}} = \nabla U_0 e^{j\phi_{0\mathrm{i}}} \tag{3.53}$$

Substitution of eq. 3.52 and 3.53 into eq. 3.51 and rearranging to collect U_0 and ∇U_0 outside the bracket, yields

$$\nabla U_0 \left(1 - \beta + \beta \sum_{\mathrm{i}} e^{j\phi_{0\mathrm{i}}}\right) = \frac{U_0 \left(m + \sum_{\mathrm{i}} e^{j\phi_{0\mathrm{i}}}\right) \cdot \mathbf{n}_{0\mathrm{i}}}{2\Gamma_0}$$

By recalling that the Euler formula for complex number gives $e^{jz} = \cos z + j \sin z$; $\cos z = 1/2\left(e^{jz} + e^{-jz}\right)$ and $j \sin z = 1/2\left(e^{jz} - e^{-jz}\right)$, it is easy to show that the summation can be rewritten as

$$\sum_{i=1}^{m} e^{j\phi_{0i}} = \cos z_1 + \cos z_2 + \ldots + j\left(\sin z_1 + \sin z_2 + \ldots\right)$$

From the definition given by eq. 3.30, it is quite clear that the diffusion error that is associated with the imaginary component of the numerical wave number transfer function is zero when

$$Re\left(\sum_{i=1}^{m} e^{j\phi_{0i}}\right) = \cos z_1 + \cos z_2 + \ldots = 0$$

which can only be satisfied when m is even. Furthermore, $\mathbf{r}_{0i}$ must be distributed in such a way that they can be paired to so that

$$Im\left(\sum_{i=1}^{m} e^{j\phi_{0i}}\right) = \sin z_1 + \sin z_2 + \ldots$$

is well defined. Unfortunately, these requirements are not generally satisfied by an unstructured mesh. To complicate the matter, failure to meet such a requirement leads to non-zero cross derivatives, which are not found in either uniform equilateral triangular or uniform rectangular mesh. This has not been taken into account in the previous analysis. It will be demonstrated on the later section that these cross derivatives will lead to undesirable results.

3.3.5 A numerical experiment for unstructured meshes

The second experiment has been designed to assess the suitability and ultimately the practicality of the unstructured mesh version of the compact scheme as suggested by ref. [144]. The parameter β is set at 0.5 since this will recover the fourth order accuracy on a uniform equilateral triangular mesh. At this stage, a linear double sine wave convection problem is considered. The convection is defined as

$$\frac{\partial \varphi}{\partial t} = -\frac{\partial \varphi}{\partial x} - 2\frac{\partial \varphi}{\partial y} \tag{3.54}$$

Notice that the convection velocity is unity along the x direction and two along the y direction. The initial scalar field φ describing the double sine wave is defined as follows

$$\varphi = \sin(2\pi x)\sin(2\pi y) \begin{cases} -0.5 \leqslant x \leqslant 0.5 \\ -0.5 \leqslant y \leqslant 0.5 \end{cases} \tag{3.55}$$

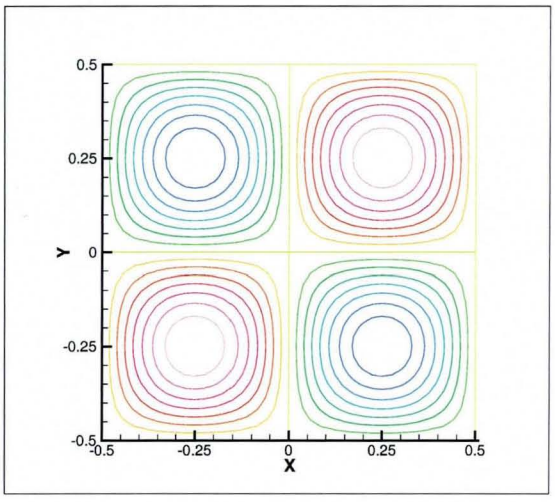


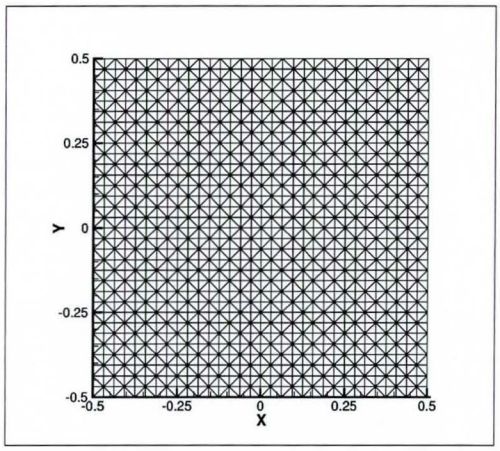
Figure 3.11: Initial contour of U describing a double sine field

which means that the computational domain has one wave length along both x and y directions as shown at fig. 3.11.

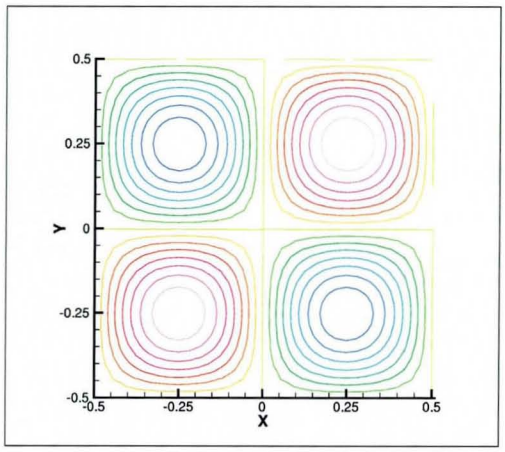
Two anisotropic triangular meshes type are employed: triangulation of a uniform structured mesh, labelled $UK - mesh$; and a non-uniform mesh based on a Delaunay triangulation, labelled $Dl - mesh$, generated using the TRI8IT code [94]. Each type consists of a series of meshes with different numbers of node to assess the refinement effect. All of the meshes are intentionally made coarse to model real LES computations where many of the flow structures are of the same order of magnitude as the cell size.

The baseline computations are carried out based on the derivative form of the governing equation, thus the gradient is employed directly to solve the convection problem. These are labelled compact4 and explicit2. Another computation based on finite volume formulation, labelled FV, is also presented for comparison since this technique is widely used in CFD. This is labelled explicit2(FV). The more detailed FV technique is described in section 4.4.

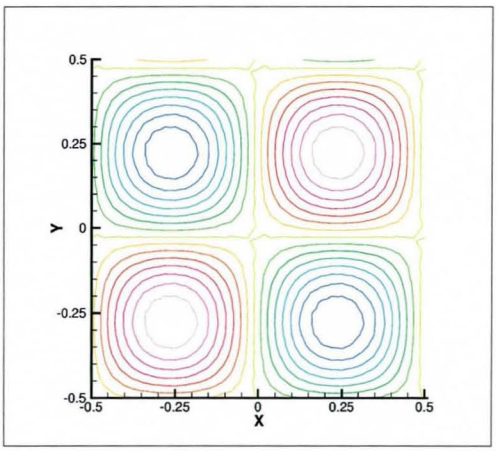
Figures 3.12, 3.13 and 3.14 suggest that some of the results simply show a phase lag whilst others show flow structure deformation in addition to the phase lag. It is quite clear that the compact formulation only shows a clear advantage over the explicit scheme on the uniform anisotropic mesh, namely the $UK - mesh$. However this advantage quickly deteriorates when the more realistic $Dl - mesh$ is employed. This finding is confirmed by table 3.1 that shows the actual accuracy based on the $L1norm$ of the



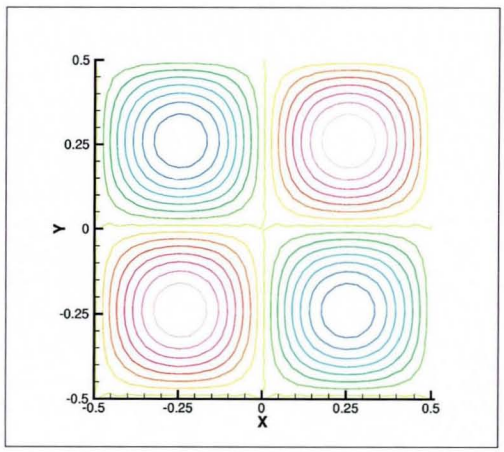
(a) UK-mesh 33×33 vertices



(b) compact4

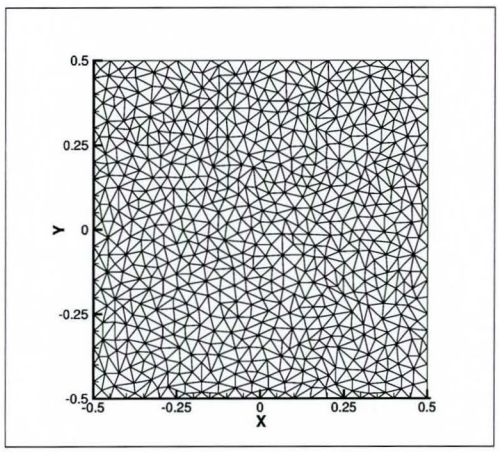


(c) explicit2

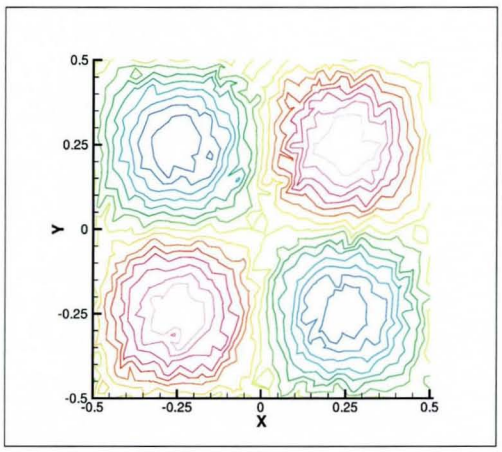


(d) explicit2(FV)

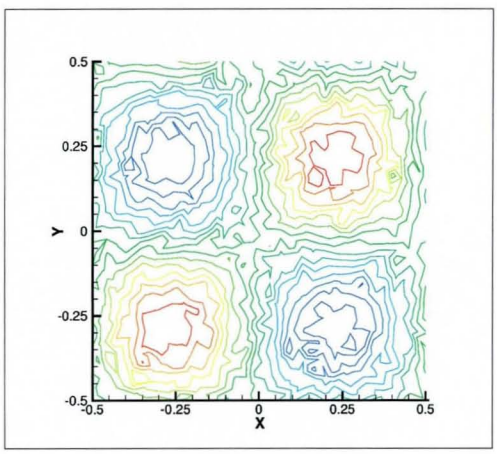
Figure 3.12: U contour at the end of 1 period convection time using UK-mesh



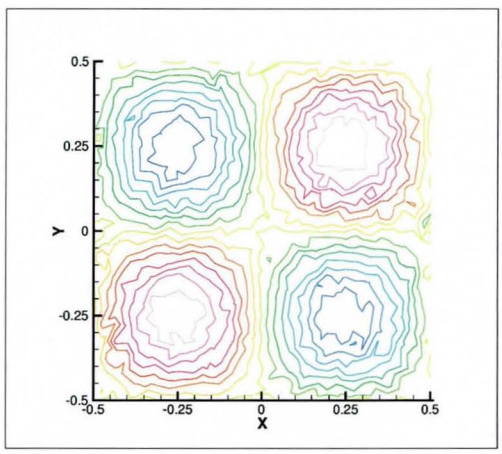
(a) Dl-mesh 745 vertices



(b) compact4



(c) explicit2



(d) explicit2(FV)

Figure 3.13: U contour at the end of 1 period convection time using Dl-mesh

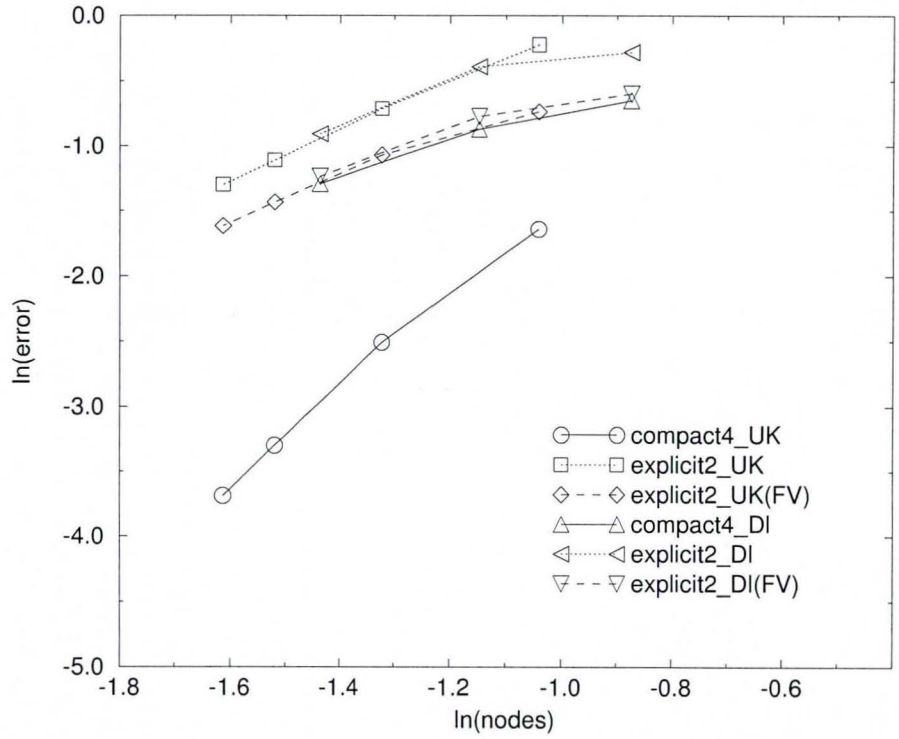


Figure 3.14: L1norm of the error, i.e max error; UK stands for UK-mesh, Dl, stands for Dl-mesh

error criterion. The table also shows that none of the schemes achieve the theoretical accuracy. This is because the mesh resolution for these computations lies on the limit of the numerical tolerance.

	UK-mesh	Dl-mesh
compact4	3.40	1.30
explicit2	1.79	1.12
explicit2(FV)	1.47	1.14

Table 3.1: Order of accuracy based on L1-norm of the error

3.4 Discussion

The implementation of the compact scheme for a structured three dimensional CFD code, which is a pre-requisite for LES, is very straightforward. The implementation of a fourth order compact scheme in an unstructured code is attractive as it only requires the knowledge of the immediate neighbour of a node (or cell, depending on the vertex centred or cell centred formulation), which is compatible with typical data structures. Furthermore, just like Zing and Lomax[144] subsection 3.3.3 has shown that Fourier analysis of linear wave convection using such an unstructured formulation on a uniform equilateral triangular mesh is actually far less sensitive to the direction of the flow (i.e. multidimensional) in comparison to a uniform structured mesh. Closer inspections in subsection 3.3.4, however, revealed some major problems. If the domain is tessellated by a uniform but highly directional triangular mesh the Fourier transform also show a highly directional discretisation. Furthermore, when a typical triangulation algorithm is employed to generate the mesh, such as Delaunay triangulation, one cannot guarantee that all of the neighbouring nodes are equidistant and distributed radially symmetrical about the node of interest. The loss of these properties destroys the high resolution capability of the compact scheme that has been suggested by Zingg and Lomax [144]. In such a case the two obvious alternative candidates to achieve higher order discretisation is either the Discontinuous Galerkin [22, 11] or the more conventional explicit differencing with large stencil [9]. Both of these are significantly more expensive than second order discretisation on an unstructured mesh. The third candidate can be developed by modifying the unstructured mesh stencil.

Observation of Lele's compact scheme, eq. 3.7, shows that the explicit term on the

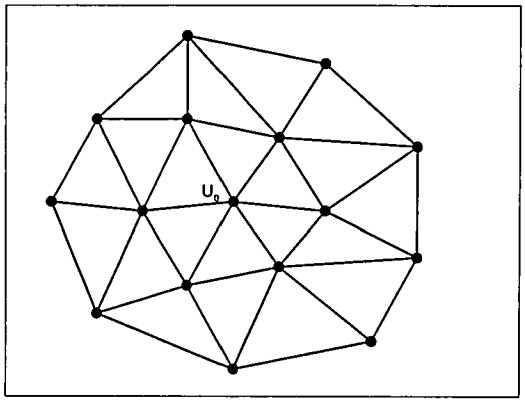
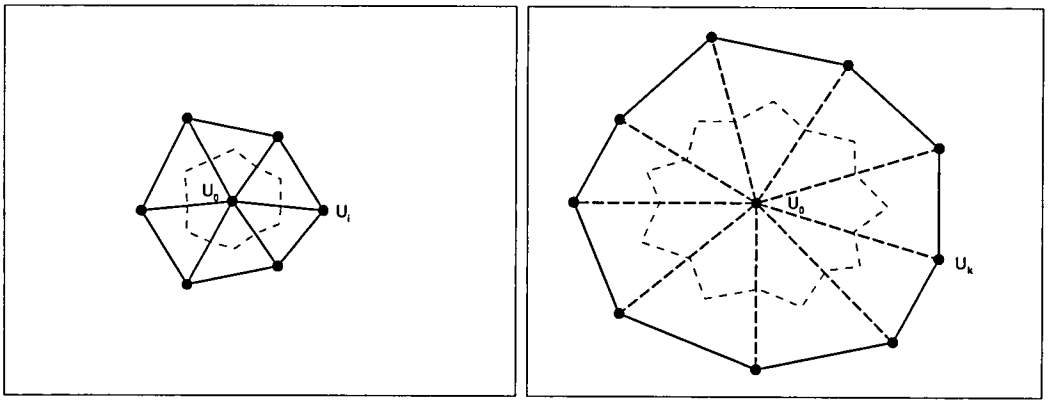


Figure 3.15: Non uniform arbitrary triangular mesh

right hand side consists of several nested stencils that have been utilised by Gamet et al [42] to discretise a non uniform structured mesh. In the light of these observation, the higher order scheme similar to that of Lele can be developed by decomposing the unstructured mesh shown in fig 3.15. into two components: primary and secondary neighbour shown in fig. 3.16



(a) primary median dual (b) secondary median dual

Figure 3.16: Decomposition of non uniform arbitrary triangular mesh of fig. 3.15

In each stencil, the following scheme is applied to calculate the first derivative.

$$\delta_x U_0 + \alpha_1 \sum_i \delta_x U_i = \frac{A_1 \sum_i (U_0 + U_i)(\hat{n}_x)_{0i}\, s_{0i}}{2\Gamma_1} + \frac{B_1 \sum_k (U_0 + U_k)(\hat{n}_x)_{0k}\, s_{0k}}{2\Gamma_2}$$

$$\delta_y U_0 + \alpha_2 \sum_i \delta_y U_i = \frac{A_1 \sum_i (U_0 + U_i)(\hat{n}_y)_{0i}\, s_{0i}}{2\Gamma_1} + \frac{B_1 \sum_k (U_0 + U_k)(\hat{n}_y)_{0k}\, s_{0k}}{2\Gamma_2}$$

where Γ_1 is the area of median dual shown in fig. 3.16(a) and Γ_2 are shown in fig. 3.16(b). It can be easily shown that the summation over the neighbouring nodes i around node 0 is $\sum_i (U_0 + U_i) = \sum_i U_i$, hence the scheme becomes

$$\delta_x U_0 + \alpha_1 \sum_i \delta_x U_i = \frac{A_1 \sum_i U_i \, (\hat{n}_x)_{0i} \, s_{0i}}{2\Gamma_1} + \frac{B_1 \sum_k U_k \, (\hat{n}_x)_{0k} \, s_{0k}}{2\Gamma_2} \tag{3.56}$$

$$\delta_y U_0 + \alpha_2 \sum_i \delta_y U_i = \frac{A_1 \sum_i U_i \, (\hat{n}_y)_{0i} \, s_{0i}}{2\Gamma_1} + \frac{B_1 \sum_k U_k \, (\hat{n}_y)_{0k} \, s_{0k}}{2\Gamma_2} \tag{3.57}$$

In two dimensions, the Taylor expansion can be expressed as

$$U_{i+1} = U_i + \left[\frac{\partial U}{\partial x}\Delta x + \frac{\partial U}{\partial y}\Delta y\right]_i + \frac{\left[\frac{\partial U}{\partial x}\Delta x + \frac{\partial U}{\partial y}\Delta y\right]_i^2}{2!} + \frac{\left[\frac{\partial U}{\partial x}\Delta x + \frac{\partial U}{\partial y}\Delta y\right]_i^3}{3!} + \cdots \tag{3.58}$$

which shows the presence of secondary cross derivatives. In uniform mesh analysis in section 3.3, these derivatives are zero so that they do not play any part on the discretisation, but this is not the case for arbitrary non uniform unstructured meshes. However, for simplicity, the cross derivative term is assumed to be negligible in comparison to the primary derivative. Thus, matching the Taylor expansion constants with that of eq. 3.56 for $\delta_x U_0$ gives

$$1 + \sum_i \alpha_1 = \frac{A_1 \sum_i (x_i - x_0) \, (\hat{n}_y)_{0i} \, s_{0i}}{2\Gamma_1} + \frac{B_1 \sum_k (x_k - x_0) \, (\hat{n}_y)_{0k} \, s_{0k}}{2\Gamma_2} \tag{3.59}$$

and for $\delta_x^2 U_0$

$$1 + \sum_i \alpha_1 \, (x_i - x_0) = \frac{A_1 \sum_i (x_i - x_0)^2 \, (\hat{n}_y)_{0i} \, s_{0i}}{2\Gamma_1} + \frac{B_1 \sum_k (x_k - x_0)^2 \, (\hat{n}_y)_{0k} \, s_{0k}}{2\Gamma_2} \tag{3.60}$$

and also for $\delta_x^3 U_0$

$$1 + \sum_i \alpha_1 \, (x_i - x_0)^2 = \frac{A_1 \sum_i (x_i - x_0)^3 \, (\hat{n}_y)_{0i} \, s_{0i}}{2\Gamma_1} + \frac{B_1 \sum_k (x_k - x_0)^3 \, (\hat{n}_y)_{0k} \, s_{0k}}{2\Gamma_2} \tag{3.61}$$

The y derivatives can be evaluated in similar manner.

If the mesh consists of uniform equilateral triangles or rectangles, the constants of the $\delta_x^2 U_0$ terms are zero in both side of the equation. Thus the system of equations become overdetermined. In this case, α_1 and α_2 can be made equal and then used as a parameter, giving one degree of freedom to the system of equation that has been exploited by eq. 3.50. In ref. [144], the parameter was chosen in such away that the condition for $\delta_x^3 U_0$ is satisfied, giving fourth order differencing as shown in eq. 3.46 and 3.47.

For an arbitrary non-uniform mesh such as shown in fig. 3.15 the values for α_1, A_1, B_1 are generally non zero to ensure fourth order accuracy in x derivatives and similarly α_2, A_2, B_2 are non zero for y derivatives. In other words the stencil for a non-uniform mesh is necessarily larger than that of a uniform mesh. This reflects the findings of Gamet et al. [42] on a non-uniform structured mesh. Therefore the advantage of high order accuracy using a small stencil that underlines the compact scheme is not readily extensible to a non-uniform mesh. However, by considering $\delta_x U_0$ and $\delta_x^2 U_0$ one can drive the system to be overdetermined thus allowing B to be fixed at 0 and use $\alpha_1 = \alpha_2$ as a parameter to obtain third order accuracy. Just as in the two previously mentioned schemes, this scheme also incurs significant additional cost that becomes more severe when a finite volume formulation is used due to the numerical integration on the control volume surface.

Critical analysis of the simple experiments that have been conducted previously also reveal that in the event of the turbulent structures interactions (with each other), the ripple following the vortical structure that was shown by the lower order scheme becomes less important. Such interactions distort the large structures to the extent that they will be broken up to smaller ones within a relatively short period of time. Furthermore, the SGS term would damp the higher wave number. This explains the success of lower order LES codes [96]. The degree of mesh coarsening that has become possible because of the higher resolution of the higher order scheme, as demonstrated by Bassi and Rebay [11] for laminar flow, is ultimately limited by the turbulent scales that need to be resolve in the LES. This qualitative, and also intuitive, argument does suggest that for practical reason a second order scheme for performing LES on unstructured meshes should be acceptable.

3.5 Closure

It has been demonstrated, both theoretically and numerically, that high order numerical schemes for the convective term do have a high resolution property that leads to small dispersion error when an anisotropic uniform mesh is employed. As far as LES is concerned this property is very desirable as one can guarantee that the SGS model is not contaminated by the numerical error. The properties exhibited by Lele's compact scheme are particularly good so that one can argue it is worth the extra computing cost that arise from the implicit formulation. However, such an optimistic view is hardly applicable to a multidimensional formulation that is required by an unstructured flow

solver, where the higher order scheme can only be implemented at a significant additional cost. Realising that the large number of cells that is coupled with a large number of iteration required for LES means that this additional cost can only be justifiable when access to high speed supercomputer is available. Even when such a machine is available, it seems to be more fruitful to exploit the computing power for solving a more realistic problem involving larger number of vertices using a lower order method rather than a higher order method on a simpler problem.

Chapter 4

Grid Transparent Numerical Method

4.1 Introduction

The general format of the fluid flow governing equations (2.1) (including the PDE of the turbulence model described by eq. A.15) can be written as follows

$$\frac{\partial}{\partial t}\iiint_{\Gamma}\mathbf{Q}dV+\iint_{\partial\Gamma}\mathbf{F}(\mathbf{Q})\cdot\hat{n}dS+\iint_{\partial\Gamma}\mathbf{G}(\mathbf{Q})\cdot\hat{n}dS=\iiint_{\Gamma}\mathbf{S}dV \tag{4.1}$$

where $\mathbf{Q}$ is the state vector, $\mathbf{F}$ is the inviscid flux vector, $\mathbf{G}$ is the viscous flux vector and $\mathbf{S}$ is the source term vector. It has been shown in Chapter 2 and Appendix A that in a conservative system, the source terms are non-zero only for the turbulence model.

It has been argued in sections 3.3.4 and 3.4 that higher than second order scheme is impractical for CFD on a general unstructured mesh using finite volume technology, in spite of its potentially high resolution capability. Thus, a second order spatial scheme is employed for the present work. The second order approximation of a volume integral is evaluated using a single quadrature point located at the control volume centroid and the mean value of the property in the control volume. This is equivalent to the mass lumping technique in the finite element method. In a median dual cell, the vertices do not in general lie at the centroid of the dual cell. The problem can be reduced by carefully generating smooth unstructured meshes.

The discretised governing equation integrated over a control volume with volume Γ

and m number of enclosing surfaces $\partial\Gamma$ can be written as follows :

$$D_t\mathbf{Q}\Gamma + \sum_{j=1}^{m}\mathbf{F}(\mathbf{Q})\cdot\hat{n}\partial\Gamma_j + \sum_{j=1}^{m}\mathbf{G}(\mathbf{Q})\cdot\hat{n}\partial\Gamma_j = \mathbf{S}(\mathbf{Q})\Gamma \tag{4.2}$$

where D_t is the time derivative operator.

The discretised equation can be simplified further by grouping the fluxes and source term in a single residual matrix $\mathbf{R}$ to form a discrete ODE [57].

$$D_t\mathbf{Q} = \mathbf{R}(\mathbf{Q}) \tag{4.3}$$

where

$$\mathbf{R}(\mathbf{Q}) = \frac{1}{\Gamma}\left[-\sum_{j=1}^{m}\mathbf{F}(\mathbf{Q})\cdot\hat{n}\partial\Gamma_j - \sum_{j=1}^{m}\mathbf{G}(\mathbf{Q})\cdot\hat{n}\partial\Gamma_j + \mathbf{S}(\mathbf{Q})\Gamma\right] \tag{4.4}$$

4.2 Data Structure

Figure 4.1 shows two adjacent primal cells, one is a tetrahedron formed by nodes `v1-v2-v3-v4` and the other is a triangular prism formed by `v1-v2-v3.v6-v5-v7`. Centroids of the primal cells are at `c1` and `c2`. `f1,f2,f3,f4`,and `f5` are centroids of the faces of the primary cells. `e12,e13,e14,e16` are midpoints of their respective edges. Two median dual components surrounding `v1` that are contributed by the tetrahedron and triangular prism are drawn as two shaded hexahedra, `v1-e12-f2-e13.e14-f1-c1-f3` and `v1-e12-f2-e13.e16-f4-c2-f5`, respectively. If `v1` is not a boundary node, it would be enclosed by an irregular polygon of the complete dual that is formed by contribution from all of surrounding cells. It can be seen clearly that each primal cell, independent of its geometric shape, contributes a hexahedron, formed by the vertex of interest, midpoints of the edges that emanate from the vertex, the centroid of the cell and their respective faces. This indicates a property of a median dual cell that can be exploited to obtain a grid-transparent method. It will be shown later that this property leads to a major simplification, that allows only an edge based data structure to be used

In the edge based data structure, an array to store edge-to-vertices information, `e2v` is used. Any edge `ie` is defined as going from `v1` to `v2`.

```
e2v(1,ie) = v1
e2v(2,ie) = v2
```

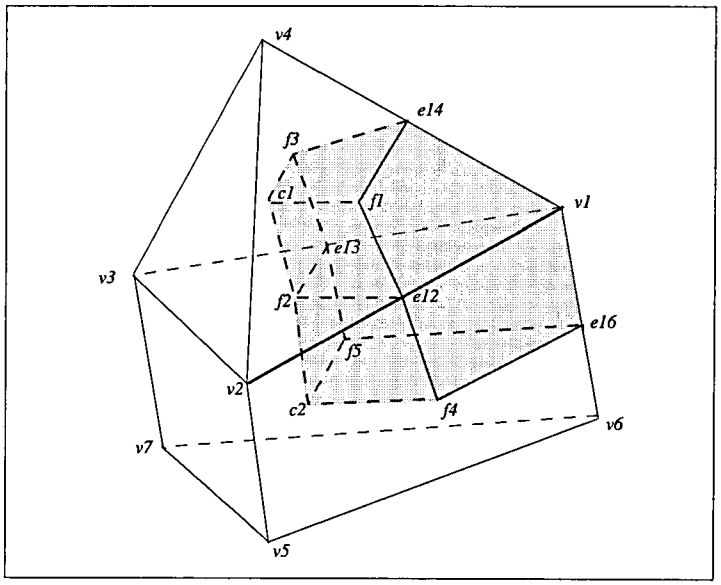


Figure 4.1: Median dual cell

The bold edge between v1-v2 in fig. 4.1 is used as an illustration of how this data structure is formed in the following discussion.

4.2.1 Surface of the Median Dual

Recall that the discrete finite volume formulation (eq. 4.2) requires that each edge connecting two vertices can be associated to a median dual surface. Thus the surface information is stored in the following array that is linked to e2v.

$$\begin{aligned} \texttt{eface(1,ie)} &= \mathbf{n}_{\mathrm{ij}_x} \\ \texttt{eface(2,ie)} &= \mathbf{n}_{\mathrm{ij}_y} \\ \texttt{eface(3,ie)} &= \mathbf{n}_{\mathrm{ij}_z} \end{aligned}$$

Since the present method is designed to be second order accurate, the surface integral can be evaluated at a single quadrature point. Thus the facets surrounding an edge can be lumped together to save storage space. A linear approximation of the face normal associated to an edge $\overline{\mathrm{ij}}$, $\mathbf{n}_{\mathrm{ij}}$, is obtained by vectorial addition of all facet normals, $\mathbf{n}_{\mathrm{ik}}$, that share the edge. Notice that the normals are defined pointing away from v1. The procedure is clearly shown by the two dimensional representation of figure 4.2 (a). The

2D sketch also illustrates the flux along edge `v1-v2` would only see the projected area of the surface in the direction perpendicular to the edge, which is unchanged by the linear approximation.

$$\mathbf{n}_{ij} = \sum_{k=1}^{k \cap \bar{i}\bar{j}} \mathbf{n}_{ik} \tag{4.5}$$

In terms of the unit normal $\hat{n}$ and surface area s, the linear approximation is written as:

$$\hat{n}_{ij} s_{ij} = \sum_{k=1}^{k \cap \bar{i}\bar{j}} n_{ik} s_{ik} \tag{4.6}$$

In two dimensions, the normal of each segment can be obtained easily by rotating the edge vector through 90deg.

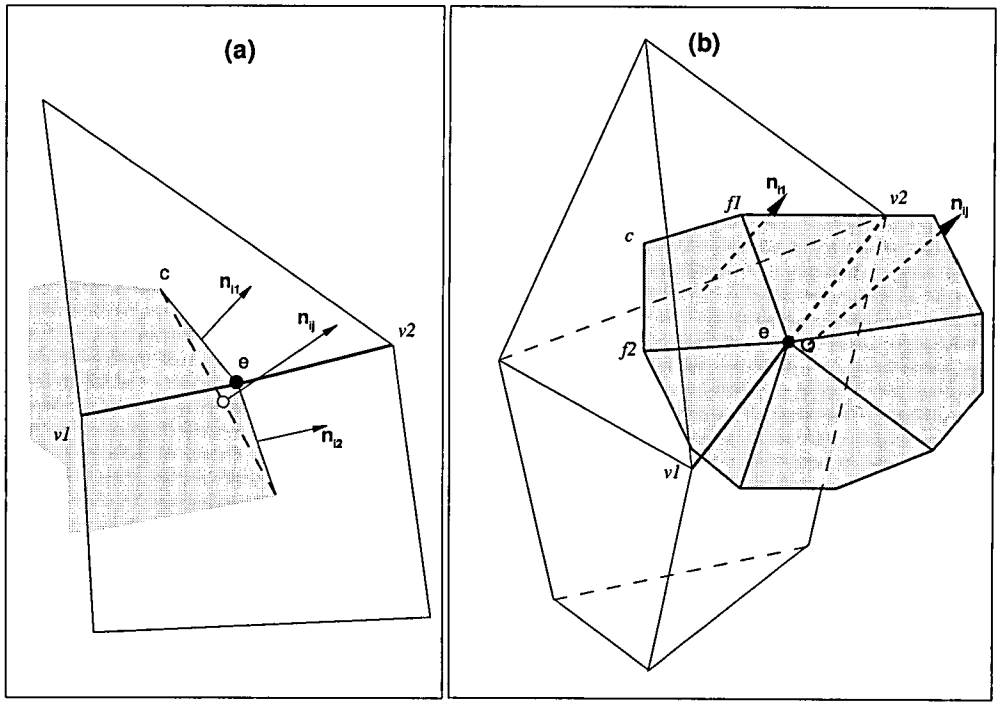


Figure 4.2: Surface of median dual cell about an internal edge

In three dimensions, each of the facets has a quadrilateral topology irrespective of the primary cell's topology. Figure 4.1 illustrates part of this surface as a combination of two quadrilaterals `c1-f2-e12-f1` and `c2-f4-e12-f2`. Figure 4.2 (b) shows a typical face surrounding an internal edge. To simplify the discussion, consider only the facet

that is contributed by the top tetrahedron. It is formed by connecting a midpoint edge (e), two centroid faces (f_1 and f_2) and the centroid of the tetrahedron cell (c). The normal of each segment is calculated by the cross product of its two diagonals, $\vec{d_1}$ and $\vec{d_2}$. For instance, at fig. 4.2 (b), n_{i1} is computed from $\vec{d_1} = \overrightarrow{ec}$ and $\vec{d_2} = \overrightarrow{f_1 f_2}$ as:

$$n_{ik} = \vec{d_1} \times \vec{d_2} \tag{4.7}$$

The quadrature point obtained from the above vector addition should be located at the median dual surface centroid, which is marked by "o" in fig 4.2. However, for convenience, it is approximated by the edge midpoint, which is symbolised "•" in the illustration. Provided that cells surrounding the edge do not vary widely, the associated error should not be significant

4.2.2 Volume of Median Dual

In two dimensions, the dual area surrounding a vertex in a triangle is always a third of the total area of the triangles. For tetrahedra, it is a fourth of the volume. Unfortunately there is no such fixed ratio for quadrilateral and prismatic cell. Figure 4.3 shows an alternative method of computing the control volume by looping over the edges. The quadrilateral formed by two cell centroids and two vertices sharing an edge can be divided into two triangles (dark and light shade) with identical area (see fig.4.3 a). This can be generalised for two and three dimensional meshes easily. Upon inspection of fig.4.1 and 4.2, the median dual control volume can be approximated by a set of arbitrary polygonal cones associated to each edge. To illustrate this, the edge `v1-v2` of figure 4.2 and its associated surface is isolated in space. Two polygonal cones with identical bases can then be drawn by connecting all nodes of the median dual surface to the edge's end (see fig.4.3 b)

The median dual control volume is therefore approximated as follows

$$V_i = \sum_{\mathtt{ie}=1}^{\mathtt{ie}\cap i} dV_{\mathtt{ie}} \tag{4.8}$$

where in 2D the segment $dV_{\mathtt{ie}}$ is calculated as a quarter of the rectangle formed by the dot product of surface normal $\mathbf{n_{ij}}$ and edge $\vec{\mathbf{ij}}$ vectors

$$dV_{\mathtt{ie}} = \frac{1}{2}\left(\frac{\mathbf{n_{ij}} \cdot \vec{\mathbf{ij}}}{2}\right) \tag{4.9}$$

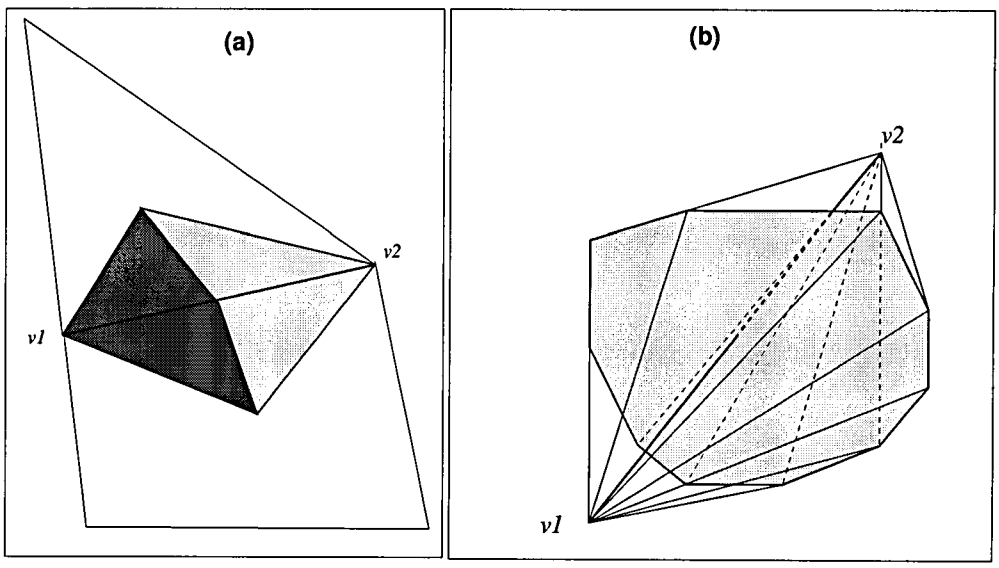


Figure 4.3: Component of median dual volume about an edge

and in 3D the segment $dV_{\mathtt{ie}}$ is calculated as a sixth of the polygonal cylinders formed by the dot product of the surface normal $\mathbf{n}_{\mathrm{ij}}$ and edge $\vec{\mathbf{ij}}$ vectors.

$$dV_{\mathtt{ie}} = \frac{1}{2}\left(\frac{\mathbf{n}_{\mathrm{ij}} \cdot \vec{\mathbf{ij}}}{3}\right) \tag{4.10}$$

4.2.3 Median Dual Control Volume on the Domain Boundary

Median dual control volumes are formed around each boundary nodes in a similar manner to those around internal nodes. Here, boundary fluxes are computed on the dual face about a boundary node. All boundary node indices are stored in a single array `bVertices(1:bound_node_max)`.

A second list is created to access data in `bVertices` using a CSR (Compact Sparse Row)-like technique [98]. Variable `vstart` indicates a starting position `ib` value of a group of nodes listed within `bVertices`. Variable `nverts` indicates a jump to the the starting `ib` value of the next group. The boundary type, which will be shown later in detail, is defined by `btype`.

```
DO indx = 1,nboundary_region
  bRegion(1,indx) = btype
```

```
  bRegion(2,indx) = vstart
  bRegion(3,indx) = nverts
ENDDO

DO ib=bRegion(2,indx),bRegion(2,indx)+bRegion(3,indx)-1
  bVertices(ib) = v
ENDDO
```

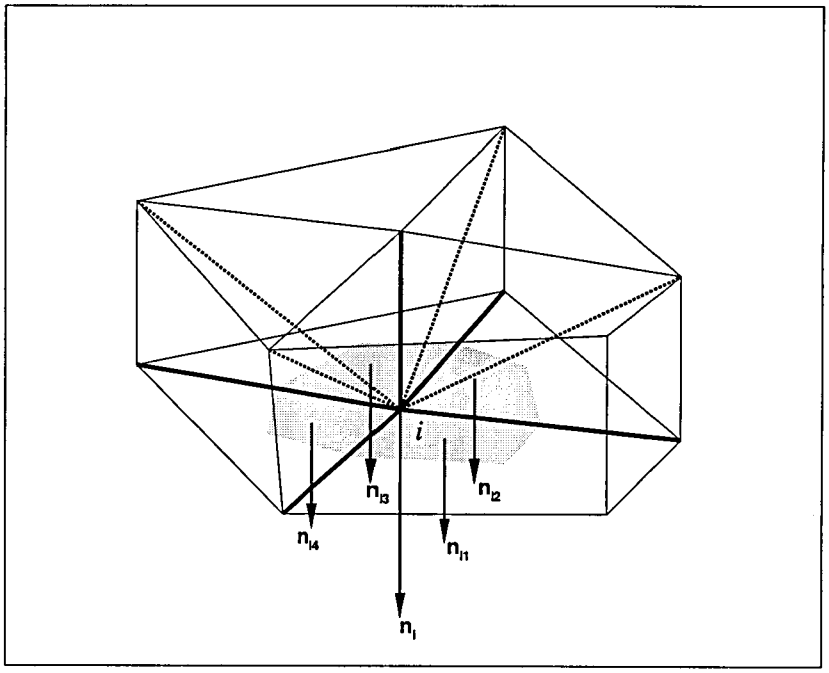


Figure 4.4: Median dual around a boundary node

Figure 4.4 shows a median dual face about boundary node i. The bold lines indicate edges that connect surrounding nodes used to reconstruct the gradient at the boundary node using a least-squares method. The dashed lines are virtual wall edges to improve the gradient reconstruction. See section 4.4.3.

Here, the face is constructed in similar way to section 4.2.1. The normal of the face, which points outside of the computational domain, is stored in the `bNorm` list which is linked to `bVertices`.

$$\texttt{bNorm(1,ib)} = \mathbf{n}_{i_x}$$

$$\texttt{bNorm(2,ib)} = \mathbf{n}_{i_y}$$
$$\texttt{bNorm(3,ib)} = \mathbf{n}_{i_z}$$

The technique illustrated by fig. 4.4 works well for any boundary surface with smooth curvature. However if any one of the segments makes an acute angle to the others, such as in a wing trailing edge, a careful treatment is required. Although this is not normally a serious problem in viscous flows, it can be serious for inviscid flows with slip-wall boundary condition. This limitation will be illustrated by the inviscid calculation shown in chapter 5.

4.2.4 Cyclic and Periodic Boundary Data Structure

When two boundary surfaces are linked by a cyclic or periodic boundary, an additional data structure is provided to connect the two surfaces topologically. Thus, the boundary nodes on these surfaces in effect become internal nodes. The simplest, although not the cheapest, way to achieve this is by introducing virtual vertices and virtual edges around the boundary nodes.

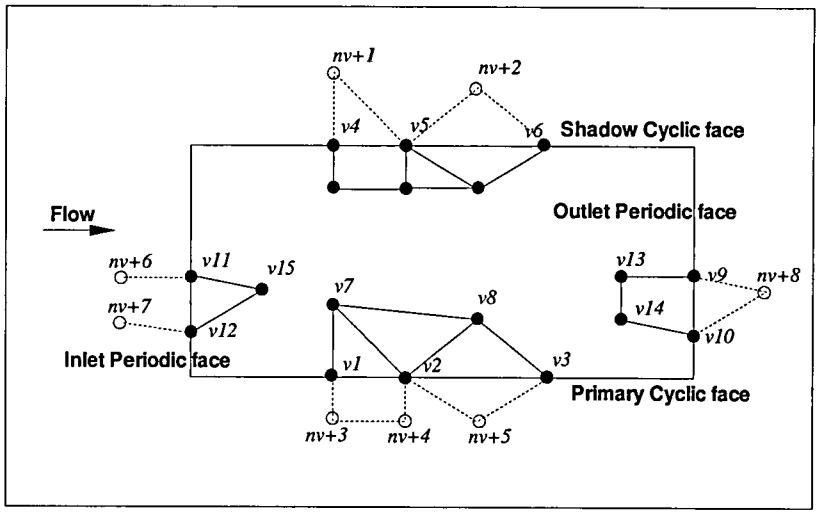


Figure 4.5: Virtual cell and edges of cyclic and periodic boundaries

Figure 4.5 illustrates the definition of virtual nodes and edges which are stored as an extension to the physical vertex list and `e2v` array. The normal vectors of the control volume surface associated to these virtual edges are also stored in an extension of the `eface` list. To simplify flux reconstruction, the virtual edge is arranged such that it goes

from the physical to the virtual node. An additional data structure is used to register the vertices pairing.

```
DO indx = 1,nPC_region
  Aliasing(1,indx) = PCtype    /* cyclic or periodic */
  Aliasing(2,indx) = vstart
  Aliasing(3,indx) = nverts
ENDDO

DO ib=bRegion(2,indx),bRegion(2,indx)+bRegion(3,indx)-1
  AliasingV(ib,1) = v_primary_face
  AliasingV(ib,2) = v_shadow_face
ENDDO
```

The strategy is similar to the data structure for other boundary conditions. `AliasingV` array lists all of the pairing vertices, whereas `Aliasing` array holds information on how to access the vertex list using CSR like method.

4.3 Time Advancement technique

The ODE of eq. 4.3 is solved by explicit time integration based on a low storage multi stage Runge-Kutta (RK) scheme [57]. Unlike the traditional m stage RK scheme that requires m-storage spaces for each variables at each node simultaneously [64], the low storage version only requires 2 storage spaces. This saving comes at the expense of accuracy. A 4 stage traditional RK scheme is 4^{th} order accurate in time whereas the low storage counterpart that is widely used in CFD is only 2^{nd} order. The low storage R-K scheme was used in conjunction with a local time step to accelerate the convergence of the steady solver.

$$\mathbf{Q}^{(1)} = \mathbf{Q}^n + 0.0833\Delta t\mathbf{R}(\mathbf{Q}^n)$$

$$\mathbf{Q}^{(2)} = \mathbf{Q}^n + 0.2069\Delta t\mathbf{R}(\mathbf{Q}^{(1)})$$

$$\mathbf{Q}^{(3)} = \mathbf{Q}^n + 0.4625\Delta t\mathbf{R}(\mathbf{Q}^{(2)})$$

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n + 1.0000\Delta t\mathbf{R}(\mathbf{Q}^{(3)})$$

The above scheme originates from the precursor of the present work [50] and is retained for convenience. However as the work on the LES code started, it was found that a faster and higher order accurate RK scheme is readily available in the literature, namely the 3 stage RK scheme of Gottlieb and Shu[46], which is 3^{rd} order accurate.

$$\mathbf{Q}^{(1)} = \mathbf{Q}^n + \Delta t \mathbf{R}(\mathbf{Q}^n)$$

$$\mathbf{Q}^{(2)} = \frac{3}{4}\mathbf{Q}^n + \frac{1}{4}\mathbf{Q}^{(1)} + \frac{1}{4}\Delta t \mathbf{R}(\mathbf{Q}^{(1)})$$

$$\mathbf{Q}^{n+1} = \frac{1}{3}\mathbf{Q}^n + \frac{2}{3}\mathbf{Q}^{(2)} + \frac{2}{3}\Delta t \mathbf{R}(\mathbf{Q}^{(2)})$$

4.4 Convective Numerical Schemes

There are three main issues in handling a convective numerical scheme. Firstly, how to approximate the fluid properties on the control volume surface from the information stored at the control volume nodes. The second issue is the discrete convective flux approximation that ensure accuracy as well as robustness. The third issue is quadrature points to perform numerical integration over the surface. The last issue has been discussed earlier by employing a single quadrature point located approximately at mid edge since the present work is only 2^{nd} order accurate in space. This section deals with the other two issues.

4.4.1 Convective Flux

The present work employs the Roe Flux Difference Splitting [97] approximate Riemann solver that has been widely regarded as a robust and accurate technique for RANS problems. The method is chosen because its strengths and limitations have been well documented [53].

At the face, left $\mathbf{Q}_{ij}^L$ and right $\mathbf{Q}_{ij}^R$ states are computed from extrapolation. The flux is then computed as an average of the flux computed from the left and right states and an artificial dissipation to provide upwind behaviour.

$$\mathbf{F}_{ij} = \frac{1}{2}\left[\mathbf{F}(\mathbf{Q}_{ij}^L) + \mathbf{F}(\mathbf{Q}_{ij}^R) - |\widetilde{\mathbf{A}}|(\mathbf{Q}_{ij}^R - \mathbf{Q}_{ij}^L)\right] \tag{4.11}$$

where $\widetilde{\mathbf{A}} = \frac{\partial \mathbf{F}}{\partial \widetilde{\mathbf{Q}}}$ is the so called Roe matrix, that provides the artificial dissipation, and is calculated from characteristic waves. Since in multi dimensional problems there are

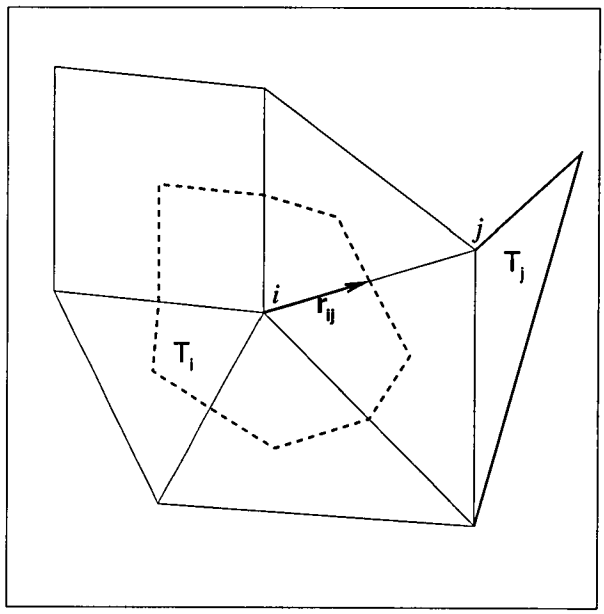


Figure 4.6: Reconstruction of variables on control volume surface

an infinite number of such waves, it is a common practice in a finite volume method to consider only the wave that is normal to control volume surface (i.e. the contravariant velocity component U_n), thus the flux computation is treated locally one dimensionally.

From linear algebra, the matrix $\widetilde{\mathbf{A}}$ can be written in term of its right eigenvector $\widetilde{\mathbf{R}}$ and its inverse that is usually called the left eigenvector $\widetilde{\mathbf{R}}^{-1}$ and a diagonal matrix consisting of its eigenvalues $\widetilde{\lambda}$.

$$|\widetilde{A}| = \widetilde{\mathbf{R}}|\widetilde{\lambda}|\widetilde{\mathbf{R}}^{-1} \tag{4.12}$$

$$\widetilde{\lambda} = \begin{pmatrix} \widetilde{\lambda}_1 & 0 & 0 & 0 & 0 \\ 0 & \widetilde{\lambda}_2 & 0 & 0 & 0 \\ 0 & 0 & \widetilde{\lambda}_3 & 0 & 0 \\ 0 & 0 & 0 & \widetilde{\lambda}_4 & 0 \\ 0 & 0 & 0 & 0 & \widetilde{\lambda}_5 \end{pmatrix} \tag{4.13}$$

where $|\widetilde{\lambda}_{1,2,3}| = \widetilde{U}_n$, $|\widetilde{\lambda}_4| = \widetilde{U}_n + \widetilde{a}$ and $|\widetilde{\lambda}_5| = \widetilde{U}_n - \widetilde{a}$.

By simplifying the contribution from the first three identical eigenvalues, the dissipation can now simply be written as follows

$$|\widetilde{A}|(\mathbf{Q}_{\mathbf{ij}}^R - \mathbf{Q}_{\mathbf{ij}}^L) = |\widetilde{A}_1| + |\widetilde{A}_4| + |\widetilde{A}_5| \tag{4.14}$$

where

$$|\widetilde{A}_1| = |\widetilde{\lambda}_1| \left[\left(\Delta\rho - \frac{\Delta p}{\widetilde{c}^2} \right) \begin{pmatrix} 1 \\ \widetilde{u} \\ \widetilde{v} \\ \widetilde{w} \\ \frac{\widetilde{U}^2}{2} \end{pmatrix} + \widetilde{\rho} \begin{pmatrix} 0 \\ \Delta u - \hat{n}_x \Delta U_n \\ \Delta v - \hat{n}_y \Delta U_n \\ \Delta w - \hat{n}_z \Delta U_n \\ \widetilde{u}\Delta u + \widetilde{v}\Delta v + \widetilde{w}\Delta w - \widetilde{U}_n \Delta U_n \end{pmatrix} \right] \tag{4.15}$$

$$|\widetilde{A}_{4,5}| = |\widetilde{\lambda}_{4,5}| \frac{\Delta p \pm \widetilde{\rho}\widetilde{c}\Delta U_n}{2\widetilde{a}^2} \begin{pmatrix} 1 \\ \widetilde{u} \pm \hat{n}_x \widetilde{a} \\ \widetilde{v} \pm \hat{n}_y \widetilde{a} \\ \widetilde{w} \pm \hat{n}_z \widetilde{a} \\ \widetilde{h} \pm \widetilde{U}_n \widetilde{a} \end{pmatrix} \tag{4.16}$$

where $\hat{n}$ is the face unit normal; $a = \sqrt{\gamma \frac{p}{\rho}}$ is the local sound speed; $h = \frac{\gamma}{\gamma - 1} \frac{p}{\rho}$ is the local enthalpy. The difference Δ and Roe averaged $\widetilde{\cdot}$ operators are defined as follows

$$\Delta = (\cdot)^R_{ij} - (\cdot)^L_{ij}$$

$$\widetilde{(\cdot)} = \frac{(\cdot)^L_{ij}\sqrt{\rho^L_{ij}} + (\cdot)^R_{ij}\sqrt{\rho^R_{ij}}}{\sqrt{\rho^L_{ij}} + \sqrt{\rho^R_{ij}}}$$

Here the speed U is simply defined as $U = u^2 + v^2 + w^2$. The difference contravariant velocity is $\Delta U = \Delta u \hat{n}_x + \Delta v \hat{n}_y + \Delta w \hat{n}_z$. The Roe averaged contravariant velocity is $\widetilde{U} = \widetilde{u}\hat{n}_x + \widetilde{v}\hat{n}_y + \widetilde{w}\hat{n}_z$

The Roe scheme was originally designed for Euler and RANS steady state computation involving strong shock waves. As such, it may be too dissipative for LES and could cause excessive decaying of eddies. Based on numerical investigations of several compressible numerical schemes with numerical dissipation, Garnier et al. [43] suggests that a numerical switch that can distinguish between shock and turbulence is desirable. Thus the Roe flux is modified by introducing a switch Φ to control the artificial dissipation.

$$\mathbf{F}_{ij} = \frac{1}{2}\left[\mathbf{F}(\mathbf{Q}^L_{ij}) + \mathbf{F}(\mathbf{Q}^R_{ij}) - \Phi|\widetilde{\mathbf{A}}|(\mathbf{Q}^R_{ij} - \mathbf{Q}^L_{ij})\right] \tag{4.17}$$

Yee et al [138] proposed a characteristic switch for Φ based on the TVB formulation of Harten. The switch can be formulated quite easily and relatively cheaply in a structured mesh. It was proposed to evaluate the switch only at the end of time step when multistage time stepping such as R-K is employed. However in an unstructured mesh the TVB switch becomes expensive, as it involves evaluation of $\widetilde{\mathbf{R}}^{-1}(\mathbf{Q}^R_{ij} - \mathbf{Q}^L_{ij})$.

Ducros et al. [29] who employ the scalar dissipation of the JST scheme [57] defined Φ as a scalar function that compares velocity divergence and vorticity.

$$\Phi = \frac{(\nabla \cdot \mathbf{u})^2}{(\nabla \cdot \mathbf{u})^2 + \Omega^2 + \epsilon} \tag{4.18}$$

The rational is that in the neighbourhood of a shock wave, the velocity divergence will dominate the switch to recover the original artificial dissipation, whereas in the turbulence region, the vorticity will minimise the dissipation. ϵ is a small number to prevent division by zero. Currently, it is set as 10^{-20}.

It is felt that the switch proposed in [29] is quite aggressive in the sense that dissipation can be scaled down close to zero. Bui [17] has investigated that there is a minimum value that can be allowed to ensure stability in channel flow simulation. Thus for the present work (4.18) is modified as follows

$$\Phi = \max\left(\varepsilon, \frac{(\nabla \cdot \mathbf{u})^2}{(\nabla \cdot \mathbf{u})^2 + \Omega^2 + \epsilon}\right) \tag{4.19}$$

where ε is used to control the lower threshold value of the switch; when set to zero, the original switch is recovered. The switch can be easily modified to make it biased (.i.e. more sensitive) towards the vorticity as follows:

$$\Phi = \max\left(\varepsilon, \frac{(\nabla \cdot \mathbf{u})^2}{(\nabla \cdot \mathbf{u})^2 + \Lambda\Omega^2 + \epsilon}\right) \tag{4.20}$$

where Λ is an adjustable constant to set the vorticity weighting. The switch at edge $\overline{ij}$ is calculated as $\Phi_{ij} = \max(\Phi_i, \Phi_j)$ since (4.19) is evaluated at the nodes rather than edges. Notice, the more elaborate gradient calculation for diffusive flux is not used here.

As a shorthand for later discussions, the switch based on eq. 4.19 is called the Ω-switch at the remainder of the thesis.

4.4.2 Reconstruction of flow property on control volume face

In structured mesh methodologies, the Monotone Upstream-centred Schemes for Conservation Laws (MUSCL) of van Leer [130] is generally regarded as a reliable way to perform higher order reconstruction. In this work the simplified version of MUSCL scheme that has been proposed by Barth and Jespersen [10], which is identical to the so called Fromm scheme in one dimension [130], is employed. Thus variable reconstruction on control volume surfaces is defined as follows:

$$\mathbf{Q}_{ij}^{L} = \mathbf{Q}_i + \nabla\mathbf{Q}_i \cdot \frac{1}{2}\mathbf{r}_{ij} \tag{4.21}$$

$$\mathbf{Q}_{ij}^{R} = \mathbf{Q}_j - \nabla\mathbf{Q}_j \cdot \frac{1}{2}\mathbf{r}_{ij} \tag{4.22}$$

Notice that the left and right interpolation is actually defined by the direction of $\mathbf{r}_{ij}$. The negative sign on the right hand side of eq. 4.22 is caused by the definition $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$. This scheme is very attractive since it requires very small stencil, assuming the nodal gradients are known. To simplify the discussion, only a generic formula based on the left state reconstruction is used for the remainder of the chapter.

Unfortunately, Godunov's Theorem states that only first-order scheme can be monotone [53]. Therefore, it is necessary to introduce a non-linear gradient limiter function in order to force the reconstruction scheme to become first order around flow discontinuities whilst still producing higher order reconstruction everywhere else. Hence the MUSCL scheme is modified as follows

$$\mathbf{Q}_{ij} = \mathbf{Q}_i + \Psi_i\nabla\mathbf{Q}_i \cdot \frac{1}{2}\mathbf{r}_{ij} \tag{4.23}$$

where Ψ_i is the limiter function at node i.

Venkatakrishnan's limiter function [132] has been chosen for the present work since it does not enforce monotonicity too strictly. Like that of Barth and Jespersen [10], the limiter is multidimensional. Whilst this property is desirable for an unstructured mesh, it is rather expensive since additional interpolation is required to compute the limiter function.

$$\varphi_{ij} = \varphi_i + \frac{1}{2}\nabla\varphi_i \cdot \mathbf{r}_{ij} \tag{4.24}$$

$$\varphi_{max} = \max\left(\varphi_i, \varphi_j\right) \qquad \forall\bar{ij} \cap i \tag{4.25}$$

$$\varphi_{min} = \min\left(\varphi_i, \varphi_j\right) \qquad \forall\bar{ij} \cap i \tag{4.26}$$

$$\Delta_{1,max} = \varphi_{max} - \varphi_i \tag{4.27}$$

$$\Delta_{1,min} = \varphi_{min} - \varphi_i \tag{4.28}$$

$$\Delta_2 = \varphi_{ij} - \varphi_i \tag{4.29}$$

$$\Psi_{ij}\left(\varphi\right) = \begin{cases} \frac{1}{\Delta_2}\frac{\left(\Delta_{1,max}^2+o^2\right)+2\Delta_2^2\Delta_{1,max}}{\Delta_{1,max}^2+2\Delta_2^2+\Delta_{1,max}\Delta_2+o^2} & \text{if} \quad \Delta_2 > 0 \\ \frac{1}{\Delta_2}\frac{\left(\Delta_{1,min}^2+o^2\right)+2\Delta_2^2\Delta_{1,min}}{\Delta_{1,min}^2+2\Delta_2^2+\Delta_{1,min}\Delta_2+o^2} & \text{if} \quad \Delta_2 < 0 \\ 1 & \text{if} \quad \Delta_2 = 0 \end{cases} \tag{4.30}$$

$$\Psi_i\left(\varphi\right) = \min\Psi_{ij}\left(\varphi\right) \qquad \forall\bar{ij} \cap i \tag{4.31}$$

where $o^2 = (\mathcal{K}L)^3$, $\mathcal{K}$ is an adjustable constant to tune the limiter sensitivity, and L is the local mesh spacing. In the present work L is taken as the cube root of the control volume.

This limiter function will only be applied for cases where robustness become a major issue as it can contaminate the solution field. This is particularly important for LES computation since the limiter function is known to be incapable of distinguishing between a high velocity gradient that is caused by flow discontinuity and vorticity [43].

4.4.3 Gradient Reconstruction

There are two simple methods that can be used to reconstruct gradients for equations 4.21 and 4.22 , namely the Green-Gauss theorem and the Least-Squares method. The former is analytical and can be reduced to second order central difference in one dimension. The later is heuristic as it is based on the following assumption :

$$q_j = q_i + \left.\frac{\partial q}{\partial x}\right|_i (x_j - x_i) + \left.\frac{\partial q}{\partial y}\right|_i (y_j - y_i) + \left.\frac{\partial q}{\partial z}\right|_i (z_j - z_i) \qquad \forall \bar{ij} \cap i \tag{4.32}$$

For an unstructured grid, such as illustrated in fig. 4.6, the above equation can be written in the following form:

$$\begin{pmatrix} \Delta x_{1i} & \Delta y_{1i} & \Delta z_{1i} \\ \Delta x_{2i} & \Delta y_{2i} & \Delta z_{2i} \\ \vdots & \vdots & \vdots \\ \Delta x_{Ni} & \Delta y_{Ni} & \Delta z_{Ni} \end{pmatrix} \left\{ \begin{array}{c} \frac{\partial q}{\partial x} \\ \frac{\partial q}{\partial y} \\ \frac{\partial q}{\partial z} \end{array} \right\}_i = \left\{ \begin{array}{c} q_1 - q_i \\ q_2 - q_i \\ \vdots \\ q_3 - q_i \end{array} \right\} \tag{4.33}$$

which can be recast as

$$[\mathbf{A}][\mathbf{x}] = [\mathbf{b}] \tag{4.34}$$

Since the matrix $\mathbf{A}$ is not square, this represents an overdetermined system, which can be solved in a Least-Squares approach, hence the name. To begin with, matrix $\mathbf{A}$ is decomposed into an orthogonal matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$.

$$[\mathbf{A}] = [\mathbf{Q}][\mathbf{R}] \tag{4.35}$$

The solution of equation 4.34 can be written simply as

$$[\mathbf{x}] = [\mathbf{R}]^1[\mathbf{Q}]^T[\mathbf{b}] \tag{4.36}$$

It must be noted that $[\mathbf{R}]^1[\mathbf{Q}]^T$ is actually a set of weighting functions. Thus the gradient computation at node i becomes as follows:

$$\left.\frac{\partial q}{\partial x}\right|_i = \sum_{j=1}^{N} w_{x_j}(q_j - q_i) \qquad \left.\frac{\partial q}{\partial y}\right|_i = \sum_{j=1}^{N} w_{y_j}(q_j - q_i) \qquad \left.\frac{\partial q}{\partial z}\right|_i = \sum_{j=1}^{N} w_{z_j}(q_j - q_i) \quad (4.37)$$

The weighting function only depends on mesh geometry. Thus the weighting functions need to be calculated only once when stationary mesh is used.

There are several QR decomposition methods that can be used here such as the Gramm-Schmidt or Householder algorithms. While the former is the simpler method of the two, the later is considered more robust [98]. Since the matrix is not expected to be ill-defined, the simpler Gramm-Schmidt algorithm was chosen. Hence, the weighting functions are defined as:

$$w_{x_j} = \frac{\Delta x_{ji}}{r_{11}^2} - \frac{r_{12}}{r_{11}r_{22}^2}\left(\Delta y_{ji} - \Delta x_{ji}\frac{r_{12}}{r_{11}}\right) + \frac{\Upsilon}{r_{33}^2}\left(\Delta z_{ji} - \Delta x_{ji}\Upsilon\right) \quad (4.38)$$

$$w_{y_j} = \frac{1}{r_{22}^2}\left(\Delta y_{ji} - \Delta x_{ji}\frac{r_{12}}{r_{11}}\right) - \frac{r_{23}}{r_{22}r_{33}^2}\left(\Delta z_{ji} - \Delta x_{ji}\Upsilon\right) \quad (4.39)$$

$$w_{z_j} = \frac{1}{r_{33}^2}\left(\Delta z_{ji} - \Delta x_{ji}\Upsilon\right) \quad (4.40)$$

where

$$\Upsilon = \frac{r_{12}r_{23} - r_{13}r_{22}}{r_{11}r_{22}} \quad (4.41)$$

$$r_{11} = \sqrt{\sum_{j=1}^{N} \Delta x_{ji}^2} \quad (4.42)$$

$$r_{12} = \frac{\sum_{j=1}^{N} \Delta x_{ji}\Delta y_{ji}}{r_{11}} \quad (4.43)$$

$$r_{13} = \frac{\sum_{j=1}^{N} \Delta x_{ji}\Delta z_{ji}}{r_{11}} \quad (4.44)$$

$$r_{22} = \sqrt{\sum_{j=1}^{N} \left(\Delta y_{ji} - \Delta x_{ji}\frac{r_{12}}{r_{11}}\right)^2} \quad (4.45)$$

$$r_{23} = \frac{\sum_{j=1}^{N} \Delta z_{ij} \left(\Delta y_{ji} - \Delta x_{ji} \frac{r_{12}}{r_{11}} \right)}{r_{22}} \tag{4.46}$$

$$r_{33} = \sqrt{\sum_{j=1}^{N} \left(\Delta z_{ji} - \Delta y_{ji} \frac{r_{23}}{r_{22}} + \Delta x_{ji} \Upsilon \right)^2} \tag{4.47}$$

Numerical studies by Anderson and Bonhaus [3] suggested that being based on an analytical method, the Green-Gauss gradient reconstruction is more accurate in approximating the derivative. However, they found that the linear reconstruction of equations 4.21 and 4.22 is more compatible to the Least-Squares method so that a more accurate reconstruction is obtained as a result. Haselbacher's studies [50] also supported these findings.

Figure 4.4 illustrates the main drawback of the Least-Squares method in dealing with nodes on the non-slip wall boundary. In this situation the weighting would be biased toward the neighbouring nodes on the wall which have zero velocity. Hence the gradient on the wall would be underestimated. To alleviate the problem, it is common practice to introduce virtual wall edges [50]. It must be noted however that such a modification may improve the flux calculation, but still underestimates the wall gradient.

4.5 Diffusive Fluxes

It has been shown in chapter 2 that for a Newtonian Fluid, the viscous flux **G**, is proportional to the velocity gradient on the control volume surface (eq. 2.4 and 2.13). In computing the velocity gradient on the face, one has to remember that numerically the diffusive flux should have a stabilising effect, i.e. it must be positive. Haselbacher [50] has discussed extensively about issues regarding the diffusive term in the Navier-Stokes equations. The following scheme is employed to calculate the gradient at the control volume face.

$$\nabla Q_{ij} = \overline{\nabla Q}_{ij} - \left[\overline{\nabla Q}_{ij} \cdot \hat{r}_{ij} - \frac{Q_j - Q_i}{|\mathbf{r}_{ij}|} \right] \hat{r}_{ij} \tag{4.48}$$

where

$$\overline{\nabla Q}_{ij} = \frac{1}{2} \left(\nabla Q_i + \nabla Q_j \right) \tag{4.49}$$

4.6 Boundary Conditions

Boundary conditions are defined using the characteristic method. Supersonic boundaries that require information from one side only are imposed using the strong form, whilst subsonic boundaries are defined using the weak form to take into account information from both sides of the boundary face. As shown in fig. 4.7, consistent with the internal faces, the left state vector is the internal information whose values are taken from the old variable values of the boundary vertices. The right state vector is specified according to the boundary condition type.

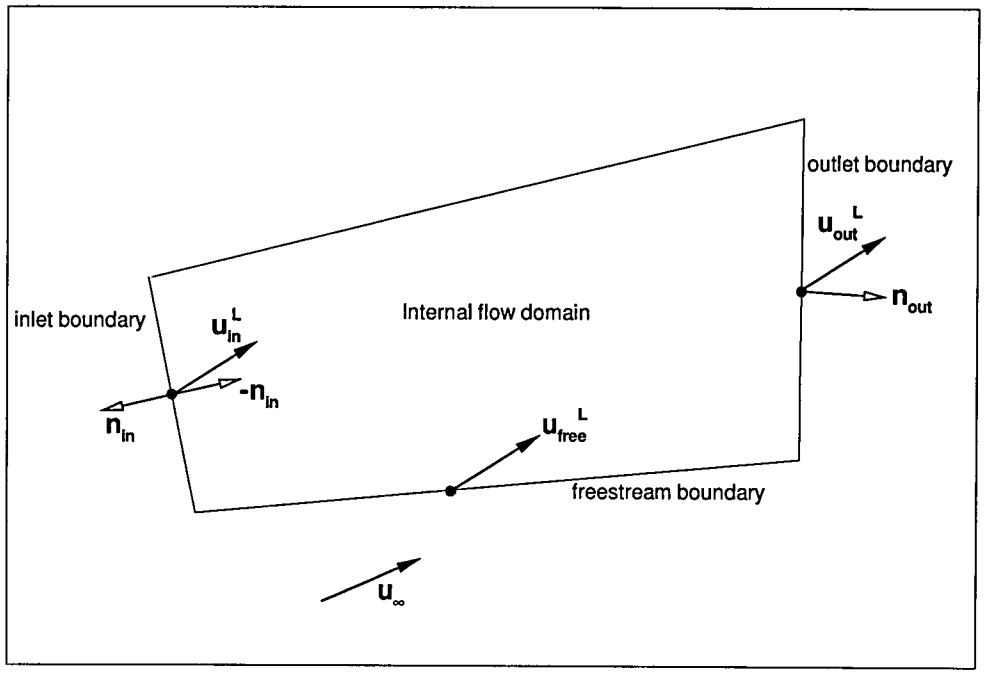


Figure 4.7: Inlet, outlet and free stream boundary conditions

4.6.1 Inlet Boundary

For a subsonic inlet, total pressure p_0, total temperature T_0 and the flow direction using unit velocity vector $\hat{\mathbf{u}}$ are specified. Following Haselbacher [50], the right state vector R is defined using a backward-propagating Riemann invariant

$$R_- = \mathbf{u}_{in}^L \cdot (-\hat{\mathbf{n}_{in}}) - \frac{2a_{in}^L}{\gamma - 1} \tag{4.50}$$

and the total speed of sound

$$a_0 = \gamma \Re T_0 = (a_{in}^L)^2 + \frac{\gamma - 1}{2} |\mathbf{u}_{in}^L|^2 \tag{4.51}$$

are used to formulate the right state speed of sound, which gives

$$a_i n^R = -R_- - \frac{\gamma - 1}{(\gamma - 1)\cos^2\theta + 2} \left[1 + \cos\phi \sqrt{\frac{(\gamma - 1)\cos^2\theta + 2}{\gamma - 1} \frac{a_0^2}{R_-^2} - \frac{\gamma - 1}{2}} \right] \tag{4.52}$$

where θ is the angle between the left state velocity vector $\mathbf{u}_{in}^L$ and inward pointing unit normal vector at inlet plane $-\mathbf{n}_{in}$.

For a wholly supersonic boundary, the incoming Mach number is prescribed to determine the right state vector.

When perturbed boundary conditions are required, a modification is performed case by case. Hence the description is in the appropriate section defining the test case.

4.6.2 Outlet Boundary

For a subsonic outlet, the static pressure p_{out}^R is specified. Once again, following Haselbacher [50], the right state vector is calculated as

$$\rho_{out}^R = \rho_{out}^L \left(\frac{p_{out}^R}{p_{out}^L} \right)^{\frac{1}{\gamma}} \tag{4.53}$$

$$u_{out}^R = u_{out}^L + \frac{2\hat{n}_{out_x}}{\gamma - 1} \left(a_{out}^L - a_{out}^R \right) \tag{4.54}$$

$$v_{out}^R = v_{out}^L + \frac{2\hat{n}_{out_y}}{\gamma - 1} \left(a_{out}^L - a_{out}^R \right) \tag{4.55}$$

$$w_{out}^R = w_{out}^L + \frac{2\hat{n}_{out_z}}{\gamma - 1} \left(a_{out}^L - a_{out}^R \right) \tag{4.56}$$

which has been based on the following Riemann invariants:

$$R_+ = \mathbf{u}_{out}^L \cdot (\hat{\mathbf{n}}) + \frac{2a_{out}^L}{\gamma - 1}$$

$$R_s = \frac{p}{\rho^\gamma}$$

$$R_t = \mathbf{u} \cdot \mathbf{t}$$

where R_t stands for the tangential Riemann invariant and $\hat{\mathbf{t}}$ is the appropriate tangential unit vector.

For a supersonic outlet, the right state is specified to be the same as the left state. Consequently the Riemann solver is not required.

4.6.3 Free Stream Boundary

The right state vector is simply taken from the specified free stream condition. Currently, for lifting flow there is no far field lift correction implemented. Consequently the free stream boundary must be placed as far away as possible from the lifting body to minimise its influence on the boundary.

4.6.4 Cyclic Boundary

A cyclic boundary is used when the flow-field has repetition in the span-wise sense. Hence the two nodes that makes up an aliasing pair should be identical topologically. The extended data structure allows each node pair to be calculated separately. The pairing nodes should have identical value in the ideal condition (i.e. computer round-off error is zero). As such an ideal condition does not exist the two values are averaged and then stored in each node to minimise the round-off error.

4.6.5 Periodic Boundary

A periodic boundary is used for flow with repetition along the stream-wise direction. It is normally used for a fully developed flow in a channel or pipe. It is assumed that such a flow is purely driven by pressure gradient to overcome wall friction. Thus a body force to model the pressure gradient is applied to eq. 2.7. The magnitude of the force must be estimated from a published data such as the Moody Chart or using the resistance law proposed by Blasius [135]. Having introduce the body force, all of the flow variables are treated in the same manner as for cyclic boundary.

4.6.6 Slip Wall Boundary

In a slip wall boundary condition, the flow is forced to be tangential to the wall. It is implemented in two stages. Firstly, the residual of the fluxes is calculated using the discrete form of eq. 2.3. Notice, the normal velocity U_n is not assumed to be zero here.

The second stage is velocity update at the end of each of the RK stage to ensure that it will be tangent to the wall.

$$\mathbf{u}_{wall} = \mathbf{u}_t = \mathbf{u} \cdot \hat{\mathbf{t}} = \mathbf{u} + (-\mathbf{u} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \tag{4.57}$$

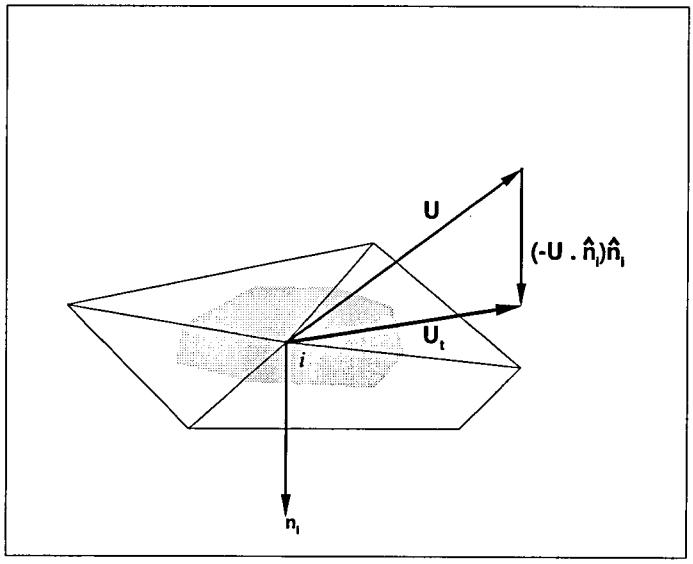


Figure 4.8: Calculation of tangential flow to node i that lies on a slip wall

where $\hat{t}$ is the appropriate tangential unit vector. This vector operation highlight the dependency on proper wall normal estimation that is sketched in fig. 4.4.

Although this treatment in general does not pose any problems for a slip wall boundary, it may not be applicable for sharp concave corner such as a trailing edge. It is obvious from fig. 4.4 that the approximation of the normal vector of the boundary nodes at the trailing edge is erroneous due to the discontinuity in the surface geometry. In turn, fig. 4.8 shows that such a normal is important to define the flow tangency, which is important for the Kutta condition. Therefore two possible modifications of the slip wall boundary condition are considered for the trailing edge problem. The first modification is to enforce the Kutta condition by simulating a stagnation condition at the trailing edge, i.e. zero velocity at the trailing edge vertices. The second modification is based on characteristic-type boundary condition that has been proposed by Anderson [2].

$$p_{wall} = p_{old} + \rho_{old} a_{old} U_n \tag{4.58}$$

$$\rho_{wall} = \rho_{old} + \frac{p_{wall} - p_{old}}{a^2_{old}} \tag{4.59}$$

where subscript old indicates the values that have been obtained at from the previous stage of the RK solver.

4.6.7 Symmetry Wall Boundary

Strictly speaking, symmetry boundary condition requires zero normal pressure gradient as well as tangential velocity on the surface boundary. The second requirement means that a symmetric boundary condition is very similar to a slip wall boundary. However the first requirement is rather complicated for an unstructured mesh. Since this boundary condition is less important for the current work it was felt that the normal pressure gradient obtain from the slip wall boundary was acceptable. Thus, in this thesis the symmetry boundary is treated identically to the slip wall boundary.

4.6.8 No-slip Wall Boundary

The most straight forward way of implementing adiabatic non-slip boundary condition is by setting velocity U equal to zero in equation 2.3. However, it was argued by Anderson and Bonhaus [3] that this simple implementation would enter a limit cycle, thus preventing convergence. They suggest an isothermal like no slip wall boundary condition where velocity is zero and temperature is defined by the following empirical formula:

$$\frac{T}{T_\infty} = 1 + \sqrt{Pr}\frac{\gamma - 1}{2}M_\infty^2 \tag{4.60}$$

Isothermal No-slip Boundary condition is virtually identical with the implementation of adiabatic no-slip wall boundary. The main difference is that the wall temperature T is specified as required, rather than using an empirical law.

4.7 Implementation of Turbulence and Sub-Grid Scale Model

All of the Sub-Grid Scale models that have been discussed in chapter 2 are based on the eddy viscosity hypothesis of Boussinesq, which is implemented by modifying the dynamic viscosity. For completeness the eddy viscosity based RANS turbulence model that is discussed in appendix A is also included.

4.7.1 Filter size

The characteristic filter size is taken as the length of a cube with equivalent volume of the polygonal control volume, thus for a median dual control volume enclosing vertex 0 this is

$$\Delta = \sqrt[3]{V_0} \tag{4.61}$$

This is similar to $\Delta = \sqrt[3]{\Delta x \Delta y \Delta z}$ that is often employed for LES on structured meshes [35].

4.7.2 Distance to the Nearest Non-slip Wall

Distance to the nearest wall is calculated by a simple method that relies on brute-force rather than the more efficient algorithm of Löhner [71]. In this approach, distances between every vertex to each non-slip wall boundary vertex is measured, thus a $Nv \times Nb$ loop is required where Nv is total number of vertices and Nb is total number of non-slip wall vertices. Fig. 4.9 illustrates how the nearest wall distance is calculated.

4.7.3 Implementation of Spalart-Allmaras model

Inspection at (A.20) shows that f_w values will quickly approach a constant value as soon as one moves away from a wall. Although this value is of order 1, its calculation depends on evaluation of r and g at (A.22) and (A.21), respectively, both can quickly approaching computer floating point limits that might cause overflow. In fact Spalart and Allmaras suggest to limit the magnitude of r to 10 [116]. To reflect that g is negative for a small range of $-r$, the following modification is employed for the present code:

$$r = sign(r_{org}) max(|r_{org}|, 10.0) \tag{4.62}$$

where r_{org} is calculated using (A.22)

Unlike scalar convective schemes, such as that of Jameson [57] or Liou's AUSM-family [70], that can be used for any transport variable, the Roe scheme must be modified when a new variable, such as the modified turbulent viscosity of the S-A model, is used. Alternatively, a separate convective scheme must be designed for the turbulence variable. To ensure that the numerical method is stable, a first order upwind scheme is used.It is felt that the error coming from such a lower order scheme is not an important issue once the solution converge to a steady state.

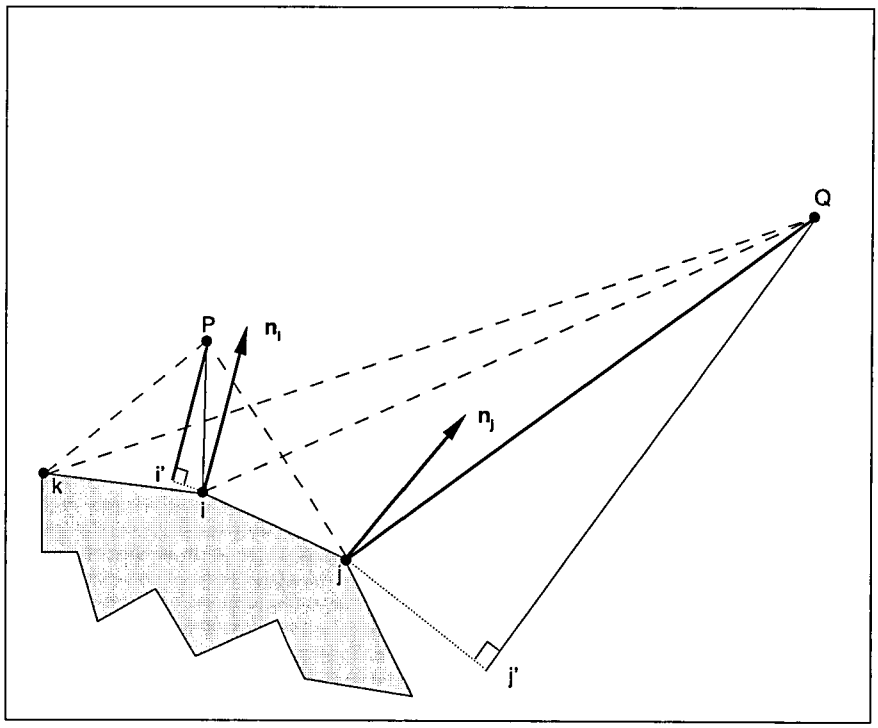


Figure 4.9: Nearest wall distance

4.8 Parallel Implementation

Large Eddy Simulations require a large number of computational nodes to resolve sufficient range of turbulence scales and computation over a large number of time steps in order to obtain an accurate statistical mean of the fluctuation that is inherent in turbulent flows. To obtain the solution in a reasonable time scale on current computer technology, parallel computer is a necessity.

4.8.1 Parallelisation Strategy

Foster [37] suggests that there are two decomposition strategies that are suitable for parallel programming. The first one is a domain decomposition, which partitions the data into a number of subsets to be handled by separate processors. The other is a functional decomposition where each processor deals with a different task of the problem in a manner similar to a factory assembly line. The nature of the present program that only consist of a small number of subtasks made the second strategy impractical as only a small number of processors could be utilised. The first decomposition is well suited to the nature of the problem that is outlined at the beginning of this chapter.

Currently there are a number of parallel computer architecture that are available, ranging from the vector machines such as CRAY to the Complex Instruction Set Chip (CISC)-based PC cluster. Hence portability is a desirable feature in building a parallel programme. As ANSI Fortran 77 and ANSI C had been used to write the sequential code to ensure portability, the MPI library, which has been in widespread use among the parallel computing community in recent years, has been chosen in spite of the fact that there are other libraries such as SHMEM that are potentially more efficient [72]. While the SHMEM library is only available for specific hardware, namely the CRAY vector machines and SGI machines, MPI implementations are available from both vendor's hardware and as open-source such as MPICH [47].

4.8.2 File input and output

Parallelisation is achieved by partitioning the input mesh into several sub-meshes, each of which is assigned to a different processor. This strategy is closely linked to input/output file handling. Prior to the MPI-2 standard, which was agreed in 1997 by the MPI committee, there had been no standard way of handling input and output

files in a parallel machine environment, hence each vendor devised different methods in dealing with the issue. The simplest and safest way to handle input and output while still maintaining portability has been demonstrated by Hansen [48] who asked each processor to read and write from separate files. This strategy in effect requires knowledge of the number of processors prior to code execution as each mesh partition is read from a separate file. Furthermore, it means that there are at least as many output files as the number of processors involved. Although this is not an issue for a RANS calculation, in LES this would lead to large amount of data files as the flow field needs to be dumped relatively frequently to allow time-history post processing. Hence, the total number of output files would be m number of solutions multiplied by the n number of mesh partitions, which is rather impractical.

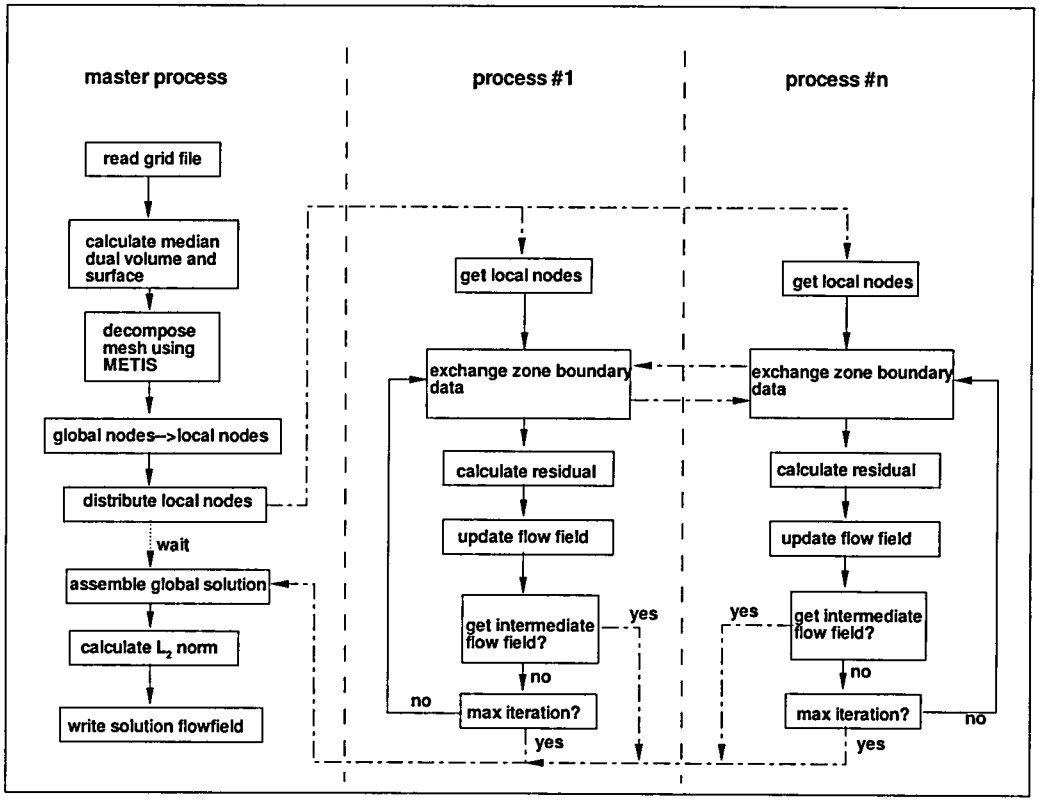


Figure 4.10: Parallel flow solver

In a multi-processor environment, the present code applies a master-worker paradigm. One processor (rank id 0) is designated as the master process while the rest are workers. The worker processes carry out the actual computation whereas the

master performs administrative task such as i/o, computing control volume surface area and volume and finally mesh partitioning right at the beginning of execution. This allows the number of processors to be decided by the user at run-time as well as keeping a single mesh file and output for each instantaneous flow-field. To ensure that the largest problem that can be handled is not limited by the memory availability in the master node [1], it was decided that the master node should not be loaded with variables for computing the flow solution, hence it is not involved in the time-stepping stages. Figure 4.8.2 illustrates the present strategy with dash-dotted lines indicating communication between processors. An obvious consequence of the present strategy is that running the present code on two processors does not bring any benefit. In fact a request for two processors computation is currently undefined in the code.

4.8.3 Domain Decomposition

The METIS 4.0 graph partitioning library [61] was used to decompose the unstructured mesh. From experiment it was found that the kV-METIS routine of the library gives the most balanced node distribution for this application. Figure 4.11 shows a typical mesh partitioning using kV-METIS for a RANS mesh around a two dimensional supercritical aerofoil. However, the load balancing is generally not optimum for mixed element meshes since the partitioning objective of the library is to balance the number of nodes, whereas the volume of work is a function of edges.

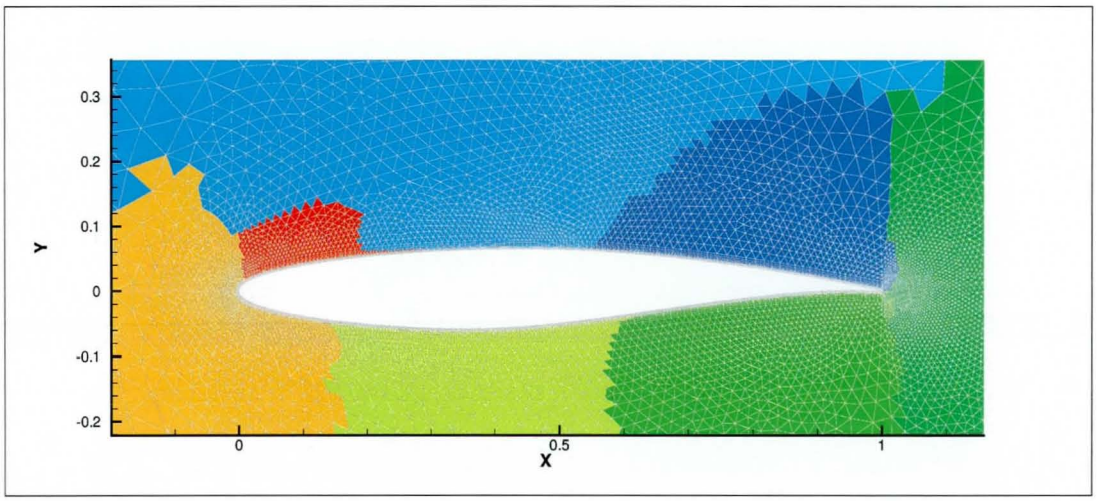


Figure 4.11: kV-METIS domain decomposition.

[1] A distributed memory machine is assumed

neighbour id(1)	...	neighbour id (n-nbor)
comm type(1)	...	comm type (n-nbor)
send start(1)	...	send start (n-nbor)
send end(1)	...	send end (n-nbor)
recv start(1)	...	recv start (n-nbor)
recv end(1)	...	recv end (n-nbor)
comm info (1)	...	comm info (n-nbor)

Figure 4.12: A two dimensional array to store communication path

4.8.4 Point to Point Communication

The most important part of parallelising a sequential unstructured code is providing point to point (processor to processor) communication. The first issue to be addressed is how a processor, which holds only part of the decomposed mesh, recognises its neighbours. For this purpose, a communication array, called `path`, is defined in each processor. Fig. 4.12 shows a schematic diagram of the two dimensional array `path` for `n-nbor` neighbouring zones. `neighbour id` holds the process number of the neighbouring zone. `send start` and `send end` are pointers to the send buffer. `recv start` and `recv end` are pointers to the receive buffer. `comm type` and `comm info` are used when the zone boundary describes a periodic or cyclic boundary condition.

Two buffer arrays are provided for each processor: `qsend`, which handles all of the variables to be send out, and `qrecv`, which handles every variable that a processor receive from its neighbours. The two of them must be synchronised.

As shown in figure 4.13 `qsend` and `qrecv` are formed by elements of the flow-field variable `q` that is local to each processor. The first buffer is connected to the field through the `sendlist` array that holds the vertex number of the flow-field whose data should be send out. The second buffer is connected to the halo cell through the `recvlist` array. The diagram illustrates processor #n sends data to processor #m.

The second issue that is MPI specific is deadlock. This may occur when the two

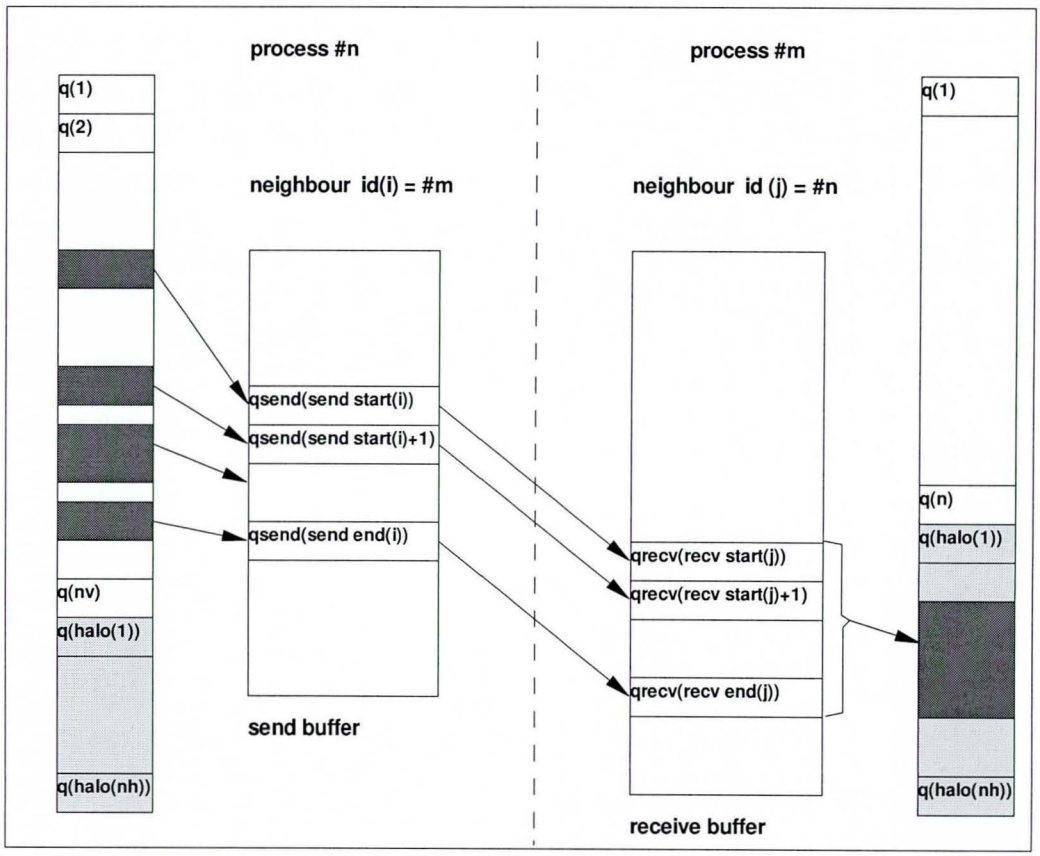


Figure 4.13: Point to point communication

communicating processes are trying to send information at the same time. For a regularly partitioned domain such as a structured mesh, it is possible to avoid deadlock by designing a communication whereby one processor sends information while the other receives the information, after which each of them change their role as receiver and sender [4]. For an unstructured mesh, such a strategy is impractical. It would be simpler if each processor tries to send and receive data regardless what the other is doing. Non-blocking sending and receiving MPI routines are employed to achieve this. The following algorithm illustrates how `nsend` (`nrecv`) data are sent (received) to (from) its neighbours.

```
DO i = 1,n_nbor
  Nonblock_SEND(qsend(istart(i)),nsend(i),i,requestS(i))
  Nonblock_RECV(qrecv(istart(i)),nrecv(i),i,requestR(i))
ENDDO
DO i = 1,n_nbor
  Finish_SEND(requestS(i))
  Finish_RECV(requestR(i))
ENDDO
```

During non-blocking communication, MPI defines that the processor can perform other tasks and determining later whether the communication has finished. If this is the case the communication process is terminated, otherwise the processor should wait. This interrogation and decision making is performed by the explicit termination call. During an MPI course run by EPCC [74], it has been suggested that the other task that a processor can do is another communication. Hence it is actually also possible to implement the communication as a pair of nonblocking send and standard blocked receive as follows:

```
DO i = 1,n_nbor
  Nonblock_SEND(qsend(istart(i)),nsend(i),i,requestS(i))
ENDDO
DO i = 1,n_nbor
  Standard_RECV(qrecv(istart(i)),nrecv(i),i,requestR(i))
ENDDO
DO i = 1,n_nbor
  Finish_SEND(requestS(i))
ENDDO
```

4.9 Parallel Performance

Table 4.1 compares the execution time of the two communication models described in the previous section when the code is used to solve an 867k nodes problem for 100 iterations on an Intel® Xeon 2.4 GHz based PC cluster. Model 1 is a pair of non-blocking send and receive, while Model 2 is a pair of nonblocking send and standard blocked receive. It is clear that the second model offers a very small improvement in term of execution time.

worker processes	Model 1			Model 2		
	run time (s)	speed up	eff. (%)	run time (s)	speed up	eff. (%)
1	4432.07	-	-	4367.64	-	-
2	2292.17	1.934	96.70	2270.11	1.924	96.20
4	1183.13	3.746	93.65	1179.91	3.702	92.55
8	678.13	6.536	81.70	675.97	6.461	80.76
15	416.25	10.648	70.99	414.92	10.526	70.41
31	242.19	18.300	59.03	239.71	18.221	58.77

Table 4.1: Parallel performance of Model 1 and 2 for a 867k nodes problem.

4.10 Closure

The present chapter described the numerical method for the compressible turbulent flow simulation using hybrid mesh in three dimension. The concept of mesh transparency based on edge data structure is developed, which is equivalent to a Petrov-Galerkin Finite element formulation [8]. This allow a post processing and solution method to be developed for a computational mesh consisting of arbitrary cell types. The parallelisation of the method has also been described. The MPI library has been utilised for this purpose to ensure portability.

Chapter 5

Assessment of base solver for steady flow problems

5.1 Introduction

The numerical method for steady flow problems developed in chapter 4 is a three dimensional extension from the earlier work of Haselbacher [50]. Since the two dimensional performance of the base solver has been extensively assessed in the earlier work, this chapter is only concerned with the three dimensional flow verification of the present code.

The convergence of the numerical method is not discussed here. Since the present work is ultimately designed for unsteady computation, namely LES, convergence is not a relevant property. Thus multi-grid convergence acceleration has not been implemented in the code.

The following test cases are designed to assess the performance of the core algorithm in three dimensional problems. A secondary objective is to demonstrate the code capability in dealing with complex geometry, which will be absent from the later LES experiments for practical reasons. Since LES needs a fine mesh and long time integration, complex geometry problems will be constrained by time and resources. For the rest of the chapter, results from the present code are labelled as Cirrus.

The first test case is an inviscid flow about the DLR-F4 wing body configuration to represent a typical geometry found in aerospace industry applications. The inviscid

flow simulation has been chosen to assess the basic numerics free from any turbulence modelling issues. Unfortunately, such a flow will be physically incorrect; for example, the shock is situated downstream of the real viscous experiment. Since the lack of viscosity has prevented direct comparison to experimental data, a set of results from a validated code was used as a reference solution. The second order coupled upwind solver of Fluent v. 6.0 (subsequently simply referred to as Fluent) is employed for this purpose. It must be noted that direct comparison of numerical accuracy between the two CFD solutions is not possible since the primal cell employed by Fluent has a smaller control volume than the median dual control volume of Cirrus. In effect, Fluent has finer spatial resolution from the same mesh in comparison to Cirrus [90].

The second test case is a viscous flow calculation to demonstrate that the discrepancy with experimental data in an inviscid computation is not caused by problems in the present algorithm. Turbulent flow over the ONERA M6 geometry was chosen for two reasons. Firstly, the relative simplicity of the geometry allows for relatively smaller mesh size than a viscous mesh over the DLR-F4 wing body configuration. Secondly, the test case is in widespread use in the CFD community for validation purposes [111], thus allowing for ease of comparison with other codes.

5.2 Inviscid flow over DLR-F4

The DLR-F4 wing body configuration was a generic subsonic transport type aircraft with a swept back wing of aspect ratio 9.5 [95]. Three sets of wind tunnel measurement were taken in the DRA (now QinetiQ) 8ft $\times$ 8ft Pressurised Subsonic/Supersonic Wind Tunnel, the NLR High Speed Wind Tunnel and the ONERA-S2MA Wind Tunnel. The wind tunnel results and half aircraft geometry for the present calculation were obtained from the AIAA drag prediction workshop which incorporated wing deformation due to aerodynamic forces [91].

5.2.1 Mesh and Boundary Condition

Gambit v. 2.04 (subsequently referred to as Gambit) was employed to generate the computational mesh. It consisted of 138.6k nodes (732.9k pyramids and tetrahedra). Dense grid clustering was applied on the high curvature surfaces – such as the fuselage nose, tail cone, windshield and wing leading edge – as well as the upper wing surface

where a shock wave was expected to occur. To achieve this goal, the Gambit size function was employed. However, Gambit only worked best when the sources were oriented along the coordinate direction. For general orientation, the source was actually defined in steps rather than a straight line. As a result of this limitation, the mesh quality was compromised. This is quite evident on the wing as shown on figure 5.1. The wing leading edge of the present geometry had high curvature in the chord-wise direction and virtually linear (i.e. negligible curvature) in the span-wise direction. Thus an anisotropic mesh would have been desirable. Since Gambit did not have such a functionality, the leading edge resolution was rather poor in comparison to the standard mesh of the workshop.

The computational domain was defined as a hemisphere enclosing the half aircraft model, rather than the cuboid shape used for the workshop standard mesh. The shape of the outer boundary was chosen to avoid any discontinuity that arose when two boundary surfaces meet at a sharp angle. As shown by section 4.2.3, such a discontinuity would compromise the computation of a median dual surface normal. Following the standard computational domain for the AIAA workshop, the hemisphere was defined with a radius of 50 mean aerodynamic chords. Symmetric boundary were applied on the symmetry plane whereas a Riemann free stream condition was applied on the concave skin of the hemisphere. A Mach number 0.75 and 0.17 ° incidence was applied at the free stream boundary. This corresponds to the workshop test case with an experimental C_L of 0.5 [1].

The slip wall boundary condition was applied on the wing-body surface. Unlike the outer boundary that was guaranteed to have a smooth surface due to the hemisphere domain, surface discontinuities could not be avoided on the wing trailing edge. Here, both modifications described in section 4.6.6 to deal with the trailing edge problem were tested. The stagnation condition to model the Kutta condition was labelled *sTE*. The correction of pressure and density was labelled *cbw*. Therefore, the pressure coefficient, C_p, at the trailing edge region was not expected to be correctly predicted by the present code in inviscid mode.

5.2.2 Numerical Solution

Following the findings of Pirzadeh and Frink [90] that showed an adverse effect of the flux limiter on their DLR-F4 problem, the present computation has been carried out

[1] Interpolation from experimental data published in [95] gave 0.177 ° incidence.

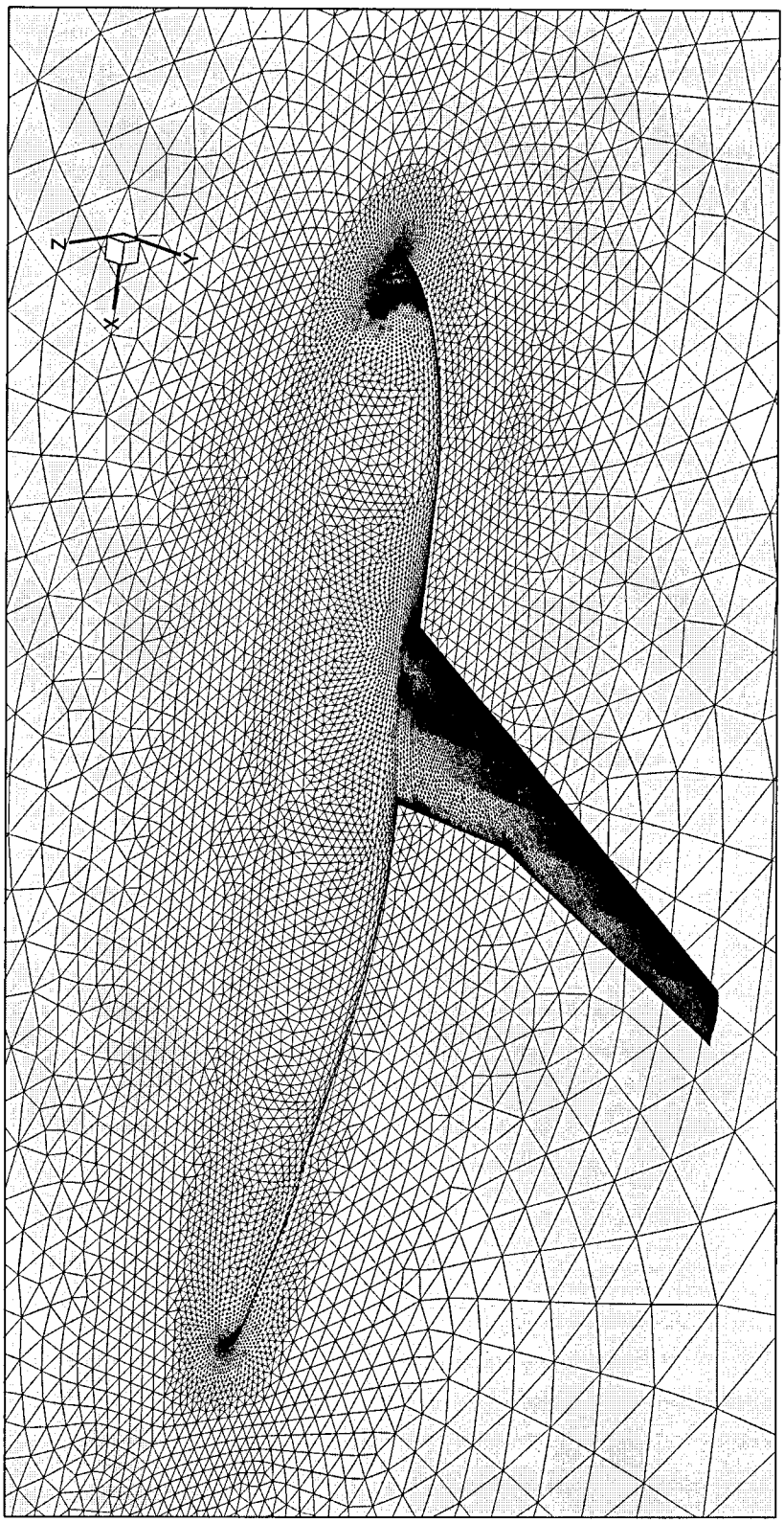


Figure 5.1: Inviscid mesh around DLR-F4 wing body configuration.

without any limiter. The computation was performed using 15 CPUs of an Intel ® Xeon 2.4 GHz based PC cluster. The solution was stopped at 8000 iterations when the residual error flattened out. No observable changes were detected when further iterations were performed.

In accordance with the common practice used in wind tunnel measurement, the coefficient of pressure C_p as a non-dimensional measure of relative pressure is used. For completeness, C_p is defined as:

$$C_p = \frac{p - p_\infty}{1/2\rho|\mathbf{u}_\infty|^2} \tag{5.1}$$

which means that negative values signified a lower pressure region than the ambient, i.e. suction region. Figure 5.2 shows the C_p distribution on the wing-body geometry as well as on the symmetry plane. The leading edge suction is clearly shown upstream of the shock bow at the front part of the wing suction surface.

Figure 5.4 shows a comparison of the pressure coefficient at 7 semi-span stations, which are shown in figure 5.3. The present computation with stagnation condition at the trailing edge is labelled as cirrus+sTe, while the pressure - density correction is labelled as cirrus+cbw. As expected all of the inviscid solutions predict a shock location further downstream of the experiment. The discrepancy of this shock position means that the inviscid calculation will have significantly larger lift–due to larger wing suction area– than that of a wind tunnel measurement.

The C_p plots show that the stagnation boundary condition at the trailing edge produces a closer shock location to the experiment than either Fluent or the corrected wall boundary calculation. The plots also shows that the pressure at the trailing edge for cirrus+sTe is close to the stagnation value. Although the trailing edge itself is behaving in accordance to inviscid aerofoil theory, severe non-physical behaviour is observed around it. On the other hand, cirrus+cbw and the Fluent boundary condition seem to mimic the viscous effect on the trailing edge. Thus, the pressure - density correction seems to be more preferable.

In general the present code behaves similarly to the second order upwind inviscid coupled solver of Fluent, especially when the wall boundary correction to the density and pressure variables are applied. It must be noted that Fluent predicts a sharper shock since its mesh is effectively finer than that of cirrus.

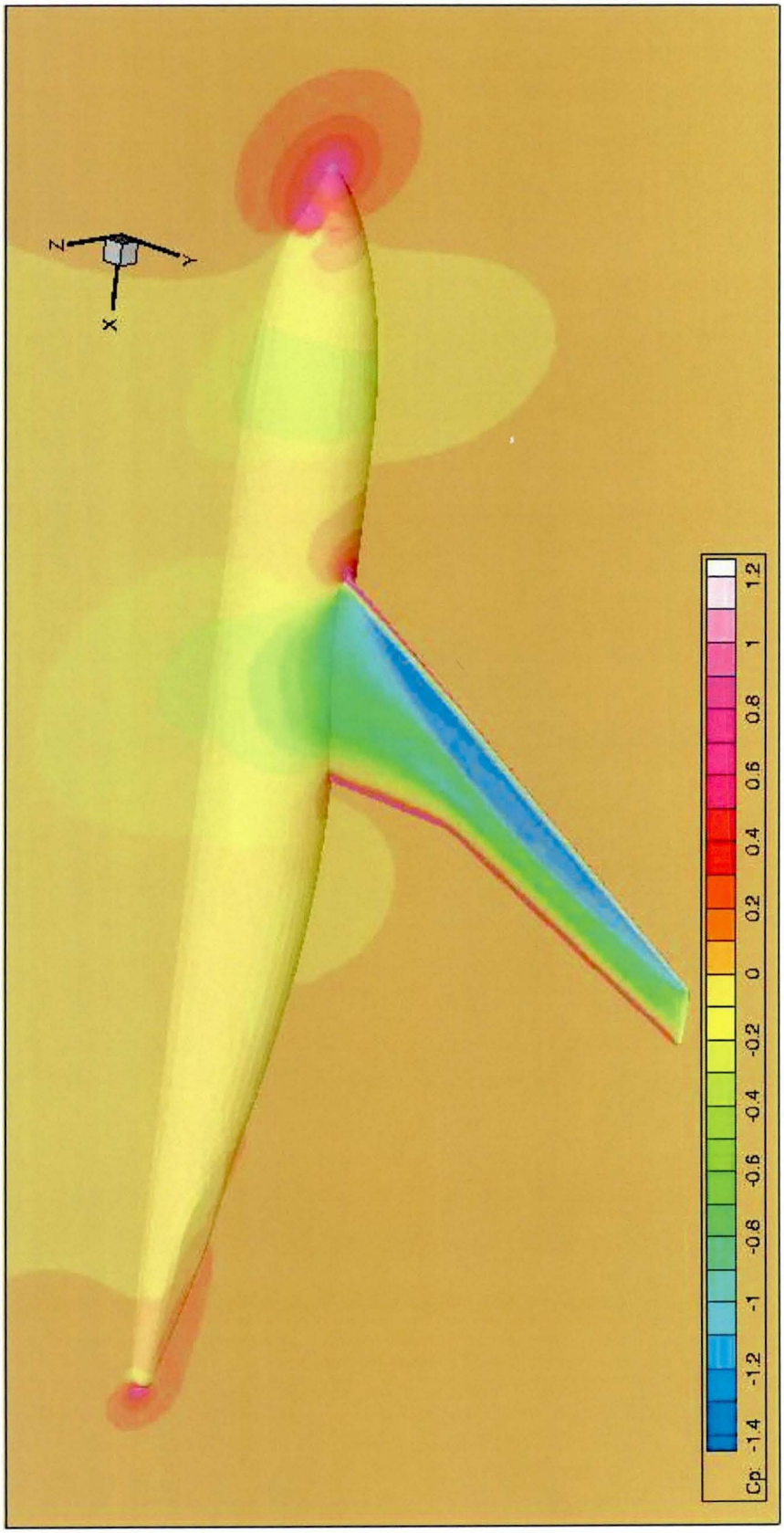


Figure 5.2: C_p distribution on DLR-F4 wing body configuration.

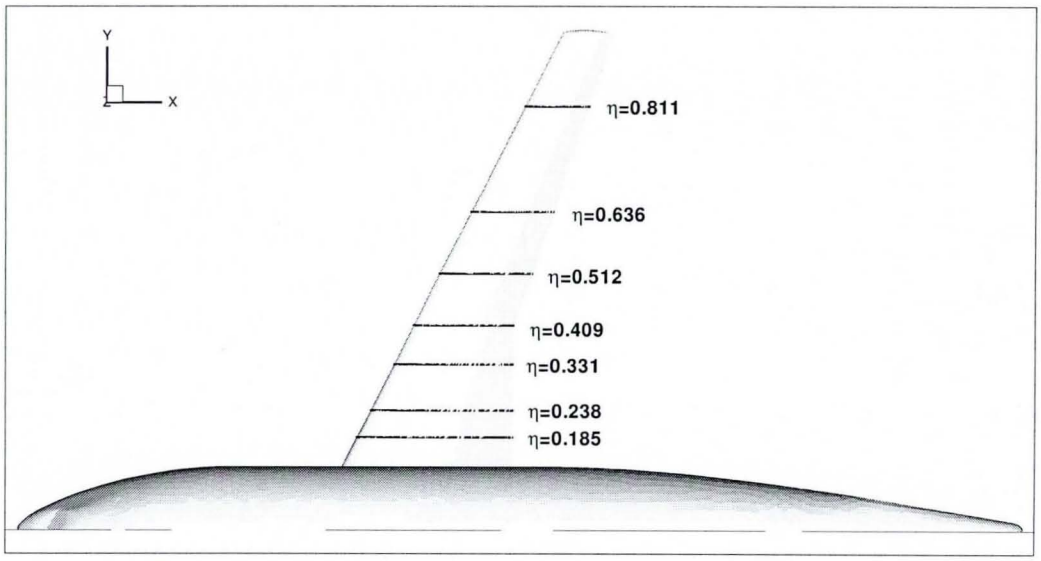
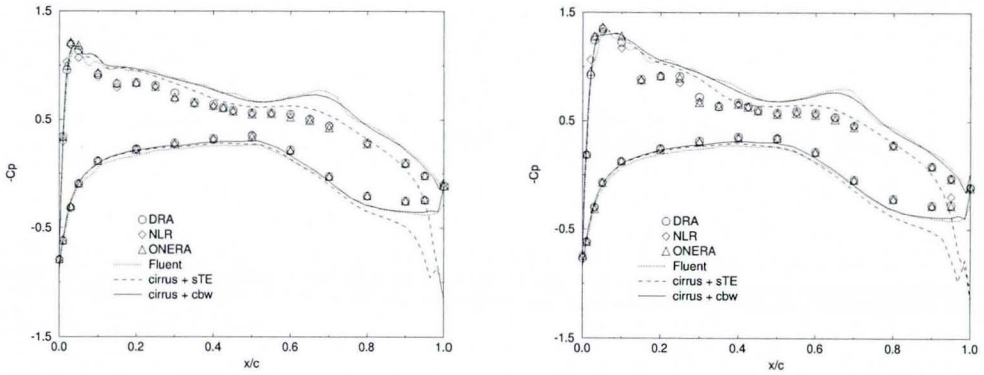


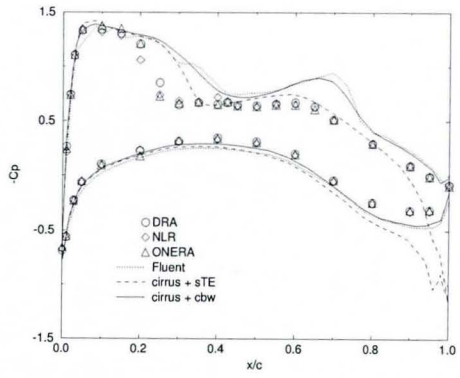
Figure 5.3: Pressure tapping locations on the wing of DLR-F4 configuration.



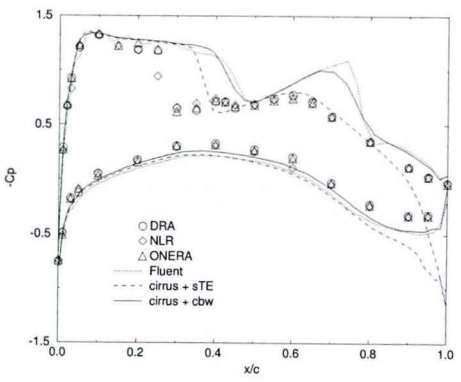
(a) 18.5% semi-span station, $\eta = 0.185$

(b) 23.8% semi-span station, $\eta = 0.238$

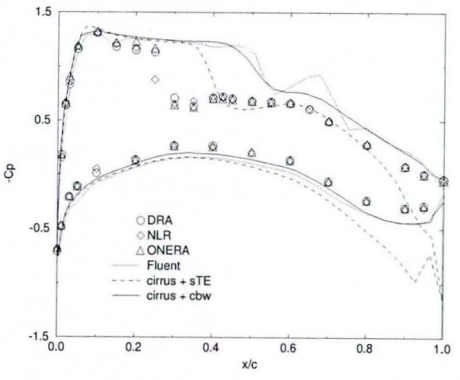
Figure 5.4: C_p distribution on DLR-F4 wing surface



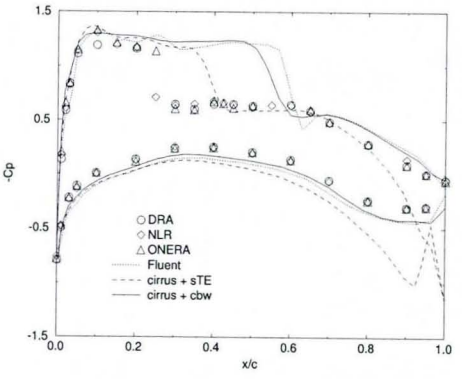
(c) 33.1% semi-span station, $\eta = 0.331$



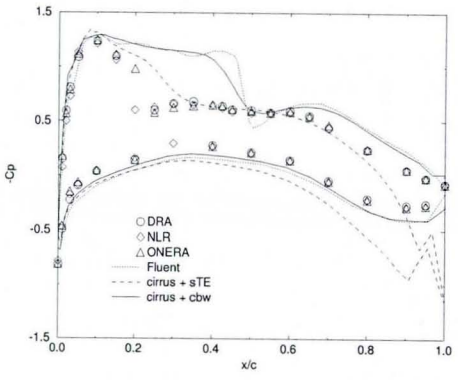
(d) 40.9% semi-span station, $\eta = 0.409$



(e) 51.2% semi-span station, $\eta = 0.512$



(f) 63.6% semi-span station, $\eta = 0.636$



(g) 81.1% semi-span station, $\eta = 0.811$

Figure 5.4: C_p distribution on DLR-F4 wing surface (Cont'd)

5.3 Turbulent flow over ONERA M6 Wing

Three-dimensional turbulent transonic flow about the ONERA M6 wing configuration was used to verify the steady viscous solver. ONERA M6 was an untwisted swept-back wing. It had a leading edge sweep angle of 30 °, an aspect ratio of 3.8 and a taper ratio of 0.562. The wing section was the symmetric conventional ONERA "D" aerofoil, which had 10% thickness-to-chord ratio. The numerical test was conducted at Mach 0.84 and incidence of 3.06 °(Test 2308). The nominal Reynolds number based on mean aerodynamic chord length, Re_c in the experiment was 11.72 million. The wing transonic flow regime proved to be quite challenging due to the presence of a λ-shock pattern on the suction surface. The complete description of the wing geometry and wind tunnel measurements can be seen in Schmitt and Charpin report [104].

The present RANS computation utilised the S-A turbulence model [116]. Since the test case was not intended as an exhaustive assessment of a turbulence model, flow with significant separation, such as Test 2564, was not chosen.

5.3.1 Computational Mesh and Boundary Condition

Due to symmetry, only a half wing needs to be simulated. A hemisphere with radius of 20 root chords was employed for the outer free stream boundary of the computational domain. The wing root was attached to the circular plane of the hemisphere, which was defined as a symmetric boundary condition. Being viscous, the wing surfaces were defined as no-slip-wall boundaries.

The computational grid consisted of 549.7k nodes (1.25M tetrahedra, pyramids and prisms), which were generated by Gambit. The lower surface was left much coarser than the upper surface as it was not expected to have significant flow features. See figure 5.5 for illustration. This mesh was quite coarse on the leading edge whose semi-circular section was on average only resolved by 12 nodes along the chord-wise direction. In spite of this, the mesh still required a large number of nodes to cover the wing surface since Gambit only allows nearly isotropic cells for the surface mesh. Thus, 300 and 200 nodes were used along the leading edge and trailing edge span, respectively. In contrast to this, the high aspect ratio cells of a structured mesh adopted by Slater [111] only used 33 nodes along the span with more than 20 nodes resolving the semi-circular leading edge along the chord. A size function was only used to control nodal clustering

around the rear shock on the upper surface. Rather than being designed for optimum node distribution, the grid was designed with simplicity in mind while maintaining a balance between number of cells and accuracy.

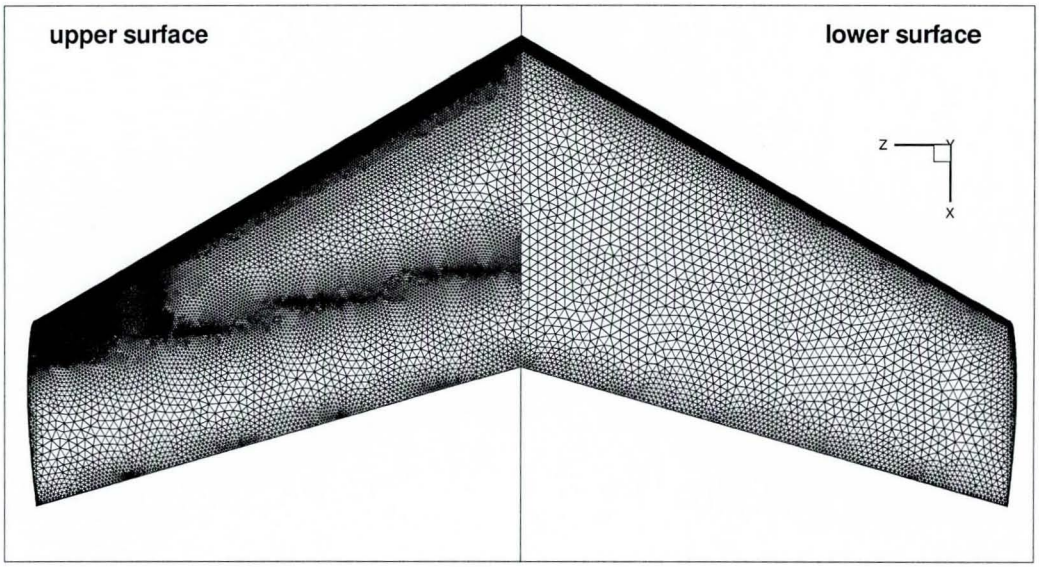


Figure 5.5: Wing surface mesh.

A prismatic region consisting of 25 layers was used to resolve the wing boundary layer. This region covers more than 75% of the nodes of the mesh. This was because the S-A model required the first node from the wall to be $y^+ \leq 5$ away to resolve the viscous sub layer. It was estimated that $y/c_{root} = 6.25 \times 10^{-7}$ corresponds to $y^+ = 1$ from the NPARC Alliance Validation Archive [111]. On the other hand, Gambit precision was limited at 1.0×10^{-6}. Considering that the computation was only performed at single precision, this limitation was deemed to be acceptable. To avoid any problem arising from Gambit resolution while satisfying requirement for the S-A model at the same time, the first node from the wall was placed at $y/c_{root} = 3.0 \times 10^{-6}$ which corresponds to $y^+ = 4.8$. To minimise the number of cell required to resolve the boundary, a rather large expansion factor of 1.4 was used at present.

Figure 5.6 shows that the trailing edge was slightly rounded in the present computation. This was needed to reduce the skewness of the prismatic cells around that region. In fact, Gambit produced very skewed prismatic cells with negative volumes when a sharp trailing edge was used in conjunction with the previously discussed y^+ for the first node from the wall.

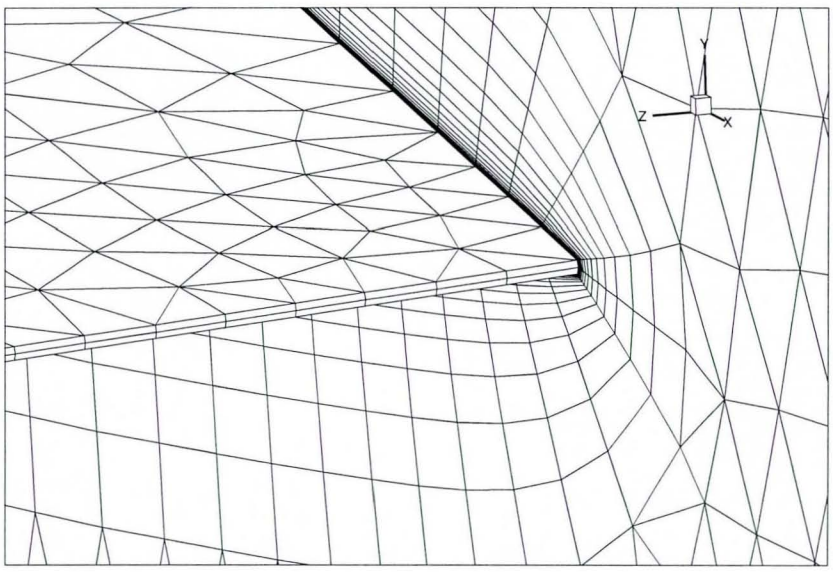


Figure 5.6: Mesh around wing trailing edge.

5.3.2 Numerical Solution

The computation was performed using 15 CPUs of an Intel® Xeon 2.4 GHz based PC cluster. The solution was stopped at 12000 iterations when the residual error decreased by 4 orders of magnitude and did not change rapidly anymore. Unlike the previous inviscid computation the boundary layer mesh of the present problem required the limiter to be activated. A K value of 2.5 for the Venkatakrishnan limiter was used.

As a reference solution, data from the NPARC Alliance Validation Archive [111] using the WIND code is employed to verify the present numerical solution. Figure 5.7 shows C_p contours on the upper surface. The λ-shock pattern that is reported in ref [111] is reasonably well resolved by the present simulation.

Schmitt and Charpin [104] only published surface pressure distributions from 7 span-wise stations, shown in figure 5.8. Hence, this is the only quantity that can evaluated here. Figure 5.9 shows a comparison of the C_p distribution along the chord at these span-wise stations between the present numerical solution, labelled cirrus, that of WIND and the experimental data. The vertical axis has been plotted such that negative C_p values lie above the origin.

The present turbulent computation generally compares very well with the wind

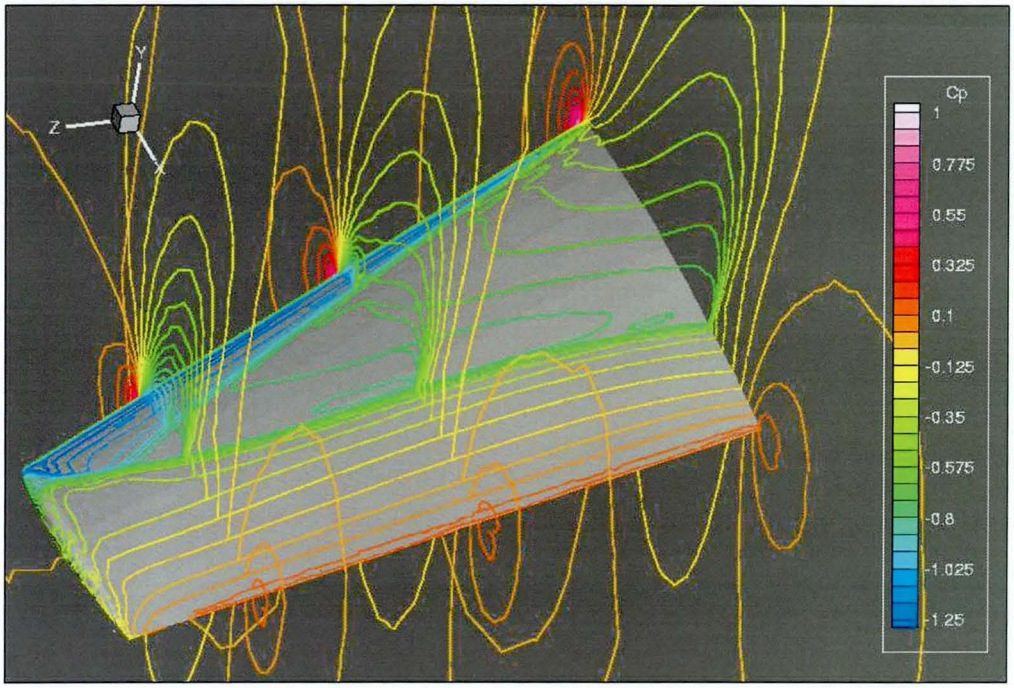


Figure 5.7: C_p contour on ONERA M6 wing surface; RANS calculation.

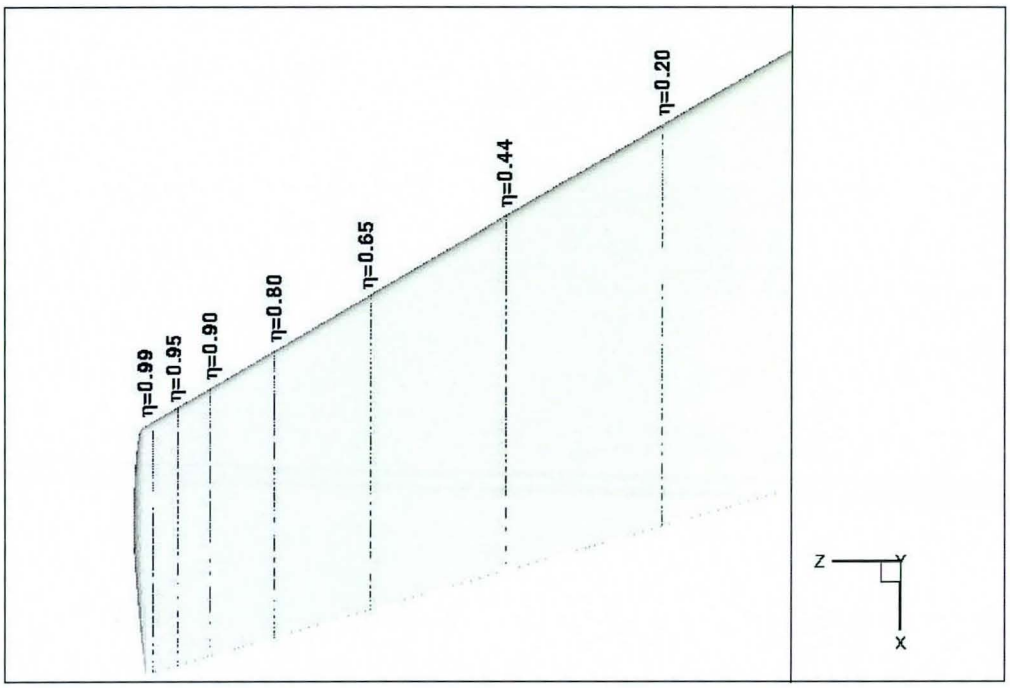
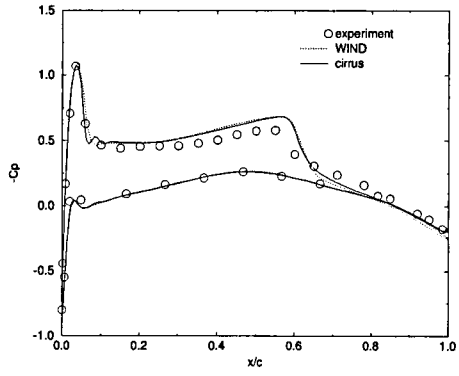
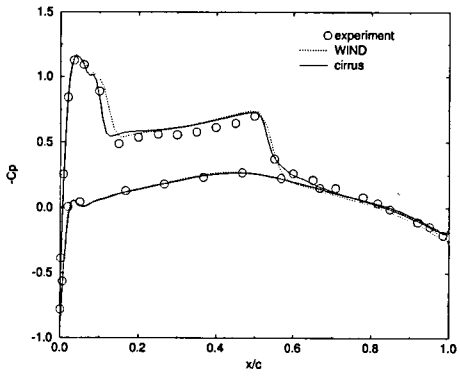


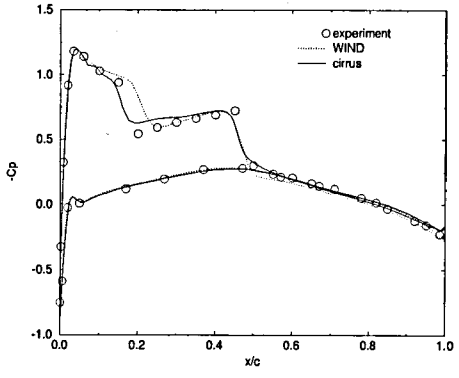
Figure 5.8: Pressure tapping locations on ONERA M6 wing.



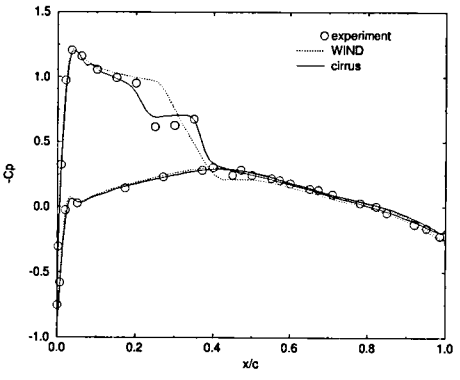
(a) 20% semi-span station, $\eta = 0.20$



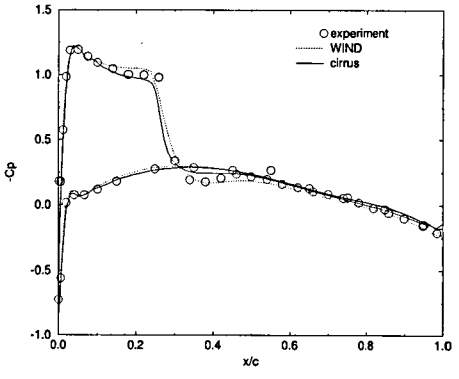
(b) 44% semi-span station, $\eta = 0.44$



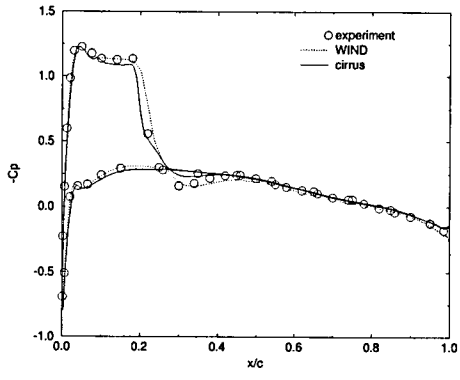
(c) 65% semi-span station, $\eta = 0.65$



(d) 80% semi-span station, $\eta = 0.80$

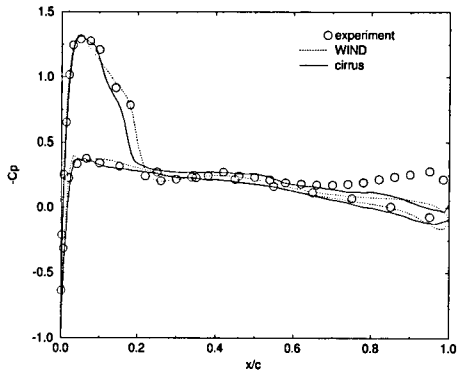


(e) 90% semi-span station, $\eta = 0.90$



(f) 95% semi-span station, $\eta = 0.95$

Figure 5.9: C_p distribution on ONERA M6 wing surface



(g) 99% semi-span station, $\eta = 0.99$

Figure 5.9: C_p distribution on ONERA M6 wing surface (Cont'd)

tunnel measurement as well as the CFD result of WIND code [111]. At 80% semi span location the present code still resolves the two shocks whereas WIND only shows a single shock. However, the present code does tend to smear the shock, hence reducing its strength. Unlike the structured mesh used in [111], the triangular cell on the wing surface has added some amount of diffusion to that of the turbulence model across the shock that is generally not-aligned with the mesh. It is felt that this drawback is compensated by the increased geometric flexibility of the unstructured mesh approach.

5.4 Closure

Two test cases presented on this chapter have verify the capability of the core algorithm in resolving complex three dimensional flow. The first one is an inviscid flow around a generic wing body configuration while the second one is a turbulent flow around a simple wing geometry. The two test case shows that the core algorithm works well in dealing with steady flow. Thus, it gives confident in extending the present code for unsteady flows and ultimately LES computations.

Chapter 6

Validation of LES code for unsteady turbulent flow

It has been shown in chapter 5 that the core numerical method developed in chapter 4 has been able to deal with steady flow problems involving complex geometry. This chapter is focused on the verification of the unsteady extension of the numerical scheme of chapter 4 in handling intermittent structures in turbulent flows. Three test cases, which are considered as prototype turbulent flows in fluid mechanics, are considered here. The first is the interaction of a shock wave with a spatially developing mixing layer. As the test case is designed for the beginning of laminar flow transition to turbulence, it can be used to assess the numerics free from any kind of sub-grid scale modelling. The second test case is a turbulent free jet flow, which is not only important in its own right for practical applications but also serves as a prototype of separated turbulent flows. The last test case is a Large-Eddy Simulation of a low Mach number flow through a round pipe to assess the code capability in solving wall bounded turbulent flow flow problems.

6.1 Shock Wave Impingement on Spatially Developing Supersonic Mixing Layer

This two dimensional test case is designed to test the code behaviour for the interaction of a shock-wave and a laminar mixing shear layer at the beginning of transition to turbulence. The case is purely a test of numerics, and there is no experimental data

available. It was used originally by Yee et al. [138] to test their characteristics filter behaviour, which was designed to give non-dissipative central schemes a shock capturing capability. The motivation is three-fold. Firstly, to validate the present unstructured upwind scheme, which is theoretically second order accurate in space, against the spatially fourth order scheme of Yee et al. [138]. Secondly, to investigate the effect of triangular meshes as compared to quadrilateral and hybrid meshes. The final motivation is to assess the effect of the limiter as well as the scaling of the Roe Dissipation matrix that have been described previously.

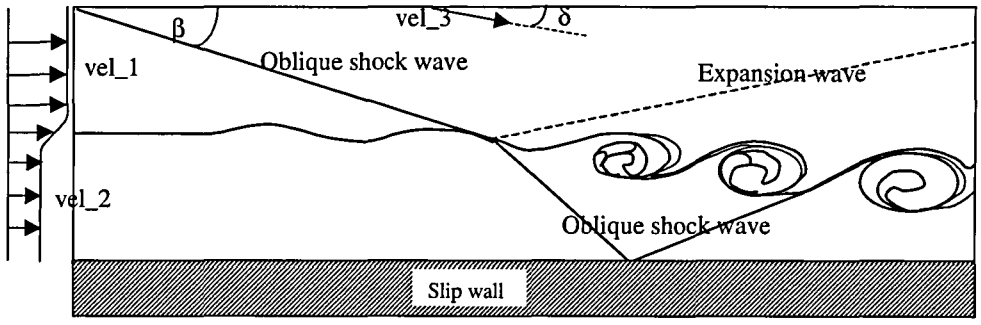


Figure 6.1: Shock-wave impingement on supersonic mixing layer

Figure 6.1 sketches the basic set up of the test case. Two perturbed supersonic streams flow from left to right. The Mach numbers are 5.625 and 1.7647 for the upper and lower streams, respectively. The two flows have identical static pressure and total temperature, thus they have different density, static temperature and sound speed. An oblique shock wave with shock angle $\beta = 12.0\,^{\circ}$ emanates from the top left corner of the domain. The oblique shock wave is created by setting the upper boundary as another supersonic inlet with flow direction of $\delta = 2.634\,^{\circ}$. As the shock wave reaches the density discontinuity, it is partially reflected upward as an expansion wave and refracted downward at a steeper angle. The refracted wave is eventually reflected upward by the solid boundary at the bottom of the domain. Meanwhile, the shock amplifies the disturbance of the mixing layer resulting in vortex structures being created downstream of the impingement point. Near the outlet region, the reflected shock merges with the vortex structure of the mixing layer.

The Reynolds number of the flow is 500, based upon the average density of the two supersonic flows, their velocity jump, and the width of the mixing layer. The mixing layer width is defined as the distance between two inflexion points in the incoming

velocity profile that is sketched on the left hand side of fig. 6.1.

6.1.1 Computational Mesh

A rectangular domain was used for the present calculation with a non-dimensional length (along the horizontal x-axis) of 200 units and width (along the vertical y-axis) of 40 units. Figure 6.2 shows a number of mesh topologies that were tested. The following meshes were named after the identification given in fig. 6.2.

The first two meshes were uniform quadrilateral with 321 × 81 vertices (*mesh a*) and 641 × 161 vertices (*mesh b*). The coarser mesh, which had 26k vertices, served as the baseline mesh while the finer one that had 52k vertices was designed to study the refinement effect. Both were also used by Yee et al. Unless indicated, the baseline mesh is used for all of the calculations in this mixing layer study.

Two purely triangular meshes were generated using Gambit. The first triangular mesh, *mesh d*, had relatively uniform cells. The cell area was roughly half of that in *mesh a*. As a result, the mesh had twice as many cells as *mesh a* whilst having a similar number of nodes. The second triangular mesh, *mesh e*, had a non uniform distribution. The mixing layer region had nearly twice as many cells as *mesh d*, similar numbers of cells at the oblique shock region, but coarser in the remaining regions; giving a similar number of nodes to that of *mesh d*.

A hybrid mesh, *mesh c*, was also created using Gambit. Outside the mixing layer region, a similar strategy to *mesh e* was employed. The mixing layer itself was meshed by quadrilateral elements that were similar to *mesh a*, since this was where the most important flow features were expected. This mesh had a total of approximately 18k nodes, making it the most efficient mesh tested for this particular test case.

6.1.2 Boundary Conditions and Non-Dimensionalisation

The flow variables were non dimensionalised by upper stream density, temperature and sound speed. It must be noted that such a non-dimensionalisation was convenient for the present code even though it was different from that of Yee et al. [138]. Hence, the numerical quantities were redefined accordingly. Table 6.1 summarises the input values at the inlet and outlet boundaries. Inlet1 is the upper stream. and inlet2 is the

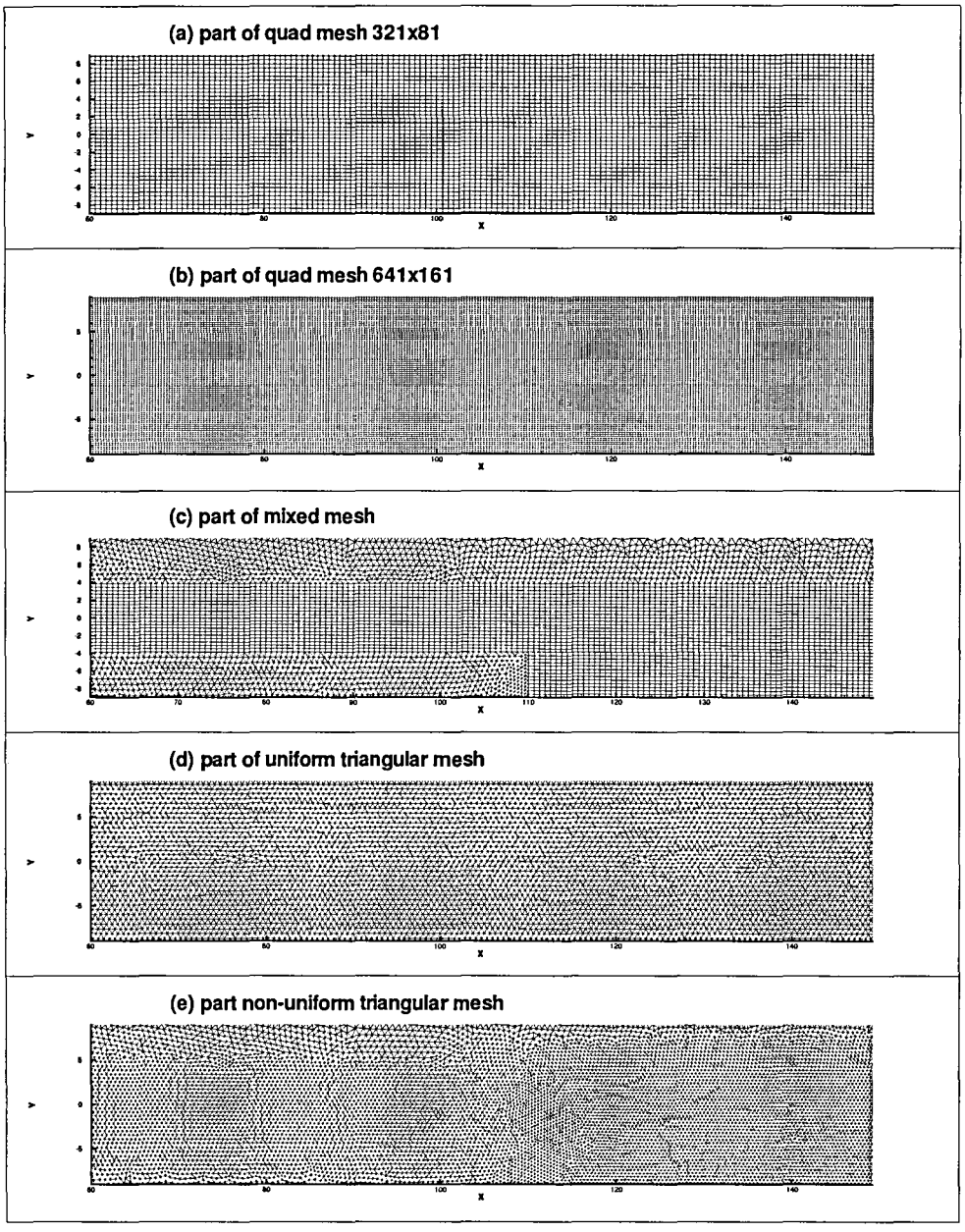


Figure 6.2: Part of computational grid. Only cells around the oblique shock impingement on mixing layer are shown

lower stream. The outlet boundary was not specified as the flow-field was completely supersonic within the computational domain.

property	inlet1	inlet2	top BC
u-velocity	5.6250	3.7502	5.5708
v-velocity	0.0000	0.0000	-0.2563
density	1.0000	0.2215	1.2887
pressure	0.7143	0.7143	1.0207
sound speed	1.0000	2.1250	1.0531
Mach number	5.6250	1.7647	5.2956

Table 6.1: Flow properties at inlet and upper boundary.

The supersonic inlet velocity profile was specified as a hyperbolic tangent function defined as follows

$$u = \frac{2.5 + 0.5\tanh(2y)}{0.5333} \tag{6.1}$$

where the mixing layer was placed at y = 0.0. Fluctuations were added by introducing a perturbation perpendicular to the mixing layer, at the inlet using the following cosine bi-harmonic function.

$$v' = \sum_{k=1}^{2} a_k \cos\left(2\pi k \frac{t}{T} + \phi_k\right) \exp(-y^2/b) \tag{6.2}$$

with period $T = 5.97$, and $b = 10$. For $k = 1$, $a_1 = 0.05$ and $\phi_1 = 0$, and for $k = 2$, $a_2 = 0.05$ and $\phi_2 = \pi/2$. No perturbation was added to the stream-wise component. Since the investigation was centred on the mixing layer and its interaction with the shock, the wall at the lower edge of the computational domain was simply defined as a slip wall boundary.

6.1.3 Results

Here, only qualitative results are presented since Yee et al. did not provided any quantitative data other than the maximum and minimum contour levels in their pressure and density plots.

In ref. [138], Yee et al. labelled TVD22 and TVD44 for computation with second order and fourth order TVD schemes, respectively. ACM22 and ACM44 was used to

indicate second order and fourth order central schemes with characteristic filter to minimise the artificial dissipation, respectively. For completion, their results are reproduced in fig. 6.3 and 6.4. All of the present results are plotted at the same instant as that of ref. [138]. Using the current non-dimensionalisation, this is 264 time units, which is equivalent to Yee et al. [138]'s non-dimensional time of 120 units.

Effect of computational mesh topology

Figure 6.5(a) and 6.6(a) show that the present computation using the baseline mesh, the 321×81 vertices, is capable of resolving the flow structures in the mixing layer, even though the Roe artificial dissipation is only scaled by 0.4 in the region. The resolution is significantly better than TVD22 and TVD44 results (fig. 6.3) and comparable to that in ACM22 and ACM44 (fig. 6.4). As expected, the finer quadrilateral mesh, fig. 6.5(b) and 6.6(b) produce a more well defined interaction between the shock wave and the vortical structure. It must be noted that an accurate comparison is difficult since the contour level used in ref. [138] is not given. The present density contours are plotted between 0.2 and 1.775 with 22 levels whereas the pressure is between 0.44 and 1.44 with 21 levels.

It is not surprising that both triangular meshes in fig. 6.5 and 6.6 give very diffusive results. However, they do not seem to be much worse than the TVD result shown in fig. 6.3. This enforces earlier work in section 3.3.5 (and also ref. [108]) which shows accuracy for time-dependent problems using a triangular mesh to be lower than an equivalent quadrilateral mesh. An improved result was obtained by clustering more nodes on the mixing layer region as shown in fig. 6.5(e) and 6.6(e).

The hybrid mesh gives comparable results to the baseline quadrilateral mesh in the mixing layer region since both of them have identical vertex density in this region. In other regions where triangles are used to discretise the flow domain, diffusive results (similar to fig. 6.5(d) and 6.6(d)) are observed. This is particularly obvious across the shock region which is not aligned with the mesh. Since the objective of the numerical experiment is to capture a shock wave mixing layer interaction and its development, this is not felt to be a problem. In fact, the hybrid mesh has actually managed to achieve the objective with a significantly smaller number of vertices than the baseline mesh.

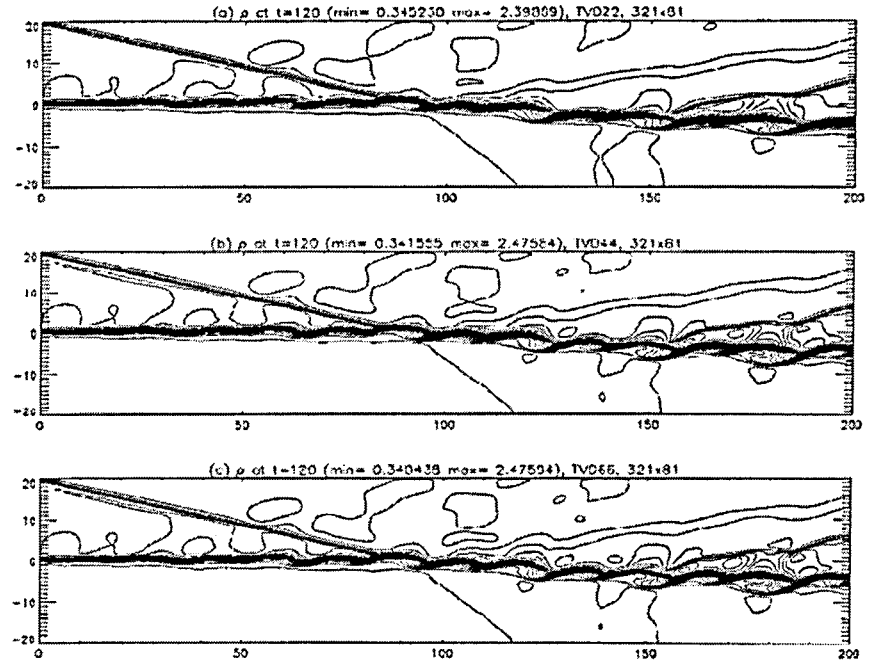


Figure 6.3: TVD results presented by Yee et al. [138]

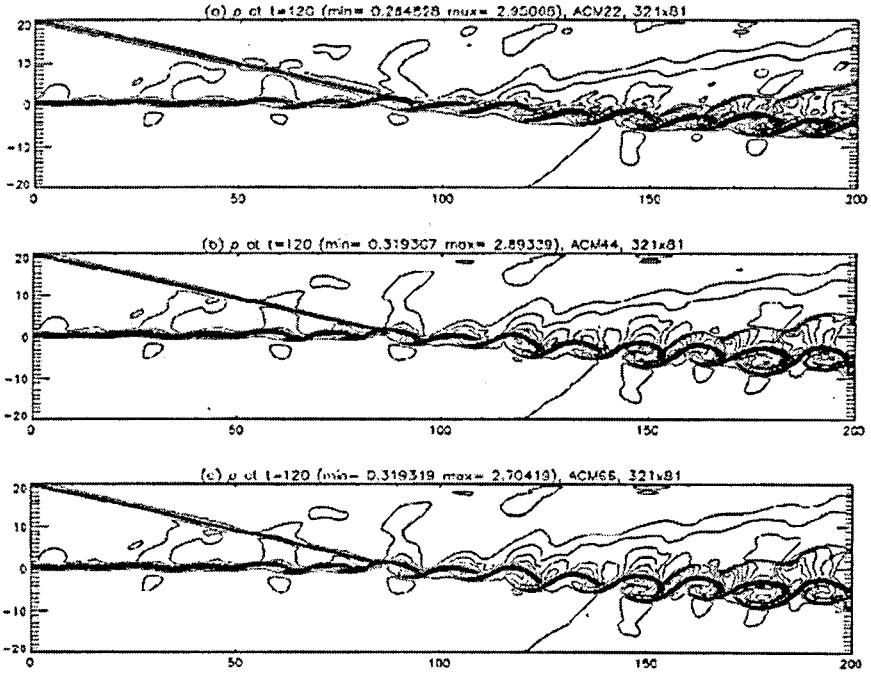
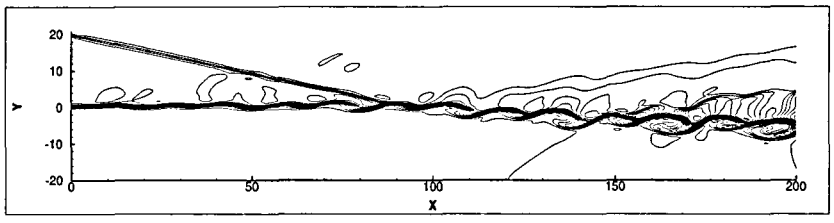
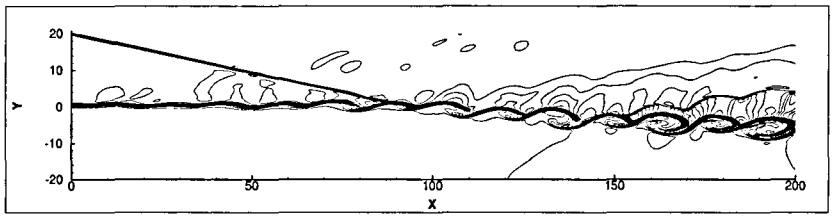


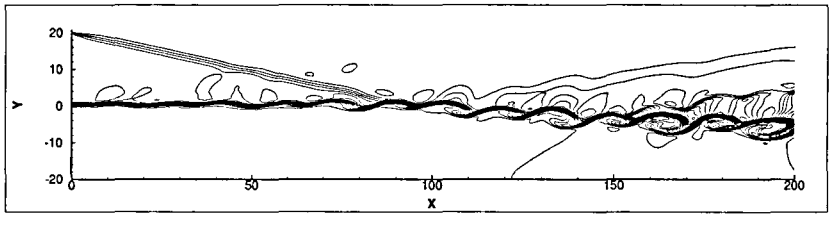
Figure 6.4: Minimum dissipation filter, ACM, results presented by Yee et al. [138]



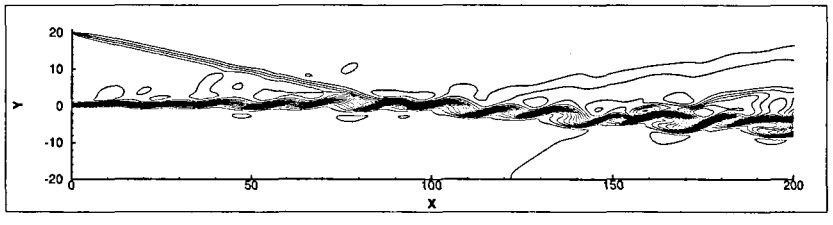
(a) baseline mesh: 321 × 81 vertices



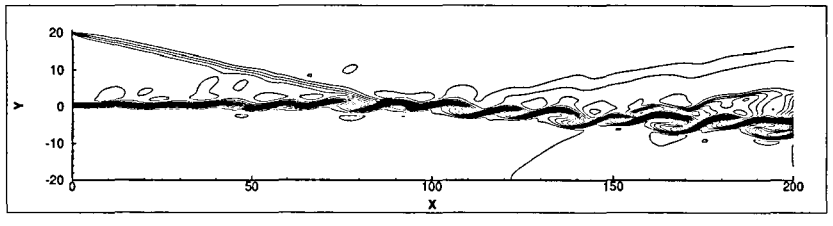
(b) 641 × 161 vertices



(c) hybrid mesh

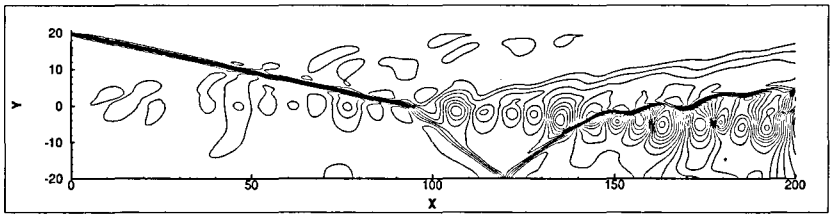


(d) uniform triangle mesh

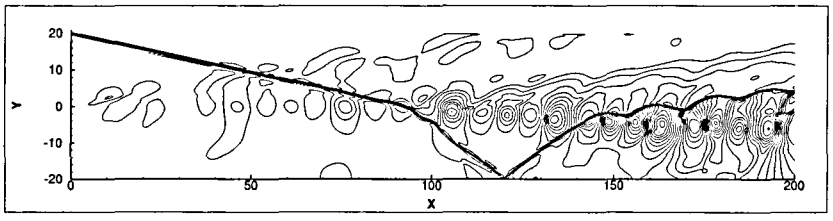


(e) non uniform triangle mesh

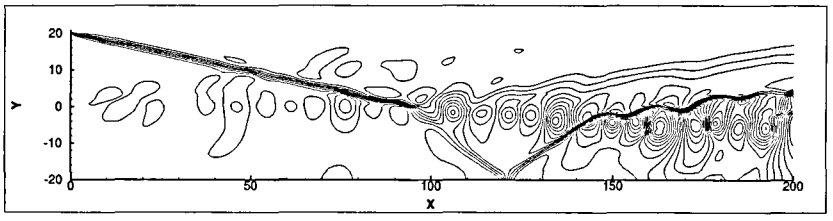
Figure 6.5: Comparison of density contours from 5 different meshes; Venkatakrisnan limiter $K = 10$ and Ω-switch with $\varepsilon = 0.4$



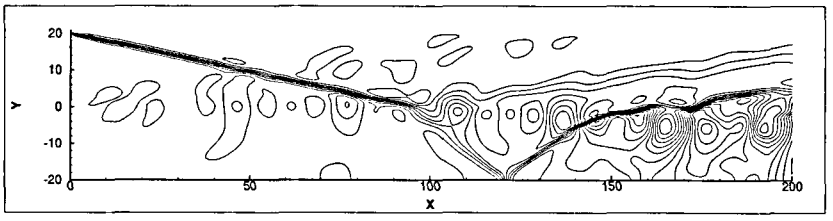
(a) baseline mesh: 321 × 81 vertices



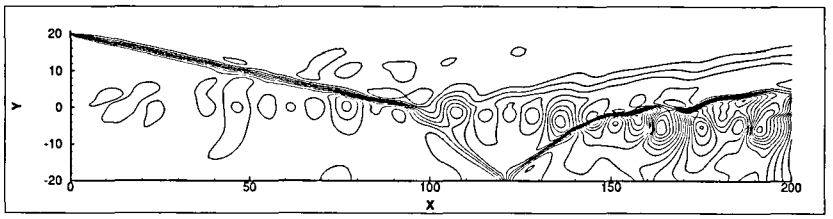
(b) 641 × 161 vertices



(c) hybrid mesh



(d) uniform triangle mesh

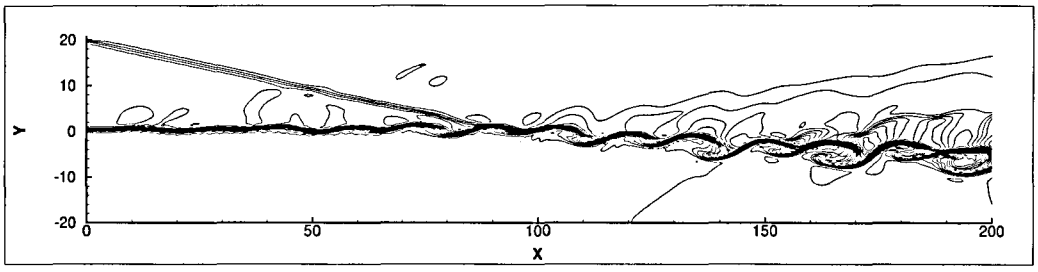


(e) non uniform triangle mesh

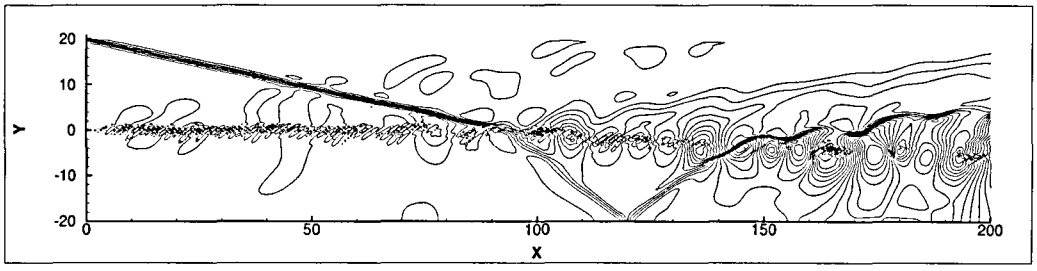
Figure 6.6: Comparison of pressure contours from 5 different meshes; Venkatakrisnan limiter $K = 10$ and Ω-switch with $\epsilon = 0.4$

Effect of artificial dissipation scaling and flux limiter

Comparisons of density plots shown in fig. 6.6(a) and fig. 6.7(b) shows that ε can only assume a certain value before the odd-even oscillation takes place in the mixing layer in front of the shock impingement point. Although less obvious, such oscillation is also exhibited in the density field when comparing between fig. 6.5(a) and fig. 6.7(a)



(a) density contour



(b) pressure contour

Figure 6.7: Odd-even decoupling on the density and pressure contour; unlimited; $\varepsilon = 0.05$ for Ω-switch

Figure 6.8 shows the instantaneous contour plot of the dissipation switch. High values indicate that 'upwinding' becomes active. It must be noticed that the switch is also activated in the irrotational flow region as well as around the shock region.

Unlike the artificial dissipation of the convective scheme, the effect of the flux limiter is hardly noticeable in fig. 6.9. However, the minor effect observed in fig. 6.9 does not necessarily mean that unlimited flux can be used in any flow problem. Thus, its application should be restricted to flow problems where robustness is important.

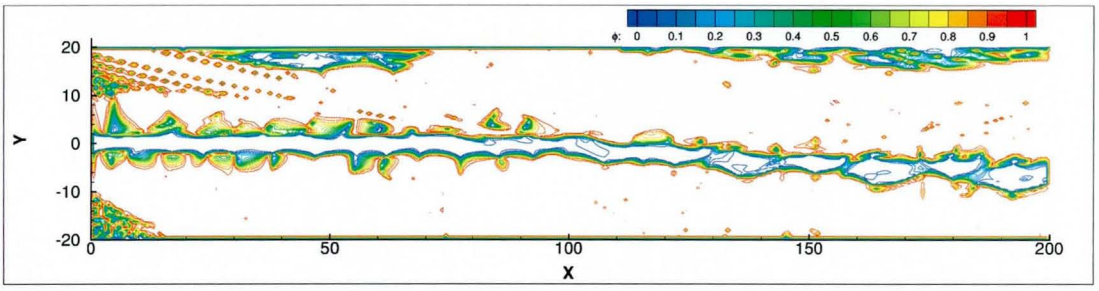
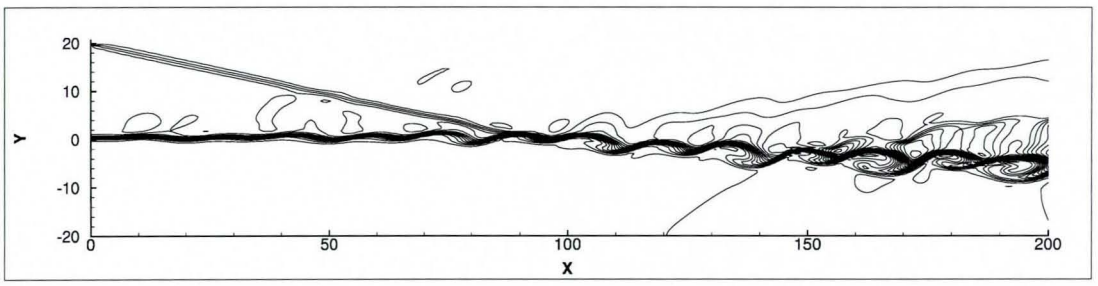
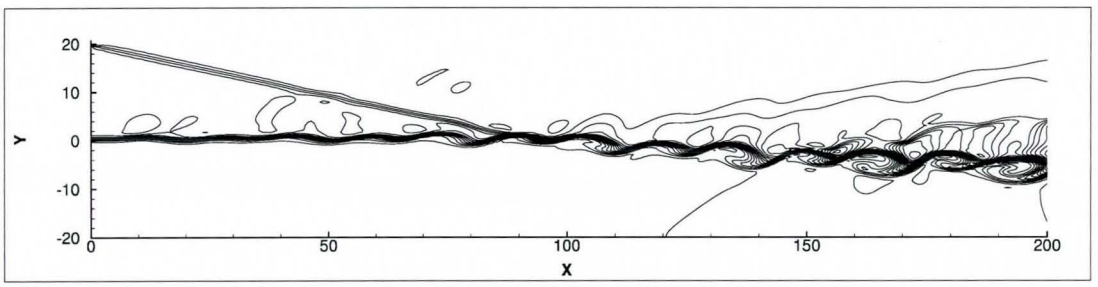


Figure 6.8: Instantaneous contour of artificial dissipation switches for shock wave-mixing layer interaction



(a) unlimited flux



(b) Venkatakhrishnan flux limiter $K = 10$

Figure 6.9: Limiter effect on shock wave-mixing layer interaction

6.1.4 Closure

All of the contour plots indicate that the present solutions capture the physics of the flow. The plots shows how the disturbance in the mixing layer is amplified by the shock and eventually develops into a vortical structure. The alternate maximum-minimum pressure regions along the mixing layer are also shown clearly in the pressure plots. The lower oblique shock and its reflection is only captured clearly on the pressure plot since the shock is too weak to show marked density changes for the chosen contour levels. These features are smeared in triangular meshes but can be observed very clearly when quadrilateral cells are used. Since the low quality resolution is acceptable in the less important region, a hybrid mesh offers the best compromise since the triangular cells can be generated easily. As shown by the experiment, the hybrid mesh uses a smaller number of nodes than the more rigid quadrilateral cells for the same resolution at the region of interest. Hence such a meshing strategy is preferred for the rest of the thesis.

6.2 Large-Eddy Simulation of Mach 0.9 Round Jet

Turbulent jet flows are important from an engineering point of view as these have been long identified as a major contributor to aircraft noise. Thus any attempt to design a quiet aircraft inevitably needs better understanding of the noise generation mechanisms in a turbulent compressible jet. Moreover, from a fluid mechanics point of view, jet flows are often seen as a prototype for complex turbulent flows with swirl, separation, recirculation, curvature, body force etc. [87]

The Mach 0.9 round jet is of particular interest here; this case is well documented and it allows the validation of the present code for a practical problem. Experimental results of Lau et al. [66], and Stromberg et al. [120] that have been conducted at Mach 0.9 serve as a reference. Each of the experiments were designed to investigate different aspects of the flow. The former was focused on the high Reynolds number aerodynamics whilst the latter was designed to study jet acoustics at a low Reynolds number ($Re_D = 3500$). For these problems, the jet Reynolds number is based on nozzle exit speed(U_J) and diameter (D). Being performed at low Reynolds number, the Stromberg experiment is particularly suitable for DNS as well as LES studies.

Although some nozzle exit Mach number effects have been reported by Lau et al. [66] and Zaman [143] regarding the potential core length and mass flux variation of a fully expanded supersonic jet, they do not extend beyond the near nozzle flow field. As jet self-similarity is only observed further downstream, where the Mach number is much lower than the nozzle exit, the extensive incompressible jet measurement of Hussein et al. [56] and Panchapakesan and Lumley [87] have also been used as reference. Freund [38] has demonstrated this through his DNS work that was set to match the Stromberg experiment [120].

There have been a number of literature reports of LES computations of the Mach 0.9 round jet using structured meshes. Ghosal's study of the interference between the convective numerical scheme and the SGS model [45] lead to the growing popularity of the higher order compact scheme of Lele [67]. Following this trend, Boersma and Lele [14] and Uzun et al. [128, 129] conducted their simulations using a sixth order numerical scheme for the convective terms with compact filtering to smooth out the velocity gradient for the strain computation. In addition, Uzun et al. [128] has also demonstrated the superiority of the sixth order compact scheme over its fourth order counterpart. This finding was merely a confirmation of the one dimensional Fourier

analysis such as presented in chapter 3. Apart from the standard Smagorinsky model in the earlier paper of Uzun et al. [128], these two studies have been quite similar, namely the application of dynamic Smagorinsky model [44] and $Re_D = 36000$. The main difference between the two studies lies in the choice of mesh topology. Whilst Boersma and Lele conducted their computation on a circular cylindrical computational domain discretised by a single block curvilinear mesh (using a polar topology), Uzun et al. employed a single block Cartesian mesh for their square cylinder domain. Bogey et al. [15] also conducted an LES computation of the same jet problem using the high order Dispersion-Relation-Preserving (DRP) scheme of Tam and Webb [121]. This was an attempt to perform both a turbulent flow simulation and an acoustic wave propagation in a single calculation. In all of these studies a velocity profile was defined at the inlet domain to model the jet exiting from the nozzle without including the nozzle itself. It is interesting to note that two compressible jet studies (not Mach 0.9) of Lupoglazoff et al. [73] and Shur et al. [107], who were motivated by the practical application of LES methods in aeronautical engineering, have chosen to employ a numerical scheme with scaled down upwinding , discarding the SGS model, but did include the nozzle geometry.

At present, two SGS models have been implemented, namely the standard Smagorinsky and the WALE model of Nicoud and Ducros [84]. Although the standard Smagorinsky model is frequently considered to be unsuitable for jet flows, Uzun et al. [128] showed that this model performed reasonably well in comparison to that with the dynamic procedure [129]. This result has also been supported by Bogey et al. [15]. Consequently, the constants for the standard Smagorinsky model have been utilised from Uzun et al. [128], $C_s^2 = 0.018$. Since the author is not aware of WALE model applications for jet flows, $C_w^2 = 0.25$ has been applied for the WALE constant as recommended by Nicoud and Ducros [84] for generic applications. In both computations the compressibility correction is taken from ref. [128], i.e. $C_I = 0.0066$. Following Boersma and Lele [14] and Uzun et al. [128], the present simulation was set at $Re_D = 36,000$.

6.2.1 Computational Domain and Mesh

The computational domain was simply a cylinder, with radius $R = 8D$, oriented along the x-axis (axial length $22.5D$). Following Freund, robustness at the outlet region was ensured by creating a sponge zone at $22.5D \geqslant x \geqslant 30D$. To reduce the computational cost, instead of modelling the nozzle geometry, a velocity profile that accounted for the

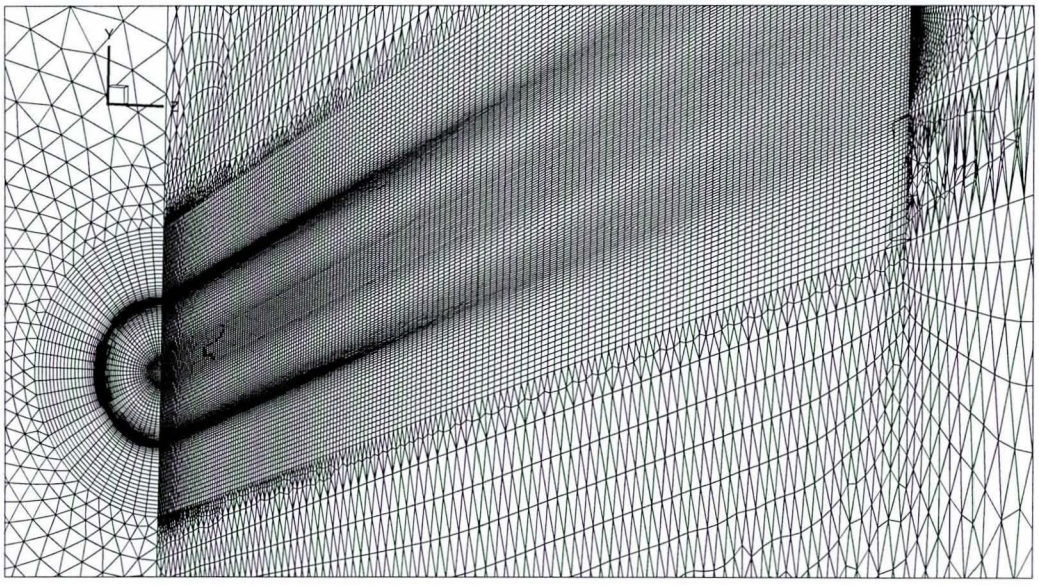


Figure 6.10: Computational mesh for the turbulent round jet simulation

nozzle effect was specified at the inlet plane.

A hybrid mesh was generated using Gambit to discretise the computational domain, as shown in fig. 6.10. Such a simple computational domain could have been discretised rather easily by a curvilinear structured mesh, but a hybrid meshing has been chosen as it offered a more efficient mesh by employing a structured like hexahedral mesh in the jet region and a tetrahedral mesh, which could rapidly coarsen, at the far field and sponge region. Using a similar strategy to section 6.1, hexahedral cells were applied in the upstream shear layer region as well as the downstream jet region resulting in a comparable mesh resolution to that of Boersma and Lele [14]; whereas the tetrahedral cells outside this region lead to a noticeably coarser discretisation in the far-field region. As a result, the present mesh only requires 800k nodes as opposed to 1.6M in the structured mesh of ref. [14]. It must be noted that the aggressive degree of coarsening in the far-field region shown in fig. 6.10 was possible since direct aeroacoustic computation was not the objective here. The singularity along the axis of the cylindrical computational domain was avoided by rearranging the hexahedral cell differently from that of the shear layer region (fig. 6.11).

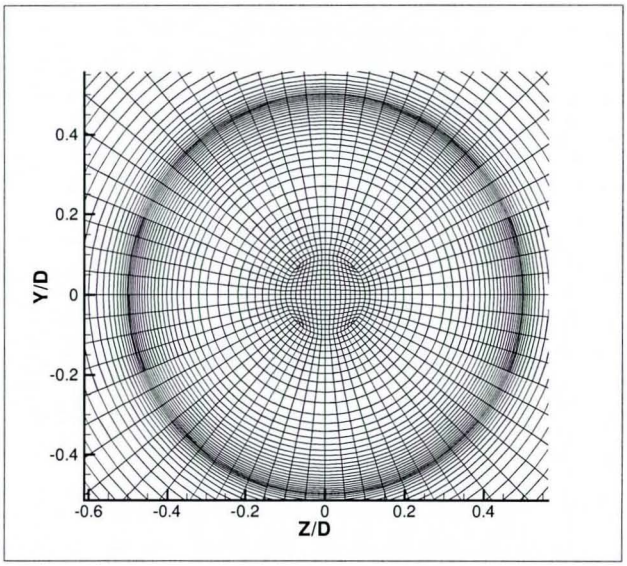


Figure 6.11: Computational mesh around jet centre line

6.2.2 Boundary Conditions and Computational Procedure

Following, Boersma and Lele [14] (also Freund [38]) a profile similar to the mixing layer test case was specified at the inflow plane $x = 0$. A hyperbolic tangent axial profile was superimposed with a harmonic function whose frequency matched that of the excitation in the Stromberg experiment, and was specified for u-velocity as follows:

$$u = U_J \left[\frac{1}{2} - \frac{1}{2} \tanh \left(b \left(\frac{r}{R} - \frac{R}{r} \right) \right) \right] (1 + \beta \sin (2\pi 0.5t)) \tag{6.3}$$

where $R = 1/2D$ was the jet radius, b was the mixing layer thickness and β was the amplitude of the harmonics. A random perturbation was added to the circumferential (tangential) velocity component ru_θ to promote three dimensional flow as follows

$$ru_\theta = 0.025 Rand \left(e^{-3(1-r/R)^2} \right) \tag{6.4}$$

Notice that eq. 6.4 describes the tangential ru_θ rather than the angular u_θ velocity component. Obviously, rather than having a turbulent initial shear layer, the jet had a perturbed laminar layer whose resulting potential core length was affected by the choice of b and β. Here, the thicker and more intensely perturbed layer of Uzun et al. [128] was chosen as oppose to that used Boersma and Lele [14]. Thus $b = 3.125$ and $\beta = 0.005$. As a result, the potential core length in the present computation was markedly shorter than that of Boersma and Lele. This adjustment of the velocity profile

has lead some authors to argue that it is better to model the nozzle geometry and its delivery pipe [26], [73]. Since most jet experiment have not been conducted with a fully developed turbulent flow in the delivery pipe, they argued that no perturbation would have been necessary in the delivery pipe inlet. The main objective here is to demonstrate that the unstructured approach can produce solution of similar accuracy to an existing structured code, and so it was felt better to use the same velocity profile approach. Nevertheless, the unstructured approach makes it extremely simple to include the nozzle geometry (in contrast to Cartesian mesh solvers).

Uzun et al. [128] and Boersma et al.'s DNS [13] indicates that the lateral boundary plays an important role in ensuring correct entrainment, which in turn affects the jet's axial velocity decay. This was particularly true since the application of eq. 6.3 as an inlet profile implies that the jet emanated from a wall. Consequently the flow entrainment only took place from the lateral boundary. Uzun et al.[128] found that the non reflective boundary condition that is often used for RANS simulations of jet flow (see [124] for instance) had been responsible for incorrect axial velocity decay. Hence, the lateral boundary was designed to act as subsonic inflow boundary (with ambient total pressure and temperature) if the flow tried to come into the domain and subsonic outflow boundary condition (with ambient static pressure) otherwise. Such treatment was intended to ensure the correct flow entrainment in order to achieve a well behaved jet decay and spreading. A similar boundary condition was also employed by Lupoglazoff et al. [73].

The subsonic outlet boundary condition – with static pressure specified at the free stream value – were applied at the outflow plane $x = 30D$. It was realised that imposing static pressure at the outlet was not strictly correct since the outlet static pressure would not be uniform. However, the weak formulation of the boundary condition did allow some pressure variations on this plane. Furthermore, this boundary condition was felt to be compatible with the sponge zone whose main task was to make the flow behaved like a steady flow as much as possible by the time it reached the outflow plane.

The LES computation was performed on an Intel® Xeon 2.4 GHz based PC cluster. As in the previous section, the timing of the parallelism was not considered to be an important issue. The computation was started from a uniform flow-field without superimposing any white noise since the non-linear disturbance was readily available from three sources: the inlet boundary, the Helmholtz instability in the shear layer, and perhaps most importantly from numerical point of view was the perturbation caused

by the initial vortex pair created when the high speed jet front advances through the quiescent fluid in the computational domain. No limiter has been applied in any of these computations.

The time step was set to 0.005 time unit, which was comparable to ref. [14]. The statistics were not collected until statistically steady kinetic energy of the flow-field had been achieved. A further 100000 iterations were then performed to collect the statistics.

6.2.3 Cylindrical Coordinate Post Processing

Statistically averaged quantities in the round jet will be axisymmetric. Thus. to increase the effective number of samples when computing mean and fluctuating statistics, circumferential averaging is employed. Whilst this is straightforward for a cylindrical polar structured mesh, in the present work using unstructured mesh the averaging is performed over an annular stripe extracted from the cross sectional slices. Detail of the procedure is given in Appendix B. As a result fig. 6.16 only shows a few points coming from such slices to estimate the spreading rate.

6.2.4 Results and Discussion

Qualitatively, both the WALE and standard Smagorinsky computations display the correct physics of a turbulent round jet. Fig. 6.12 shows instantaneous vorticity coloured by ω_x direction for the WALE SGS solution. Blue stands for negative rotation (clockwise) and yellow for positive. Initially, the jet shear layer enters the domain in a laminar state. Eventually the instability in the shear layer causes vortical structures to be formed and transition to a turbulent shear layer commences. The thickening shear layers soon merge to end the potential core.

The mean local Mach number profile is shown in fig. 6.13. The development of the jet from a top hat profile near the nozzle (fig. 6.13(a)) through the transition of the shear layer (fig. 6.13(b) to a fully developed turbulent flow 6.13(c)) is clearly shown. The potential core in both the Smagorinsky and WALE calculations are close to that of the Stromberg experiment [120]. The Smagorinsky computation shows a slightly longer potential core length suggesting that the shear layer is less energetic. Hence the standard Smagorinsky model with $Cs^2 = 0.018$ is more dissipative than the WALE model with $Cw^2 = 0.25$.

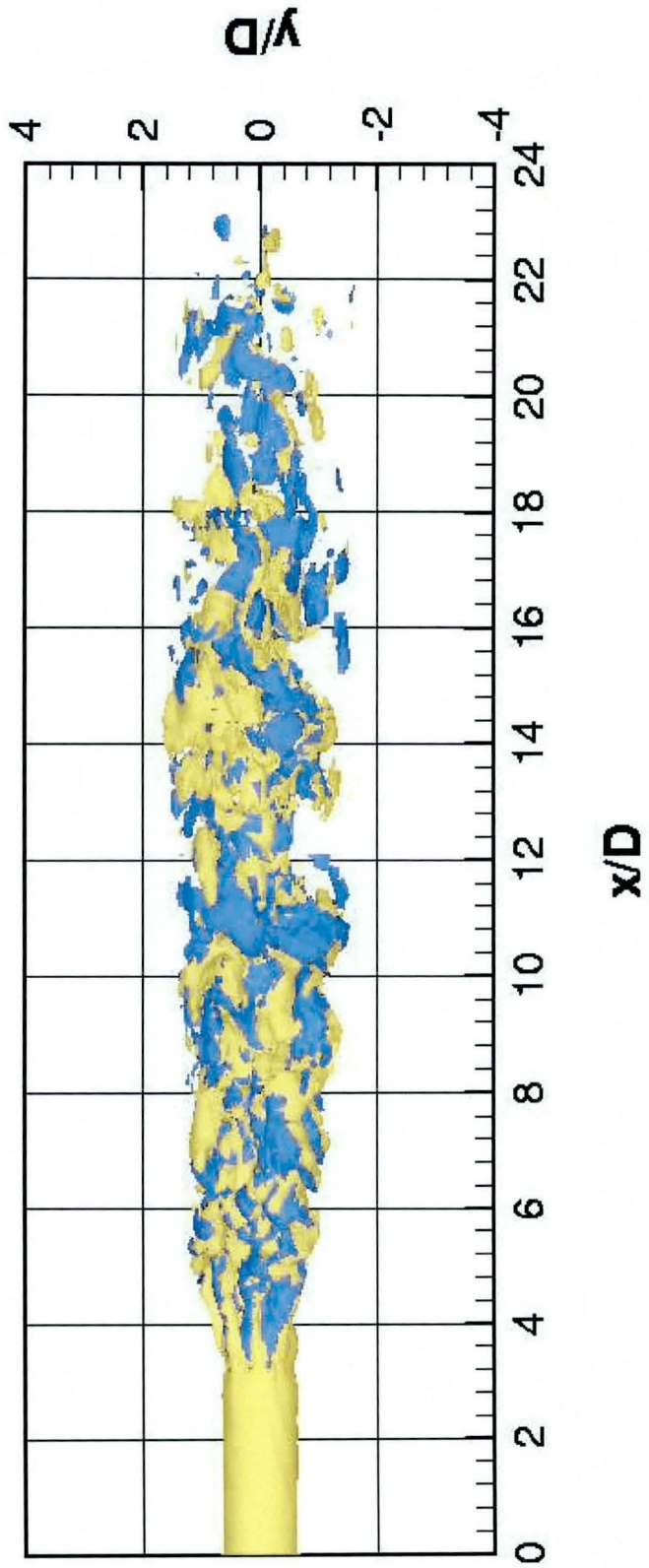


Figure 6.12: Instantaneous vorticity based on $|\omega| = 1.0$ from the WALE test.

It is important to mention that DeBonis and Scott [26], who included the nozzle geometry, found that their potential core length is shorter than experimental data. The same result had also been observed by Lupoglazoff et al. [73], who did not implement any explicit SGS model. On the other hand, potential core length tends to be over predicted when a prescribed velocity profile is used, This is not only found in LES computation (see Boersma and Lele, Gamet) but also in DNS (see Boersma et al. [13] and Freund [38]). This suggests that these low Reynolds number jet problems are sensitive to the state of the initial shear layer. This may be less important for high Reynolds number jets.

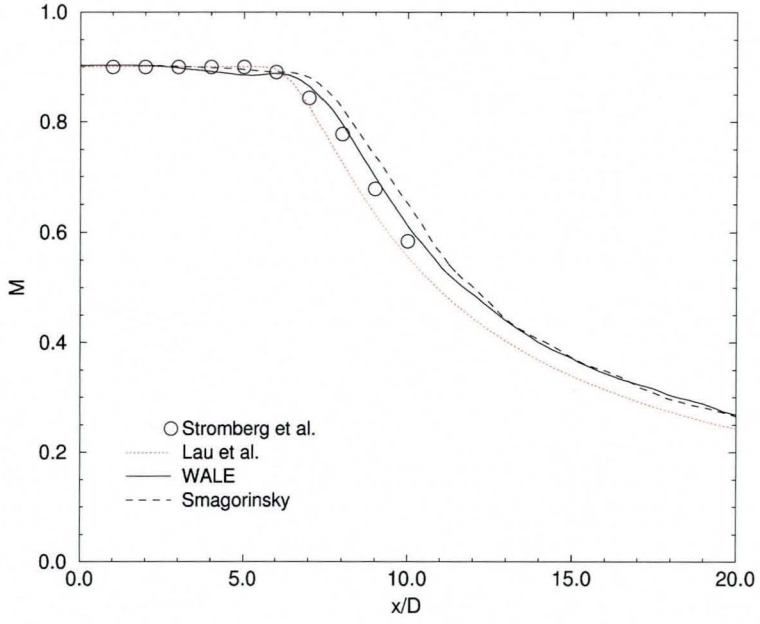
The sensitivity to the inlet condition is also shown in fig. 6.14. Here the absence of turbulence fluctuation within the jet core at the inlet boundary condition of eq. 6.3 and 6.4 manifests as zero RMS of axial and radial fluctuations around the inlet region of the jet centre line. When the turbulent shear layers merge, the fluctuation is observed on the jet centre line. However, the magnitude of the peak fluctuation and the rate of increase upstream of the peak suggests a higher turbulence level in the computation, which indicates that the break-up occurs at a faster rate than that of observed by Lau et al. [66]. From this it can be deduced that, while the experimental shear layer is turbulent, the LES computation is laminar and hence more susceptible to a disturbance that leads to shear layer break up.

Experiments on high Reynolds number fully developed turbulent round jets [87] have shown that the inverse of axial velocity decay can be expressed as

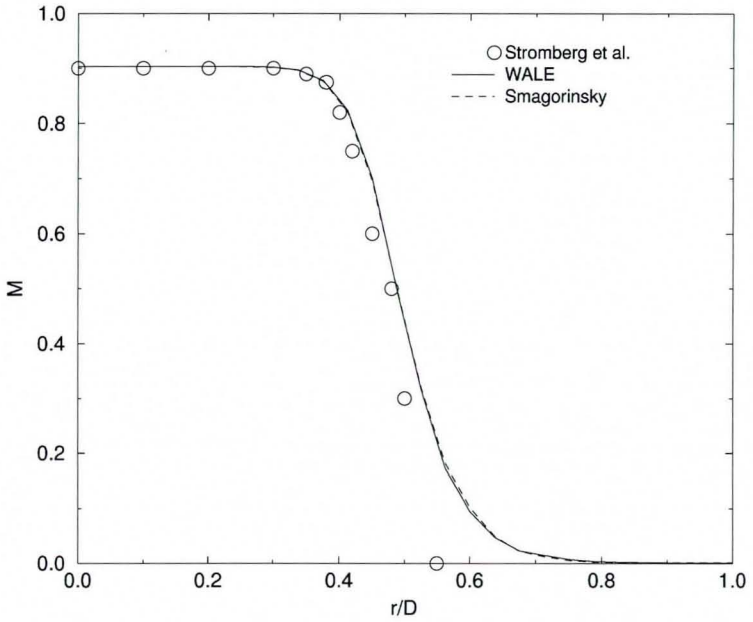
$$\frac{U_J}{U_c} = \frac{1}{Bu}\frac{x - x_0}{D} \tag{6.5}$$

where U_c is the local centre line axial velocity and x_0 is the datum position of the decay. Figs. 6.15 show the U_J/U_c slope is 0.169 and 0.185 for WALE and standard Smagorinsky model, respectively. These values are within the range of experimental measurements which is 0.165 to 0.185 [87]. In term of decay constant Bu, these values are 5.92 and 5.41 for WALE and Smagorinsky model, respectively, whereas the experimental range are 6.06 to 5.41. Figure 6.15 also indicate that $x_0/D = 1.59$ for the WALE and $x_0/D = 2.8235$ for the standard Smagorinsky models.

Figure 6.16 shows the predicted spreading rate. The trend line is computed using linear regression while forcing the x-axis cross over point x_0 to be identical to that obtained from fig. 6.15. The present spreading rates are 0.0979 and 0.1085 for the WALE and standard Smagorinsky models, respectively, which are slightly higher than

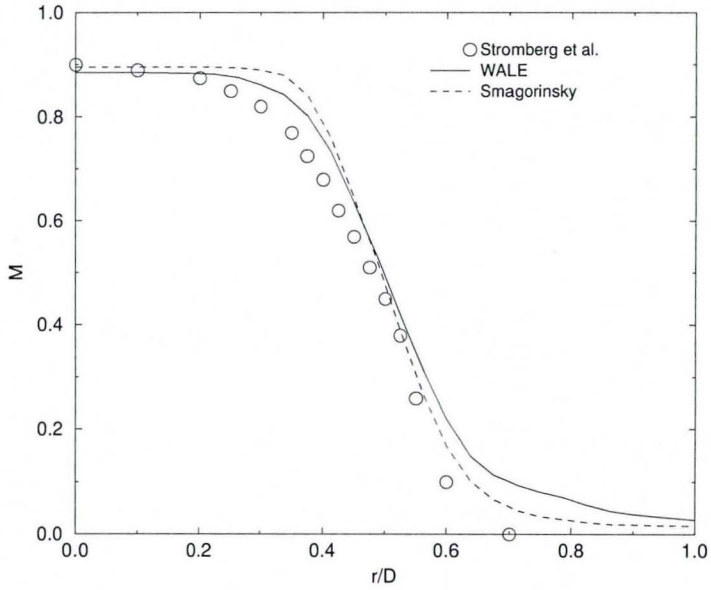


(a) Jet centre line

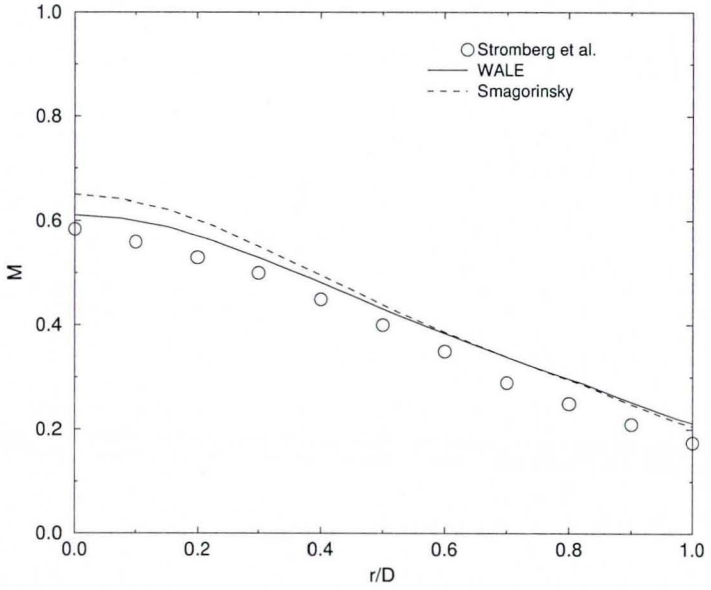


(b) x/D=1.0

Figure 6.13: Local Mach number profile

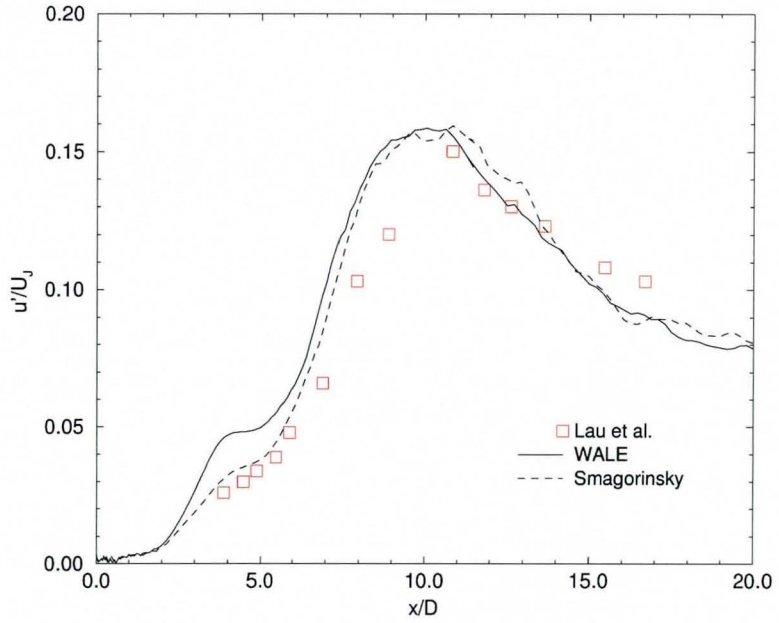


(c) x/D=5.0

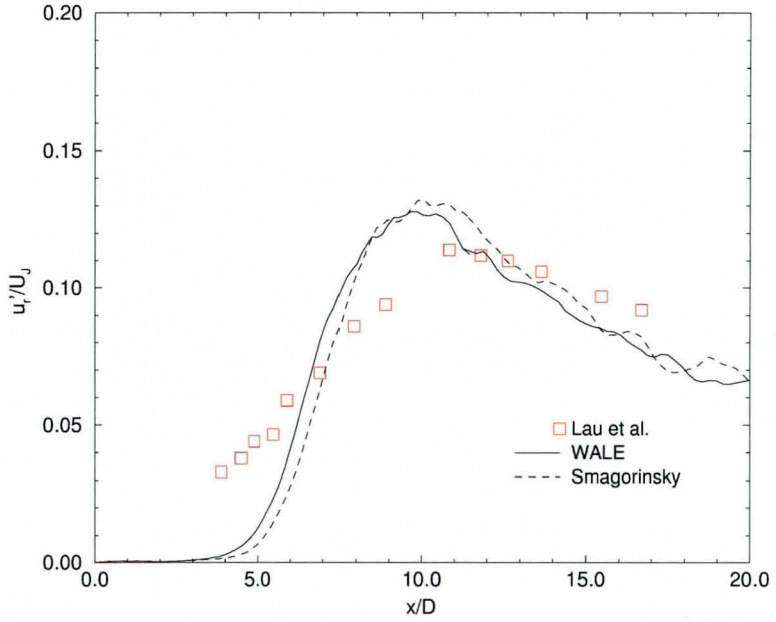


(d) x/D=10.0

Figure 6.13: Local Mach number profile (cont'd)

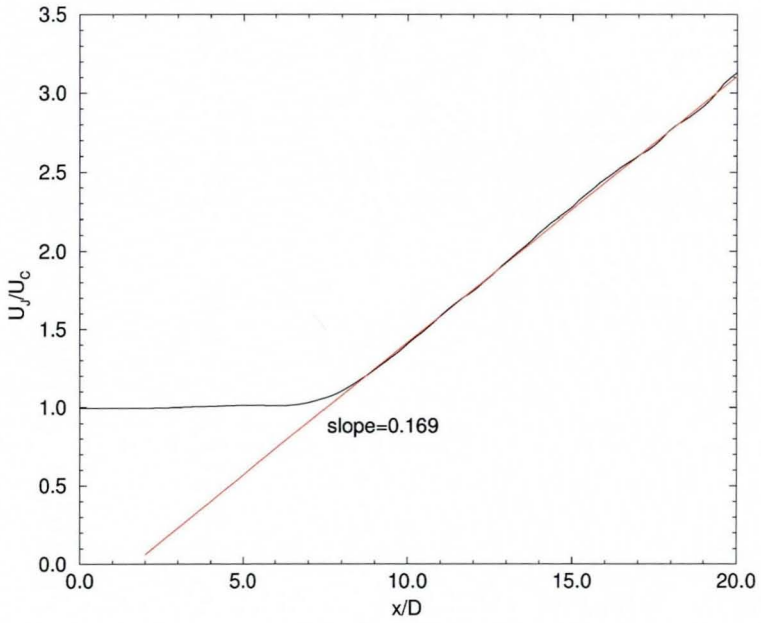


(a) axial fluctuation component

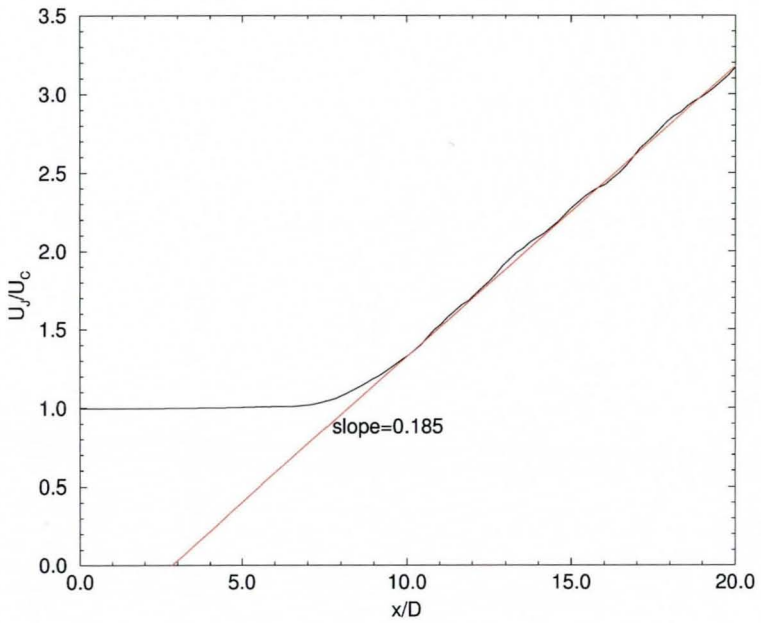


(b) radial fluctuation component

Figure 6.14: Root-mean-square turbulence velocity distribution along the centre line



(a) WALE



(b) Standard Smagorinsky

Figure 6.15: Variation of local centre line velocity, U_C

experimental data ranging from 0.086-0.096 [87]. However, this result is comparable to the LES computation of Uzun et al. [128] and the DNS computation of Freund [38].

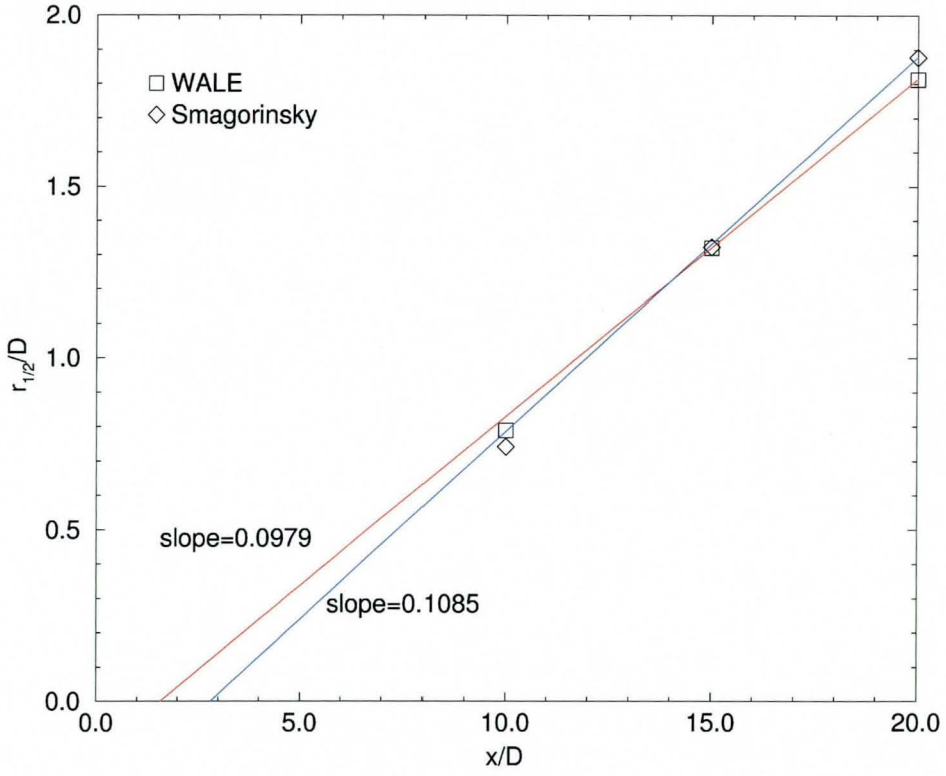
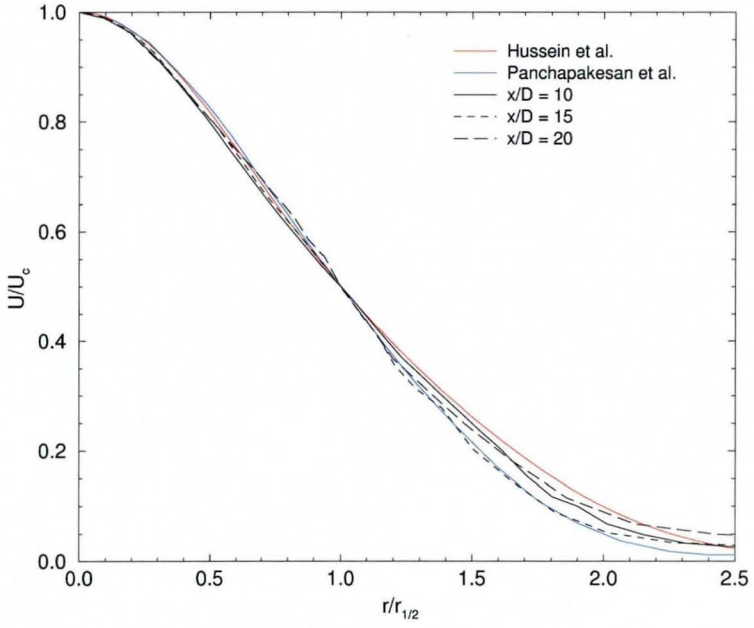


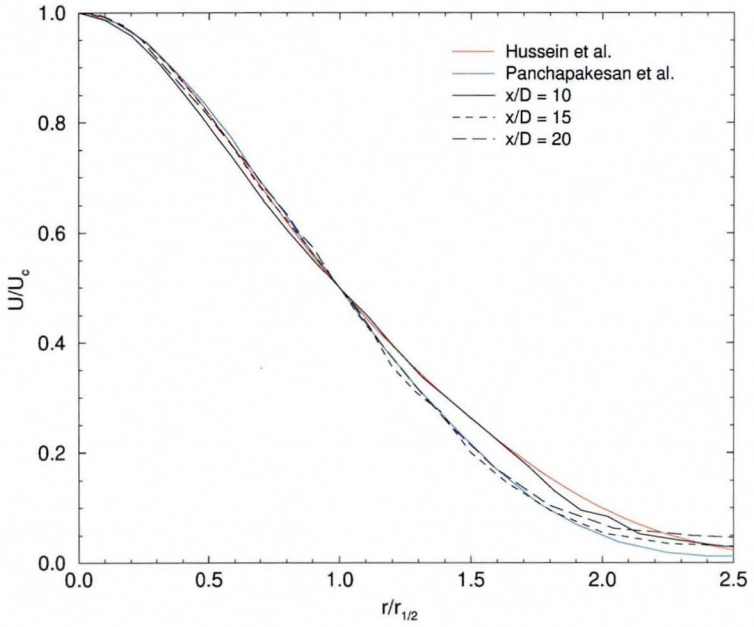
Figure 6.16: Comparison of spreading rate based on $r_{1/2}$, the radial coordinates where the velocity is half of that in the centre line

Self-similarities are checked at three downstream stations, namely $x/D = 10, 15$ and 20. At these stations the local Mach number has decayed to the extent that it is reasonable to compare it with the low speed experiments of Hussein et al. [56] and Panchapakesan and Lumley [87]. Figure 6.17 shows self-similarity of the mean velocity downstream of the shear layer merging point. Although Hussein et al. [56] and Panchapakesan et al. [87] did not start their measurements before $x/D = 30 - 40$, the present computation shows remarkably good self-similarity of the main flow as early as $x/D = 10$, which is similar to other LES computations. Similar to the findings of Uzun et al. [129], fig. 6.17 clearly shows that the present computations match the Panchapakesan et al. measurements [87] better than that of Hussein et al. [56].

Figure 6.18 shows Reynolds stress, non dimensionalised by the square of skin friction

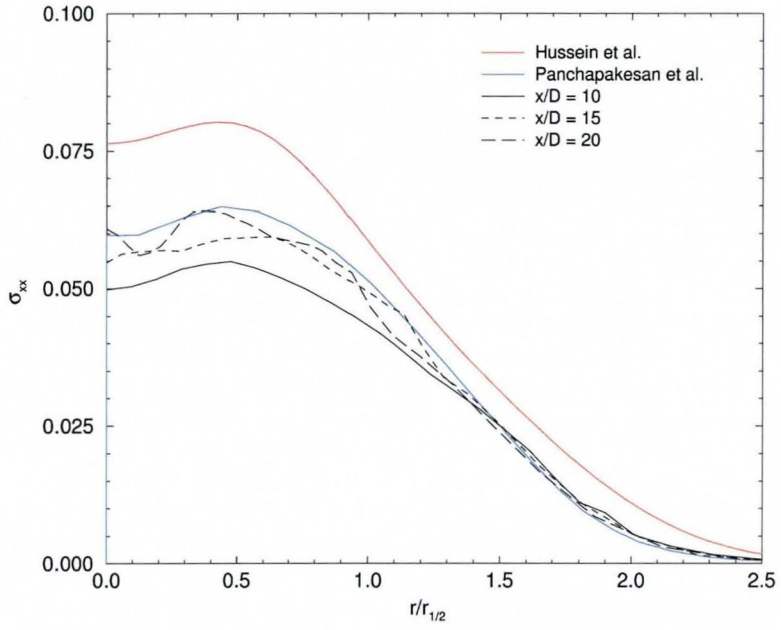


(a) WALE

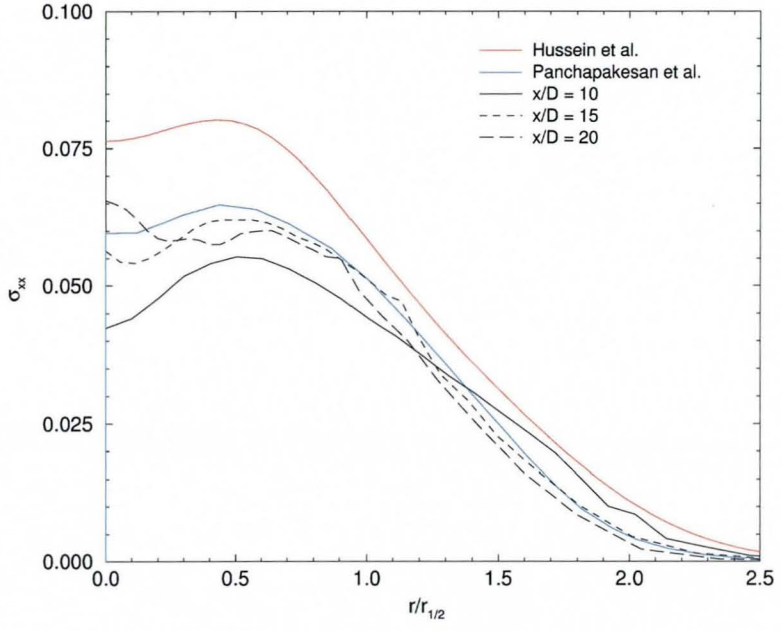


(b) Standard Smagorinsky

Figure 6.17: Self similarity of the mean axial velocity

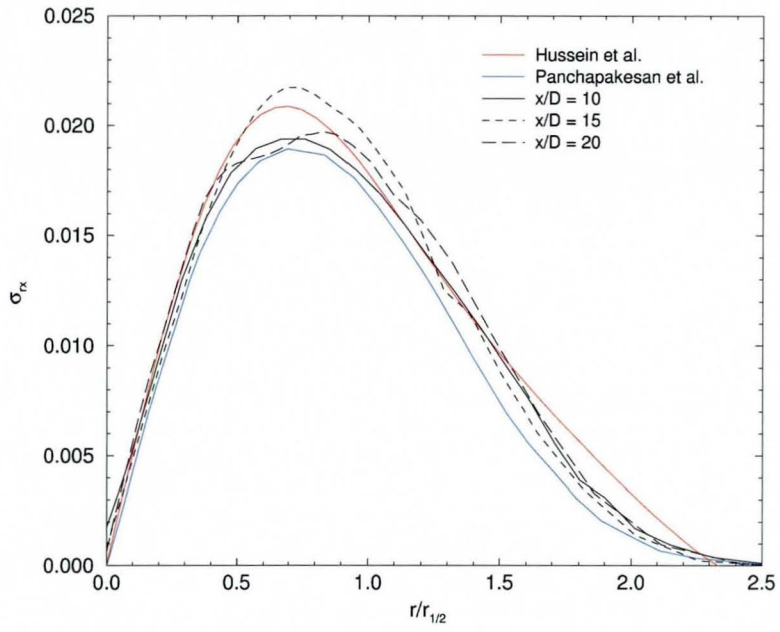


(a) WALE

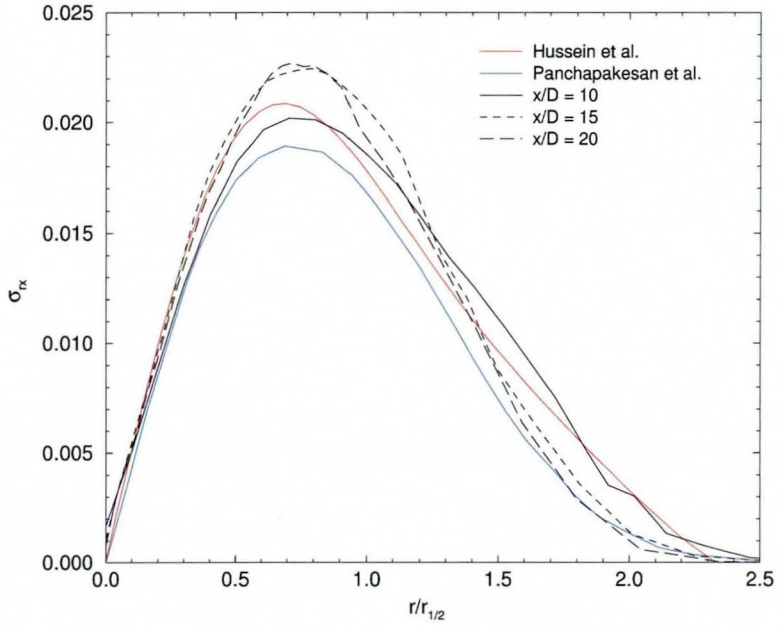


(b) Standard Smagorinsky

Figure 6.18: Self-similarity of σ_{xx} Reynolds stress component



(a) WALE



(b) Standard Smagorinsky

Figure 6.19: Self-similarity of σ_{rx} Reynolds stress component

velocity, at several axial stations typically reported by many other LES computations. Here, self-similarity is also observed for the Reynolds stresses even though it is not as good as the mean velocity profile, indicating that the self similarity of the Reynolds stresses is only observed at further downstream region. In fact, Boersma et al. [13] identified that self-similarity of the mean flow should be observed from $x/D = 10$ whereas that of Reynolds stress is observed much later at $x/D = 30$. Once again, the self-similarity data shows a better agreement with the Panchapakesan measurements [87] than that of Hussein et al. [56].

6.2.5 Closure

In general the present code has captured the flow physics of the compressible turbulent jet flow well. Although some discrepancies are still observed, these are similar to other LES computations, typically based on higher order (fourth or sixth order accurate) spatial numerical schemes using structured meshes. This proves that the application of a hybrid unstructured mesh does not have any detrimental effect on the LES resolution in comparison to the more conventional structured mesh approach, in spite of the fact that the former employs a smaller number of nodes.

6.3 Large-Eddy Simulation of a low Mach number fully developed turbulent pipe flow

In the light of recent works published by Mossi and Sagaut [83], a Large-Eddy Simulation of an infinitely long pipe or channel flow is regarded as a good test case to validate the capabilities of a numerical method to sustain turbulence. This is particularly relevant for the present code due to the inherent numerical dissipation of the base scheme. Numerically, such a simulation is carried out using a pair of periodic boundary conditions for the inlet and outlet of the computational domain, thus the high strain rate near the wall is the sole mechanism of turbulence generation. This means that such a flow is also useful in assessing the near wall behaviour of a SGS model. Whenever the numerical or SGS dissipation is too high, after some number of time steps the turbulence will dissappear completely as the flow re-laminarises numerically.

A fully developed pipe flow is used to validate the present code for wall bounded turbulent flow. Despite being not as popular for this task as a channel flow for LES computations, a large body of data on fully developed turbulent pipe flow has recently been documented by an AGARD committee [82]. Interestingly, the report suggests that the popularity of the channel flow is mainly caused by difficulty in dealing with cylindrical coordinates and the consequent singularity at the centre line, when using a typical single block structured mesh. Furthermore, Eggels et al. [33] has shown that pipe and channel flows shared many similar characteristics.

In this section, a test case similar to that of Nicoud and Ducros [84], which was carried out at a Reynolds number $Re_b = 10000$ based on the bulk axial velocity u_b and pipe diameter. With the absence of experimental measurement at this particular Reynolds number, Durst et al. [32] measurements, which were performed a $Re_b = 7442$ are used as a comparison. In spite of this, the higher Reynolds number experiments of Durst et al. [32] shows similar trend in velocities mean and RMS fluctuation profiles among the various Reynolds number data. Since Durst data is incompressible, the present computation is designed to have nominal bulk Mach number $M_b = 0.25$, which is also used by ref. [84].

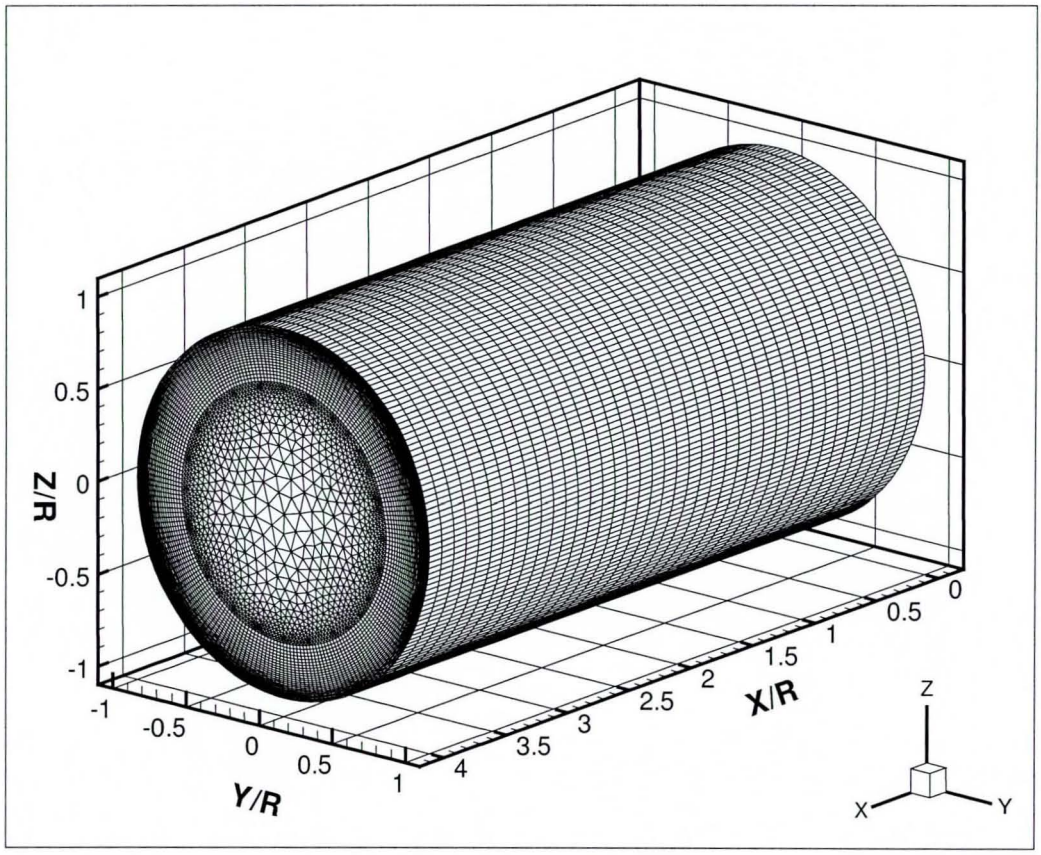


Figure 6.20: Computational mesh for the turbulent pipe flow simulation

6.3.1 Computational Mesh

A section of an infinitely long round pipe was modelled by a cylindrical computational domain of radius R length $4R$, which was discretised by a hybrid mesh consisting of a hexahedral mesh to discretise an annulus near the wall region to capture the eddies in the turbulent boundary layer and triangular prisms in the core region where the eddy structure is less important. It must be noted that having triangular prisms in the core region has alleviated the severe time step restriction typically found in curvilinear structured meshes due to the small cell size along the centre line [122]. The hybrid mesh was generated using Gambit. Following the description given by Nicoud and Ducros [84], the interface between the annular hexahedral region and the cylindrical triangular prism region was placed at r=0.7R. The hexahedral cells were arranged in a structured manner, such that the pipe wall is discretised into 40 uniform cells along the pipe axis and 200 uniform cells along the pipe circumference. From the cylinder wall to

the interface, 25 non uniform cells were generated. The first four cell from the wall had uniform distribution with a spacing of $0.0075R$ ($\Delta d^+ = 2.4$) whereas the others were stretched toward the pipe core. This radial coarsening was continued into the centreline. The resulting mesh consisted of 200k hexahedral and 100k prisms giving a total of 260k vertices. Figure 6.20 shows that the computational mesh is relatively coarse.

6.3.2 Boundary Condition and Computational Procedure

The adiabatic non-slip wall boundary was applied for the pipe wall. Periodic boundaries were applied at the inlet and outlet region. It was assumed that the external force to overcome the skin friction was solely provided by the stream-wise pressure gradient.

$$\rho f_x = \frac{\partial p}{\partial x} = \frac{\lambda}{2R}\frac{\rho u_b^2}{2} \tag{6.6}$$

whereas f_y and f_z were set to zero, and λ is approximated by Blasius' resistance law [135]

$$\lambda = \frac{0.3164}{Re_b^{1/4}} \tag{6.7}$$

as Re_b was well below 10^5, which is the upper limit of the Blasius law applicability. A similar strategy has also been applied in ref. [84].

The cut-off level for the dissipation switch ϵ was set at 0.05. For comparison purposes, the fixed 5% scaling of the Roe Matrix artificial dissipation prescribed by Bui [17] was also used. It must be noted that Nicoud and Ducros [84] employed a scalar dissipation so that separate scaling for each equation of the Navier-Stokes system was possible. In their case, zero for the momentum equations and unity, hence full dissipation, for the continuity and energy equations. Such a separate scaling is hardly advisable for the Roe scheme whose dissipation is closely coupled together. In all of the previously mentioned computations, an unlimited MUSCL flux reconstruction based on eq. 4.21 and 4.22 has been used to ensure that the Roe dissipation matrix does not render the spatial scheme first order accurate. A further reference computation was also performed to mimic a centred second order accurate calculation typically found in structured mesh LES by employing a fixed zero scaling factor without any reconstruction. This was particularly relevant to assess the near wall behaviour that was discretised by hexahedral cells.

Two SGS models described in chapter 2 were tested in this section, namely the WALE model and the Spalart-Allmaras based DES. Despite being the most popular SGS model for LES, the standard Smagorinsky model was not tested. The Van Driest

damping, that is needed by standard Smagorinsky model, requires a data structure to store a pairing between each internal node to the nearest node on the wall. In practical unstructured mesh geometries this can be an expensive and complex pre-processing step. This was felt to be a major drawback as compared to the WALE model. Following the suggestion that had been given by ref. [84], the WALE model was implemented with $C_w^2 = 0.25$. Because of the low Mach number, the compressibility correction in the SGS model was set to zero. DES computation was conducted with the commonly used $C_{DES} = 0.65$.

All of the computation were performed on 15 processors of an Intel® Xeon 2.4 GHz based PC cluster resulting in rapid turn-around in this coarse mesh. A white noise superimposed on a Poiseulle solution was implemented as the initial flow field. The flow field was non-dimensionalised by sound speed, giving a non-dimensional U_b to be identical with Mach number. The time step was set to $0.01R/a_\infty$ for the LES computations, which was comparable to ref. [84]. The DES computation required a smaller time step, i.e. $0.008R/a_\infty$ to ensure that the solution did not diverge. The statistics were not collected until statistically steady kinetic energy of the flow-field has been achieved. A further 100000 (120000 for DES) iterations were then performed to get the statistical data.

6.3.3 Results and Discussion

Computations performed in this section are summarised in table 6.2. The computations can be divided into three main groups. The first group have been performed with the Ω-switch to scale down the artificial dissipation of the Roe matrix. For this purpose the cut-off value ε is set at 0.05 to ensure that the numerical dissipation is small for the majority of the flow. The baseline computation for this group has been carried out with the WALE SGS model (case WALE-Φ). To isolate the upwind effect an embedded LES computation was performed by relying solely on the scaled down Roe dissipation (case MILES-Φ). Another calculation with the modified Ω-switch, where the vorticity term is magnified 5 times to make the switch more sensitive in detecting the turbulence structure, with the WALE SGS model (case WALE-Φs5) has also been performed.

The second group has been performed with a scaling of the Roe matrix fixed to 0.05. Here, two SGS models described in chapter 2 are compared, namely the WALE model (case WALE-05), and DES based on the Spalart-Allmaras turbulence model (case DES-SA-05). The single computation of the third group is denoted as CENTR2 is based on

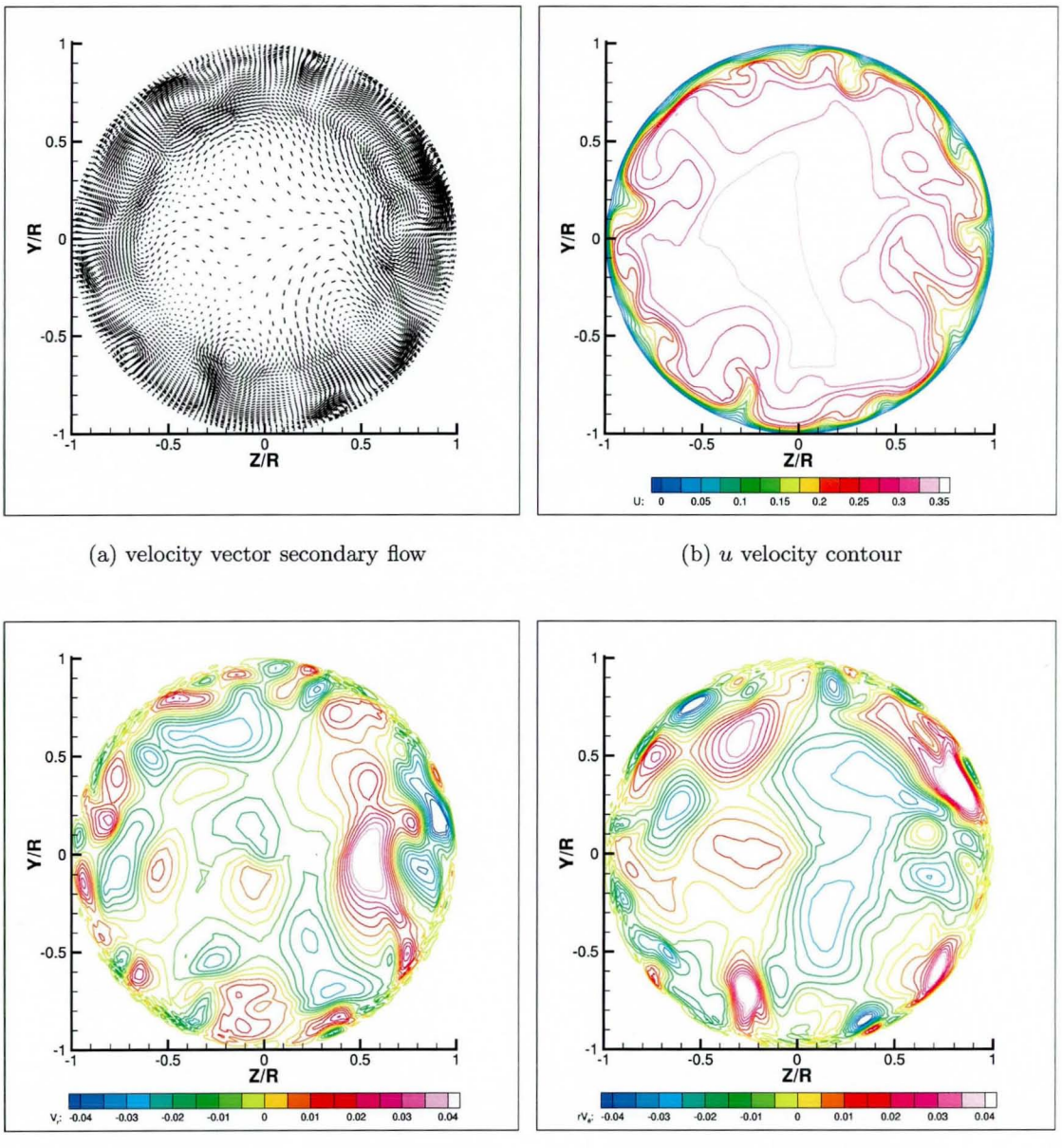
linear interpolation to the control volume face with the absence of numerical dissipation. This particular computation is perform only as a reference to the numerical behaviour to the fixed 5% numerical dissipation.

case	recons.	ϕ	SGS model	Re_τ
WALE-Φ	MUSCL	eq. 4.19; $\varepsilon = 0.05$	WALE	327
MILES-Φ	MUSCL	eq. 4.19; $\varepsilon = 0.05$	(embedded)	315
WALE-Φs5	MUSCL	eq. 4.20; $\varepsilon = 0.05$ $\Lambda = 5$	WALE	328
WALE-05	MUSCL	0.05	WALE	310
DES-SA-05	MUSCL	0.05	DES	321
CENTR2	–	0.0	–	N/A

Table 6.2: Pipe computations for nominal $Re_\tau \approx 320$ based on friction velocity and pipe Diameter [84]

There is no doubt that the numerical schemes described in chapter 4 are capable of sustaining the turbulence. Fig. 6.21 shows the instantaneous velocity field obtained from case WALE-Φ. Similar instantaneous turbulence in the velocity field are also observed in all other calculations. Furthermore, the familiar coherent structures near the wall with vortex stretching followed by ejection is observed to various degrees in all computations. Figure 6.3.3 shows instantaneous vortical structures (using Q criteria [55, 59]) from three selected cases: WALE-Φ, WALE-05 and DES-SA-05. The colouring indicates rotation about x-axis, with blue stands for negative rotation (clockwise) and yellow for positive. WALE-E05 shows more noise than the other two due to near wall oscillation that will be discussed later. DES-SA-05 shows unphysical structures that are longer and more widely spaced than has been reported previously by Bagget [5], probably due to the unsteady RANS that dominate the near wall behaviour.

Fig. 6.23 shows that the mass flow rate obtained from the first group (and DES-SA-05) of computations do not give the correct value. The plots are shown against wall distance $d = R - r$ presented in wall units $d^+ = (\rho_{wall} u_t d)/\mu$, where the skin friction velocity is $u_t = \sqrt{(\tau_{wall}/\rho_{wall})}$. For instance, rather than following the widely accepted empirical velocity distribution of $2.5\ln(d^+) + 5.5$, the velocity profile of case WALE-Φ actually follows $2.5\ln(d^+) + 8$ more closely. Other researchers, for instance Mossi and Sagaut [83], often express this phenomenon as lack of skin friction since the u^+ velocity profile can indeed be made to follow the empirical law when a higher skin friction velocity is used.



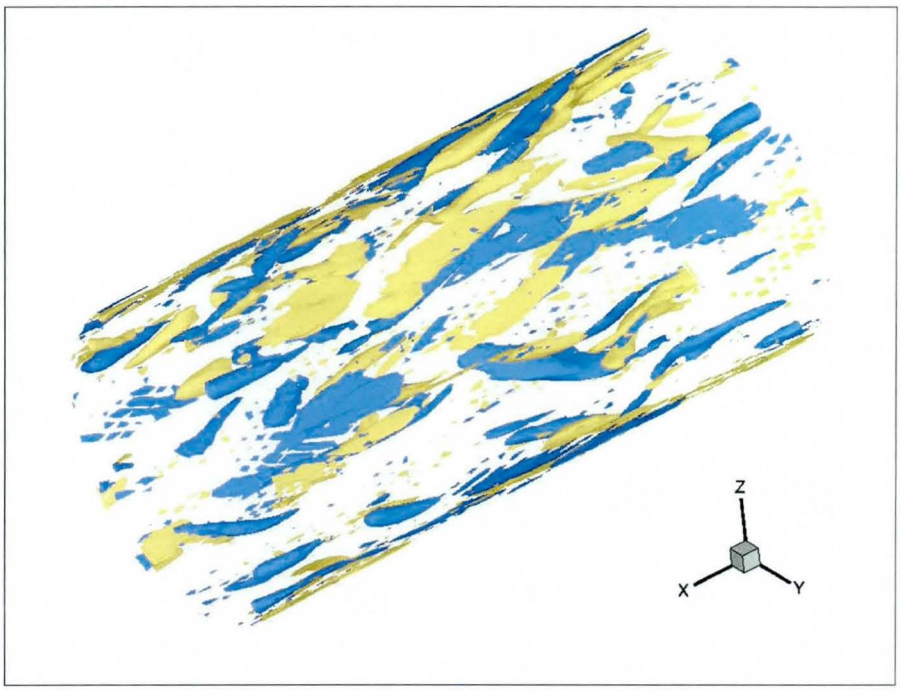
(a) velocity vector secondary flow

(b) u velocity contour

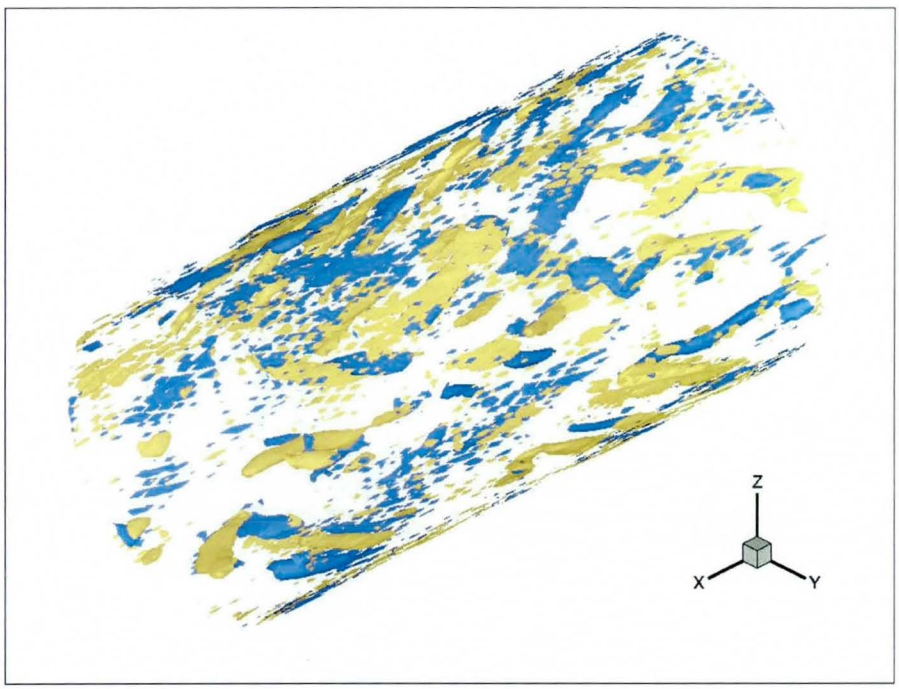
(c) u_r velocity contour

(d) ru_θ velocity contour

Figure 6.21: Instantaneous velocity using Ω-switch with ϵ=0.05

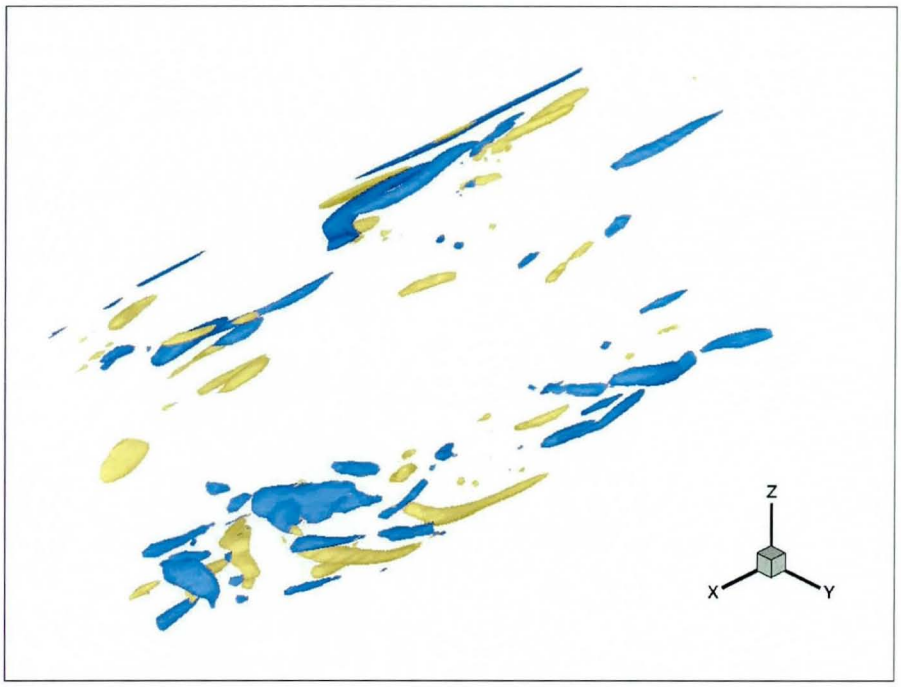


(a) WALE-Φ



(b) WALE-E05

Figure 6.22: Instantaneous vortical structures based on $\mathcal{Q} = 0.25$.



(c) DES-SA-05

Figure 6.22: Instantaneous vortical structures (Cont'd)

Fig. 6.24 shows the variation of the RMS turbulence velocity against wall distance d^+. As for the mean velocity profile, some discrepancies are also observed at the first group of computation (as well as DES-SA-05). It must be noted that the peak level value of Durst et al. [32] measurement lies on lower d^+ since it has been taken at lower Reynolds number.

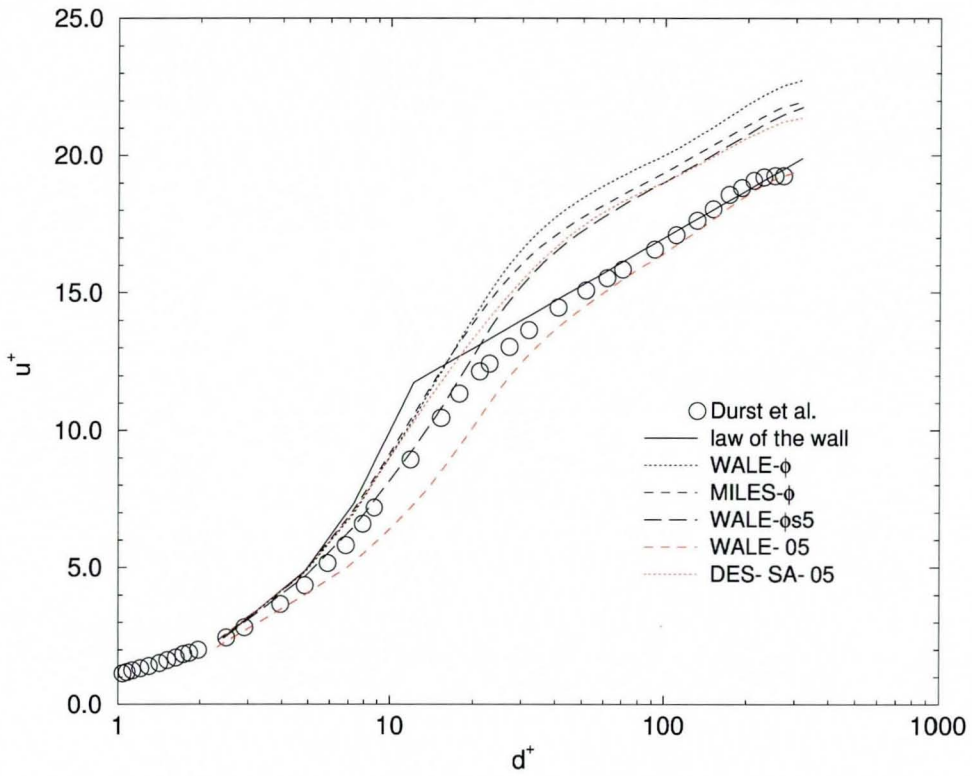
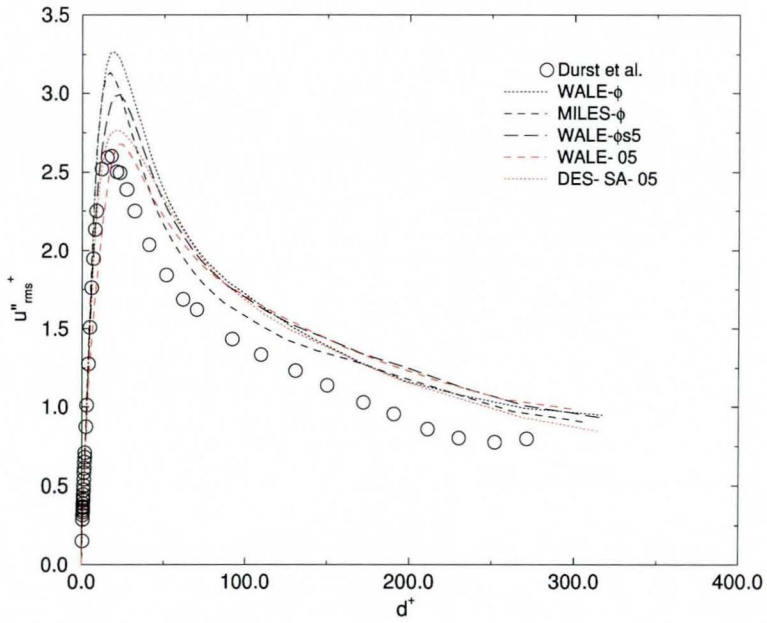
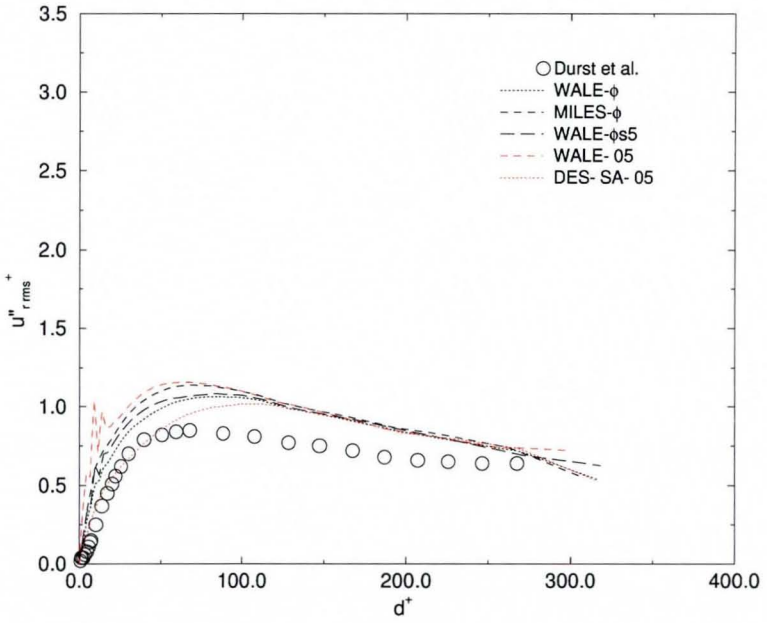


Figure 6.23: Mean flow velocity profile normalised by skin friction velocity

The switch actually scales down the artificial dissipation to 5% (the cut-off value ε) in most of the field apart from the very few spots near the wall where rapid changes of the secondary flow direction, i.e. the fluctuations, causes the divergence of velocity $\nabla \cdot \mathbf{u}$ to be the same order of magnitude, if not higher, than the vorticity ω. This in turn produces higher numerical dissipation that can be as high as 70%. Although those spots only affects very small region at any instant the effects is clearly very important as the fixed 5% scale (both with WALE and Smagorinsky models) actually predicts better mean velocity profiles. It is also easily seen that by biasing the switch toward vorticity, as in case WALE-Φs5, the upwinding effect can be reduced so that the solution is driven towards the fixed small scale. Although Nicoud and Ducros [84] did not report



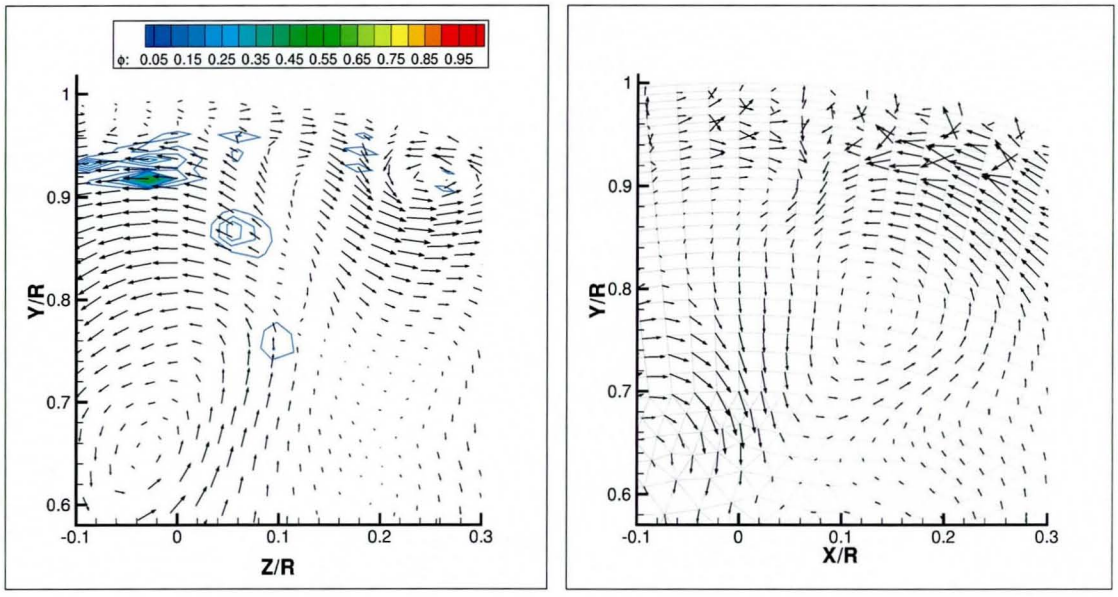
(a) stream-wise velocity component



(b) radial velocity component

Figure 6.24: Root-mean-square turbulence velocity profile

such problems, a series of LES computations on channel flow using upwind schemes conducted by Mossi and Sagaut [83] suggested that the present discrepancies are not an isolated incident. Due to their full scale upwinding (in some cases the flux limiter was also activated), their results were generally worse than the present work. Bui [17] reports similar result when comparing the full upwinding of the Roe scheme against that of scaled by fixed 5% value. Furthermore, similar problems are also observed even using centred structured schemes, including the fourth order scheme with symmetric TVD-based characteristic filter of Yee et al. [138].



(a) WALE-Φ

(b) WALE-05

Figure 6.25: Secondary flow near wall. a) includes contour of ϕ

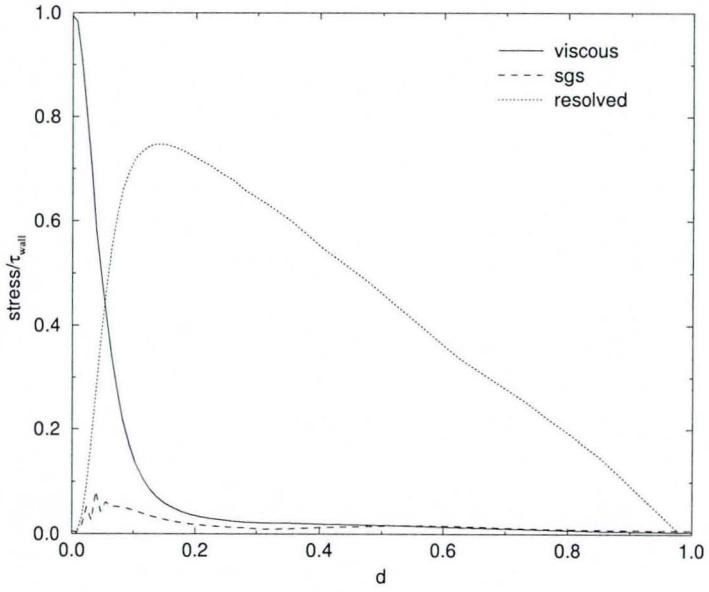
Having laid out the detrimental effect that upwinding may have on the statistics of an LES computation, closer inspection at the near wall velocity flow-field tells a rather different story. Two cases, namely WALE-05, exhibit unphysical oscillations due to odd-even decoupling typically found in centred scheme near the wall region. When the Ω-switch is activated these oscillations triggered high dissipation spots (before spreading more widely as in fig 6.25), thereby removing the unphysical behaviour. The same oscillation can also be removed by introducing a wall function such as the (unsteady) RANS mode of the Spalart-Allmaras turbulent model in DES, which effectively increases the near wall dissipation. This indicates that the viscous numerical scheme employed by the present method (see section 4.5) does not seems to be able to remove the odd-even

decoupling phenomena of the centred scheme without resorting to numerical dissipation.

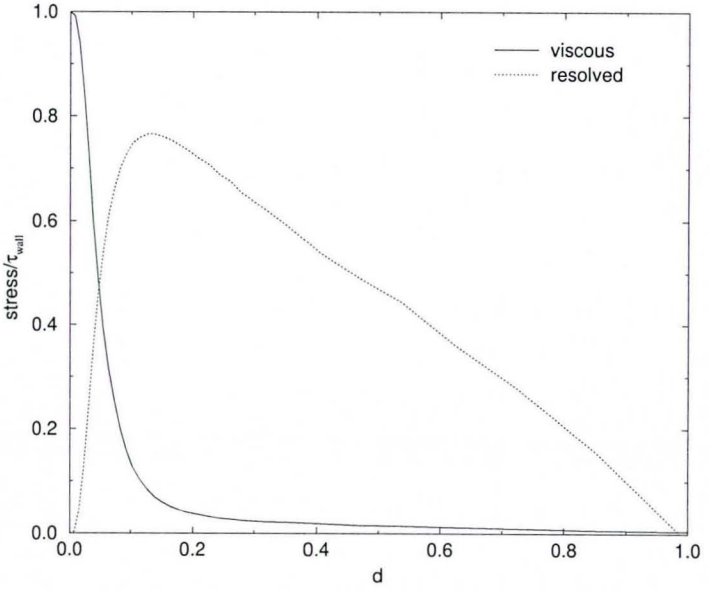
Kravchenko and Moin [63] showed that the two most widely used discretisations of convective term, namely the divergence and skew symmetric forms can strongly affect the robustness of the computer code. Although the divergence form by definition retained the mass conservation law, it was found that its application to a second order finite difference scheme was only stable on a staggered mesh while the skew symmetric form was stable on both co-located and staggered meshes. Morinishi et al. [80] argued that the reason behind this different behaviour lay in the fact that the skew symmetric discretisation conserves the kinetic energy as well as momentum. Later, Ducros et al. [29] showed that in the flux evaluation of a control volume surface in a finite volume code, a divergence form is obtained when it is taken as the average of the fluxes from the two sides of the face, while the skew symmetric form is obtained when the flux is computed from the average of the primitive variables. Unsurprisingly, the divergence form is obtained when eq. 4.17 is employed to evaluate the flux in centred scheme mode as carried out in case CENTR2. As a result CENTR2 also suffers from the oscillation. In short the oscillation found in WALE-05 is not caused by the reconstruction, but the divergence form of the discrete convective flux evaluation.

Generally speaking the discretised form of the convective scheme is rarely discussed in detail in compressible flow numerics such as those developed in refs. [57, 70, 97] since these oscillations are generally removed by the smoothing term in the numerical dissipation. In agreement with the present finding as presented by WALE-Φ, WALE-Φs5 and WALE-05, full Roe upwinding has never been reported to produce any boundary layer oscillation in a steady viscous Navier-Stokes solver. Bui [17], did mention in passing the stability issue but did not mention whether such oscillation has been observed or not in his computation. On the other end of the spectrum, Mossi and Sagaut [83] implemented a full Roe upwinding with min-mod flux limiter resulting in a very dissipative numerical scheme such that the flow was actually beginning to re-laminarise.

The consequence of the oscillation affects the behaviour of the SGS stress near the wall. From sections 2.3.1 and 2.3.2, it can be deduced that the SGS eddy viscosity should fluctuate when oscillation occurs in the flow. Fig. 6.26, demonstrates this unphysical behaviour, which in turn affects the shear stress distribution, which would otherwise show the correct behaviour. As a note, Bui [17] did not mention whether such oscillation had been observed, but stated the 5% numerical dissipation was employed to keep the solution robust. It is worth noting that the consequence of unphysical near wall

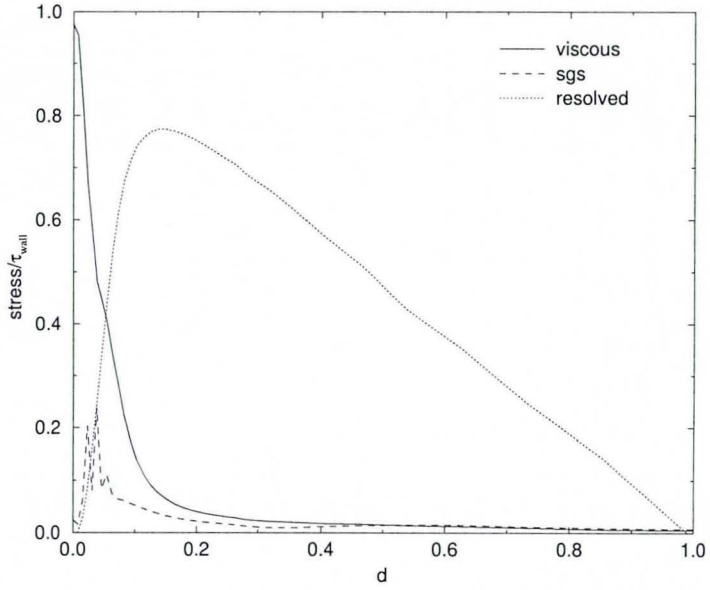


(a) WALE-Φ

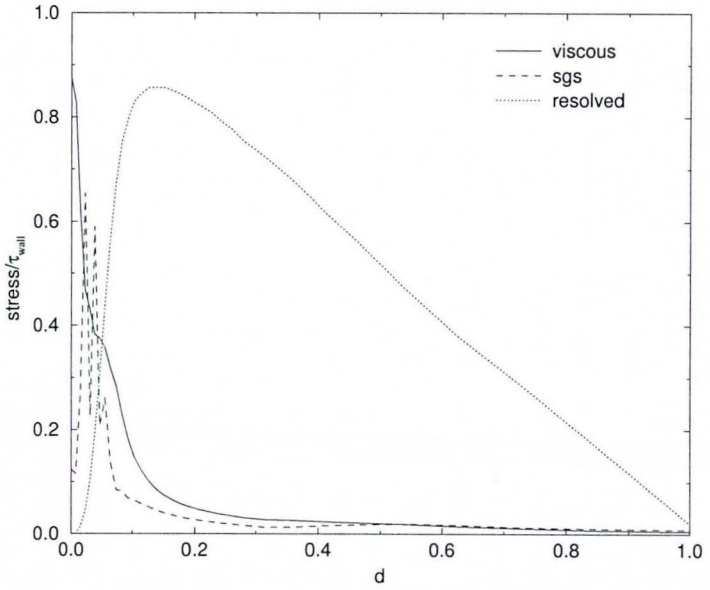


(b) MILES-Φ

Figure 6.26: Time averaged shear stresses τ_{xr}

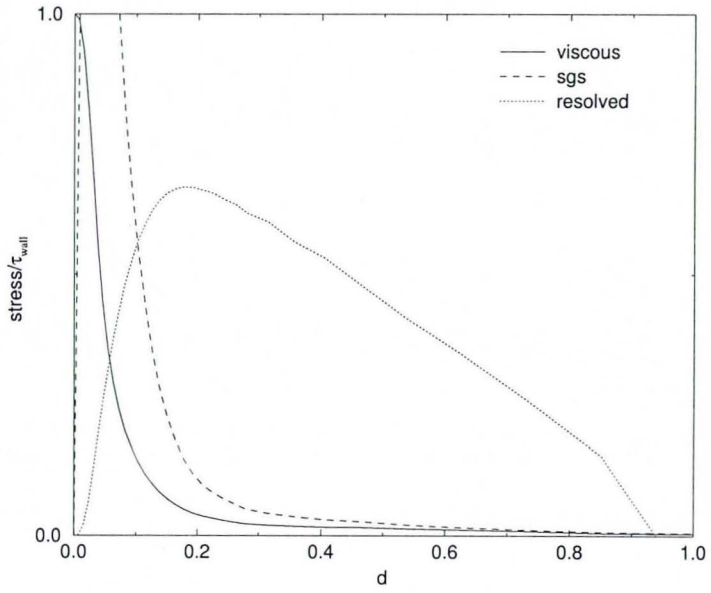


(c) WALE-Φs5



(d) WALE-05

Figure 6.26: Time averaged shear stresses (Cont'd)



(e) DES-SA-05

Figure 6.26: Time averaged shear stresses (Cont'd)

flow structures observed in fig. 6.22(c) that has been noted by Bagget [5] manifest as extremely high SGS stress near the wall before settling to a lower value in the core flow. Consequently, the peak resolved stress of the DES computation is significantly lower than the other LES computations.

Bui [17], shows that the dissipative upwind-biased schemes tend to produce higher RMS of stream-wise velocity fluctuations than the centred scheme. The picture is reversed for the other velocity components, suggesting that the artificial dissipation actually redistributes the turbulence kinetic energy. Therefore it can be said that the numerical dissipation is indeed behaving in a similar manner to an SGS model as claimed by the proponents of embedded SGS models. This view is reinforced by the flow behaviour when the code is operated in embedded SGS mode (case MILES-Φ). This embedded LES computation behaviour agrees with the analysis by Drikakis [28] and results from other wall bounded flow such as Fureby and Grinstein channel flow [40] and Urbin and Knight boundary layer [127]. The DES computations with fixed scaling of the Roe matrix, case DES-SA-05, supported this view even further. However the numerical discrepancies observed from these schemes suggest that the embedded SGS

model may not behave accurately in modelling the flow physics. Similar conclusion has been reached by Garnier et al. [43] from their study on homogeneous turbulence decay.

6.3.4 Closure

The predictions shown here have demonstrated that the present code is capable of computing a pipe flow type problem where sustainance of the turbulence is normally an issue for a code with built in numerical dissipation. The resulting flow field behaves in similar manner to other works that has been based on coupled solver on structured mesh. Qualitatively, the worm vortices near the wall and their stretching and ejection towards the core region are captured very well. The high turbulent kinetic region near the wall due to stream-wise velocity fluctuations are also predicted without any serious problem.

Quantitatively, the numerical dissipation has compromised the numerical accuracy of the mean flow. Removing the dissipation completely proved to be difficult as it promotes the odd-even decoupling near the wall. Despite this, the DES computation suggests that by appropriate use of wall functions these difficulties could be reduced.

6.4 Remarks on unsteady flow validation

A distributed memory parallel code with a mesh-transparent algorithm for LES of compressible flow has been presented. As the present work is aimed at practical industrial applications, a second order accurate numerical scheme based upon a Roe scheme [97] has been modified in order to minimise the upwinding outside flow discontinuity region. Qualitative study of the isolated numerical scheme using a spatially developing mixing layer indicates that the present method is comparable to the higher order numerical scheme with low dissipation property of Yee et al. [138] originally designed for LES/DNS computation using a structured mesh. Despite having similar resolution for the quadrilateral cells at the shear layer region, the triangles for the smooth flow region allows the hybrid mesh to employ significantly smaller numbers of nodes than that of a structured computation. LES computation for a compressible round jet has also been conducted. Once again the triangular cell for the far field has allowed smaller numbers of nodes than a structured mesh with similar resolution for the jet region. Comparison with Uzun et al. [128] who utilised a structured mesh with Lele's fourth order compact

scheme shows similar mean flow behaviour. These have given confidence in the present method for free turbulent flows. The present code still suffers from the small inherent dissipation that leads to quantitative discrepancy for wall bounded problems, although the qualitative flow physics has been captured well. Consequently, a more sophisticated scheme to control the level of upwinding in near wall region is required. Future research will be directed to tackle this problem, especially for hybrid unstructured mesh application.

Chapter 7

Unsteady impingement of a supersonic round jet on a flat plate

It has been shown in the previous chapter that the computational procedure developed in this thesis is capable of capturing the physics of turbulence flows. The limitations that have been highlighted are typical of a LES code based on a compressible flow algorithm. With these in mind, the present chapter deals with the prediction of a supersonic jet impingement problem. Such a problem has many engineering applications. As illustrated in Chapter 1, the new generation of STOVL aircraft operating in ground effect is a good example as the thrust requirement leads to highly under-expanded (supersonic) jet [65]. In industrial processing, the manufacturing of a quenched glass panel, typically used for vehicle wind screens, normally employ supersonic jet of cold air impinging on the heated glass to provide sudden cooling [100]. In both cases the noise generated via a feed-back loop mechanism that is originally proposed by Powell [92] has become a major concern.

A supersonic jet decelerates through a normal stand-off shock before impingement. Under certain conditions of nozzle pressure ratio (jet total pressure to ambient static pressure) and impingement height a very intense discrete tone can be produced [93] which may be 20 dB above the broadband noise. Henderson [52] observed that oscillations of the stand-off shock were connected to the production of tones. Recently Henderson et al. [51] carried out a comprehensive experimental study using phase-locked

shadowgraph and phase averaged Digital Particle Image Velocimetry to observe the unsteady behaviour and postulated how this relates to noise production. Whilst experimental study of the unsteady jet impingement problem is widely available, numerical study has mainly centred on unsteady inviscid flows [100]. Hence, the LES computation presented in this chapter is a preliminary work in nature.

The present study is based on the PIV measurements of Henderson et al. [51]. The nozzle pressure ratio is 4.0 and the impingement height is $2.02D$ in the experiment. They used a 25.4mm exit diameter nozzle giving a jet Reynolds number of approximately 6.0×10^5. In the experiment the tonal noise frequency was found around 10kHz. This corresponds to a Strouhal number of approximately 0.7 when the frequency was non-dimensionalised by the jet exit diameter and the speed of sound at the far field.

To ensure that good spatial resolution can be achieved achieved for the present LES computation using reasonable computational resources, the Reynolds number is lowered by one order of magnitude. Clearly, this would lead to some inaccuracies and a finer grid would allow higher Reynolds numbers to be studied.

7.1 Computational Domain and Mesh

A radially symmetric domain about x-Cartesian axis was created for the present computation. Figure 7.1 shows the x-y plane providing a cross sectional view of the computational domain(left) and the close up of the computational mesh around the nozzle and jet region on the x-y plane (right). For convenience the unit length has been non-dimensionalised by the nozzle exit diameter.

A generic conical nozzle with a 30° subtended angle was used for the present computation to model the external nozzle geometry used by ref [51]. In the absence of clear definition of the inner nozzle geometry, the present computation employed nurbs curvature for the inner profile to represent the convergent nozzle. The impingement plate was also simplified. Rather than using a rectangular plate of the experiment, the present computation assumed an infinitely large plate so that it could be extended all the way to the outer boundary of the computational domain.

A hybrid mesh, that was generated using Gambit, was used to discretise the flow domain. As shown at fig. 7.1, structured like hexahedral cells were used to resolve the annular region of the jet shear layer. triangular prisms were used for the jet core as

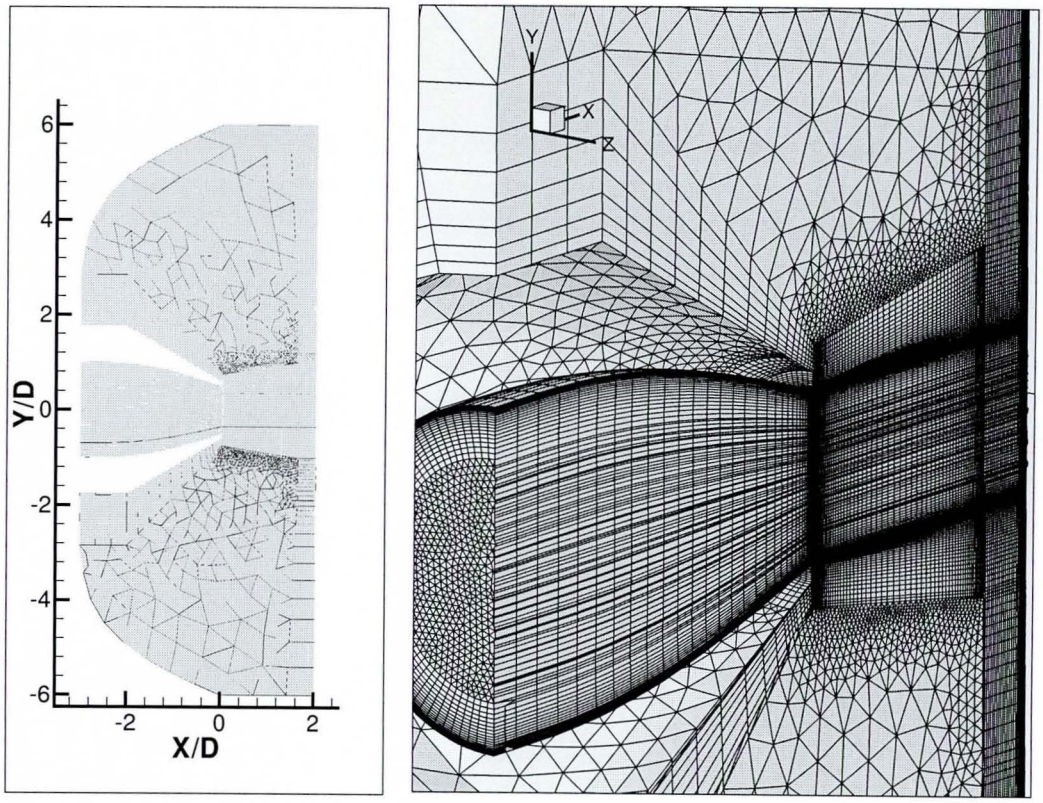


Figure 7.1: Computational domain and mesh of supersonic impinging jet

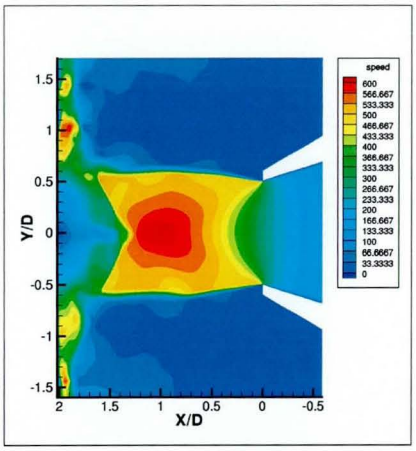
well as the flat plate boundary layer. The axial distance between the nozzle lip and the flat plate was resolved by 86 vertices, 30 of them were used to resolve the flat plate boundary layer. The rest of the domain was discretised using tetrahedral with rather aggressive degree of coarsening. The resulting mesh consist of 820k nodes, most of them were used to resolve the main and wall jets.

7.2 Boundary Condition and Computational Procedure

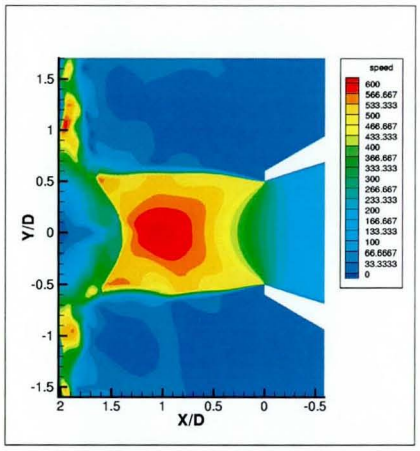
A steady state inlet boundary condition based on Riemann characteristics as described in section 4.7 was used for this preliminary study. Unlike LES of shock-turbulence interaction in ref. [29] that seeded the upstream flow with turbulence, the present work relied on the ability of the numerical scheme to generate the disturbance via instantaneous velocity strain and the feed back loop mechanism. Inlet boundary was also applied for the concave part of the outer domain whilst outlet boundary was applied for the cylindrical part. Obviously, the nozzle and flat plate surfaces are treated with non-slip wall boundary condition.

Unlike LES computations that had been undertaken in the previous chapter, the presence of strong shock waves required the limiter to be active in this case. Here, Venkatakrishnan limiter (see chapter 4) with $K = 0.2$ was used to ensure monotonicity across the shock. As a result the computational time for each time step is nearly doubled.

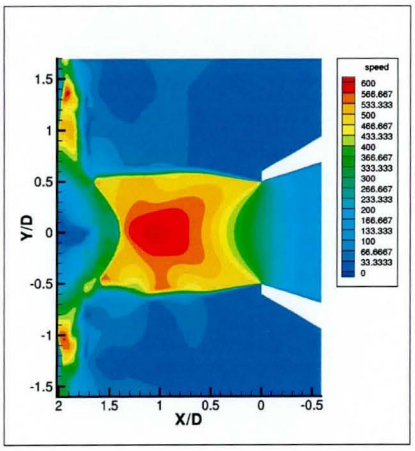
In consistent with the non dimensional mesh, other flow parameter were non-dimensionalised by the ambient density and sound speed as reference density and velocity. It must be noted that neither the density nor the temperature of the experiment had been reported. As a result, it would be impossible to get an exact match for the velocity magnitude that has been observed from the PIV since the choice of these reference values could not be made to In non dimensional time units, a small time step value of 1.2×10^{-4} has been applied. An initial run of 100,000 time steps (around three plate shock oscillations from a converged RANS solution have been performed. A further 244000 time steps were then carried out to collect approximately 32 time units worth of data which was sampled every 20 time steps.



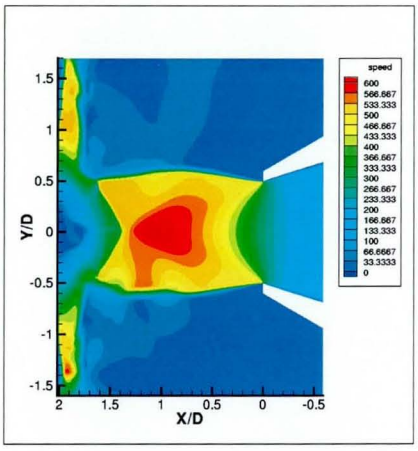
(a) $t = t_{start}$



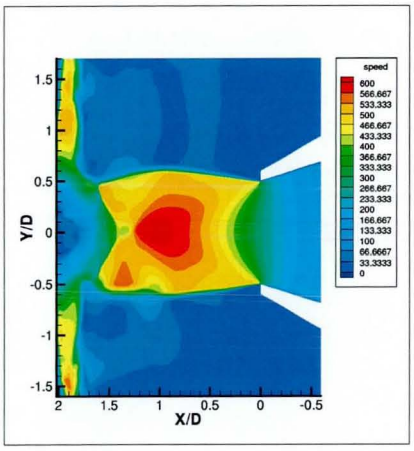
(b) $t = t_{start} + 0.24$



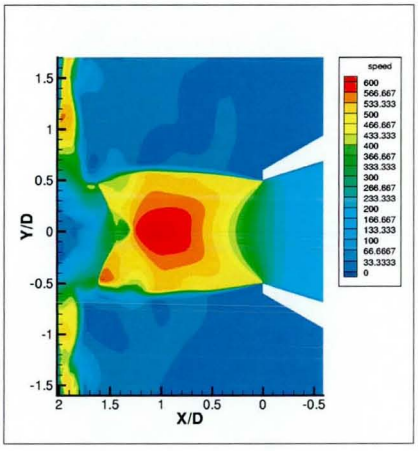
(c) $t = t_{start} + 0.48$



(d) $t = t_{start} + 0.72$

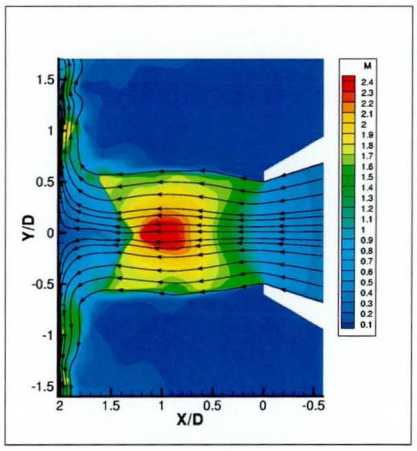


(e) $t = t_{start} + 0.96$

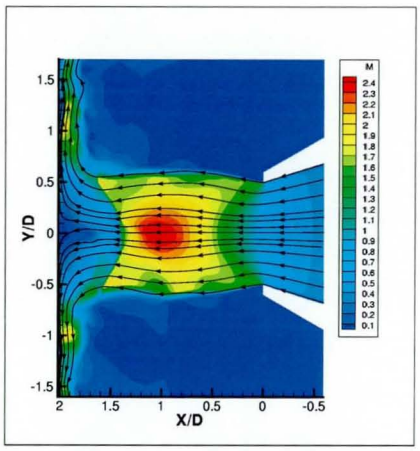


(f) $t = t_{start} + 1.20$

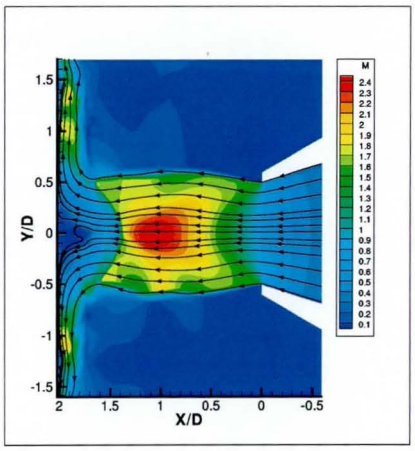
Figure 7.2: Velocity magnitude contour fluctuation



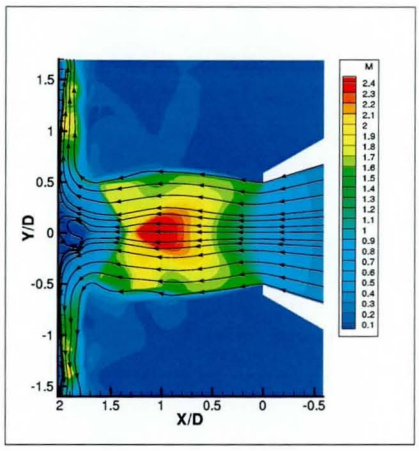
(a) $t = t_{start}$



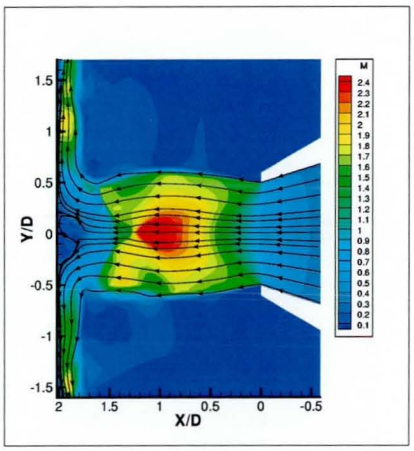
(b) $t = t_{start} + 0.24$



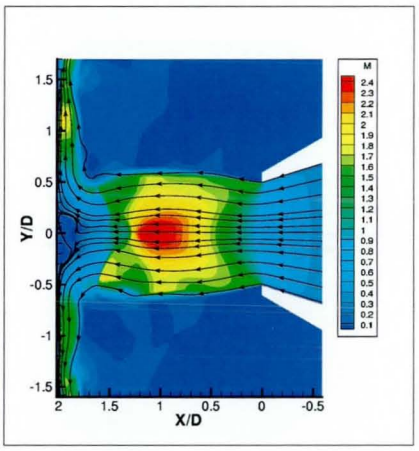
(c) $t = t_{start} + 0.48$



(d) $t = t_{start} + 0.72$

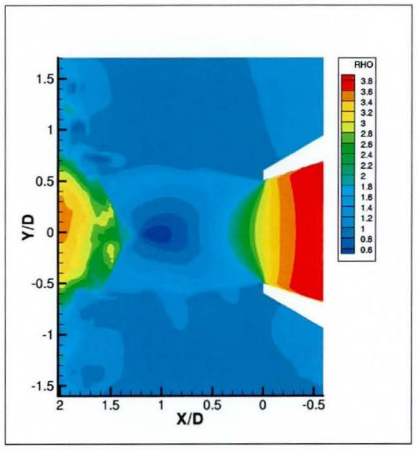


(e) $t = t_{start} + 0.96$

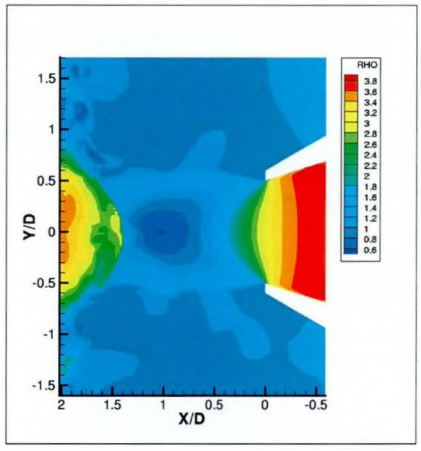


(f) $t = t_{start} + 1.20$

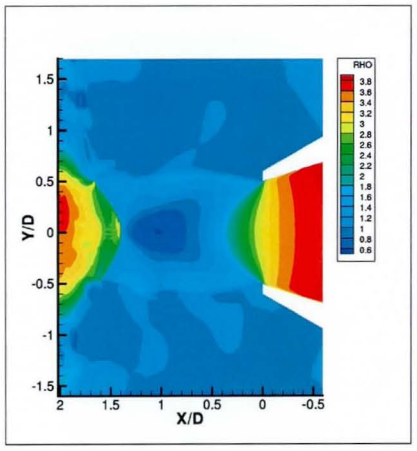
Figure 7.3: Mach number contour fluctuation



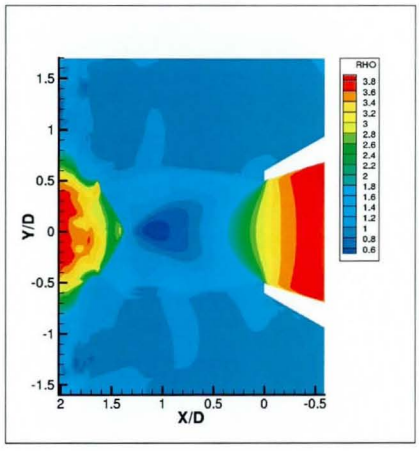
(a) $t = t_{start}$



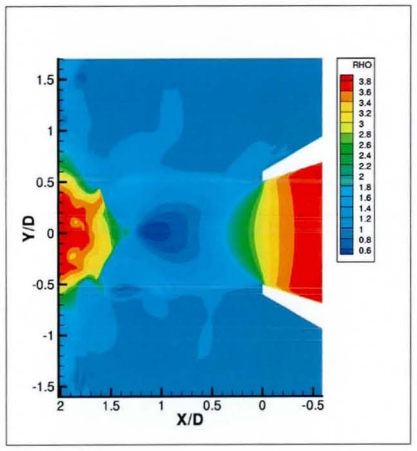
(b) $t = t_{start} + 0.24$



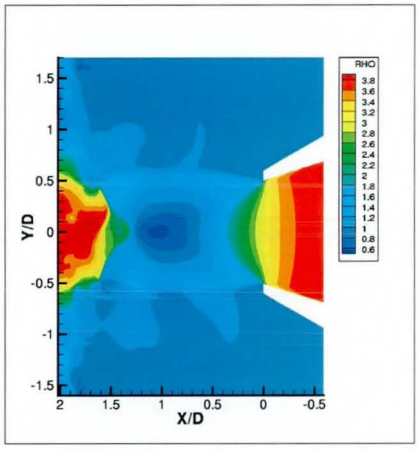
(c) $t = t_{start} + 0.48$



(d) $t = t_{start} + 0.72$

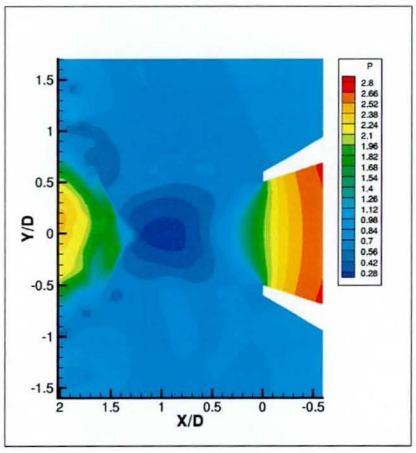


(e) $t = t_{start} + 0.96$

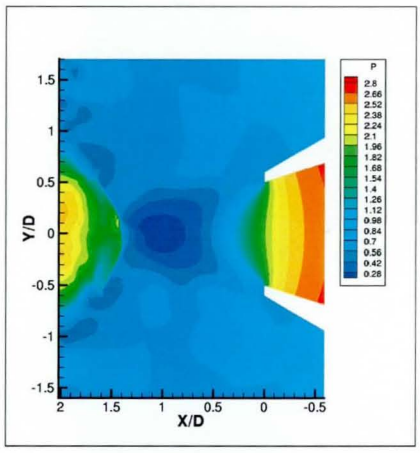


(f) $t = t_{start} + 1.20$

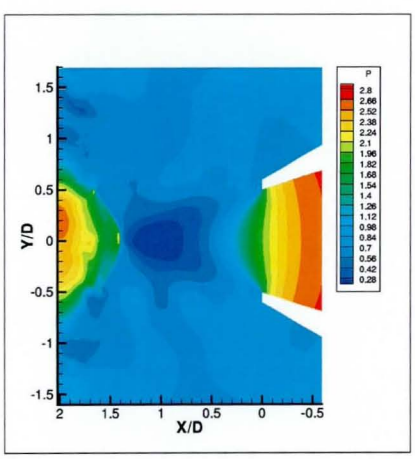
Figure 7.4: Density contour fluctuation



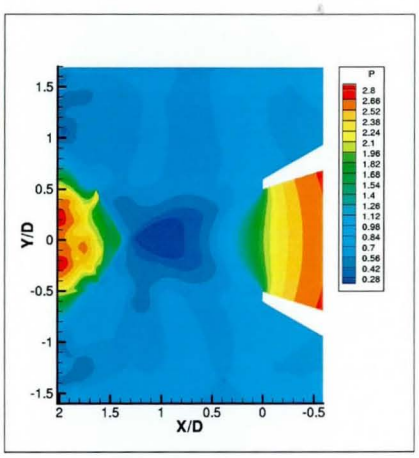
(a) $t = t_{start}$



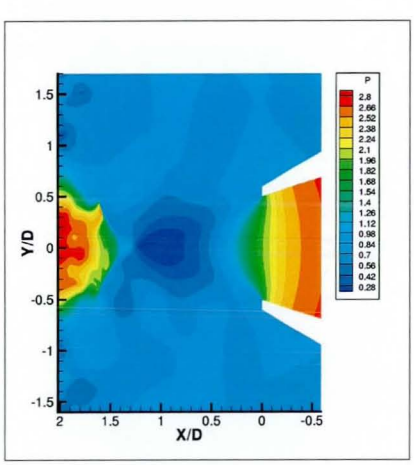
(b) $t = t_{start} + 0.24$



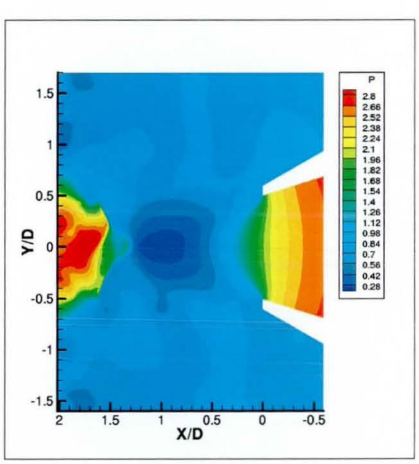
(c) $t = t_{start} + 0.48$



(d) $t = t_{start} + 0.72$



(e) $t = t_{start} + 0.96$



(f) $t = t_{start} + 1.20$

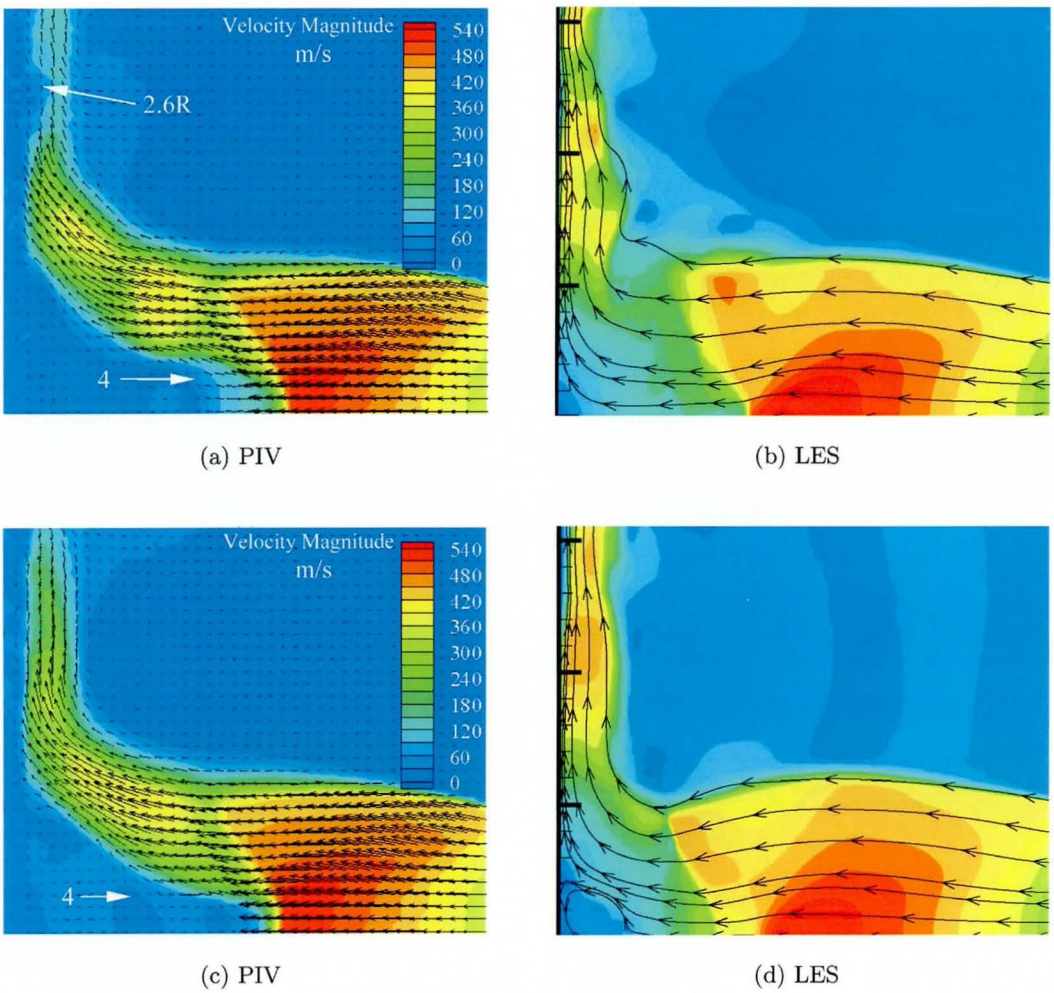
Figure 7.5: Pressure contour fluctuation

7.3 Results and Discussion

Figure 7.2– 7.5 are a series of still picture frames describing the unsteady turbulent flow in the highly under-expanded jet. Although 'streamlines' have been added to the Mach number plots, these are not valid in a time varying flow, they have merely been included to help highlight the recirculation bubble in the impingement region. Since the Mach number is highest on the jet centreline and the stand-off shock is normal, the shock total pressure loss is greatest on the centreline. Consequently, the static pressure observed at the impingement point is lower than in an annular region on the plate around impingement. This creates a recirculation zone in which the flow is inboard towards the impingement point. This is most easily observed by the 'streamlines' in fig. 7.3. Animation of the time history of these contour plots shows that the recirculation bubble is unstable and grows and collapses in time. Henderson et al. [51] observed a contact surface between the recirculation zone and the flow downstream of the normal shock, the flow is then deflected around the recirculation zone and towards the shear layer. The stand-off shock and contact surface were seen to be moving with periodic collapses of the recirculation zone. The motion distorts the curved jet shear layer and large changes in velocity in the wall jet were observed at 2.6 jet radii from the centreline and this was postulated as a major source of noise. Similar behaviour is observed in the LES predictions with a peak velocity bubble occurring in the wall jet at 2 radii from the centre line which then convects outwards. Furthermore, the time history pressure contour plot (fig. 7.5) clearly demonstrates the perturbation of the jet near nozzle exit due to the feed back mechanism.

An attempt has been made to compare directly the LES instantaneous velocity field with the PIV data in figure 7.6. It is not possible to directly match the two sets of data for a given time in the periodic flow and so two arbitrary times have been chosen for the PIV and LES contours. Similarly, the colourmap used to present the PIV contours differs somewhat from the LES plots. However, reasonable agreement is still observed. The experiments tend to show the stand-off shock being at a slightly greater distance from the plate and the recirculation bubble to be larger (in the experiment '4' indicates the observed penetration of the recirculation bubble). Indeed in figure 7.6(b) the recirculation bubble is absent. The LES predictions indicate higher peak velocities in the beginning of the wall jet and these are occurring in the region at 2.6 jet radii indicated by Henderson et al.

An instantaneous plot of the vortical structures using the Q-criteria [55] is shown



(a) PIV

(b) LES

(c) PIV

(d) LES

Figure 7.6: Supersonic impinging jet comparison with PIV measurements [51]

in fig. 7.7. The characteristic helical nature of eddies in the jet shear layer and wall jet is observed. It can also be seen that the recirculation bubble that has been reported in the PIV study [51] consists of a number of separate dynamic structures that has been noted previously rather than a single annular structure. This is in agreement with the flow pattern that has been observed by Donaldson and Snedeker [27].

Figure 7.7: Instantaneous vortex structures around impingement point using iso-surface of Q

Time histories of pressure were collected at various monitoring points that are shown in fig. 7.8. Four numerical probes were positioned in the jet. Pt1 and Pt2 were placed on the wall diametrically opposite to each other at 0.7 jet radius from the centre line. Pt3 was located on the wall at 2.6 jet radius from centre line. Pt4 was placed in the jet shear layer upstream to the impingement shock. Far field sound pressure level can not be extracted from the present computation since neither an acoustic code has been available nor that direct computation is possible as the mesh is not fine enough to propagate the pressure wave to a distant that can be considered far enough from the impingement point. Hence only the pressure fluctuation in the near field has been recorded. Pt5 and Pt6 were located close to the main and wall jet, respectively, whereas Pt 9 was

situated close to the nozzle exit. It must be noted that these probes were located at the tetrahedral region of the computational mesh.

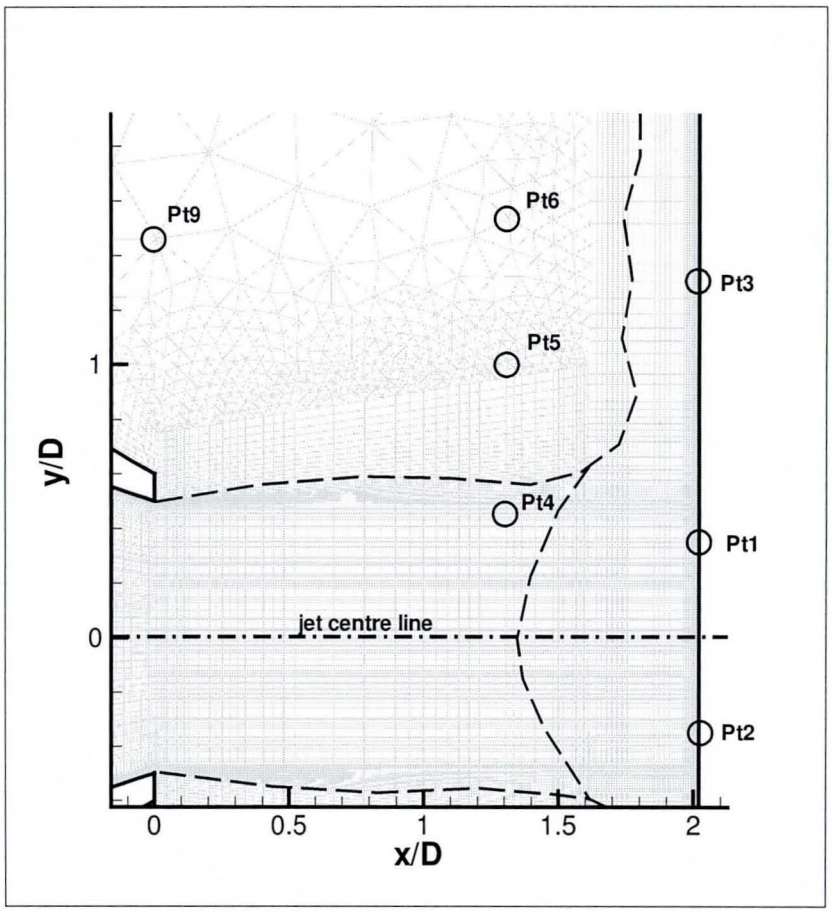


Figure 7.8: Numerical pressure probes position

Pressure fluctuations around the impingement point are obtained from Pt1 and Pt2. Figure 7.9 shows the correlation of the fluctuation measured by the two probes. The corelation exhibits axisymmetric mode of an impinging jet simulation (see fig. 7.10), which suggests that the qualitative behaviour of the impingement agrees with the widely accepted flow physics [65, 100].

Figure 7.11– 7.16 show the time history of the non-dimensional pressure fluctuations and the corresponding power spectra that have been obtained by all of the probes. The power spectra $|p'|$ are computed as

$$|p'| = \frac{Pz \cdot conj(Pz)}{M}$$

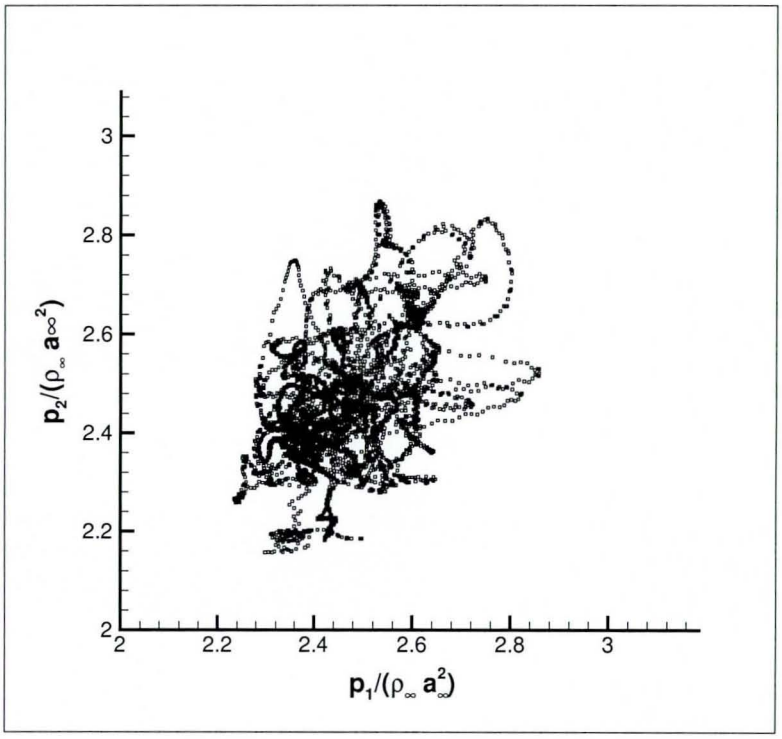


Figure 7.9: Pressure correlation on impingement plate (Pt1 and Pt2)

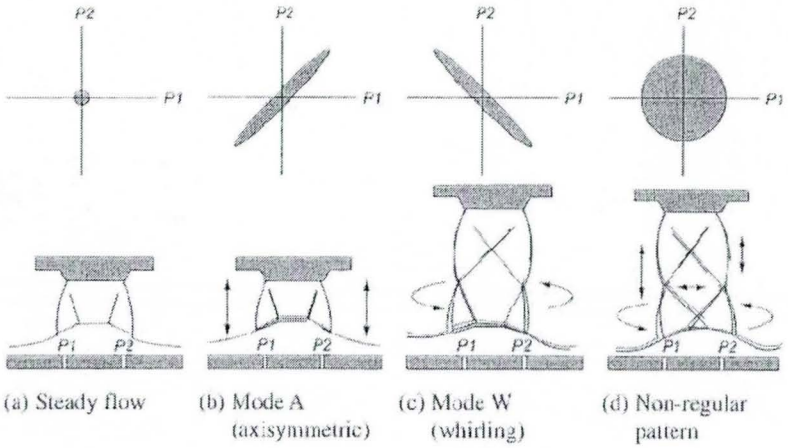


Figure 7.10: Oscillation modes of impinging jet [100]

Here, Pz is a complex number array consisting of M number of sample data which are obtained from the following fast Fourier Transform using MATLAB v5.2:

$$Pz = fft(p_d)$$

where p_d is the non-dimensional pressure fluctuation. It must be noted that the different cell size for each numerical probe means that the cut-off number of the numerical resolution are different.

Four numerical probes, namely Pt1, Pt2, Pt3 and Pt4, are used to investigate the pressure fluctuation inside the jet. The power spectra of the pressure oscillation of Pt1 and Pt2 shows a peak at a Strouhal number of approximately 0.5, which correspond to the shock oscillation. Inspection of all other probes data also shows pressure oscillation at this frequency. Another peak at a Strouhal number of approximately 1.1 has also been observed very strongly at Pt3 and Pt4, which corresponds to the formation of the ring vortices (See fig. 7.7). Since the shock oscillation occurs downstream of Pt4, which is supersonic, the oscillation at St=0.5 indicates the disturbance of the main jet through Powell's feedback-loop mechanism [92] that is illustrated by the contour plots (fig. 7.5).

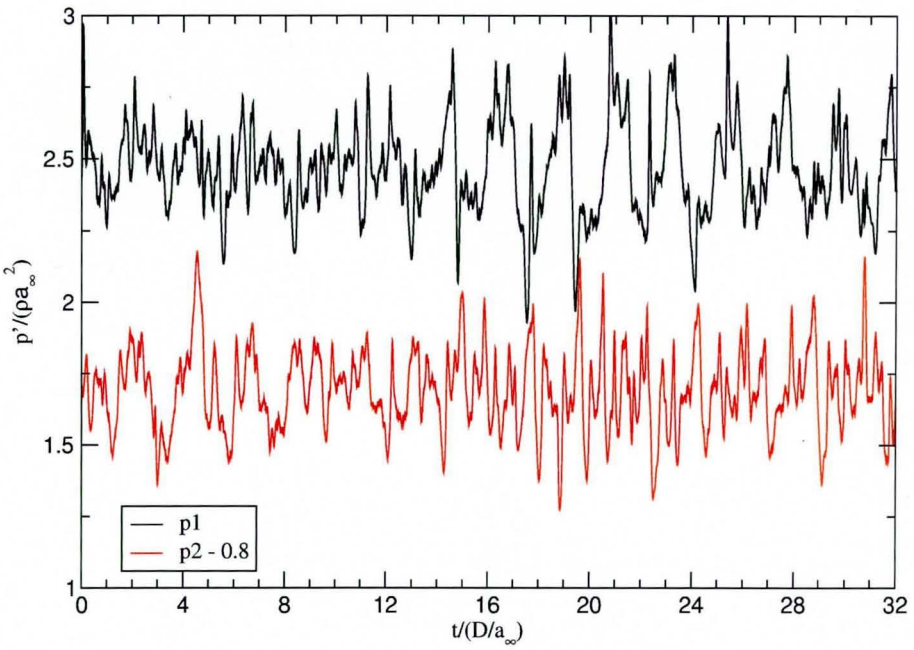
Observation of the pressure contour development suggests that only low frequency harmonics, thus the tonal sound, would have been distinctively propagated to the far field whereas the higher noise frequency would simply become broadband noise. This assessment is supported by the fact that the strength of the pressure fluctuation with a Strouhal number of 1.1 has decayed significantly in probes Pt6 and Pt9. An inspection of the time history fluctuation (See fig. 7.11(a) and the subsequent figures) suggests that the jet rotates slowly about its axis at approximately St=0.2. This confirm Sakakibara and Iwamoto observation of their three dimensional computation [100].

The power spectra suggest that the frequency of the present shock oscillation is lower than the tonal frequency observed by Henderson et al. [51], which was 0.7 as opposed to 0.5. Although a similar discrepancy between experimental measurement and unsteady numerical simulation has also been reported by Sakakibara and Iwamoto [100] in their unsteady inviscid computation, further research to investigate the LES behaviour is still required. It has been noted earlier that the LES prediction of the recirculation bubble around the impingement point is smaller than the experiment. This indicates that the bubble (and the shock that is located upstream) is not as energetic as in the experiment. As such a phenomena normally suggest that the computation is too diffusive, there are two possible reason for this. Firstly, the lower Reynolds number used for the LES may allows molecular diffusion to play more significant role than that in experiment.

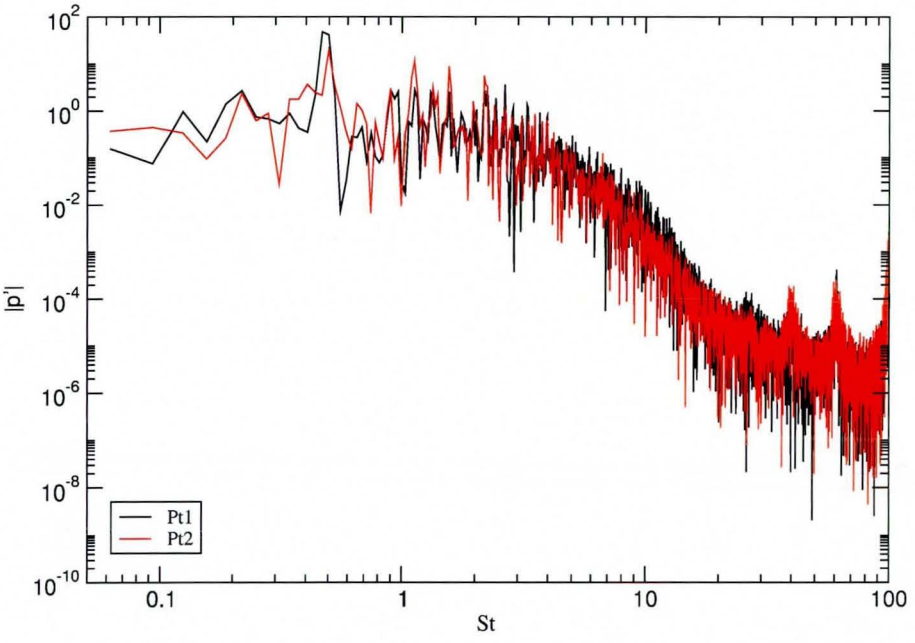
Secondly, the limiter that has been used to ensure monotonicity across the shock may have produced too much numerical diffusion.

7.4 Closure

A preliminary study of a highly under expanded jet with promising results has been presented. This particular test case is of special interest since there is no known LES computation that has been published in the literature. Qualitative analysis of the time dependent fluctuation in a highly under expanded supersonic jet has established the capability of the present LES methodology in capturing the important flow physics.

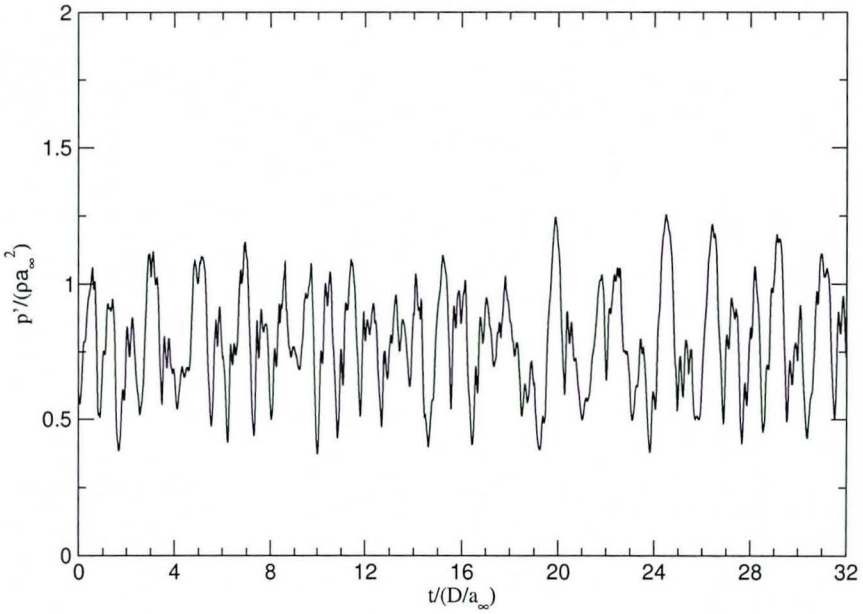


(a) time history

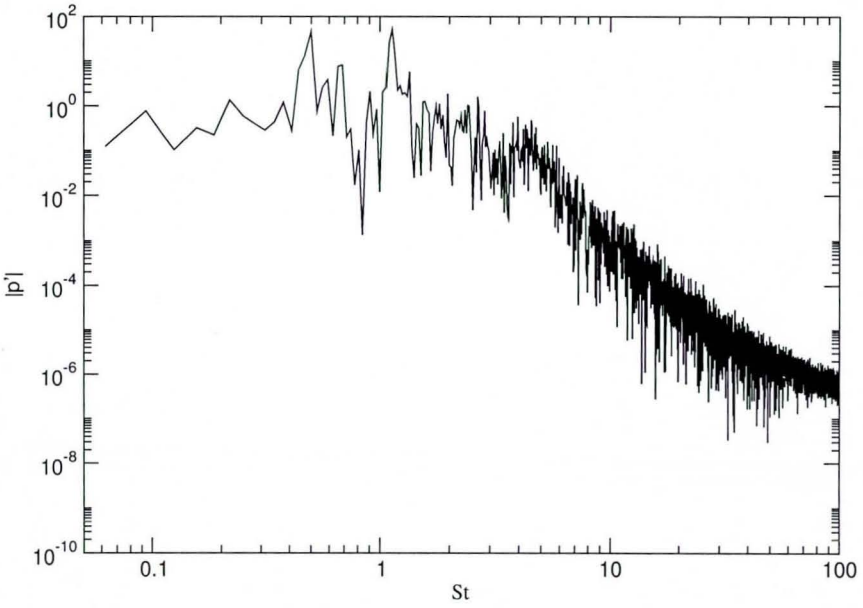


(b) power spectrum

Figure 7.11: Pressure fluctuation at Pt1 and Pt2

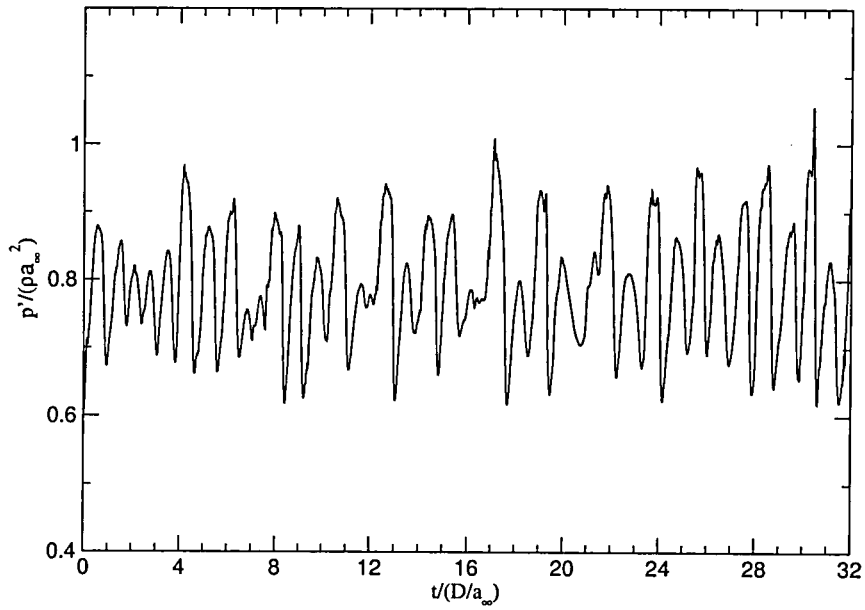


(a) time history

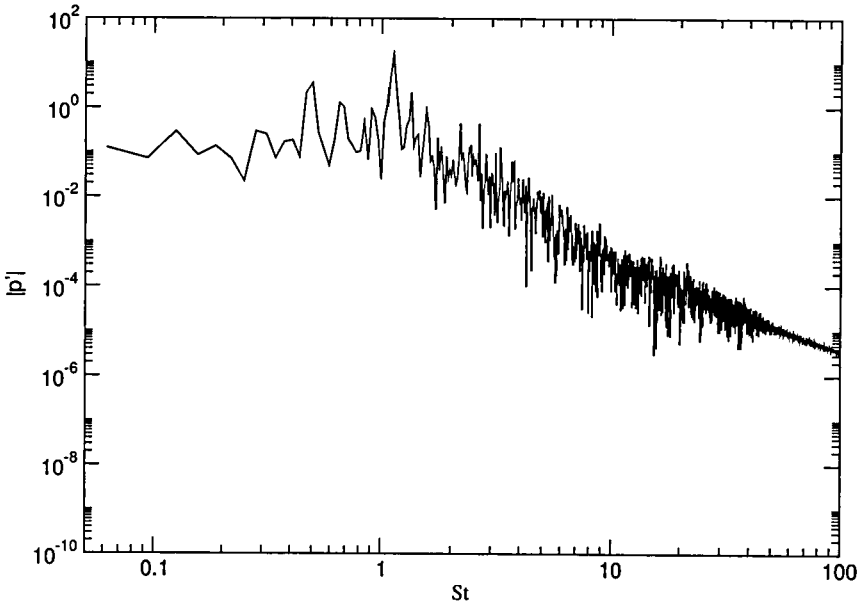


(b) power spectrum

Figure 7.12: Pressure fluctuation at Pt3

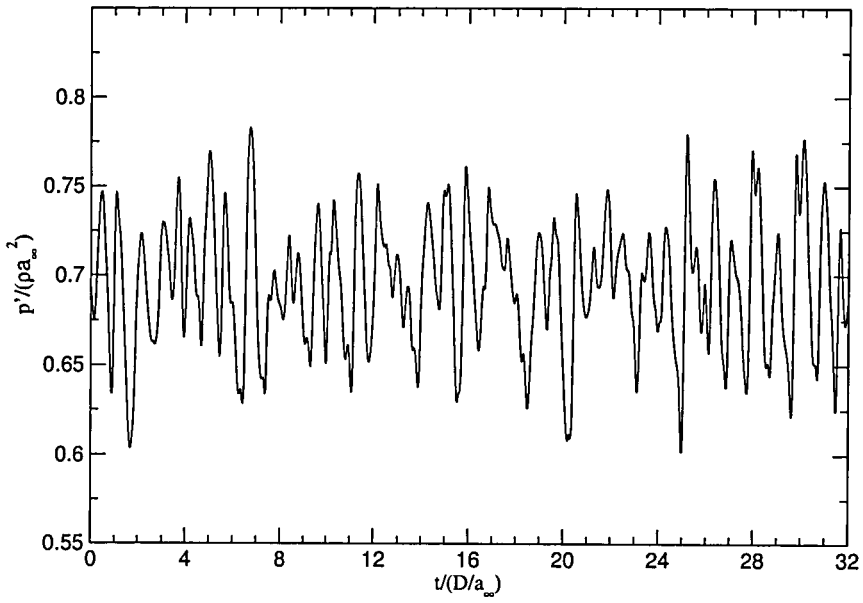


(a) time history

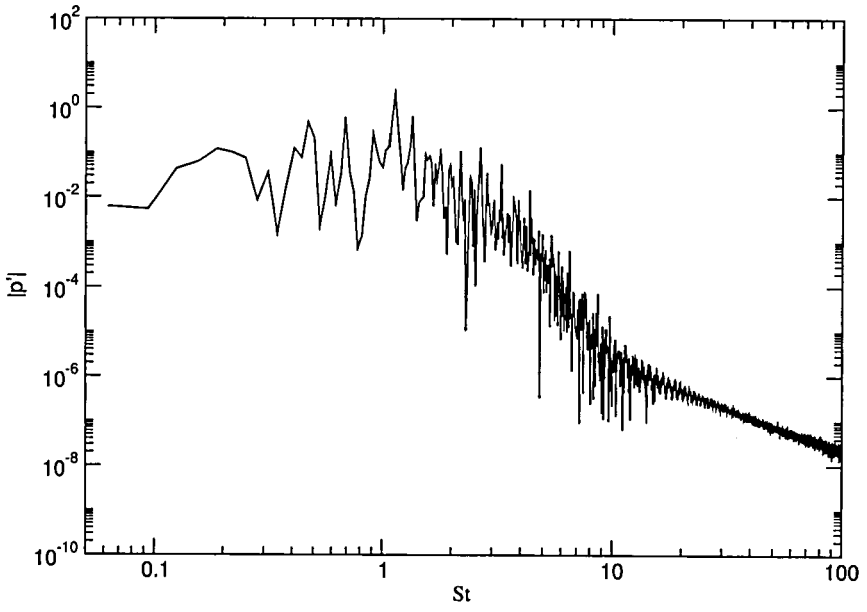


(b) power spectrum

Figure 7.13: Pressure fluctuation at Pt4

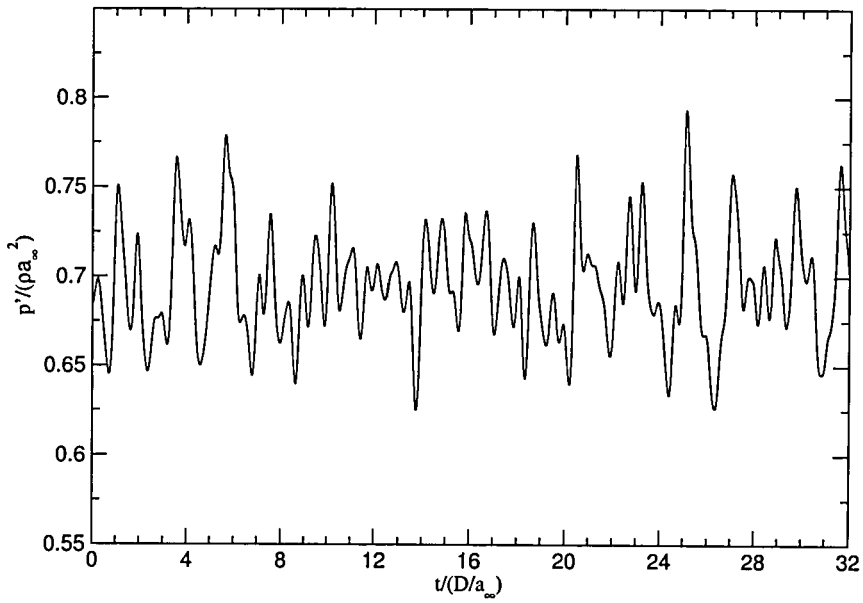


(a) time history

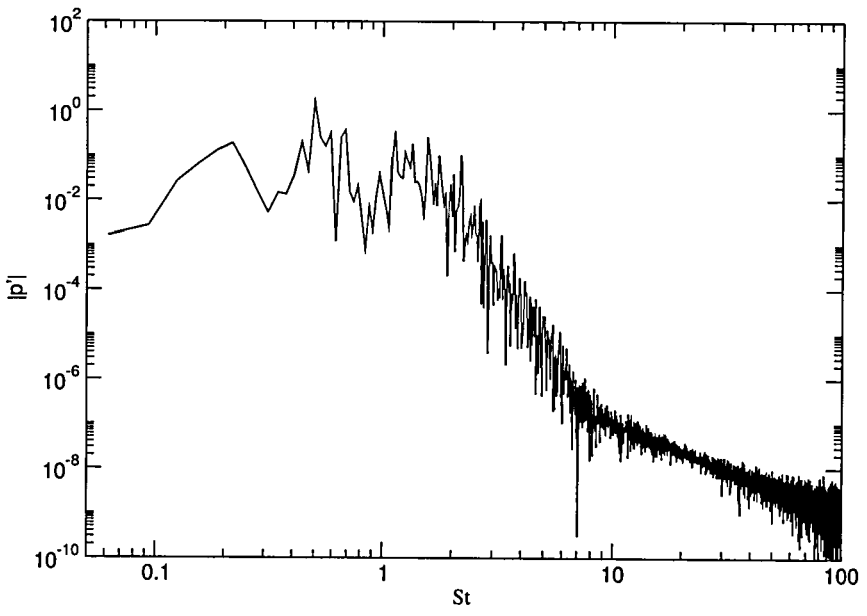


(b) power spectrum

Figure 7.14: Pressure fluctuation at Pt5

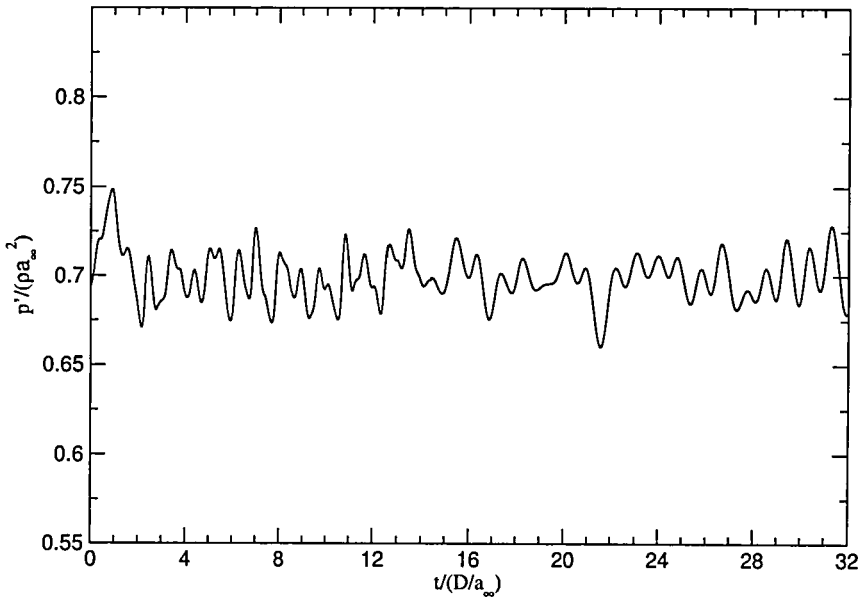


(a) time history

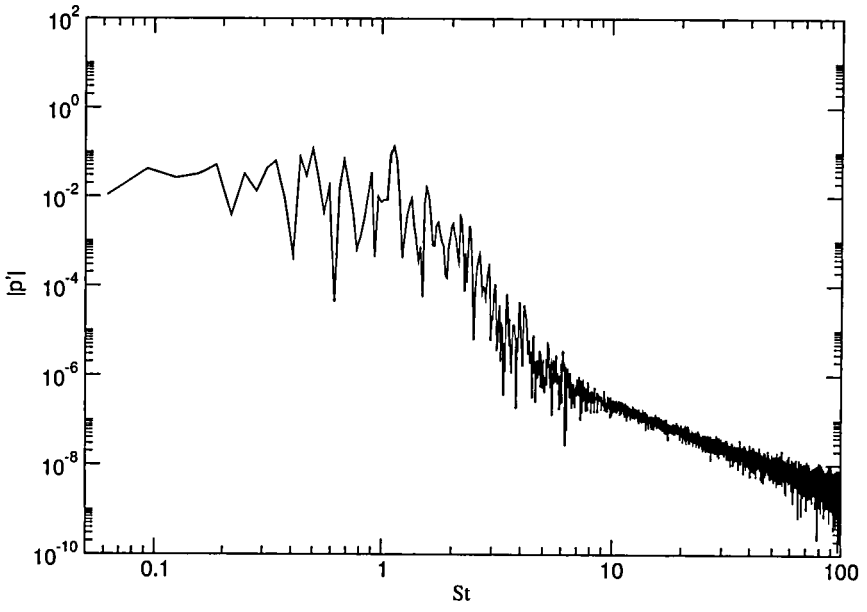


(b) power spectrum

Figure 7.15: Pressure fluctuation at Pt6



(a) time history



(b) power spectrum

Figure 7.16: Pressure fluctuation at Pt9

Chapter 8

Conclusion

A parallel mesh transparent numerical method for Large-Eddy Simulation of compressible turbulent flow has been presented. The hybrid unstructured mesh method is designed to handle engineering flow problems involving complex geometry at reasonable computing time. The results are generally comparable to previously published works using structured meshes that involve a more complicated mesh generation process than unstructured meshes.

The accomplishments and findings are presented in accordance to the objectives laid out in Chapter 1. Subsequently, further research works are recommended.

8.1 Accomplishments and Findings

Fourier analysis of the finite volume discretisation of a generic linear differential equation has been performed. Although it shows improvement can be gained from having a higher than second order compact scheme on a uniform triangular stencil, such improvement is not directly extendable to general triangulation. Hence, the application of the higher order scheme for a code aimed at solving practical flow problems is not felt justified.

A second order accurate mesh-transparent method using a modified upwind scheme has been developed to perform LES and RANS computation of three dimensional compressible turbulent flow. A simple algebraic function is used to activate the upwinding only in the vicinity of a shock wave. The computer code has necessarily been parallelised so that an LES solution can be obtained within a reasonable time scale. The

MPI library has been used for the parallelisation to ensure that the code can be ported easily to either distributed or shared memory parallel machines.

Validation of the basic numerics for steady flow using the mesh-transparent numerical method has been presented. The two steady flow cases illustrate the ability of the present code in handling complex 3D geometries typically found in practical engineering flow problems.

The present unsteady algorithm has been assessed on three flow problems. The first test case is a purely numerical two dimensional test case of the interaction between an oblique shock wave and laminar mixing layer, where the unstructured method shows comparable resolution to a previously published computation using structured fourth order compact scheme method on identical quadrilateral mesh. The test case also highlighted the substantial saving in computational nodes that can be gained from employing a hybrid mesh with quadrilateral cells in the mixing layer region and triangular cells for the region that is coarsened rapidly towards the far field boundary. The second test case is a turbulent round compressible free jet, where the hybrid meshing strategy has demonstrated that a solution of similar quality and resolution to an LES computation on structured mesh of 1.6M nodes can be obtained with only 800k nodes. The third test case is a low Mach number turbulent flow through a round pipe, where the code exhibit results typical to LES methods that is based upon coupled solver technology.

Having managed to reproduce the previously published LES computation, a simulation of a turbulent flow with a shock wave that is still relatively rare for LES is attempted. A preliminary LES computation of a highly under-expanded impinging jet with unsteady impingement shock is used for this purpose. The results show that the present technique of localising numerical dissipation to shock region works reasonably well. In comparison to previous numerical computations on this problem which have been conducted inviscidly, the present method produces a much more complete picture of the flow physics.

8.2 Further Works

In general, research aimed at more practical problems such as aeroacustics and combustion can be developed as extensions to present code.

The two jet test cases that have been used here highlight the need for a better inlet

condition that is more representative of the turbulent nozzle flow found in experiments. These cases also call for an extension of the present research in aeroacoustics field. Typically a computational aeroacoustic problem only concerns with propagation of pressure wave to the far field, which is almost always a linear problem. However the mechanisms of noise production in the near field is highly non-linear. For noise generated by bluff bodies, such as aircraft landing gear, it is the flow separation involving highly non linear vortex shedding that is the most important factor. Thus, utilisation of LES (or even DNS) for generating the instantaneous turbulence fluctuation rather than the time independent empirical model of RANS as a noise source has become increasingly popular in recent years.

Despite being able to show reasonable flow physics around the impingement point, the preliminary solution of the highly under expanded impinging jet suggests that further research is indeed required to investigate the discrepancy of the frequency of the shock oscillation.

Combustion, which involves appreciable change of density even when the fluid is confined in a closed chamber, is another area of research that the present code can be developed into. There has been clear evidence that the physics of combustion, notably the flame propagation, requires a non-linear approach rather than the linear one that is currently offered by many eddy viscosity based RANS. Thus, the extension of the present code would offer a more realistic combustion research capability that typically involves a rather complex combustor geometry such as those investigated by Moin and his co workers.

The present works also provided a platform upon which a further research towards better sub-grid scale model can be aimed. There are few doubts that despite being very popular for LES computations, the standard Smagorinsky model has some inherent limitations. The WALE model that has been used for the present work does address the standard Smagorinsky problem near a solid wall, but it still requires fine mesh spacing near the wall. Here a hybrid RANS/LES approach such as DES that is based on non-physical turbulent viscosity or turbulent kinetic energy that has been proposed by some authors should be a particular interest.

Appendix A

Reynolds Averaged Navier Stokes Equations

The Navier Stokes-Equations presented in eqns. 2.1, 2.2, 2.3, 2.4 can be written in differential form using vector notation as follows.

The continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{A.1}$$

The momentum equation:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\tau} = 0 \tag{A.2}$$

The total energy equation:

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot [\rho(E+p)\mathbf{u}] - \boldsymbol{\tau} : \nabla \cdot \mathbf{u} + \nabla \dot{\mathbf{q}} = 0 \tag{A.3}$$

where : is the scalar inner product of two symmetric tensors such that $\boldsymbol{\tau} : \nabla \cdot \mathbf{u} = \tau_{x_i x_j} \frac{\partial u_j}{\partial x_j}, i = 1,3$

RANS filtering is used to decompose the flow field into the time-averaged flow-field that is solved numerically and the fluctuating component that is to be modelled. Consider that field φ is a function of time t in the interval $(t-T, t+T)$ as well as coordinate position ξ. The filtering operation is defined by

$$\bar{\varphi}(\xi) = \frac{1}{2T} \int_{t-T}^{t+T} G(t)\varphi(\xi, t)dt \tag{A.4}$$

where the convolution filter G is unity in the interval.

Alternatively, RANS filtering can be performed over large number of N independent fields to obtain ensemble average as follows

$$\bar{\varphi}(\xi,t) = \frac{1}{N}\int_0^N G(n)\varphi(\xi,t,n)dn \tag{A.5}$$

where once again G is unity in $(0,N)$ interval. If N approaches infinity, ergodic hypothesis renders the two filtering techniques identical [135].

Unsteady RANS (URANS) can be interpreted as having an ensemble average over N samples, which is large but not sufficiently large to satisfy the ergodic hypothesis. Thus, unlike time filtering, the filtered variables still retain some time dependent fluctuations, which in turn can be decomposed into a time averaged mean flow and a very large scale fluctuations that are often called coherent fluctuation (see ref [99]). Hence this simulation is also called Very Large-Eddy Simulation (VLES) by some authors [114] as the resulting simulation behaves as if a very coarse spatial filter had been employed. As in RANS, the random, high frequency part of the fluctuation is modeled.

A density based averaging is then applied to the filtered equation to simplify the resulting RANS equations. The Favre averaging leads to the following decomposition :

$$\varphi = \widetilde{\varphi} + \varphi"$$

where the first term on the right hand side is the filtered variable, i.e. the low frequency component of the flow variable. The second term is the high frequency component that is removed by the filter, hence its effect will be modelled. Generally speaking the modelled term of τ and $\dot{\mathbf{q}}$ are negligible.

Applying the decomposition to continuity equation leads to

$$\frac{\partial\overline{\rho}}{\partial t} + \nabla\cdot(\overline{\rho}\widetilde{\mathbf{u}}) = 0 \tag{A.6}$$

Thus filtering does not alter the continuty equation. However, the same can not be said for the other two equations.

A.1 Filtered Momentum Equations

The decomposition of the convective term $\rho\mathbf{uu}$ yields

$$\overline{\rho u_i u_j} = \overline{\rho}\widetilde{u}_i\widetilde{u}_j + \widetilde{\overline{\rho}\widetilde{u}_i u_j"} + \widetilde{\overline{\rho}u_i"\widetilde{u}_j} + \widetilde{\overline{\rho}u_i"u_j"}$$

In RANS, the high frequency component, known as the turbulence fluctuation, has zero averaged value. This means that both the second and the third terms are zero. Thus the filtered convective term becomes

$$\overline{\rho u_i u_j} = \overline{\rho}\widetilde{u}_i\widetilde{u}_j + \overline{\rho}\widetilde{u_i'' u_j''}$$

Meanwhile the decomposition of the pressure gradient and the viscous flux terms simply yield

$$\overline{\nabla p} = \nabla \overline{p}$$

$$\overline{\nabla \cdot \tau} = \nabla \cdot \overline{\tau}$$

Thus the filtered momentum equation can be written as follows:

$$\frac{\partial \overline{\rho}\widetilde{\mathbf{u}}}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{\rho}\widetilde{u}_i\widetilde{u}_j + \overline{p}\delta_{ij}\right) - \frac{\partial}{\partial x_j}\left(\overline{\tau}_{x_j x_i} - \overline{\rho}\widetilde{u_i'' u_j''}\right) = 0 \tag{A.7}$$

A.2 Filtered Energy Equation

It must be recalled that the total internal energy can be written in term of internal and kinetic energy (eqn. 2.8). Thus, filtering the convection of total internal energy yields

$$\overline{\rho E u_j} = c_v \overline{\rho T u_j} + \frac{1}{2}\overline{\rho u_k u_k u_j}$$

As the time averaged of flucatuation quantities are zero, the above expression can be simplified as

$$\overline{\rho}\widetilde{Eu_j} = c_v\overline{\rho}\widetilde{T}\widetilde{\mathbf{u}} + c_v\overline{\rho}\widetilde{T''u_j''} + \frac{1}{2}\overline{\rho}\left[\widetilde{u}_k\widetilde{u}_k\widetilde{u}_j + \widetilde{u_k''u_k''}\widetilde{u}_j + 2\widetilde{u_j''u_k''}\widetilde{u}_k + \widetilde{u_k''u_k''u_j''}\right]$$

Notice that the filtered E can be written as follows

$$\widetilde{E} = c_v\widetilde{T} + \frac{1}{2}\overline{\rho}\widetilde{u}_k\widetilde{u}_k + \frac{1}{2}\overline{\rho}\widetilde{u_k''u_k''}$$

Thus, the filtered energy equation is defined as follows

$$\frac{\partial \overline{\rho}\widetilde{E}}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\overline{\rho}\widetilde{E} + \overline{p}\right)\widetilde{u}_j - \left(\overline{\tau}_{x_j x_i} - \overline{\rho}\widetilde{u_k''u_k''}\right)\widetilde{u}_j + \overline{q}_j + c_v\overline{\rho}\widetilde{T''u_j''} + \overline{pu_j''} - \mathcal{D}\right] = 0$$

where

$$\mathcal{D} = \overline{\left(\tau_{x_j x_i} - \frac{1}{2}u_k''u_k''\right)u_j''}$$

It is common practice for RANS to neglect the last term, thus the final form of the filtered energy equation is simply

$$\frac{\partial \overline{\rho}\widetilde{E}}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\overline{\rho}\widetilde{E} + \overline{p}\right)\widetilde{u}_j - \left(\overline{\tau}_{x_j x_i} - \overline{\rho}\widetilde{u_k'' u_k''}\right)\widetilde{u}_i + \overline{\dot{q}}_j + c_v\widetilde{\overline{\rho}T'' u_j''} + \overline{p u_j''}\right] = 0 \quad \text{(A.8)}$$

A.3 Eddy Viscosity Hypothesis

In this thesis, the RANS turbulence model is based on eddy-viscosity hypothesis of Boussinesq. Hence it is assumed that the filtered kinematic viscosity $\overline{\nu}$ can be mimicked by an eddy viscosity ν_t. Using analogy that kinematic viscosity and molecular viscosity μ is related to $\mu = \rho\nu$, the eddy viscosity can be related to "turbulence viscosity" as $\mu_t = \rho\nu_t$. Thus the turbulence fluctuation terms can be modelled as

$$\widetilde{\overline{\rho}u_i'' u_j''} = -\left[\overline{\rho}\mu_t\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i}\right) - \overline{\rho}\frac{2}{3}\mu_t\frac{\partial \widetilde{u}_k}{\partial x_k}\delta_{ij} - \frac{2}{3}\widetilde{\overline{\rho}u_k'' u_k''}\delta_{ij}\right]$$

As shown by Warsi [135], the last term on the right hand side of eqn. A.9 involving turbulent kinetic energy, $\widetilde{u_k'' u_k''}$, is required to give the correct trace of the Reynolds stress, which takes into account compressibility effect on the turbulence field. Consequently, this term is not significant on low turbulent Mach number, thus

$$\widetilde{\overline{\rho}u_i'' u_j''} = -\left[\overline{\rho}\mu_t\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i}\right) - \overline{\rho}\frac{2}{3}\mu_t\frac{\partial \widetilde{u}_k}{\partial x_k}\delta_{ij}\right] \quad \text{(A.9)}$$

As a result the viscous shear stress for RANS computation can be written as

$$\overline{\tau}_{x_i x_j} - \overline{\rho}\widetilde{u_i'' u_j''} = \overline{\rho}\left(\mu + \mu_t\right)\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \widetilde{u}_k}{\partial x_k}\delta_{ij}\right) \quad \text{(A.10)}$$

Similarly, the last two terms on the right hand side of eq. A.8 is modelled as "turbulence conduction"

$$\widetilde{\overline{\rho}c_v T'' u_i''} + \overline{p u_j''} = \frac{\mu_t c_p}{Pr_t}\frac{\partial \widetilde{T}}{\partial x_i} \quad \text{(A.11)}$$

where turbulent Parndtl Number Pr_t is fixed at 0.9 for the present work.

A.4 Spalart-Allmaras Turbulence Model

The Spalart-Allmaras (S-A) model [116], which was based on a single transport equation for modified turbulent viscosity is chosen for the present work as it offers good

compromise between accuracy and simplicity. While the fact that this model and its constants were developed from a dimensional analysis, Galilean invariance and selected empirical results rather than flow physics is acknowledged, the work of Bardina et al. [7], which suggested that the model performed reasonably well for a range of turbulent flows, should justify its application for the present work. Having said that, it must me noted that modified turbulent viscosity is not a physical quantity.

The eddy viscosity is defined in term of the modified eddy viscosity variable $\tilde{\nu}$ and viscosity function f_{v1} as follows:

$$\nu_t = \tilde{\nu} f_{v1} \tag{A.12}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{A.13}$$

$$\chi = \frac{\tilde{\nu}}{\nu} \tag{A.14}$$

The incompressible non-conservative form of transport equation that was used in the original paper [116] is employed. Unless the main flow field is hypersonic, the turbulence Mach number is unlikely to be high enough to warrant a compressible turbulent model.

$$\frac{\partial}{\partial t}\iiint_{\Gamma} \tilde{\nu} dV + \iint_{\partial\Gamma} \tilde{\nu} U_n dS - \iint_{\partial\Gamma} (\nu + \nu_t)\frac{1}{\sigma}\nabla\tilde{\nu} \cdot \hat{n} dS = Source \tag{A.15}$$

$$Source = \iiint_{\Gamma} \left[\underbrace{c_{b1}(1 - f_{t2})\tilde{\nu}\tilde{\Omega}}_{production} + \underbrace{\frac{c_{b2}}{\sigma}(\nabla\tilde{\nu} \cdot \nabla\tilde{\nu})}_{diffusion} - \underbrace{c_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2}_{destruction} \right] dV \tag{A.16}$$

where d is the nearest distance to a wall, κ is Von Karman constant, and σ is the Prandtl number. The transition trip term is not implemented in the present work, because it is difficult to define a trip location in a three-dimensional unstructured mesh. Here the magnitude of vorticity Ω with the summation convention implied is defined as follows

$$\Omega_{ij} = \sqrt{\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)^2} \tag{A.17}$$

The modified vorticity $\tilde{\Omega}$ is defined using the original formulation of Spalart-Allmaras model given in [116]

$$\tilde{\Omega} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \tag{A.18}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{A.19}$$

The empirical functions for the no-slip wall effects are defined as follows

$$f_w = g\left(\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right)^{1/6} \tag{A.20}$$

$$g = r + c_{w2}\left(r^6 - r\right) \tag{A.21}$$

$$r = \frac{\tilde{\nu}}{\tilde{\Omega}\kappa^2 d^2} \tag{A.22}$$

The empirical constants that are suggested by Spalart and Allmaras are used for the present work without modification. The values are shown in table A.1

constant	value
c_{b1}	0.1355
c_{b2}	0.622
τ	2/3
κ	0.41
c_{w1}	$\frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\sigma}$
c_{w2}	0.3
c_{w3}	2.0
c_{v1}	7.1

Table A.1: Empirical constants for Spalart-Allmaras model

Appendix B

Post processing the LES data

B.1 Transformation between Cartesian and Cylindrical Coordinates Systems

The cylindrical nature of all the domain means that the cylindrical velocity components, rather than the Cartesian velocity components are more useful for the post processing. Since the present code is written in Cartesian system, transformation operator is needed to relate the two coordinates systems.

The cylindrical coordinates system is oriented along the x-axis. Thus this chapter section with transformation between $x-y-z$ and $x-r-\theta$ coordinate systems. Using fig. B.1, the v and w velocities can be transfrom into its polar counterpart u_r and ru_θ as follows:

$$u_r = v\sin\theta + w\cos\theta \tag{B.1}$$

$$ru_\theta = -v\cos\theta + w\sin\theta \tag{B.2}$$

where

$$\sin\theta = \frac{y}{r} \tag{B.3}$$

$$\cos\theta = \frac{z}{r} \tag{B.4}$$

$$r = \sqrt{y^2 + z^2} \tag{B.5}$$

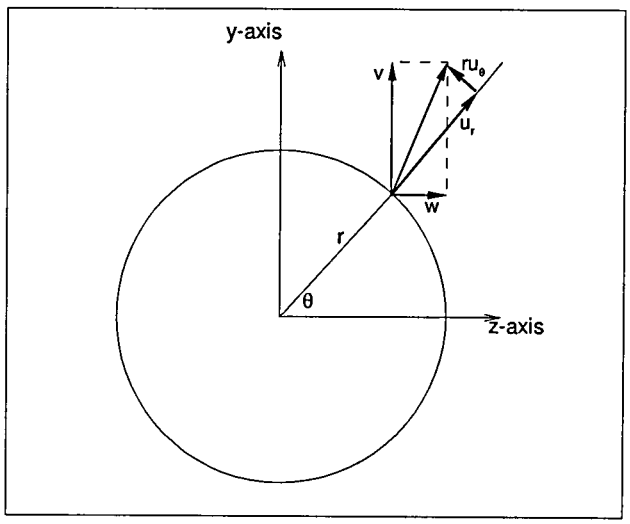


Figure B.1: Cartesian velocities and polar velocities

The transformation can also be applied to the Cartesian Reynolds stresses $\widetilde{u"v"}$ and $\widetilde{u"w"}$ and their cylindrical counterpart $\widetilde{u"u_r"}$ and $\widetilde{u"ru_\theta"}$ since u can be regarded as a scalar multiplier.

$$\widetilde{u"u_r"} = \widetilde{u"v"} \sin\theta + \widetilde{u"w"} \cos\theta \tag{B.6}$$

$$\widetilde{u"ru_\theta"} = -\widetilde{u"v"} \cos\theta + \widetilde{u"w"} \sin\theta \tag{B.7}$$

However, the transformation of the Reynolds stresses associated with the square of the fluctuation velocity, namely $\widetilde{v"v"}$ and $\widetilde{w"w"}$ is not trivial since such an operation lead to a positive definite Cartesian component in all of the four polar quadrant. This in turn removes any trace of the velocity direction, which is essential for the transformation defined previously. To overcome this problem, it is assumed that the $\widetilde{u"v"}$ and $\widetilde{u"w"}$ retain the direction for $\widetilde{v"v"}$ and $\widetilde{w"w"}$, respectively. Therfore the transformation is modified as follows:

$$\widetilde{u_r"u_r"} = \widetilde{v"v"} \sin\theta sign\left(\widetilde{u"v"}\right) + \widetilde{w"w"} \cos\theta sign\left(\widetilde{u"w"}\right) \tag{B.8}$$

$$\widetilde{u"ru_\theta"} = -\widetilde{v"v"} \cos\theta sign\left(\widetilde{u"v"}\right) + \widetilde{w"w"} \sin\theta sign\left(\widetilde{u"w"}\right) \tag{B.9}$$

B.2 Turbulent statistics

The mean value could be calculated by averaging all of the sample taken during the LES computation at post processing stage.

$$\widetilde{\varphi} = \frac{\varphi_1 + \varphi_2 + \cdots \varphi_n}{n} \tag{B.10}$$

However, this would lead to enormous amount of data being stored. Although a snapshot of the flow is inevitably dumped as the LES progress, its frequency is normally much less that that of sampling frequency necessary to obtain a good statistics. The most widely used alternative is to urolling averaged, i.e. performing the averaging as the computation progresses:

$$\widetilde{\varphi}_n = \frac{(n-1)\widetilde{\varphi}_{n-1}}{n} \tag{B.11}$$

where n is number of samples when the rolling averages is performed. Thus only the mean values are stored.

The second order statitics for the Reynolds stresses are computed in two stages. Firstly the rolling averaged of the product is computed as the simulation progresses

$$\left(\widetilde{\varphi\varphi}\right)_n = \frac{(n-1)\left(\widetilde{\varphi\varphi}\right)_{n-1}}{n} \tag{B.12}$$

Secondly, the mean of the product of the fluctuation is calculated at the end of the simulation

$$\left(\widetilde{\varphi''\varphi''}\right)_n = \left(\widetilde{\varphi\varphi}\right)_n - \widetilde{\varphi}_n\widetilde{\varphi}_n \tag{B.13}$$

Bibliography

[1] J. D. Anderson. *Computational Fluid Dynamics: The Basics with Applications.* McGraw-Hill Book Company, 1995.

[2] W. K. Anderson. Grid generation and flow solution method for euler equations on unstructured grids. Technical Memorandum TM-1992-4295, NASA, 1992.

[3] W. K. Anderson and D. L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers & Fluids*, 23:1–21, 1994.

[4] Y. Aoyama and J. Nakano. *RS/6000 SP: Practical MPI Programming.* IBM Corporation, International Technical Support Organization, 1999. www.redbooks.ibm.com.

[5] J. S. Bagget. On the feasibility of merging LES with RANS for the near-wall region of attached turbulent flows. In *Annual Research Briefs 1998*, pages 267–277. Center for Turbulence Research, Stanford University, 1998.

[6] J. Bardina, J. H. Frziger, and W. C. Reynolds. Improved subgrid scale models for large-eddy simulations. AIAA Paper 80–1357, 1980.

[7] J. E. Bardina, P. G. Huang, and T. J. Coakley. Turbulence modelling validation, testing and development. Technical Memorandum TM-1997-110446, NASA, 1997.

[8] T. J. Barth. On unstructured grid and solvers. In *Computational Fluid Dynamics*, VKI Lecture Series 1990-03. Von Karman Institute, 1990.

[9] T. J. Barth and P. O. Frederickson. Higher order solution of the euler equations on unstructured grids using quadratic reconstruction. AIAA Paper 90–0013, 1990.

[10] T. J. Barth and D. C. Jespersen. The design and application of schemes on unstructured meshes. AIAA Paper 89–0366, 1989.

[11] F. Bassi and S. Rebay. A high order accurate discontinuous finite element method for numerical solution of compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

[12] P. Beaudan and P. Moin. Numerical experiments on the flow past a circular cylinder at sub-critical reynolds numbers. Technical Report TF-62, Department of Mechanical Engineering, Stanford University, 1994.

[13] B. J. Boersma, G. Brethouwer, and F. T. M. Nieuwstadt. A numerical investigation on the effect of the inflow conditions on the self-similar region of a round jet. *Physics of Fluids*, 10:899–909, 1998.

[14] B. J. Boersma and S. K. Lele. Large eddy simulation of compressible turbulent jet. In *Annual Research Briefs 1999*, pages 365–377. Center for Turbulence Research, Stanford University, 1999.

[15] C. Bogey, C. Bailly, and D. Juvé. Flow field and sound radiation of a mach 0.9 jet computed by LES. In *RTO/AVT Symposium on Development in Computational Aero- and Hydro- Acoustics*, 2001.

[16] J. P. Boris, F. F. Grinstein, E. S. Oran, and R. L. Kolbe. New insights into large eddy simulations. *Fluid Dynamics Research*, 10:199–228, 1992.

[17] T. T. Bui. A parallel finite-volume algorithm for large-eddy simulation of turbulent flows. Technical Memorandum TM-1999-206570, NASA, 1999.

[18] S. Camarri and M. S. Salvetti. Towards the large-eddy simulation of complex engineering flows with unstructured grids. Rapport de Recherche 3844, INRIA, 1999.

[19] F. E. Camelli and R. Löhner. Combining the baldwin lomax and smagorinsky turbulence models to calculate flows with separation regions. AIAA Paper 2002–0426, 2002.

[20] D. Carati and A. A. Wray. Large-eddy simulations with explicit equations for subgrid-scale quantities. In *Proceedings of the Summer Program 2002*, pages 379–86. Center for Turbulence Research, Stanford University, 2002.

[21] B. Cockburn and C. W. Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. Technical Report 97-32, ICASE, 1997.

[22] B. Cockburn and C. W. Shu. Runge-kutta discontinuous galerkin methods for convection dominated problems. Technical Report SC2000-12, Brown University, 2000.

[23] L. Davidson. LESFOIL: A european project on large eddy simulation around high-lift airfoil at high reynolds number. In *European Congress on Computational Fluid Dynamics Conference ECCOMAS 2000*, September 2000.

[24] L. Davidson. Hybrid LES-RANS: A combination of a one-equation sgs model and a k-ω model for predicting recirculating flows. In *European Congress on Computational Fluid Dynamics Conference ECCOMAS 2001*, September 2001.

[25] A. Deardorff. Numerical study of three-dimensional turbulent channel flow at large reynolds numbers. *Journal of Fluid Mechanics*, 41:453–480, 1970.

[26] J. R. DeBonis and J. N. Scott. A large-eddy simulation of a turbulent compressible round jet. AIAA Paper 2001–2254, 2001.

[27] C. D. Donaldson and R. S. Snedeker. A study of free jet impingement. part 1. mean properties of free and impinging jet. *Journal of Fluid Mechanics*, 45:281–319, 1971.

[28] D. Drikakis. Embedded turbulence model in numerical methods for conservation laws. *International Journal for Numerical Methods in Fluids*, 39:763–781, 2002.

[29] F. Ducros, V. Ferrand, F. Nicoud, C. Weber, D. Darracq, C. Gacheriu, and T. Poinsot. Large-eddy simulation of the shock/turbulence interaction. *Journal of Computational Physics*, 152:517–549, 1999.

[30] F. Ducros, F. Nicoud, and T. Schönfeld. Large eddy simulation of compressible flow on hybrid mesh. In *Proceedings of the Eleventh Symposium on Turbulent Shear Flows*, pages 28.1–28.6, 1997.

[31] P. A. Durbin. Separated flow computations with the k-ϵ-v^2 model. *AIAA Journal*, 33:659–664, 1995.

[32] F. Durst, J. Jovanovic, and J. Sender. Lda measurements in the near-wall region of a turbulent pipe flow. *Journal of Fluid Mechanics*, 295:305–335, 1995.

[33] J. G. M. Eggels, F. Unger, M. H. Weiss, J. Westerweel, R. J. Adrian, and F. T. M. Nieuwstadt. Fully developed turbulent pipe flow: A comparison between direct

numerical simulation and experiment. *Journal of Fluid Mechanics*, 268:175–209, 1994.

[34] J. H. Ferziger. Higher-level simulations of turbulent flows. In J. A. Essers, editor, *Computational Methods for Turbulent, Transonic, and Viscous Flows*, pages 93–182. Hemisphere Publishing Corp., 1983.

[35] J. H. Ferziger. Introduction to the physics and simulation of turbulence. In *Introduction to the Modelling of Turbulence*, VKI Lecture Series 1997-03. Von Karman Institute, 1997.

[36] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics.* Springer-Verlag, second edition, 1999.

[37] I. Foster. *Designing and Building Parallel Programs.* Addison-Wesley Publishing Company, 1995.

[38] J. B. Freund. Noise sources in a low-reynolds-number turbulent jet at mach 0.9. *Journal of Fluid Mechanics*, 438:277–305, 2001.

[39] C. Fureby and F. F. Grinstein. Monotonically integrated large eddy simulation of free shear flows. *AIAA Journal*, 37:544–556, 1999.

[40] C. Fureby and F. F. Grinstein. Large eddy simulation of high reynolds number flows. In C. Dopazo, editor, *Proceedings of the Eight European Turbulence Conference*, pages 595–598, June 2000.

[41] C. Fureby, G. Tabor, H. G. Weller, and A. D. Gosman. Differential subgrid stress model in large eddy simulations. *Physics of Fluids*, 9:3578–3580, 1997.

[42] L. Gamet, F. Ducros, F. Nicoud, and T. Poinsot. Compact difference schemes on non uniform meshes aplication to direct numerical simulation of copmpressible flows. *International Journal for Numerical Methods in Fluids*, 29:159–191, 1999.

[43] E. Garnier, M. Mossi, P. Sagaut, P. Comte, and M. Deville. On the use of shock capturing schenes for large-eddy simualtion. *Journal of Computational Physics*, 153:273–311, 1999.

[44] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic sub-grid scale eddy viscosity model. *Physics of Fluids A*, 3:1760–1765, 1991.

[45] S. Ghosal. An analysis of numerical errors in large-eddy simulation of turbulence. *Journal of Computational Physics*, 125:187–206, 1996.

[46] S. Gottlieb and C. W. Shu. Total variation diminishing runge-kutta schmes. Technical Report 96-50, ICASE, 1996.

[47] W. Gropp. Tutorial on mpi: The message-passing interface. Mathematics and Computer Science Division, Argonne National Laboratory.

[48] R. P. Hansen. *Unsteady Analysis of Bluff Bodies Using the Nonlinear Disturbance Equation.* PhD thesis, Pennsylvania State University, 2001.

[49] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8:2182–2189, 1965.

[50] A. C. Haselbacher. *A Grid-Transparent Numerical Method for Compressible Viscous Flows on Mixed Unstructured Grid.* PhD thesis, Loughborough University, 1999.

[51] B. Henderson, J. Bridges, and M. Wernet. An investigation of the flow structure of tone producing supersonic impinging jets. AIAA Paper 2002–2529, 2002.

[52] Brenda E. Henderson. An experimental investigation into the sound producing characteristics of supersonic impinging jets. AIAA Paper 2001–2145, 2001.

[53] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 2. John Willey & Sons, 1990.

[54] K. Horiuti. Large-eddy simulation of turbulent channel flow by one-equation modelling. *Journal of the Physical Society of Japan*, 54:2855–2865, 1985.

[55] J. C. R. Hunt, A. A. Wray, and P. Moin. Eddies, stream, and convergence zones in turbulent flows. In *Center for Turbulence Research Report CTR-S88*, page 193, 1988.

[56] H. J. Hussein, S. P. Capp, and W. K. George. Velocity measurements in a high-reynolds-number, momentum-conserving, axisymmetric, turbulent jet. *Journal of Fluid Mechanics*, 258:31–75, 1994.

[57] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the euler equations finite volume methods using runge-kutta time-stepping. AIAA Paper 81–1259, 1981.

[58] K Jansen. Large-eddy simulation using unstructured grids. In C. Liu and Z. Liu, editors, *Advances in DNS/LES*, 1st AFOSR International Conference on DNS/LES, pages 117–128. Greyden Press, August 1997.

[59] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.

[60] G. S. Karamanos and G. E. Karniadakis. A spectral vanishing viscosity methods for large-eddy simulations. *Journal of Computational Physics*, 163:22–50, 2000.

[61] G. Karypis and V. Kumar. Metis - a software package for partioning unstructured graphs, partioning meshes, and computing fill-reducing ordering of sparse matrices, version 4.0. Technical report, University of Minnesota, Department of Computer Science / Army HPC Research Centre, 1998. http://www.cs.umn.edu/ metis.

[62] D. Knight, Zhou G., N. Okong'o, and V. Shukla. Compressible large eddy simulation using unstructured grids. AIAA Paper 98–0535, 1998.

[63] A. G. Kravchenko and P. Moin. On the effect of numerical errors in large eddy simulation of turbulent flows. *Journal of Computational Physics*, 131:310–322, 1997.

[64] E. Kreyzig. *Advanced Engineering Mathematics*. John Willey & Sons, seventh edition, 1993.

[65] A. Krothapalli, E. Rajkuperan, F. Alvi, and L. Lourenco. Flow field and noise characteristics of a supersonic impinging jet. *Journal of Fluid Mechanics*, 392:155–181, 1999.

[66] J. C. Lau, P. J. Morris, and M. J. Fisher. Measurements in subsonic and supersonic free jets using a laser velocimeter. *Journal of Fluid Mechanics*, 93:1–27, 1979.

[67] S. K. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103:16–42, 1992.

[68] M. A. Leschziner. LES of separated high-lift aerofoil flow at Re=2.2M, some lessons from EU project LESFOIL. LESUK2 Workshop at Loughborough University, November 2002.

[69] M. A. Leschziner and D. Drikakis. Turbulence modelling and turbulent-flow computation in aeronautics. *The Aeronautical Journal*, 0000:349–384, 2003.

[70] M. S. Liou. Ten years in the making – ausm-family. AIAA Paper 2001–2521, 2001.

[71] R. Löhner. *Applied CFD Techniques: An Introduction based on Finite Element Method.* John Wiley & Sons, 2001.

[72] G. R. Luecke, B. Raffin, and J. J. Coyle. Comparing the communication performance and scalability of a sgi origin 2000, a cluster of origin 2000's and a cray t3e-1200 using shmem and mpi routines. Iowa State University, October 1999.

[73] N. Lupoglazoff, A. Biancherin, F. Vuillot, and G. Rahier. Comprehensive 3d unsteady simulations of subsonic and supersonic hot jet flow-fields. part i: Aerodynamic analysis. AIAA Paper 2002–2599, 2002.

[74] N. MacDonald, E. Minty, J. Malard, T. Harding, S. Brown, and M. Antonioletti. Writing message passing parallel programs with mpi, course notes version 1.8.3. Edinburgh Parallel Computing Centre, University of Edinburgh.

[75] M. P. Martin, U. Piomelli, and G. V. Candler. Subrid-scale model for compressible large-eddy simulations. *Theoretical and Computational Fluid Dynamics*, 13:361–376, 2000.

[76] W. D. McComb. *The Physics of Turbulence.* Oxford University Press, 1990.

[77] O. Métais and M. Lesieur. Spectral large-eddy simulation of isotropic and stably stratified turbulence. *Journal of Fluid Mechanics*, 256:157–194, 1992.

[78] R. Mittal and P. Moin. Suitability of upwind-biased finite difference schemes for large-eddy simulation of turbulent flows. *AIAA Journal*, 35:1415–1417, 1997.

[79] P. Moinier. *Algorithm Developments for an Unstructured Viscous Flow Solver.* PhD thesis, University of Oxford, 1999.

[80] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin. Fully conservative higher order finite difference scheme for incompressible flow. *Journal of Computational Physics*, 143:90–124, 1998.

[81] S. A. Morton, J. R. Forsythe, K. D. Squires, and K. E. Wurtzler. Assessment of unstructured grids for detached eddy simulation of high reynolds number separated flows. In *Proceedings of 8th International Conference on Numerical Grid Generation in Computational Field Simulation*, 2002.

[82] R. D. Moser, B. Cantwell, and L. P. Purtell. Chapter 5: Pipes and channels. In *A Selection of Test Cases for the Validation of Large-Eddy Simulations of Turbulent Flows*, number AGARD AR-345, pages 23–28, April 1998.

[83] M. Mossi and P. Sagaut. Numerical investigation of fully developed channel flow using shock-capturing schemes. *Computers & Fluids*, 32:249–274, 2003.

[84] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62:183–200, 1999.

[85] N. V. Nikitin, F. Nicoud, B. Wasistho, K. D. Squires, and P. R. Spalart. An approach to wall modelling in large-eddy simulations. *Physics of Fluids*, 12:1629–1632, 2000.

[86] G. J. Page, J. J. McGuirk, D. Jiang, L. R. Harper, and C. J. Penrose. Application of computational fluid dynamics to hot gas ingestion modelling. In *International Powered Lift Conference*, pages 23.1 – 23.11, September 1998.

[87] N. R. Panchapakesan and J. L. Lumley. Turbulence measurements in axisymetruc jets of air and helium. part 1. air jet. *Journal of Fluid Mechanics*, 246:197–223, 1993.

[88] S. Parneix, M. Behnia, and P. A. Durbin. Prediction of heat transfer in a jet impinging on a heated pedestal. *ASME Journal of Heat Transfer*, 121:43–49, 1998.

[89] U. Piomelli and J. R. Chasnov. Large eddy simulations : Theory and application. In D. Henningson, M. Hallbaeck, H. Alfreddson, and A. Johansson, editors, *Transition and Turbulence Modelling*, pages 269–336. Kluwer Academics Publishers, 1996.

[90] S. Pirzadeh and N. T. Frink. Assessment of the unstructured grid software tetruss for drag prediction of the dlr-f4 configuration. AIAA Paper 2002–0839, 2002.

[91] S. Z. Pirzadeh. Aiaa cfd drag prediction workshop, June 2001. http://www.aiaa.org/tc/apa/dragpredworkshop/dpw.html.

[92] A. Powell. On the mechanism of choked jet noise. In *Proceedings of the Physical Society of London*, volume B, pages 1039 – 1037, 1953.

[93] A. Powell. The sound-producing oscillations of round underexpanded jets impinging on normal plates. *Journal of the Acoustical Society of America*, 83:515–533, 1988.

[94] R. D. Rausch. Tri8it user guide. Aerodynamic and Acoustic Method Branch, NASA Langley Research Center, 1994.

[95] G. Redeker. Dlr-f4 wing body configuration. In *A Selection of Experimental Test Cases for the Validation of CFD Codes Vol. II*, number AGARD AR-303, pages B4–1 – B4–21, August 1994.

[96] W. Rodi, J. H. Ferziger, M. Breuer, and Porquiré. Status of large eddy simulation: Results of a workshop. *ASME Journal of Fluids Engineering*, 119:248–262, 1997.

[97] P. L. Roe. Approximate riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[98] Y. Saad. *Iterative Methods for Sparse Linear System.* PWS Publishing Company, 1996.

[99] P. Sagaut. *Large Eddy Simulation for Incompressible Flows.* Springer-Verlag, 2002.

[100] Y. Sakakibara and J. Iwamoto. Oscillation of impinging jet with generation of acoustic waves. *International Journal of Aeroacustics*, 1:385–402, 2002.

[101] H. Salman, J. J. McGuirk, G. J. Page, and P. Moinier. The influence of unstructured mesh type on prediction of convoluted shear layer. In *European Congress on Computational Fluid Dynamics Conference ECCOMAS 2000*, September 2000.

[102] T. Schönfeld and M. Rudgyard. Steady and unsteady flows simulations using the hybrid flow solver avbp. *AIAA Journal*, 37:1378–1385, 1999.

[103] U. Schumann. Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of Computational Physics*, 18:376–404, 1975.

[104] V. Scmitt and F. Charpin. Pressure distribution on the onera-m6-wing at transonic mach numbers. In *Experimental Data Base for Computer Program Assessment*, number AGARD AR-138, pages BI – BI–38, May 1979.

[105] L. Selle, G. Lartigue, T. Poinsot, P. Kaufmann, W. Krebs, and D. Veynante. Large-eddy simulations of turbulent combustion for gas turbines with reduced chemistry. In *Proceedings of the Summer Program 2002*, pages 333–344. Center for Turbulence Research, Stanford University, 2002.

[106] M. Shur, P. R. Spalart, M. Strelets, and A. Travin. Detached-eddy simulation of an aerofoil at high angle of attack. In W. Rodi and D. Laurence, editors, *4th International Symposium on Engineering Turbulence Modelling and Experiments*, pages 669–678. Elsevier, May 1999.

[107] M. L. Shur, P. R. Spalart, M. Kh. Strelets, and A. K. Travin. Towards the prediction of noise from the jet engines. *International Journal of Heat and Fluid Flow*, 24:551–561, 2003.

[108] D. Sidilkover. A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible euler equations. Technical Report 94-84, ICASE, 1994.

[109] T. A. Simons and R. H. Pletcher. Large eddy simulation of turbulent flows using unstructured grids. AIAA Paper 98–3314, 1998.

[110] K. Siniša and L. Davidson. Large eddy simulation of the flow around a three dimensional bluff body. AIAA Paper 2001–0432, 2001.

[111] J. W. Slater. Onera m6 wing: Study#1. NPARC Alliance Validation Archive. http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing01/m6wing01.html.

[112] J. Smagorinsky. General circulation experiments with the primitive equations, i. the basic experiment. *Monthly Weather Review*, 91:99–164, 1963.

[113] S. Smirnov, C. Lacor, B. Lessani, J. Meyers, and M. Baelmans. A finite volume formulation for compact schemes on arbitrary meshes with applications to RANS and LES. In *European Congress on Computational Fluid Dynamics Conference ECCOMAS 2000*, September 2000.

[114] P. R. Spalart. Strategies for turbulence modelling and simulations. *International Journal of Heat and Fluid Flow*, 21:252–263, 2000.

[115] P. R. Spalart. Young-person's guide to detached-eddy simulation grids. Contractor Report CR-2001-211032, NASA, 2001.

[116] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. AIAA Paper 92–0439, 1992.

[117] P. R. Spalart, W. H. Jou, M. Strelets, and S. R. Allmaras. Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach. In C. Liu and Z. Liu, editors, *Advances in DNS/LES*, 1st AFOSR International Conference on DNS/LES, pages 137–147. Greyden Press, August 1997.

[118] K. D. Squires, J. R. Forsythe, S. A. Morton, W. Z. Strang, K. E. Wurtzler, R. F. Tomaro, M. J. Grismer, and P. R. Spalart. Progress on detached-eddy simulation on massively separated flows. AIAA Paper 2002–1021, 2002.

[119] M. Strelets. Detached eddy simulation of massively separated flows. AIAA Paper 2001–0879, 2001.

[120] J. L. Stromberg, D. K. McLaughlin, and T. R. Troutt. Flow field and acoustic properties of a mach number 0.9 jet at a low reynolds number. *Journal of Sound and Vibration*, 72:159–176, 1980.

[121] C. K. W. Tam and J. C. Webb. Dispersion-relation-preserving finite difference schemes for acoustics. *Journal of Computational Physics*, 107:262–281, 1993.

[122] G. Tang, Z. Yang, and J. J. McGuirk. Large eddy simulation of flow in lean premixed prevapourised combustor geometries. In C. Dopazo, editor, *Proceedings of the Eight European Turbulence Conference*, pages 227–230, June 2000.

[123] L. Temmerman and M. A. Leschziner. Studies of LES-RANS hybrid startegiews for simulation of high reynolds wall-bounded flows. LESUK2 Workshop at Loughborough University, November 2002.

[124] R. H. Thomas, K. W. Kinzie, and S. P. Pao. Computational analysis of pylon-chevron core nozzle interaction. AIAA Paper 2001–2185, 2001.

[125] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation: Foundation and Applications.* Elsevier Science Publishing Company, 1985.

[126] I. H. Tristanto. Investigation of transient-convective numerical schemes for large-eddy simulation. MPhil to PhD Transfer Report, Loughborough University, 2001.

[127] G. Urbin and D. Knight. Large-eddy simulation of s supersonic boundary layer using an unstructured grid. *AIAA Journal*, 39:1288–1295, 2001.

[128] A. Uzun, G. A. Blaisdell, and A. S. Lyrintzis. Recent progress towards a large eddy simulation code for jet aeroacoustics. AIAA Paper 2002–2598, 2002.

[129] A. Uzun, G. A. Blaisdell, and A. S. Lyrintzis. 3-d large eddy simulation for jet aeroacoustics. AIAA Paper 2003–3322, 2003.

[130] B. van Leer. Towards the ultimate conservative difference scheme v: A second-order sequel to godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.

[131] O. V. Vasilyev, T. S. Lund, and P. Moin. A general class of commutative filter for LES in complex geometries. *Journal of Computational Physics*, 146:82–104, 1998.

[132] V. Venkatakrishnan. Convergence to steady-state solutions of the euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118:120–130, 1995.

[133] B. Vreman, B. Geurts, and H. Kuerten. Large eddy simulation of turbulent mixing layer. *Journal of Fluid Mechanics*, 339:357–390, 1997.

[134] B. Vreman, B. Geurts, H. Kuerten, and P. J. Zandbergen. A finite volume approach to large eddy simulation of compressible, homogeneous, isotropic, decaying turbulence. *International Journal for Numerical Methods in Fluids*, 15:799–816, 1992.

[135] Z. U. A. Warsi. *Fluid Dynamics: Theoretical and Computational Approaches.* CRC Press, 2000.

[136] A. Wood. Multi-dimensional upwind fluctuation splitting scheme with mesh adaption for hypersonic viscous flow. Technical Publication TP-2002-211640, NASA, 2001.

[137] Z. Yang and P. Voke. Large eddy simulation of transition under turbulence. http://www.vortex.mech.surrey.ac.uk/database/test73/test/test.html.

[138] H. C. Yee, N. A. Sandham, and M. J. Djomehri. Low-dissipative high-order shock-capturing method using characteristic-based filters. *Journal of Computational Physics*, 150:199–238, 1999.

[139] A. Yoshizawa. A statistical theory for compressible turbulent shear flows, with the application to subgrid modeling. *Physics of Fluids A*, 29:2152–2164, 1986.

[140] A. Yoshizawa. Subgrid-scale modelling with a variable length scale. *Physics of Fluids A*, 1:1293–1295, 1989.

[141] A. Yoshizawa. Bridging between eddy-viscosity-type and second-order models using a two scaled dia. In *9th International Symposium on Turbulent Shear Flow*, volume 3, pages 23.1.1–23.1.6, 1993.

[142] A. Yoshizawa and K. Horiuti. A statistically-derived subgrid-scale kinetic energy model for the large-eddy simulation of turbulent flows. *Journal of the Physical Society of Japan*, 54:2834–2839, 1985.

[143] K. B. M. Q. Zaman. Asymptotic spreading rate of initially compressible jets – experiment and analysis. *Physics of Fluids*, 10:2652–2660, 1998.

[144] D. Zingg and H. Lomax. Finite-difference schemes on regular triangular grids. *Journal of Computational Physics*, 108:306–313, 1993.