The Extraction of Shock Waves and Separation and Attachment Lines From Computational Fluid Dynamics Simulations Using Subjective Logic

Matthew C. Lively
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd
Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation
Lively, Matthew C., "The Extraction of Shock Waves and Separation and Attachment Lines From Computational Fluid Dynamics Simulations Using Subjective Logic" (2012). All Theses and Dissertations. 3287.
https://scholarsarchive.byu.edu/etd/3287

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen amatangelo@byu.edu.
ABSTRACT

The Extraction of Shock Waves and Separation and Attachment Lines From Computational Fluid Dynamics Simulations Using Subjective Logic

Matthew C. Lively
Department of Mechanical Engineering, BYU
Master of Science

The advancement of computational fluid dynamics to simulate highly complex fluid flow situations have allowed for simulations that require weeks of computation using expensive high performance clusters. These simulations often generate terabytes of data and hinder the design process by greatly increasing the post-processing time. This research discusses a method to extract shock waves and separation and attachment lines as the simulation is calculating and as a post-processing step. Software agents governed by subjective logic were used to make decisions about extracted features in converging and converged data sets. Two different extraction algorithms were incorporated for shock waves and separation and attachment lines and were tested on four different simulations. A supersonic ramp simulation showed two shock waves at 10% of convergence, but did not reach their final spatial locations until 85% convergence. A similar separation and attachment line analysis was performed on a cylinder in a cross flow simulation. The cylinder separation and attachment lines were within 5% of their final spatial locations at 10% convergence, and at 85% convergence, much of the cylinder and trailing separation and attachment lines showed probability expectation values of approximately 0.90 - 1.00. An Onera M6 wing simulation was used to investigate the belief tuples of the two separate shock waves at full convergence. Probability expectation values of approximately 0.90 - 1.00 were displayed within the two shock waves because they are strong shock waves and because they met the physical requirements of shock waves. A separation and attachment line belief tuple analysis was also performed on a delta wing simulation. The forward portions of these lines showed probability expectation values of approximately 0.90 - 1.00, but dropped to approximately 0.60 - 0.75 as a consequence of their respective vortices breaking down and losing their strength. Similar to shock waves, high probability expectation values meant the separation and attachment lines were strong and physically met separation and attachment line physics. The subjective logic process presented in this research was able to determine which shock waves and separation and attachment lines were most probable, making it easier to view and further investigate these important features.

Keywords: feature detection, subjective logic, shock waves, separation lines, attachment lines, CFD
ACKNOWLEDGMENTS

The author would like to thank his thesis advisor, Dr. Steven Gorrell, for his constant guidance, support and editing. The author would also like to thank committee members, Dr. Daniel R. Maynes and Dr. Julie Vanderhoff, for their input and aid. The author would like to thank the Air Force Office of Scientific Research for sponsoring part of this project with Contract # FA9550-08-C-0058. The author would also like to thank Dr. Robert Woodley and 21st Century Systems, Inc. for their help and support. Finally, the author would like to thank his wife Christy and our daughter Emily for their belief and support, and for allowing me the time to complete this research.
### TABLE OF CONTENTS

**LIST OF TABLES** ................................................................. viii

**LIST OF FIGURES** .............................................................. x

**NOMENCLATURE** ................................................................. xiv

**Chapter 1 Introduction** ......................................................... 1
  1.1 Motivation ................................................................. 1
  1.2 Feature Extraction ....................................................... 2
  1.3 Software Agents .......................................................... 5
  1.4 Objective ................................................................. 6
  1.5 Method ................................................................. 6
  1.6 Overview ............................................................... 7

**Chapter 2 Background and Literature Review** ................................ 9
  2.1 Fluid Shock Waves ....................................................... 9
    2.1.1 Shock Wave Extraction Methods .................................. 10
    2.1.2 Lovely-Haines Algorithm ........................................ 11
    2.1.3 Ma-Rosendale-Vermeer Algorithm ................................ 14
  2.2 Fluid Separation and Attachment Lines .................................. 16
    2.2.1 Separation and Attachment Extraction Methods ................. 19
    2.2.2 Parallel Vectors Algorithm ..................................... 21
    2.2.3 Phase Plane Algorithm ........................................ 22
  2.3 Subjective Logic ....................................................... 25
    2.3.1 Belief Triangle .................................................. 26
    2.3.2 Probability Expectation ......................................... 26
  2.4 Trust Networks .......................................................... 28
    2.4.1 Subjective Logic Operators .................................... 29
    2.4.2 Example Trust Network ......................................... 30
  2.5 Current Dataset Post-Processing Concepts ................................. 31
    2.5.1 CAFÉ ............................................................. 31
    2.5.2 Evita ............................................................ 32

**Chapter 3 General Numerical Feature Extraction Method** ....................... 35
  3.1 Extracting and Filtering Features ..................................... 35
  3.2 Filtering Extracted Features ........................................... 37
  3.3 Creating Opinions of Extracted Features ................................. 38
    3.3.1 Intelligent Agent Structure ..................................... 39
    3.3.2 Algorithm Agent Opinions ...................................... 40
    3.3.3 Extracting Algorithm Agent ...................................... 42
    3.3.4 Non-Extracting Algorithm Agent .................................. 43
    3.3.5 Master Agent .................................................. 44
3.4 Aggregating Final Feature Set Opinions .................................................. 46

Chapter 4 Shock Wave Feature Extraction Method ........................................ 47
4.1 Shock Wave Characteristics ................................................................. 47
  4.1.1 Shock Strength .............................................................................. 47
  4.1.2 Parallel Normal Vectors ............................................................... 47
  4.1.3 Normal Mach Number .................................................................... 48
4.2 Extracting and Filtering Shock Waves .................................................... 48
  4.2.1 Pressure and Density Gradient Filter .............................................. 50
  4.2.2 Normal Parallel Vectors Filter ....................................................... 51
  4.2.3 Normal Mach Number Filter ........................................................ 52
4.3 Forming Opinions on Extracted Shock Waves ......................................... 52
  4.3.1 Lovely-Haines Strengths, Weaknesses, and Feature Characteristics .... 53
  4.3.2 Belief Tuple Value for Lovely-Haines Extracting Algorithm Agent ...... 55
  4.3.3 Ma-Rosendale-Vermeer Strengths, Weaknesses, and Feature Characteristics 57
  4.3.4 Belief Tuple Value for Ma-Rosendale-Vermeer Extracting Algorithm Agent 58
  4.3.5 Non-extracting Algorithm Agent Opinion ...................................... 59
  4.3.6 Master Agent Opinion ................................................................. 60
4.4 Aggregating Final Shock Wave Feature Sets ............................................ 61

Chapter 5 Separation and Attachment Line Feature Extraction Method .............. 63
5.1 Separation and Attachment Line Characteristics ...................................... 63
  5.1.1 Pressure Difference ....................................................................... 63
  5.1.2 Velocity Difference ....................................................................... 64
  5.1.3 Curvature .................................................................................... 64
  5.1.4 Local Velocity ............................................................................... 64
  5.1.5 Wall Shear Stress ......................................................................... 65
5.2 Extracting and Filtering Separation and Attachment Lines ......................... 65
  5.2.1 Normal Pressure Difference Filter ............................................... 66
  5.2.2 Normal Velocity Magnitude Filter ............................................... 67
  5.2.3 Normal Velocity Difference Filter ................................................. 67
5.3 Forming Opinions on Extracted Separation and Attachment Lines .............. 68
  5.3.1 Parallel Vectors Strengths, Weaknesses, and Feature Characteristics ... 68
  5.3.2 Belief Tuple Value for Parallel Vectors Extracting Algorithm Agent ...... 70
  5.3.3 Phase Plane Strengths, Weaknesses, and Feature Characteristics ....... 72
  5.3.4 Belief Tuple Value for Phase Plane Extracting Algorithm Agent .......... 73
  5.3.5 Non-extracting Algorithm Agent Opinion ...................................... 74
  5.3.6 Master Agent Opinion ................................................................. 75
5.4 Aggregating Final Separation and Attachment Feature Sets ........................ 76

Chapter 6 Method Validation ........................................................................ 77
6.1 Supersonic Ramp ..................................................................................... 77
  6.1.1 Shock Waves in Converging Datasets ............................................ 80
  6.1.2 Shock Waves in Converging Datasets Processed by Agents ................ 82
  6.1.3 Comparison of Shock Waves Processed by Agents from Converged Solutions 83
6.2 Swept Onera M6 Wing ....................................................... 85
   6.2.1 Definition of Extracted Shock Waves ............................. 85
   6.2.2 Comparison of Shock Waves Processed by Agents from Converged Solution 88
6.3 Cylinder In A Cross Flow .................................................. 96
   6.3.1 Separation and Attachment Lines in Converging Datasets ....... 98
   6.3.2 Separation and Attachment Lines in Converging Datasets Processed by Agents ........................................................................... 102
   6.3.3 Comparison of Separation Lines Processed by Agents from Converged Solutions ................................................................. 106
   6.3.4 Comparison of Attachment Lines Processed by Agents from Converged Solutions ........................................................................... 108
6.4 Delta Wing ................................................................. 110
   6.4.1 Definition of Delta Wing Separation and Attachment Lines .......... 113
   6.4.2 Comparison of Separation Lines Processed by Agents from Converged Solution ........................................................................... 116
   6.4.3 Comparison of Attachment Lines Processed by Agents from Converged Solution ........................................................................... 126

Chapter 7 Conclusions ..................................................... 135
   7.1 Summary ............................................................................. 135
   7.2 Conclusions ........................................................................ 136
   7.3 Recommendations ............................................................... 138

REFERENCES ................................................................. 141
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Definition of AA_E belief tuple</td>
<td>42</td>
</tr>
<tr>
<td>3.2</td>
<td>Definition of AA_NE belief tuple</td>
<td>44</td>
</tr>
<tr>
<td>3.3</td>
<td>Definition of AA_NE belief tuple</td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td>Definition of AA_E belief tuple for the Lovely-Haines shock extraction algorithm</td>
<td>54</td>
</tr>
<tr>
<td>4.2</td>
<td>Definition of AA_E belief tuple for the Ma-Rosendale-Vermeer shock extraction algorithm</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>Definition of AA_E belief tuple for the Parallel Vectors separation and attachment line extraction algorithm</td>
<td>68</td>
</tr>
<tr>
<td>5.2</td>
<td>Definition of AA_E belief tuple for the Phase Plane separation and attachment line extraction algorithm</td>
<td>72</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 Extracted shock wave surfaces surrounding a hypersonic vehicle [9]. The red surfaces represent the shock waves, while the bisecting plane is colored by pressure where red represents high pressure and blue represents low pressure. .......................... 3

2.1 A shock wave forming around a transonic aircraft [17]. ............................................. 9
2.2 Lovely-Haimes shock detection test quantity [19]. ....................................................... 12
2.3 Representation of a Lovely-Haimes shock detection shock “shell”. ................................. 13
2.4 Steps of developing flow separation around a curved surface [8]. ............................... 17
2.5 Separated and reattaching flow around a backward facing step [23]. ........................... 18
2.6 A topological skeleton of flow around a cylinder [8]. ................................................... 19
2.7 Skin friction on a blunt fin from a flow simulation at Mach 5 [8]. ................................. 20
2.8 Critical points classified by the eigenvalues of the Jacobian [8]. ................................. 21
2.9 Three phase portraits have tangent curves that asymptotically converge on an axis in the phase plane. Triangles that straddle the axes contribute line segments to either separation or attachment lines [28]. ......................................................... 25
2.10 Opinion triangle with \( \omega_x \) as an example [14]. ....................................................... 27
2.11 Simple trust network showing A’s derived trust in C from B [12]. ............................... 28
2.12 Trust network showing A’s derived trust in D from B and C requiring both consensus and discounting operators. .......................................................... 30
2.13 CAFÉ conceptual picture showing how the physics domain maps to nodes in the information space. These nodes communicate among each other, direct the data-mining activity, and interact with the operator. ......................................................... 32

3.1 2-dimensional representation of an extracted shock wave without interpolation colored by normal Mach number. ................................................................. 36
3.2 Comparison of filtered (a) and unfiltered (b) extracted shock wave feature sets. ....... 37
3.3 Representation of a two algorithm AA structure. .......................................................... 39
3.4 Representation of an additional agent being incorporated into the existing two algorithm AA structure. ................................................................. 41
3.5 Representation of two separate features extracted from the two separate AAs, AA\(_1\) shown in black and AA\(_2\) shown in red. ................................................................. 41
3.6 Representation of the same feature extracted at 200 and 300 iterations, where feature displacement measures the distance between them. ......................... 45

4.1 Representation of a swept Onera M6 wing dataset, (a) a slice of the dataset colored by Mach number and (b) the unfiltered extracted dataset generated by the MRV algorithm, many erroneous extractions exist as a result. ................................. 49
4.2 Representation of the normal parallel vectors parameter. The vectors on the left are parallel at the actual shock location while those in regions away from the actual shock location are not. ................................................................. 51
4.3 Extracted shock waves of the Swept Onera M6 wing dataset after pressure gradient, normal parallel vectors, and normal Mach number filters have been applied. ......... 53
4.4 Relationship between flow directions and the shock wave angle [22]. ........................ 54
4.5 (a) Proper extraction spacing. (b) Poor extraction spacing. The scale on either may represent belief or probability expectation. .............................................. 59

5.1 (a) Represents a PV separation line extracted feature set without filtering. (b) Represents the same feature set after filtering. ......................................................... 66

5.2 Representation of a circumscribed circle fit to three disconnected points P₀, P₁, and P₂. .......................................................... 69

5.3 (a) Proper extraction spacing. (b) Poor extraction spacing. The scale on either may represent belief or probability expectation. ................................. 74

6.1 Representation of the mesh generated for the Supersonic Bump dataset. Flow is moving from left to right. .............................................................. 78

6.2 Residual convergence history of the supersonic ramp dataset. ....................... 79

6.3 Representation of the Supersonic Ramp dataset contours colored by Mach number. 80

6.4 Comparison of MRV extracted shock waves from converged supersonic ramp dataset (black) and converging datasets (red). (a) At 10% converged, both shock waves have already formed. (b) At 25% converged, both shocks have formed but are misplaced. (c) At 50% converged, the trailing shock is still slightly displaced. (d) At 85%, both shock waves spatially match the converged dataset. In this case, flow is moving from right to left. .......................................................... 81

6.5 Comparison of shock waves extracted by MRV at 15% (a), 30% (b), 50% (c), and 85% (d) convergence increments. The black dots represent the final converged dataset. Flow is moving from the bottom right to the top left. .............................................. 83

6.6 Comparison of belief tuple values and probability expectation values for the shock waves in the converged dataset. Flow is moving from the bottom right to the top left. 86

6.7 Representation of the mesh generated for the Swept Onera M6 wing dataset. ...... 87

6.8 Residual convergence history of the swept Onera M6 dataset. ......................... 87

6.9 Representation of the Swept Onera M6 wing dataset contours colored by Mach number. ........................................................... 88

6.10 Display of shock waves extracted by LH and MRV algorithms at full solution convergence. ................................................................. 89

6.11 Display of shock waves extracted by LH and MRV algorithms colored by normalized density/pressure gradient and parallel vectors at full convergence. ........ 90

6.12 Display of shock waves extracted by LH and MRV algorithms colored by normalized density/pressure gradient and parallel vectors at full convergence. .......... 91

6.13 Comparison of belief and disbelief components of the belief tuple for the shock waves in the converged dataset. Flow is moving from left to right. ............... 93

6.14 Comparison of uncertainty and probability expectation components of the belief tuple for the shock waves in the converged dataset. Flow is moving from left to right. 95

6.15 Representation of the mesh generated for the cylinder in a cross flow dataset. ...... 96

6.16 Residual convergence history of the cylinder in a cross flow simulation. ............ 97

6.17 Close up representation of the cylinder in a cross flow data set. Surface LIC lines represent the different separation and attachment lines. .............................. 98
6.18 Comparison of PV extracted separation lines from converged cylinder in a cross flow dataset (black) and converging datasets (red). (a) At 15% converged, the cylinder separation lines are formed but misplaced. (b) At 30% converged, the aft separation lines take more form, but all are misplaced. (c) At 55% converged, both separation lines are formed and closer to their final locations. (d) At 85%, the separation lines nearly spatially match the converged dataset.

6.19 Comparison of PV separation lines on the Cylinder in a Cross Flow dataset. (a) Shows the unfiltered feature set with obviously erroneous point and (b) shows the filtered feature set.

6.20 Comparison of PV extracted attachment lines from converged cylinder in a cross flow dataset (black) and converging datasets (red). (a) At 15% converged, the stagnation attachment line has formed. (b) At 30% converged, the aft attachment line begins to take more form, but is rather messy. (c) At 50% converged, both attachment lines are formed and closer to their final locations. (d) At 85%, the real attachment lines nearly spatially match the converged dataset.

6.21 Comparison of separation lines extracted by PV at 15% (a), 30% (b), 55% (c), and 85% (d) convergence. The black dots represent the final converged dataset. Flow is moving from the top left to the bottom right.

6.22 Comparison of attachment lines extracted by PV at 15%, 30%, 55%, and 85% convergence. The black dots represent the final converged dataset. Flow is moving from the bottom left to the top right.

6.23 Comparison of belief and disbelief components of the belief tuple for the separation lines in the converged cylinder in a cross flow dataset. Flow is moving from left to right.

6.24 Comparison of the uncertainty and probability expectation components of the belief tuple for the separation lines in the converged cylinder in a cross flow dataset. Flow is moving from left to right.

6.25 Comparison of belief and disbelief components of the belief tuple for the attachment lines in the converged cylinder in a cross flow dataset. Flow is moving from right to left.

6.26 Comparison of the uncertainty and probability expectation components of the belief tuple for the attachment lines in the converged cylinder in a cross flow dataset. Flow is moving from right to left.

6.27 Representation of the mesh generated for the delta wing dataset. The red represents the mesh on the top of the delta wing, while the blue lines represent the mesh sliced in two directions with the delta wing removed.

6.28 Residual convergence history of the delta wing dataset.

6.29 Close-up representation of the delta wing dataset. Surface LIC lines represent the different separation and attachment lines. Additional lines are added to better show the primary and secondary separation and attachment lines.

6.30 A backside looking forward view of the primary and secondary vortices forming on top of the delta wing [37].

6.31 A top view of the primary and secondary vortices LIC lines forming on top of the delta wing [37].
6.32 Display of separation and attachment lines extracted by PV and PP algorithms at full solution convergence. .......................................................... 116
6.33 A cross-section of the velocity vectors above the delta wing, colored by helicity at 75% chord. ................................................................. 117
6.34 Display of separation lines extracted by PV and PP algorithms colored by normalized pressure differences at full convergence. ......................... 117
6.35 Side views of the delta wing dataset at full convergence with the surface being colored by \( C_p \), and two cut planes showing the primary vortices at 75% and 90% chord colored by helicity. ...................................................... 118
6.36 Display of separation lines extracted by PV and PP algorithms colored by normalized pressure and velocity differences at full convergence. ................. 119
6.37 Side views of the delta wing dataset at full convergence with the surface being colored by \( C_p \), and two cut planes showing the secondary vortices at 75% and 90% chord colored by helicity. ...................................................... 119
6.38 Display of separation lines extracted by PV and PP algorithms colored by normalized velocity magnitude and wall shear stress at full convergence. ................. 121
6.39 Display of separation lines extracted by PV and PP algorithms colored by normalized curvature at full convergence. ........................................... 122
6.40 Comparison of belief and disbelief components of the belief tuple for the separation lines in the converged dataset. Flow is moving from left to right. .... 123
6.41 Comparison of uncertainty and probability expectation components of the belief tuple for the separation lines in the converged dataset. Flow is moving from left to right. ................................................................. 125
6.42 Display of attachment lines extracted by PV and PP algorithms colored by normalized pressure and velocity differences at full convergence. ................. 127
6.43 Display of attachment lines extracted by PV and PP algorithms colored by normalized velocity magnitude and wall shear stress at full convergence. ................. 129
6.44 Display of separation lines extracted by PV and PP algorithms colored by normalized curvature at full convergence. ............................................... 130
6.45 Comparison of belief and disbelief components of the belief tuple for the attachment lines in the delta wing converged dataset. Flow is moving from left to right. 131
6.46 Comparison of uncertainty and probability expectation components of the belief tuple for the attachment lines in the delta wing converged dataset. Flow is moving from left to right. ................................................................. 132
NOMENCLATURE

$a$ Atomicity
$at$ Attachment Point
$AA$ Algorithm Agent
$AA_E$ Extracting Algorithm Agent
$AA_{NE}$ Non-Extracting Algorithm Agent
$b$ Belief
$c$ User defined threshold value between zero and one
$CFD$ Change in Feature Displacement
$C_p$ Coefficient of Pressure
$d$ Disbelief
$\Delta$ Discriminant
$de$ Detachment Point
$det$ Determinant
$E$ Probability Expectation
$FD$ Feature Displacement
$I_1$ $1^{st}$ Imaginary Eigenvalue
$I_2$ $2^{nd}$ Imaginary Eigenvalue
$\ell$ Length
$M$ Mach number
$MA$ Master Agent
$n$ Normal Vector
$r$ Radius
$R$ Region in a feature
$R_1$ $1^{st}$ Real Eigenvalue
$R_2$ $2^{nd}$ Real Eigenvalue
$LH$ Lovely-Haimes
$MRV$ Ma-Rosedale-Vermeer
$P$ Pressure
$PMV$ Product of the Local Normal Velocity and Local Pressure Gradient
$PP$ Phase Plane
$PV$ Parallel Vectors
$PVV$ Parallel Vectors Value
$T$ Temperature
$tr$ Trace
$S$ Separation Point
$u$ Uncertainty
$v$ Velocity vector
$V$ Velocity Field
$\dot{x}$ $x$ Component of Tangential Velocity
$x$ $x$ Cartesian Coordinate
$x_{cp}$ $x$ Component of Critical Point
$X$ $x$ Component of Cannonical Coordinate
$\dot{y}$ $y$ Component of Tangential Velocity
$y$  y Component of Tangential Velocity
$y_{cp}$  y Component of Critical Point
$Y$  y Component of Cannonical Coordinate
$\alpha$  Angle of attack
$\gamma$  Ratio of Specific Heats
$\delta_1$  1st Derivative
$\delta_2$  2nd Derivative
$\tau_w$  Wall Shear Stress
$\mu$  Dynamic Viscosity
$\rho$  Density
$\omega$  Opinion; Belief tuple
$\otimes$  Discounting operator
$\oplus$  Consensus operator
$\nabla$  Gradient operator
CHAPTER 1. INTRODUCTION

This chapter defines the motivation for this research and the importance of its development. A brief description will be given of feature extraction and software agents. The objective and method of this research will be outlined, and an overview of the succeeding chapters will be introduced.

1.1 Motivation

Computational fluid dynamics (CFD) is the analysis of fluid flow, heat transfer, and mass transfer numerically, based on the means of a computer-based simulation. The continual advancement of computational resources (parallel processing, supercomputing, etc) has allowed for the calculation of enormous and very complex CFD simulations. List [1] and Yao [2] have run unsteady RANS (Reynolds Average Navier-Stokes) simulations of gas turbine engine transonic fan stages containing 166 million grid points. Simulations of entire engine fans typically contain over 300 million grid points. These simulations can generate terabytes of data that need to be analyzed by an engineer. The post-processing time required to analyze such data becomes correspondingly large.

One challenge when conducting these high-fidelity simulations is the required post-processing and analysis of such large amounts of data. In some cases, the time required to analyze a massive data set is equivalent to the time needed to compute the solution. The post-processing of these large datasets may require a variety of software programs and techniques which can often be discipline dependent. A common method used in the post-processing of massive time-accurate CFD datasets in turbomachinery simulations requires the researcher to slowly sift through data in order to find useful information or features based on the researchers’ own intuition and/or previous experience. Existing software packages such as FieldView [3], Ensight [4], Tecplot [5], and ParaView [6] were developed to aid in post-processing and visualization of large datasets. These programs of-
ten include feature extraction techniques such as the construction of iso-surfaces using pertinent parameters and automated visualization based on researcher input criteria.

Because CFD simulations will undoubtedly continue to increase in size and complexity, new post-processing techniques are needed to help a researcher quickly parse through data to find useful information needed for design improvement. The research discussed in this paper is part of the Concurrent Agent-enabled Feature Extraction (CAFÉ) [7] concept used for feature extraction and visualization which meets this challenge by reducing post-processing time. The CAFÉ concept is currently being jointly developed by Brigham Young University and 21st Century Systems, Inc. CAFÉ is an agent-based data mining software system designed to be an after market plug-in for CFD packages that will perform concurrent analysis of CFD data, allowing the researcher to quickly find the needed information. This concept will be discussed in greater detail in Section 2.5.1.

1.2 Feature Extraction

Post [8] defines features as “phenomena, structures or objects in a data set, that are of interest for a certain research or engineering problem.” Some important features of interest in CFD simulations are vortices, shock waves, and separation and attachment lines. Often, these flow features can be located in CFD datasets using visual inspection of streamlines or flow properties such as pressure gradients. However, these methods can be a time-consuming iterative process of visualization and searching. A simple visual inspection may not always reveal all pertinent features within a dataset, nor include important physical feature characteristics such as the size of a shock area and shock strength. Feature extraction works by determining the location of relevant features within a dataset isovalue range without visualization. This is done in an automated manner that requires little or no researcher input. Feature extraction is an automated process by which a feature is precisely located in a dataset using its physical characteristics. This extraction process is especially useful because it can prioritize data for further analysis and provide insight to the relevant flow physics. Also, if a data set is too large to be saved to a hard disk, the file size of the extracted features can be orders of magnitude smaller, allowing them all to be stored with ease.

After the features of interest are extracted, they can then be easily visualized and made useful by displaying information for each feature such as feature location and strength. Extracted
features have no real significance until they can be visualized to show where spatially and when temporally they occur within each dataset and how they affect the surrounding flow. Vortices and separation and attachment lines may be visualized by lines or a series of points forming lines. However, shock waves need be visualized using an opaque surface such as a 2D or 3D Delaunay surface (or more simply, a cloud of disconnected points forming a surface). Figure 1.1 gives an example visualization of an extracted shock surface surrounding a hypersonic vehicle [9]. When visualized, extracted features not only give a researcher quick insight into where design improvements can be made, but also allows for better understanding of how particular features affect other features within the flow domain.

![Figure 1.1: Extracted shock wave surfaces surrounding a hypersonic vehicle [9]. The red surfaces represent the shock waves, while the bisecting plane is colored by pressure where red represents high pressure and blue represents low pressure.](image)

One of the most significant drawbacks to feature extraction from massive datasets is the amount of time it takes to extract the features of interest. The amount of time required to perform the extractions can often be long enough that it actually hinders post-processing rather than helping. Suppose you are using a software package that cannot run in parallel, requiring one processor to be used on a entire dataset that just used hundreds of processors to compute. The time required to extract the features of interest would be too large to take advantage of the reduced post-processing time associated with feature extraction. Also, feature extraction is currently be-
ing performed after the CFD simulation is converged, requiring extra computation time after the simulation is complete. Feature extraction can be advanced by extracting features in parallel while running concurrently alongside the simulation. This allows the features to be available while the simulation is still running, as well as when the simulation is completed. The concurrent extraction process will decrease post-processing time, allowing for lower design costs and faster design-to-production (DTP) times. The CAFÉ concept will incorporate these concurrent and parallel feature extraction methods, while extracting and determining the believability of each extraction.

Feature extraction is often performed algorithmically where each feature requires its own unique extraction algorithm. Unfortunately, for each feature there is not one markedly superior algorithm that accurately extracts all features within the spacial and temporal flow domain. One shock algorithm could accurately extract the shock wave under certain circumstances, but may not extract the full feature or extract false detections under other circumstances. Ma, Rosendale, and Vermeer [10] noted these many false detections by stating, “it is clear that there is no single best shock detection...algorithm.” Similarly, Roth [11] states, “none of the [vortex extraction] methods are clearly superior in all the tested data sets.” This leaves the problem of needing to run the same dataset through multiple extraction algorithms and parse through the extractions in order to find the relevant features and determine which are real and which are erroneous.

Roth theorized when extracting vortices that “an idea for a follow-up project situated in computer science is adding methods from computer vision and AI [artificial intelligence] techniques to combine the various proposed definitions into a single system. Such a system would calculate the vortex cores according to a set of definitions, and then try to use knowledge about the strengths and weaknesses of each method to determine a single set of vortex cores. For example, as long as the resulting vortices are sufficiently strong or almost straight, the zero curvature definition produces very good results. So by adding higher-level post-processing and considering the various feature detection algorithms as specialized knowledge bases, one could use a rule-based AI system to decide which definitions are most likely to give the best results in each particular situation [11].”

Even though Roth’s statement was specifically directed towards extracting vortices, the same idea can be extended to any flow feature of interest as long as an appropriate extraction algorithm is applied. In this research, multiple extraction algorithms are used to locate features
instead of using only one extraction algorithm per feature. Intelligent software agents governed by subjective logic are then used to determine the believability of each extraction.

1.3 Software Agents

An intelligent software agent is “a piece of software that can act autonomously. It is able to make decisions and decide the outcome of situations without being told by an end user what actions to take. An intelligent agent may use a predefined set of information to decide what action to take in any given situation or it may use a form of machine learning to identify what course of action is best [12].” Agents are used because they can properly account for the strengths and weaknesses of each algorithm. In this research, agents will be given a predefined set of information in which to govern their behavior which will then be quantified and input into agent opinions defined by subjective logic [13–15].

Subjective logic is defined as a mathematics-based logic system that forms opinions that can account for the uncertainty in a system using four basic elements: belief ($b$), disbelief ($d$), uncertainty ($u$), and atomicity ($a$). Atomicity is used to give a proper weighting if an opinion is more believable overall than another. In this research, the atomicity is set to give equal weight to each agent, allowing atomicity to be dropped from the agent opinion. This setting was made so that this method could be used generally on any CFD dataset. The three opinion elements are shown below in Equation 1.1 where $\omega$ represents the entire opinion, or belief tuple.

$$\omega = f(b, d, u)$$

(1.1)

Each individual agent is able to make an intelligent decision based on the opinions developed by each. “Three opinion values in subjective logic allow agents to form opinions that are not strictly one way or the other. In other words, an agent has some subjectivity about the outcome of a situation. An agent can find, based on given information, how probable an outcome is rather than simply reducing the outcome to a binary situation of will, or will not occur. [12]” Subjective logic is also useful when needing to make a decision about uncertain situations and/or when data is missing or incomplete. Missing or incomplete data is considered as part of an agent’s uncertainty
value. During concurrent feature extraction, some data will contain very high uncertainty requiring the agents to make suitable decisions.

1.4 Objective

The objective of this research is to continue the development of a software package for the extraction of flow features from CFD data sets while simulations are converging, and as a post-processing step when simulations are converged. This research is a continuation of work presented by Mortensen [12] for the concurrent detection and extraction of vortex core lines but will focus on shock waves and separation and attachment lines. The continued development of this software will be designed as a part of the CAFÉ concept.

Multiple shock wave and separation and attachment extraction algorithms will be investigated. Since these algorithms generate many false extractions, it can be difficult to determine which are real and which are not. In order to determine which extractions are real, the strengths and weaknesses of each will be detailed. Using these strengths and weaknesses, a subjective logic architecture will be developed. Subjective logic will be used to determine the believability of each extraction.

In addition to the feature locations, important information concerning feature size and strength will be readily available. Most importantly, this research provides a means to properly calculate and convey the probability of each feature in converging data sets, allowing an interpretation of features and their interactions with the flow before a CFD simulation has fully converged. The CFD simulations will provide a means to follow the evolution and development of flow features giving a good idea when these features may be extracted and when flow features are spatially correct.

1.5 Method

The general method of this process will be similar to Mortensen’s [12] use of software agents governed by subjective logic to determine the expected probability of extracted features from in-process converging data sets and to aid in decisions made about features from converged data sets. It will be shown how each value in an agent opinion was set so that an appropriate final
opinion may be formed for that extracted feature. This general method will be validated by applying the method to both shock waves and separation and attachment lines. Two CFD simulations for each feature will be presented that replicate concurrent feature extraction of shock waves and separation and attachment lines, showing it is possible to extract these features before CFD simulations are fully converged. Also, these simulations will be used to verify the implementation in VTK [6] and use of subjection on the shock wave and separation and attachment line extraction methods.

1.6 Overview

This document is organized as follows: Chapter 2 gives a background on shock wave and separation and attachment line extraction, as well as a review of subjective logic, trust networks, and some large data set post-processing programs. Chapter 3 gives the general method to extract flow features from CFD data sets using software agents governed by subjective logic. Chapter 4 gives a specific application of the general method to shock waves and separation and attachment lines. Chapter 5 gives results of CFD simulations that have shock waves and separation and attachment lines extracted from partially converged and fully converged datasets using the method described in Chapter 4. Chapter 6 discusses some recommendations for future research, and Chapter 7 provides conclusions about the research.
CHAPTER 2. BACKGROUND AND LITERATURE REVIEW

This chapter defines shock waves and separation and attachment lines and their characteristics. Feature extraction algorithms for both are reviewed. A background is also provided on subjective logic, trust networks, and novel large data set post-processing concepts.

2.1 Fluid Shock Waves

The shock process is represented by an abrupt change in flow properties (pressure, density, velocity, temperature) as the flow crosses the shock wave. Fox [16] details a commonly accepted definition of a shock wave: “irreversible discontinuities can occur in any supersonic flow field, in either internal flow or external flow... Large changes in pressure, temperature, [velocity], and other properties occur across small distances.” An example of this behavior can be seen in Figure 2.1 where a shock wave is formed around a jet traveling within the transonic regime. On a humid day the shock wave is visible because of the large vapor pressure jump across the shock. It can also be seen that the shock wave is a 3-dimensional cone that encompasses much of the aft portions of the aircraft.

Figure 2.1: A shock wave forming around a transonic aircraft [17].
Another common definition of a shock wave “is to create an iso-surface of the points where the Mach number, \( M \), is one” [8]. This occurs when the flow transitions from supersonic to subsonic (below unity in a normal shock wave, and below unity for the normal component of an oblique shock wave). As the flow quickly slows across the shock wave to subsonic, it crosses unity. It is at this point that the shock wave occurs.

Shock waves are common occurrences in many high Mach compressible flow situations. A normal shock is defined as a planar shock that develops normal to the flow direction, while an oblique shock is a shock that develops inclined at an angle to the flow. An oblique shock can also be either straight or curved. Shock waves can be destructive or useful depending on the application. They are a major source of aerodynamic losses. Shock waves can cause structural damage as well as cause flow separation, thus increasing aerodynamic losses. However, shocks can be useful in the design of aircraft gas turbine engine diffusers by decreasing the flow velocity into the diffuser. Because shock waves can be beneficial or detrimental, it becomes extremely important to know the physical location and strength of the shock waves in the flow.

The definitions described above can lead to a physical shock wave structure that contains many large internal discontinuities. This structure gives different ideas for the extraction of shock waves: extraction of points where \( M = 1 \), extraction of high pressure gradient regions and extraction of high density gradient regions.

### 2.1.1 Shock Wave Extraction Methods

A simple method to extract a shock surface is to connect all the points within the flow domain where the Mach number equals one [10]. As the flow velocity decreases across the shock wave from supersonic to subsonic, it crosses \( M = 1 \). There are many issues with this method because the Mach number can be one elsewhere. Also, if the shock wave is oblique, the downstream Mach number is not required to drop below one (although the normal component does drop below one). This method is not often used because the resultant “sonic-surfaces” do not generally represent a shock wave. For example, extractions can be generated when the undisturbed freestream flow is exactly equal to one, but these are obviously not shock waves.

Another method is to find regions in the flow where the Mach number normal to the shock, using the local pressure gradient approximately equals unity [10]. Since the local pressure gradient
is always normal to the shock wave, the thought is to find regions where the Mach number in the
direction of the pressure gradient crosses unity.

An additional idea for the extraction of a shock wave region is to locate regions containing
a maxima in the density gradient because a shock wave resides in areas with the largest density
gradient. Zero-level isosurfaces are then created from these extractions to form the shock wave.
Pandergram and Seitz used this definition to extract shock wave surfaces [18]. However, this may
not always be true when other large regions of density gradient occur such as the leading edge on
an airfoil.

For this research it was determined that two algorithms were better suited than the others.
Two criteria helped to determine which algorithms fit our application. 1) How accurately did the
algorithm identify all shock waves within the flow domain? 2) Would the algorithm adequately
identify shock waves in applications where concurrent data mining would be beneficial or applica-
able?

2.1.2 Lovely-Haimes Algorithm

One shock detection algorithm used for this research was the Lovely-Haimes (LH) algo-
rithm [19]. The LH algorithm was developed as a simple yet robust method for use in either 2D or
large 3D datasets. Additional information is given by the authors on how to transform this method
into a transient method by including a time derivative of the pressure. It is used in many CFD post-
processing packages such as EnSight [4] and Tecplot [5]. This algorithm was originally outlined
by David Darmofal [20] and was originally implemented in PLOT3D [21]. The stationary shock
portion of this algorithm was developed with knowledge of general shock geometry as shown in
Figure 2.2, where $|\vec{M}_1|$ is the freestream flow and $\hat{V}P$ is the local non-dimensional unit pressure
gradient. This Mach number normal to the unit pressure gradient ($|\vec{M}_n|$) is calculated on each node
within the dataset and used as a shock test value in determining the shock wave location.

Since the local unit pressure gradient is always normal to the shock wave, it can be used
to find the shock wave. The local unit pressure gradient is also found at each node, then dotted
with the local Mach number in order to calculate the local shock test value, $|\vec{M}_n|$ as shown in
Equation 2.1. The locations where this $|\vec{M}_n|$ value equals one are extracted as a boundary that
surrounds the shock wave location. An isosurface is then constructed using these extracted regions to define the shock wave.

$$
\vec{M}_n = \vec{M}_1 \cdot \nabla P = 1
$$

A boundary surrounding the shock is extracted because of numerical approximations. An example of this surrounding shock “shell” is shown in Figure 2.3 (in this figure, flow is moving from the bottom right to the upper left). Lovely and Haimes realized that local grid density has an effect on the thickness of the extracted region surrounding the shock. They performed a grid density study in which they simulated three 2D grids of the same geometry with differing grid densities. They found that as the grid density increased, the thickness of the “shell” decreased. This “shell” thickness can actually be a good thing considering it can give the researcher a good visual of whether or not the grid is refined enough surrounding the shock region.

One of the benefits of the LH method is that it does not extract expansion waves. However, it still may extract some false shock regions. This is partly due to small numerical errors in the gradient away from the shock. Three different filtering techniques were suggested to help eliminate false extractions. Each of these techniques starts with an isosurface of $|\vec{M}_n|$ values approximately equal to one. The first filtering method is to enforce the property that the pressure gradient is normal to the shock wave. This technique eliminates all extractions that do not meet the criteria...
Figure 2.3: Representation of a Lovely-Haimes shock detection shock “shell”.

given in Equation 2.2, where $\hat{n}$ is the normal vector to the cell surface, and c is a user defined threshold value between zero and one.

\[
\frac{|\nabla P \cdot \hat{n}|}{|\nabla P|} \leq c
\]  

(2.2)

The second filtering method removes all extractions that fall below a certain pressure gradient magnitude threshold. This technique is based on the knowledge that a shock resides in regions with very high (discontinuous) pressure gradients. A glaring problem with this technique is defining a “very high” pressure gradient and setting an appropriate threshold value. They suggest to find the largest extracted pressure gradient value and calculate a user-defined percentage of the maximum value, and use this as the threshold value.

The third filtering technique removes all extractions that do not experience jumps in density and temperature corresponding to the Rankine-Hugonoit [22] relations (Equation 2.3 and Equation 2.4). These relations state the necessary relationships between the pressure jump and the density and temperature ratios respectively across the shock. This can be difficult to implement in three dimensions because density and temperature information are needed both in front of and behind the shock. They suggest taking up to twenty measurements of pressure and density on both sides of the shock, and then use the two measurements that best fit these relationships. If any of the extractions do not meet these criteria within a certain user-defined threshold, they are removed from the extracted dataset.
\[ \frac{\rho_2}{\rho_1} = \frac{1 + \frac{\gamma+1}{\gamma-1} \frac{P_2}{P_1}}{\frac{\gamma+1}{\gamma-1} + \frac{P_2}{P_1}} \]  
(2.3)

\[ \frac{T_2}{T_1} = \frac{P_2}{P_1} \frac{\frac{\gamma+1}{\gamma-1} \frac{P_2}{P_1}}{1 + \frac{\gamma+1}{\gamma-1} \frac{P_2}{P_1}} \]  
(2.4)

Only the pressure gradient magnitude thresholding technique is used for this research. This is because Lovely and Haimes performed a filtering study to determine which method produced the best shock surface and found this to be the most accurate. With this in mind, the researcher still must define a suitable thresholding value that may eliminate some portions of the shock wave or leave false extractions.

### 2.1.3 Ma-Rosendale-Vermeer Algorithm

The second shock wave detection algorithm used for this research was the Ma-Rosendale-Vermeer (MRV) algorithm [10]. The first version of this method was generated by Pagedarm and Sietz [18], but Ma-Rosendale-Vermeer made a few additions to better filter erroneous points. The MRV method is also used as a shock wave extraction option in Ensight [4].

The general idea of this method is that a shock wave surface exists in the presence of a discontinuity in the density gradient. The shock location can be approximated by taking the position of the steepest gradient of the density gradient curve throughout the domain. Ma et al. [10] theorize that these locations can be found in a 1D example by numerically finding the second spatial derivative of the scalar \( f: \frac{\partial^2 f}{\partial x^2} \). Where \( f \) represents a fluid scalar parameter such as pressure or density. The points where this second derivative equal zero, and the first derivative \( \frac{\partial f}{\partial x} \) is non-zero represent a discontinuity and are thus extracted as part of the shock surface.

In order to apply this process to the 3D density gradient field, the 1st derivative of the density gradient needs to be taken in the direction of the local velocity vector, \( v \). This is calculated using Equation 2.5, where \( \delta_1 \rho \) denotes the first derivative of the density gradient in the direction of the local velocity vector.

\[ \delta_1 \rho = \frac{v}{||v||} \cdot \nabla \rho \]  
(2.5)
Much information can be gleaned from this $\delta_1 \rho$ value. First, for extracting shock waves this method is only interested in extracting the extrema of this quantity. Secondly, the sign of the $\delta_1 \rho$ value indicates whether or not the local flow is compressing or expanding. When $\delta_1 \rho > 0$, the local density gradient and velocity vector are aligned in the same direction, meaning that the flow is compressing. When $\delta_1 \rho < 0$, the local density gradient and velocity vector are now aligned in opposite directions, meaning that the flow is expanding. Since a shock wave is a compression wave, the negative values of $\delta_1 \rho$ are eliminated. Additionally, instead of looking for the extrema of $\delta_1 \rho$, the search is limited to the local maxima.

The location of the shock wave is found by searching for the local maxima of the quantity $\delta_1 \rho$. This is accomplished by finding the zero-value locations of the second directional derivative $\delta_2 \rho$ using Equation 2.6:

$$\delta_2 \rho = \frac{\nu}{||\nu||} \cdot \nabla \delta_1 \rho = \frac{\nu}{||\nu||} \cdot \nabla (\frac{\nu}{||\nu||} \cdot \nabla \rho)$$ (2.6)

This value is simply the second derivative of the density gradient in the direction of the local velocity vector. Similar to the one-dimensional example used in Equation 2.5, an isosurface can be constructed using the locations where the quantity $\delta_2 \rho$ equals zero.

While this method is very effective at finding the shock location, many more erroneous detections are extracted as compared to the Lovely-Haimes algorithm. Although it is true that a shock occurs when $\delta_2 \rho$ equals zero, the algorithm also calculates this to be the case when the $\delta_2 \rho$ vanishes in regions of undisturbed freestream flow. A couple of attempts were made by the authors to eliminate these erroneous extractions. The first attempt suggested by Pagedarm and Sietz [18] was to threshold the $\delta_1 \rho$ value by some user-defined constant $\varepsilon > 0$, thus eliminating those areas that are relatively undisturbed. Even though this attempt works fairly well, it can be problematic. If $\varepsilon$ is set too large, holes will be created in the shock surface. However if $\varepsilon$ is set too small, false extractions remain in non-shock areas. Choosing a proper value for $\delta_1 \rho$ can be difficult and is not a universal value and must be changed for each dataset extracted.

Ma-Rosendale-Vermeer [10] suggest an alternative to manually choosing an $\varepsilon$ value. Similar to the Lovely-Haimes algorithm [19], instead of looking at the $\delta_1 \rho$ value to use for thresholding, they use the normal Mach number (the component of the Mach number in the direction of the den-
sity gradient instead of pressure gradient). The normal Mach number will decrease from greater than one before the shock wave to below one after the shock. There is a point at which the normal Mach number will equal one as the flow crosses a shock wave. The location of this value is found using an interpolation method described in Section 3.1. However, they realize that this value is often not equal to exactly one, therefore the normal Mach number approach still needs a user-defined threshold value near one. They argue that if a proper value of the normal Mach number is chosen, better results are produced than the $\delta_l \rho$ threshold method.

While this method does not produce the “shell” shape that is produced by the LH algorithm, it does extract more points that are in regions of relatively low flow disturbance. With this in mind, the researcher still must define a proper normal Mach number thresholding value that may actually eliminate some portions of the shock wave. It can be seen that by investigating both of these algorithms, neither one is better than the other, since both require a user-defined thresholding value. The strengths and weaknesses of the LH and MRV algorithms are reviewed later (Tables 4.1 and 4.2 respectively).

### 2.2 Fluid Separation and Attachment Lines

Separation lines or points can be defined in a few different manners, which again leads to different extraction methods. A very common definition is that separation occurs where the reversed flow meets the streamwise flow. Another definition used to define separation comes from the idea that when the two flow directions meet, the local flow will be zero. It makes sense then that at this point of convergent flow, the local flow velocity is zero since a local stagnation point is created. Another important feature associated with separation is that the local wall shear stress $\tau_w$ vanishes (see Equation 2.7).

$$\tau_w = \mu \frac{\partial u}{\partial y} = 0$$  \hspace{1cm} (2.7)

Attachment lines or points are important in most engineering flows because they mark the location of boundary layer creation. Attachment and reattachment occur when a free shear layer becomes increasingly affected and finally dominated by the presence of an adjacent solid surface causing the flow at the wall to come to a stop. This free shear layer can be a freestream flow
approaching an object in the flow (attachment) or the downstream portion of a separation bubble (reattachment).

Attachment lines or points can be defined in a similar manner to separation lines, which again leads to numerous extraction methods. A very common definition is that attachment occurs where the local flow vectors are in opposite directions (pointing away from each other) [23]. Similar to flow separation, attachment can be defined using the previous flow direction idea, that when the two flow directions diverge, the local flow will be zero. Once again local wall shear stress $\tau_w$ vanishes at the points of reattachment.

Separation and attachment lines are a very common occurrence in fluid dynamic engineering situations. Separation lines are of the utmost interest because they increase drag and reduce lift, therefore their occurrence should be prevented or at least minimized [8]. Separation lines are defined as a region where the boundary layer of the flow leaves the adjacent surface creating a separation bubble or backflow section (Figure 2.4). Separation lines can only occur in the presence of an overall adverse pressure gradient (increase of pressure in the streamwise direction). “Under a strong enough adverse pressure gradient, the flow next to the wall [separates and] reverses direction, resulting in a region of backward flow. The reversed flow meets the forward flow at some point $S$ at which the fluid near the surface is transported out into the mainstream” [24]. Flow separation location occurs when the reversed flow meets the forward flow. Separation not only occurs in external flows such as over an airfoil, but also in internal flow such as a highly divergent diffuser. Additionally, laminar flows have poor resistance to adverse pressure gradients and separate easily, while turbulent boundary layers can resist separation longer.

![Figure 2.4: Steps of developing flow separation around a curved surface [8].](image-url)
Flow separation and reattachment can be broken down into 5 different zones as shown in the backwards-facing step of Figure 2.5 [23]. Zone I is the detached shear flow separating Zone II and Zone III (recirculating flow regions) from the freestream. An interesting feature at the end of Zone III is the presence of a substantial adverse streamwise pressure gradient. As the boundary layer separates from the edge of the step, a two-dimensional shear layer rolls into a core, or rotating vortex. This shear layer exhibits a Kelvin-Helmholtz type instability which gives rise to a vortical sub-structure that drives the reversed flow in Zone III. Also, the reversed flow resembles laminar-like near wall flow and sees an effective favorable pressure gradient just forward of the reattachment point. Zone IV is the reattachment point, and is characterized by very low mean velocity, an adverse pressure gradient, and the wall skin friction vanishes. Interestingly at this point, in a time-accurate solution, the flow will often reverse directions. After the attachment point, the reattached flow recovers the structure of a flat-wall boundary layer (Zone V).

Flow Zones:
1. Separated Shear Layer
2. Corner Eddy
3. Recirculation Zone
4. Reattachment Zone
5. Redeveloping Boundary Layer

Figure 2.5: Separated and reattaching flow around a backward facing step [23].

An additional common example used to define separation and attachment is laminar flow around a cylinder. Figure 2.6 [8] shows a topological skeleton of flow around a cylinder. At the leading edge is the point of attachment (at). As the flow moves downstream around the cylinder, the adverse pressure increases until the adverse pressure gradient becomes large enough that the flow separates from the cylinder (de) creating the separation bubble (ce). Under some flow conditions, the flow will reattach to the surface (at).
2.2.1 Separation and Attachment Extraction Methods

Many different methods have been developed for the extraction of separation and attachment lines. Some methods use a phenomenological approach while others use topographical approaches. The phenomenological looks for physical phenomena in the flow, while topographical is more mathematic projection based. “The phenomenological approaches allow researchers to find flow patterns on a surface, then use their insight or experience to identify separation or attachment lines. This approach is similar to mimicking flow visualization techniques used in wind tunnel experiments (similar to surface oil flow techniques). The topographical approaches are based on the mathematics of Poincaré (phase planes) and identify specific features that originate from critical points.” [8]

A very common phenomenological method for determining separation and attachment lines is to use surface streamlines or skin friction lines [25]. The skin friction approach is to seed particles near a body in the flow and compute integral curves (streamlines) that are constrained to the body. These curves converge or diverge at separation and attachment lines respectively. Figure 2.7 [8] shows a blunt fin with skin friction lines running along the side wall of the fin. This method appears to work well when many particles are released, causing a very defined separation line. This approach only works by researcher observation and does no real extraction. Additionally, since separation and attachment lines look similar in a streamline plot, it can be difficult to determine which is which.

Texture synthesis methods such as Line Integral Convolution (LIC), a texture based technique for visualizing vector fields, can create continuously-shaded images that are effectively able to display the global structure of the flow. This is commonly used to replicate experimental results obtained from oil streak flows. Similar to the skin friction method, the LIC methods require
careful researcher analysis to determine which are separation lines and which are attachment lines. This can become a very complex and time-consuming problem when working with time-accurate solutions.

Helman and Hesselink [26] were able to generate separation and attachment lines using vector field topology. The topology of a vector field consists of critical points (where the velocity is zero), and tangent curves (streamlines) that connect the critical points. Numerous vector topology critical points are shown in Figure 2.8 [8] and are classified by their first and second eigenvalues of the Jacobian matrix. However, only closed separation and attachments are found (curves that start and end at critical points). Open separations are not required to start and stop at critical points, and would thus be missed by this technique.

There are numerous algorithms that have been developed to locate separation and attachment lines. For this research it was determined that two algorithms were markedly superior than the others discussed here. The same two criteria used to select the shock wave algorithms were used for the separation and attachment algorithms. 1) How accurately did the algorithm identify all separation and attachment lines within the flow domain? 2) Would the algorithm adequately identify separation and attachment in applications where concurrent data mining would be required or if the extraction process was conducted using a concurrent architecture, would the algorithm still perform accurately?
2.2.2 Parallel Vectors Algorithm

The first separation and attachment line detection algorithm used within this research is the Parallel Vectors (PV) algorithm developed by Kenwright, Henze, and Levit [27]. This algorithm is able to detect both open and closed separation and attachment lines. Closed separation and attachment lines begin and end at a saddle or node (Figure 2.8) whereas open separation and attachment lines are not required to begin or end at a saddle or node. This method compares the eigenvectors, $e_i$, of the local velocity gradient tensor ($\nabla V$) (see Equation 2.8), also called the Jacobian Matrix, with the local velocity vector. Separation or attachment lines exist when the local streamline curvature is zero (see Equation 2.9), where $V$ is the vector field. Note that this is only performed when both of the eigenvalues are real.

$$\nabla V = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix} \quad (2.8)$$

$$e_i \times V = 0 \quad (2.9)$$

The distinction between separation and attachment lines can be easily made with the PV algorithm. If the local velocity vector is parallel to the largest eigenvector $e_1$, then the point in
question is identified as part of an attachment line. Conversely, if the velocity is parallel to the smallest eigenvector $e_2$, then the point is part of a separation line.

An additional set of lines are extracted with this algorithm, the inflection lines between the separation and attachment lines. These inflection lines occur where one of the eigenvectors is locally parallel to the velocity vector, but the line is not an asymptote (convergent or divergent) of the neighboring streamlines. These inflection lines can be easily filtered by using the simple check in Equation 2.10 because this will not be true.

$$\nabla (e_i \times V) \cdot V = 0 \quad (2.10)$$

Even though this algorithm is able to eliminate inflection lines, many erroneous separation and attachment lines are still extracted, making it difficult to determine which are real and which are false. Additionally, the authors mention that this algorithm does not accurately extract curved lines. They argue that the extractions are usually satisfactory using the assumption that the flow is locally linear. This assumption will be accurate in some situations of medium curvature, but with very curved lines it still struggles to accurately predict the correct locations of the separation and attachment lines.

### 2.2.3 Phase Plane Algorithm

The second separation and attachment line detection algorithm used within this research is the Phase Plane (PP) algorithm developed by Kenwright [28]. Similar to the Parallel Vectors algorithm, the PP algorithm can detect both the open and closed separation and attachment lines. This algorithm was developed to be local, meaning that an independent test could be applied to every node along the surface of an object. The theory of this algorithm is based on the concepts of linear phase plane analysis. It is assumed that the surface mesh of the object is subdivided into triangles (mesh with tetrahedral), and that the Cartesian vector components are stored at the vertices or nodes. A linear vector field can now be constructed that passes through each triangle and satisfies the prescribed vectors at the vertices:
\[
\begin{pmatrix}
\dot{x} \\
\dot{y}
\end{pmatrix} =
\begin{pmatrix}
a_1 \\
a_2
\end{pmatrix} 
+ 
\begin{pmatrix}
b_1 & c_1 \\
b_2 & c_2
\end{pmatrix} 
\begin{pmatrix}
x \\
y
\end{pmatrix}
\] (2.11)

Where \(x\) and \(y\) are the local Cartesian coordinates, and the vector \((\dot{x}, \dot{y})\) is the tangential velocity or shear stress. The coefficients \(a_1\) and \(a_2\) and those in the 2x2 matrix on the right hand side of the equation are the constants defined in Equation 2.12.

\[
\begin{pmatrix}
a_1 \\
b_1 \\
c_1
\end{pmatrix} = 
\begin{pmatrix}
1 & x_1 & y_1 \\
1 & x_2 & y_2 \\
1 & x_3 & y_3
\end{pmatrix}^{-1} 
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{pmatrix}
\] and

\[
\begin{pmatrix}
a_2 \\
b_2 \\
c_2
\end{pmatrix} = 
\begin{pmatrix}
1 & x_1 & y_1 \\
1 & x_2 & y_2 \\
1 & x_3 & y_3
\end{pmatrix}^{-1} 
\begin{pmatrix}
\dot{y}_1 \\
\dot{y}_2 \\
\dot{y}_3
\end{pmatrix}
\] (2.12)

The Jacobian matrix is the 3x3 matrix on the right hand side and the vectors on the farthest right are the 2D components of velocity at each point in the triangle. Next, by differentiating Equation 2.11 with respect to time and algebraically manipulating the results, two second order non-homogeneous ordinary differential equations are generated in Equation 2.13.

\[
\ddot{x} - (b_1 + c_2)\dot{x} + (b_1 c_2 - b_2 c_1) x = (a_2 c_1 - a_1 c_2)
\] and

\[
\ddot{y} - (b_1 + c_2)\dot{y} + (b_1 c_2 - b_2 c_1) y x = (a_1 b_2 - a_2 b_1)
\] (2.13)

The solutions to these kinds of equations are found in most math texts on differential equations. The eigenvalues of the system of equations in Equation 2.13 are given by \(\lambda^2 - (b_1 + c_2)\lambda + (b_1 c_2 - b_2 c_1) = 0\).

The discriminant \((\Delta)\) of the Jacobian is now calculated using Equation 2.14. Where \(tr\) is the trace of the Jacobian \((b_1 + c_2)\), and \(det\) is the determinant \((b_1 c_2 - b_2 c_1)\).

\[
\Delta = tr^2 - 4(det)
\] (2.14)
Processing stops for that triangle if the discriminant is negative, which occurs when the eigenvalues are complex numbers. If the discriminant is non-zero, the eigenvectors of the Jacobian are calculated and the eigenmatrix is assembled in Equation 2.15.

\[
\begin{pmatrix}
  x(t) - x_{cp} \\
  y(t) - y_{cp}
\end{pmatrix} =
\begin{pmatrix}
  \xi_1 & \eta_1 \\
  \xi_2 & \eta_2
\end{pmatrix}
\begin{pmatrix}
  \alpha e^{\lambda t} \\
  \beta e^{\mu t}
\end{pmatrix}
\] (2.15)

Where \( \lambda \) and \( \mu \) are the eigenvalues of the Jacobian matrix, and \( (\xi_1, \xi_2)^T \) and \( (\eta_1, \eta_2)^T \) are the eigenvectors. It is important at this point to scale the eigenvectors such that the determinant of the eigenmatrix equals 1. The coordinates of the critical points \( x_{cp} \) and \( y_{cp} \) are then calculated using Equation 2.16.

\[
x_{cp} = \frac{a_2 c_1 - a_1 c_2}{b_1 c_2 - b_2 c_1}
\]

\[
y_{cp} = \frac{a_1 b_2 - a_2 b_1}{b_1 c_2 - b_2 c_1}
\] (2.16)

Once the coordinates of the critical points have been determined, the triangle is projected onto the phase plane by transforming each vertex into canonical coordinates \( (X, Y) \) using Equation 2.17.

\[
\begin{pmatrix}
  X \\
  Y
\end{pmatrix} =
\begin{pmatrix}
  \xi_1 & \eta_1 \\
  \xi_2 & \eta_2
\end{pmatrix}^{-1}
\begin{pmatrix}
  x \\
  y
\end{pmatrix} =
\begin{pmatrix}
  \alpha e^{\lambda t} \\
  \beta e^{\mu t}
\end{pmatrix}
\] (2.17)

Finally, the phase portrait for each point is determined using the eigenvalue tests shown in Figure 2.9 [28]. Only phase portraits that are saddles, or repelling/attracting nodes are considered at this point. If the local phase portrait is a saddle or a repelling node, determine where \( X \) term in Equation 2.17 equals zero. This is then converted back into 2D Cartesian coordinates as an attachment point. If the local phase portrait is a saddle or attracting node, determine where \( Y \) term in Equation 2.17 equals zero. This is converted back into 2D Cartesian coordinates as a separation point. The respective points are then connected into semi-continuous lines.

This algorithm appears to accurately extract separation and attachment lines, but like most extraction algorithms it does extract some false lines as well. In fact, this algorithm tends to produce even more erroneous extractions than the PV algorithm. The authors mention that this problem occurs “when flow separation/attachment is relatively weak and becomes diffused over
Figure 2.9: Three phase portraits have tangent curves that asymptotically converge on an axis in the phase plane. Triangles that straddle the axes contribute line segments to either separation or attachment lines [28].

several cells. This causes the phase plane algorithm to either detect multiple ghost lines or leave gaps.” Similar to the PV algorithm, one cannot take these results alone as being completely accurate because of all the false extractions.

2.3 Subjective Logic

Subjective logic is a mathematics-based logic system that forms separate opinions on a given situation and accounts for uncertainty in a system [29]. These opinions and uncertainty are then combined into final human recognizable opinion. For example, what if you were asked if your favorite football team was going to win its game today? Your answer would probably not be a confident “yes” or “no” but would depend on some given variables. If the opposing team’s starting quarterback is injured, I would be more confident that my team would win. If my team’s starting quarterback was injured, I would probably say my team would lose the game. All the events that affect the outcome of the game (weather, hometown crowd, etc.) often do not have a strict if/then occurrence. Subjective logic accounts for all these variables and provides a mathematical solution to a question that can be difficult to answer with absolute certainty. Since it is not likely that the opposing teams’ starting quarterback will be injured since their team is very experienced and has won all their games so far, there is a low belief that my team will win the game and a high disbelief that my team will win. But there is some uncertainty as to the outcome because some unexpected
things might happen. These three values of belief, disbelief, and uncertainty make up the subjective logic final opinion (see Equation 1.1) on who will win the game.

A numerical value is assigned to each component of the final opinion tuple, allowing the final opinion to formulate an exact answer. The sum of each member of the final opinion tuple must equal one, as shown in Equation 2.18 [14]. With the final opinion tuple equaling one, uniformity in an opinion is maintained, meaning each opinion is on the same scale. Equation 2.18 can mathematically determine the final opinion on each extraction.

\[ b + d + u = 1 \]  

(2.18)

2.3.1 Belief Triangle

Jøsang [14] used Figure 2.10 as a simple example to define the final opinion on an extraction if \( b = 0.40, \ d = 0.10, \ u = 0.50 \) and \( a = 0.60 \). Here \( b \) is belief, \( d \) is disbelief, \( u \) is uncertainty, \( a \) is the atomicity, and \( \omega_x \) is the final opinion. \( \omega_x \) is a function of the four different values \( b, d, u, \) and \( a \) and is defined in Equation 2.21 and/or 2.25 (depending on the problem at hand). The \( b,d,u \) lines run from the mid-point of one edge to the opposite vertex indicated by the “Belief” (\( b \)) line, “Disbelief” (\( d \)) line, and “Uncertainty” (\( u \)) line labels. The final opinion can be located by moving along these lines. These lines are broken into ten sections, each representing one tenth of the line, and each line is labeled from 0 to 1 to denote the beginning and end of each line. To locate \( \omega_x \), we will use the disbelief line, move 0.10 (or 1 step) along the Disbelief line starting from 0. At this point, there is a dotted line perpendicular to the Disbelief line. By definition, \( \omega_x \) must lie along this line. Now move 0.50 (or 5 steps) along the Uncertainty line starting from 0. Again at this point there is line perpendicular to the uncertainty line. The location of the \( \omega_x \) is the point where the two dotted lines cross, thus the probability equals approximately 0.66. Any \( \omega_x \) can be located given two of the three belief, disbelief, and uncertainty values.

2.3.2 Probability Expectation

The probability expectation (PE) is an important value when evaluating any opinion as it defines the expected probability of a situation. Probability expectation defines what an agent “ex-
pects” the probability to be, not simply straight probability. \( PE \) describes this expected probability of an outcome based on its belief, disbelief, and uncertainty. It is calculated using Equation 2.19. This equation takes into account the entire belief tuple and some of the uncertainty tuple. Uncertainty accounts for the unknowns of an outcome. Some of the unknowns will positively affect the outcome, while some unknowns will negatively affect it. The \( a \) in Equation 2.19 is the atomicity. Atomicity defines how much of the uncertainty will positively affect the final outcome.

\[
PE = b + au
\]  (2.19)

The horizontal line in Figure 2.10 is the probability axis that contains all probability expectation values ranging from 0 to 1. The director is the dashed line that extends from the top of the triangle and intersects the probability axis. This point of intersection denotes the value of the atomicity. In this case, atomicity is equal to 0.60. The projector line is the dashed line that begins at the point \( \omega_x \), and extends to intersect the probability axis. This point of intersection denotes the probability expectation value. It should be noted that the projector is always parallel to the director. Going back to the previous example, \( \omega_x \), the probability expectation value becomes 0.70.

A common assumption in subjective logic is to always set the atomicity equal to 0.50. When the atomicity is 0.50, the director in Figure 2.10 will be normal to the probability axis. In this research the \( PE \) will be evaluated using this assumption. Thus, Equation 2.19 becomes:

\[
PE = b + \frac{1}{2}u
\]  (2.20)
Mortensen [12] gave a great example to better illustrate this point: “when a school class starts and most grades given in the class are B grades, the probability that student 1 will get a B grade is higher than probabilities that student 1 will get any other grade. What if student 1 has a history of getting good grades and is in a subject where he/she excels? This information does not change the probability that student 1 will get a B grade but it can change the expected probability. If it is input into subjective logic the expected probability that student 1 will get an A grade could be higher than the expected probability that student 1 will get a B grade.” With the information about a situation (unknowns, etc.) an agent can expect the probability of an outcome to be higher or lower than the standard probability.

2.4 Trust Networks

Now that the feature extraction algorithms have been chosen and defined, the intelligent agents need to be designed to include the outputs of each algorithm and determine a way to combine them. In this research, the intelligent agents combine the outputs in the form of a trust network. Trust networks [29] provide a means to assign a numerical value to trust and pass that from one individual to another. Figure 2.11 shows an example of a simple trust network where individual A seeks an opinion on individual C. Individual A does not know individual C, but has trust in individual B, and individual B has trust in individual C. Therefore individual A can gain a “derived trust” in individual C through individual B. In this research, each individual is a software agent that gains trust based on the trust of other agents. This is subjective logic at its core.

![Figure 2.11: Simple trust network showing A’s derived trust in C from B [12].](image)
2.4.1 Subjective Logic Operators

In trust networks, two different operators are needed to accurately transfer trust: the discounting operator and the consensus operator. The discounting operator is defined by Jøsang [13] and is used when the trust network lies along a straight path such as that shown in Figure 2.11. The discounting operator uses the symbol $\otimes$ giving Equation 2.21. In this equation, the superscripts denote an agent having the trust, and the subscripts denote the agent or piece of information on which the trust is based. Equation 2.21 represents the trust that $A$ has in $C$ based on the linear combination of trust that $A$ has in $B$ and the trust that $B$ has in $C$ (see Section 2.4). Following the derivations provided by Jøsang, the belief, disbelief, and uncertainty that $A$ has in $C$ can be calculated using Equations 2.22 - 2.24.

$$\omega_A^C = \omega_B^A \otimes \omega_C^B$$  \hspace{1cm} (2.21)

$$b_A^C = b_B^A b_C^B$$  \hspace{1cm} (2.22)

$$d_A^C = b_B^A d_C^B$$  \hspace{1cm} (2.23)

$$u_A^C = d_B^A + u_B^A + b_B^A u_C^B$$  \hspace{1cm} (2.24)

The consensus operator is defined by Jøsang [14] and is used when one agent holds two opinions on the same piece of information and needs to combine them into one single opinion. The consensus operator uses the symbol $\oplus$ giving Equation 2.25. Here, $X$ and $Y$ are the agents having trust, and $Z$ is the opinion. The trust that $X$ has in $Z$ is represented $\omega_X^Z$, and the trust that $Y$ has in $Z$ is denoted $\omega_Y^Z$. These two separate opinions are combined using Equation 2.25 to describe the trust that $X$ and $Y$ have in $Z$. Again, following the derivations provided by Jøsang, the belief, disbelief, and uncertainty that $X$ and $Y$ have in $Z$ can be calculated using Equations 2.26 - 2.28.

$$\omega_X^Z = \omega_X^Y \oplus \omega_Y^Z$$  \hspace{1cm} (2.25)

$$b_Z^{XY} = \frac{b_Z^Y u_Z^Y + b_Z^X u_Z^X}{\kappa}$$  \hspace{1cm} (2.26)

$$d_Z^{XY} = \frac{d_Z^X u_Z^X + d_Z^Y u_Z^Y}{\kappa}$$  \hspace{1cm} (2.27)
\[
\kappa = u_Z^X + u_Z^Y - u_Z^X u_Z^Y
\]  
\hspace{1cm} (2.29)

2.4.2 Example Trust Network

An example of a more complex trust network is shown in Figure 2.12. Individual A seeks a final opinion on D, but does not possess any information about D, however both B and C have opinions concerning D. It can be seen that the two different agent opinions form two different linear paths - the left side path and the right side path. The left path would use Equations 2.22 - 2.24, but replacing the C’s with D’s. Similarly, the right side path would use the same equations replacing the B’s with C’s. These two paths each create their own opinion and thus need to be combined into one final opinion D. This combination can be done using Equations 2.26 - 2.29, where X and Y are the opinions just created, and replacing the Z’s with D’s. After these have been calculated, individual A has now has a final opinion on D based on its agents’ opinions.

![Figure 2.12: Trust network showing A’s derived trust in D from B and C requiring both consensus and discounting operators.](image)
2.5 Current Dataset Post-Processing Concepts

The intelligent architecture created by Mortensen [12] for vortex core line detection and used in this research for shock wave and separation and attachment lines will be incorporated into the CAFÉ concept. This section will briefly discuss the CAFÉ concept, as well as a similar concept called Evita.

2.5.1 CAFÉ

The CAFÉ concept is an agent-based architecture that was designed to be an after-market plug-in for CFD packages (FLUENT, Overflow, Openfoam, etc.) for decision support in extractions applications. CAFÉ uses multiple feature extraction algorithms to increase the accuracy of extracted features and determines which extractions are real and which are not based on feature characteristics.

Figure 2.13 shows a layout view of the CAFÉ concept. Enormous amounts of data are generated when performing a large high-fidelity physics-based simulation. Features are mined from the data using various algorithms contained in agents. Feature extraction changes the complex data into a more simple dataset using the discounting and consensus operators mentioned in Section 2.4.1. In addition to simplifying the variables associated with each extraction, the size of the dataset is greatly reduced because only those associated with the feature of interest are communicated to the other agents.

A current limitation to feature extraction from massive CFD datasets is that the time it takes to extract features is often long enough to hinder post-processing rather than help it. Typically, feature extraction is done after a CFD simulation is converged. This requires extra computation time after a simulation is complete. Intelligent agents aid in extracting flow features concurrent with a running simulation and as a post-processing step. CAFÉ’s ability to perform concurrent analysis (during the simulation run time) can eliminate excessive post-processing storage needs and time by targeting specific regions where features have been detected. This concurrency is performed by exporting datasets at user-specified frequencies during the convergence process, then running the extraction and subjective logic algorithms on that dataset using a separate processor. This process is repeated until convergence is attained. In addition to concurrency, the software
agents allow CAFÉ to work using multiple processors. This process is performed alongside the CFD software programs, taking full advantage of the parallel processing environment.

### 2.5.2 Evita

The closest program to CAFÉ is a concept called Evita [30] which is designed specifically for large data set exploration. Evita consists of three components: an off-line preprocessor, a server, and a client. While CAFÉ uses known feature extraction techniques, Evita provides two alternate techniques for feature mining: point classification and aggregate classification. Point classification verifies points as features before they are aggregated, while aggregate classification does the opposite. It aggregates points before they are verified and then verifies the aggregate. Evita also provides a method that uses wavelet transforms to focus on the areas of high interest. Evita uses one feature extraction algorithm to create a binary classification of the flow domain which is then given to supplied data mining algorithms to classify, cluster, and categorize identified features before they are presented to a researcher.
The biggest difference between CAFÉ and Evita is that CAFÉ uses two separate feature extraction algorithms to find an opinion on each extraction, while Evita uses only one to verify and aggregate the features of interest. Another difference is the manner in which erroneous extractions are removed. CAFÉ’s algorithms filter erroneous extractions based on the strengths and weaknesses of each algorithm, while Evita uses wavelet transforms to filter out erroneous extractions.
CHAPTER 3. GENERAL NUMERICAL FEATURE EXTRACTION METHOD

In this chapter, the general numerical method for feature extraction from CFD datasets using subjective logic will be discussed. Chapter 4 will discuss how this general method is specifically applied to the extraction of shock waves and separation and attachment lines. The general method was originally developed by Mortensen [12] and involves the following steps:

1. Extract the feature of interest using the feature extraction algorithms.
2. Filter obviously erroneous features.
3. Create agent opinions for each of the extracted datasets.
4. Combine the agent opinions into one final opinion.
5. Aggregate one final dataset from the final opinions.

3.1 Extracting and Filtering Features

Once a CFD simulation has been calculated, the resultant datasets are run through the extraction algorithms contained in the software agents. A much smaller, more manageable dataset is created containing only the features of interest, called the feature sets. For example, if three extraction algorithms were used to extract certain features, then three different feature sets would be created (one per algorithm). Because each algorithm calculates the features differently, each feature set could be markedly different or slightly different, but somewhat similar features should be extracted because they came from the same dataset.

The shock wave and separation and attachment line detection algorithms are point-based schemes that look at each node within the domain and attempt to determine if they exist in the region of a feature. However, during the programing of this process it was determined that it was rare for a shock wave or separation or attachment line to pass exactly through a node in the domain. Figure 3.1 shows a 2-dimensional shock wave representation of cells extracted by the LH method colored by normal Mach number. It can be seen that the normal Mach number drop associated
with the shock wave occurs between two nodes (where the normal Mach number changes from red to blue). In order to extract points along the shock wave rather than extracting regions surrounding the shock wave, it was determined that interpolation was required.

In order to accomplish the interpolation, an algorithm was developed that searches for the point at which the normal Mach number equals unity as the flow moves from upstream to downstream of the shock wave. This is done by finding computational nodes where the normal Mach number decreases from above 1 to below 1 across the computational nodes. Once these nodes are located, linear interpolation is used to calculate the spatial location where the normal Mach number equals unity. After these points have been located, they are stored separately from the original dataset, resulting in a new feature set. The interpolation not only determines the spatial location of the extractions, but it also interpolates values of interest needed for future calculations such as pressure, velocity, density, etc. These interpolated values are then stored along with the spatial location of each extraction.
3.2 Filtering Extracted Features

Since each algorithm calculates the same features differently, there will usually be a difference in each feature extracted. Along with these differences, many extraneous extractions are made, some being very obvious. For example, Figure 3.2 shows the shock wave extracted by the MRV algorithm around a blunt fin [31]. The blunt fin simulation consists of airflow over a flat plate with a blunt fin rising normal to the plate. The inlet Mach number is 2.95, and a Reynolds number based on fin width of 2.1e6. A separated shock wave develops in front of the fin.

Figure 3.2a shows many shock wave extractions, while Figure 3.2b shows the real shock wave after filtering. In this simulation, the erroneous extractions were removed by only keeping those with high pressure gradient values. Computational time and resources can be saved by filtering out these erroneous extractions before continuing with other calculations by using a simple threshold criteria. These criterion can vary per feature, but should be feature specific. For example a shock wave extraction with a very low local pressure gradient is obviously not a real shock and should be eliminated. These filtering criterion should be set low enough to filter out only the obviously erroneous detections, but let the rest of the extracted features through. The agents work better when the obviously erroneous extractions are eliminated first.

Figure 3.2: Comparison of filtered (a) and unfiltered (b) extracted shock wave feature sets.
3.3 Creating Opinions of Extracted Features

After the features have been extracted and run through a simple filter to remove the obviously erroneous extractions, the agents can begin to develop opinions on the extracted features. While developing these opinions, the belief, disbelief, and uncertainty values are created within each agent adhering to Equation 2.18. The agents form their opinions based on predefined parameters such as algorithm strengths and weaknesses (high or low pressure and density gradients, etc.) known to affect the extraction of each feature.

Once the agents have finished calculating an opinion for each extraction, they need to be combined into one final opinion for each agent. This final opinion is calculated using the discounting and consensus operators discussed in Section 2.4.1. The final opinion consists of three separate values: belief, disbelief, and uncertainty. These values are used in calculating the probability expectation of each feature (see Equation 2.20). Probability expectation can be easily quantified as follows: if a feature has a high probability expectation it will have a high belief, as well as low disbelief and uncertainty. Conversely, if a feature has a low probability expectation it will have high disbelief and uncertainty, but have a low belief.

With subjective logic there is no black or white, yes or no answer. This circumstance exists with feature extraction. For example, if a feature has a belief of 0.82, a disbelief of 0.09, and an uncertainty of 0.09, is this a real or false extraction? Subjective logic can be used to develop an opinion on this feature when there is no clear-cut, yes or no answer. Using this example, the belief tuple is defined as follows: \( \omega = (0.82, 0.09, 0.09) \). Using Equation 2.20, the probability expectation becomes 0.865, which means that for this case the feature is probably real. Bear in mind that the 0.865 value being labeled as “probably real” is not a concrete definition, and may change depending on the other probability expectation values within your current simulation.

It can be difficult to set a definite probability expectation value in which the feature in question is “real” or not. Obviously a probability expectation value of 0.99 is probably real, and a value of 0.6 is probably not, but in between it becomes the choice of the user to determine what is “real” and what is not based on the probability expectation values. Higher probability expectation values mean the feature is probably real, however lower values mean the feature is probably not.
### 3.3.1 Intelligent Agent Structure

A visual example of the agent structure used to calculate opinions on each extracted feature is shown in Figure 3.3. AA represents an agent algorithm; and each AA contains one feature extraction algorithm with the subscripts 1 and 2 denoting the two separate algorithms. MA is the master agent that eventually combines the opinions created from the separate AA’s. R refers to an extraction region within the computational domain that is being investigated by the AA’s to determine whether it is probable. If the feature under investigation is a separation or attachment line, then R refers to a point contained in the extracted lines. Similarly, for a shock wave R refers to a point within the extracted shock surface. Ultimately MA will develop belief, disbelief, and uncertainty values that can be used to calculate a final opinion on a feature inside R.

![Figure 3.3: Representation of a two algorithm AA structure.](image)

It can be seen that Figure 3.3 has the same architecture as Figure 2.12. It was discussed in Section 2.4.2 that this architecture is a combination of two separate linear opinion branches that needed to be combined. Using this same idea and combining the opinions as defined by Jøsang [14], Equations 3.1- 3.4 are used to find the belief, disbelief, and uncertainty of a feature. The subscripts denote the information under investigation, and the superscripts denote the opinion being formed. These equations were derived by Jøsang and Mortensen see references [14] and [12] for a more detailed derivation.
\[ b^M_A = \frac{(b^M_A \cdot b^{AA_1}_R)(d^{MA}_{AA_1} + u^{MA}_{AA_1} + b^{MA}_{AA_2}u^{AA_2}_R) + (b^M_A \cdot b^{AA_2}_R)(d^{MA}_{AA_1} + u^{MA}_{AA_1} + b^{MA}_{AA_2}u^{AA_1}_R)}{\kappa} \]  \hspace{1cm} (3.1) \\
\[ d^M_A = \frac{(b^M_A \cdot d^{AA_1}_R)(d^{MA}_{AA_1} + u^{MA}_{AA_1} + b^{MA}_{AA_2}u^{AA_2}_R) + (b^M_A \cdot d^{AA_2}_R)(d^{MA}_{AA_1} + u^{MA}_{AA_1} + b^{MA}_{AA_2}u^{AA_1}_R)}{\kappa} \]  \hspace{1cm} (3.2) \\
\[ u^M_A = \frac{(d^{MA}_{AA_1} + u^{MA}_{AA_1} + b^{MA}_{AA_1}u^{AA_1}_R)(d^{MA}_{AA_2} + u^{MA}_{AA_2} + b^{MA}_{AA_2}u^{AA_2}_R)}{\kappa} \]  \hspace{1cm} (3.3) \\
\[ \kappa = (d^{MA}_{AA_1} + u^{MA}_{AA_1} + b^{MA}_{AA_1}u^{AA_1}_R) + (d^{MA}_{AA_2} + u^{MA}_{AA_2} + b^{MA}_{AA_2}u^{AA_2}_R) \]
\[ - (d^{MA}_{AA_2} + u^{MA}_{AA_2} + b^{MA}_{AA_2}u^{AA_2}_R)(d^{MA}_{AA_1} + u^{MA}_{AA_1} + b^{MA}_{AA_1}u^{AA_1}_R) \]  \hspace{1cm} (3.4)

Even though Figure 2.12 is specified to contain only two AA’s, any number of AA’s could be implemented into the structure allowing for the use of more feature extraction algorithms. Figure 3.4 shows a visual representation of how additional AA’s could be implemented into the existing two algorithm AA structure. For example, if another shock detection algorithm was defined, it could easily be added to the current structure. Adding other algorithms could be greatly beneficial because features that were missed and not extracted by the other AA’s could now be found by the new addition. It can be seen that adding the extra agent AA_N does not affect the other agents, but will only add to the confidence of the final opinion. This process is fairly flexible because as many as N AA’s could be used or as few as one. With the addition of another AA, Equations 3.1- 3.4 would need to be altered to account for all of the AA’s. Even though adding more AA’s will increase the confidence in the final opinion, it can be seen that the calculations of the belief, disbelief, and uncertainty would become more complex and time consuming.

### 3.3.2 Algorithm Agent Opinions

Within this research, the MA and AA’s need to be set individually. The first ones to be set are the AA’s. The two AA structure (Figure 3.3) generates two separate extraction datasets - one from each AA. Since two separate extraction datasets were generated from the different AA’s, it is important to think of these as completely separate, meaning that initially they do not affect each other. A simple example of this is shown below in Figure 3.5 where the two lines represent two
different separation or attachment lines. In this case $\text{AA}_1$ represents the feature from the original dataset 1 (shown as the top black line). Now consider that $\text{AA}_2$ represents a similar feature from the original dataset 2 (shown as the bottom red line). Even though these extracted datasets are similar they are still different.

These two feature sets can now be used to calculate the opinions of $\text{AA}_1$ and $\text{AA}_2$ for feature set 1. To do this $\text{AA}_1$, needs to form an opinion on every point defining the line in feature set 1. Also, $\text{AA}_2$ needs to form an opinion on every point defining the line in feature set 1. $\text{AA}_2$ needs an opinion on feature set 1 because Equations 3.1- 3.3 require this calculation in the right hand side of the equations. Each AA forms an opinion on every point in feature set 1, giving what appears to be a mutually contradictory method for defining the AA's. This is all taken into account by defining the $\text{AA}_1$ points from feature set 1 as the extracting algorithm ($\text{AA}_E$). Since $\text{AA}_2$ does not extract points from feature set 1, it is termed the non-extracting algorithm ($\text{AA}_{NE}$).
Now consider the feature set 2 extracted by the AA$_2$. Once again both AA’s need to form an opinion on each point contained in feature set 2. However, this time the naming convention will be slightly different. Since AA$_2$ is the extracting algorithm, it will be deemed as such (AA$_E$), and AA$_1$ becomes the non-extracting algorithm (AA$_{NE}$). After these two extracted feature sets have an opinion formed on them by each AA, there are no more AA’s to define.

This mutually contradictory method for defining the AAs works with the simple 2 AA structure (Figure 3.3) as well as it does with multiple AAs (Figure 3.4). Each point within the respective feature set will contain one AA$_E$ opinion and as many AA$_{NE}$ opinions as there are other AA’s. Now that this structure has been defined, the AA$_E$ and AA$_{NE}$ need to be defined.

### 3.3.3 Extracting Algorithm Agent

The belief tuple that defines the extracting algorithm agent (AA$_E$) contains three variables: belief, disbelief, and uncertainty. Belief is set by the strengths of each algorithm, disbelief is set by the weaknesses of each algorithm, and uncertainty is set by a physical flow feature characteristic (see Table 3.1). Remember that these values must sum to unity according to Equation 2.18.

<table>
<thead>
<tr>
<th>AA$_E$ Defined by</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>AA$_E$ Strengths</td>
</tr>
<tr>
<td>$d$</td>
<td>AA$_E$ Weaknesses</td>
</tr>
<tr>
<td>$u$</td>
<td>Feature Characteristics</td>
</tr>
</tbody>
</table>

The belief is set based on whether or not the extraction in question contains that strength. For example Kenwright [28] suggests that the phase plane algorithm adequately extracts separation and attachment lines in the presence of a high local pressure gradient and when the lines are nearly straight. These two strengths are then passed to the AA$_E$ and a high belief is set if both cases are true. However if neither of the strengths are present, a low belief value will be set. The belief will be calculated at each extracted point in each feature set.

The disbelief is similar to the belief except it accounts for the weaknesses of each algorithm, meaning situations where a false extraction may have been made. The weakness characteristics
may be opposite of the strengths. For example, the phase plane algorithm does not perform well when the extracted line is very curved and in regions of low pressure gradient. Therefore, if an extraction is very curved and does not display a large local pressure gradient, the disbelief will be set high. Conversely, if the line is straight and a high local pressure gradient exists then the disbelief will be set low.

Uncertainty is set from physically known characteristics of the specific feature of interest. For example, it is well documented that as fluid flow crosses a shock wave that the local flow velocity will decrease. It is also known that depending on the angle of the shock and the upstream Mach number, a downstream Mach number can be calculated. If the downstream point just after extraction is close to equaling the theoretically calculated value, then the extraction will have a low calculated uncertainty, or the AA will be quite certain the extraction is a real shock wave. Conversely, if the two downstream values are not similar, a high uncertainty will be calculated. There are many uncertainty parameters that can be used as inputs for the agents. The only criterion is that the parameter be quantifiable, not a simple yes or no answer. This criteria also applies to belief and disbelief.

One of the advantages to setting the AA\textsubscript{E} in this manner is that it can be easily extended to accept multiple additional agent algorithms. For example, if it was found that an additional algorithm for shock detection exists, it may be beneficial to add it to the existing structure. As long as the algorithm has defined strengths and weaknesses and contains physical information about the feature, it can be added to the structure without affecting the other AA\textsubscript{E} opinions. Yet it provides more confidence in the resultant probability expectation.

### 3.3.4 Non-Extracting Algorithm Agent

Once all the AA\textsubscript{E} opinions have been set, the AA\textsubscript{NE}’s of each feature need to be defined. The belief tuple of the AA\textsubscript{NE} consists of 3 separate values. The belief is set by the extracting algorithm’s (AA\textsubscript{E}) strengths, while the disbelief is set by the extracting algorithm’s (AA\textsubscript{E}) weaknesses. The uncertainty is set by the distance between the closest extracted regions (see Table 3.2). The uncertainty parameter and calculations were originally defined by Mortensen [12].

Uncertainty is set according to the minimum distance between any point extracted by the AA\textsubscript{NE} and the point being investigated. Going back to the two AA structure example, two separate
feature sets will be generated. If the investigated point is within feature set 1, then the minimum distance would be calculated as the distance from its closest point in feature set 2. If a point is a real feature it will exist in both the $AA_E$ and $AA_{NE}$ agents. When the distance between the two points is small, a low uncertainty value is calculated. Conversely, if the two points are far from each other, a higher uncertainty value is calculated.

### 3.3.5 Master Agent

The master agent (MA) can be thought of as the governing or controlling agent because it has the most influence on the believability of the extracted features. The job of the MA is to take the AA opinions and generate one final opinion of each extraction. The belief tuple of the MA is based on the thought that as a feature converges, it will settle on a final spatial location. This characteristic is quantitatively defined through the feature displacement (FD) parameter. Feature displacement was originally defined by Mortensen [12], and is a measure of the distance a feature moved from one iteration to another. Feature displacement is calculated using Equation 3.5 where $i$ refers to the current iteration and $i-1$ refers to the previous iteration which could be ten to thousands of iterations prior.

$$FD_i = P_i - P_{i-1} \tag{3.5}$$

An example of feature displacement is given in Figure 3.6 where the same line feature is extracted at two different time steps – 200 and 300 iterations. If a similar point is considered in each line, the distance between the two divided by the reference length will be defined as the feature displacement. Each point in the 300 iteration line has a feature displacement value based on the 200 iteration line.
Figure 3.6: Representation of the same feature extracted at 200 and 300 iterations, where feature displacement measures the distance between them.

Another parameter used in the MA is the change in feature displacement ($\Delta FD$), and was originally defined by Mortensen [12]. This value refers to the absolute value of the slope of the feature displacement vs. iterations. $\Delta FD$ is calculated using the following:

$$\Delta FD_i = \frac{|FD_i - FD_{i-1}|}{\text{# of iterations}}$$ \hspace{1cm} (3.6)

The belief tuple of the MA is defined as follows: belief is set by the feature displacement and change in feature displacement, disbelief is set by the feature displacement, and uncertainty is set by the change in feature displacement. The MA belief tuple is summarized in Table 3.3.

Table 3.3: Definition of AA$^{NE}$ belief tuple.

<table>
<thead>
<tr>
<th>MA</th>
<th>Defined by</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>$FD &amp; \Delta FD$</td>
</tr>
<tr>
<td>d</td>
<td>$FD$</td>
</tr>
<tr>
<td>u</td>
<td>$\Delta FD$</td>
</tr>
</tbody>
</table>

If the feature moves slightly between iterations (low feature displacement), and the previous iterations’ feature movement is also small (low change in feature displacement), then the belief value will be high. This suggests that the feature will not move much during the upcoming iterations. Conversely, if the feature moves a large distance between iterations, then the belief will be small. If the feature moves a large distance between iterations (high feature displacement) and a large discrepancy exists between the current and previous feature displacements (high change in feature displacement), then a high disbelief will be calculated for that feature. This suggests that
the feature has not yet converged to a spatial location. The uncertainty will also be high when a large change in feature displacement exists. This suggests that the MA is uncertain about the feature because the feature could potentially move substantially during future iterations.

3.4 Aggregating Final Feature Set Opinions

This process is not currently automated in the program, but can be performed manually by visual inspection. Aggregation will be implemented within the later stages of the CAFÉ project. This can be accomplished manually by creating a search criterion that locates the same feature in each feature set and then selecting only the feature with the highest probability expectation. The search criterion could be performed by parsing through the minimum distance values calculated in Section 3.3.4 and comparing those with the smallest minimum distance values. The features that do not have a similar feature extracted by a separate algorithm may be selected or discarded based on some combination of their belief tuple values and their probability expectation.
CHAPTER 4. SHOCK WAVE FEATURE EXTRACTION METHOD

In Chapter 3, the general method for feature extraction and opinion generation was discussed. This method could theoretically be used for any type of feature. This chapter will focus on how to use this method specifically for the extraction of shock waves.

4.1 Shock Wave Characteristics

Shock wave characteristics provide useful inputs to the agents that can help them make decisions about the expected probability of each feature. Even though there are many ways to characterize a shock wave, the specific shock wave characteristics used in this research are shock strength, the Parallel Normal Vectors parameter, and the normal Mach number value.

4.1.1 Shock Strength

Shock wave strength is the measure of the static pressure and density jump across a shock wave. The larger the jump, the stronger the shock wave. This definition will allow for a simple yet effective measure of the shock strength that can be used later in defining the strengths and weaknesses of each algorithm.

4.1.2 Parallel Normal Vectors

The Parallel Normal Vectors parameter was developed by Hesselink, Levy, and Batra, who suggest to “reject all the triangles in the shock surface that have normals that are not aligned with the pressure gradient.” [32] This parameter was disregarded by Lovely-Haines because if used as a filter it would eliminate the extractions in front of the shock wave. Because of this, the researcher would no longer have the added information provided by the thickness of the “shell” surrounding the shock.
Even though Lovely and Haimes dismiss this parameter it is used in this research as an additional strength/weakness of each algorithm for two reasons. First, the thickness of the “shell” surrounding the shock gives the researcher an idea of the fineness or coarseness of the local mesh is that modeling the shock. The larger the gap between the “shell” walls, the more coarse the local mesh. A large gap between the shell walls would suggest to the researcher the need to locally refine the mesh to obtain more accurate results. No scale will be given here to define how much of a gap should require mesh refinement because this determination will vary per application and will thus remain the responsibility of the user.

Secondly, in the presence of a real shock wave, the cell normals of the extractions should be nearly parallel to the pressure or density gradient. This parameter will be useful in helping the researcher determine the expected probability of the shock wave extractions while still retaining information about the local grid resolution.

Both of these reasons justify our inclusion of this parameter in determining the expected probability of the extractions because it provides additional valuable information needed to make an accurate decision both visually and computationally.

4.1.3 Normal Mach Number

The Normal Mach Number value is defined as the dot product between the local Mach vector and the local density gradient. A shock wave exists when this value is approximately equal to one. Since this parameter was used in the LH algorithm to define a shock wave, it cannot be used to help determine the expected probability of an extraction generated by using the LH method.

Ma-Rosendale-Vermeer suggest using this value as an effective filtering technique. This research will not use this value for filtering, but will use it to help determine the probability of an MRV extracted shock wave. This will eliminate the issues that Ma-Rosendale-Vermeer encountered in picking a suitable range of thresholding.

4.2 Extracting and Filtering Shock Waves

Two different shock wave extraction algorithms are used in this research, the Lovely-Haimes algorithm (LH) and the Ma-Rosendale-Vermeer algorithm (MRV). These algorithms were de-
scribed in Sections 2.1.2 and 2.1.3 respectively. Two separate feature sets were generated from the algorithms (one each), with each feature set consisting of point clouds that form the shape of the shock wave. Each extracted point is not connected to the others, but can later be combined into a 3-dimensional surface using a 3D-Delaunay interpolator if desired.

Figure 4.1 details airflow past a swept Onera M6 airfoil [33] at an inlet Mach number of 0.8395 with flow moving from left to right. This simulation will be discussed in greater detail in Section 6.2. Figure 4.1a shows a slice of the resultant velocities colored by Mach number. Shocks can often be located in such a figure by finding large drops in velocity (when the supersonic flow drops below Mach=1). It can be seen that two shocks exist, one near the tip and one mid-chord. These shocks are characterized by the local sonic flow decreasing to subsonic flow within a very short distance. Figure 4.1b shows the shock wave extractions from the MRV algorithm. Some obviously erroneous extractions reside below the leading edge of the wing and extend upstream. A real shock wave is not likely to develop here because the freestream flow is subsonic, hence a bow shock is not expected. Also, shock waves physically do not extend upstream, whereas these extractions do, leading them to be unlikely. Three different filters will be discussed to eliminate these erroneous points: pressure gradient, normal parallel vectors, and normal Mach number.

Figure 4.1: Representation of a swept Onera M6 wing dataset, (a) a slice of the dataset colored by Mach number and (b) the unfiltered extracted dataset generated by the MRV algorithm, many erroneous extractions exist as a result.
Comparing the CFD results to the extracted results can lead to a fairly accurate determination of the location of the shock waves. However, doing so can be misleading when judging which are real shock waves and which could be false. For example, a stagnation region on an object in a flow could easily be mistaken for a shock wave. In order to obtain this information, the extraction and subjective logic procedure will be used.

4.2.1 Pressure and Density Gradient Filter

The pressure and density gradient filters remove all extractions that contain small pressure or density gradient values. This filter eliminates the lower values because a shock is not likely to exist within these low regions. The pressure gradient filter applies only to the LH algorithm, and the density gradient filter only applies to the MRV algorithm.

Selecting a suitable minimal filtering threshold can be a tricky task because if the filter is set too low, many false extractions will remain. If it is set too high, part of the actual shock could be eliminated. If a value for pressure gradient or density gradient were chosen by taking a certain percentage of the maximum gradients, it may work for some situations but not others. For example, if a shock resides in a shock tube, it is pretty safe to assume that the pressure gradient filter could be set fairly high since there will be few other regions of high gradients. However if a shock wave is forming externally on a wing, the highest gradients may exist at the leading edge stagnation points rather than in the shock region, thus the filtering value would end up removing sections of the actual shock wave.

A better option used to determine a suitable filtering limit is the geometric mean. The geometric mean is the \( n^{th} \) root of the product of \( n \) numbers, meaning it is the average of the logarithmic values of a data set, converted back to a base 10 number. This allows the program to sift through the extracted feature set and automatically determine a value that is best suited for that particular feature set, rather than just using the maximum values. Once this value is determined, all extractions that fall below this value are removed from the feature set. In this research this filter is applied inside each respective algorithm.
4.2.2 Normal Parallel Vectors Filter

The normal parallel vectors filter is based on work performed by Hesselink, Levy, and Batra [32]. The normal parallel vectors parameter is defined as a measure of how parallel the extracted cell normal vector is in reference to the local pressure gradient vector. According to Hesselink, when the cell normal and pressure gradient vectors are nearly parallel there is a strong probability that the extraction is part of the shock wave. Figure 4.2 shows a close-up of a Lovely-Haimes shock shell formed on the suction side of an Onera M6 wing (flow is moving from the right to the left). The LH shock shell was previously described in Section 2.1.2. The gray face on the left side of the figure is the inside face of the aft shock shell wall, while the red face on the right side is the outside face of the forward shock shell wall. The red vectors represent the local pressure gradient and the black arrows represent the local cell normals. It can be seen that on the left (aft wall of shock shell) the vectors are parallel (appearing to be one vector), and will be given a value near 1. The vectors on the right are clearly not parallel, so these extractions will be given a lower value.

![Figure 4.2: Representation of the normal parallel vectors parameter.](image)

The normal parallel vectors filter removes those extractions whose cell normal and velocity vectors are not close to parallel. In Figure 4.2, the extractions on the right (non-parallel portions) could be removed. In this case however, the limit for filtering should be set relatively low because
the full shock shell produced by LH is often helpful to the research. In this research, this filter is applied within each algorithm before the subjective logic process is applied.

4.2.3 Normal Mach Number Filter

The normal Mach number filter is used to eliminate those extractions whose Mach number normal to the local density gradient is not equal to unity. This filter is only applied to the MRV algorithm because it is directly used to locate a shock region in the LH algorithm. The idea of this filter is that as the local Mach vector normal to the density gradient crosses a shock, it will cross unity. However, it cannot be expected that the normal Mach number be always exactly equal to one due to CFD approximations, so the geometric mean of the normal Mach values is calculated to find a suitable filtering value. Once again, this value is calculated and filtering is performed inside the extraction algorithm.

The pressure gradient, normal parallel vectors, and normal Mach number filters were then applied to the extracted feature set in Figure 4.1b. The resultant feature set is shown in Figure 4.3. It can be seen that the extractions that are not within the expected forward and aft shock regions have been eliminated giving a clear and more accurate feature set.

4.3 Forming Opinions on Extracted Shock Waves

Now that all the feature sets have been extracted and filtered, four opinions need to be formed on each extracted point: $\omega_{MA}^{AA_1}$, $\omega_{MA}^{AA_2}$, $\omega_{R}^{AA_1}$, $\omega_{R}^{AA_2}$. Here, R refers to the extracted point, and AA$_1$ and AA$_2$ refer to the LH and MRV extraction algorithms respectively. These opinions were previously discussed in Sections 3.3.3, 3.3.4, and 3.3.5. As previously mentioned, the basic method for these calculations was previously defined by Mortensen [12] for vortex core extraction. This method was altered to accept shock wave algorithms. It should be noted that only three of these four opinions are created because $\omega_{MA}^{AA_1}$, $\omega_{MA}^{AA_2}$ are the same.

In order to create the agent opinions on each extraction, Table 3.1 needs to be quantified. Both Lovely-Haijmes and Ma-Rosendale-Vermeer discuss in their research instances where each respective algorithm works well and when they don’t. These situations will be used to define the strengths and weaknesses of each algorithm.
Figure 4.3: Extracted shock waves of the Swept Onera M6 wing dataset after pressure gradient, normal parallel vectors, and normal Mach number filters have been applied.

4.3.1 Lovely-Haimes Strengths, Weaknesses, and Feature Characteristics

The strengths and weaknesses of the LH shock extraction algorithm are shown in Table 4.1. The LH method works well when the extracted points reside in an area of high shock strength; thus a high shock strength will be used as an algorithm strength. Shock strength is commonly referred to as the measure of pressure gradient across the shock wave. A “high” pressure gradient value is subjective, however, and there is no definitive cut-off between a high pressure gradient and a low pressure gradient. Thus each extraction is normalized by the geometric mean of the extracted pressure gradient, making each extraction on the same scale. A high normalized value would generally be above approximately 0.85. The LH method also works well when the extracted points’ cell normal and local velocity vectors are nearly parallel. This is referred to as the normal parallel vectors value. The normal parallel vectors value is independent of the LH and MRV extraction algorithms and is used as either a strength or weakness for both algorithms.

The weaknesses of this algorithm are the opposite of its strengths. For example, if an extracted point possesses a weak pressure gradient it will have a low shock strength. This will decrease the believability of the opinion of that extraction. A normalized pressure gradient would
Table 4.1: Definition of AA_E belief tuple for the LH shock extraction algorithm.

<table>
<thead>
<tr>
<th>AA_E</th>
<th>Defined by</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>AA_E Strengths high shock strength, high normal parallel vectors</td>
</tr>
<tr>
<td>d</td>
<td>AA_E Weaknesses low shock strength, low normal parallel vectors</td>
</tr>
<tr>
<td>u</td>
<td>Feature Characteristics downstream Mach number comparison</td>
</tr>
</tbody>
</table>

be low if it were below approximately 0.70. Additionally, when the normal parallel vectors value is small, it will also have a negative effect on its opinion.

While there are many different physical feature characteristics that could be used to define uncertainty, only one is used in this research - the downstream Mach number comparison. This parameter comes from concepts of compressible flow. The downstream Mach number \( M_2 \) is calculated theoretically based on the upstream Mach number \( M_1 \) using Equation 4.1 [22]. This value is then compared to the CFD simulated downstream Mach number. The closer these two values are, the lower the uncertainty becomes for that point. Equation 4.1 is derived from Figure 4.4 [22] where \( \theta \) is the shock wave angle, and \( \gamma \) is the ratio of specific heats of the working fluid, \( \alpha_1 \) is the angle of the upstream flow, \( \alpha_2 \) is the angle of the flow downstream of the shock, and \( \delta \) is the angular difference between \( \alpha_1 \) and \( \alpha_2 \), also known as the deflection angle.

\[
M_2^2 = \frac{1 + \frac{\gamma - 1}{2} M_1^2}{\gamma M_1^2 \sin^2 \theta - \frac{(\gamma - 1)}{2}} + \frac{M_2^2 \cos^2 \theta}{1 + \frac{(\gamma - 1)}{2} M_1^2 \sin^2 \theta}
\]  

(4.1)

Figure 4.4: Relationship between flow directions and the shock wave angle [22].
4.3.2 Belief Tuple Value for Lovely-Haines Extracting Algorithm Agent

When LH is the extracting algorithm, each points’ AA\(_E\) belief tuple is calculated using Equations 4.2 - 4.4. The constants in these equations were originally determined by Mortensen [12] and have not been altered here. Mortensen determined these constants because they gave a good range of probability expectation values as described in Section 4.3.4.

\[ b = 0.4 \times \text{NormalAverage}_{LH} + 0.6 \]  
\[ d = -0.4 \times \text{NormalAverage}_{LH} + 0.4 \]  
\[ u = 0.5 \times \frac{M_2_{\text{calculated}} - M_2_{\text{simulated}}}{M_2_{\text{calculated}}} \]

where

\[ \text{NormalAverage}_{LH} = \frac{\text{NormalPressureGradient} + \text{NormalParallelVectors}}{2} \]

and

\[ \text{NormalPressureGradient} = \begin{cases} \frac{\text{PressureGradient}}{\text{meanPressureGradient}} & \text{if } \text{NormalPressureGradient} < 1, \\ 1 & \text{if } \text{NormalPressureGradient} \geq 1. \end{cases} \]

\[ \text{NormalParallelVectors} = \begin{cases} 1 - \frac{|PVV| - |PMV|}{|PVV|} & \text{if } |PVV| \geq |PMV|, \\ 1 - \frac{|PMV| - |PVV|}{|PMV|} & \text{if } |PVV| < |PMV|. \end{cases} \]

Where PVV is the local Parallel Vectors Value, and PMV is the product of the local normal velocity and local pressure gradients. Equations 4.6 and 4.7 are thus normalized by their own respective values so that each value will be on a scale from one to zero. A value of one means that the algorithm is operating under its strongest conditions, while a value of zero means the algorithm is operating away from its strengths. With these values being on the same scale they can be averaged to create a normalized average (Equation 4.5). This value can then be used in Equations 4.2 and 4.3.

Equations 4.2, 4.3, and 4.4 represent the belief, disbelief, and uncertainty of each point extracted by the LH algorithm. These equations are not set in stone and other variations may be
used if it is determined they give better results. If the $NormalAverage_{LH}$ is equal to one, meaning all of the LH strength characteristics are present, then $b = 1$ and $d = 0$, representing total belief. If $NormalAverage_{LH}$ is equal to zero, meaning none of the LH strengths are present, then $b = 0.6$ and $d = 0.4$. Belief is not zero because LH still extracted that point for a reason. If $NormalAverage_{LH}$ is equal to 0.5, meaning some strengths are present but not others, then $b = 0.8$ and $d = 0.2$. This suggests a fairly high belief in the extraction.

Most of the time $b + d + u \neq 1$. For example, almost always $u > 0$. When $u > 0$, then belief is held constant while disbelief and uncertainty are decreased until the condition holds. Equation 4.4 will be greater than zero at every point except where the two downstream Mach numbers exactly match. This will be rare since CFD is an approximation of flow properties.

Equation 4.6 produces the local normalized pressure gradient value at each extracted point in the computational domain. The local pressure gradient is normalized by the geometric mean of the pressure gradient ($GmeanPressureGradient$), which is calculated across all extracted points. If the $NormalPressureGradient$ is close to unity, it has a high shock strength and will calculate a value near one. Conversely, if it is low, the extraction has a low shock strength and will calculate a lower value. Depending on the dataset, if $NormalPressureGradient$ is calculated to be above 0.9, it should be within the shock region, but below that it may or may not.

Equation 4.7 calculates the $NormalParallelVectors$ value to be near 0 when the local velocity and cell normal vectors are far from parallel, but equal to 1 when they are parallel. Here $PVV$ is the local $ParallelVectorsValue$, and $PMV$ is the $ParallelMagnitudesValue$, which is the magnitude of the product of the unit normal velocity and unit pressure gradient vectors.

If $NormalAverage_{LH}$ value in Equation 4.5 = 1, then LH is operating at its optimal conditions. In this case the $b^{AA}_1 = 1$ and $d^{AA}_1 = 0$, meaning that the AA$_E$ has complete belief in the extracted point. This makes sense because LH is operating with optimal conditions. Conversely, if Equation 4.5 = 0.1, then $b^{AA}_1 = 0.6$ and $d^{AA}_1 = 0.4$. This means that the AA$_E$ has little belief and high disbelief in the extracted point. This result again makes sense because the LH is operating well away from its optimal conditions. The belief does not equal exactly zero because the point was extracted for a reason, meaning it has to have “some” belief.
4.3.3 Ma-Rosendale-Vermeer Strengths, Weaknesses, and Feature Characteristics

The strengths and weaknesses of the MRV shock extraction algorithm are shown in Table 4.2. The MRV method also works well when the extracted points possess high shock strength. A high shock strength will be used as an algorithm strength. The MRV method also works well when the extracted points have cell normal and local velocity vectors that are nearly parallel. The normal parallel vectors value is used because it is independent of both the LH and MRV extraction algorithms. An additional strength/weakness parameter is the value of the local normal Mach number based on density. This is a strength when the value is near unity (Equation 2.1).

<table>
<thead>
<tr>
<th>$AA_E$ Defined by</th>
<th>$b$</th>
<th>$d$</th>
<th>$u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$AA_E$ Strengths</td>
<td>high shock strength, high normal parallel vector, high normal Mach number</td>
<td>low shock strength, low normal parallel vector, low normal Mach number</td>
<td>downstream Mach number comparison</td>
</tr>
</tbody>
</table>

In the MRV algorithm, the weaknesses are opposite the strengths. A shock wave with a low strength in MRV is when the density gradient is low, while a high strength means the extraction possesses a large local density gradient. Similar to the LH algorithm, the MRV algorithm is strong when the local velocity and cell normal vectors are nearly parallel. This becomes a weakness when the two vectors are far from parallel. The normal Mach number is calculated similar to the extraction process used by the LH method (Equation 2.1), but uses density instead of pressure. It measures how close the normal Mach number is to unity as it crosses the shock in the direction of the density gradient. A high value is a strength while a low value is a weakness.

Similar to the LH algorithm, the MRV algorithm uses Equation 4.1 [22] to calculate the theoretical downstream Mach number based on the upstream Mach number. This is again compared to the CFD simulated value. When the two values are similar, the outcome on the final opinion is positively affected, meaning that the uncertainty will be minimal and that the certainty will be increased. However, when the values are not similar, it increases the uncertainty of the outcome, meaning that the agent will not be as certain that the extraction is real.
4.3.4 Belief Tuple Value for Ma-Rosendale-Vermeer Extracting Algorithm Agent

When MRV is the extracting algorithm, each point’s AA’s belief tuple is calculated using Equations 4.8 - 4.10.

\[ b = 0.4 \times \text{NormalAverage}_{\text{MRV}} + 0.6 \]  
\[ d = -0.4 \times \text{NormalAverage}_{\text{MRV}} + 0.6 \]  
\[ u = 0.5 \times \frac{M_{2\text{calculated}} - M_{2\text{simulated}}}{M_{2\text{calculated}}} \]  

where

\[ \text{NormalAverage}_{\text{MRV}} = \frac{\text{NormalDensityGradient} + \text{NormalParallelVectors} + \text{NormalMach}}{3} \]  

(4.11)

and

\[ \text{NormalDensityGradient} = \begin{cases}  
\text{DensityGradient} & \text{if NormalDensityGradient} < 1, \\
\text{GmeanDensityGradient} & \text{if NormalDensityGradient} \geq 1. 
\end{cases} \]  

(4.12)

\[ \text{NormalParallelVectors} = \begin{cases}  
1 - \frac{|PVV|-|PMV|}{|PVV|} & \text{if } |PVV| \geq |PMV|, \\
1 - \frac{|PMV|-|PVV|}{|PMV|} & \text{if } |PVV| < |PMV|. 
\end{cases} \]  

(4.13)

\[ \text{NormalMach} = \begin{cases}  
1 - \frac{\text{NormalMach}_{\text{MRV}} - 1}{\text{NormalMach}_{\text{MRV}}} & \text{if NormalMach}_{\text{MRV}} \geq 1, \\
1 - \frac{1 - \text{NormalMach}_{\text{MRV}}}{1} & \text{if NormalMach}_{\text{MRV}} < 1. 
\end{cases} \]  

(4.14)

Equations 4.8, 4.9, and 4.10 represent the belief, disbelief, and uncertainty of each point extracted by the MRV algorithm. Equation 4.12 works in the same manner as Equation 4.6, but the local density is used instead of pressure. Equation 4.13 is the same as that used in the LH method because this parameter is algorithm independent. Each variable within Equation 4.11 was already normalized, and then normalized again by 3 so that the resultant NormalAverage\text{MRV} value is forced to be between zero and one. When the NormalAverage\text{MRV} equals one, this acts as a strength, but when it is much lower than one, the believability of the resulting opinion decreases.
The constants provided by Mortensen [12] in Equations 4.2 - 4.4 and Equations 4.8 - 4.10 were chosen in an attempt to evenly space the resulting normalized values. According to Mortensen [34], “the desire is to spread out the shock wave opinions and avoid bunching”. For example, Figure 4.5 gives a hypothetical representation of shock wave opinions with good and poor spacing. A circle represents a shock wave opinion and the scale of the figures may represent the belief value of each opinion or the probability expectation value for the entire opinion. It is important to have well-spaced opinions so that the most probable shock waves are more easily selected since they are not clustered with erroneous extractions. For example, in Figure 4.5a, the red circle is clearly the shock wave with the highest belief or probability expectation and the blue circle is lower with the second highest value. In Figure 4.5b, this spacing behavior is not as easily distinguished. With a good extraction opinion spacing, it is much simpler to decide which shock waves are more probable. Additional details are provided by Mortensen [12].

Figure 4.5: (a) Proper extraction spacing. (b) Poor extraction spacing. The scale on either may represent belief or probability expectation.

4.3.5 Non-extracting Algorithm Agent Opinion

Each value in the AA\textsubscript{NE}’s belief tuple is set using the information described in Table 3.2. These terms are quantified in a set of linear functions shown in Equations 4.15-4.17.

\[ b = 0.8 \times NormalAverage\textsubscript{NE} + 0.2 \]  

(4.15)
\[ d = -0.8 \times \text{NormalAverage}_{NE} + 0.8 \] (4.16)

\[ u = 0.5 \times \text{NormalMinimumDistance} \] (4.17)

Where the \( \text{NormalAverage}_{NE} \) is computed using either Equation 4.5 or 4.11 depending on which algorithm is being used, and

\[
\text{NormalMinimumDistance} = \left| \frac{\text{MinimumDistance}}{\text{MinimumDistanceMax}} \right|
\] (4.18)

The \( \text{NormalMinimumDistance} \) value is the distance from one point within the \( \text{AA}_{NE} \) to its closest point in the \( \text{AA}_{E} \). The \( \text{MinimumDistanceMax} \) has been set to 0.1 (based on the length scale of the simulations presented in Chapter 6) for both algorithms because it appears to give the best opinion scale spread. It should be noted that this value should change depending on the length scale of the simulation being investigated. As the length scale of the simulation increases, the \( \text{MinimumDistanceMax} \) increases by the same scale, and as the length scale decreases, the \( \text{MinimumDistanceMax} \) decreases by the same scale.

The constants in Equations 4.15 and 4.16 were chosen in a similar manner as those in the \( \text{AA}_{E} \)'s. Note the differences between them however. Here the agent forming the opinion did not extract the points, so it begins with a lower belief that the region is correct. This is shown by a belief of 0.2 in Equation 4.15, while it was 0.6 in Equation 4.2.

### 4.3.6 Master Agent Opinion

The elements of the MA belief tuple were defined in Table 3.3 and are quantified in Equations 4.19 - 4.21. The disbelief and uncertainty values are represented by linear functions, while the belief is set by a planar function. The planar function was required because the feature displacement and change in feature displacement values were both needed to accurately define the belief. The constants within these equations were previously set by Mortensen [12]. Feature displacement was previously defined in Section 3.3.5 as the measure of movement between iterations, and change in feature displacement as the rate of change of movement between iterations.

\[
b = \frac{-2.25 \times \Delta FD - 0.02 \times FD}{2} + 1
\] (4.19)
It can be seen from Equation 4.21 that if the \( FD = 0 \), meaning there is no movement between iterations, then \( u = 0 \). However if \( FD \) is large, the uncertainty will be large also. This same pattern holds true for the disbelief. When both the \( FD = \Delta FD = 0 \), Equation 4.19 calculates \( b = 1 \), meaning that the MA has complete belief in the extracted point. Even though the MA has total belief in an extraction, the individual AA agents may not. For example, a converged erroneous shock wave extraction can occur if it is extracted with a very high local pressure gradient, but has low parallel vectors and normal Mach number values. This feature would be fully converged and no longer spatially move within the solution domain, meaning that the MA has a high belief. However, since the parallel vectors and normal Mach number values are low, the AA’s would have low belief.

If MA values are low, a correspondingly low belief value will be assigned. This often occurs early in the convergence process since feature displacement and change in feature displacement are related to feature convergence. As the simulation reaches convergence, these values should become very close to zero.

### 4.4 Aggregating Final Shock Wave Feature Sets

The aggregation of the final datasets will be completed in later stages of the CAFÉ program. The following method is a suggestion to perform this aggregation. When aggregating a final feature set, there are two different feature sets to choose from (one from each algorithm). Each point in each feature set has an associated belief, disbelief, and uncertainty. These values can be used to calculate the probability expectation of each point using Equation 2.19.

Aggregating the extracted points involves searching to find if both extraction algorithms have extracted the same points. It is common for the separate algorithms to extract the same feature, but to extract the points at slightly different locations. Therefore a tolerance must be set to determine if a similar point was extracted by both algorithms. Since both algorithms tend to extract very similar points in the highly probable shock regions, this tolerance value can be set quite low. Once similar
points have been found, the point with highest probability expectation will be selected for the final feature set.

After the similar points have been selected, each individual feature set is searched manually for high values of probability expectation. If it is determined that that a high enough probability expectation region was left out of the final feature set, it can be added at this time. This brings up an interesting question as to what is a high enough probability expectation. A good rule of thumb is to consider those features with a probability expectation above 0.85 to be real.

This feature set aggregation process is to be added to the CAFÉ concept during its later stages and is not automated in this research. However, the process can be mimicked manually by using visual inspection.
CHAPTER 5. SEPARATION AND ATTACHMENT LINE FEATURE EXTRACTION METHOD

In Chapter 4, the method for feature extraction and opinion generation of shock waves was discussed. This chapter will focus on how to use the method specifically for the extraction of separation and attachment lines.

5.1 Separation and Attachment Line Characteristics

Separation and attachment line characteristics can provide useful inputs to the agents that can help them make their decisions about the expected probability of each feature. Even though there are many ways to characterize a separation or attachment line, the specific separation and attachment characteristics used in this research are pressure difference, velocity difference, and curvature. The uncertainty measures are the local wall shear stress and local velocity magnitude.

5.1.1 Pressure Difference

The pressure difference across a separation or attachment line is somewhat similar to shock waves in that there is a pressure increase across a separation or attachment line. This pressure increase is evidenced in both separation and attachment lines because of the development of a new boundary layer both approaching and leaving the line.

Both the Parallel Vectors and Phase Plane algorithms mention that they tend to extract erroneous lines when there is a weak separation or attachment line. The higher the pressure difference, the stronger the separation or attachment line, making the extractions more likely to be real. This definition can allow for a simple yet effective measure of the separation or attachment strength that can be used later in defining strengths and weaknesses of each algorithm.
5.1.2 Velocity Difference

Similar to the pressure difference, the velocity difference across a separation or attachment line is measured because as this difference increases, the separation or attachment strength increases. Another key characteristic to a separation line is that the local velocity vectors across the line point towards each other, while an attachment line consists of vectors that point away from each other.

Since a large velocity difference across a separation or attachment line equates to a strong feature, this value can be used as an additional parameter to measure feature strength. The different velocity directions can also be used as a test if the extracted feature matches physical properties. These definitions allow for another simple yet effective measure of the shock strength.

5.1.3 Curvature

Curvature is defined as the reciprocal of the radius of a circle, as shown in Equation 5.1 where \( r \) is the radius of the circle. Since separation and attachment lines are straight line segments connected end to end, a curvature definition is needed that can be applied to the combination of these straight segments. The Parallel Vectors and Phase Plane algorithms do not perform well when extracting curved separation or attachment lines. This is because the algorithms assume the flow to be locally linear. With this assumption in mind, low curvature lines may be acceptable, however highly curved lines may tend to diverge away from the physical feature in an attempt to follow a linear path. The local curvature of a separation or attachment line is generally calculated by fitting a circle to a point and the surrounding two points. The radius is then found from this circle.

\[
\text{curvature} = \frac{1}{r} \tag{5.1}
\]

5.1.4 Local Velocity

Separation and/or attachment can only occur in the presence of a local zero velocity. At a separation point, the upstream flow vectors will be in the direction of the local flow, while those downstream will be in the opposite direction because they are within the associated separation
bubble. The velocity vectors will meet, creating a localized stagnation point, or a point of zero velocity. At an attachment point, the velocity vectors point away from each other causing a small localized stagnation point, where the upstream and downstream flow is moving away from this point. This occurs because the upstream flow is controlled by the separation bubble, and the downstream flow is now free to continue in the overall flow direction.

5.1.5 Wall Shear Stress

Wall shear stress ($\tau_w$) is a measure of the shear stress on the wall of an object within a fluid flow (see Equation 2.7). This term is dependent on the local velocity of the flow and the dynamic viscosity ($\mu$) of the fluid. Wall shear stress should be equal to zero at a separation or attachment point. “Consider the velocity $u$ in the boundary layer at an infinitesimal distance $dy$ above a plate. This will be:

$$u_{y=dy} = u_0 + \frac{\partial u}{\partial y} y=0$$

(5.2)

where $u_0 = 0$ is the velocity at the surface of the plate. It is clear that $u_{y=dy}$ will be zero (i.e., separation will occur) only when $\frac{\partial u}{\partial y} y=0 = 0 [16]$. This definition can be used as a test delimiter if $\frac{\partial u}{\partial y} y=0 = 0$, and in Equation 2.7 $\tau_w$ will also be equal to zero.

5.2 Extracting and Filtering Separation and Attachment Lines

Two different separation and attachment extraction algorithms are used in this research - the Parallel Vectors algorithm (PV) and the Phase Plane algorithm (PP). Both of these algorithms were briefly described in Sections 2.2.2 and 2.2.3 respectively. Similar to shock wave extraction, two separate feature sets are generated from the algorithms. The feature sets consist of points that align to form a line. Each extracted point is not connected to the others but can later be combined into a 2-dimensional line using a 2D-Delaunay interpolator. Figure 5.1 displays the extracted separation lines from the same blunt fin simulation used in Section 3.2 where the separation lines are associated with the horseshoe vortex. This feature provides an example of an extracted feature set that contains many clearly erroneous separation line detections and displays the post-filter dataset. Some of the erroneous points exist near the boundaries of the flow domain in regions of relatively
low flow disturbances. Real separation lines are not likely to develop here. Three different filters will be discussed to eliminate these erroneous points: pressure difference, curvature, and local velocity magnitude.

Figure 5.1: (a) Represents a PV separation line extracted feature set without filtering. (b) Represents the same feature set after filtering.

5.2.1 Normal Pressure Difference Filter

The normal pressure difference $\Delta P$ filter removes all extractions that contain a small pressure rise across each line. According to Westphal [23], a separation or attachment line occurs in the presence of a local pressure rise; the larger the pressure rise, the stronger the separation or attachment line. The pressure difference value is normalized using the geometric mean of the extracted pressure difference values. This filter eliminates the lower values because separation and attachment lines are not likely to exist within these lowest regions. The pressure difference filter will be used for both the PV and PP algorithms.

Similar to shock waves, selecting a suitable minimal filtering threshold can be a tricky task. If the filter is set too low, many false extractions remain. But if it is set too high, part of the actual separation or attachment lines could be eliminated. If a value for pressure difference were chosen by taking a certain percentage of the maximum differences, it may work for some situations
but not others. The geometric mean is again used to find the value that is best suited for that particular feature set, rather than just using the maximum values. Once this value is determined, all extractions that fall below this value are removed from the feature set. In this research this filter is applied inside each respective algorithm.

5.2.2 Normal Velocity Magnitude Filter

By definition [23], the local velocity in the region of a separation or attachment line is very near zero. This occurs because, as discussed in Section 2.2, the velocity vector directions are opposite, causing either a small stagnation point (separation) or evacuated region (attachment). The velocity magnitude is calculated at each extracted point and is then normalized by the geometric mean of these values. The normal velocity magnitude filter is used to eliminate those extractions whose local velocity magnitude is high. This filter also applies to both algorithms and is performed inside the extraction algorithm.

5.2.3 Normal Velocity Difference Filter

The velocity difference $\Delta \vec{V}$ filter removes all extractions that contain a large absolute value of velocity difference across each line. The highest values are eliminated because by definition, the velocity near a separation or attachment line should be near zero. If the difference is large it means that one point surrounding the extraction is much larger than the other. This violates the definition that these velocities be very small. This value is normalized using the geometric mean of the extracted velocity difference values. An additional key to this parameter is that along a separation line, the velocity vectors will be pointing toward each other, while along an attachment line, the velocity vectors will be pointing away from each other. In order to account for this, a “sign check” was implemented to see if this “sign change” condition exists. All points are eliminated that do not have a sign change across the line. This filter is applied within each algorithm before the subjective logic process is applied.
5.3 Forming Opinions on Extracted Separation and Attachment Lines

After all the feature sets have been extracted and filtered, four opinions need to be formed on each extracted point: \( \omega_{MAA_1}^{MA}, \omega_{MAA_2}^{MA}, \omega_{R}^{AA_1}, \omega_{R}^{AA_2} \). This process is similar to that used for shock waves, where \( R \) refers to the extracted point, and \( AA_1 \) and \( AA_2 \) refer to the PV and PP extraction algorithms. Similar to shock wave extraction, only three of these four opinions are created because \( \omega_{MAA_1}^{MA}, \omega_{MAA_2}^{MA} \) are the same.

The agent opinions on each extraction for separation and attachment lines are set by quantifying Table 3.1. Both Kenwright [28] and Kenwright-Henze-Levit [27] give examples in their research of instances in which each algorithm works well and when they don’t. Similar to shock wave extraction, these situations will be used to define the strengths and weaknesses of each algorithm for this research.

5.3.1 Parallel Vectors Strengths, Weaknesses, and Feature Characteristics

The strengths and weaknesses of the PV separation and attachment line extraction algorithm are shown in Table 5.1. The PV method works well when the extracted points reside in an area of high separation or attachment strength. Thus, a high strength will be used as an algorithm strength. Separation and attachment strength is commonly referred to as the measure of pressure difference across the separation or attachment line. The PV method also works well when the extracted points display a low velocity magnitude. Additionally, the PV algorithm can be trusted when the extracted line is straight, but this belief drops when the line is curved. These values are independent of the PV and PP extraction algorithms and are used as a strength or weakness for both algorithms.

<table>
<thead>
<tr>
<th>AA(_E)</th>
<th>Defined by</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>AA(_E) Strengths</td>
</tr>
<tr>
<td>( d )</td>
<td>AA(_E) Weaknesses</td>
</tr>
<tr>
<td>( u )</td>
<td>Feature Characteristics</td>
</tr>
<tr>
<td></td>
<td>high ( \Delta P ), low ( \Delta \vec{V} ), low curvature</td>
</tr>
<tr>
<td></td>
<td>low ( \Delta P ), high ( \Delta \vec{V} ), high curvature</td>
</tr>
<tr>
<td></td>
<td>velocity magnitude, wall shear stress</td>
</tr>
</tbody>
</table>
The weaknesses in this case are the opposite of strengths. For example, if an extracted point possesses a weak pressure difference it will have a low separation or attachment strength and thus lower belief. If the velocity difference is large, it will cause the belief of the resulting opinion to drop. Additionally, when the local curvature is large, the belief decreases.

Since both algorithms do not perform well when the extracted separation lines are curved, it becomes imperative to develop a dependable method to calculate the curvature of a set of disconnected points. A method was selected that fits a circle to 3 consecutive points (see Figure 5.2 where a, b, and c are the distances between the points $P_0$, $P_1$, and $P_2$, and O is the center of the circle).

![Figure 5.2: Representation of a circumscribed circle fit to three disconnected points $P_0$, $P_1$, and $P_2$.](image)

The radius and curvature of the circumscribed circle was calculated using Equations 5.3 and 5.4 [35]. It was also found that in order to get more reliable, less “scattered” or discontinuous results, an average curvature from the 5 surrounding points of each point was calculated. This approach will tend to generate incorrect results at the starting and ending points of each separation or attachment line. Since these incorrect results only exist at the beginning and end of the lines, these results are simply accepted and are still used in the probability expectation calculations as normal.
\[ radius = \frac{abc}{\sqrt{(a+b+c)(-a+b+c)(a-b+c)(a+b-c)}} \]  

(5.3)

\[ curvature = \frac{1}{radius} \]  

(5.4)

Even though there are many different physical feature characteristics that could be used to define uncertainty, only two are used in this research - the local velocity magnitude and local wall shear stress \( \tau_w \). The local velocity magnitude was defined in Section 5.2.2. When this value is near zero, a low uncertainty is assigned to that point. However if the velocity magnitude is high, it will be assigned a higher uncertainty. By definition of a separation or attachment line, the wall shear stress \( \tau_w \) should be \( \approx 0 \). When the extractions have a very low wall shear stress values, a low uncertainty value will be calculated. Conversely, if the extractions have higher wall shear stress values, then a higher uncertainty value will be calculated (see Equations 5.7 and 5.9).

### 5.3.2 Belief Tuple Value for Parallel Vectors Extracting Algorithm Agent

When PV is the extracting algorithm, each points’ \( AA_{E} \) belief tuple is calculated using Equations 5.5 - 5.7. The constants in these equations were originally determined by Mortensen [12] and have not been altered. Here \( \Delta x, \Delta y, \) and \( \Delta z \) represent the changes in \( x, y, \) and \( z \) velocity directions across a separation or attachment line respectively.

\[ b = 0.4 \times NormalAverage_{PV} + 0.6 \]  

(5.5)

\[ d = -0.4 \times NormalAverage_{PV} + 0.4 \]  

(5.6)

\[ u = UncertaintyNormalAverage_{PV} \]  

(5.7)

where

\[ NormalAverage_{PV} = \frac{Normal\Delta P + Normal\Delta \vec{V} + NormalCurvature}{3} \]  

(5.8)

\[ UncertaintyNormalAverage_{PV} = \frac{NormalVelocityMagnitude + Normal\tau_w}{2} \]  

(5.9)

and

\[ \Delta P = ||P_1|| - ||P_2|| \]  

(5.10)
\[
\Delta \vec{V} = \begin{cases} 
\sqrt{\Delta x^2 + \Delta y^2} & \text{if } \Delta x > \Delta z, \Delta y > \Delta z, \\
\sqrt{\Delta x^2 + \Delta z^2} & \text{if } \Delta x > \Delta y, \Delta z > \Delta y, \\
\sqrt{\Delta y^2 + \Delta z^2} & \text{if } \Delta y > \Delta x, \Delta z > \Delta x,
\end{cases} 
\]  
(5.11)

Velocity Magnitude = \sqrt{V_x^2 + V_y^2 + V_z^2} 
(5.12)

Equation 5.10 produces the local pressure difference value across each extracted point in the computational domain. The pressure difference value is normalized by the geometric mean of the pressure difference \((GmeanPressureDifference)\). This is calculated across all extracted points. If the \textit{NormalPressureDifference} is close to one, it has a high separation or attachment strength and will be given a value near one. Conversely, if it is low, the extraction has a low separation or attachment strength and will be given a lower value.

Equation 5.11 calculates the \(\Delta \vec{V}\) value, and is normalized using its geometric mean and then calculates the \textit{NormalVelocityDifference} value to be near one when the local velocity difference is very small, but is set lower when the velocity difference is higher.

The \textit{VelocityMagnitude} in Equation 5.12 is calculated using the definition of the magnitude of a vector and is again normalized using its geometric mean. It becomes near zero when the local velocity magnitude is near 0, and will be calculated higher when the velocity magnitude is not near 0.

The \textit{Curvature} is calculated using the method defined in Section 5.3.1. Once the curvature for each point is calculated, it is normalized using the geometric mean of the curvature values. Due to the normalization, the \textit{NormalCurvature} value becomes approximately equal to one when the line segments are very straight, but will be lower depending on how curved the line segments are.

The wall shear stress \(\tau_w\) is calculated within the CFD package or separately, depending on whether the CFD package has the ability to export the wall shear stress. This value is also normalized using its own geometric mean. Since a low \textit{Normal}\(\tau_w\) means that the uncertainty is low, the \textit{Normal}\(\tau_w\) is subtracted from 1. This means that the resulting \textit{Normal}\(\tau_w\) is normalized to be near zero when the local wall shear stress is near zero, yet will be larger when the wall shear stress is not near zero.
Equations 5.5, 5.6, and 5.7 represent the belief, disbelief, and uncertainty of each point extracted by the PV algorithm. These equations are not set in stone and other variations may be used if they are determined to give better results. If the $\text{NormalAverage}_{PV}$ value in Equation 5.8 = 1, meaning all of the PV strength characteristics are present, then $b = 1$ and $d = 0$. This represents total belief, and also that PV is operating at its optimal conditions. If $\text{NormalAverage}_{PV}$ is equal to 0, meaning none of the PV strengths are present, then $b = 0.6$ and $d = 0.4$. This occurs because the PV is operating well away from its optimal conditions. Belief is not zero because PV still extracted that point for a reason. If $\text{NormalAverage}_{PV}$ is equal to 0.5 meaning some strengths are present but not others, then $b = 0.8$ and $d = 0.2$. This suggests a fairly high belief in the extraction.

Similar to shock wave extractions, most of the time $b + d + u \neq 1$. A simple example is that almost always $u > 0$. So even if $b = 1$ and $d = 0$, then $b + d + u$ would be greater than 1. Equation 5.7 will be greater than zero at every point except where the normalized velocity magnitude and Normalized Wall Shear Stress are exactly equal to zero. This will be rare, however, since CFD is an approximation of flow properties. When $b + d + u \neq 1$, the belief is held constant while disbelief and uncertainty are decreased until $b + d + u = 1$.

### 5.3.3 Phase Plane Strengths, Weaknesses, and Feature Characteristics

The strengths and weaknesses of the PP separation and attachment line extraction algorithm are the same as those in the PV algorithm and are shown below in Table 5.2. The PP method works well when the extracted points reside in an area of high separation or attachment strength (high pressure difference). The PP method also works well when the extracted points display a low velocity magnitude and curvature, but not so well when the velocity magnitude and curvature are high.

<table>
<thead>
<tr>
<th>AA$_E$</th>
<th>Defined by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$ AA$_E$ Strengths</td>
<td>high $\Delta P$, low $\Delta \vec{V}$, low curvature</td>
</tr>
<tr>
<td>$d$ AA$_E$ Weaknesses</td>
<td>low $\Delta P$, high $\Delta \vec{V}$, high curvature</td>
</tr>
<tr>
<td>$u$ Feature Characteristics</td>
<td>velocity magnitude, wall shear stress</td>
</tr>
</tbody>
</table>
The weaknesses in this case are also the opposite of strengths. For example, if an extracted point possesses a low pressure difference, it will have a low separation or attachment strength, decreasing the believability of the opinion of that extraction. If the velocity difference is large, and the local curvature is large, that will also decrease the believability of the resulting opinion.

The radius and curvature for the PP algorithm are calculated in the same manner as those used by the PV algorithm.

The local velocity magnitude (Section 5.2.2) and local wall shear stress $\tau_w$ (Section 5.1.5) used in the PV algorithm are also used in the PP algorithm. When these values are near zero, a low uncertainty is calculated to that point using Equations 5.15 and 5.17. But if the velocity magnitude is high, a higher uncertainty value is calculated. The wall shear stress $\tau_w$ should be $\approx 0$. When the extractions have a very low wall shear stress, then a low uncertainty value is calculated. Conversely, if the extractions have higher wall shear stress, a higher uncertainty value is calculated.

### 5.3.4 Belief Tuple Value for Phase Plane Extracting Algorithm Agent

When PP is the extracting algorithm, each point’s $A_{AE}$ belief tuple is calculated using Equations 5.13 - 5.15. These equations are the same as those used for the PV algorithm.

\[
b = 0.4 \times \text{NormalAverage}_{PP} + 0.6 \tag{5.13}
\]

\[
d = -0.4 \times \text{NormalAverage}_{PP} + 0.4 \tag{5.14}
\]

\[
u = \text{UncertaintyNormalAverage}_{PP} \tag{5.15}
\]

where

\[
\text{NormalAverage}_{PP} = \frac{\text{Normal}\Delta P + \text{Normal}\Delta \vec{V} + \text{NormalCurvature}}{3} \tag{5.16}
\]

\[
\text{UncertaintyNormalAverage}_{PP} = \frac{\text{NormalVelocityMagnitude} + \text{Normal}\tau}{2} \tag{5.17}
\]

Where $\Delta P$, $\Delta \vec{V}$, and $\text{VelocityMagnitude}$ are defined the same as those used in the PV algorithms Equations 5.10 - 5.12.

Equations 5.13, 5.14, and 5.15 represent the belief, disbelief, and uncertainty of each point extracted by the PP algorithm. The $\text{Normal}\Delta P$, $\text{Normal}\Delta \vec{V}$, $\text{NormalCurvature}$, $\text{Normal}\tau$, and
NormalVelocityMagnitude, that are used in Equations 5.13, 5.14, and 5.15 are all defined the same as those in the PV algorithm in Section 5.3.2.

The constants provided by Mortensen [12] in Equations 5.5 - 5.7 and Equations 5.13 - 5.15 were chosen in order to evenly space the resulting normalized values. Similar to shock wave extraction, the desire is to spread out separation and attachment line opinions and avoid bunching [34]. Consider again the hypothetical representation of separation and attachment line opinions with good and poor spacing in Figure 5.3. In Figure 5.3a the red circle is clearly the separation or attachment line with the highest belief or probability expectation and the blue circle is lower than it with the second highest value. This spacing behavior in Figure 5.3b is not as easily determined. Again, similar to shock wave extraction, a good spacing is much simpler to decide which separation or attachment lines are more probable.

![Figure 5.3: (a) Proper extraction spacing. (b) Poor extraction spacing. The scale on either may represent belief or probability expectation.](image)

5.3.5 Non-extracting Algorithm Agent Opinion

Each value in the AA_{NE}’s belief tuple is set using the information described in Table 3.2. These terms are quantified in a set of linear functions shown as Equations 5.18-5.20 and are similar to those used by the PV algorithm:

\[
b = 0.8 \times \text{NormalAverage}_{NE} + 0.2 \quad (5.18)
\]
\[ d = -0.8 \ast \text{NormalAverage}_{NE} + 0.8 \]  
\[ u = 0.5 \ast \text{NormalMinimumDistance} \]  

Where the \text{NormalAverage}_{NE} is computed using either Equation 5.8 or 5.16 depending on which algorithm is being used, and

\[ \text{NormalMinimumDistance} = \left| \frac{\text{MinimumDistance}}{\text{MinimumDistanceMax}} \right| \]  

The \text{NormalMinimumDistance} value is the distance from one point within the \text{AA}_{NE} to its closest point in the \text{AA}_{E}. The \text{MinimumDistanceMax} has been set to 0.1 (based on the length scale of the simulations presented in Chapter 6) for both algorithms because through trial and error it appears to give the best opinion scale spread. It should be noted that this value should change depending on the length scale of the simulation being investigated. As the length scale of the simulation increases, the \text{MinimumDistanceMax} increases by the same scale. As the length scale decreases, the \text{MinimumDistanceMax} decreases by the same scale.

The constants in Equations 5.18 and 5.19 were chosen in a similar manner as those for the \text{AA}_{E}’s. However, in this case the agent forming the opinion did not extract the points, so it begins with a lower belief that the region is correct. This is shown by the 0.2 in Equation 5.18, where in Equation 5.5 it was 0.6.

### 5.3.6 Master Agent Opinion

The elements of the MA belief tuple were defined in Table 3.3 and are shown in Equations 5.22 - 5.24. The disbelief and uncertainty values are represented by linear functions while the belief is set by a planar function. The planar function was required because the feature displacement and change in feature displacement values were both needed to accurately define the belief. The constants within these equations were previously set by Mortensen [12]. Recall that feature displacement was defined in Section 3.3.5 as the measure of movement between iterations, and change in feature displacement as the rate of change of movement between iterations.

\[ b = \frac{-2.25 \ast \Delta FD - 0.02 \ast FD}{2} + 1 \]  

75
\[ d = -0.02 \times FD \]  
\[ u = 2.25 \times FD \]

Similar to shock wave extraction, from Equation 5.24 it can be seen that if the \( FD = 0 \), meaning there is no movement between iterations, then \( u = 0 \). However if \( FD \) is large, the uncertainty will be large also. This same pattern holds true for the disbelief. When both the \( FD = \Delta FD = 0 \), it can be seen from Equation 5.22 that \( b = 1 \) meaning that the MA has complete belief in the extracted point. If these values are low, a correspondingly low belief value is calculated. Feature displacement and change in feature displacement are often related to feature convergence, so as the simulation reaches convergence, these values should become very close to 0.

5.4 Aggregating Final Separation and Attachment Feature Sets

The suggested method for this aggregation is the same as that discussed in Section 4.4. This feature set aggregation process is to be added to the CAFÉ concept during its later stages but is not automated in this research. However, the process can be mimicked manually by using visual inspection.
CHAPTER 6. METHOD VALIDATION

Two CFD simulations were run to test both the shock wave and the separation and attachment line extraction algorithms to verify that concurrent extraction is possible and validate the shock wave and separation and attachment algorithm methods defined in Chapters 4 and 5. The two geometries for the shock wave extraction validation are common CFD datasets — the supersonic ramp [19] and the Swept Onera M6 wing [33]. The two geometries used for separation and attachment extraction were the cylinder in a cross flow [36] and a highly swept delta wing [37]. All of these cases were selected because they are complex enough to provide many of the features of interest, but are simple enough to solve and visualize on a standard desktop computer.

6.1 Supersonic Ramp

The supersonic ramp simulation was used to replicate a common bow shock forming in front of a wedge in a flow. This geometry is the same as that used by Lovely-Haines [19] with a base width of 0.25 m, height of 0.25 m, and a length of 0.35. The Reynolds Averaged Navier-Stokes (RANS) equations were calculated using FLUENT 12. The grid was generated in Gambit as an unstructured tetrahedral mesh with 882,325 nodes (see Figure 6.1). The simulation was run to convergence, and then the mesh was refined in regions of high pressure gradient to better capture the shock waves. The inlet Mach number was set as $M=3$ at an angle of attack of $0^\circ$. Following the Lovely and Haines method, the viscous model was set to laminar for simplicity in this simulation. Running the model as inviscid will neglect wall effects and will not model the boundary layer. Since this simulation is not accounting for the boundary layer, we can disregard its effect by not modeling it. Air was the working fluid, and the simulation utilized the pressure-based, Least Square Cell Based Gradient solution method. The FLUENT FMG (Full Multigrid Initialization) technique was employed in order to generate a good initial condition. The FLUENT FMG process is completed by constructing the desired number of geometric grid levels. The inviscid Euler
equations are then solved using first order discretization in order to obtain the approximate solution at the coarsest level. This approximation is then interpolated onto the more fine grids. This process is repeated until the finest level is reached with a suitable initial condition. The simulation was then run to full convergence (1000 iterations), and the residuals are displayed in Figure 6.2. The drop in residual values at about 250 iterations represents the switch to 2\textsuperscript{nd} order approximation, and a decrease in over-relaxation factors. While this simulation is not large enough to necessitate concurrent feature extraction, it provides a dataset that can validate the shock wave extraction method.

Figure 6.1: Representation of the mesh generated for the Supersonic Bump dataset. Flow is moving from left to right.

In order to replicate concurrent extraction, the entire flow field was exported to the hard drive from FLUENT every 50\textsuperscript{th} iteration. The shock waves were then extracted from each dataset using the Lovely-Haimes and Ma-Rosendale-Vermeer methods defined in Chapter 4 — each creating its own unique extraction dataset. The extracted datasets were then run through the subjective logic architecture to calculate a belief tuple Probability Expectation value for each extraction.

The fully converged supersonic ramp dataset is represented in Figure 6.3. This 2D slice of the 3D dataset is colored by Mach number and labels the bow and trailing shocks. As the
supersonic freestream flow contacts the forward portion of the ramp, it begins to slow down. When the velocity of the air moving past the ramp reaches a point when it can no longer transmit upstream information about the ramp, a shock wave occurs. The shock wave occurs because the downstream conditions still needs to satisfied. The flow then “shocks” in order to satisfy the higher pressure downstream conditions. This is the case in the bow shock that occurs in front of the bump and is attached to the nose of the ramp. The bow shock is characterized by a large abrupt drop in Mach number. The upstream Mach number is approximately 3, and drops to approximately 1.75. Since the shock is oblique, the trailing Mach number is allowed to be greater than one. But the downstream Mach number normal to the shock wave will be less than one. After the bow shock, the flow begins to speed up again and a much weaker shock occurs where the back side of the bump meets the floor. This region of compression creates the trailing shock wave. These results resemble those developed by Lovely and Haimes [19] because the spatial locations closely match the shock locations in the simulation developed by Lovely and Haimes.
6.1.1 Shock Waves in Converging Datasets

Figures 6.4(a)-6.4(d) are side views of the supersonic ramp showing the shock wave extraction results from the MRV algorithm. Many of the extraneous extractions have already been eliminated to make the images more readable (as demonstrated in Section 3.2). The black dots that form the two separate waves represent the extractions at full convergence while the red dots that form another two waves represent the extractions at 10%, 25%, 50%, and 85% convergence. Each percentage was obtained by dividing the current iteration number by the number of iterations at full convergence (1000) and multiplying by one hundred. In this case the flow is moving from left to right. For comparison, the intermediate and fully converged datasets are plotted together.

The computational results of the supersonic ramp dataset are similar to the bow shock generated by Lovely and Haimes [19], however the trailing shock was filtered out by Lovely and Haimes by their pressure gradient filtering. Since an actual shock occurs here, this demonstrates the difficulty of just using simple filtering techniques to determine real shock waves.

It can be seen that as the iterations continue, the waves in Figure 6.4 begin to converge upon each other. With this side view, this can most easily be seen by looking at how spatially similar the red and black trailing shock waves are to each other. As the iterations continue, the closer
Figure 6.4: Comparison of MRV extracted shock waves from converged supersonic ramp dataset (black) and converging datasets (red). (a) At 10% converged, both shock waves have already formed. (b) At 25% converged, both shocks have formed but are misplaced. (c) At 50% converged, the trailing shock is still slightly displaced. (d) At 85%, both shock waves spatially match the converged dataset. In this case, flow is moving from right to left.

they become. This is evident in Figure 6.4(d) where the two datasets are nearly identical. At 10% convergence, both of the shock waves have appeared (in red), but the trailing shock appears to reside approximately 0.05 m above the converged dataset (in black). The same phenomenon can be seen at the upper edge of the bow shock, and occurs because the pressure gradients have not yet properly developed spatially. At 30% convergence, some of the bow shock wave appears to mostly match the converged dataset, while the trailing shock wave has moved about 0.015 closer to the converged dataset. At 50% convergence, the bow shock wave appears to be approximately 0.005 m aft of the fully converged dataset, and the trailing shock still appears approximately 0.01 m above
the converged solution. At 85% convergence, both shock waves appear to be lying directly on top of each other. This is suggesting that at 85% convergence, the shock waves have settled into their final spatial locations within 0.01%.

### 6.1.2 Shock Waves in Converging Datasets Processed by Agents

Figures 6.5(a)- 6.5(d) show a comparison of the shock wave extraction results from the MRV algorithm colored by probability expectation at 15%, 25%, 50%, and 85% convergence versus the fully converged dataset (represented by black dots). These figures are now shown parametrically to give a better representation of the probability expectation. Each dataset is displayed as a series of points forming an overall shock wave as discussed in Section 4.2. At first glance it appears that each of these frames are the same. This is because the shock waves developed very early in the convergence process. The differences between each will be discussed here. Figure 6.5(a) shows that the Probability Expectation is only high ($\approx 1.0$) within the bow shock region. This makes sense because this region is the closest spatially to the converged dataset and its large density gradient. The rest of the extractions are lower, suggesting that the other extractions farther away from the walls are probably not accurate shocks. A similar analysis could be made of the extraction in Figure 6.5(b) at 25% convergence. The extractions far away from the wall now have higher probability expectation values ($\approx 0.8$), meaning that they are probably part of the bow shock, but are very weak portions of it. The trailing shock now also shows higher probability expectation values $\approx 0.8 - 0.9$, meaning that the extractions are real, but are part of a much weaker shock than the bow shock.

The 50% and 85% convergence feature sets in Figures 6.5(c) and 6.5(d) are very similar in showing that the probability expectation of the bow shock is very high ($\approx 0.95 - 1.0$) and decrease as the wave extends away from the wall. This decrease occurs because the density gradient is decreasing as it extends away. The trailing shock has lower probability expectation values ($\approx 0.80 - 0.85$). The reason these are so low within a region that one would expect a shock to be is because the density gradient is approximately 80% smaller than those in the bow shock, meaning it is a much weaker shock. The results from the LH algorithm are similar and show the same trends.
Figure 6.5: Comparison of shock waves extracted by MRV at 15% (a), 30% (b), 50% (c), and 85% (d) convergence increments. The black dots represent the final converged dataset. Flow is moving from the bottom right to the top left.

### 6.1.3 Comparison of Shock Waves Processed by Agents from Converged Solutions

A comparison of the shock waves extracted at full solution convergence by the MRV and LH algorithms after the agents have formed their final opinions will be detailed here. Both algorithms present high belief values ($\approx 0.90 - 1.0$) within the bow shock region, but lower values ($\approx 0.60 - 0.70$) in the trailing shock region (Figures 6.6(a) and 6.6(b)). High values in these regions were expected because they contain characteristics that are strengths of both extraction algorithms (high pressure, density gradients, etc.) that were discussed in Sections 4.3.1 and 4.3.3. The low values
that appear in front of the high values (≈ 0.90 – 1.0) in Figure 6.6(b) are associated with the LH shock shell as described in Section 2.1.2 and are thus lower in belief value. Because they have low belief values, these are not considered part of the actual bow shock.

The regions that do not contain, or only partially contain, the strengths of each algorithm display lower belief values (≈ 0.50 – 0.60). These regions exist within the trailing shock, upper portions of the bow shock, and forward portions of the LH shock shell. Disbelief values of ≈ 0.00 – 0.15 exist in the bow shock region, but are higher (≈ 0.20 – 0.35) in the trailing shock wave and forward portion of the LH shock shell (Figures 6.6(c) and 6.6(d)). Physically, a low disbelief value for a shock wave means that either the shock is not very strong, the parallel vectors value is low (≳ 0.5), the normal Mach vector is not normal to the shock, or a combination of these.

The uncertainty in Figures 6.6(e) and 6.6(f) display low values of uncertainty (≈ 0.0) in the most of the extracted regions. Low uncertainty values inform the researcher that those features contain numerical and theoretical $M_2$ values matching within 8% (Sections 4.3.1 and 4.3.3), meaning that the agents are fairly certain that the calculated belief and disbelief values are accurate representations of a real shock wave. This also means that the extractions have settled on their final spatial location, making $FD$ and $\Delta FD$ almost zero. Since almost all of the extractions display very low uncertainty values, it can be assumed that each extraction is part of a physical shock wave. These values however are higher in the forward portions of the LH shock shell. This makes sense because they are not actually a part of the shock wave. The value is still low however, because the $FD$ and $\Delta FD$ are almost zero.

Figures 6.6(g) and 6.6(h) show that both algorithms display high probability expectation values (≈ 0.90 – 1.0) in the bow shock regions. This occurs because according to Equation 2.19, if the belief is high and the uncertainty is low, then a high probability expectation will result. This is the case in the bow shock region, but not in the other extractions. Since the belief is low and disbelief is high, the extractions comprising the forward portions of the LH shock shell and trailing shock wave will have lower probability expectation values. In this case, it can be safely assumed that any extraction with a probability expectation greater than 0.90 is probably part of an actual shock wave because they are the most probable of all the extractions. However, the trailing shock wave values need to be carefully considered by the researcher because theoretically a very weak
shock wave does exist here. But since the density and pressure gradients are so much smaller than the bow shock, it receives much lower probability expectation values.

6.2 Swept Onera M6 Wing

A Swept Onera M6 wing [33] CFD simulation was also run for shock wave detection. The Reynolds Averaged Navier-Stokes (RANS) equations were solved using FLUENT 12. The grid was generated in Gambit as an unstructured tetrahedral mesh with 553,447 nodes (see Figure 6.7). The wing span length is 0.86 m, and the chord length sweeps from 1.0 m at its root to 0.56 m at the tip. The inlet Mach number was set as M=0.8395 at an angle of attack of 3.06°. The Spalart-Allmaras turbulence model was used for this simulation. These inlet conditions match those used by Slater [33]. Air was the working fluid, and the simulation utilized the pressure-based, Least Square Cell Based Gradient solution method. The FLUENT FMG (Full Multigrid Initialization) technique was employed in order to generate a good initial condition. The simulation was then run to full convergence (2000 iterations) and the residuals are displayed in Figure 6.8.

Using the Lovely-Haimes and Ma-Rosendale-Vermeer methods defined in Chapter 4, shock waves were then extracted from each dataset, each creating its own unique extraction dataset. Subjective logic was then used to calculate a belief tuple (belief, disbelief, uncertainty, and probability expectation) for the extraction shock waves from each dataset.

6.2.1 Definition of Extracted Shock Waves

The fully converged Swept Onera M6 wing simulation is presented in Figure 6.9. This 2D slice (at 65% wing span) of the 3D dataset is colored by Mach number and shows forward and aft shock waves. As the subsonic freestream flow moves across the upper portion of the wing, it begins to speed up due to the wing geometry. As the flow accelerates, it creates an adverse pressure gradient. In the case of this Onera M6 wing, two shock waves are formed – the forward and aft shock waves.

Figure 6.10 shows the shock waves extracted by the LH and MRV extraction algorithms. In these figures, flow is moving from left to right. The forward shock occurs at approximately 15% of the chord length and runs close to the leading edge, because of the velocity is increasing so rapidly.
Figure 6.6: Comparison of belief tuple values and probability expectation values for the shock waves in the converged dataset. Flow is moving from the bottom right to the top left.
Figure 6.7: Representation of the mesh generated for the Swept Onera M6 wing dataset.

Figure 6.8: Residual convergence history of the swept Onera M6 dataset.

along the forward portion of the wing. After the forward shock slows the flow, it begins to speed up again due to the wing geometry. As the velocity begins to increase after the forward shock, the aft shock wave results. Both the forward and aft shock waves are characterized by an abrupt drop in Mach number and form a “v” or “:\textlambda ;” shape. These results resemble the CFD Mach numbers generated by Slater [33]. Additionally, the spatial locations of the forward and aft shock waves
are very similar to the those generated by Slater. He describes the locations of the shock waves by showing a 2D slice of the pressure contours. These spatial locations closely match the Mach number jump locations in the simulation developed for this research.

These extracted shock waves are shown in both the LH and MRV feature sets (Figure 6.10(a) and Figure 6.10(b) respectively). The LH feature set looks like there are no erroneous extractions, but erroneous forward portions of the LH shock shell exist in addition to the more believable portions of the shell. This LH shell was discussed in Section 2.1.2. The MRV dataset appears to be somewhat cluttered with erroneous extractions, but the forward and aft shock waves are shown in similar spatial locations as those in the LH feature set. Even though these feature sets look vastly different, the real portions of the forward and aft shocks are spatially similar. However, these two feature sets extract different shock wave points because each extraction algorithm uses different criteria for extraction.

6.2.2 Comparison of Shock Waves Processed by Agents from Converged Solution

Figures 6.11(a) and 6.11(b) show the density and pressure gradients of the shock waves from the converged solution. Figure 6.11(b) shows only the aft faces of the shocks because the forward
portion of the LH shock shell blocks the view of the aft extractions when viewed in the same position as Figure 6.11(a). The pressure and density gradients are $\approx 0.85 - 1.0$ near the bottom portions (closest to the wing) of the forward and aft shocks. The outer portions of these shock waves begin to show lower gradient values ($\approx 0.5 - 0.8$) as they extend farther away from the surface of the wing. Physically, this means that the forward and aft shock waves are dissipating as they extend away from the wing. The erroneous extractions (near the leading edge) presented in Figure 6.11(a), possess very high normalized density gradient values ($\approx 0.90 - 1.00$). These are high because they are associated with the stagnation flow near the leading edge. Looking only at gradient strength might mislead a researcher to believe these are shock waves, however it will be shown later in this section that these are not actually shock waves based on their other characteristics.

Figures 6.11(c) and 6.11(d) display the parallel vectors values of the extracted shock waves from the converged solution. Both feature sets display parallel vectors values $\approx 0.95 - 1.0$ within the aft shock. The forward shock wave displays somewhat lower parallel vectors values ($\approx 0.85 - 0.90$). This means that the extractions with high parallel vectors values meet the criteria described by Lovely and Haimes [19] that the Mach number normal to the shock be approximately equal to one. Figure 6.11(c) also shows that the extractions near the leading edge (flow stagnation region)
have very low parallel vectors values of $\approx 0.50 - 0.65$. This suggests that according to Lovely and Haimes [19], these extractions are not real shock extractions. Interestingly however, the extractions near the trailing edge of the wing are $\approx 0.85 - 0.95$. Again, this might mislead a researcher to think these are actual shock waves if considering only the parallel vectors criteria.

Figures 6.12(a) and 6.12(b) display the Mach uncertainty values of the extracted shock waves. Both extractions display Mach uncertainty values of $\approx 0.0 - 0.20$ within the aft shock. Low Mach uncertainty values mean that the theoretical and numerical $M_2$ values of these extractions were similar. For this simulation they matched within 8% - 12% of each other (Sections 4.3.1 and 4.3.3). These percent error values were determined by comparing the computational $M_2$ value with the theoretical $M_2$ value. This suggests that the agents can be fairly certain the extractions
are part of real shock waves. The forward shock wave displays a similar pattern, with Mach uncertainty values \( \approx 0.00 - 0.20 \). Figure 6.12(a) also shows that the extractions near the leading and trailing edges have higher Mach uncertainty values (\( \approx 0.30 - 0.50 \)). These large uncertainty values correspond to a 35% - 55% difference between the computational \( M_2 \) and theoretical \( M_2 \) values. This suggests that these extractions are not real shock extractions because they do not meet the physical definition of a shock wave.

![Figure 6.12](image)

Figure 6.12: Display of shock waves extracted by LH and MRV algorithms colored by normalized density/pressure gradient and parallel vectors at full convergence.

Figure 6.12(c) displays the normalized Mach value calculated by the MRV algorithm. Recall from Section 4.1.3 that the normalized Mach value is the dot product between the local Mach vector and the local density gradient. A shock wave exists when this value is approximately equal.
to one. This parameter was used in the LH algorithm to define a shock wave, and is not used as a strength or weakness of the LH algorithm. It can be seen that in both the forward and aft shock waves, the normalized Mach value \( \approx 0.90 - 1.00 \). This means that they meet a physical requirement to be a shock wave, meaning that the normal Mach number just before the shock needs to be approximately equal to one. However, the normalized Mach value is much lower in the extractions near the leading and trailing edges \( \approx 0.50 - 0.70 \) because it appears they do not meet this physical definition of a shock wave.

With the strengths and weaknesses of each algorithm calculated, they are combined into a belief tuple using the subjective logic architecture defined in Sections 4.3.2 and 4.3.4. The belief and disbelief components of the belief tuple are shown in Figure 6.13. Figures 6.13(a) and 6.13(b) display the belief values of each algorithm. The belief is \( \approx 0.95 - 1.00 \) within the bulk of the shock waves. In Figure 6.13(a), the belief is high in the forward and aft shocks because the normalized density gradient (Figure 6.11(a)), parallel vectors (Figure 6.11(c)), and normalized Mach number (Figure 6.12(c)) values are high \( \approx 0.90 - 1.00 \) in these regions. Since all the strengths of the MRV algorithm are high, the belief becomes associatively high. The extractions near the leading and trailing edges have lower belief values \( \approx 0.75 - 0.85 \) because not all of the strengths of the MRV algorithm are represented with high values. The normalized density gradient is high \( \approx 0.95 - 1.00 \) near the leading edge (Figure 6.11(a)), so the belief will not be very low \( \approx 0.50 \) even though the parallel vectors (Figure 6.11(c)) and normalized Mach values (Figure 6.12(c)) are low \( \approx 0.50 - 0.70 \). The same thing can be said about the trailing edge extractions. The parallel vectors values are high (Figure 6.11(c)), but the normalized density gradient and normalized Mach values aren’t, making the belief to be calculated at \( \approx 0.75 - 0.85 \).

A similar conclusion can be drawn for the LH extractions in Figure 6.13(b). The normalized pressure gradient (Figure 6.11(b)) and parallel vectors values are high \( \approx 0.90 - 1.00 \) within the forward and aft shock regions, making the belief high in these waves. Since the normalized pressure gradient (Figure 6.11(b)) is not very high in the forward shock region \( \approx 0.50 - 0.80 \), it will have correspondingly lower belief values \( \approx 0.85 - 0.90 \). This is important to note because if the researcher were only considering the LH algorithm, this forward shock might be disregarded as a questionable shock wave.
The disbelief values of the shock wave extractions are shown in Figures 6.13(c) and 6.13(d). Since the normalized density gradient, parallel vectors, and normalized Mach value are all high ($\approx 0.90 – 1.00$), the disbelief is low ($\approx 0.00 – 0.05$) in the MRV forward and aft shock extractions (Figure 6.13(c)). For the LH algorithm, the normalized pressure gradient and parallel vectors values are high ($\approx 0.90 – 1.00$), making the disbelief $\approx 0.00 – 0.10$ within the forward and aft shock extractions (Figure 6.13(d)).

The extractions near the leading edge and forward portion of the forward shock shell in Figures 6.14(a) and 6.14(b) have uncertainties higher ($\approx 0.15 – 0.25$) than the forward and aft shocks because their respective parallel vectors and normalized Mach values are low. However,
these values are not even lower because the normalized pressure gradient values near the leading edge are high ($\approx 0.90 - 1.00$). A similar pattern is shown at the leading edge in Figure 6.13(c). The normalized density gradient and normalized Mach values are low ($\approx 0.50 - 0.60$) near the trailing edges, but the parallel vectors values are high ($\approx 0.90 - 1.00$), making the disbelief $\approx 0.00 - 0.15$ but not $\approx 0.0$.

Figure 6.14 displays the uncertainty and probability expectation values of the belief tuple of the extracted shock waves. The uncertainty in Figures 6.14(a) and 6.14(b) display low values of uncertainty in the forward and aft shock regions ($\approx 0$). These low values are a result of the extractions having settled on their final spatial location, making $\Delta F$ and $\Delta F_D$ almost zero. These values are also low because the Mach uncertainty values are low ($\approx 0$) within the forward and aft shocks. Figure 6.14(a) shows that the extractions near the leading and trailing edges have higher uncertainty ($\approx 0.1 - 0.25$). These extractions have higher values than the ones in the shock waves because they are still moving and because their Mach uncertainty values are higher ($\approx 0.00 - 0.20$) than the forward and aft shock waves. Even though the solution is converged, the extraneous features can still be moving because they are interpolated between computational nodes. When the feature is not strong between these nodes, the algorithms can’t fully decide where to place the extracted point. The higher uncertainty values suggest that the agents are not as certain about these extractions as those extractions in the forward and aft shock waves.

Figures 6.14(c) and 6.14(d) show the shock waves extracted by the MRV and LH algorithms colored by probability expectation. The expected probability is $\approx 0.9 - 1.0$ for both algorithms across most of the forward and aft shock waves. This makes sense because the belief values are high ($\approx 0.90 - 1.00$), and the uncertainty is low ($\approx 0.00$) in these regions. The extractions near the leading and trailing edges of Figure 6.14(c) have lower probability expectation values than the forward and aft shock waves because their belief is lower ($\approx 0.75 - 0.85$). The lower belief in the leading edge extractions occurs because the parallel vectors values are low ($\approx 0.50 - 0.70$). Likewise, the belief is lower in the trailing edge extractions because the normalized density gradient values are low ($\approx 0.50 - 0.70$). The uncertainty is higher in the leading and trailing edges at $\approx 0.00 - 0.15$ compared to the forward and aft shock waves because their Mach uncertainty values are high at $\approx 0.35 - 0.50$. 
Figure 6.14: Comparison of uncertainty and probability expectation components of the belief tuple for the shock waves in the converged dataset. Flow is moving from left to right.

It can be safely assumed that any extraction with a probability expectation greater than 0.90 is probably part of an actual shock wave because these extractions contain all or most of the algorithm strengths as defined in Sections 4.3.1 and 4.3.3. This value of 0.90 is not a concrete number to be as a cut-off used on all datasets. It is specific to this dataset only and is subjective to the specific user, others may deem a value of 0.85 would be more fitting. These results coincide with the simulated and experimental shock waves described by Slater [33]. They also show that using only one of the above characteristics can be misleading. This method effectively shows the real shock waves while showing which extractions are probably not real shock waves.
6.3 Cylinder In A Cross Flow

A CFD cylinder in a cross flow simulation was created to generate a dataset that can validate separation and attachment extractions. The Reynolds Averaged Navier-Stokes (RANS) equations were simulated using FLUENT 12. The grid was generated as an unstructured tetrahedral mesh with 932,804 nodes (see Figure 6.15). The cylinder diameter was 0.075 m and was 0.525 m tall. The inlet velocity was set to 0.1 m/s, giving a Reynolds number of 513.56. These inputs were based on simulations performed by Fornberg [36]. The viscous model was set to “laminar” (meaning the solver will neglect the viscous terms in the governing equations) for simplicity in this simulation. The working fluid was air, and the simulation utilized the pressure-based, Least Square Cell Based Gradient solution method. The simulation was run to full convergence (950 iterations). The residuals are displayed in Figure 6.16. Like the supersonic ramp, this simulation is not large enough to require concurrent feature extraction, but it can be used to verify the separation and attachment line extraction methods.

Figure 6.15: Representation of the mesh generated for the cylinder in a cross flow dataset.

Concurrent extraction was replicated in this simulation by exporting the entire flow field from FLUENT every 50th iteration. The separation and attachment lines were then extracted from each dataset using the Parallel Vectors and Phase Plane methods defined in Chapter 5. The subjective logic architecture was then used to calculate the belief tuple for each extraction.
Figure 6.16: Residual convergence history of the cylinder in a cross flow simulation.

A close-up of the fully converged cylinder in a cross flow simulation is shown in Figure 6.17. Figure 6.17 shows the Surface LIC lines (computational simulation of surface oil streaks) that represent the separation and attachment lines. The separation and attachment lines are characterized by the LIC lines converging and diverging from each other. There are two main pairs of separation lines for the cylinder in a cross flow dataset: the cylinder and the trailing separation line pairs. The cylinder separation lines form as a result of the adverse pressure gradient present as the flow moves around the cylinder. The trailing vortices are formed as the flow moves around the cylinder and meets again behind it. The black dots in Figure 6.18(a) represent the fully converged separation lines. This figure shows that quickly after the trailing vortices are formed, they begin to break down and the lines become more disrupted. The vortex breakdown region is characterized by the vortices losing their strength and reducing to disorganized turbulent flow. When this occurs, the separation and attachment lines become broken and develop holes. This behavior is associated with vortex breakdown, meaning when vortical structures lose strength due to turbulent diffusion. This erratic behavior can be seen in Figure 6.18(a) when the black dotted separation lines begin
to become more sporadic. Experimental and computational results of this cylinder in a cross flow have shown the formation of the cylinder separation line pair [36].

Figure 6.17: Close up representation of the cylinder in a cross flow data set. Surface LIC lines represent the different separation and attachment lines.

### 6.3.1 Separation and Attachment Lines in Converging Datasets

Figures 6.18(a)- 6.18(d) represent the cylinder in a cross flow with the extracted separation lines from the PV algorithm. Extraneous extractions were previously filtered out to clean up the images using the same process demonstrated in Section 5.2. This is represented in Figure 6.19 where some of the obviously erroneous extractions were filtered out. The black lines represent the extractions at full convergence while the red lines represent the extractions at 15%, 30%, 55%, and 85% convergence. For comparison the intermediate and fully converged datasets are plotted together.

As the iterative process continues, the non-converged separation lines begin to move closer to the fully converged lines. This is shown in Figure 6.18(d) where the two feature sets are nearly identical. At 15% convergence, the cylinder separation line has formed, but appears to be approx-
Figure 6.18: Comparison of PV extracted separation lines from converged cylinder in a cross flow dataset (black) and converging datasets (red). (a) At 15% converged, the cylinder separation lines are formed but misplaced. (b) At 30% converged, the aft separation lines take more form, but all are misplaced. (c) At 55% converged, both separation lines are formed and closer to their final locations. (d) At 85%, the separation lines nearly spatially match the converged dataset.

approximately 5% farther aft than the final spatial location. The trailing separation lines are beginning to form, but are approximately 8% from their final locations. Additionally, many questionable extractions are present on the aft portions of the cylinder. These extractions are obviously false, because physically they can not exist on this surface. At 30% convergence, both the cylinder and trailing separation lines have formed, but the trailing lines are closer than before. At 55% convergence,
both lines appear to have moved within approximately 2.5% of their final location, but are still not there yet. Fewer questionable extractions are present along the back side of the cylinder at this point of the convergence process. At 85% convergence, the two feature sets appear to be lying directly on top of each other. This analysis shows that at 85% convergence, the separation lines have reached their final locations.

Figures 6.20(a)- 6.20(d) display the cylinder in a cross flow with the extracted attachment lines from the PV algorithm at multiple iterations. Similar to the separation line analysis, numerous extraneous extractions were previously filtered as discussed in Section 5.2. The black lines represent the attachment lines at full convergence while the red lines represent the attachment lines at 15%, 30%, 55%, and 85% convergence.

There are two main attachment lines for the cylinder in a cross flow dataset: the stagnation and the trailing attachment lines. The stagnation attachment line forms as a result of the flow
Figure 6.20: Comparison of PV extracted attachment lines from converged cylinder in a cross flow dataset (black) and converging datasets (red). (a) At 15% converged, the stagnation attachment line has formed. (b) At 30% converged, the aft attachment line begins take more form, but is rather messy. (c) At 50% converged, both attachment lines are formed and closer to their final locations. (d) At 85%, the real attachment lines nearly spatially match the converged dataset.

contacting the cylinder. The flow begins to separate as it begins to move around the cylinder farther downstream. The trailing vortices form a single attachment line where both vortices reattach in a similar manner forming the trailing attachment lines. Figures 6.20(a)- 6.20(d) show that as the trailing vortices begin to break down, the line becomes very distorted. Experimental and
computational results of similar cylinder in cross flow simulations have shown the formation of the 
stagnation attachment line at the same location [36].

The attachment lines begin to converge upon each other as the iterative process continues. 
This is evident in Figure 6.20(d) where the two feature sets appear to be mostly the same. At 
10% convergence, the cylinder attachment line has formed and appears to be lying on top of its 
final spatial location. The trailing attachment line is beginning to form, but is far from being 
completely formed. At 30% convergence, both the cylinder and trailing attachment lines have 
formed. The trailing attachment line appears very distorted downstream because many of these 
eextractions reside in the vortex breakdown region. The vortex breakdown region creates very 
distorted separation and attachment results. At 55% convergence, the trailing line appears to have 
moved even closer to its final location but is still not there yet. Since at 85% convergence, the 
separation line extractions found their final spatial locations, it makes sense then that this same 
conclusion is drawn for the attachment lines since they are physically tied together because a 
separation and attachment line pair exists for each vortex.

6.3.2 Separation and Attachment Lines in Converging Datasets Processed by Agents

The separation lines extracted by the PV algorithm are shown in Figures 6.21(a)- 6.21(d), 
and are colored by probability expectation at 15%, 30%, 55%, and 85% convergence. These 
feature sets are plotted versus the fully converged dataset (represented in black) for comparison. 
Figure 6.21(a) shows that the probability expectation is only high (≈ 0.90 – 0.95) along the side 
of the cylinder at 15% convergence. The other extractions have probability expectation values of 
≈ 0.5 – 0.75, suggesting that these extractions have low confidence in these separation lines at this 
point of the iterative process.

At 30% convergence, Figure 6.21(b) shows that the main separation line along the side of 
the cylinder possesses very high probability expectation (≈ 1.0) where it is straight, but has lower 
values (≈ 0.75 – 0.9) near the top and bottom of the cylinder. The lower values occur because these 
lines are more curved than the lines along the center of the cylinder. Curved separation lines are 
a weakness of the PV algorithm. Additionally, the trailing separation lines are beginning to take 
proper shape closest to the cylinder as evidenced by the high probability expectation (≈ 0.9 – 1.0).
Figure 6.21: Comparison of separation lines extracted by PV at 15% (a), 30% (b), 55% (c), and 85% (d) convergence. The black dots represent the final converged dataset. Flow is moving from the top left to the bottom right.

The region in front of the cylinder is the separation line from the formation of the horseshoe vortex, but the rest of the line is not strong enough to have been detected here.

The 55% convergence in Figure 6.21(c) shows a larger probability expectation in the downstream portions of the trailing separation lines (from $\approx 0.5 - 0.6$ to $\approx 0.65 - 0.80$). This increase occurs because the separation lines are close to their final spatial locations and because the pressure
and velocity differences are beginning to properly develop. The same occurs for the line in front of the cylinder.

Finally, the extractions in Figure 6.21(d) at 85% convergence have high probability expectation values ($\approx 0.9 - 1.0$) along the side of the cylinder. This suggests that these extractions are real separation lines. Near the bottom and top of the cylinder, the probability expectation is lower ($\approx 0.7 - 0.8$) because of their curvature. The trailing separation lines have probability expectation values $\approx 0.8 - 1.0$, suggesting that the portions just aft of the “v” split are real but less believable as they progress downstream. Finally, the line from the horseshoe vortex has probability expectation values ($\approx 0.7 - 0.9$), but since the rest of the horseshoe vortex is not extracted it is difficult to visualize the rest of the separation associated with this vortex.

Figures 6.22(a) – 6.22(d) show the attachment line extractions generated by the PV algorithm colored by probability expectation at 15%, 30%, 55%, and 85% convergence. Similar to separation lines, these iterations are plotted against the fully converged dataset (represented in black). Figure 6.22(a) shows that the probability expectation is very high ($\approx 0.95 - 1.0$) on the cylinder attachment lines on the front of the cylinder. This makes sense because it is very near its final spatial location. The trailing attachment line in the middle of the domain has a few extractions where probability expectation $\approx 0.70 - 0.80$ near its upstream formation point, but overall the probability expectation is low ($\approx 0.50 - 0.7$). The trailing attachment line probability expectation is low because the trailing vorticies are far away from this line, thus the pressure and velocity differences are lower than those in the cylinder attachment line and are not fully developed yet.

At 30% convergence, Figure 6.22(b) shows that the entire trailing attachment line now has increased probability expectation values ($\approx 0.60 - 0.9$) compared to the previous iteration ($\approx 0.60 - 0.75$). The large value range is because the line is more probable near its inception, but decreases greatly as it proceeds downstream. This increase is due to the trailing vorticies beginning to take a more definite shape.

The 55% convergence in Figure 6.22(c) shows a similar probability expectation distribution in both the cylinder and trailing attachment lines. This makes sense because the simulation residuals are beginning to level off, making the changes in the extractions minimal.

Figure 6.22(d) shows the attachment lines at 85% convergence colored by probability expectation. A probability expectation of $\approx 0.75 - 0.95$ within the forward portions of the trailing
Figure 6.22: Comparison of attachment lines extracted by PV at 15%, 30%, 55%, and 85% convergence. The black dots represent the final converged dataset. Flow is moving from the bottom left to the top right.

separation line suggests that these extractions may be a real attachment line, although weak. At approximately 1/2 of the trailing line length, the trailing vorticies begin to break down, giving the attachment line lower probability expectation values ($\approx 0.70 - 0.80$). Since these are lower, they are less believable. The cylinder attachment line displays high probability expectation ($\approx 0.95 - 1.0$) for much of the line and does not deviate from this. Because the probability expectation is so high, it will be considered a real attachment line. However, the extractions in front of the cylinder.
are relatively low ($\approx 0.60 - 0.75$) and a low confidence leads to them not being considered real attachment lines.

### 6.3.3 Comparison of Separation Lines Processed by Agents from Converged Solutions

The agents form their final opinions on the fully converged separation lines extracted by the PV and PP algorithms. These are displayed in Figure 6.23. Both algorithms present high belief values within the separation lines along the side of the cylinder ($\approx 0.9 - 1.0$) and at the beginning of the trailing vorticies ($\approx 0.85 - 0.95$). This is shown in Figures 6.23(a) and 6.23(b). High values in these lines suggest that these extractions contain multiple strengths of both extraction algorithms (high pressure and velocity differences), as discussed in Sections 5.3.1 and 5.3.3.

The lines that only have some or none of the strengths of the algorithms at all display higher disbelief values. This means that these separation line extractions are physically weak (having low pressure and velocity differences). Disbelief values of $\approx 0.0 - 0.2$ exist on the cylinder and forward portions of the trailing separation lines (Figures 6.23(c) and 6.23(d)). These values are low because they contain at least some of the strengths of each algorithm. The extractions near the bottom of the aft portion of the cylinder are higher $\approx 0.3 - 0.5$ suggesting that these extractions do not contain all or any of the strengths of the extraction algorithms.

Figures 6.24(a) and 6.24(b) display low values of uncertainty ($\approx 0.0 - 0.2$) in the separation lines along the side of the cylinder and at the beginning of the trailing vorticies. With the uncertainty values being low, a researcher can be assured that the features have low values of wall shear stress and local velocity magnitude. Recall in Sections 5.3.1 and 5.3.3 that when these values are low, the agents can expect the extractions to be real separation lines, weak or not. Similar to shock wave extractions, low uncertainty values tell the researcher that the $FD$ and $\Delta FD$ values are near zero, meaning that extractions have reached their final locations. Farther downstream as the trailing vorticies break down, the uncertainty is higher ($\approx 0.3 - 0.5$). This occurs because the physical attributes of a separation line (pressure difference, velocity difference, and local velocity magnitude) are not strong in these breakdown regions. These higher uncertainty values suggest that the agents are not as certain that these extractions are real separation lines.

The probability expectation values of the separation lines are shown in Figures 6.24(c) and 6.24(d). These values are high at $\approx 0.85 - 1.0$ along the side of the cylinder and at the beginning
Figure 6.23: Comparison of belief and disbelief components of the belief tuple for the separation lines in the converged cylinder in a cross flow dataset. Flow is moving from left to right.

of the trailing vortices. Equation 2.19 says that if the belief is high and the uncertainty is low, high probability expectation will result. This is the case in strong separation lines, but not in the weaker extractions. Thus, the weaker extractions have lower probability expectation values. In this case, it is assumed that any extraction with a probability expectation greater than 0.85 is probably a real separation line because they are the most probable of all the extractions. Once again, 0.85 is not a concrete number to be as a cut-off used on all datasets and is specific to this dataset only and is subjective to the experience and opinion of the user.
Figure 6.24: Comparison of the uncertainty and probability expectation components of the belief tuple for the separation lines in the converged cylinder in a cross flow dataset. Flow is moving from left to right.

6.3.4 Comparison of Attachment Lines Processed by Agents from Converged Solutions

Figure 6.25 shows the attachment lines extracted by the PV and PP algorithms after the agents have formed their final opinions on the extractions. The feature sets’ belief values are high within the attachment lines along the front of the cylinder ($\approx 0.9 - 1.0$) and at the beginning of the trailing vorticies ($\approx 0.8 - 0.9$). The high belief of the forward portions of the trailing vorticies makes sense because the belief of the forward portions of the trailing separation lines are also high.
\( \approx 0.85 - 0.95 \), as shown in Figures 6.25(a) and 6.25(b). According to Sections 5.3.1 and 5.3.3, high belief values are calculated because these extractions contain at least some of the strengths of the extraction algorithms (high pressure and velocity differences).

If no algorithm strengths are present or only some of them are present, the lines will be calculated to have higher disbelief values. Figures 6.25(c) and 6.25(d) show that disbelief values of \( \approx 0.0 - 0.2 \) exist along the cylinder and trailing attachment lines. These low disbelief values

Figure 6.25: Comparison of belief and disbelief components of the belief tuple for the attachment lines in the converged cylinder in a cross flow dataset. Flow is moving from right to left.
occur because these extractions contain many or all of the strengths of the extraction algorithms. Conversely, the extractions near the bottom of the cylinder have higher disbelief values because the extractions do not contain any, or only have some, of the extraction algorithms’ strengths.

Figures 6.26(a) and 6.26(b) show low uncertainty values ($\approx 0.0 - 0.2$) in the attachment lines along the front of the cylinder and at the beginning of the trailing vorticies. Similar to separation lines, low uncertainty values mean that the features contain low values of wall shear stress and local velocity magnitude. When these uncertainty values are low, the agents are fairly confident that the extractions are real attachment lines. Also, the $FD$ and $\Delta FD$ are nearly equal to zero, meaning they are not moving anymore between iterations. As the trailing vorticies break down, it can also be seen that the uncertainty is higher at $\approx 0.3 - 0.5$ farther downstream. This occurs because the velocity and pressure difference values are not strong in these breakdown regions.

Figures 6.26(c) and 6.26(d) show that both of the algorithms display high probability expectation values ($\approx 0.85 - 1.0$) along the front of the cylinder and at the beginning of the trailing vorticies. Similar to the shock wave analysis, Equation 2.19 informs the researcher that if the belief is high and the uncertainty is low, then a high probability expectation will be calculated. This occurs in the strong separation and attachment lines, but not in the weaker extractions. Thus, the weaker extractions will have a lower probability expectation. For these particular attachment lines, any extraction with a probability expectation value greater than 0.85 is probably a real separation line because according to subjective logic they are the most probable of all the extractions.

6.4 Delta Wing

A highly swept (75°) delta wing [37] CFD simulation was used to generate a dataset that could validate the developed separation and attachment line belief tuple results. The Reynolds Averaged Navier-Stokes (RANS) equations were calculated using FLUENT 12. The delta wing has a chord length of 0.567 m, a span width of 0.304 m, and a thickness of 0.03 m. The grid was generated in Gambit as an unstructured tetrahedral mesh with 1,389,474 nodes. A 0.010” radius was added to the edges of the wing to reduce the number of highly skewed elements. A portion of the mesh is shown in Figure 6.27, where the blue represents the flow domain and the red represents the mesh on top of the wing. The inlet Mach number was set as $M = 0.3$ with an angle of attack of 20.5°. The Reynolds number was set to $Re = 1e6$. The viscous model was set to “laminar”
Figure 6.26: Comparison of the uncertainty and probability expectation components of the belief tuple for the attachment lines in the converged cylinder in a cross flow dataset. Flow is moving from right to left.

(meaning the solver will neglect the viscous terms in the governing equations) in order to match the results generated by Ekaterinaris and Schiff [37]. Air was the working fluid, and the simulation utilized the pressure-based, Least Square Cell Based Gradient solution method. The simulation was then run to full convergence (3500 iterations), and the residuals are displayed in Figure 6.28. This simulation provides a more complex dataset that can validate the separation and attachment extraction methods and belief tuple calculations.
Similar to the cylinder in a cross flow analysis, the separation and attachment lines from this delta wing simulation were extracted using the Parallel Vectors and Phase Plane methods defined in Chapter 5 – each creating its own unique extraction feature set. The feature sets were then processed by the subjective logic equations to find a belief tuple for each extraction.

A close-up of the fully converged delta wing simulation is represented in Figure 6.29. Figure 6.29 shows the Surface LIC lines that represent the separation and attachment lines. There are three pairs of separation and attachment lines for the delta wing dataset: the primary, secondary, and tertiary. However, in this dataset, the tertiary separation and attachment lines are small and difficult to spot without feature extraction and are not shown in this figure. The three separation and attachment line pairs form as a result of the vorticies forming along the upper surface of the wing. As the vorticies rotate, they pull the fluid away from the surface in order to supply flow to the vorticies, causing the fluid to separate from the surface. Experimental and computational results of this delta wing have shown the formation of these separation and attachment line pairs. The spatial locations and strengths of each of these lines are similar to the results generated by Ekaterinaris [37]; this is shown in a graphical representation of these vorticies in Figure 6.30.
each vorticity there is an associated separation and attachment line. These are extracted in the same areas as shown in Figures 6.30 and 6.31. Ekanterinaris left the tertiary lines out of Figures 6.30 and 6.31 to keep the figures simple.

6.4.1 Definition of Delta Wing Separation and Attachment Lines

Figure 6.32 shows the separation and attachment lines extracted by the PV and PP extraction algorithms. In these figures, flow is moving from the bottom left to the upper right. The red lines near the outer portions of the delta wing (shown in Figures 6.32(a) and 6.32(b)) represent the primary separation line pair. These separation lines are associated with the primary vortex pair formed along the upper surface of the wing. The blue lines represent the secondary separation lines. These are associated with the smaller secondary vortex pair. The green lines represent the tertiary separation line pair. These lines are associated with the even smaller tertiary vortex pair. The black extractions will later be shown to be erroneous extractions.
Figure 6.29: Close-up representation of the delta wing dataset. Surface LIC lines represent the different separation and attachment lines. Additional lines are added to better show the primary and secondary separation and attachment lines.

Figure 6.30: A backside looking forward view of the primary and secondary vortices forming on top of the delta wing [37].

The red line along the center of the delta wing in Figures 6.32(c) and 6.32(d) represents the primary attachment line. This attachment line is shared by the primary vortex pair as shown in Figure 6.30. This is a common occurrence on slender (70 or higher) delta wing geometries. However, many of these extractions do not form a coherent straight line as compared to the separation lines; this is difficult to see zoomed out because the line simply appears thicker. A close-up analysis or a very zoomed-in view of the primary attachment line would show that the extractions
Figure 6.31: A top view of the primary and secondary vorticies LIC lines forming on top of the delta wing [37].

are more widely scattered. This spreading or scattering occurs because the primary attachment line is weaker than the other attachment lines. Figure 6.33 helps describe why this line is weak throughout most of the chord length. The black line represents the mid-line of the wing, where we would expect the attachment line to be. The left and right ($\pm z$) components of the velocity vectors surrounding the black line are almost non-existent, making them appear parallel to the line. This leads to low calculated velocity difference values, and a weak attachment line. It must be noted at this point that even though this line is weak, it is still a real attachment line and subjective logic will calculate a low probability for it.

The blue lines in Figures 6.32(c) and 6.32(d) near the outer portions of the wing represent the secondary attachment lines which are associated with the smaller secondary vorticies. These secondary vorticies are also shown in Figure 6.30. These lines are just inside of the wing tips. The green lines represent the tertiary attachment line pair, and are associated with the small tertiary vortex pair. These attachment lines are associated with each respective separation line. The separation lines that appear along the trailing edge of the delta wing are extracted because the flow separates from the wing at this point. The extracted attachment lines in this same area have been erroneously extracted, this will be shown later when their probability expectation values are calculated. Again, the black extractions will be proven erroneous.
Figure 6.32: Display of separation and attachment lines extracted by PV and PP algorithms at full solution convergence.

6.4.2 Comparison of Separation Lines Processed by Agents from Converged Solution

Figures 6.34(a) and 6.34(b) show the pressure difference values of the separation lines from the converged solution. Recall that Sections 5.3.1 and 5.3.3 describe that high pressure difference values are a strength of both algorithms. The pressure differences are \( \approx 0.90 - 1.0 \) throughout all of the primary separation lines. The primary vortex pair breaks down at approximately 80% chord length. This is evidenced by the primary vortices suddenly decreasing in strength as shown in Figure 6.35(b) between the 75% and 90% chord planes. Additionally, Soemarwoto and Boelens [38] used the Coefficient of Pressure \( (C_p) \) as a parameter to help detect vortex breakdown. They showed that when a vortex breaks down, \( C_p \) suddenly increases from a negative value to a higher
Figure 6.33: A cross-section of the velocity vectors above the delta wing, colored by helicity at 75% chord.

value. This happens in Figure 6.35(a) where the marked dark blue region transitions to green just aft of the 75% chord plane. Even though primary vortex break down is occurring, the primary separation lines are not affected much by the primary vortex breakdown because these separation lines are directly related to the flow detaching as it leaves the underside of the wing.

Figure 6.34: Display of separation lines extracted by PV and PP algorithms colored by normalized pressure differences at full convergence.

The secondary separation lines undergo an interesting outward shift at approximately 45% chord. This outward shift begins as the boundary layer of the outward bound (±z) velocity sup-
Figure 6.35: Side views of the delta wing dataset at full convergence with the surface being colored by $C_p$, and two cut planes showing the primary vorticies at 75% and 90% chord colored by helicity.

plied from the primary attachment region transitions from laminar to turbulent (the boundary layer, not the vortex). Since a turbulent boundary layer stays attached longer than a laminar boundary layer, the secondary separation line forms farther outboard than it does upstream. The secondary separation line shift is represented in this dataset by an intermediate stepped pattern rather than a curved continuous line; this is because neither extraction algorithm performs well at extracting curved separation or attachment lines. The secondary separation lines farther downstream begin to lose their strength at about 90% chord. They show a high pressure difference $\approx 0.90 - 1.0$ along the forward portions, but begin to lose strength at approximately 90% chord. This loss of strength is a consequence of the secondary vorticies breaking down at about 85% chord. The evidence of this breakdown is shown in Figure 6.37(a) as the marked $C_p$ values transition from blue to green, while the vortex suddenly increases in size in Figure 6.37(b), and loses strength and structure. Once the vortices break down, it takes some distance for the separation and attachment lines to be affected, in this case about 5% chord. This distance can vary depending on the size and strength of the associated vorticies. Since the secondary separation lines have now shifted outward, the line that appears to continue straight downstream appropriately decreases in strength quickly after the boundary layer transition because they are no longer associated with the secondary vorticies.

Similar to the primary and secondary lines, the tertiary separation lines retain high pressure difference values ($\approx 0.85 - 1.0$) until approximately 95% chord where the values drop to $\approx 0.50 - 0.70$ as a result of tertiary vortex breakdown. The lines that are extracted on the trailing edge of
the delta wing are separation lines formed from the flow leaving the delta wing, both on the top and bottom. The attachment lines extracted here will be shown later to not be probable extractions. The other extractions that appear between the secondary and tertiary lines will also be shown to be not very probable.

Figures 6.36(a) and 6.36(b) display the normalized velocity difference values of the extracted separation lines from the PV and PP algorithms. Sections 5.3.1 and 5.3.3 demonstrate that high velocity difference values are also strengths of both algorithms. Both feature sets display normalized velocity difference values $\approx 0.95 - 1.0$ throughout much of the forward portions of the separation
lines. As the secondary lines extend downstream and approach 1/2 chord (the boundary layer transition region) they begin to lose strength, dropping to $\approx 0.70 - 0.90$. They regain their strength though after the transition. Similar to Figures 6.34(a) and 6.34(b), the tertiary lines maintain high velocity difference values $\approx 0.95 - 1.0$ until approximately 95% chord, as a result of break down.

The normalized velocity magnitude values of the separation lines extracted by the PV and PP algorithms are shown in Figures 6.38(a) and 6.38(b). Low normalized velocity magnitude values physically describe a separation line (Sections 5.3.1 and 5.3.3). Both feature sets show normalized velocity magnitude values of $\approx 0.0 - 0.10$ within the forward portions of the primary, secondary, and tertiary separation lines. The primary separation lines display a consistently low normalized velocity magnitude because they are very strong throughout their length. Because of the transition of the secondary vorticies, the normalized velocity magnitude values in the secondary separation line begin to increase to $\approx 0.10 - 0.25$ approaching 1/2 chord and decrease back to $\approx 0.0 - 0.10$ after transition. Here the line that continues downstream of the forward portion of the secondary separation lines can be deceiving. If a researcher did not look at these parameters, he might be led to believe these extractions, but according to the pressure and velocity values they are not likely to be real or accurate separation lines. These parameters will be combined later to form a final probability expectation value to help with this analysis.

Figures 6.38(c) and 6.38(d) show the normalized wall shear stress values of the separation lines. Similar to the normalized velocity magnitude, low values of wall shear stress describe a separation line physically (Sections 5.3.1 and 5.3.3). Normalized wall shear values of $\approx 0.0 - 0.10$ are represented throughout the entirety of the primary separation lines. However, similar to the normalized velocity magnitude feature sets, the secondary separation line wall shear stress extractions increase to $\approx 0.20 - 0.40$ as they reach the 1/2 chord line but decrease after transition. If a researcher only used this parameter, an erroneous belief may be formed on these midstream extractions and disregard them as false. The tertiary separation lines maintain their low normalized wall shear values until approximately 90% chord where all the lines begin to break down.

The final parameter to consider is the normalized curvature of each line. Figures 6.39(a) and 6.39(b) show the normalized curvature of the separation lines extracted by the PV and PP algorithms. A low normalized curvature value is a strength, according to both PV and PP algorithms (Sections 5.3.1 and 5.3.3). Both feature sets show high normalized curvature values of
≈ 0.40 – 0.50 near the nose of the wing because the lines are bunched together and the algorithms had difficulty determining which point belonged to which line. This is an inherent challenge with the curvature calculations. Since each point is not numerically tied to another point, the curvature is calculated from its 2\textsuperscript{nd} closest point, even if it is not associated with the same particular line. If the lines are straight, the curvature is appropriately low (below approximately 0.01). Farther downstream the feature sets show normalized curvature values of ≈ 0.0 – 0.20 within the most of both the separation lines. Vortex breakdown does not appear to affect the curvature calculations of these extractions until past the 75% chord where many of the extractions downstream are high (≈ 0.40 – 0.50), particularly the primary separation lines.
The strengths and weaknesses of each algorithm were then combined into a belief tuple using the subjective logic architecture defined in Sections 5.3.2 and 5.3.4. Figure 6.40 shows the belief and disbelief components of the delta wing belief tuple. Figures 6.40(a) and 6.40(b) display the belief values of each algorithm. Within the forward portions of all three separation line pairs, the belief values are \( \approx 0.85 - 1.00 \). The belief is high in both the PV and PP feature sets because the normalized pressure difference and the normalized velocity difference (Figure 6.11(c)) are high (\( \approx 0.90 - 1.00 \)) in these regions. Additionally, the normalized curvature is low (\( \approx 0.0 - 0.20 \)). Because these are strengths of the PV and PP algorithms, the belief becomes associatively high. The primary separation line extractions downstream of 90% chord have lower belief values \( \approx 0.70 - 0.80 \) because not all of the strengths of the PV and PP algorithms are represented with high values. The secondary separation lines approaching the 1/2 chord line have belief values slightly lower than their upstream portions (\( \approx 0.80 - 0.95 \)) as a result of the secondary vortex beginning transition. The downstream portions of the secondary lines have belief values \( \approx 0.90 - 0.10 \). The extractions near the trailing edge of the wing and between the real separation lines have understandably low belief values because most of the algorithm strengths were not well represented. The tertiary lines show a similar pattern as the primary separation lines, in that they have high belief upstream (\( \approx 0.85 - 1.0 \)), but decrease in belief (\( \approx 0.50 - 0.65 \)) around 90% chord.
Figure 6.40: Comparison of belief and disbelief components of the belief tuple for the separation lines in the converged dataset. Flow is moving from left to right.

The disbelief values of these separation lines are shown in Figures 6.40(c) and 6.40(d). Because the normalized curvature is $\approx 0.00 - 0.20$ and normalized pressure and velocity differences are high ($\approx 0.90 - 1.00$) within the forward portions of the three main separation line pairs, their disbelief values are associatively low ($\approx 0.00 - 0.15$). Similar to the belief values, the disbelief values of the primary separation lines increase to values of $\approx 0.20 - 0.35$ just after 90% chord as a result of breakdown of their respective vorticies. This increase occurs because as the vorticies break down, their respective strengths decrease, causing the belief to increase.

The uncertainty and probability expectation values of the delta wing belief tuple of the extracted separation lines are shown in Figure 6.41. The uncertainty in Figures 6.41(a) and 6.41(b) display low values of uncertainty $\approx 0.00 - 0.10$ in the forward portions of the primary, secondary,
and tertiary separation lines. These values are low because the lines have settled onto their final spatial locations. The values are also low because the normalized velocity magnitude and wall shear stress uncertainty values are low ($\approx 0.00 - 0.20$) within the forward portions of the lines. The secondary line shows slightly higher uncertainty values ($\approx 0.10 - 0.25$) at the beginning and end of the 45% chord boundary layer transition region because wall shear stress and normalized velocity magnitude are higher here. For all of the separation lines, the uncertainty increases to $\approx 0.35 - 0.50$ around 90% - 95% chord due to the effects of vortex breakdown. Vortex breakdown causes an increase in uncertainty because vortices are now weaker, meaning that they no longer have low values of velocity magnitude and wall shear stress. This occurs because more turbulence is induced as the vortices transition to breakdown. Similar to the cylinder in a cross flow simulation, higher uncertainty values mean the agents are not as confident about these extractions as those extractions in the forward portions of the three main separation lines.

Figures 6.41(c) and 6.41(d) show the extracted separation lines colored by probability expectation. By combining the belief, disbelief, and uncertainty values, the probability expectation of the separation lines is calculated (Section 2.3.2). The expected probability is $\approx 0.9 - 1.0$ for both algorithms within the forward portions of the primary, secondary, and tertiary separation lines. Because their belief values are high, at $\approx 0.90 - 1.00$, and the uncertainty is low ($\approx 0.0 - 0.10$), it makes sense that the probability expectation is high in these regions. This means that these separation lines have high strength (high pressure and velocity differences) and also meet the physical definitions of a separation line. The primary separation line extractions, as well as the secondary and tertiary lines downstream of the 90% chord, all display a decrease in probability expectation to $\approx 0.50 - 0.70$ because as a consequence of the associated vortices breaking down, they lose their strength. As they lose their strength, each extraction is no longer represented with high values of the algorithms’ strengths, decreasing their belief, and increasing the disbelief and uncertainty. All these lead to the decrease in probability expectation. As shown in the belief and disbelief values, the associated probability expectation values are slightly lower ($\approx 0.85 - 0.90$) near the beginning and end of the transition regions of the secondary separation lines. The extractions between the secondary and tertiary lines mostly have low calculated probability expectation values ($\approx 0.50 - 0.70$), again this shows that they are probably not real separation lines.
A researcher can safely assume that an extraction with a probability expectation value larger than 0.75 is probably an actual separation line because these extractions contain all or most of the algorithm strengths as defined in Sections 5.3.1 and 5.3.3. Similar to the previous datasets, this value is subjective to the user and is not a concrete value. Any extraction below 0.75 appears to be questionable and would need greater inspection by the researcher. These results verify the separation lines described by Ekaterinaris [37]. Similar to shock wave extraction, this method shows that using only one of the strength characteristics can be misleading. For example an extraction may have one high strength but another strength may be low. Without combining these, it would be difficult to determine if it was a valid extraction. Combining all strengths and weaknesses into a single parameter effectively shows which extractions are real and which are not.
The separation line subjective logic algorithms showed that for the delta wing simulation, the strong lines were represented with high probability expectation values. However, the lines with lower probability expectation values were shown to be weaker. Even though the algorithms did well at calculating an expected probability, a few false lines were still extracted with relatively high probability that can be somewhat misleading to the researcher.

6.4.3 Comparison of Attachment Lines Processed by Agents from Converged Solution

The attachment lines from the converged delta wing simulation are shown in Figures 6.42(a) and 6.42(b) colored by pressure difference. Similar to separation lines, high pressure difference values are strengths of both algorithms (Sections 5.3.1 and 5.3.3). The primary attachment line pressure differences are \( \approx 0.90 - 1.0 \) throughout only the forward-most portions of the line. After approximately 1/10 chord length, the pressure difference drops to \( \approx 0.50 \) downstream. This occurs because the primary vorticies move parallel to the diverging outer wing tips. Near the nose of the delta wing, the vorticies are close together, making the middle primary attachment line strong. But as the vorticies move away from each other downstream, the strength of the attachment line decreases. The pressure differences are \( \approx 0.90 - 1.0 \) throughout most of the secondary and tertiary attachment lines. Similar to the separation lines, past the 90% of the chord length, the secondary and tertiary attachment lines begin to lose strength. Physically, this shows that the secondary and tertiary vorticies dissipate as they extend farther downstream of the initiated vortex breakdown. This phenomenon is picked up by both the PV and PP feature sets. Notice that the secondary attachment line does not shift like the separation line. This shows that the vortex is changing size, not location. The other extractions that appear between the secondary and tertiary lines will be shown later to be false extractions because they are not well represented by high algorithm strengths.

Figures 6.42(c) and 6.42(d) display the normalized velocity difference values of the attachment lines extracted by the PV and PP algorithms. Sections 5.3.1 and 5.3.3 describe why high velocity difference values are also strengths of both algorithms. As discussed previously, this value is also a strength of separation lines, therefore we would expect to see similar patterns with the associated attachment lines. The velocity difference values throughout most of the forward portions of the attachment lines are \( \approx 0.95 - 1.0 \) for both feature sets displayed. As the primary attachment
line extends downstream to approximately 90% chord, the normalized velocity difference drops to about \( \approx 0.50 - 0.70 \) as a consequence of the associated primary vortices breaking down at about 80% chord. A similar pattern can be seen as the tertiary attachment lines extend downstream, they begin to drop to \( \approx 0.50 - 0.75 \) at approximately 90% chord. This again occurs because the tertiary vortices are breaking down. The secondary attachment lines retain high velocity difference values (\( \approx 0.85 - 1.0 \)) throughout all of the line. These patterns seem to follow the same separation line pattern found in Figures 6.36(a) and 6.36(b). This shows that since both patterns are somewhat similar, their respective extraction values actually validate each other.

Figures 6.43(a) and 6.43(b) display the normalized velocity magnitude values of the attachment lines extracted by the PV and PP algorithms. Similar to separation lines, low normalized velocity magnitude values physically describe an attachment line (Sections 5.3.1 and
5.3.3). The forward portions of the primary, secondary, and tertiary attachment lines from both feature sets contain normalized velocity magnitude values of \( \approx 0.0 - 0.10 \). These results coincide with their respective separation lines. The vortex breakdown of the primary and tertiary vortices causes the normalized velocity magnitude values in these attachment lines to begin to increase to \( \approx 0.20 - 0.50 \) at approximately \( 85\% - 90\% \) chord. This occurs because the local turbulence induced from the vortex breakdown causes the local velocity vectors to no longer directly diverge from each other. This means that the local velocity will increase rather than form a small stagnation region which defines an attachment line. Interestingly however, the secondary attachment lines maintain low normalized velocity magnitude values \( \approx 0.0 - 0.25 \) throughout all of their length like the separation lines do, while their respective separation lines do not. One would expect the attachment values to maintain a pattern similar to their associated separation lines. This is possibly because energy is lost due to turbulent boundary layer diffusion before before it separates; more so than that lost before re-attachment.

Figures 6.43(c) and 6.43(d) display the normalized wall shear stress values of the attachment lines extracted by the PV and PP algorithms. Low values of wall shear stress physically describe an attachment line (Sections 5.3.1 and 5.3.3), as was the case for separation lines. Normalized wall shear values \( \approx 0.0 - 0.10 \) throughout the first half of the primary and tertiary attachment line are shown for both feature sets. Similar to the normalized velocity magnitude feature sets, the extractions in these lines increase to \( \approx 0.20 - 0.50 \) downstream near \( 85\% - 90\% \) chord. The secondary attachment lines maintain their low normalized wall shear values \( \approx 0.0 - 0.10 \) throughout the whole lines.

The last parameter considered is the normalized curvature of each line. Figures 6.44(a) and 6.44(b) show the normalized curvature of the attachment lines extracted by the PV and PP algorithms. Highly curved lines are a weakness of each algorithm, therefore low normalized curvature becomes a strength of each algorithm (Sections 5.3.1 and 5.3.3). The feature sets display the normalized curvature values of \( \approx 0.45 - 0.50 \) within the primary attachment line. The curvature is high not because the line is necessarily curved, but because the points are spread widely. The points are widely spread because this line is weak, and thus the agents have a hard time determining the actual location of the line. This is a potential problem with the code in determining the location of a weak attachment line. The secondary and tertiary attachment lines show fairly low normalized
curvature values of $\approx 0.10 - 0.25$ throughout most of each line. However, as the tertiary line extends downstream of 90% chord, a familiar pattern of weakening values is found as a results of the respective vorticies breaking down.

The method for combining the strengths and weaknesses into the belief tuple using the subjective logic architecture is the same used by the separation line analysis. The belief and disbelief components of the delta wing attachment line belief tuple are shown in Figure 6.45. Figures 6.45(a) and 6.45(b) display the belief values of each algorithm. High belief of $\approx 0.85 - 1.00$ exists within the forward portions of the primary, secondary and tertiary attachments. These high values correspond to the high beliefs in their associated separation lines. For both feature sets, the belief is high because the normalized pressure difference and normalized velocity difference values are also
Figure 6.44: Display of separation lines extracted by PV and PP algorithms colored by normalized curvature at full convergence.

high ($\approx 0.90 - 1.00$). As the secondary and tertiary lines extend beyond approximately 90% and 95% chord respectively, their belief decreases as a result of their vorticies breaking down at about 85% and 90% chord respectively (6.4.2). The primary attachment line is widely spread because it is a weak attachment line, meaning its pressure and velocity differences are low ($\approx 0.0 - 0.20$). This leads the belief values to be calculated low ($\approx 0.60 - 0.70$). The extractions near the nose of the wing are slightly higher ($\approx 0.75 - 0.90$) because the attachment line is stronger here, meaning the pressure and velocity differences are higher ($\approx 0.75 - 0.90$). These values begin to decrease rapidly after 1/10 chord because the primary vortices are moving farther apart.

The disbelief values of the attachment lines are shown in Figures 6.45(c) and 6.45(d). The disbelief is low ($\approx 0.00 - 0.15$) along most of the secondary and tertiary attachment lines because the normalized pressure and velocity differences are high ($\approx 0.90 - 1.00$) and the normalized curvature is $\approx 0.00 - 0.20$. Similar to the belief values, the disbelief values of the secondary attachment lines increase to $\approx 0.20 - 0.35$ at approximately 90% chord. A similar pattern can be seen with the tertiary vortices. This increase occurs because as the vorticies break down, their respective strengths decrease, causing the belief to decrease. The primary attachment line has higher disbelief ($\approx 0.10 - 0.50$) than the other lines because it is so spread out due to the primary attachment line being so weak. This causes the attachment strength to decrease. Near the nose, the disbelief is $\approx 0.10 - 0.25$, but increases as the line extends past the 1/10 chord line due to the vorticies growing in physical size.
Figure 6.45: Comparison of belief and disbelief components of the belief tuple for the attachment lines in the delta wing converged dataset. Flow is moving from left to right.

The uncertainty and probability expectation values of the belief tuple of the extracted attachment lines are shown in Figure 6.46. The uncertainty in Figures 6.46(a) and 6.46(b) display low values of uncertainty ($\approx 0.00 - 0.20$) in the forward portions of all three attachment lines. Because the extractions have stopped moving ($FD \approx 0$ and $\Delta FD \approx 0$) and the normalized velocity magnitude and wall shear stress uncertainty values are low ($\approx 0.00 - 0.20$), the uncertainty will also be low. This is found within the forward portions of the attachment lines. These uncertainty values show a striking similarity to their separation line counterparts because the separation and attachment lines are physically tied to each other.

Figures 6.46(c) and 6.46(d) display the extracted attachment lines colored by probability expectation. The probability expectation is $\approx 0.9 - 1.0$ within the forward portions of the secondary
Figure 6.46: Comparison of uncertainty and probability expectation components of the belief tuple for the attachment lines in the delta wing converged dataset. Flow is moving from left to right.

and tertiary attachment lines. This occurs because the belief values are high (≈ 0.90 – 1.00) and the uncertainty is low (≈ 0.00) throughout most of these lines. Physically, this suggests that these lines have high attachment strength (high pressure and velocity differences and low curvature), and that they also meet the physical requirements for an attachment line (low wall shear stress and local velocity magnitudes). The secondary attachment lines maintain high probability expectation values (≈ 0.80 – 1.00) until approximately 90% – 95% chord because as a result of the associated vortices breaking down, they lose their strength at this point. As they lose their strength, each extraction is no longer represented with high values of the algorithms’ strengths — decreasing their belief, and increasing their disbelief and uncertainty. This leads to the decrease in probability expectation. A similar pattern is found with the primary attachment line. Near the nose, the proba-
bility expectation is $\approx 0.75 - 0.90$. However, after $1/10$ chord, the primary vorticies begin to grow in size causing a large loss of flow energy and the probability decreases to $\approx 0.50 - 0.70$.

Similar to the separation lines, any extraction with a probability expectation greater than 0.75 is likely an actual attachment line because these extractions contain all or most of the algorithm strengths. The separation and attachment lines demonstrated by Ekaterinaris [37] can be used to verify these results. These attachment lines also seem to correspond to their associated separation lines (see Figure 6.30). Similar to shock wave and separation line extraction, this analysis shows that using only one or two of the strength parameters can lead to erroneous conclusions without accounting for physical uncertainty. Similar to separation lines, if an extraction has only some strengths, but not others it can be difficult to determine which are real by looking at the parameters separately. Combining the strengths, weaknesses, and physical attributes, a final value helps the user determine which are real attachment lines and which are not. This extraction method effectively shows the real separation and attachment lines within a physically complex flow that can be difficult/impossible to interpret without feature extraction. Similar to the separation lines, the extractions between the secondary and tertiary lines have low probability expectation values ($\approx 0.50 - 0.70$), this shows that these are probably not real attachment lines.

The attachment subjective logic algorithms also showed that the strong lines were represented with high probability expectation values. The lines with lower probability expectation values were also shown to be weaker. Additionally, a few false lines were also extracted with relatively high probability that can be somewhat misleading to the researcher.

In general, the PV and PP algorithms performed very similarly by extracting separation and attachment lines with similar spatial locations. However, not every line was extracted the same by each algorithm. For example, if the PV left out a few parts of the line, the PP would extract them. The PP algorithm consistently extracted a few more erroneous points when compared to the PV algorithm. These differences will be beneficial when combining probability expectation values from the two algorithms because the points that were not extracted by the PV or PP algorithms can later be eliminated either by the user or within the CAFÉ architecture. Additionally, the two algorithms calculated slightly different probability expectation values for the weaker extracted lines, but maintained high values for the strong separation and attachment lines.
No one algorithm consistently extracted and calculated higher or lower values than the other. Nor did one perform better than the other in any particular situation, such as a stronger separation line. Since the algorithms performed similarly, yet did not always extract the exact same portions of the lines, it can be assumed that the algorithms are different enough to compliment each other within the agent algorithm architecture.
CHAPTER 7. CONCLUSIONS

This chapter will outline a brief summary of this research, as well as detail some conclusions and recommendations for future improvement.

7.1 Summary

This research has developed a method to extract shock wave and separation and attachment lines from converging and fully converged CFD simulations using intelligent software agents governed by subjective logic. This was done to reduce post-processing time and to enable a researcher to quickly parse through data to find desired information. This methodical combination of multiple algorithms to form an expected probability for each shock wave and separation and attachment line extraction based on each algorithms’ strengths and weaknesses is a new approach to determine the believability. The general feature extraction method used contains four basic steps which may be applied to any CFD flow feature with corresponding feature extraction algorithms. These four steps are as follows:

1. Extract features using feature extraction algorithms.
2. Filter obviously extraneous features.
3. Create agent opinions at regions contained in each extracted feature.
4. Combine agent opinions to form final opinions of features.

The shock wave and separation and attachment line extraction algorithms were then used by the agents to determine their expected probability using subjective logic. Subjective logic provided an effective method for the analysis of concurrent feature extraction by providing three logic values that were combined into a final probability expectation value. The probability expectation value accounts for belief, disbelief, and physical uncertainty and was used to help determine which extractions were real and which were not. However, because subjective logic does not give a clear
“yes” or “no” answer for these features, a researcher will still need to exercise some engineering judgment to further analyze the results.

This method shows that using subjective logic is possible to determine which extracted features are real based on individual algorithm strengths and weaknesses. The supersonic ramp and the cylinder in a cross flow simulations were used to track convergence, while the swept Onera M6 wing and delta wing simulations were used to demonstrate the method of calculation for the belief tuple.

7.2 Conclusions

The supersonic ramp simulation showed that as early as 15% into the simulation, the bow shock wave had an expected probability above 0.90 for most of the feature. This is significant because it is extracted very early on in the convergence process, giving a good indication of strength and location without waiting until 100% convergence. However, it also showed that the aft shock may not be strong enough to be considered a real shock wave. This helps the researcher because it shows that for very strong shock waves, accurate locations and strengths can be studied in greater detail early in the convergence process. The intelligent agents were able to effectively determine which portions of each shock were real and even demonstrated that the trailing shock is weak. This information is useful because even though it is in a location where compressible flow theory suggests a shock should be, disregarding it can save the researcher from further investigation of this weak shock wave.

The analysis of the cylinder in a cross flow simulation also showed that as early as 15% into the simulation, the cylinder separation and attachment lines were found to have an expected probability above 0.90 for most of the features. This discovery is again very helpful to researchers because they can be confident that the locations and strengths of the high probability lines will remain constant throughout the remainder of the convergence process.

Based on the analysis of these two simulations, it can be concluded that using intelligent agents and subjective logic is effective in determining location and strengths of extracted shock waves and separation and attachment lines.

On the fully converged swept Onera M6 wing simulation the shock wave extractions with probability expectation values greater than 0.90 were determined to be part of an actual shock. The
intelligent extraction method proved to efficiently extract the regions with the strongest, or highest, gradients (pressure, density, velocity, etc.). Extractions with large gradients and low uncertainty (meaning they meet the physical definition of a shock wave) had the highest belief and probability expectation values based on their subjective logic belief tuples. The weaker extractions and those with high uncertainty values calculated a lower probability. Using belief tuples to calculate a probability expectation quantifies the physical flow features, allowing the researcher to clearly see which extractions are real, along with the physical strength of each feature, instead of relying on experience and intuition. This directs the researcher to focus efforts on these extractions rather than focusing on misleading high pressure regions.

The CFD delta wing simulation concluded that an extraction with a probability expectation value larger than 0.75 is likely an actual separation line. The intelligent agent method performed well at calculating the strengths and weaknesses of each algorithm, as well as the strength of their physical characteristics. They showed that the probability expectation was high in regions of strong separation lines, but was low in weak lines and in the vortex breakdown and transition regions. These results prove that this method is capable of quantifying the physical strength of a separation line and assign an expected probability to it. This is significant because it shows which extractions are most probable and which are least probable at first glance without the additional time and experience required to investigate surface LIC plots and local velocity vectors.

Similar to the separation lines, any delta wing attachment line extraction with a probability expectation greater than 0.75 is safely considered an actual attachment line. The strong attachment lines with well represented physics displayed high probability expectation values. Similar to separation lines, this method allows the researcher to determine the most probable attachment lines quickly and effectively in a physically complex flow. It should be noted however that the 0.75 designation as being real is not universal, but applies only to this delta wing dataset. Other datasets may have a higher or lower cut off depending on the normalized strengths, weaknesses and physical characteristics of each extraction. These values are subjective to each user based on their experience and knowledge of attachment lines.

The shock wave and separation and attachment line subjective logic algorithms showed that for the swept Onera M6 wing and delta wing simulations, the strong features were represented with high probability expectation values. However, the features with lower probability expectation
values were shown to be weaker. Subjective logic can help interpret the flow features by showing for example that large gradients exist in the presence of a shock wave, and that local velocity is nearly zero in along a separation line. Information regarding these parameters is very useful in helping the researcher understand what physical characteristics make up the feature, and how their respective strengths affect it.

It should be noted that while the algorithms effectively calculated an expected probability for the real extractions, a few false features were still extracted and their expected probability was calculated. These false features were extracted due to the weaknesses of the individual algorithms. However, far more real features were extracted than erroneous ones because the clearly erroneous extractions were eliminated from the feature set using multiple filtering techniques before the belief tuples were calculated.

This research shows that using the combination of multiple algorithms is more effective at determining which extractions are real than either standalone algorithm. This is because it uses the multiple strengths and weaknesses of each extraction algorithm to accurately calculate the strength, uncertainty, and spatial locations of each feature. Knowing which extractions are real and their physical locations can help a researcher efficiently visualize the real features without manually sifting through large datasets searching for these features of interest.

7.3 Recommendations

This research and the CAFÉ concept are still in development. Currently the normalization of the pressure and density gradients in the shock wave algorithms are normalized by the geometric mean of the respective gradients (Sections 4.3.2 and 4.3.4). A better approach may be to determine a single global value in which to normalize these gradients, doing this may eliminate false calculation within the normalization and produce more accurate results. However, this would require the determination of a concrete gradient value at which a shock would undoubtedly exist.

The separation and attachment lines extracted by the algorithms are currently single points with no relation to their neighbors. This can cause issues when calculating the curvature (Section 5.3.1) of these lines if two lines lie very close to each other. If they are too close, then a false curvature value may be calculated. Connecting the points into a single line would allow the curvature to be calculated on that line only, eliminating possible false calculations. This could be
accomplished by determining a points neighbor and connecting them with a line. The difficulty is determining which points belong to which lines. A similar option could be applied to shock waves, making them a complete surface.

As explained in Section 3.4, the process to aggregate a final feature set is not yet automated. Future work will be to implement this aggregation process by using a simple search criterion to locate common shock waves or separation and attachment lines between the separate feature sets and then selecting the ones with the highest expected probabilities. This is a function that will be included in the final stages of the CAFÉ project.

A final recommendation would be to decrease the code run time. For example, the shock wave extraction and subjective logic calculations for the supersonic ramp simulation (Section 6.1) took approximately 36 hours to calculate, while the CFD simulation took approximately 7 hours to run. The reason for this large time difference was because the extraction and subjective logic calculations were being performed using only one processor, whereas the CFD simulation used 8 processors in parallel. Ultimately the code will be both concurrent and parallel. Converting the current code to run within a parallel architecture would eliminate this large amount of time, making it more usable and time-friendly. CAFÉ is expected to ultimately be a parallel code running on the same number of processors as the simulation.
REFERENCES


