AUTOMATED DETECTION OF FEATURES IN CFD DATASETS

By

Satya Sridhar Dusi Venkata

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computational Engineering
in the College of Engineering

Mississippi State, Mississippi

December 2001

AUTOMATED DETECTION OF FEATURES IN CFD DATASETS

By

Satya Sridhar Dusi Venkata

Approved:

David S. Thompson Associate Research Professor of Computational Engineering (Director of Thesis)	Bharat K. Soni Professor of Aerospace Engineering (Committee Member)
Edward A. Luke Assistant Professor of Computer Science (Committee Member)	Michael J. Pearson Associate Professor of Mathematics (Committee Member)
Boyd Gatlin Associate Professor of Aerospace Engineering and Engineering Mechanics (Graduate Coordinator)	A. Wayne Bennett Dean of the College of Engineering

Name: Satya Sridhar Dusi Venkata

Date of Degree: December 14, 2001

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr. David S. Thompson

Title of Study: AUTOMATED DETECTION OF FEATURES IN CFD DATASETS

Pages in Study: 62

Candidate for Degree of Master of Science

Typically, computational fluid dynamic (CFD) solutions produce large amounts of data that can be used for analysis. The enormous amount of data produces new challenges for effective exploration. The prototype system EVITA, based on ranked access of application-specific regions of interest, provides an effective tool for this purpose. Automated feature detection techniques are needed to identify the features in the dataset. Automated techniques for detecting shocks, expansion regions, vortices, separation lines, and attachment lines have already been developed. A new approach for identifying the regions of flow separation is proposed. This technique assumes that each pair of separation and attachment lines has a vortex core associated with it. It is based on the velocity field in the plane perpendicular to the vortex core. The present work describes these methods along with the results obtained.

DEDICATION

I would like to dedicate this work to my parents.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my major professor, Dr. David Thompson, for his guidance and continuing assistance throughout this work. I would like to offer a special "thank you" to Dr. Bharat Soni for supporting me in this work. I would also like to thank the rest of my committee, Dr. Edward Luke and Dr. Michael Pearson. In addition, I would like to thank Dr. Boyd Gatlin and the Engineering Research Center for providing the facilities for this work. Finally this work was supported by the NSF Large Data and System Software Visualization Program (9982344).

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

NOMENCLATURE

Symbol:	
a	Speed of Sound
$\tilde{B}$	Bitstream
E	Energy
i, j, k	Orthogonal coordinate axes
I_1, I_2	Imaginary eigenvalues
L	Characteristic length associated with the size of the region of complex eigenvalues
$\vec{M}$	Mach number vector
M_n	Normal Mach number
$\vec{n}$	Unit vector parallel to the real eigenvector or vorticity vector
$\vec{n}_v$	Unit vector tangent to the vortex core
P	pressure
$\vec{q}$	Velocity vector
R_1, R_2	Real eigenvalues
$\tilde{S}$	Segmentation Map
$\bar{\bar{S}}$	Antisymmetric spin tensor
t_{conv}	Time to convect through region of complex eigenvalues
t_{orbit}	Period of motion of the spiral path of a swirling point
$\vec{V}$	Velocity vector
V_{conv}	Magnitude of velocity along L

$\vec{V}_i$	Velocity in computational space
$\vec{V}_r$	Velocity of a point in a plane normal to the vortex core
$\vec{V}_{reduced}$	Reduced velocity
$W_{feature}$	Weight assigned for a particular feature
$\dot{x}$	Velocity in x direction
(x_{cp}, y_{cp})	Coordinates of a critical point
ρ	Density
γ	Ratio of specific heats
τ	Swirl parameter
λ	Eigenvalue
∇	Gradient operator
∇P	Pressure gradient vector
$\nabla \vec{V}$	Velocity gradient tensor
$\bar{\bar{\Omega}}$	Symmetric deformation tensor

CHAPTER I

INTRODUCTION

Computational fluid dynamics (CFD) is the process of digitally simulating complex fluid flows. An analysis of this type typically produces large amounts of data. For example, the results of a steady three-dimensional CFD simulation on a 100 x 100 x 100 grid produces five scalars at 1e+06 points. For unsteady simulations, where multiple time steps are of interest, the data size increases further. Because of the amount of data, analysis of these results is typically performed using visualization techniques. The objective of visualization is to interactively transform the data into a pictorial form for better understanding. Visualization of very large datasets is often termed terascale visualization. Traditional visualization techniques do not work well for such large datasets. Very large amounts of data also present new challenges in storage and transmission. The increase in the size of the data that must be stored and transmitted produces a similar increase in the needed memory and bandwidth.

The EVITA project [1] proposes a systematic solution to the problem of terascale visualization. The key technology in the EVITA system prototype is the facility to provide ranked access to application-specific features in compressed representations. The EVITA project focuses on features of interest in CFD datasets. EVITA is similar to a client-server model with three main components - a pre-processor, a server, and a client. The pre-processor is responsible for identifying the regions of interest (ROIs), ranking and then, compressing them offline. The server schedules the transmission of these ROIs to the client in a compressed state based on the ranking of the features. The client receives these ROIs and decompresses them. The decompressed ROIs are then displayed by the client. This way, the visualization of relevant features in the data is optimized by reducing the amount of data transferred. The client needs to process only the relevant features rather than the whole dataset. Thus, the memory requirements on the client side are minimized. A detailed description of the EVITA system is presented in the next chapter. The present work deals with the development of a feature detection module for the EVITA system.

A feature is defined as a pattern occurring in the dataset that is of interest [2]. Shocks, vortices, and flow separations are among the important features that occur in fluid flows. For experimentally obtained data, visualization is done as a part of the measurement process [3]. In such techniques, the flows are studied using seeding with particles or using photographs of surface oil streaks. The seeding technique requires a large number of seeds to obtain high resolution of the features. The features are identified by observing the particles in the flow. The use of oil streaks requires tracing tangent curves on the surface. The features are identified by observing these tangent curves. These techniques also yield two-dimensional images that may not capture all the features of a three-dimensional flow. In numerical simulations, icons like streamlines, streak lines, glyphs, and iso-surfaces can be used for visualization. There are some difficulties associated with using these icons for feature detection [4]. For example, when a large number of streamlines or streak lines are needed near a point to resolve a feature, it may be difficult to visualize the feature because of the crowding of the icons at the point. The direction of a vector field cannot be deduced accurately with the use of glyphs in a three-dimensional flow field. Using these techniques, the user has to identify a feature by observing the pattern of the streamlines, streak lines, glyphs, or iso-surfaces. Hence, techniques that require minimum user interaction and do not need a prior knowledge of the locations of the features are desirable. Such types of methods are called automated techniques. A detailed description about the work being done in the field of visualization for CFD datasets is presented in [2].

Features may be characterized either by scalar or vector fields [2]. A feature detection algorithm may use scalar fields as in the case of a shock or use vector fields as in the case of a flow separation. Features can be classified as point-type, line-type, or region-type. Point type features are features that can be described using a single point. Line type features are features located on a curve in the flow field. Region type features are features that are defined for a surface or a volume in the flow field. There exists another type of classification of features based on the locality of the data the feature depends on. In this context, features can be classified as local or global. If a point can be identified as a particular feature by using the flow field at that point and its immediate neighbors only, this feature is called a local feature. Global features are those whose local characteristics may depend on the values at any point in the flow field. Another type of feature called set-type feature is defined in [5] . In [5], a vortex is defined as a set of points swirling around another set of points. Table 1.1 lists examples for the different types of features discussed. A region-type feature can be detected by identifying the points in the region independently and then performing a segmentation. If an algorithm identifies a point as a feature, then this algorithm is called point-based technique. If the algorithm identifies a region as a feature, it is a region-based technique. If the algorithm uses the solution field of a point and its neighbors only, it is a local-type feature detection algorithm. A global-type feature detection technique may use the solution at any point in the flow field.

Type of feature	Feature
Point type	Critical point
Line type	Attachment and separation lines
Region type	Vortex
Local type	Critical point
Global type	Streamlines
Set type	Vortex

Table 1.1: Examples for different type of features.

There has been much work in the general area of feature detection. Different techniques for shock detection are presented in [6], [7], and [8]. These techniques are based on scalar field variables or derived quantities such as Mach number and pressure, which depend on the conservative flow variables. These methods can be classified as point-based, local-type feature detection algorithms. Different algorithms for detecting vortex cores are presented in [5], [9], [10], and [11]. The techniques for detection of separation and attachment lines based on velocity vector field topology are discussed in [3], [12], and [13]. Except for [5], these techniques are either local-type or global-type techniques. The vortex is defined as a set-type feature in [5]. Some of these techniques identify features as line-type and others as region-type.

The following specifications are considered for developing the feature detection module in EVITA. This module is designed to take a structured grid and its solution field as inputs. Since the segmentation map generator requires a scalar value for every grid node as an input, the feature detection module should generate this as an output. Since EVITA deals with ROIs, the feature should be identified as a region in the flow field. Segmentation is not required if a region-type technique is used to identify the features. But, if a point-type algorithm is used, a segmentation is necessary. Additionally, it is desired to indicate the relative strength of the feature at each point identified by the scalar value assigned to the point. The present work discusses existing techniques that can be used to identify shocks, expansion regions, vortices, separation lines, and attachment lines as well as a new approach for identifying regions of separated flow subject to the above specifications. Since the features should be identified as a region, some of the existing techniques may not be sufficient for the present work. Hence, this work attempts to develop automated techniques that can identify the features in the dataset as a point-type or region-type features and assign a scalar value to every node identified to represent the relative strength of a feature at that point. This work deals with some of the existing techniques that can be used for this purpose along with a new technique for identifying separation layer.

Chapter II provides a brief overview of the EVITA system. Chapter III describes the existing automated feature detection algorithms that can be used to identify shocks, expansion regions, vortices, separation lines, and attachment lines. The new approach for identifying the separation layer is also discussed in this chapter. The results obtained using these algorithms on a few datasets are presented in chapter IV. Chapter V summarizes the present work and addresses the future work that can be done in automated feature detection.

CHAPTER II

THE EVITA SYSTEM

This chapter explains briefly the workings of the EVITA system [1]. EVITA is a prototype system for efficient storage, representation, and visualization of terascale datasets. The focus of EVITA is on time-varying, computational fluid dynamic and oceanographic solutions on structured rectilinear or curvilinear grids. The EVITA system is a client-server model which consists of a pre-processor, a server, and a client as shown in figure 2.1. Each of the shaded blocks in the figure can be considered as a module.

The pre-processor takes a structured rectilinear or curvilinear grid and its solution dataset as inputs. The *Feature Detector* module performs automatic feature detection and outputs a scalar value for every grid node. The *Significance Map Encoder* module takes in the scalar value output of the *Feature Detector* module as an input and generates a significance map. The significance map generator identifies the features as regions of interest (ROIs) and labels them. The labeling or ranking of these ROIs is based on a certain criteria. The *Wavelet Transform* module generates multi-scale wavelet coefficients for the field. Since the wavelet coefficients arise from multiple scales, a region cannot be identified using a single resolution significance map. Hence, a multi-scale significance map is used to robustly resolve the features at multiple levels. The *Embedded Encoder* module takes the grid, the wavelet transformations, and the significance map as input and generates an embedded bitstream $\tilde{B}$. The embedded bitstream consists of bitplanes of compressed ROIs in a particular order. The compression system chosen for encoding the data should be suitable for both scalar and vector fields and must allow perfect reconstruction of the original data. This compressed representation should also support partial reconstruction of the original dataset as desired by the user. The ranking of the ROIs is obtained from the

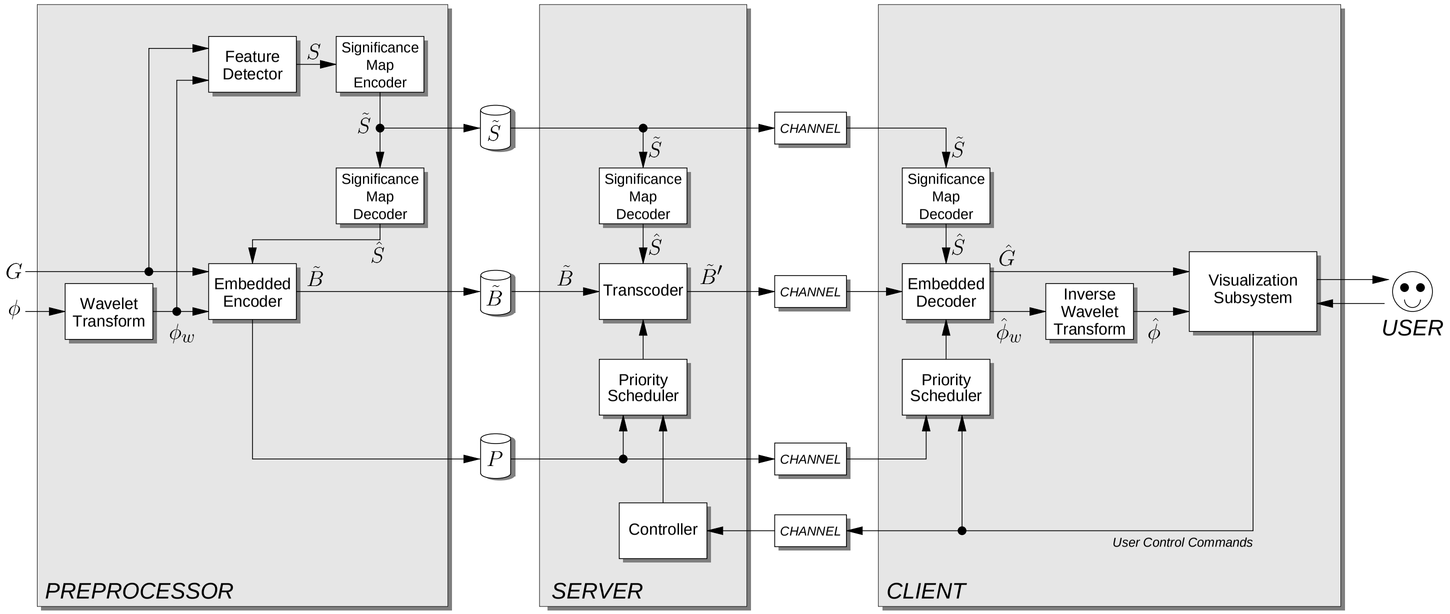


Figure 2.1: EVITA - A system for exploring terascale datasets.

significance map to establish a baseline priority of the ROIs in $\tilde{B}$. The computational work of feature detection, wavelet transformation, significance map generation, and generation of the embedded bitstream is done in the pre-processor in an offline manner.

The server is responsible for transmitting the bitstream $\tilde{B}$ to the client upon initialization of the visualization subsystem by the client. For the purpose of transmitting the bitstream in a manner requested by the client, a *Transcoder* is used. A *Transcoder* is defined as a processor that converts, or translates, data encoded in one encoding into another encoding [1]. The information about the ranking of the ROIs is obtained through a decoding of the segmentation map, $\tilde{S}$, by the *Significance Map Decoder.* At the start of the visualization, the server transmits each bitplane to the client in the same order as received in $\tilde{B}$. A user request to change the ROI rankings is sent to the *Priority Scheduler* through the *Controller.* Based on the requested priority, the *Transcoder* transmits the bitstream in a different encoding by simply relocating the bitplanes in $\tilde{B}$.

The client is responsible for decoding the bitplanes transmitted by the server using the significance map and the priority schedule. The *Embedded Decoder* performs the reverse operations of the *Transcoder* in the server and the *Embedded Encoder* in the pre-processor on the bitstream received. It produces the wavelet coefficients for each of the ROIs received. The *Inverse Wavelet Transform* performs an inverse wavelet transformation on the wavelet coeffients to generate the ROIs. Since the client knows the priority schedule on which the transcoder is operating, it can reorder the bitplanes as desired. This data is passed onto the *Visualization Subsystem* at frequent time-intervals. The *Visualization Subsystem* is responsible for displaying the features. It is also designed to provide an user interface to control the display of the features and the priority schedule of the ROIs. The *Priority Scheduler* in the client is responsible for communicating the user control commands to the server. The results for a two-dimensional vector field using the EVITA system are presented in [1].

The objective of present work is to enhance the *Feature Detector* module in the pre-processor to facilitate automatic feature detection of selected features. The module is designed to take a grid and its solution field as input and to identify the feature requested by the user. It outputs a scalar value (*weight*) to represent the strength of the feature at that particular point. A detailed description of this library can be found in [14].

CHAPTER III

FEATURE DETECTION ALGORITHMS

In this chapter, feature detection algorithms for shocks, expansion regions, vortices, and regions of separated flows are discussed. Each section discusses the various techniques that are available for detecting a specific feature and the technique that was actually implemented.

3.1 Shocks

A shock wave is a compression wave that may occur in fluid flows when the velocity of the fluid exceeds the local speed of sound. A shock is characterized by abrupt changes in flow quantities such as pressure, velocity, and density. The Mach number decreases while the pressure, density, and temperature increase across a shock. The properties of shocks are explained in more detail in [15]. The change in physical quantities is often very abrupt and hence a physical shock is nearly singular. In the case of numerical data, however, the discontinuity is typically smeared over several grid cells given the numerical approximation of the fluid dynamic equations and ensuing errors of the numerical implementation. A shock detection problem can be treated as an edge detection problem as in [6] or can be approached using the dynamics of fluid flow as in [7], [8], and [16].

In [6], a shock detection method based on the density gradient is used to detect steady shocks. The first and second derivatives of density in the direction of the local flow velocity are calculated. If the second derivative is zero and the first derivative is positive, a discontinuity is assumed to be present at that point. This technique is applied to all the nodes in the flow and the nodes where the shock is present are identified. Due to the numerical approximation associated with the computation of the derivatives, the second derivative may not be exactly equal to zero at many points where a shock is present. Therefore, a threshold value must be chosen for this purpose. Some of the problems with this technique are discussed in [8]. According to [8], this technique would also detect other features such as expansion waves when the pressure gradient is in opposite direction of the velocity ($\nabla P \cdot \vec{V} < 0$). In [7], three quantities called feature detectors are defined for a two-dimensional case to represent expansions and compressions in the flow direction and shear gradients normal to the flow. The feature detectors used are the negative, positive, and normal components of density gradient with respect to the local velocity vector. A threshold value is chosen for each of the detectors. The feature detectors are evaluated at all the nodes and if the value of the detector is less than the corresponding threshold value, it is zeroed. A given node is said to be a part of a relevant feature if any of its feature detectors are non-zero. The set of points that originate from the same feature detector and are physically connected to one another form a group. Each group is treated independently and tests are conducted to identify the group as a shock. This technique involves tracing of lines similar to streamline tracing and choosing three threshold values. A method based on this technique is used to identify the shocks in [16], where the nodes identified as a part of shock are further segmented to form regions of interest. A wavelet analysis is performed on each region of interest to produce a denoised and significantly localized description of the feature. Image processing techniques are used to extract the exact shape of the shock.

Another method using the dynamics of the flow is described in [8]. This technique uses the Mach number normal to the shock, i.e., the Mach number based on the velocity component normal to the shock. The normal Mach number is taken to be the Mach number in the direction of the local pressure gradient vector, as the local pressure gradient is assumed to be aligned with the shock surface normal. In a two dimensional case, contours of normal Mach number are created and the $M_n = 1$ curve forms a boundary for the shock. For a three-dimensional case, an iso-surface of $M_n = 1$ is used to visualize the shock. A correction term for a transient shock is also suggested in the paper. This technique may produce better results than the technique described in [6], as it does not identify other features such as expansions and it does not involve tracing of lines as in [7]. The algorithm described below borrows heavily from this technique.

3.1.1 Stationary Shocks

Consider the oblique shock wave shown in figure 3.1. The upstream flow, region 1, is horizontal with a Mach number $M_1 > 1$ and the downstream flow, region 2, is inclined upward at an angle ϕ with a Mach number $M_2 < M_1$. The Mach number components normal to the shock are $M_{n1} > 1$ and $M_{n2} < 1$ for the upstream and downstream flows respectively. The technique implemented is based on the pressure gradient and the normal Mach number. It is known from the dynamics of fluid flow that the velocity component normal to the shock changes from supersonic to subsonic across a shock. It is assumed that the local pressure gradient vector is in the direction of the normal to the shock. Thus, the normal Mach number is the Mach number component that is aligned with the local pressure gradient. It is also known that the shock is compressive in nature. It can be deduced from [7] that the scalar dot product of the local pressure gradient and local velocity vector can be used to identify compression or expansion at a node. If this product is greater than zero, the node is assumed to be located in a region of compression, otherwise it is assumed to be in a region of expansion. The normal Mach number can be calculated using

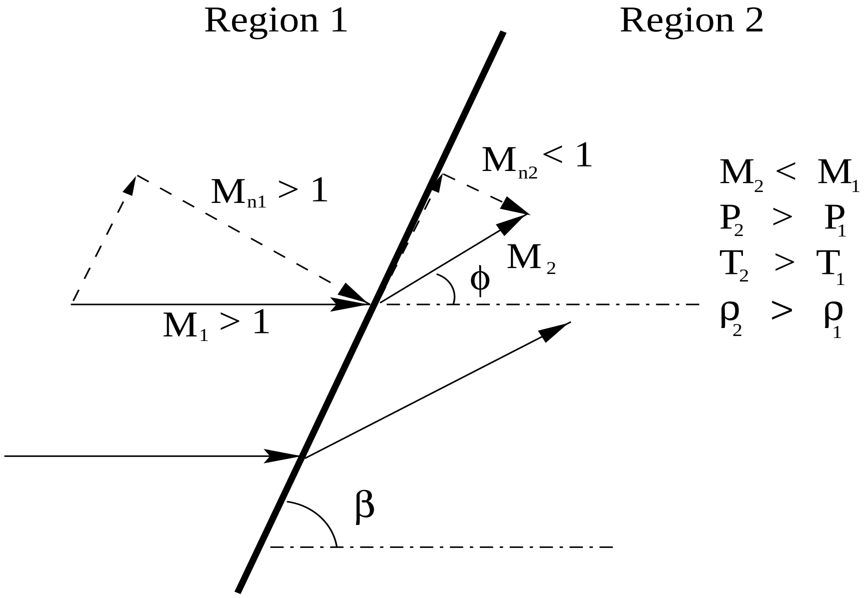


Figure 3.1: An oblique shock wave

equations 3.1, 3.2, and 3.3 as shown.

$$V_n = \vec{V} \cdot (\frac{\nabla P}{|\nabla P|}) \tag{3.1}$$

$$a = \sqrt{\frac{\gamma P}{\rho}} \tag{3.2}$$

$$M_n = \frac{V_n}{a} = \frac{\vec{V}}{a} \cdot \frac{\nabla P}{|\nabla P|} = \vec{M} \cdot \frac{\nabla P}{|\nabla P|} \tag{3.3}$$

The nodes where the normal Mach number changes from greater than one to less than one in the direction of the flow are included in the set of points representing a shock. The implementation of this technique is described in the following paragraph.

Consider a two-dimensional curvilinear grid as in figure 3.2. If the scalar dot product of $\vec{r}$ and $\vec{V}$ is greater than zero, where $\vec{r}$ is the position vector of the neighboring node with respect to node 5 and $\vec{V}$ is the local velocity vector at node 5, then the neighboring node is taken to be in the direction of the local flow. If the normal Mach number at node 5 is greater than one and if any of its neighbors are in the direction of the flow and have a normal Mach number less than

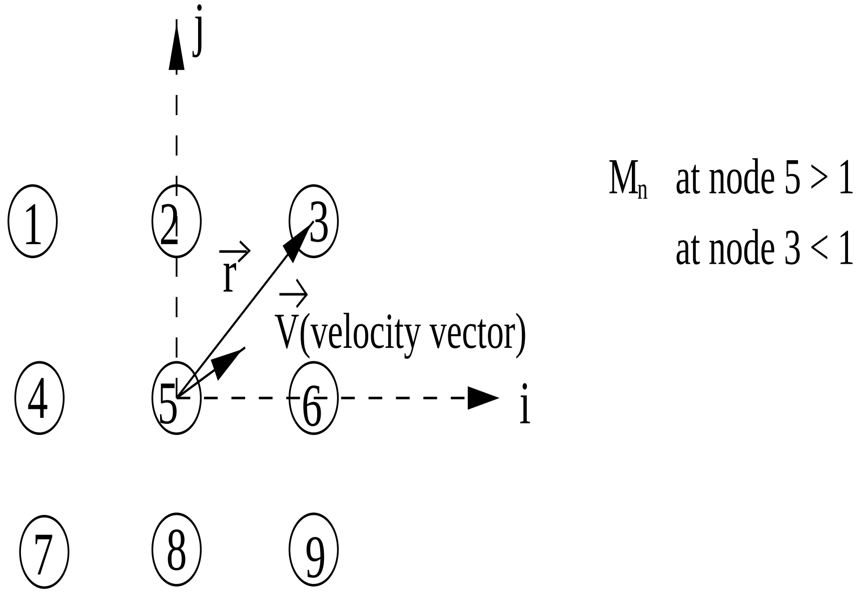


Figure 3.2: Grid nodes in two dimensions

one then node 5 and that neighbor are associated with the region of a shock. It is assumed that the normal Mach number changes from greater than one to less than one somewhere between these nodes.

There may be small disturbances in the pressure gradient due to numerically induced Mach waves. To avoid detection of these small disturbances in pressure as part of a shock, a filtering based on pressure gradient is applied. If the pressure gradient at a node is less than a threshold value, the node is discarded. For nondimensional pressure gradients, the threshold value is chosen to be 0.1 as in FAST[17] and PLOT3D[18]. The same technique is applied to all the nodes in the field and the shock is identified. A similar technique is applied for a three-dimensional case. Instead of considering the eight neighboring nodes, all the twenty six possible neighbors are taken into account.

Once the nodes are identified, a normalized scalar value denoted by W_{shock} is assigned to each node to represent the strength of the shock at that node. The value assigned for a node is given by the equation 3.4. The term $|\nabla P|_{max-shock}$ is the maximum magnitude of the pressure gradient in the region identified as shock.

$$W_{shock} = \frac{|\nabla P|}{|\nabla P|_{max-shock}} \tag{3.4}$$

3.1.2 Transient Shocks

The assumptions used to detect a steady or a stationary shock are not applicable to moving or transient shock. However, if the observer is traveling with the shock and all the velocities are measured with respect to this translating frame of reference, the technique used for steady shock detection can be used. As the velocity of the shock is not known initially, the velocity field with respect to the required reference frame cannot be obtained. The Lagrangian derivative of pressure, given by equation 3.5, provides the time derivative in the appropriate reference frame.

$$\frac{DP}{Dt} = \frac{\partial P}{\partial t} + \vec{V} \cdot \nabla P \tag{3.5}$$

Division by $(|\nabla P|a)$ yields,

$$\frac{1}{|\nabla P|}\frac{1}{a}\frac{DP}{Dt} = \frac{1}{|\nabla P|}\frac{1}{a}\frac{\partial P}{\partial t} + \vec{M} \cdot \frac{\nabla P}{|\nabla P|}. \tag{3.6}$$

Note that the expression on the right side of the equation 3.6 is same as equation 3.3 with the addition of an unsteady term.

Since the solution field may easily be available at only a single time step, the time derivative of pressure must be calculated indirectly. Thus, the time derivative of pressure is approximated using known variables as done in [8]. For this purpose, an isentropic flow is assumed across a shock. Equations 3.7 and 3.8 state the isentropic flow condition and the conservation of mass.

$$\frac{\partial P}{\partial t} = a^2 \frac{\partial \rho}{\partial t} \tag{3.7}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{q}) = 0 \tag{3.8}$$

Equation 3.9 is obtained from equations 3.7 and 3.8.

$$\frac{\partial P}{\partial t} = -a^2 \nabla \cdot (\rho \vec{q}) \tag{3.9}$$

From equations 3.6 and 3.9, we obtain,

$$\frac{1}{|\nabla P|}\frac{1}{a}\frac{DP}{Dt} = -a\frac{1}{|\nabla P|}\nabla \cdot (\rho \vec{q}) + \vec{M} \cdot \frac{\nabla P}{|\nabla P|}. \tag{3.10}$$

However, it is well known that the flow is not isentropic across a shock. Another approximation term can be obtained from the laws of conservation of mass and momentum [8]. Based on this assumption, equations 3.11 and 3.12 are obtained.

$$\frac{dP}{dt} = (\gamma - 1)[-\nabla \cdot (\rho \vec{q} E) - \vec{q} \cdot (\nabla P + \nabla \rho \vec{q} \vec{q}) - q^2 \nabla \cdot (\rho \vec{q})] \tag{3.11}$$

$$\begin{aligned}\frac{1}{|\nabla P|}\frac{1}{a}\frac{DP}{Dt} = & \frac{1}{|\nabla P|}\frac{1}{a}(\gamma-1)[-\nabla\cdot(\rho\vec{q}E) - \vec{q}\cdot(\nabla P + \nabla\rho\vec{q}\vec{q}) - q^2\nabla\cdot(\rho\vec{q})] \\ & + \frac{\vec{M}\cdot\nabla P}{|\nabla P|}\end{aligned} \tag{3.12}$$

The term on the right side of the equation 3.12 is evaluated at all the nodes. The nodes where the value of this term changes from greater than one to less than one can be identified as a part of a moving shock. This is similar to the steady shock detection technique where the normal Mach number is considered instead of this term. The same filtering technique based on pressure gradient is used. The nodes identified are assigned a scalar value given by equation 3.4 to represent the strength of the shock at that node.

The algorithm for shock detection can be summed up as follows:

1. If the flow is steady, calculate the normal Mach number given by equation 3.3 at all nodes. If it is unsteady, calculate the term given by equation 3.12 at every node.

2. Identify all the nodes where the value of the term calculated in step 1 changes from greater than one to a value less than one in the direction of the velocity vector at that point (Figure 3.2).

3. Calculate the maximum value of $|\nabla P|$ at the nodes identified in step 2.

4. Assign a scalar value W_{shock} given by equation 3.4 to these nodes.

3.2 Expansion Region

Regions of expanding flow occur naturally in many flows. A special case of an expanding region, an expansion fan, is a continuous expansion region formed by an infinite number of Mach waves. Figure 3.3 shows a typical expansion fan. The Mach number increases while the pressure, density, and temperature decrease through an expansion fan. In contrast to a shock, there are no discontinuities in the flow variables. The streamlines are continuous and the flow is isentropic across an expansion fan. The expansion fan is bounded by two Mach waves - the forward and the rear-ward Mach waves. Expansion fans are extensively described in [15].

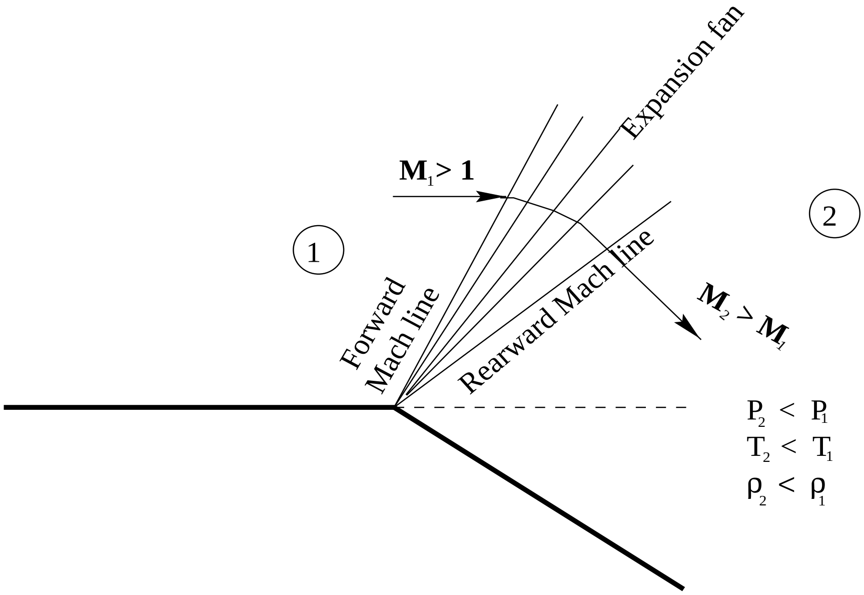


Figure 3.3: Expansion fan

As mentioned earlier, the scalar dot product of the pressure gradient and the velocity vector gives information about the flow being compressive or expansive. If the value of $\nabla P \cdot \vec{V}$ is less than zero, the fluid is flowing in the direction opoosite to that of increasing pressure, and is thus expanding. The nodes where the flow is supersonic and expanding can be identified as an expansion region. Thus, if equations 3.13 and 3.14 are satisfied at a node, the node is associated with an expansion region and a normalized weight $W_{expansion}$ given by equation 3.15 is assigned to it.

$$\nabla P \cdot \vec{V} < 0 \tag{3.13}$$

$$M = \frac{|\vec{V}|}{a} > 1 \tag{3.14}$$

$$W_{expansion} = \frac{|\nabla P|}{|\nabla P|_{max-expan}} \tag{3.15}$$

The term $|\nabla P|_{max-expan}$ is the maximum magnitude of the pressure gradient of the points identified as an expansion region.

The algorithm for identifying an expansion region can be summed up as follows:

1. Calculate the Mach number and value of $\nabla P \cdot \vec{V}$.

2. If the value of M is greater than one and the value of $\nabla P \cdot \vec{V}$ is less than zero at a node, then that point is identified as a part of expansion region.

3. The nodes identified are assigned a value $W_{expansion}$.

3.3 Separated Flows

Flow separation and vortical flow are important features in fluid flow that have received considerable attention. A large number of methods for detecting these features have been published. A new technique to detect regions of separated flow, described in the following sections, uses the location of the vortex cores and separation and attachment lines in the flow field. Hence, before discussing the technique used for detecting the regions of separated flow, a discussion about the various methods that can be used for detecting vortex cores, separation lines, and attachment lines is presented.

3.3.1 Vortices

Informally, a vortex is defined as a swirling flow pattern around a central point. In many reported publications, a vortex is defined as a region that satisfies a particular set of conditions. Various definitions for a vortex are presented in [5], [9], [10], [11], [19], [20], and [21]. Excellent reviews of the existing methods are given in [2] and [22]. Each of the techniques is either a region-type or a line-type feature detection algorithm. These can be re-classified as a local-feature, a global-feature, or a set-feature based algorithms as described earlier.

The method described in [11] can be classified as a region-type, local-feature based algorithm. This technique is based on the local velocity gradient tensor. It is assumed that a point is located in the region of swirling flow if the eigenvalues of the velocity gradient tensor at that point are complex. The same technique is applied to all the points in the flow to obtain the regions of swirl. A parameter called "intrinsic swirl parameter" (τ) is defined at a point to estimate the tendency of that point to stay in a region of the swirl. The value of the parameter is given by equation 3.16,

$$\tau = \frac{t_{conv}}{t_{orbit}} \tag{3.16}$$

The quantity, t_{orbit}, given by equation 3.17, is the period of motion of the spiral path of a swirling particle,

$$t_{orbit} = \frac{2\pi}{|Im(\lambda_{1,2})|} \tag{3.17}$$

where $Im(\lambda_{1,2})$ is the imaginary part of the complex conjugate pair of eigenvalues. The quantity, t_{conv} is obtained using equation 3.18

$$t_{conv} = \frac{L}{V_{conv}} \tag{3.18}$$

where

$$V_{conv}^2 = |\vec{V}|^2 - (\vec{n} \cdot \vec{V})^2. \tag{3.19}$$

L is a characteristic length associated with the size of the region of complex eigenvalues and V_{conv}, given by equation 3.19, is the velocity of a point in a plane whose normal is parallel to the real eigenvector ($\vec{n}$). Combining equations 3.16, 3.17, and 3.18, we get

$$\tau = \frac{|Im(\lambda_{1,2})|L}{2\pi V_{conv}}. \tag{3.20}$$

This parameter is assigned to every point identified to represent the strength of swirl at that point. According to [11], the swirl has a nonzero value in the regions containing vortices and attains a local maximum in the core region.

The method described in [10] can be classified as a local-feature, line-type detection algorithm. The algorithm is based on the critical point theory. Critical points are the points where the streamline slope is indeterminate because the velocity is zero. According to this theory, the eigenvalues and eigenvectors of the matrix $\frac{\partial u_i}{\partial x_j}$, define the flow pattern about a critical point. If the eigenvalues are complex, the flow is assumed to spiral about the eigenvector corresponding to the real eigenvalue in a plane normal to this eigenvector. For swirling flows whose centers are not critical points, a parameter called reduced velocity is defined. The reduced velocity is the velocity obtained after subtracting the velocity in the direction of the eigenvector corresponding to the real eigenvalue from the velocity at that point. The reduced velocity is given by the equation 3.21 where $\vec{n}$ is a unit vector parallel to the real eigenvector.

$$\vec{V}_{reduced} = \vec{V} - \left(\vec{V} \cdot \vec{n}\right) \vec{n} \tag{3.21}$$

If the reduced velocity is zero at a point, that point is assumed to be at the center of the swirling flow. A line segment is formed by connecting all such identified points to form a vortex core. This method is similar to the one suggested in [11]. The term V_{conv}, considered in [11], is the magnitude of $\vec{V}_{reduced}$ used here. This method assumes a linear vector field. A correction term is provided for curved or bent vortices in [9].

A region-type, local-feature based detection technique is described in [20]. It extracts the region where the equation 3.22 is satisfied and considers this region to be a vortex.

$$|\frac{\vec{V} \cdot \nabla \vec{V}}{|\vec{V}||\nabla \vec{V}|}| = 1 \tag{3.22}$$

In equation 3.22, $\vec{V}$ is the velocity vector and $\nabla \vec{V}$ is the velocity gradient tensor. It assumes that a vortex core occurs when the velocity vector is parallel to the vorticity vector. In [19], it is assumed that a vortex core is a streamline of vorticity with the pressure being minimum at the core. This is a global, line-type feature detection algorithm. A region-based, local-feature based technique is described in [21]. This approach is based on the symmetric deformation tensor $\bar{\bar{S}} = \frac{1}{2}\left(\nabla \vec{V} + \left(\nabla \vec{V}\right)^T\right)$ and the antisymmetric spin tensor $\bar{\bar{\Omega}} = \frac{1}{2}\left(\nabla \vec{V} - \left(\nabla \vec{V}\right)^T\right)$. According to [21], if the second largest eigenvalue of $\bar{\bar{S}}^2 + \bar{\bar{\Omega}}^2$ is negative or if the second invariant of $\frac{1}{2}\left(|\bar{\bar{\Omega}}|^2 - |\bar{\bar{S}}|^2\right)$ is positive, that point is contained within a vortex.

The method proposed in [5] is a set-feature method based on winding angle. The winding angle is a measure of the rotation of the streamline with respect to a point not belonging to the streamline. The winding angle of a streamline swirling around a center is greater than 360 degrees. It is assumed for a two-dimensional case that streamlines spiral about the center. For a grid node, if a streamline starting from any of its neighbors has a winding angle greater than 360 degrees, then that node is assumed to be located in the core. In a three dimensional case, it can be assumed that the streamlines spiral about the center in a plane normal to the vortex core. One of the techniques suggested in [10] or [11] can be used to obtain the location of the vortex cores. The method described in [11] is used in [1] and [23] for detecting regions of swirl and vortex core.

3.3.2 Separation and Attachment lines

The various methods that can be used to identify the separation and attachment lines are discussed in this section. Flow separation or attachment occurs when a flow abruptly leaves or returns to a solid body respectively. The line along which the flow leaves the body is called the separation line and the line along which flow attaches is called the attachment line. The conventional method for visualizing flow separation is to construct integral curves such as streak lines or streamlines using a large number of seeds placed near the surface. These curves converge along a separation line and diverge along an attachment line as shown in figures 3.4 and 3.5. Automated methods for the detection of separation and attachment lines, presented in [3], [12], and [13] are based on critical point theory. A brief description of critical point theory, as described in [3], is presented here.

A tangent curve of a vector field is a curve for which the tangent vector at any point along the curve is parallel to the vector field at that point. In the case of fluid flows, the vector field is the velocity field. The tangent curves near a critical point define the behavior of the flow around that point. As noted previously, the nature of the flow around a critical point is defined by the eigenvalues and eigenvectors of the Jacobian matrix of the velocity vector, $\frac{\partial u_i}{\partial x_j}$, at that critical point. A critical point can be classified based on the eigenvalues. Table 3.1 lists the possibilities for a two-dimensional case. A detailed description of the critical point theory and vector field topology is presented in [24].

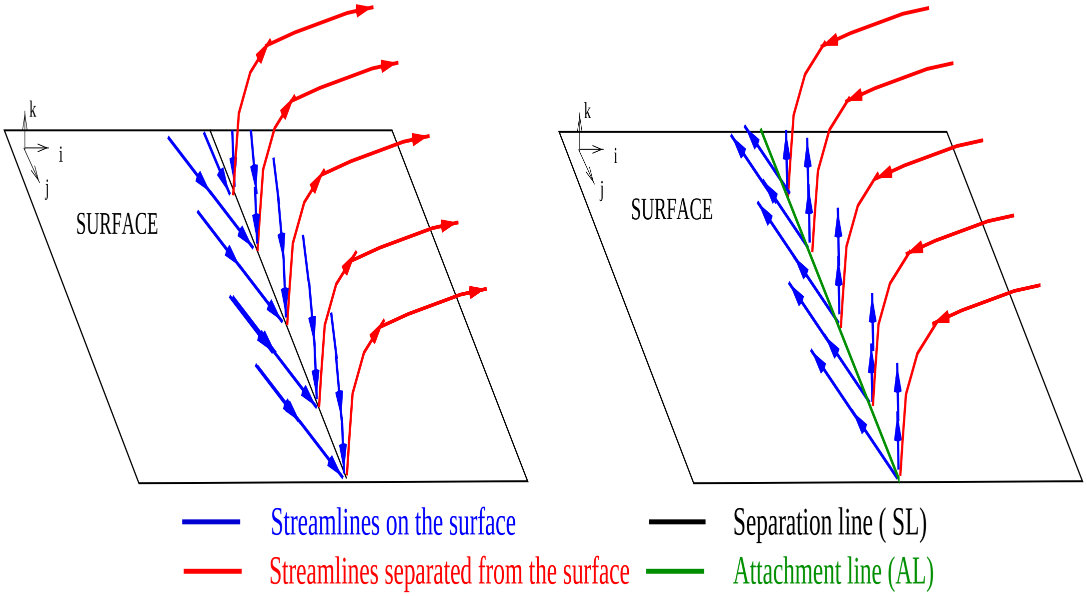

k
i
j
SURFACE
k
i
j
SURFACE
Streamlines on the surface
Streamlines separated from the surface
Separation line (SL)
Attachment line (AL)

Figure 3.4: Streamlines near separation and attachment lines

Type of point	Eigenvalues I_1, I_2 - imaginary part R_1, R_2 - real part	
Attracting node	$R_1, R_2 < 0$	$I_1 = I_2 = 0$
Attracting focus	$R_1 = R_2 < 0$	$I_1 = -I_2 <> 0$
Repelling node	$R_1, R_2 > 0$	$I_1 = I_2 = 0$
Repelling focus	$R_1 = R_2 > 0$	$I_1 = -I_2 <> 0$
Center	$R_1 = R_2 = 0$	$I_1 = -I_2 <> 0$
Saddle	$R_1 < 0, R_2 > 0$	$I_1 = I_2 = 0$

Table 3.1: Classification criteria for critical points

According to [3], the saddle point and the attachment and detachment nodes are the only points where unique tangent curves end on the point itself. Tangent curves are drawn from all the critical points in the flow field. This generates one curve for each attachment and detachment node and four curves for each saddle point. Based on this classification, the attachment and detachment nodes are identified. The above algorithm is extended to three-dimensional case in [12]. The vector field of a plane one computational unit above the surface of the body is projected onto the surface. Tangent curves are generated from the saddle or the node-type critical points in the real eigenvector directions. These are classified as separation or attachment lines based on the sign of the eigenvalues. This approach assumes that the separation and attachment lines start at a saddle point or a node and end at another saddle point or a node. This type of separation is called a closed separation. A technique to detect both open and closed types of separations is presented in [13]. This technique is based on the critical point theory and assumes a linear vector field. This technique is chosen because it considers both types of separations. The implementation is discussed in the following paragraph.

It is assumed that the computational domain consists of triangles and the vector field is available at every vertex. The velocity field of a plane one computational unit above the surface in computational space is considered. The coordinates and the velocity vectors of the triangles are transformed from a three-dimensional to a two-dimensional basis. Based on the transformed data, a linear vector field is constructed for each of the triangle using equation 3.23, where (x, y) are the cartesian coordinates and $(\dot{x}, \dot{y})$ is the tangential velocity or shear stress vector. The coefficients a_1, a_2, b_1, b_2, c_1, and c_2 are constants. These constants are computed by substituting the coordinates and the velocity vectors of each vertex of the triangle in the equation 3.24. In equation 3.24, (x_i, y_i) and $(\dot{x}_i, \dot{y}_i)$ are the position coordinates and the velocity of the ith vertex of a triangle.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 & c_1 \\ b_2 & c_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.23}$$

$$\begin{gathered} \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_2 & y_2 \end{pmatrix}^{-1} \begin{pmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{pmatrix} \\ and \\ \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_2 & y_2 \end{pmatrix}^{-1} \begin{pmatrix} \dot{y_1} \\ \dot{y_2} \\ \dot{y_3} \end{pmatrix} \end{gathered} \tag{3.24}$$

If the discriminant of the Jacobian matrix is greater than zero and both the eigenvalues are non-zero, the eigenvectors of the Jacobian are calculated. The critical point for the triangle is calculated and the triangle is projected onto a phase plane using equations 3.25 and 3.26. The coordinates of the triangle are transformed to a canonical coordinate system by projecting onto the phase plane. In this coordinate system, the eigenvectors are perpendicular to each other. Substituting $\dot{x} = 0$ and $\dot{y} = 0$ in equation 3.23 for a critical point, we obtain,

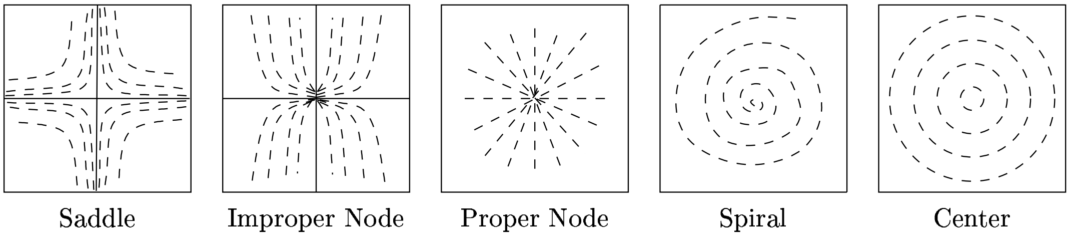
$$x_{cp} = \frac{a_2c_1 - a_1c_2}{b_1c_2 - b_2c_1} \quad y_{cp} = \frac{a_1b_2 - a_2b_1}{b_1c_2 - b_2c_1} \ . \tag{3.25}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \epsilon_1 & \eta_1 \\ \epsilon_2 & \eta_2 \end{pmatrix}^{-1} \begin{pmatrix} x - x_{cp} \\ y - y_{cp} \end{pmatrix} \tag{3.26}$$

The left side of the equation 3.26, x and y, are the position coordinates in the phase plane. $(\epsilon_1, \epsilon_2)^T$ and $(\eta_1, \eta_2)^T$ are the eigenvectors of the Jacobian. The phase portrait of the triangle can be classified as a saddle point, a repelling node, or an attracting node based on the eigenvalues of the Jacobian. Table 3.2 lists the conditions for each kind. Figure 3.5 shows the different types of phase portraits that arise in linear vector fields. The phase portrait of a proper node, a spiral, and a center do not have limiting streamlines as in the case of a saddle point, a repelling, or an attracting node. A triangle with either one eigenvalue equal to zero or with imaginary eigenvalues gives rise to a center, spiral, or a proper node which does not have limiting streamlines. Since

Type of Phase Portrait	Eigenvalues - e_1, e_2
Saddle	$e_1 < 0 < e_2$
Repelling Node	$0 < e_1 < e_2$
Attracting Node	$e_1 < e_2 < 0$

Table 3.2: Conditions for different phase portraits


Saddle

Improper Node

Proper Node

Spiral

Center

Figure 3.5: Different types of phase portraits

a separation line is a limiting streamline on which adjacent streamlines converge, the Jacobians with either one eigenvalue equal to zero or with imaginary eigenvalues are not considered. If the phase portrait contains a saddle point or a repelling node and the line x = 0 intersects the triangle, the attachment line is assumed to pass through the triangle. If the phase portrait contains a saddle point or an attracting node and the line y = 0 intersects the triangle, the separation line is assumed to pass through the triangle. This technique is applied to all the triangles on the surface and the separation and attachment lines are identified.

The algorithm can be summed up as the following steps:

1. If a structured grid is provided, triangulate the grid on the surface. Project the velocity vectors of a plane one computational unit away from the surface onto the surface.

2. Transform the postion coordinates and velocity vectors of the triangle vertices from a three-dimensional basis to a two-dimensional basis.

3. Calculate the coefficients in equation 3.23 using equation 3.24 and assemble the Jacobian.

4. Calculate the determinant of the Jacobian. Stop if the discriminant is negative.

5. Evaluate the eigenvalues of this Jacobian. Stop processing if one of the eigenvalues is zero.

6. Calculate the critical point using equation 3.25.

7. Project the triangle onto the phase plane using equation 3.26.

8. Using Table 3.1, determine the phase portrait.

9. If the phase portrait conatines a saddle or a repelling node and the line x = 0 passes through the triangle, attachment line is assumed to pass through the triangle.

10. If the phase portrait is a saddle or an attracting node and line y = 0 passes through the triangle, it is assumed that a separation line passes through the triangle.

3.3.3 Separation Layer

In this section, the new method for detecting a separation layer is discussed. The vortex cores, separation lines (SLs), and attachment lines (ALs) discussed in the previous sections are line-type features. To identify separation as a region-type feature, a new technique for detecting a separation layer is required for the present work. The separation layer can be defined as the boundary of a separated flow formed by streamlines that separate from the surface of the body. The method is designed to identify the separation layer assuming that a vortex core is present for every SL/AL pair. Figure 3.6 shows the cross-section of such a three-dimensional flow separation. For convenience, the surface of the body is assumed to be k = 0 plane or x-y plane in computational space and the directions in the surface are the i and j axes. In cases where the separating streamlines reaches the surface at an AL as in figure 3.6, it is sufficient to identify the surface traced by the streamlines starting from different points on the SL. The above condition may not be true in a cross-flow separation with high axial velocity because the separating streamlines may exit the boundaries of the flow field before reaching the surface. It is

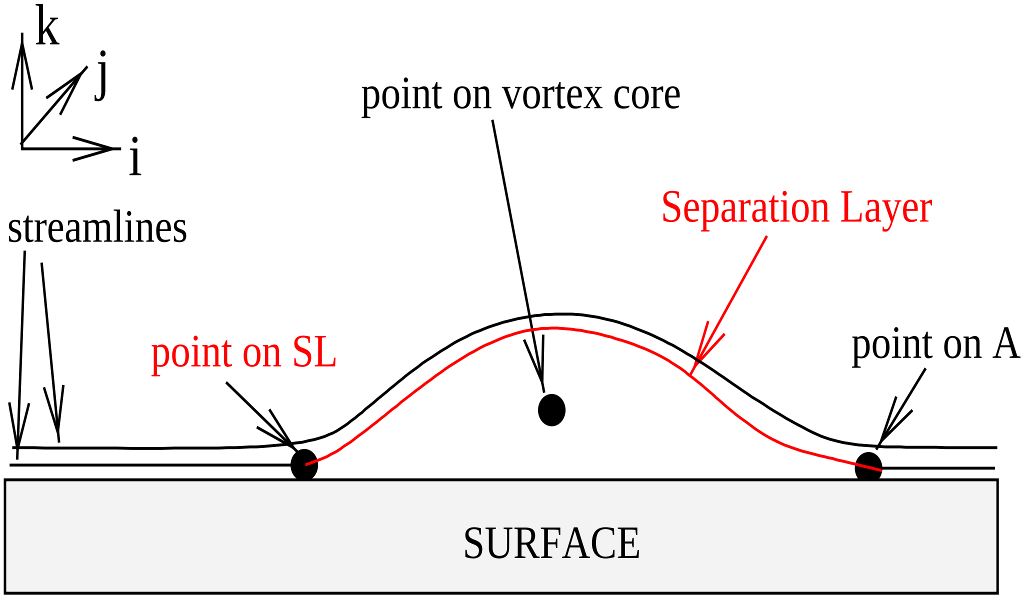


Figure 3.6: Cross section of a three-dimensional flow separation.

reiterated that this technique is designed to only identify those separated flows that are described by the schematic shown in figure 3.6.

One possible technique, based on the direction of the flow at the SL and AL, attempts to pair an SL with an AL initially and then identify the region in-between these lines. This is explained using figure 3.7 which shows a simple flow separation with only one SL/AL pair. A streamline can be integrated starting from a point on the SL to identify the direction of the flow leaving the SL. A streamline can be traced backwards from a point on an AL to identify the direction from which the streamlines approach the AL. If the direction of the separating streamlines is same as the direction of the attaching streamlines as in figure 3.7, the SL and the AL are paired up. In cases where the velocity component normal to the surface of a point on an AL is positive due to computational inaccuracies, this technique fails because the streamline traced backwards from that point ends on the surface. This technique can also cause problems if there is more than one SL/AL pair. Hence, a new technique to detect separation layer based on the vortex cores is proposed. A definition of vortex as given in [25] is as follows:

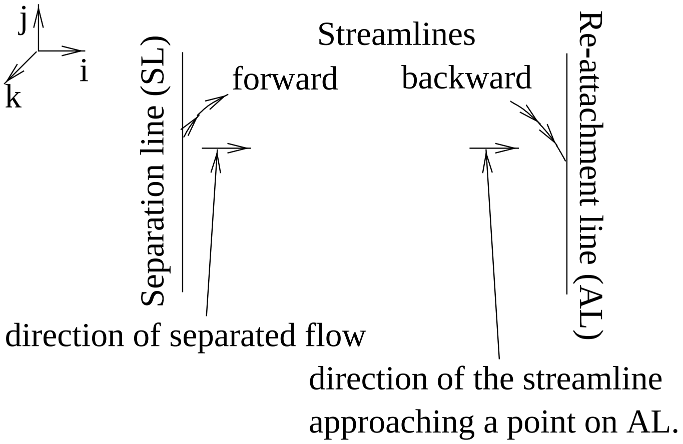

j
i
k
Separation line (SL)
Streamlines
forward
backward
Re-attachment line (AL)
direction of separated flow
direction of the streamline
approaching a point on AL.

Figure 3.7: A simple flow separation.

> A vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern, when viewed from a reference frame moving with the center of the vortex.

From [12], it is deduced that an SL/AL pair is associated with a vortex core. In [5], it is stated that the winding angle of a streamline projected onto a plane normal to the vortex core is greater than 360 degrees. Hence, it can be assumed that a streamline starting from a point on an SL restricted to a plane normal to the associated vortex core re-attaches to the surface at a point on the corresponding AL. This is shown in figure 3.8. If the vortex core considered is contained in the region traced by the above streamline, then the SL and the AL are paired. A number of streamlines starting from different points on the SL in this local restricted plane can be used to obtain the surface traced by the separating streamlines. The implementation details of a technique based on this approach are explained in the following paragraphs.

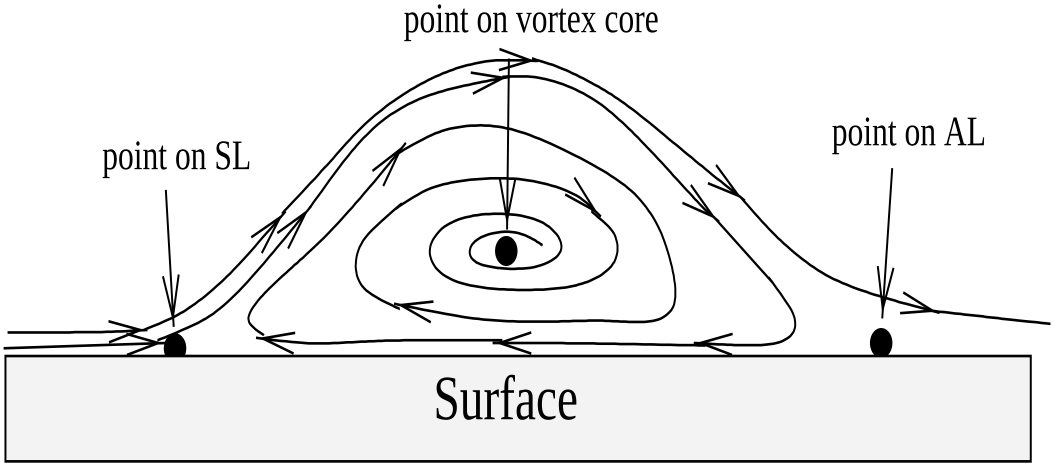


Figure 3.8: Typical streamlines in a plane normal to the vortex core

The SLs and ALs are identified using the technique discussed in the previous section [13]. If there exists more than one surface in the flow field, the method is applied to each surface. The locations of the vortex cores are obtained using one of the line-type feature detection algorithms discussed earlier. As previously stated, the SLs and ALs are sets of points on a surface. Due to the no-slip condition, the velocity vector of a point on the surface is zero. Hence, a streamline starting from a point on an SL actually refers to a streamline starting from a point one computational unit away from the surface. For example, if the point chosen on the SL in computational coordinate system is $(i_1, j_1, 0)$, the streamline is actually integrated from the point $(i_1, j_1, 1)$.

A streamline initiated from a point on an SL is restricted to the local plane normal to one of the vortex core using the velocity component of the point in that plane. Consider figure 3.9 which shows an SL and a vortex core. $\vec{s}$ is the position vector, with respect to origin, of the point on the SL from where the streamline is drawn. Let $\vec{c}$ be the position vector of a point on the vortex core and $\vec{n}_v$ be the local tangent to the vortex core at $\vec{c}$. Since the line joining $\vec{s}$ and $\vec{c}$ should be normal to $\vec{n}_v$, $\vec{c}$ should be chosen such that $|\vec{s} - \vec{c}|$ is a minimum. The local tangent $(\vec{n}_v)$ is considered to be normal to the required plane. The velocity at the point $\vec{s}$ in the required plane is obtained by subtracting the velocity component in the direction of $\vec{n}_v$ from the velocity of the point. The reduced velocity $(\vec{V}_r)$ for any velocity vector $(\vec{V}_i)$ and local tangent vector $(\vec{n}_v)$ can be obtained using equation 3.27.

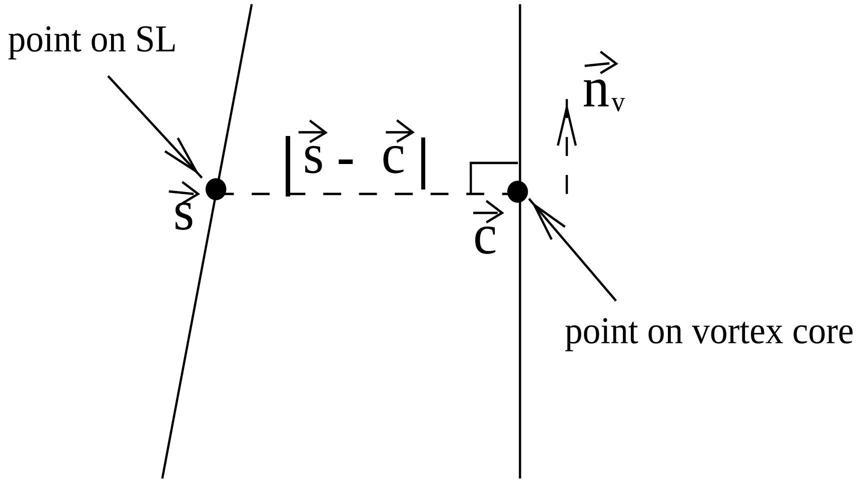


Figure 3.9: The geometry of an SL and a vortex core.

$$\vec{V}_r = \vec{V}_i - \vec{n}_v \left(\vec{n}_v \cdot \vec{V}_i\right) \tag{3.27}$$

It should be noted that this reduced velocity is same as the reduced velocity described in [10]. $\vec{V}_i$ is the velocity of a point in computational space and is obtained using tri-linear interpolation. The streamline integration is done using a fourth-order Runga-Kutta scheme.

Figure 3.8 illustrates a case in which a separated streamline reaches the surface. However, there are other cases where a separated streamline does not reach the surface as shown in figure

3.10. This type of behavior may be due to the computational inaccuracy involved in dealing with

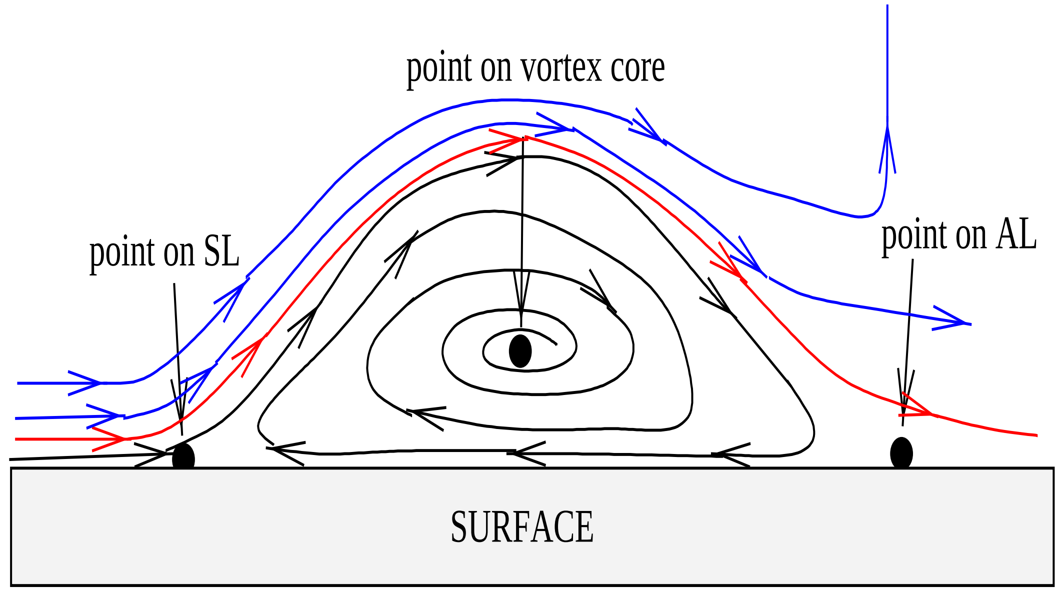


Figure 3.10: Various paths a streamline can take.

the small velocity magnitudes near the surface which are usually small numbers. This may also arise due to the location of the point chosen on the SL. Consider figure 3.10 where streamlines are integrated from different points near an SL. In some cases, the streamlines re-attach to the surface, in some cases, the streamline moves into a vortex and, in some cases, it leaves the flow field before re-attaching to the surface. Hence, depending on the point from where the streamline is initiated, the streamline could take one of the paths shown in 3.11(a), (b), or (c). In figure 3.11(a), the separated streamline reaches a point from where the streamline moves only in the k direction and leaves the flow field. In such cases, the streamline integration is stopped once it moves less than a threshold value in both i and j directions. The last point traced is taken to be the endpoint. In figure 3.11(b), the separated streamline moves in a spiral fashion and ends at a point away from the surface. The streamline integration is stopped if its position does not move more than a threshold value in any direction. The farthest point traced by this streamline from the initial point is considered as the endpoint. The threshold value chosen for these cases is 10^{-6}. In figure 3.11(c), the separated streamline moves in a circular fashion. The streamline

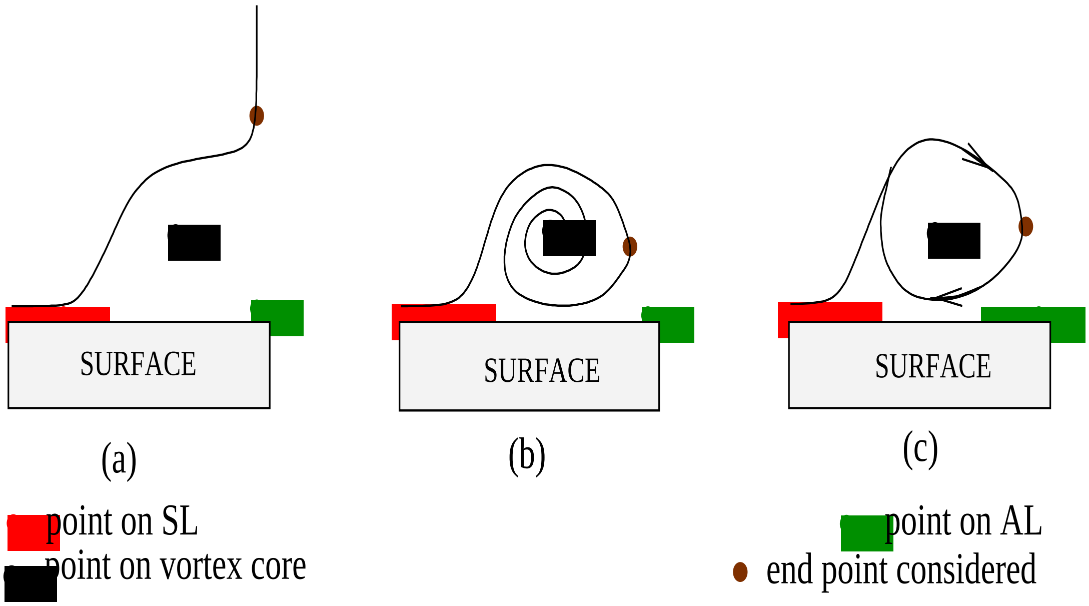


Figure 3.11: Other paths a separated streamline can take.

integration is stopped if it passes through any cell more than 10,000 times. In this case too, the farthest point traced from the starting point is considered as the endpoint. In some cases, the streamline may behave similarly to the one shown in figure 3.8 but may not reach the k = 1 plane. For such cases, the streamline integration is stopped once it reaches the k = 3 plane and has a negative velocity component in the direction normal to the surface. For a typical viscous simulation, the k = 3 surface is very close to the no-slip boundary.

Three points on an SL - first, middle, and end point are considered in the algorithm. A streamline is traced from the first point on the SL in a plane normal to the vortex core considered and the endpoint is obtained. This endpoint is projected onto the surface. If the projected endpoint lies on an AL or within three cells away from a point on an AL, the SL and that particular AL are considered as a possible pair. The bounding box of the region traced by the streamline can be obtained from the maximum and the minimum values of i, j, and k of the points traced by the streamline as shown in figure 3.12. If at least one point of the vortex core lies in this bounding box, then the SL and the AL are paired. If there is more than one surface in the flow field, then the flow may separate from one surface and re-attach on another surface.

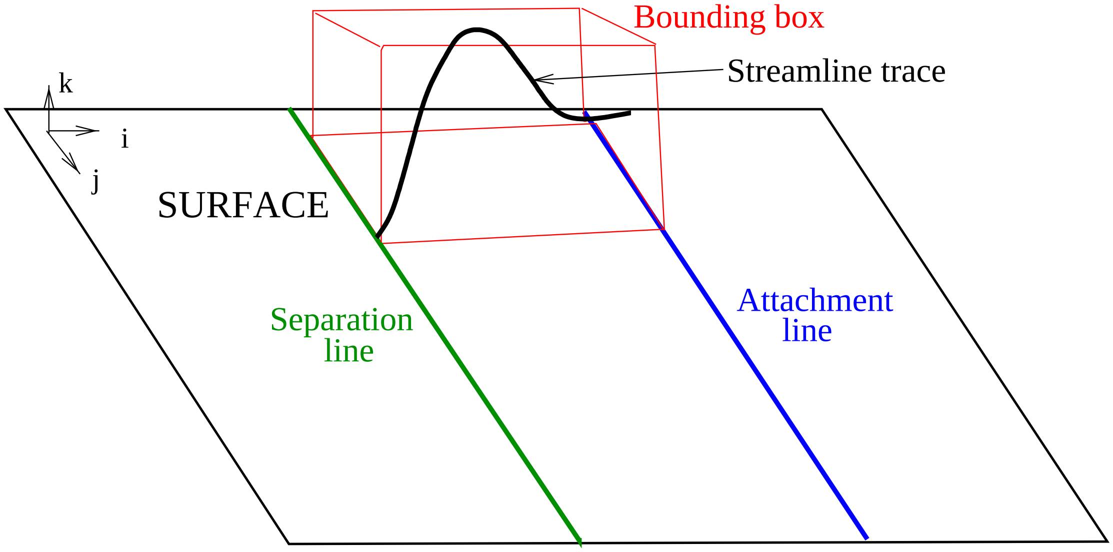


Figure 3.12: The bounding box of a streamline trace.

Hence, the endpoint obtained from the streamline integration is projected onto all the surfaces present. If the streamline from the first point of the SL does not satisfy the above conditions, then the same procedure is applied to the middle and the last point of the SL. If the SL is still not paired, another vortex core is considered. The same procedure is applied to every SL considering all the vortex cores present until the SL is paired with an AL. Once an SL and an AL are paired, a number of points (for example ten), are chosen on the SL. Streamlines are constructed from these points and the points lying in between a streamline and the surface are assigned a scalar value.

To differentiate the separation layers associated with different SL/AL pairs and to identify secondary and tertiary separations, different scalar values can be used for different vortex cores considered. To illustrate this, consider figure 3.13 where there are three SL/AL pairs and three vortex cores. If a vortex core is associated with an SL/AL pair, a scalar value corresponding to that vortex core is assigned to all the nodes lying in between the streamline trace and the surface. Let f_1, f_2, and f_3 be the scalar values corresponding to the three vortex cores. Hence in figure 3.13, when SL_1 is considered with the vortex core 1, f_1 is assigned to the nodes in the regions 1, 2, and 3. When SL_1 is considered with vortex core 2, f_2 is assigned to the nodes in the

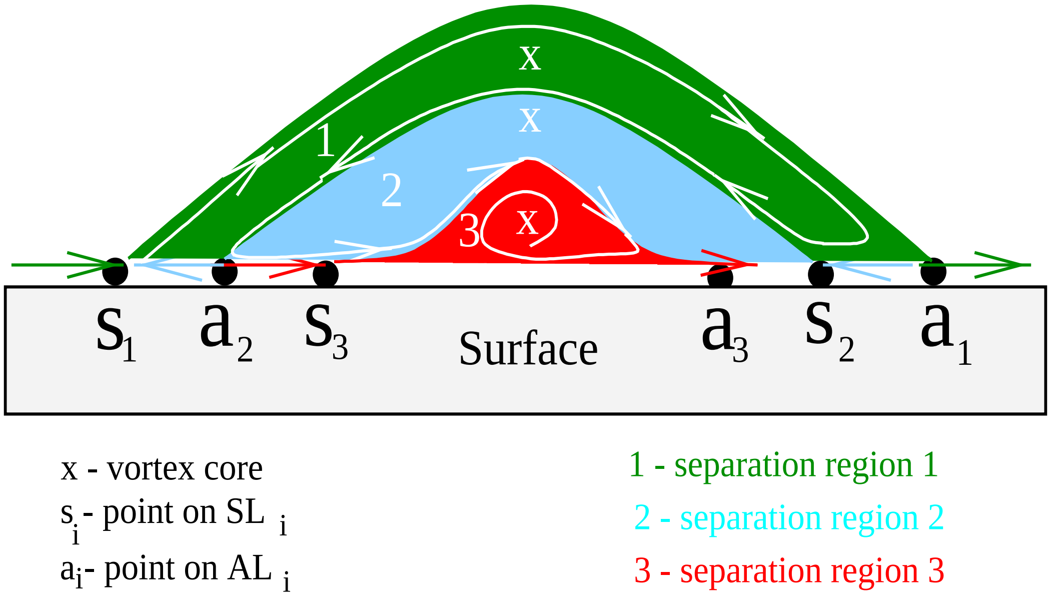


Figure 3.13: The primary, secondary, and tertiary separations.

regions 1, 2, and 3. Similarly, when vortex core 3 is considered, the nodes in region 1, 2, and 3 are assigned f_3. Hence, when all the SLs are considered with every vortex core, different values are assigned to the nodes. Table 3.3 shows the values assigned to the nodes in region 1, 2, and 3 after each SL is considered with every vortex core. It can be observed that the three regions

SL	Region 1			Region 2			Region 3		
	VC 1	VC 2	VC 3	VC 1	VC 2	VC 3	VC 1	VC 2	VC 3
SL$_1$	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3
SL$_2$	-	-	-	-	f_2	f_3	-	f_2	f_3
SL$_3$	-	-	-	-	-	-	-	-	f_3
After three SLs	f_1, f_2, f_3			f_1, f_2, f_2, f_3, f_3			$f_1, f_2, f_2, f_3, f_3, f_3$		
Value assigned	f_1			f_2			f_3		

Table 3.3: The different values assigned to nodes in a tertiary separation. Legend: VC - vortex cores.

can be distinguished based on the values assigned to them. The value f_3 is assigned thrice to the nodes in region 3 and value f_2 is assigned twice to the nodes in region 2. Thus the nodes in different regions can be differentiated and can be assigned a single value. An iso-surface of the values assigned is used to display the separation layer. The strength of the vortex associated with the separation layer can be taken as the strength of that separation layer.

This technique assumes that the location of the vortex cores is known initially. It should be noted that an approximate plane normal to the vortex core considered is sufficient for this technique. The threshold values used for streamline integration and the number of points chosen on an SL for identifying a possible pairup (three points are used here) worked well for the test cases implemented. These values are expected to work well in most cases but different values may be needed in some cases.

The algorithm can be summed up as follows:

1. Identify the vortex cores, SLs, and ALs using the algorithms discussed in previous sections. If there is more than one surface, apply the technique to all the surfaces.

2. Consider a vortex core. Consider three points on the SL - first, middle, and last point.

3. Start a streamline from the first point on the SL restricting it to a plane normal to the vortex core considered.

4. The streamline is restricted to the plane normal to the vortex core by using the reduced velocity given by equation 3.27.

5. Obtain the endpoint of the streamline and project it onto the surface. If the endpoint is on an AL or within three cells away from a point on an AL proceed with the algorithm. Else, go to step 3 with streamline starting from the next point until all the three points have been tried.

6. Obtain the bounding box traced by the streamline as shown in figure 3.12. If at least one point on the vortex core is in this bounding box, pair the SL and the AL. Assign the nodes between the point traced by the streamline and the surface a flag value corresponding to the vortex core considered. If any point of the vortex core does not lie in the bounding box go to step 3. If all the three points on an SL have been tried, go to step 2 and choose a different vortex core.

7. Perform the steps for all the SLs considering every vortex core. In case of a secondary or a tertiary separation, some nodes will be assigned more than one flag value. In such cases, assign the correct value based on the explanation given in the preceding paragraph.

CHAPTER IV

RESULTS AND DISCUSSION

The feature detection algorithms discussed in the previous chapter are tested for a few datasets. The test cases considered are a blunt fin, a finned missile, and two-dimensional single and double ramp cases. A brief description of each of the above cases, before presenting the results obtained, is provided in the following sections.

4.1 Single and Double ramp case

The grids for the two-dimensional single and double ramp cases are shown in figures 4.1 and 4.2 respectively. The flow is from left to right. The free-stream Mach number in both the cases is 4.

The pressure distribution for the single ramp case is shown in figure 4.3. The sharp change in pressure observed in the figure corresponds to a shock. Figure 4.4 shows the normal Mach number distribution. The nodes where the value of normal Mach number changes from greater than one to less than one form a shock. The shock detection algorithm is applied to this case, and the nodes identified as shock nodes are shown in figure 4.5. The nodes identified are assigned W_{shock}, defined by equation 3.4, as shown in figure 4.6. The algorithm for identifying an expansion region is also applied and the result with $W_{expansion}$, defined by equation 3.15, assigned to the nodes is shown in figure 4.7. From figure 4.1, it can be observed that there is no pressure variation corresponding to an expansion here. However, the algorithm identifies the small disturbances or oscillations in pressure behind the shock as expansions. Except for the nodes near the boundary, the weights assigned to the nodes identified are very small. The relatively large values of $W_{expansion}$ occurring at the downstream boundary can be attributed to nonphysical oscillations that occur due to an interaction of the shock with the boundary. The above results are obtained using the pressure gradient filter discussed earlier. Figure 4.8 shows the normal Mach number distribution without using the pressure gradient filter. It can be seen that there are small disturbances away from the shock where the normal Mach number varies.

The pressure distribution in the double ramp case is shown in figure 4.9. There are two sharp changes in pressure that correspond to two intersecting shocks. A variation in pressure can also be observed behind the shock. An expansion region can be expected behind the shock. Thus both shock and expansion detection algorithms are applied to this case. Figures 4.10, 4.11, and 4.12 show the normal Mach number distribution, the shock identified, and normalized shock weight respectively. In this case, the intersecting shocks produce an expansion that reflects off the ramp surface. This expansion and a non-physical expansion at the downstream boundary are detected as shown in figure 4.13. The normal Mach number distribution obtained without using the pressure gradient filter is shown in figure 4.14.

4.2 Expansion Fan

The case shown in figure 4.15 results in an expansion fan. The flow is from the left to right. The upstream flow is supersonic with free-stream Mach number of 4 and the flow accelerates across the field. Hence an expansion fan is expected. The result is shown in figure 4.15.

4.3 Blunt Fin

Figure 4.16 shows the standard blunt fin case. In this dataset, the airflow is over a flat plate with a blunt fin rising from the plate. The free stream flow direction is parallel to the plate and to the flat part of the fin with a free-stream Mach number of 2.95 and a unit Reynolds number of 6.3x10e+7m^{-1}. The flow is assumed to be symmetrical about a plane through the center of the fin, so one half of the geometry (as shown in the figure) is considered. The results of a flow simulation on the blunt fin are presented in [26]. Some of the results shown in [26] are discussed here. In this simulation, the flow is considered to be steady, although in experimental measurements the flow is determined to be inherently unsteady and highly oscillatory. The main features that occur in this case are a lambda shock, a pair of vortices, and separation of flow. A bow shock exists in front of the fin. This causes the boundary layer to separate from the surface ahead of the fin resulting in a separated flow region near the surface and a separation shock ahead of the fin. The separation shock intersects with the bow shock resulting in a supersonic jet which separates the two sub-sonic regions to form a lambda shock. The locations of the vortex cores used in the technique detecting separation layer are obtained from [9]. One vortex is present on the surface and another one is present near the nose of the fin.

Figure 4.17 shows the pressure distribution in the i = 0 plane. Figure 4.18 shows the normal Mach number distribution in this plane. The nodes where the normal Mach number changes value from greater than one to less than one are identified as shock as shown in figure 4.19. Figure 4.20 shows the pressure distribution in a plane parallel to the plate (k = constant plane). The nodes identified as a part of a shock are assigned a scalar value of 1. Figures 4.21 and 4.22 show the shock as detected using an iso-surface of an iso-value = 0.99. The lambda shock detected in front of the fin can be seen better in figure 4.21. The bow shock in front of the fin and the lambda shock are detected as expected. Figure 4.23 shows the iso-surface of the expansion region.

Figures 4.24 - 4.33 show the separation, attachment lines, and separation layer as detected. In these figures, the white lines are the separation lines and the black lines are the attachment lines. As seen in figures 4.24 and 4.25, the velocity vectors in the plane one computational unit away from the surface of the plate and the fin converge along the separation lines and diverge along the attachment lines as expected. The SLs and ALs detected agree with those reported in [26]. The vortex core locations taken as input for identifying the separation layer from [9] are also shown in the figure. Two separation layers - one on the plate and one on the fin are expected. Figures 4.26 and 4.27 show streamlines, starting from a few points on the SL on the plate, restricted to an i = constant plane. The streamlines separate from the surface ahead of the fin. Figure 4.28 shows a streamline from a point on another SL on the fin restricted to an i = constant plane that separates from the surface of the fin and re-attaches on the plate. Figures

4.29, 4.30, and 4.31 show the region traced by a streamline starting at different points on an SL restricted in an i = constant plane. The nodes in such regions are assigned a scalar value of 1. Figures 4.32 and 4.33 show the separation layers identified using an iso-surface of 0.99. The results agree with the previously known location of regions of flow separation [2, 9, 26].

4.4 Finned Missile

Figure 4.34 shows the finned missile case which is a multi-block chimera grid. It is an "iblanked" grid since there are holes in some zones. The flow direction is shown in the figure and the free-stream Mach number is 2.5. A cross flow separation is expected in zone 1 (the red part in the figure). Due to the limitations of the algorithm for identifying the separation layer, it is applied to zone 1 only. The shock and expansion detection algorithms are also applied on zone 1. As the shock and expansion detection algorithms are local-type methods, these can be applied to the entire grid. The results are shown in figures 4.35 - 4.39. Figure 4.35 shows the region identified as shock. Figure 4.36 shows the expansion region as identified. The velocity vectors in the plane one computational unit away from the surface converge along the separation line and diverge along the attachment line as shown in figure 4.37. The white lines are separation lines and the black lines are attachment lines. The vortex core location used for this purpose is also shown in the figure. Figure 4.38 shows the streamlines drawn from a few points on the SL restricted to an i = constant plane. It can be seen that the streamline separates from the surface and re-attaches at the attachment line. The nodes in the region traced by the streamlines from SL are assigned a scalar value of 1. Figure 4.39 shows the separation layer identified using an iso-surface of 0.99.

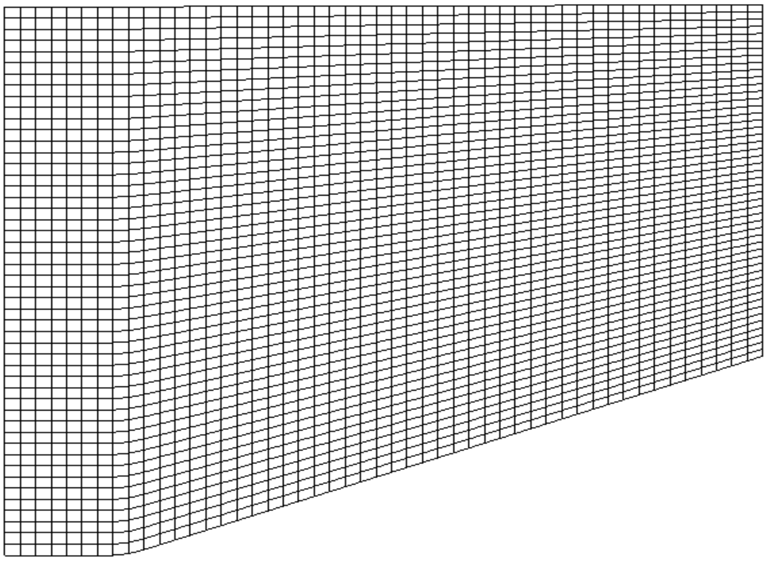


Figure 4.1: The single ramp case

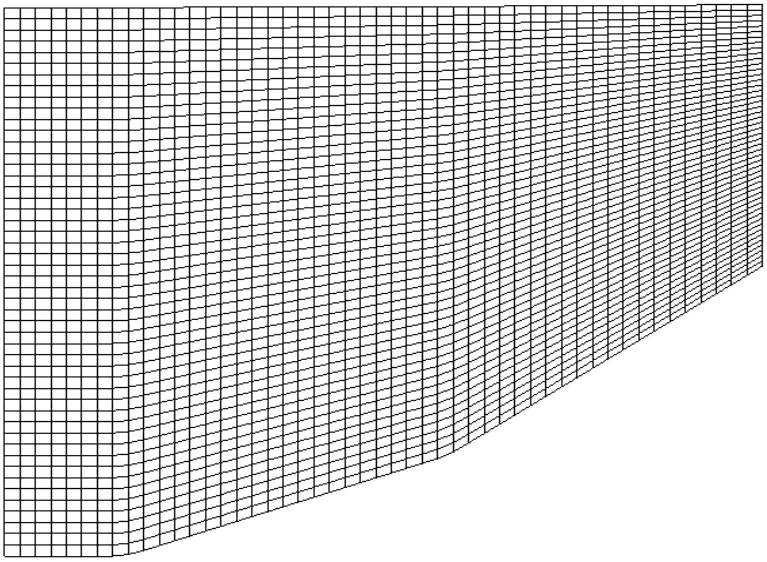


Figure 4.2: The double ramp case

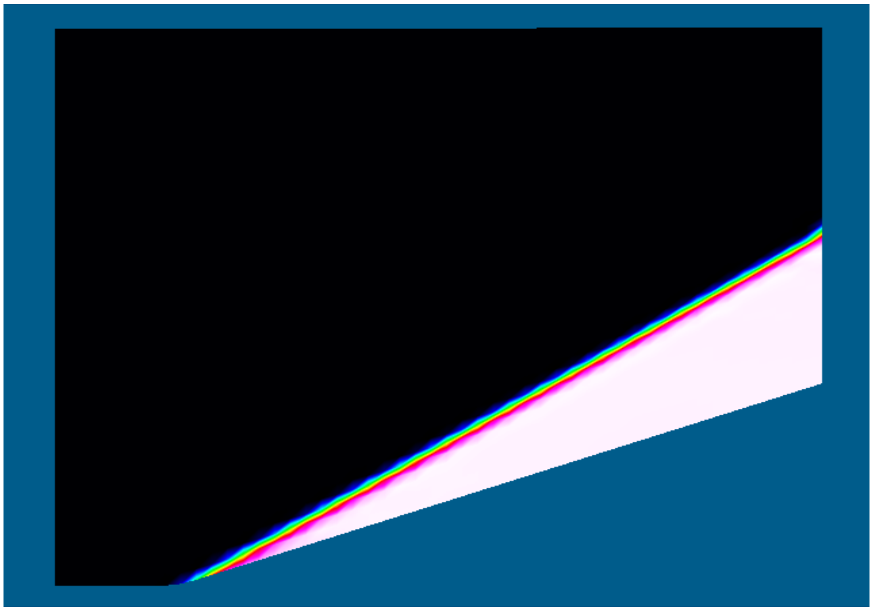


Figure 4.3: Single ramp: Pressure distribution.

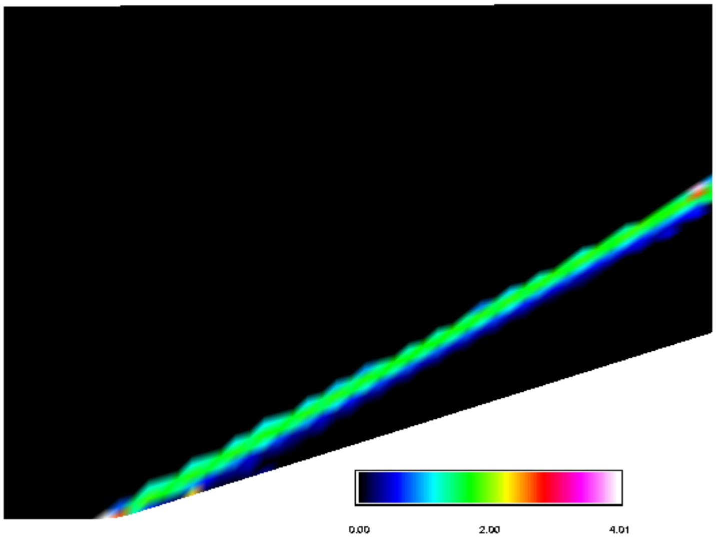


Figure 4.4: Single ramp: Normal Mach number distribution.

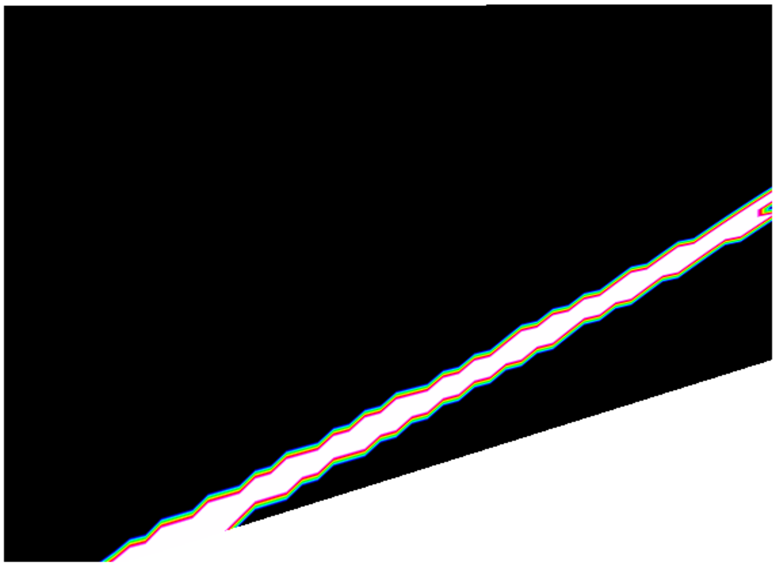


Figure 4.5: Single ramp: Nodes identified as shock.

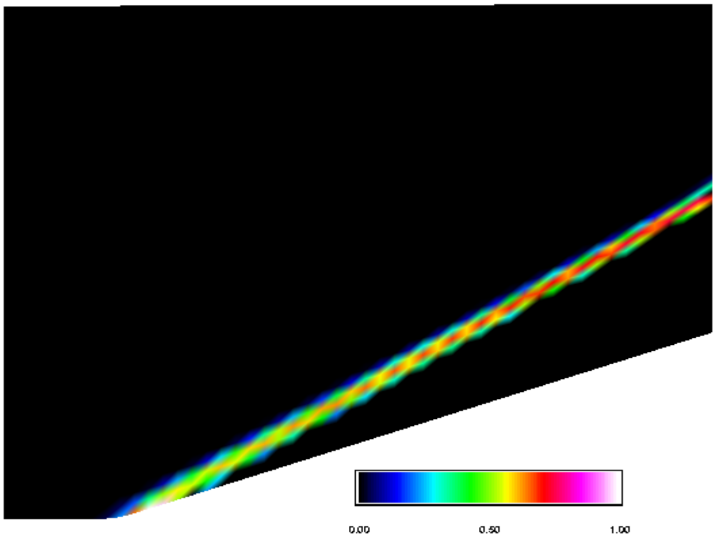


Figure 4.6: Single ramp: The identified shock with W_{shock} assigned to the nodes.

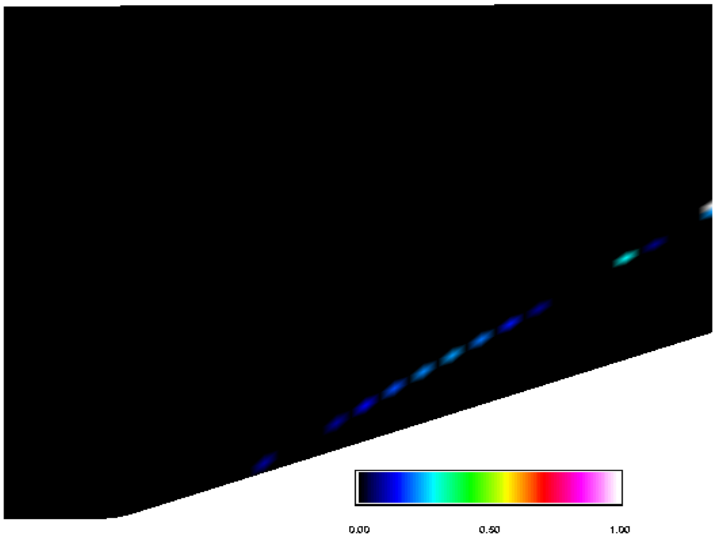


Figure 4.7: Single ramp: The expansion region identified with $W_{expansion}$ assigned to the nodes.

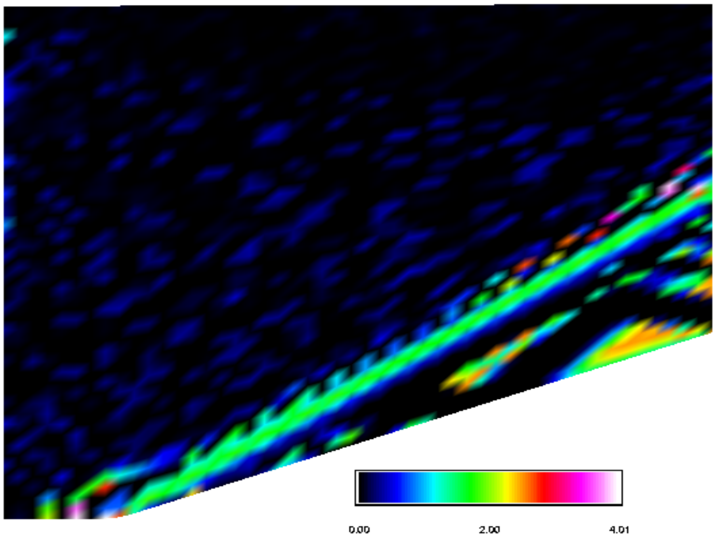


Figure 4.8: Single ramp: Normal Mach number distribution without using the pressure gradient filter.

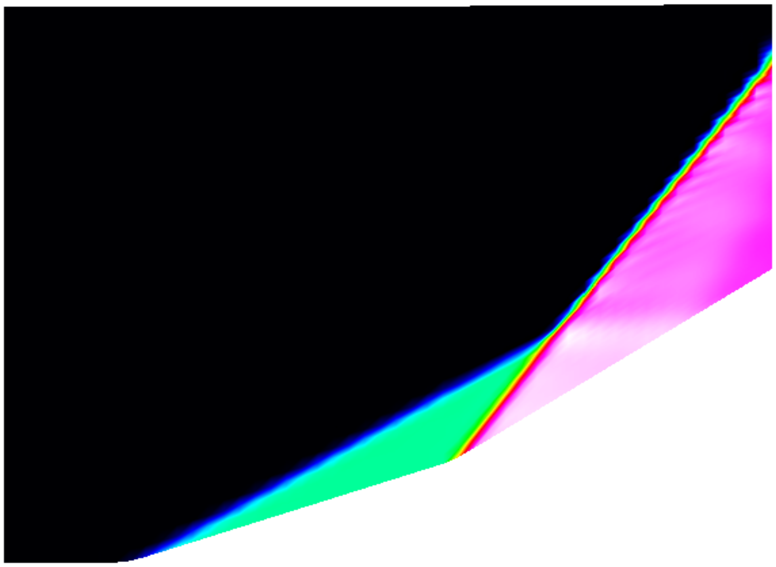


Figure 4.9: Double ramp: Pressure distribution.

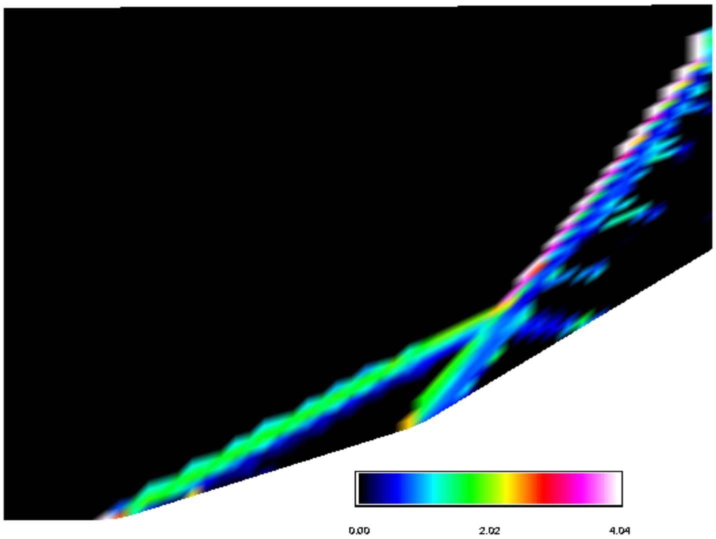


Figure 4.10: Double ramp: Normal Mach number distribution.

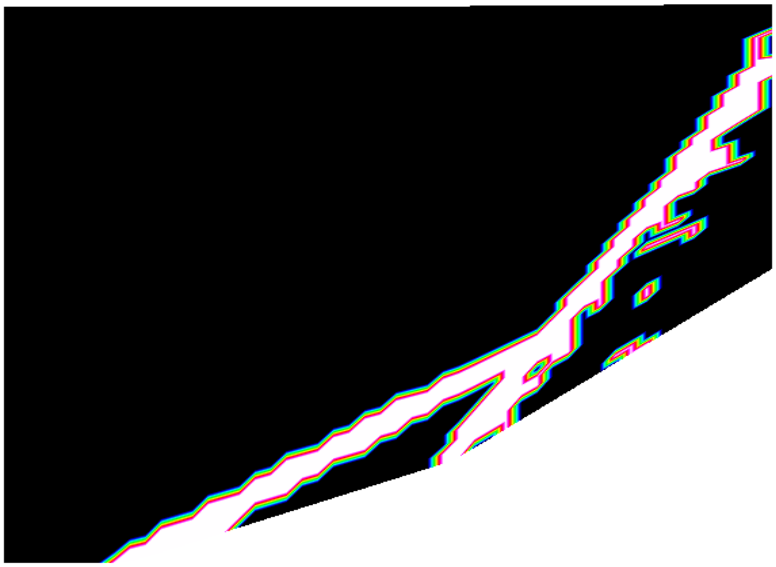


Figure 4.11: Double ramp: Nodes identified as shock.

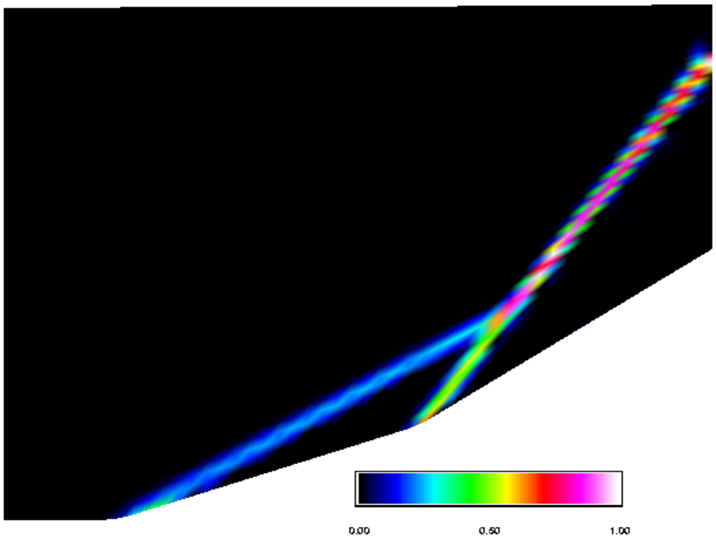


Figure 4.12: Double ramp: The identified shock with W_{shock} assigned to the nodes.

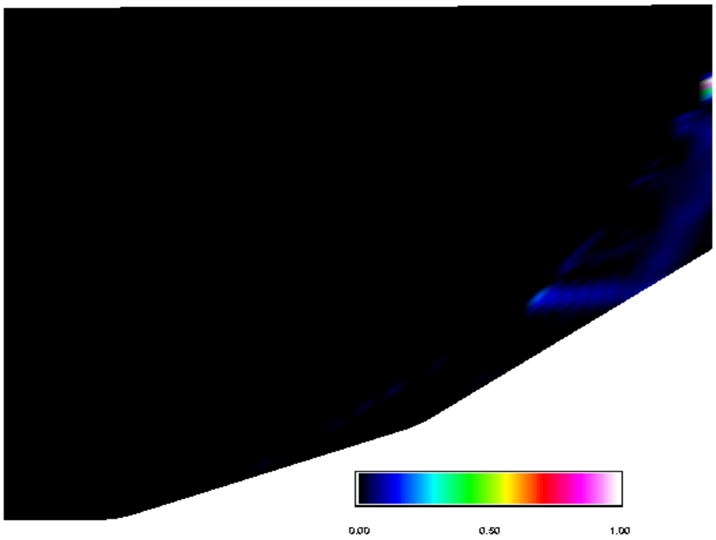


Figure 4.13: Double ramp: The expansion region identified with $W_{expansion}$ assigned to the nodes.

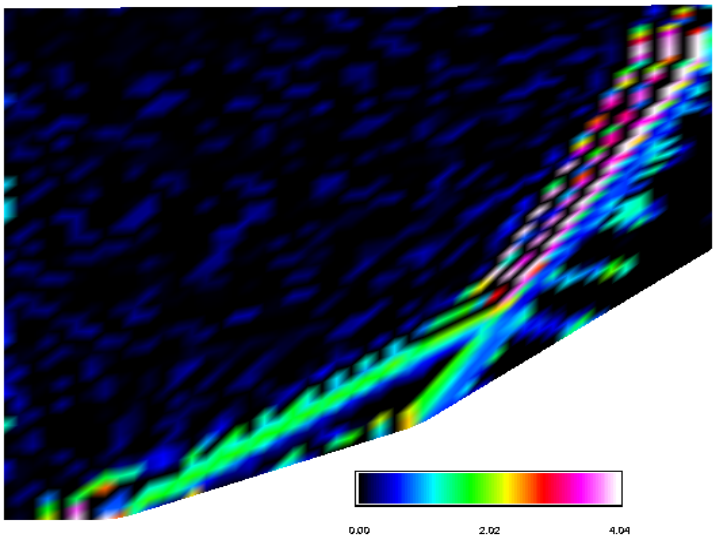


Figure 4.14: Double ramp: Normal Mach number distribution without using the pressure gradient filter.

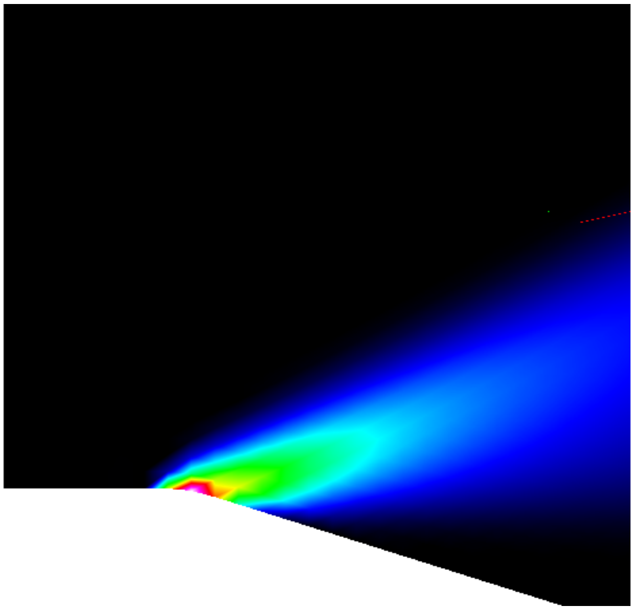


Figure 4.15: The expansion fan case.

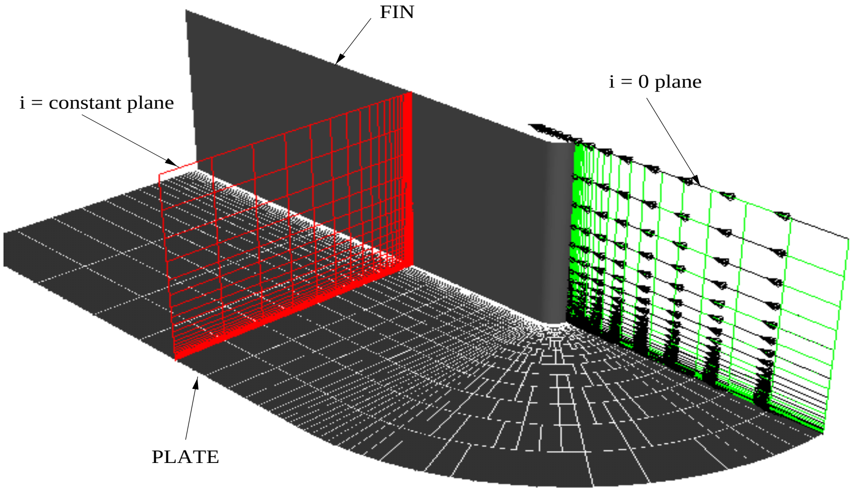


Figure 4.16: The geometry of the blunt fin test case.

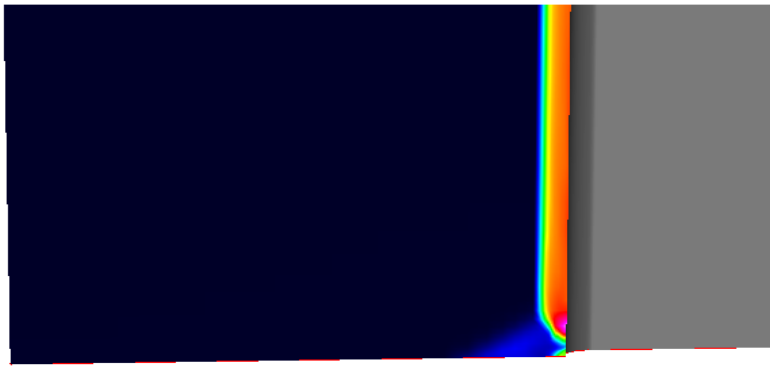


Figure 4.17: Blunt Fin: Pressure distribution in the i = 0 plane.

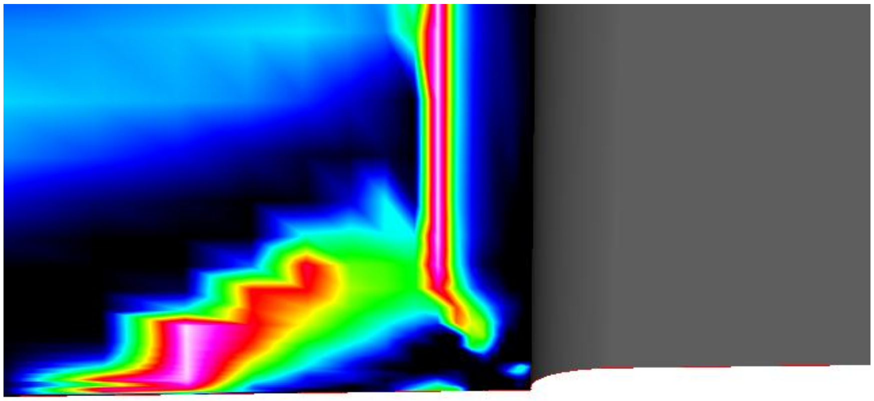


Figure 4.18: Blunt Fin: Normal Mach number distribution in the i = 0 plane.

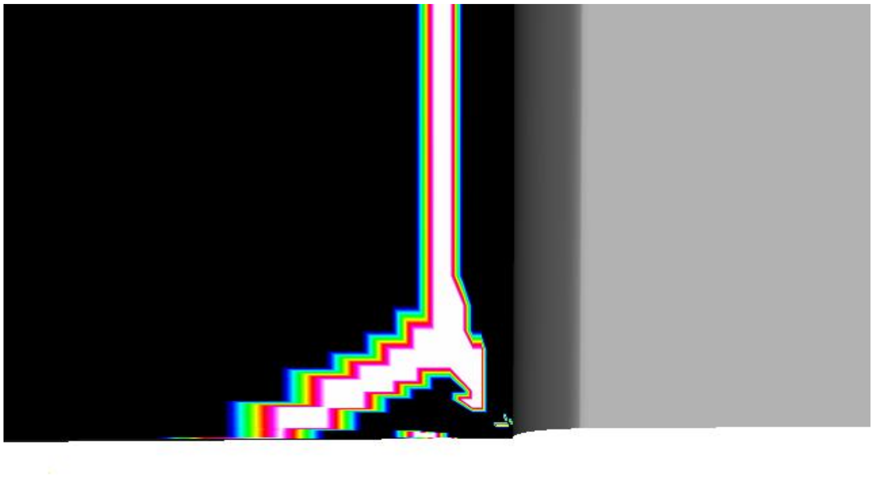


Figure 4.19: Blunt Fin: Nodes identified as shock in the i = 0 plane.

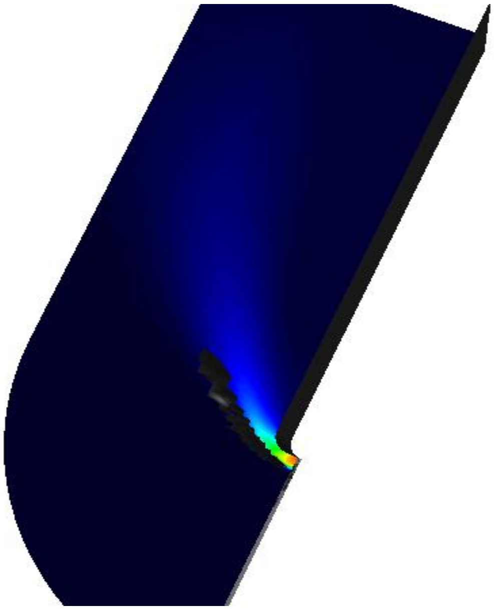


Figure 4.20: Blunt Fin: Pressure distribution in a k = constant plane.

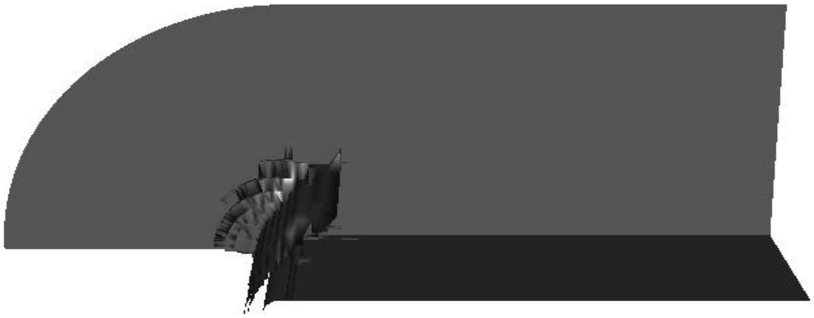


Figure 4.21: Blunt Fin: The iso-surface of the region identified as shock.

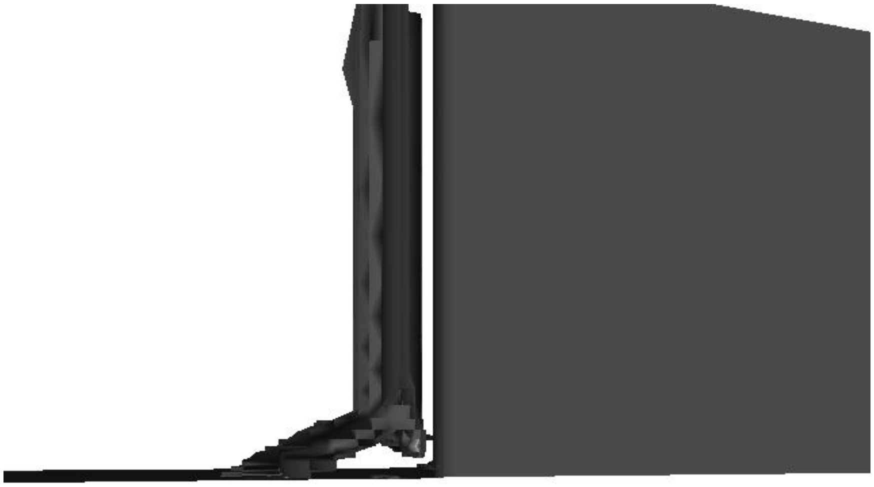


Figure 4.22: Blunt Fin: The iso-surface of the region identified as shock.

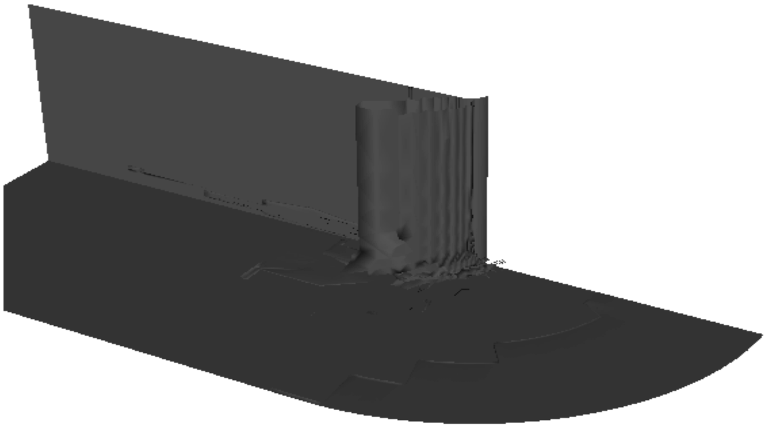
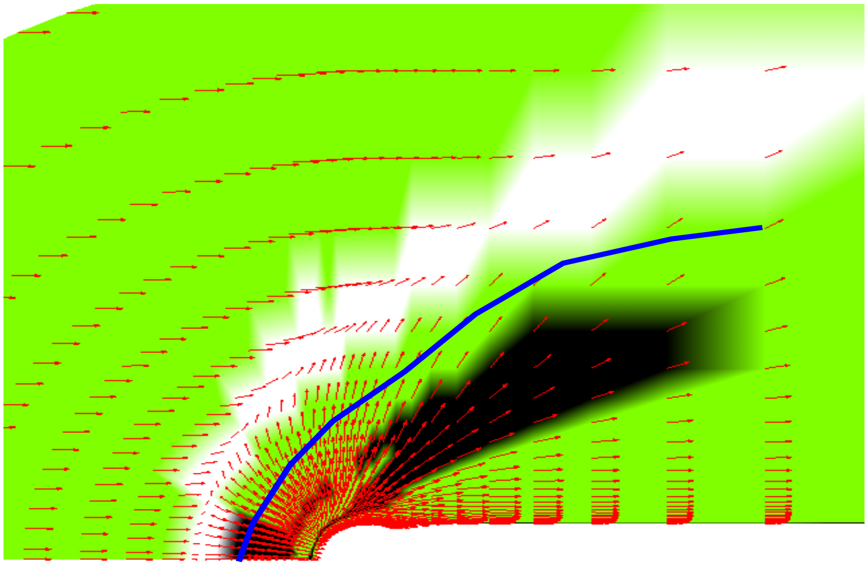
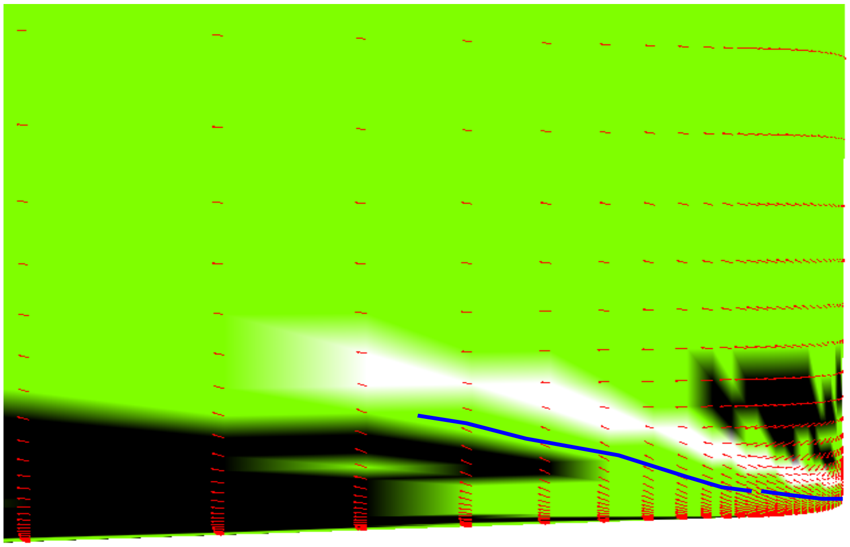


Figure 4.23: Blunt Fin: The iso-surface of expansion region.



White: Separation line, Black: Re-attachment lines, Blue: Vortex core

Figure 4.24: Blunt Fin: The velocity field in the plane one computational unit away from the surface of the plate.



White: Separation lines, Black: Re-attachment lines, Blue: Vortex cores

Figure 4.25: Blunt Fin: The velocity field in the plane one computational unit away from the surface of the fin.

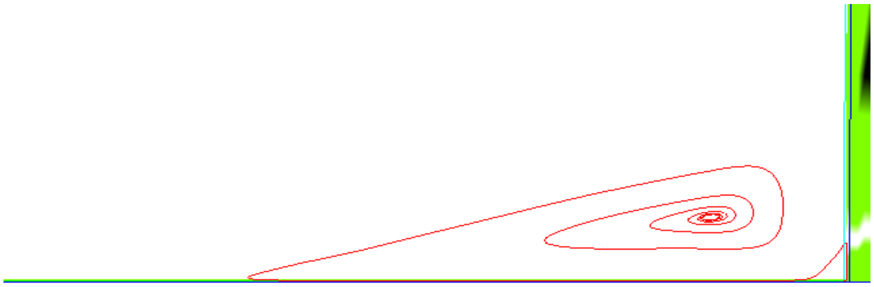


Figure 4.26: Blunt Fin: Streamline starting from points on an SL on the plate restricted to an i = 0 plane.

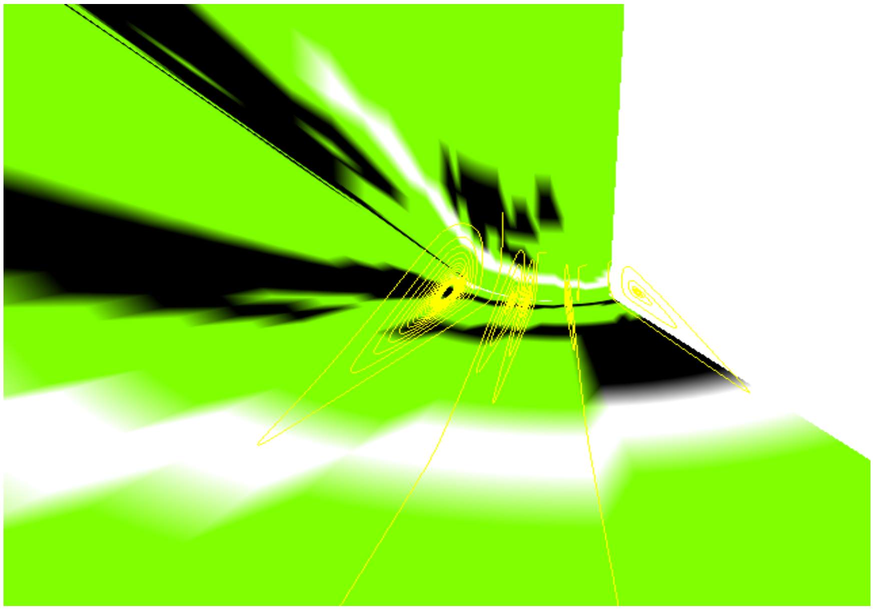


Figure 4.27: Blunt Fin: Streamlines starting from points on an SL on the plate restricted to an i = constant plane.

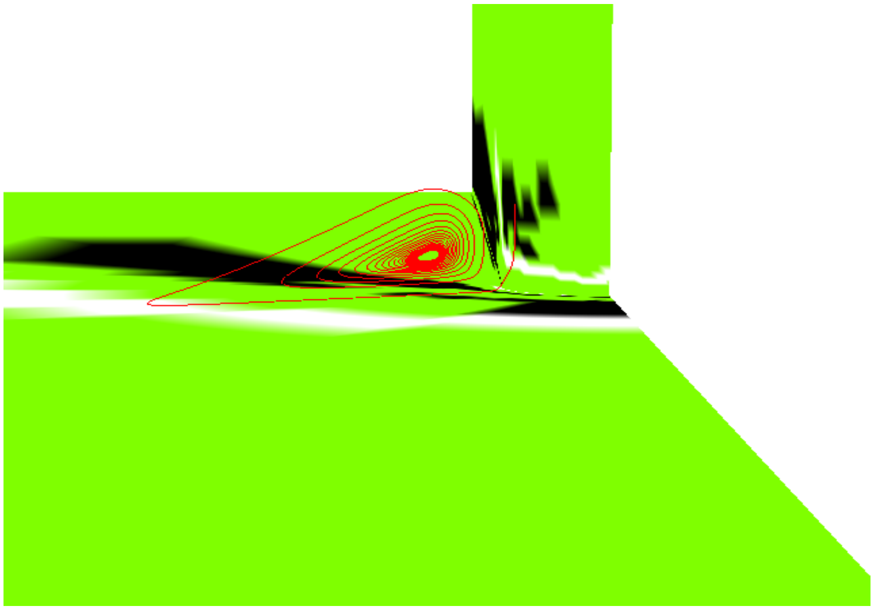


Figure 4.28: Blunt Fin: Streamlines starting from a point on an SL on the fin restricted to an i = constant plane.

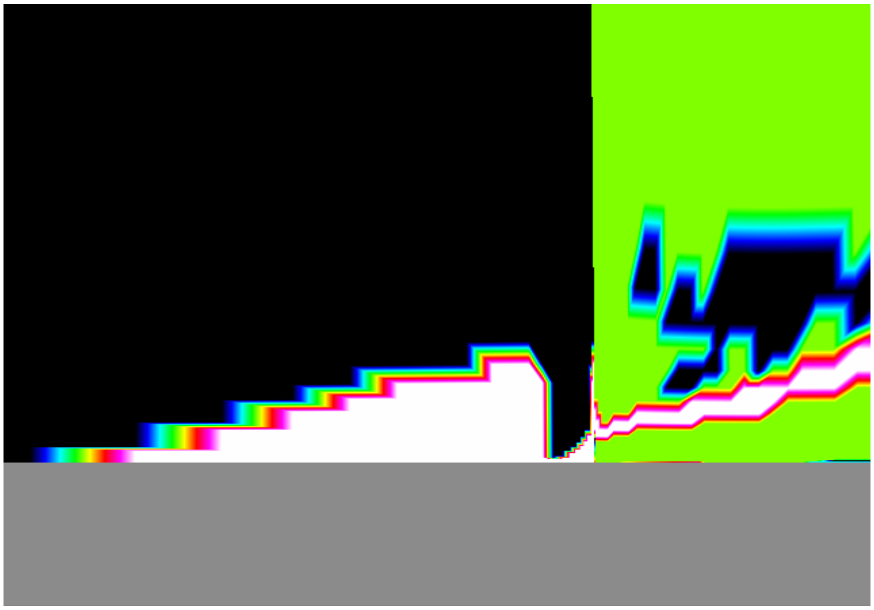


Figure 4.29: Blunt Fin: The region identified as separation layer in i = 0 plane.

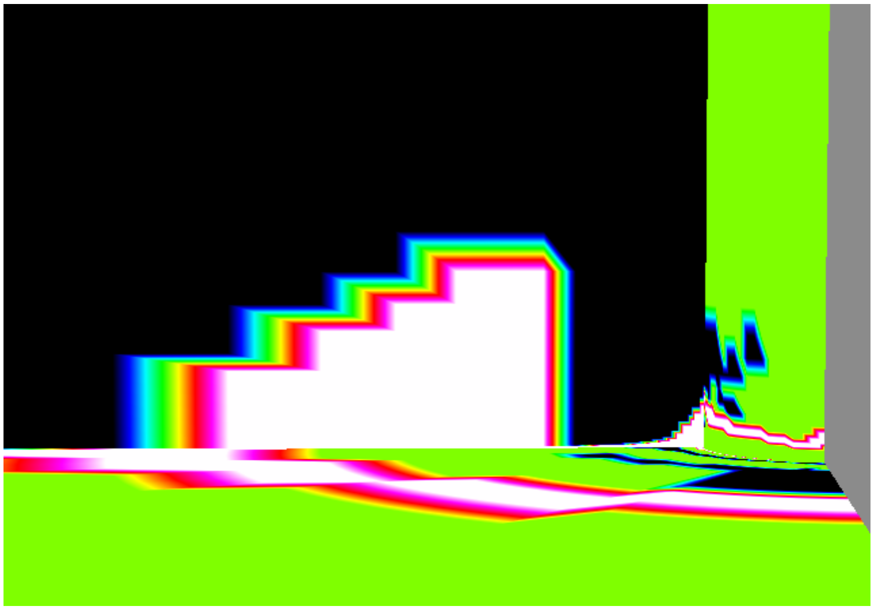


Figure 4.30: Blunt Fin: The region identified as separation layer in an i = constant plane.

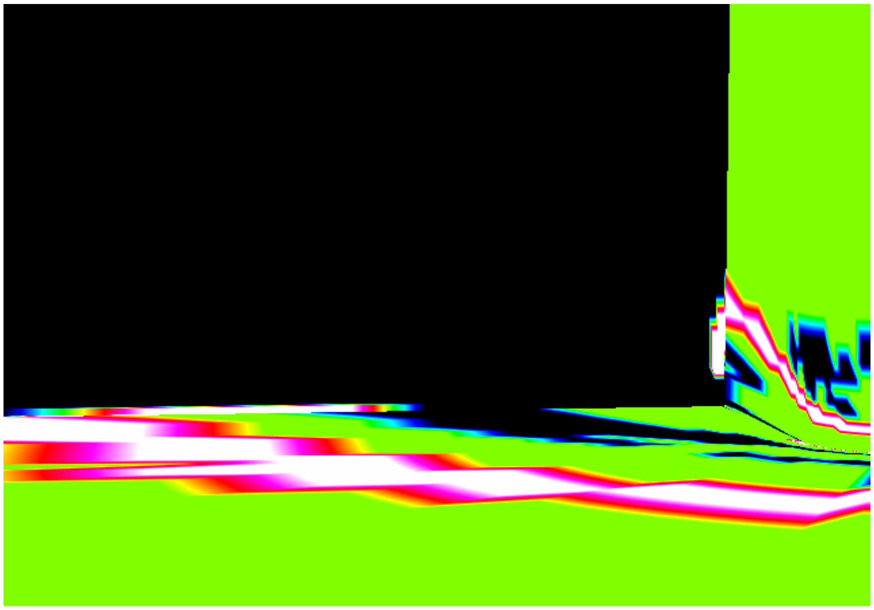


Figure 4.31: Blunt Fin: The region identified as separation layer in an i = constant plane.

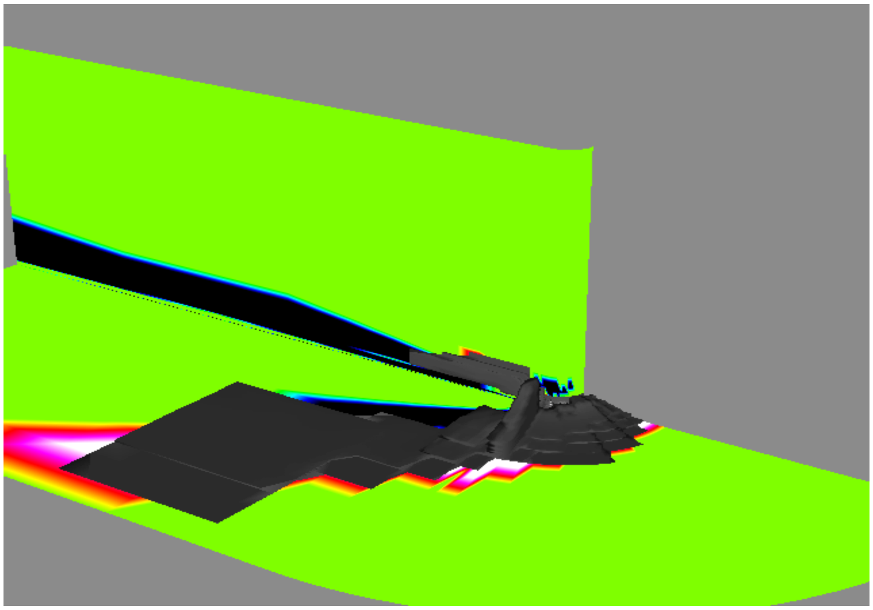


Figure 4.32: Blunt Fin: The iso-surface of separation layers.

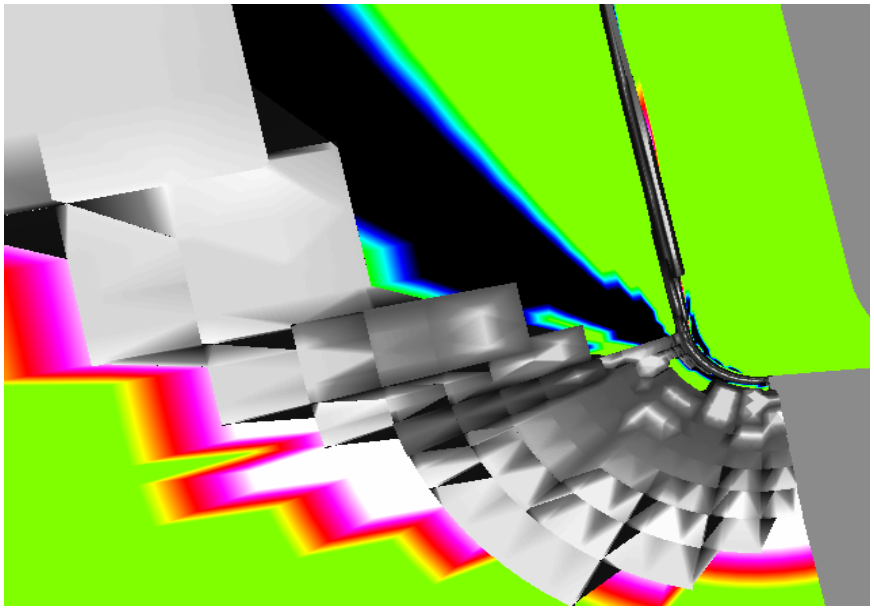


Figure 4.33: Blunt Fin: The separation layer near the nose of the fin.

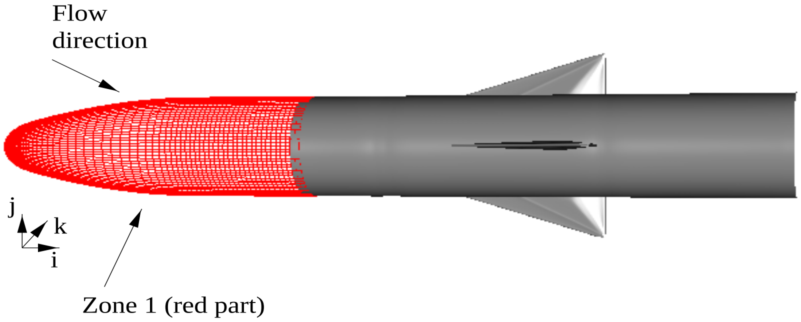


Figure 4.34: The geometry of the finned missile.

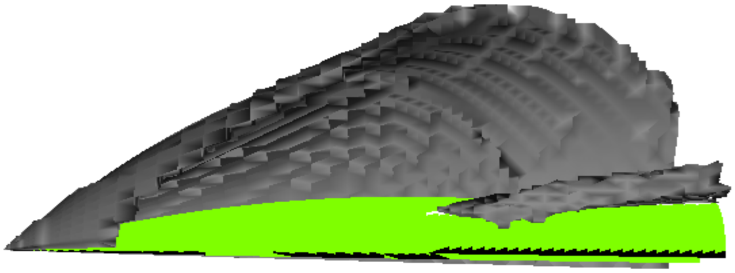


Figure 4.35: Finned Missile: The iso-surface of the region identified as shock.

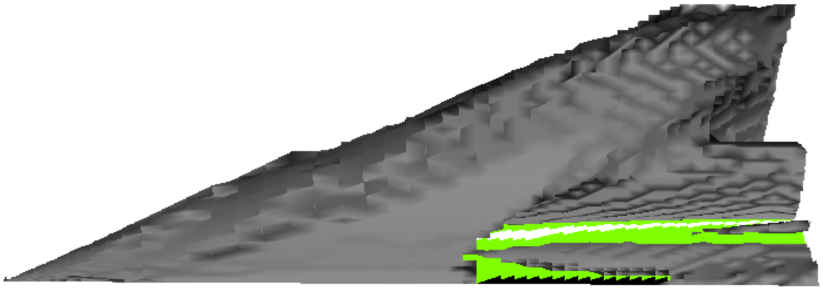
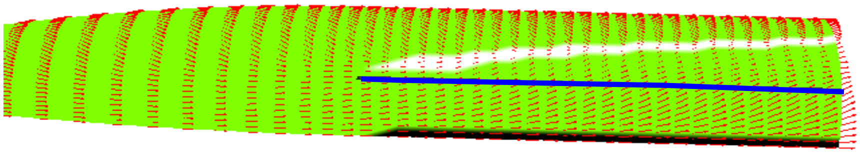


Figure 4.36: Finned Missile: The iso-surface of expansion region.



White: Separation lines, Black: Re-attachment lines, Blue: Vortex core

Figure 4.37: Finned Missile: The velocity field in the plane one computational unit away from the surface.

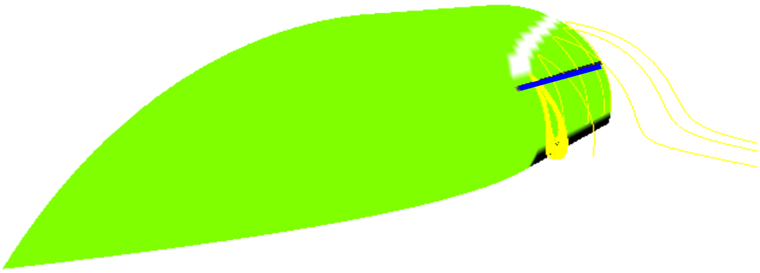


Figure 4.38: Finned Missile: Streamlines starting from an SL restricted to an i = constant plane.

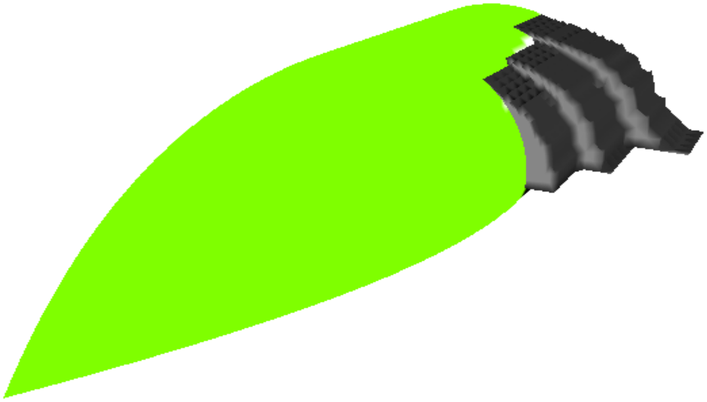


Figure 4.39: Finned Missile: The iso-surface of separation layer.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

The present work deals with automated feature detection that is appropriate for the EVITA system. The feature detection module should identify the important features in CFD datasets automatically and assign a scalar value to every point identified to represent the relative strength of the feature at that point. As a part of the feature detection module, a library for automatic feature detection has been developed [14]. Shocks, expansion regions, separation lines, attachment lines, vortices, and regions of flow separation are considered in this work.

The algorithm for shock detection based on [8] can be used for both steady and transient shocks. The results for the test cases are as expected. It is observed from the results that the filter based on the pressure gradient is effective in discarding the small disturbances in the pressure gradient. The nodes identified are assigned a scalar value as required by the EVITA system. The strength of the shock at a particular point can be deduced from the scalar value assigned to the point. The algorithm for detecting expansion region also produced results as expected.

A new algorithm for identifying a separation layer is developed. This technique uses the locations of vortex cores, separation lines, and attachment lines as input. The separation and attachment lines are identified using the algorithm given in [13]. The algorithm for detection of separation layer is designed to work on cases where there is a vortex core associated with every pair of SL/AL. The results obtained for the test cases are as expected. The technique works well even when there is more than one surface in the flow field. It should be noted that an approximate plane normal to the vortex core considered is sufficient for this technique. The threshold values are chosen worked well for the test cases implemented. Different values may be required for other cases.

However, the above technique has some limitations. A case where the flow separates from a surface in one block of a multi-block grid and re-attaches on a surface in another block cannot be solved using the current implementation. Since this approach is a global-type feature detection technique, information between two blocks is needed for streamline integration and endpoint projection. The present algorithm does not communicate this information between the two blocks. Also, in the technique implemented, it is assumed that an SL/AL pair always has a vortex core associated with it. This may not occur for some flow separations. These cases may be included for the future work in the field of feature detection.

REFERENCES

[1] R. Machiraju, J. E. Fowler, D. Thompson, B. Soni, and W. Schroeder, "EVITA - Efficient visualization and interrogation of tera-scale data," in *Data Mining for Scientific and Engineering Applications* (R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. R. Namburu, eds.), pp. 257 – 279, Kluwer Academic Publishers, 2001.

[2] M. Roth, *Automatic extraction of vortex core lines and other line-type features for scientific visualization.* PhD thesis, Swiss Federal Institute of Technology, 2000.

[3] J. L. Helman and L. Hesselink, "Representation and Display of Vector Field Topology in Fluid Flow Data Sets," *IEEE Computer*, vol. 22, no. 8, pp. 27–36, August, 1989.

[4] R. Haimes and D. Kenwright, "On the Velocity Gradient Tensor and Fluid Feature Extraction," *AIAA Paper 99-3288*, Norfolk, VA, June 1999.

[5] L. M. Portela, *On the Identification and Classification of Vortices.* PhD thesis, Stanford University, 1997.

[6] K. L. Ma, J. V. Rosendale, and W. Vermeer, "3D Shock Wave Visualization on Unstructured Grids," in *Proc. Volume Visualization '96*, pp. 87–94, October 1996.

[7] D. L. Marcum and K. P. Gaither, "Solution Adaptive Unstructed Grid Generation Using Pseudo-Pattern Recognition Techniques," *AIAA Paper 97-1860*, June 29 - July 2, 1997.

[8] D. Lovely and R. Haimes, "Shock Detection from Computational Fluid Dynamics Results," *AIAA Paper 99-3285*, Norfolk VA, June, 1999.

[9] M. Roth and R. Peikert, "A Higher-Order Method For Finding Vortex Core Lines," in *Proc. Visualization '98*, pp. 143–150, October 1998.

[10] D. Sujudi and R. Haimes, "Identification of Swirling Flow in 3D Vector Fields," *AIAA Paper 95-1715*, San Diego CA, June 1995.

[11] C. H. Berdahl and D. S. Thompson, "Eduction of Swirling Structure Using the Velocity Gradient Tensor," *AIAA J.*, vol. 31, no. 1, pp. 97–103, January 1993.

[12] J. L. Helman and L. Hesselink, "Surface Representation of Two- and Three- dimensional Fluid Flow Topology," in *Proc. Visualization '90*, pp. 6–13, 1990.

[13] D. N. Kenwright, "Automatic Detection of Open and Closed Separation and Attachment Lines," in *Proc. Visualization '98*, pp. 151–158, 1998.

[14] S. Dusi, D. Thompson, and B. Soni, "Automated Detection of Features in CFD Datasets," Tech. Rep. MSSU-COE-01-13, ERC, Mississippi State University, December 2001.

[15] J. D. Anderson, *Fundamentals of Aerodynamics.* McGraw-Hill Book Company, 1984.

[16] V. Juvvigunta and R. Machiraju, "Shock Detection and Analysis Using Wavelet Based Techniques," in *7th International Conference on Numerical Grid Generation in Computational Field Simulations* (B. K.Soni, J. Hauser, J. F. Thompson, and P. Eiseman, eds.), pp. 559 – 568, September 25 - 28, 2000.

[17] G. Bancroft, F. Merritt, T. Plessel, P. Kelaita, R. McCabe, and A. Globus, "FAST:A Multi-Processing environment for Visualization of CFD," *Proc. Visualization, IEEE Computer Society*, San Francisco, 1990.

[18] L. Pierce and P. G. Buning, *Plot 3D Manual*, 3.6 ed., 1989.

[19] D. C. Banks and B. A. Singer, "Vortex tubes in turbulent flows: Identification, representation and reconstruction," in *Proc. Visualization '94*, pp. 132–139, 1994.

[20] Y. Levy, D. Degani, and A. Seginer, "Graphical Visualization of Vortical Flows by Means of Helicity," *AIAA J.*, vol. 28, no. 8, pp. 1347–1352, August 1990.

[21] J. Jeong and F. Hussain, "On Identification of a Vortex," *J. Fluid Mechanics*, vol. 285, pp. 69–94, 1995.

[22] A. Sadarjoen, F. H. Post, B. Ma, D. C. Banks, and H.-G. Pagendarm, "Selective Visualization of Vortices in Hydrodynamic Flows," in *Proc. Visualization '98*, pp. 419–422, 1998.

[23] B. Nakshatrala, "Feature-based embedded representation of vector fields," Master's thesis, Mississippi State University, December 1999.

[24] M. S. Chong, A. E. Perry, and B. J. Cantwell, "A general classification of three-dimensional flow fields," *Phys. Fluids A*, vol. 2, no. 5, pp. 765 – 777, May 1990.

[25] S. K. Robinson, "Coherent motions in turbulent boundary layer," *Ann. Rev. Fluid Mechanics*, vol. 23, pp. 601–639, 1991.

[26] C. M. Hung and P. G. Buning, "Simulation of Blunt-Fin Induced Shock Wave and Turbulent Boundary Layer Interaction," *J. Fluid Mechanics*, vol. 154, pp. 163–185, May 1985.