TOWARD THE DIRECT DESIGN OF Waverider Airframes

by

Dawn Daniél Kinsey

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering )
in The University of Michigan
1998

Doctoral Committee:
Professor Philip L. Roe, Chairperson
Professor Noboru Kikuchi
Associate Professor Kenneth G. Powell
Professor Bram van Leer
To the little girl who dreamt of being a Doctor and exploring outer space.
ACKNOWLEDGEMENTS

“The only way to discover the limits of the possible is to go beyond them into the impossible.”
- Arthur C. Clarke

The Ph.D. experience provided more than an academic challenge. In the past 7 years I pushed and moved (or at least went around) many perceived limits. I owe thanks to those who watched, accompanied and aided these adventures. I am thankful to my parents and family for helping me learn to look beyond boundaries.

I would like to thank the Aerospace Department for funding my graduate work while at the University of Michigan (U of M). I sincerely enjoyed all the professors I knew in the department. The departmental secretary, Margaret Fillion, was wonderfully supportive and has been my lighthouse through the university formalities.

I definitely could not have taken this thesis across three states without the generosity of many people. For allowing me to use their computers, space and technical experts, I very heartily thank Dick McGehee at the Geometry Center, Paul Woodward at the Laboratory for Computing in Science and Engineering (LCSE), at the University of Minnesota and Joe Guthrie and Mike O’Neill in the Mathematics Department at the University of Texas at El Paso (UTEP). I especially want to thank the Technical Staff at the Geometry Center, Russell Cattelan and Joe Haberman at LCSE, Maria Barraza at UTEP and Eric Charlton, Darren DeZeeuw, and Jeff Hittinger at U of M who maintained my computers and accounts, and accommodated my needs for large amounts of disk space. I definitely pushed the limits of my available computer resources.

I am honored that Phil Roe allowed me to be his student. I greatly appreciate his guidance and allowing me to work at my own pace.
My work is more substantial thanks to Sami Bayyuk’s grid generation code and his assistance in converting it to a space-marching method.

During the time I was at U of M, I had the honor of working with some other very talented people: Ken Powell, Bram van Leer, and the members of the CFD group. It was humbling, inspiring, educational and fun to work with such well-rounded people. I will always appreciate the understanding, compassion, kind words (and, in the case of Hailey, food, lodging, transportation and a shoulder to cry on) and encouragement of the friends I made through the CFD group: Rob, Mohit, Jeff B., Jens, Hailey, Suzy W., Sami and Lisa.

The empathy that oozed from Bob Sr. and Dot helped sustain me through some pretty tough times. Thanks for sharing your experiences and “just being there” for me.

I hope I never forget the patience and support of my husband. He helped me find the will to finish by reminding me I didn’t have to. He never complained when I usurped his new computer and forced him to share his new office. He was always willing to let me talk out my research problems and edit my writings. And when deciding on a new job, he made sure I would have adequate computer resources available to finish my research. But these pale in comparison to the emotional support he provided throughout these long 7 years. Thanks hun.

And to the rest of my friends around the world, thanks for being supportive and patient.
# TABLE OF CONTENTS

DEDICATION .......................................................... ii

ACKNOWLEDGEMENTS ................................................. iii

LIST OF FIGURES .................................................. viii

LIST OF TABLES ................................................... xiii

LIST OF APPENDICES .............................................. xiv

CHAPTER

I. INTRODUCTION ................................................... 1

1.1 Historical Perspective ........................................ 1
1.2 A New Design Approach ....................................... 9
1.3 This Work ......................................................... 12

II. GOVERNING EQUATIONS ........................................... 14

2.1 The Three Dimensional Steady Euler Equations ............. 15
2.2 Discretization .................................................. 20
2.3 Calculation of Primitive Variables ......................... 21

III. COMPUTATIONAL GRID ........................................... 22

3.1 Overview of Code Algorithm ................................. 23
3.2 Adaptation of the Unsteady Grid Method to 3-D Steady Space Marching ................... 26
3.3 Geometric Refinement ......................................... 27
3.4 Gasdynamic Refinement ....................................... 30
3.5 Merging .......................................................... 31
3.6 Gasdynamics for the Computational Groups ................ 34
3.7 Summary ........................................................ 35

IV. METHOD OF FLOW SOLUTION ................................. 36
5.9 Summary ................................................. 98

VI. A COMPARISON OF WAVERIDER COMPRESSION SURFACES ............................................. 101

6.1 Preliminaries ........................................... 102
   6.1.1 Choosing the Body Shapes .......................... 102
   6.1.2 Initial Conditions ................................. 106
   6.1.3 Convergence ...................................... 107
   6.1.4 Comparison Criteria: Figure of Merit .......... 108
       6.1.4.1 Computation of Lift and Drag Forces ..... 110
       6.1.4.2 Lift-to-Drag Ratio as a Figure of Merit  111
6.2 Summary of Case Study Results ....................... 112
   6.2.1 Evaluation of the Process ....................... 118

VII. BEYOND EULER FLOWS AND EULER CODES ............ 121

7.1 Important Second-Order Effects In Waverider Flow .... 122
   7.1.1 Viscous Effects .................................. 122
   7.1.2 Effects of Turbulence ............................ 124
   7.1.3 Real Gas Effects .................................. 124
   7.1.4 The Continuum Assumption ....................... 127
   7.1.5 Inclusion of Viscous and Real Gas Effects .... 127
7.2 Benefits of Direct-Design Approach ................... 130

VIII. FUTURE WORK AND CONCLUSIONS ..................... 132

8.1 Suggested Improvements to Current Methods .......... 133
   8.1.1 Initial Condition Specification .................. 133
   8.1.2 Body Shapes ..................................... 135
   8.1.3 Grid Code Improvements ......................... 135
   8.1.4 Optimization .................................... 136
   8.1.5 Unexpected Flow Features ....................... 136
8.2 Advantages of Direct-Design Approach ................ 137
8.3 Enhancing the Usefulness of Direct-Design .......... 138
8.4 Conclusions ........................................... 138

APPENDICES ................................................. 139

BIBLIOGRAPHY .............................................. 174
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Examples of Simple Waverider Shapes. (The top right body is a caret wing as described in Appendix A.)</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>A Wind Tunnel Model of a Waverider Designed Using Osculating Cones by Miller and Argrow [48].</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Some Examples of Complicated Waverider Shapes</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Examples of Waverider-Like Designs. These Figures Originated From Reference [14].</td>
<td>10</td>
</tr>
<tr>
<td>2.1</td>
<td>The Computational Coordinate System - x is the Marching Direction</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>An Uncut Cell</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Stacking Computational Planes to Form a Body in Three Dimensions</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>Examples of How Cells are Cut by the Body to Produce Cut-Cells</td>
<td>24</td>
</tr>
<tr>
<td>3.3</td>
<td>A Grid Plane Which Shows the Automatic Grid Refinement on a Body Cross-Section and Along the Shock Region</td>
<td>25</td>
</tr>
<tr>
<td>3.4</td>
<td>Representation of Sharp Leading Edge on Cartesian Grid</td>
<td>29</td>
</tr>
<tr>
<td>3.5</td>
<td>Small Cut-Cells Disappear From the Computational Grid as the Body Cross-Section Changes Position Relative to the Grid Between the i and i + 1 Plane.</td>
<td>31</td>
</tr>
<tr>
<td>3.6</td>
<td>Cut-Cells are Merged into Computational Groups so That The Computational Group Exists in Both the i and i + 1 Planes.</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>CFL Condition in y-direction</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Cut Cell With Projected Wall Face Areas</td>
<td>44</td>
</tr>
</tbody>
</table>
A Close-up of Density Contours on a Caret Wing with an Embedded Shock

A Close-up of Pressure Contours on a Caret Wing with an Embedded Shock

The Computed Pressure Contours on a Delta Wing: \( \Lambda=55.715, \delta=9.265, \text{ Mach}=6.0 \)

The Computed Pressure on the Surface of a Delta Wing: \( \Lambda=55.715, \delta=9.265, \text{ Mach}=6.0 \)

The Computed Density on the Under-Surface of a Delta Wing: \( \Lambda=55.715, \delta=9.265, \text{ Mach}=6.0 \)

A Close-up of the Leading Edge Density Profile on the Surface of a Delta Wing

A Zoom of the Transition Region Between the Uniform and Non-uniform Regions on the Surface of a Delta Wing

Examples of the Range of Shapes Achievable With a Quartic Definition

Base Caret Wing Compared to Quartic Under-Surface Test Shape

Quartic Under-Surface Parametric Constraints Shown in Parameter Space

An Example of a Truncated Leading Edge, The Initial Points and The Body After Splination

Typical Convergence History on Quartic Test Bodies

Comparison of Norm-Converged Solution and Steady-State Solution

Projected Areas of Body and Differential Surface Element for the Calculation of Lift and Drag

Comparison of Test Cases to the Thin Shock Layer Hypersonic Limit and Exact On-Design Caret Wings for \( \text{Mach}_\infty=7.199 \)

Increase in Volume and \( C_l \) Over Computed Original Caret Wing
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Approximate Number of Iterations and Run-Time Hours Needed for Different Levels of Convergence</td>
<td>56</td>
</tr>
<tr>
<td>5.2</td>
<td>A Comparison of Expected and Computed Solution Over a Caret Wing: $\omega=6.733$, $\Lambda=80.0$, Mach=7.199.</td>
<td>67</td>
</tr>
<tr>
<td>5.3</td>
<td>The Effects of Blunting the Leading Edges on a Caret Wing</td>
<td>74</td>
</tr>
<tr>
<td>5.4</td>
<td>A Comparison of Expected and Computed Average Solution Over a Delta Wing: $\Lambda=55.715$, $\delta=9.265$, Mach=6.0</td>
<td>93</td>
</tr>
<tr>
<td>5.5</td>
<td>A Comparison of Expected and Computed Within the Uniform Region of a Delta Wing Solution: $\Lambda=55.715$, $\delta=9.265$, Mach=6.0: $\Lambda=55.715$, $\delta=9.265$, Mach=6.0</td>
<td>96</td>
</tr>
<tr>
<td>6.1</td>
<td>Results of Parameterized Under-surface Test Cases</td>
<td>113</td>
</tr>
</tbody>
</table>
LIST OF APPENDICES

Appendix

A. Inverse Method of Waverider Design ........................................ 140

B. Using Parameter Vector in Derivation of Eigenvalues and Eigenvectors ........ 144

C. Body Parametrizations .......................................................... 146

D. Calculation of Leading Edge Flow on Waverider Bodies Using Oblique Shock Theory ................................................................. 151
   D.1 Body Angle Definitions for Caret and Delta Wings ..................... 151
   D.2 Approximating The Flow State on a Caret Wing Using 2-D Oblique Shock Theory ................................................................. 154
      D.2.1 Computing Using a Coordinate System with X-axis Along Bottom Ridge Line ................................................................. 160
      D.2.2 Calculating The Shocked State ........................................ 162

E. Range of Influence of Leading Edge in Post-Shock Flow ..................... 163

F. Calculation of Expected Uniform Flow Region on Delta Wing ............... 166

G. Calculation of the Limits Within Parameter Space For Body Definition ... 169

H. A Relation Between Lift and Volume ......................................... 172
CHAPTER I

INTRODUCTION

1.1 Historical Perspective

In the 1950’s and 60’s aerodynamicists were looking toward a future of global unification. Automobiles, trains, ships and aircraft were already beginning to bring people from around the world together, to blend the different cultures and expand trade boundaries. Hypersonic travel was thought to be a clear next step to increase the distance that humans could travel in a reasonable amount of time [41]. To this end, much time and effort was devoted to designing supersonic and hypersonic aircraft.

Without the computational facilities of today, early researchers were forced to find analytical, calculable methods for analyzing the flowfields around aircraft. For example, potential flow theory was used to describe the flow around subsonic airfoils, the method of characteristics for determining some simple (and not so simple) supersonic flowfields, and Thin Shock Layer theory for approximating strong shock conditions. In this vein Hilton and Nonweiler in 1958 developed a method of hyper-
sonic airframe design which automatically allowed the flowfield to be known [54][55]. The idea was to first choose a flow field and then define the body surface as a series of stream-surfaces from that flow. Hypersonic vehicles designed in this way were termed waveriders since they kept bow shocks attached to the leading edge of the body. This inverse design approach became the traditional method for waverider design. Examples of using this method for vehicle design can be found in Appendix A and references [41] [55] [64] and [69].

The attached shock wave is advantageous in that it isolates the upper and lower surface flows. This allows a body to be designed using inverse methods and the flow over both surfaces to be known. The attached shock is also desirable in that it maintains the high pressure flow under the wing to create a much higher lift-to-drag ratio than conventional hypersonic configurations.

Waverider body geometries are driven by the requirement of shock attachment at the leading edge. Waveriders are highly swept, smoothly varying wing-body vehicles which require sharp leading edges in order to maintain the attached shocks. They often have more anhedral than other supersonic bodies in order to increase the shock strength without creating cross-flow velocities. Their smooth surfaces are free of sudden adverse pressure gradients, making them good candidates for aero-propulsive technology. Some typical geometries are shown in Figures 1.1, 1.2, and 1.3.

Designing bodies through the inverse method made possible beautiful, simple approximations to the hypersonic performance of these vehicles - both inviscid and viscous. Through hypersonic small disturbance theory, expressions for surface pres-
Figure 1.1: Examples of Simple Waverider Shapes. (The top right body is a caret wing as described in Appendix A.)
Figure 1.2: A Wind Tunnel Model of a Waverider Designed Using Osculating Cones by Miller and Argrow [48].
Figure 1.3: Some Examples of Complicated Waverider Shapes
sure (hence lift and drag) could be calculated directly. Then, given an inviscid flowfield, body shapes could be optimized using the calculus of variations [22][40].

While computational simplicity was an original motivator for the waverider concept, simplicity of the flow field might be expected to correlate with flowfield efficiency making waveriders outstanding candidates for hypersonic flight. Thus, a new class of hypersonic vehicles developed out of this inverse design principle which today is being considered for many applications.

Classical, subsonic, wing-body aircraft are very effective at low speeds but are not well suited to supersonic flight. The swept wings used for supersonic flight are meant to minimize the strength of the shock waves. But at hypersonic speeds the wings would have to be swept to such an extent as to be impractical. Waveriders, on the other hand, exploit the shock waves by maintaining an attached shock at the leading edge and capturing the highly compressed flow entirely under the body. In this way waveriders provide a significantly higher lift coefficient for a given lift/drag ratio (or a higher L/D for a given lift coefficient) than conventional swept aircraft designs at these speeds [41].

Waveriders have now been designed out of many different types of analytical flows: wedges, cones, yawed cones, elliptic cones and axisymmetric bodies with longitudinal curvature [38][62][63]. Methods have also been developed to derive a waverider shape out of arbitrary shock shapes [45]. This adds an extra degree of freedom to the design process over the regular inverse design method.

The oil crisis of the 1970's severely limited the amount of work devoted to high
speed civil aircraft. But the burgeoning space race helped to continue some hypersonic research. Early in the space program the U.S.A. and Europe worked together to develop a space shuttle. Most European designs were based on the waverider concept; American designs were not. The U.S.A. and Europe (primarily Great Britain) parted ways in the high speed research area with the Europeans continuing to investigate waveriders in a limited capacity [79].

Recently there has been renewed interest in hypersonic vehicles and the expansion of possible waverider applications (Single-Stage-to-Orbit vehicles, missiles, artillery, gliders, aero-braking, aero-maneuvering, aerospace planes). Consequently, U.S. and European waverider research has again begun to flourish. Several American universities as well as NASA and U.S. companies regularly publish waverider results, adding to the continued work in Europe. Asian countries have also recently begun to look at waverider-type airframes for hypersonic applications.

In the past, waveriders were criticized for their thin wings, poor volume distribution, large wetted surface areas and large anhedral which leads to poor stability characteristics. There was also a lack of knowledge about how they would perform under real mission conditions. These concerns have been addressed through the use of different flow generating shapes and optimization programs. As for "real-world" data, recent studies have addressed waverider performance with the addition of control surfaces\(^1\), propulsion units\(^2\), leading edge blunting\(^3\), at off-design conditions and in real flows having viscous and real gas effects[3]. There have also been great strides

\(^{1}\)so far only in wind tunnel tests [21][57]
\(^{2}\) [15] [32][50]
\(^{3}\)via wind tunnel tests [30][21][34]
in the development of high temperature materials and cooling systems [54].

There have recently been a number of wind tunnel tests of waveriders [20][21][30][57][46][39][50][49]. These have all served to validate the waverider concept as a high lift vehicle and shown that at off-design conditions the waverider can still be a practical vehicle, even with the addition of control and propulsion systems. In fact, X. He [34] demonstrates that waveriders flown below their design Mach number often have higher $L/D$.

The XB-70 Valkyrie, a waverider-type vehicle, was built and flight tested at the end of the 1950’s for the U.S. Air Force. It was a flat-top waverider designed to have a maximum range of 7000 nautical miles at a cruise speed of Mach three. Two flight vehicles were built. The flight tests demonstrated the high L/D performance and the practical applicability of the waverider concept as a full-scale aircraft [26]. The primary test aircraft was lost in a mid-air collision. This, in addition to changing military requirements, resulted in a termination of the XB-70 program [26].

Since then only two waverider-type vehicles have been planned or built. One is LoFlyte, a NASA contracted remotely piloted vehicle built by Accurate Automation Corporation. If this vehicles is successful, NASA will build a full-scale Mach 5.5 vehicle [28]. Still in the concept stage is NASA’s Hyper-X plane which Lewis says is “based on a vehicle that was based on a vehicle that was based on a conically-derived waverider [44]”. Any practical airframe application will require nontrivial departures from the idealized waverider configuration. Center [14] makes some observations

---

4The term waverider probably didn’t come into existence until after the XB-70 was built. However, the XB-70 design was based on the principles that later became standard waverider design concepts.
about the similarities of some operational vehicles to proposed waverider designs, though none of these vehicles are claimed to be waverider inspired. See Figure 1.4.

1.2 A New Design Approach

Limiting the waverider design process to the classical inverse method is unnecessarily restrictive given the increased computational power we have today. The original design methods have served well in furthering the understanding of hypersonic flow and providing insight into optimum vehicles for this regime and as first steps toward practical vehicles. However, the design space of waveriders is limited if one is confined to analytically obtainable flow fields.

Due to the original inverse design method, the term "waverider vehicle" typically refers to those shapes originating from streamlines of a known flow field. The design approach of this work requires an expansion of that definition so that a waverider is any vehicle whose bow shock remains attached to the leading edge.

I hope to show in this work that a waverider can be designed using a more direct approach and that this method frees the designer to explore more of the design space. Any vehicle which maintains supersonic flow in the marching direction can be analyzed as a potential waverider vehicle using a space-marching code. This method is termed a direct approach since the body shape can be chosen before the flow field around it is known.

Another benefit of the direct approach is that the viscous effects can be included in the vehicle design. Molvik et. al. [50] found that in a wind tunnel test a Mach 8
Figure 1.4: Examples of Waverider-Like Designs. These Figures Originated From Reference [14].
waverider had a boundary layer on the compression surface that was $\frac{1}{3}$ as thick as the shock layer. Thus, the inclusion of viscous effects is a necessary step toward the practical design of a waverider.

The inverse-design method bases vehicle design on analytic, inviscid flow fields. Incorporation of viscous effects as well as real gas effects, aero-propulsion and control surfaces, must be done by adding these effects to the inviscid flow calculations. Viscous and real gas effects are added using analytic approximations after the inviscid flow has been computed\(^5\). A direct design approach can automatically incorporate many of these practical design issues by solving equations which more closely model the flow being simulated (e.g. The Navier-Stokes equations with real gas effects) over bodies including any necessary design modification.

While direct design does open the design space, it introduces new challenges and difficulties. An explicit space-marching scheme requires accurate initial conditions. Any errors in the initial conditions will be carried downstream to the rest of the solution. A naive shock-capturing code on a coarse grid may not recognize when the shock should be attached or detached due to the differences in length scales between the leading edges and the rest of the body. In order to capture the flow details at the leading edge and to sharply resolve the shock, the grid should be locally refined, preferably in an automatic, adaptive manner close to the leading edges. The 3-D nature of this problem means that the grid must be able to handle cell geometries

\(^5\)At the University of Maryland, Bowcutt et. al. began optimizing inversely designed waverider shapes incorporating viscous skin friction. Chang [16] furthered this work and found that for higher Mach number flows and higher altitude flights, viscous interactions such as the induced pressure due to boundary layer displacement thickness plays a significant role in drag prediction. McLaughlin [47] further extended the optimization methods to account for chemical equilibrium flow as opposed to the perfect gas assumptions previously used [47][3].
which do not remain constant in the marching direction.

1.3 This Work

As a proof of concept, a CFD code has been developed which uses an explicit space-marching scheme to solve the Euler equations to evaluate the performance of a series of waverider bodies. In addition to the unique direct design approach, this work uses a 2-D unsteady Cartesian grid adapted for use in this 3-D steady, space-marching problem so that the cross-sectional shapes of the bodies may be arbitrary and be allowed to vary along the body.

An explicit space-marching scheme requires that the step size be dependent upon the cell size. For sharp nosed bodies this would create a prohibitively small marching step size. Initial conditions are thus required at some later x-station on the body. I chose to find these initial conditions by assuming a self-similar fore-body and using a guess for the initial solution plane. I then march the equations until a self-similar solution is achieved. This self-similar solution is then the correct initial condition required for that shape. This solution can then be used as the initial condition for the simulation of any vehicle whose aft-body smoothly blends with the required fore-body. However, the generation of initial conditions became a major part of the work and therefore only self-similar body shapes are used for the entire body length. This method eliminates errors due to transferring another computed or approximate solution to the space-marching grid.

The combined use of a shock-capturing scheme and an automatically adaptive
grid permits good resolution of the body and shock. It does this without an excessive number of cells in uninteresting regions of the flow. The grid generation procedure only requires the body definition and maximum refinement levels to be specified, though the solution refinement criteria may also be modified.

The following chapters present details regarding the assumptions, numerical methods and grid used in the CFD code. A discussion of some simple analytical techniques for caret wings is also included since they facilitate a basic understanding of waverider flowfields. The CFD code is used to evaluate a sequence of simple waverider body shapes. This leads to an improved under-surface compression shape for a given waverider planform and free-stream flow.
For hypersonic cruise flight, the assumptions of steady, inviscid flow provide enough accuracy for first-cut design studies. These assumptions allow the gasdynamic governing equations to be reduced to the steady Euler equations. We will also require that the flow remains supersonic in the marching direction (see Figure 2.1). This means that the weak shock solution to the gasdynamic equations must
be assumed rather than the strong shock solution. The supersonic flow assumption makes the Euler equations hyperbolic (with respect to the $\hat{x}$ direction\(^1\)) and thus they can be solved by “marching” in that direction. In this way the flow solution is found at each new $\hat{x}$-station directly from the previous station’s data; just as the unsteady equations are updated at each new time step.

### 2.1 The Three Dimensional Steady Euler Equations

I will use the common notation of primitive variables as follows:

- $u = $ flow velocity along $\hat{x}$,
- $v = $ flow velocity along $\hat{y}$,
- $w = $ flow velocity along $\hat{z}$,
- $p = $ static flow pressure,
- $\rho = $ flow density.

The assumptions of steady, inviscid flow imply that total enthalpy is constant (even across shocks) so that there is no need to solve an energy equation. For a calorically perfect gas this constant can be expressed as

\[
h_o = \text{const} = \frac{\gamma p}{(\gamma - 1)\rho} + \frac{1}{2} q^2 = \frac{H_0}{\rho_\infty}
\]  

(2.1)

where

\[
q^2 = u^2 + v^2 + w^2
\]  

(2.1)

and $\gamma = $ ratio of specific heats.

\(^1\) vectors with a $\hat{\cdot}$ represent unit vectors in that direction.
The three-dimensional, steady Euler equations in Cartesian coordinates are

\[ F_x + G_y + H_z = 0. \]  

(2.2a)

\( F, G, H \) are fluxes in the \( \hat{x}, \hat{y} \) and \( \hat{z} \) directions

\[ F = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
\rho uw 
\end{pmatrix}, \quad G = \begin{pmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
\rho vw 
\end{pmatrix}, \quad H = \begin{pmatrix}
\rho w \\
\rho uw \\
\rho vw \\
\rho w^2 + p
\end{pmatrix}. \]

(2.2b)

The non-conservative form of the equations,

\[ AU_x + BU_y + CU_z = 0 \]

(2.2c)

will later be used to derive the numerical flux function. \( U \) is the state vector consisting of

\[ U = \begin{pmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w
\end{pmatrix}. \]

(2.2d)

and

\[ A = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-u^2 + \gamma^{-1}(h_o + \frac{1}{2}q^2) & u^2 & -v^2 & -w^2 \\
-vu & v & u & 0 \\
wv & w & 0 & u
\end{bmatrix}. \]

(2.2e)
To formulate an eigen problem, equation 2.2c can be rewritten as

$$U_x + A^{-1}B U_y + A^{-1}C U_z = 0.$$  \hspace{1cm} (2.2h)
\[ D(3, 1) = \frac{(\gamma - 1)R(-R(\gamma - 1) + 2\gamma q^2 - 2w^2)}{-2\gamma u T} \]  
(2.9i)

\[ D(3, 2) = \frac{(\gamma - 1)((R - 2w^2)(\gamma - 1) - 2u^2\gamma) - 2\gamma^2 v^2}{-T\gamma} \]  
(2.9j)

\[ D(3, 3) = \frac{v((1 - \gamma)(R - 2u^2) + 4\gamma u^2)}{-T\gamma u} \]  
(2.9k)

\[ D(3, 4) = \frac{w((1 - \gamma)(-R(\gamma - 1) + 2q^2\gamma - 2w^2))}{-\gamma u T} \]  
(2.9l)

\[ D(4, 1) = \frac{-v w (\gamma - 1) R}{\gamma u T} \]  
(2.9m)

\[ D(4, 2) = \frac{-2vw (1 - 2\gamma)}{\gamma T} \]  
(2.9n)

\[ D(4, 3) = \frac{-2w(- (\gamma - 1)v^2 + \gamma u^2)}{\gamma u T} \]  
(2.9o)

\[ D(4, 4) = \frac{v(-R(\gamma - 1) + 2\gamma q^2 + 2w^2(1 - 2\gamma) - 2\gamma v^2)}{-\gamma u T}, \]  
(2.9p)

where

\[ R = 2h_o + q^2, \]  
(2.9j)

\[ T = R(\gamma - 1) - 2u^2. \]  
(2.9k)

Derivation of \( D \) and its eigenvalues and eigenvectors is much easier in a parametric form. Appendix B includes the necessary matrices to carry this out.

\( D \) has real eigenvalues \( \lambda^k \) and right eigenvectors, \( r^k \), which satisfy the relation

\[ \Delta G = D \Delta U = \sum_k \lambda^k \alpha^k r^k; \]  
(2.9l)

where the \( \alpha^k \) are the wave strengths. The right eigenvectors are

\[ r^1 = \begin{pmatrix} 1 \\ u + \frac{v}{\beta} \\ v - \frac{u}{\beta} \end{pmatrix}, \quad r^2 = \begin{pmatrix} 1 + \frac{q^2}{2h_o} \\ 2u \\ 2v \end{pmatrix}, \quad r^3 = \begin{pmatrix} w \\ 0 \\ 2h_o \end{pmatrix}, \quad r^4 = \begin{pmatrix} 1 \\ u - \frac{v}{\beta} \\ v + \frac{u}{\beta} \end{pmatrix}. \]  
(2.9m)
The corresponding eigenvalues are

\[ \lambda_1 = \frac{v - \frac{u}{\beta}}{u + \frac{v}{\beta}} \]  
(2.14a)

\[ \lambda_2 = \frac{v}{u} \]  
(2.14b)

\[ \lambda_3 = \frac{v}{u} \]  
(2.14c)

\[ \lambda_4 = \frac{v + \frac{u}{\beta}}{u - \frac{v}{\beta}} \]  
(2.14d)

with

\[ \beta = \sqrt{\frac{u^2 + v^2}{a^2} - 1} \]  
(2.15)

\[ a = \frac{\gamma P}{\rho} \]  
(2.16)

The integral form of the equations is needed since we are looking for discontinuous solutions to these equations:

\[ \oint_{\Omega} F \cdot dA = 0. \]  
(2.17)

\( F \) is the tensor of fluxes

\[ F = \{ F, G, H \}, \]  
(2.18)

and \( dA \) is the differential area vector. \( dA \) can be rewritten as the unit normal vector times the differential area, \( \hat{n} dS \), so that

\[ dA = dA_x \hat{x} + dA_y \hat{y} + dA_z \hat{z} \]  
(2.19)

with

\[ dA_x = dA \cdot \hat{x} \]

\[ dA_y = dA \cdot \hat{y} \]

\[ dA_z = dA \cdot \hat{z}. \]
Using these equations, 2.17 can be written as

$$ \oint_{\Omega} (F dA_x + G dA_y + H dA_z) = 0. $$  \hspace{1cm} (2.20)

### 2.2 Discretization

To numerically solve the governing equations, 2.17, they must first be discretized. To do this a finite volume method is used where the state vectors are referenced to the cell centers. The spatial domain is discretized using 3-D cells. The subscripts $i, j, k$ will represent this discretization in the $\hat{x}, \hat{y}$ and $\hat{z}$ directions respectively.

The discretized form of equations 2.17 is

$$ \sum_{l=1}^{\text{no. faces}} \mathcal{F}_l \cdot A_l = 0. $$  \hspace{1cm} (2.21)

Where $A_l$ is the area vector for the $l^{th}$ face of a cell and $\mathcal{F}_l$ is the flux through the face. For a square cell, as in Figure 2.2, equation 2.21 becomes

$$ F_{i+1} A_x^{i+1} - F_i A_x^i + G_{j+\frac{1}{2}} A_y^{j+\frac{1}{2}} - G_{j-\frac{1}{2}} A_y^{j-\frac{1}{2}} + H_{k+\frac{1}{2}} A_z^{k+\frac{1}{2}} - H_{k-\frac{1}{2}} A_z^{k-\frac{1}{2}} = 0 $$  \hspace{1cm} (2.22)

and is the discrete form of 2.20. Since the flow is always supersonic in the $\hat{x}$ direction we can choose the $\hat{x}$ direction flux vector at the latest $\hat{x} = \text{constant}$ plane, $F_{i+1}$, as
the vector of unknowns. The flux at each new cross-plane is then found from

\[
F_{i+1} = F_i A_x^i - \sum_{j,k \in \text{faces}} \left( \frac{F_i}{A_x^i} \right) \cdot A_l
\]

(2.23)

The primitive variables can be found from the flux vector as discussed in the next section. And the grid on which these computations will be carried out will be discussed in the next chapter.

### 2.3 Calculation of Primitive Variables

There are two issues regarding the calculation of the primitive variables (defined in Section 2.1) from the flux vector. One, there can be only four independent quantities for these equations (2.2a and 2.2b), yet we are using 5 primitive variables: \( p, \rho, u, v \) and \( w \). \( p \) was chosen to be the dependent variable and its value is calculated from 2.1. Second, computing the primitive variables from the flux function (2.2b) is a non-unique process - a quadratic is obtained for the density. The root which results in supersonic flow is chosen. The components of the flux vector \( F \) are given in 2.2b. Using these we can solve for density and the velocity components from the following equations.

\[
\rho = \frac{F(2) - \sqrt{F(2)^2 - 4^{\frac{\gamma-1}{\gamma}} (h_o - \frac{F(3)^2 + F(4)^2}{2F(1)^2}) (1 - \frac{\gamma-1}{2\gamma}) F(1)^2}}{2^{\frac{\gamma-1}{\gamma}} (h_o - \frac{F(3)^2 + F(4)^2}{2F(1)^2})}
\]

(2.24)

\[
U = \frac{F(1)}{\rho}
\]

(2.25)

\[
V = \frac{F(3)}{\rho U}
\]

(2.26)

\[
W = \frac{F(4)}{\rho U}
\]

(2.27)
CHAPTER III

COMPUTATIONAL GRID

When simulating waverider flow it is important to capture the solution details at the leading edges and to sharply resolve the shock. In order not to over-refine the rest of the flow, it is necessary to use local grid refinement. These local refinement zones must be moved in order to follow the leading edge and shock movements between body cross-sections. A 2-D unsteady adaptive grid generation technique developed by Bayyuk, Powell, and Van Leer [7] [8] possessed the capabilities I needed. The code written by Bayyuk to implement and test these techniques had the following capabilities: automatic, geometric-adaptive grid generation about arbitrary bodies; automatic, solution-adaptive grid refinement for shock capturing; and the ability to handle body and shock motion. By using the analogy between 3-D steady flow and 2-D unsteady flow we were able to adapt this grid generation procedure for use with my space-marching scheme.

The disadvantage of the Cartesian method is that more geometry errors are generated due to the way the grid approximates the body. These errors would not exist if a body-fitted grid were used. However, the automatic refinement capabilities would
have to be added to such a method.

If a 3-D body being simulated is viewed as a composite of stacked, computational cross-planes in the marching direction, as shown in Figure 3.1, 2-D cross-sections of the body appear to move and/or grow on the grid as each plane is reached in the marching procedure. This is analogous to marching procedures in the solution of 2-D unsteady problems.

![Figure 3.1: Stacking Computational Planes to Form a Body in Three Dimensions](image)

The grid generation techniques used are discussed in detail in Bayyuk’s PhD thesis [8] and so here the discussion will focus on the differences and difficulties encountered in adapting this code for 3-D, steady, space marching over sharp leading edge bodies.

### 3.1 Overview of Code Algorithm

Bayyuk’s grid generation code is a Quadtree-based adaptive Cartesian grid method. To form 3-D cells out of the 2-D cells in each cross-plane the 2-D cells in consecutive
cross-planes are made to form the fronts and backs of the 3-D cells. By connecting the front and back faces of a cell, its sides are formed. In this method the grid remains stationary and the body is “cut” out of the grid as seen in Figure 3.1. This results in a grid having only two basic types of cells, square cells and cut-cells which appear only along the body wall. The cut-cells result from the body intersecting the squares in the grid. Some examples of cut-cell shapes are shown in Figure 3.2. The cut-cells must be treated separately due to their shape and because one of their faces (the body wall face) will require imposition of a solid wall boundary condition.

Bayyuk allows each body cross-section to be input as a function or set of points. In the latter case, splines are fit through the points in each computational plane. The mesh is refined according to body geometry by dividing cells into four children until a user specified maximum refinement level is reached. The grid is also refined based on the current flow solution and the specified gasdynamic refinement criteria, as discussed later. The refinement criteria are responsible for ensuring a good distribution of cells over the allowable number of refinement levels. A maximum gasdynamic refinement level is specified such that the code is not allowed to refine, based on the

Figure 3.2: Examples of How Cells are Cut by the Body to Produce Cut-Cells
solution, beyond a given level. An example of a grid plane which was automatically refined to resolve the body geometry and gasdynamic features of a solution is shown for a caret wing cross-section (figure 1.1) in figure 3.3. The sharp leading edges of the body are well resolved and the regions of the planar shock and its attachment at the leading edges were automatically resolved.

Cells cut by the body can be arbitrarily small and so are merged or grouped in order to form reasonably sized computational cells (more on this later). This process intimately links the gasdynamic and geometric parts of the simulation. Merging cells requires the state vectors to be combined in a conservative manner. After this
is completed the flux vectors through the interfaces can be computed and the residual summed for all the cells in a computational group. The group flux vectors are then updated via equation 2.23. The state vector is computed from the flux vector and then redistributed to the elements in the group as explained in section 3.6. The grid is then unmerged. This completes one computational step. In Bayyuk's code, points defining the body cross-section are allowed to shift (within certain limits) with respect to the grid from one computational plane to the next.\(^1\) This allows for flexible airframe design. Appendix B discusses how the points on successive cross-planes are found once the body is defined. The grid is then refined to the new geometry and then further refined according to the solution just computed, cut-cells are merged and the flow solution on the next plane computed. The process continues until the desired station is reached.

### 3.2 Adaptation of the Unsteady Grid Method to 3-D Steady Space Marching

There are two primary differences between a 2-D unsteady code and a 3-D space marching code. The first is the formulation of the governing equations. In unsteady computations the unknown variable is the state vector. In space marching the flux vector in the marching direction is the unknown and the state vector is determined later from this flux vector. Since the flux vector is the unknown it must be stored. It could replace the state vector in the 2-D version, but the rest of the code is

\(^1\)When computing unsteady flow it may not be customary to talk of the different solutions as lying on a succession of computational planes. However, when space marching it is.
very memory efficient and the state variables are used often for refinement criteria, merging and unmerging cells, etc. so the state vector is also stored.

The second major difference is the inclusion of the third spatial dimension. The flux through the faces needs to be multiplied by a face area rather than a face length. The inclusion of the cell faces between computational planes was the biggest and most difficult change to the unsteady code in terms of maintaining efficiency and accuracy. These computations are discussed in Section 4.4. The third velocity component also needed to be added to the state, flux and gradient vectors. However, in terms of memory storage, this component simply replaced the energy variable used in the unsteady problem.

3.3 Geometric Refinement

A representation of the true body on the grid is accomplished by splining the body points in each cross-section. The computational body is created by connecting the points where the splined body intersects the grid. Splines don’t exactly represent flat surfaces especially when they must connect smoothly with a curve such as a leading edge and so small oscillations in flat surfaces may appear. The magnitude of the body position errors is on the order of $10^{-7}$. In addition, curved surfaces are represented by approximations. For example, a cone at one cross-section will be represented by a polygon. In fact, it will be represented by a different polygon at subsequent cross-sections since at each cross-section the splined body is approximated by where it crosses grid lines. Thus, the amount of the cone’s cross-section that is lost in one cell
by the linear approximation will not be the same amount after the body position is advanced on the grid. The shape of the body between cross-sections is also a linear, faceted approximation. The body between planes is represented by connecting the grid lines of the bounding cross-planes. The true curvature of the body is lost between cross-planes. Errors made in the cross-sections are passed along to the formation of the cell’s side faces. Errors in the the side faces change according to how much of the body was truncated in the cross-sectional approximation. Changes in the representation of the body lead to oscillations in the solution on the body’s surface. These oscillations don’t exist in the free-stream flow but increase as the free-stream Mach number increases. They can be decreased by increasing the number of cells on the body. The magnitude of the oscillations is less than 2.0% of the surrounding solution. This is evident in the solutions presented in chapters V and VI.

To obtain perfectly attached shocks the leading edges must be represented as perfectly sharp. This can be accomplished with either a body-fitted grid or sub-cell resolution on non-body fitted grids. While not perfectly sharp, the geometric adaptation performed by this grid code makes the leading edges for all intents and purposes sharp even without using sub-cell resolution. But the current grid code cannot merge cells (form computational groups) around sharp corners. The merging algorithm requires that the body everywhere be at least two cells wide. This means that the leading edge radius of curvature must be greater than two cell widths. Therefore, the leading edges were blunted slightly. Unless otherwise specified, the blunted leading edges have a radius which is .75% of the span in the initial cross-
section. Moreover, the bluntness, as a percentage of the local span, decreases as the cross-section of the body grows since the leading edge radius is held constant at its original value. The effect of blunted leading edges on the flow solution is discussed in 5.5.1.

It is necessary for a good representation of shock attachment to maintain relatively sharp leading edges. But without sub-cell resolution some errors at the wing tips are expected. These errors are caused by the finite resolution of the tip. Figure 3.4 shows how the grid would resolve the tip at two successive computational planes. The amount of area truncated from the tip varies from plane to plane. This oscillation of the position of the tip, especially in severe flight conditions, causes very large errors at the tips. Blunting the leading edges helps reduce these errors as does increasing the body refinement level. However, each grid refinement level increase slows the code by a factor close to four\(^2\). The errors generated at the tip stay confined to the tip region and do not appear to affect the rest of the flow. An example of the tip errors and their limited range of influence will be given in section

\(^2\)2-8 in principle [10]
The extra cells on the body required to resolve the sharp leading edges is a limiting factor in the speed of the computation. The grid methods can be improved to better resolve the sharp leading edges and to allow multiple refinement levels on the body [9]. This would drastically reduce the number of cells in the grid for bodies with primarily flat surfaces.

### 3.4 Gasdynamic Refinement

In each computational plane the grid is also refined according the solution just computed on the previous plane. This refinement is done automatically according to specified refinement criteria. Using results of De Zeeuw [24] and Paillére [56], the criteria chosen for this work were the density gradient, the divergence and the curl of velocity. A waverider solution is expected to have only attached shocks. But the truncated leading edges induce expansions on the upper surface. In some solutions embedded secondary shocks exist under the wing. These embedded shocks interact with the leading edge shock to produce contact surfaces. All these features are well captured by the chosen refinement criteria.

Cells were flagged for refinement if any of the above quantities in that cell were greater than a specified fraction of the distance between the chosen maximum and minimum threshold values. The threshold value is adjusted independently for each refinement level. This affords great flexibility in the distribution of cells of different sizes. How often to check and whether to refine or unrefine the grid is also specifiable.
The solution can be non-smooth over grid areas that have been recently refined or coarsened. But smoothness returns after a few iterations.

3.5 Merging

The apparent body motion between successive computational planes and consequent cutting of the grid cells can create arbitrarily small cut-cells. Since the CFL condition (see Section 4.2) must be satisfied for every computational cell and the spatial step size must be computed globally rather than locally, these small cut-cells could dramatically slow the simulation. The motion of the body across grid lines and the emergence and disappearance of cells from the computational domain, see figure 3.5, also causes difficulties. These are discussed in section 4.4. These problems are eliminated by merging the small cut-cells to form a “group” of cells which does not

![Figure 3.5: Small Cut-Cells Disappear From the Computational Grid as the Body Cross-Section Changes Position Relative to the Grid Between the i and i + 1 Plane.](image)
disappear from the computational grid between two successive computational planes. An example is shown in Figure 3.6. The size of these groups can be restricted to

![Diagram showing computational groups]

Figure 3.6: Cut-Cells are Merged into Computational Groups so That The Computational Group Exists in Both the $i$ and $i + 1$ Planes.

be any multiple of the size of the smallest uncut-cell (but usually between 0.5 and 2.0). Due to this merging, the computational step size is restricted by the size of the smallest uncut-cell rather than the smallest cut-cell size.

Cut-cells are merged preferentially in a direction perpendicular to the body to preserve accuracy [8]. They are merged with neighboring cut and uncut-cells until the group’s area falls within a specifiable range. The upper bound on the group area preserves body resolution and accuracy of the computation at the level of the smallest uncut-cell size and the lower bound maintains a reasonable marching step size.

There is no noticeable difference in the solution (except at the sharp leading edges) when the group’s area thresholds are changed. This is in agreement with [7] where Bayyuk, Powell and van Leer stated that the merging and unmerging of cells
can be proven not to lower the order of accuracy or the resolution of the body [7] 3.

Since the body geometry is resolved before the merging is done, the area of the merging groups does not result in more geometry errors at sharp leading edges than already exists due to leading edge truncation (discussed in 3.3). But merging the cells around the corners causes some diffusion of the solution since a computational cell can have only one state vector. Thus, the larger the groups at the leading edge the more the upper and lower surface flows seem to interact. Due to the high resolution of the body, the amount of diffusion is very small - much smaller than would be achieved using a practical, uniformly refined grid.

The current merging algorithm restricts the apparent body motion (cross-sectional body displacements between planes) to not more than the width of the smallest uncut-cell between two successive cross-planes. This is a restriction that can be lifted if the merging algorithm is upgraded. In practice, I found that the merging would often not allow the body to move more than 0.2 times the smallest uncut-cell width since I was often close to the radius of curvature restriction mentioned above.

The merging algorithm tries to merge cells perpendicular to the surface of the body to maintain accuracy. In sharp convex corners, this may result in computational groups with a large number of cells [10]. The grid generation code is not robust for this type of geometry [9].

---

3The magnitude of the error is increased but the order of accuracy is maintained.
3.6 Gasdynamics for the Computational Groups

In order to maintain conservation, the group state vector is computed by area-weighting the state vector of the group’s elements. For example, a group with \( k = 3 \) elements is assigned the density found from

\[
\rho_{\text{grp}} = \frac{\sum_{k=1}^{3} \rho_k A_{x_k}}{\sum_{k=1}^{3} A_{x_k}},
\]

(3.1)

where \( A_x \) is the cell face area in the current computational plane. Redistribution of the group state vector to elements is done by assigning the group’s conserved quantities to the elements. For density this looks like

\[
\rho_k = \rho_{\text{group}}
\]

(3.2)

where \( k \) is the number of elements in the group. Bayyuk has chosen to over-resolve the body such that when cells are merged into groups, the merged groups comprise the desired resolution level. In this way the above assignment of the group quantities to its elements is not a loss of resolution. A gradient extrapolation from the group centroidal value to the centroids of the elements can be done conservatively [59] and could be implemented to improve code efficiency by not requiring as many cells for the same effective resolution. However since the driving force in body refinement here is the leading edges, until multiple body refinement levels are allowed, this would only add more computations and unneeded increases in resolution.

\(^{4}\) Only \( \rho, \rho U, \rho V, \rho W \) are constructed, \( P \) is computed from these and \( H_o \) as discussed in 2.3.
3.7 Summary

The advantages of the adaptive mesh code are, for the most part, obvious. The arbitrarily high resolution of body geometry and automatic, solution-adaptive capabilities make the grid efficient and preferable to uniform grids especially for design type problems where efficiency is very important.

Most of the difficulties in using and adapting Bayyuk's code for steady space-marching have already been discussed: resolution and truncation of the leading edges and consequent over-refinement of flat body surfaces; merging on highly concave bodies; body oscillations due to the linear approximation of body curvature and splination of flat surfaces. The high refinement allowed by the grid also contributes to another problem, odd-even decoupling of the pressure and density fields in strong shocks for some flow solvers. The mechanisms of the breakdown are discussed in [61].

Because the boundary layer is a large percentage of the shock layer for hypersonic flow, there does not need to be as much concern about the shapes of cells and efficiency of the grid in the boundary layer. If there was a need for grid modifications near the body the Cartesian grid could be stretched. Another idea is to use a body fitted grid in the vicinity of the body and then pass the boundary of this domain into the Cartesian grid generator.
CHAPTER IV

METHOD OF FLOW SOLUTION

The steady flow solution is computed using a first-order Godunov method and a MUSCL type extension to achieve second-order accuracy. Accuracy is also enhanced through the use of the adaptive Cartesian mesh developed by Bayyuk [8].

I used two different Godunov-type methods, Flux Difference Splitting (FDS) [67] and HLLE (Harten, Lax, Van Leer, Einfeldt) [33], [27] to compute the fluxes $G$ and $H$ in equation 2.22 at the cell interfaces in cross-sectional planes. Both FDS and HLLE provide good shock capturing. HLLE smears contacts but these were not usually of importance in this work. HLLE was useful in extreme flow situations when more dissipation was required. HLLE was also useful in situations where strong shocks aligned with the grid. In this situation, FDS exhibits odd-even decoupling of the pressure and density fields as discussed in [61].
4.1 Calculation of the Interface Fluxes

Godunov’s method of evolving the equations of motion uses a piecewise, spatially averaged approximation to the solution in a series of cells, and solves a Riemann problem at the cell interface to obtain the solution at the next time/space level. For two- and three-dimensional problems a one-dimensional physical problem is assumed to exist in each direction and then super-imposed. This allows the interface flux computations for each direction, \( \dot{y} \) and \( \dot{z} \), to be done independently. Experience shows that codes which employ this technique are robust and give good results for shock problems. This method does however smear shear and entropy layers \([73][35]\), but these are not prominent features of waverider flow.

Once derivation of the numerical flux function is completed in one direction, say \( \dot{y} \), the flux in the \( \dot{z} \) direction is computed using the same algorithm with the flow rotated to a local face-coordinate system. Once the flux is computed in this local system it can be rotated back to the body-fixed coordinate system and used to update \( \mathbf{F} \). This procedure allows the flow solver code to be used for flow across a face orientated in an arbitrary direction.

The remainder of this section will discuss the computation of the interface fluxes through one face of a non-boundary cell.

4.1.1 Two-Dimensional Numerical Flux Computation

Let \( f \) be the flux solution at iteration levels \( i \) and \( i + 1 \) which yields the solution state vector \( \mathbf{v} \). Let \( g_{j+\frac{1}{2}} \) be the numerical interface flux and \( \tau \) and \( \Delta \) be the marching
direction discretization and discretization in the cross-plane respectively. The update equation 2.23 for a uniform grid can be rewritten in two-dimensions as

\[ f_j^{i+1} = f_j^i - \tau \left( g_{j+\frac{1}{2}}^i - g_{j-\frac{1}{2}}^i \right), \] (4.1)

with \( j \) being the current cell. Solving the Riemann problem exactly to obtain \( g \) involves iterative procedures which are computationally intensive. Also, the exact solution is unnecessary since approximations are already being made in averaging the states.

Both HLLE (and its variants) and FDS solve the Riemann problem approximately through a special choice of the numerical flux. The basic nature of these choices is the same upwind principle: Central differencing plus an upwind term, \( Q \), which acts as the needed artificial dissipation. At marching level \( i \), the numerical interface flux is found via

\[ g_{j+\frac{1}{2}} = \frac{1}{2} [g(v_j) + g(v_{j+1}) - Q_{j+\frac{1}{2}}(f_{j+1} - f_j)]. \] (4.2)

HLLE and FDS choose different \( Q \) matrices.

### 4.1.2 HLLE Approximate Riemann Solver

HLLE uses the dissipation matrix:

\[ Q_{j+\frac{1}{2}} = \left[ \begin{array}{c} b_r^{j+\frac{1}{2}} + b_l^{j+\frac{1}{2}} \\ b_r^{j+\frac{1}{2}} - b_l^{j+\frac{1}{2}} \end{array} \right] D_{j+\frac{1}{2}} - 2 \left[ \begin{array}{c} \frac{b_r^{j+\frac{1}{2}}}{b_r^{j+\frac{1}{2}} - b_l^{j+\frac{1}{2}}} I \\ \frac{b_r^{j+\frac{1}{2}}}{b_r^{j+\frac{1}{2}} - b_l^{j+\frac{1}{2}}} I \end{array} \right], \] (4.3)

where \( b_r \) and \( b_l \) are the numerical wave speeds. The least amount of dissipation is provided if \( b_r \) and \( b_l \) equal the maximum and minimum eigenvalues, \( \lambda^4, \lambda^1 \), of the linearized \( D_{j+\frac{1}{2}} \) matrix, given in Section 2.1. Following [27] the wavespeeds are
modified as
\[ b^+_j = \max \left\{ \lambda^4_{j+\frac{1}{2}}, \lambda^4_{j+1} \right\} \] (4.4)
\[ b^-_j = \min \left\{ \lambda^1_{j+\frac{1}{2}}, \lambda^1_j \right\} \] (4.5)
to avoid admitting non-physical discontinuities.

The matrix \( D_{j+\frac{1}{2}} \) and \( \lambda_{j+\frac{1}{2}} \) are approximated using Roe averaged [67] values for the primitive variables defined as
\[
\tilde{\rho} = \sqrt{P_j \rho_{j+1}}
\] (4.6a)
\[
\text{var} = \frac{(\text{var} \sqrt{\tilde{\rho}})_{j+1} + (\text{var} \sqrt{\tilde{\rho}})_j}{\sqrt{P_{j+1}} + \sqrt{\tilde{\rho}}}, \text{var} = u, v, w.
\] (4.6b)

From Quirk [61], the numerical flux function 4.2 and the dissipation matrix 4.3 can be combined to yield
\[
g^{n}_{i+\frac{1}{2}} = \frac{b^+ g^{n}_{i+1} - b^- g^{n}_i}{b^+ - b^-} - \frac{2b^+ b^-}{b^+ - b^- (f^{n}_{i+1} - f^{n}_i)},
\] (4.6c)
where
\[
b^+ = \max \left\{ \lambda^4_{j+\frac{1}{2}}, \lambda^4_{j+1}, 0 \right\}
\] (4.4a)
\[
b^- = \min \left\{ \lambda^1_{j+\frac{1}{2}}, \lambda^1_j, 0 \right\}.
\] (4.4b)
This is what I actually implemented to reduce the opportunity for errors in the coding of \( D_{j+\frac{1}{2}} \).

4.1.3 Roe’s Approximate Riemann Solver

The upwind term for Roe’s Flux Difference Splitting approximate Riemann solver is
\[
Q_{j+\frac{1}{2}}(f_{j+1} - f_j) = \sum_{k=1}^{\text{num. waves}} |\alpha^{k}_{j+\frac{1}{2}}| r^{k}_{j+\frac{1}{2}} \lambda^{k}_{j+\frac{1}{2}}
\] (4.5)
If we use 4.5 in 4.2 the numerical flux computation is written and implemented as

$$g_{j+\frac{1}{2}} = \frac{1}{2} (g_{j+1} + g_j) - \frac{1}{2} \sum_{k=1}^{\text{num. waves}} r^k \alpha^k \chi^k. \quad (4.6)$$

\(\alpha^k\) are the wave strengths associated with each wave and are found from

$$(f_{j+1} - f_j) = \sum_{k=1}^{\text{num. waves}} \alpha^k r^k. \quad (4.7)$$

Roe showed in [67] that it is possible to write analytic expressions for the \(\alpha^k\), but my choice of eigenvector normalizations didn’t yield simple expressions. After using Gaussian elimination I was still left with unwieldy expressions and left it to the computer to solve these.

### 4.2 Stability

To ensure convergence of the numerical method it is necessary to satisfy the CFL stability condition. The essence of this, for all numerical methods, is that the numerical domain of dependence of the method must include the domain of dependence of the partial differential equations being modeled. For linear equations this is easily implemented using the straight line characteristics of the problem. For this non-linear problem we will assume our mesh is fine enough so that the characteristics may be approximated as linear, as in Figure 4.1, and then use a safety factor, \(\nu\), to account for the non-linearity.

In space-marching the interior of the Mach cone defines the (linear) physical domain of dependence. Our task is to compute the depth of the cells in the marching direction such that we encompass this region in our computations. Let us look at
how this is done for one direction, say, the y-direction. Figure 4.1 shows the planes that would be tangent to the Mach cones generated by a disturbance at one edge of a cell. When this is projected into the $x$-$y$-plane we see that the axis of the Mach cone is along the streamline and that the projection of the Mach cone into the $x$-$y$ plane represents the slowest and fastest wave speeds when the computation is constrained to this plane. To ensure the numerical domain of the problem includes the domain of the grid or cell we must satisfy

$$\Delta x \leq \frac{\Delta y}{|\lambda_{max}|}$$  \hspace{1cm} (4.8)

for all cells. To ensure this in practice

$$\Delta x \leq \nu \frac{\Delta y}{|\lambda_{max}|}$$  \hspace{1cm} (4.9)
is used, where $\nu$ is called the CFL number and is some positive number less than or equal to one. The state of the cell at the last computed cross-plane is used to calculate the wave speeds.

For this cell-centered code it is reasonable to calculate and enforce the above condition only in the center of the faces, $x_{i\pm \frac{1}{2}}$, rather than over the entire cell width.

The above criteria must also be satisfied in the third dimension ($z$). To include the $z$-direction we use the smallest spatial step that results from the application of (4.8) in both directions over all cells in the last updated plane,

$$\Delta x \leq \min \left\{ \nu \frac{\Delta y}{|\lambda_y|}, \nu \frac{\Delta z}{|\lambda_z|} \right\}.$$  \hspace{1cm} (4.10)

### 4.2.1 Implementation of CFL Condition

Computation of the maximum spatial step satisfying the CFL (gasdynamic) and geometric (grid generation) stability conditions is limited by the smallest cells in the grid. Basing the spatial step on the size of the small cut cells would severely slow the program. Since the flux is calculated on groups of cells instead of for each cell separately, the $\Delta y$ and $\Delta z$ above are either the cell dimensions, if the cell is uncut, or the dimensions of the computational group (on the $i$-plane $^1$) if the cell is the member of a group.

$^1$Formally, this should be at $i + \frac{1}{2}$ but this approximation saves computational effort and the CFL number is only approximate anyway.
4.3 Boundary Conditions

4.3.1 Outer Boundary Conditions

Since this is a hyperbolic problem the distance of the outer boundary from the body need not be large. The largest distance needed is the size of a Mach cone emanating from the nose. Aside from the initial condition generation, there should be no spurious errors of any significant size introduced that could reflect off the outer boundary. If the solution in the outermost cells is set to match the solution in their neighboring cells an outflow boundary condition is created and helps make the outer boundary as unobtrusive on the simulation as possible.

4.3.2 Wall Boundary Conditions

The assumption of inviscid flow implies a slip boundary condition on the body. This can be implemented using a ghost cell interior to the body which is assigned the same density, pressure and tangential velocity but oppositely signed normal velocity as its neighbor. But since the resultant flux is simply the normal momentum lost to the wall, the flux can be directly implemented with less diffusion as \( P \cdot dA \). The flux through the wall is therefore computed as

\[
\mathcal{F}_{\text{wall}} \cdot A_{\text{wall}} = \begin{pmatrix}
0 \\
P_{\text{wall}}A_{\text{wall},x} \\
P_{\text{wall}}A_{\text{wall},y} \\
P_{\text{wall}}A_{\text{wall},z}
\end{pmatrix} \quad (4.11)
\]
where $P_{\text{wall}}$ is the pressure at the wall, discussed below. The projected areas, $A_{\text{wall},x,y,z}$, of the wall-face are shown in Figure 4.2. Each projected area is a component of $A_{\text{wall}}$, the wall normal area vector.

On this type of Cartesian grid, where the body is “cut” out of the grid (see Chapter III), there ends up being small, irregularly-cut cells next to the body. These small cut-cells are merged with neighbor cells to create larger cells that are more reasonable for computations. This is discussed in more detail in Chapter III. When a cell is part of a group the state vector for that cell is either the group’s centroid value (in a first-order computation) or the group’s centroid value extrapolated to the desired point using the group’s gradient value (in a second-order computation).

Referring to Figure 4.3, $P_{\text{wall}}$, the pressure at the midpoint of the wall face, is simply the pressure at the center of the cut cell’s group in the $i$-plane (we are updating the $(i+1)$-plane) for first-order or the gradient-extrapolated value of the pressure for second-order.

The tricky part of the wall flux computation is calculation of the wall face areas.
4.4 Face Area Calculation

Let us keep in mind that the grid’s representation of the actual body is only a faceted approximation. Any curvature of the body between grid lines is lost both in the cross-section and between cross-sections, refer to Section 3.3. But the planar edges of cells makes area and length computations fairly easy. For the cells in the grid which are not cut, the calculation of the front, back and side areas \(^2\) is simple. The front and back areas are computed by squaring the length of one edge. Side face areas are calculated from the product of the face edge length in the \(i\)-plane and the spatial step size, \(\Delta x\). When cells are intersected by the body they are probably merged with other cells. This brings many interesting challenges when adapting a 2-D unsteady grid code for 3-D work.

\(^2\)Front and back refer to the \(i\) and \(i+1\) planes, respectively. Side planes refer to all planes of a cell that connect the \(i\) and \(i+1\) planes.
4.4.1 Intersected Cell Face Area Calculations

Unlike first-order, 2-D unsteady flow, which only requires information from the last computed solution plane. Even for first order, 3-D steady computation requires geometric information from the plane currently being updated in order to correctly compute the cell face areas. The body position changes between the two computational planes. This means that cell edges and cells appear or disappear\(^3\) between planes. In Figure 4.4, cell face ABC in the \(i\) plane is consumed by the body in the \(i+1\) plane. Thus ABC disappears from the gasdynamic computations. To avoid this problem cut cells are "merged" into topologically invariant groups, as discussed in 3.5. These merged groups always exist in the \(i\) and \(i+1\) planes and so can act as faces for the computational cells. When the cells are merged, the front and back faces of the new computational group will no longer be made up of squares (see Figure 4.4) and thus the face’s area cannot be computed as simply as above.

---

\(^3\)By appearing and disappearing, I mean that they move interior to the body and thus disappear from the gasdynamic computations.

Figure 4.4: An Example of A Disappearing Cell and Cell Edge Between Cross-Planes.
4.4.1.1 Front and Back Faces

The position of the body and its intersection with grid lines in each cross-plane is known exactly. Therefore, given the straight line approximation to the body’s surface the front and back cell’s face areas of the merged cell groups can be computed exactly. These areas are found by summing the face areas of the member cell’s faces. For example, in Figure 4.4 the computational cell’s front area would be the sum of the area of triangle ABC and the area of the uncut square cell below it.

4.4.1.2 Side Faces

![Figure 4.5: A Side View of a Computational Group - The Actual Geometry (AB-DEFx) and the Approximation (ABDEFG) Used. ABDEFx′ is a Computational Group composed of Cells ABx and x′BDEF.](image)

The computational group geometry is a transient quantity which varies each iterations. As such the actual geometry computations are performed on a cell by cell basis even though the gasdynamics are computed for the merged computational groups. Because of this and without sub-cell resolution the side areas of cut cells
cannot be exactly computed. In figure 4.5 the body wall moves below grid line $BG$ within the computational group. The point of crossing will be labeled $x'$. Without sub-cell resolution $x'$ is not known. Therefore, cell face $ABx'$ is approximated as $ABG$ and cell $x'BDEF$ is approximated as $BDEF$. Thus, the area of $ABx'$ is over estimated and $x'BDEF$ is under estimated. Since the actual gasdynamic computation is performed on the computational group these errors almost cancel out when added to the computational group. The needed quantity is the contribution to the residual from each cell face, flux times face area, $G_k A_{y_k}^k$. So, the residual contribution for the group’s cell face shown in figure 4.5 is obtained by the following method:

$$
(A_y G)_{\text{group}} = \frac{ABG_1 + 0G_1}{2} \Delta X + \frac{BDG_2 + EFG_2}{2} \Delta X. \quad (4.12)
$$

This quantity is then added to the computational group’s residual. This method would be exact for cut cells if the state vectors in the cells were the same. This method is exact for uncut cells. The geometric errors generated for cut cells are so small that the $\sum \overline{\nu} < 10^{-12}$ for each computational group.

The above procedure is a by-product of the 2-D unsteady method, where geometric information is only needed from the current cross-plane. In that case only one face edge length computation is necessary for each cell’s side residual to be computed. For 3-D space marching it is necessary to compute two face edge lengths (in both the $i$ and $i + 1$ planes) so that the cell side face area can be computed. The current code requires two flux computations (when one would suffice) to be computed in order to obtain these lengths. This merely makes the code inefficient and can be changed.

---

4The flux through the cell side faces is $G$. And $G_k$ is the flux through the $k^{th}$ cell face.
4.4.1.3 Boundary Wall Faces

For wall faces the area vector has three components which can be computed separately. Again, the lengths have to be calculated separately on the $i$ and $i+1$ planes and then averaged before the area is computed. In Figure 4.2 $L_y$ and $L_z$ are the lengths of the cell’s edges in the cross-plane. On each plane $A_{wall_y}$ and $A_{wall_z}$ can be calculated as

$$A_{wall_y} = L_z \Delta x,$$  \hfill (4.13)

$$A_{wall_z} = L_y \Delta x.$$  \hfill (4.14)

$A_{wall_x}$ is simply

$$A_{wall_x} = |A_{xi+1}| - |A_{xi}|,$$  \hfill (4.15)

the area of the front cell face minus the area of the back cell face. The areas on the $i$ and $i+1$ planes are averaged and then multiplied by the pressure to obtain a wall flux.

4.5 Initial Conditions and Convergence

Solving hyperbolic equations requires that initial data be known. Ideally, the initial data would be free-stream conditions placed just before the nose of the body and the simulation would seamlessly march down the body computing the flow solution resulting from the body’s influence. However, the explicit method used in this work requires a CFL condition (discussed in Section 4.2) which, with the the finite grid
resolution, prohibit starting before the sharp nose of the body. The simulation must therefore be started, and hence initial conditions given, at some \( x_0 \) station down the body.

For this project it was decided to obtain these conditions (at \( x_0 \)) using a pseudo-iterative approach which assumes a self-similar shape prior to \( x_0 \). A self-similar shape implies that the flow solution over it will also be self-similar. Flow self-similarity provides a convergence criterion for the iterative method. The iterative approach begins with a guess to the exact solution at \( x_0 \). The code then computes and changes this solution until the correct self-similar flow solution is achieved. This self-similar solution can then be re-mapped (rescaled) to the \( x_0 \)-station and used as the correct initial conditions to begin a simulation of a body which has that same self-similar fore-body shape to \( x_0 \) but has an arbitrary shape for the remainder of its length.

Visually it is easy to see if the flow solutions at two cross-planes are self-similar. But it is preferable to have the code automatically decide when convergence is reached. For design comparison work, usually only integral quantities (e.g. lift and drag) are needed. These quantities converge rather quickly, implying that on average the flow has reached a steady state. But they are not sensitive enough to pick up convergence, or the lack thereof, for individual flow features such as shock location. A density difference norm between computational planes can better describe when the details of the flow have converged. In this project a difference norm

---

5If the simulation begins before the nose of the body, the first appearance of the body would be some shape having a finite area. The CFL condition shows that \( \Delta x \) is limited by the planar cell dimensions (\( \Delta y \) and \( \Delta z \)). To resolve a sharp-nosed body on this Cartesian grid would require very small cells and the the CFL condition would produce a prohibitively small marching step size.

6This procedure is analogous to knowingly starting from incorrect solutions to unsteady equations and marching until a steady solution is reached.
needs to allow for the apparent motion of the body and shock on the grid and the increase in shock resolution as the body “grows” on the grid. This can be done using area-weighted densities. The type of difference norm also needs to allow for the spurious errors generated by the face-area calculations (see Section 4.3) and leading edge truncation (see Section 3.5). The norm used for this work is as follows: The solution on the current plane is mapped back to a previous solution plane using conical similarity and the area-weighted density difference is computed for each cell. The root-mean-square of these differences is taken to be the norm for the current plane. \((old = (i - 1) \text{ plane, } new = i \text{ plane, } A=\text{cell area, } \rho=\text{density.})\)

\[
norm = \sqrt{\sum_{k=1}^{\text{numcells}} \frac{1}{2} \left( A_{\text{old}} + A_{\text{new}} \right)_k \left( \rho_{\text{old}} - \rho_{\text{new}} \right)_k^2} \\
\sum_{k=1}^{\text{numcells}} \frac{1}{2} \left( A_{\text{old}} + A_{\text{new}} \right)_k
\]

(4.16)

4.6 Extensions To Second-Order Accuracy

The extension to second-order, spatially, is achieved by reconstructing the solution at the cell interfaces as a piecewise linear function rather than a piecewise constant function. Whether to reconstruct primitive variables or conserved variables has been the subject of debate. Primitive variables were chosen as the reconstructed variable for this work since this seems to give better results for strong shocks [60]. The gradients of the primitive variables are computed at the centroid of each computational cell. The gradient is then used to extrapolate the cell average value from the centroid \((c)\) to the cell interfaces. For example, the density at the left interface, \(\rho_l\), is computed from

\[
\rho_l = \rho_c + \left( \frac{\partial \rho}{\partial y} \right)_c \Delta y + \left( \frac{\partial \rho}{\partial z} \right)_c \Delta z
\]

(4.17)
where $\Delta y$ and $\Delta z$ are the distances between the centroid $(c)$ of the cell and the midpoint of the cell interface.

This extrapolation process can introduce new extrema, thereby causing oscillations in the solution. To remedy this, the values of the reconstructed data are forced to lie within the range of existing data. This process is termed “limiting the data”, and is discussed in 4.6.2.

### 4.6.1 Calculation of Solution Gradient

The gradient of the primitive variables is calculated at the center of each cell only using information from the current $x = const$ plane. A gradient can be computed using out of plane information but the small gain in accuracy doesn’t warrant the increase in work. The group gradient value is obtained by using some conserved quantity to average the group’s cell values. This work used area as the conserved quantity.

Computation of the cell gradients can be computed either by a contour integration around the current cell or a least squares approach. One method does not have great advantage over the other [60]. The least squares method was used for this work.

### 4.6.2 Method of Limiting

There are a number of ways to force the reconstructed data to be monotone. A standard for Godunov-type methods developed by Van Leer [80] forces the data (in 2-D) to be monotone through smoothing the gradients of the variables. A great deal of work has been done in comparing limiters of this type and discussing their
methods of implementation. Therefore, I will refer the reader to [35] [43] and [81] for this information. Results for this thesis were obtained using the Min-Mod limiter or the Superbee limiter for reconstructing the primitive variables. Min-Mod was used the majority of the time since the extra diffusion it caused was easily countered by the adaptive grid. The extra diffusion also may have helped stabilize the solution during initial condition generation.

Bayyuk found that he obtained good results on his grid when the $\hat{y}$ and $\hat{z}$ directional gradients were limited independently using slope information only from the same direction. I used his approach in my work.

4.6.3 Achieving Second-Order in Marching Direction

Second-order in the marching direction is obtained using a two-stage method. Using the notation of 4.1 the solution at level $i + 1$ is found from

\begin{align*}
    f^{(0)} &= f^i \quad (4.18) \\
    f^{(1)} &= f^{(0)} + \frac{\tau}{2} Res(f^{(0)}) \quad (4.19) \\
    f^{(2)} &= f^{(0)} + \tau Res(f^{(1)}) \quad (4.20) \\
    f^{i+1} &= f^{(2)} \quad (4.21)
\end{align*}

Where $Res$ is the flux residual $\sum_{l=1}^{np, jk \text{ faces}} F_l \cdot A_l$. 
CHAPTER V

CODE VALIDATION and CARET WING

THEORY

The 3-D code was checked using known results from cones, delta wings and caret wings. The following sections show that the code produces accurate shock locations and shocked-state values.

The caret wing, while not a practical waverider shape, is useful for understanding and analyzing general features of waverider flow; especially off-design flight. In this work, caret wings played an important role in code validation and as initial condition generators for more complicated waverider shapes. In addition, I will use the caret wing to introduce some waverider analysis techniques and to discuss some off-design flight flow features. The leading edge analysis procedures carry over to analysis of arbitrary waverider shapes.

This chapter will also address the effects of rounding the leading edges. The 3-D code used in this work requires a blunted leading edge. However, this results in a slight detachment of the shock at the leading edges, and a consequent expansion of
the high pressure flow around the tips.

Though the leading edge is slightly rounded, relative to the span of the body it is still sharp. This chapter will address the extent of the errors generated as the Cartesian grid tries to resolve the sharp corners.

5.1 Initial conditions

Free-stream initial conditions were specified for all validation cases. The code was then used in a pseudo-iterative manner to find the correct initial conditions, see Section 4.5. Since the bodies are all self-similar these initial conditions were then the final solution sought. The solutions will only be shown in cross-plane views, since the other solution planes are the same except for a geometric scale factor.

5.2 Convergence

The number of iterations necessary for convergence depends on the definition of convergence. Ideally we want the initial conditions to achieve self-similarity. However, in the initial design stage usually only integral quantities such as lift and drag are required. Convergence of the lift and drag values occur much sooner than self-similarity on most of the shapes I used. Another measure of convergence is the density norm as defined in Section 4.5. The number of iterations required to meet any of the above definitions could be the same (as in the case of the cone flow) or they could be drastically different (as in the case of a caret wing). A general pattern has emerged, however. The lift and drag generally converge first or very soon after
the density norm converges. Surface profile convergence is generally last and can occur at roughly the same time as the lift, drag and norm or it may occur much later. The approximate amount of time necessary for each convergence definition is listed in table 5.1. For a usual refinement level these times are roughly the same if run on different platforms using at least a 200 MHz processor\(^1\)

<table>
<thead>
<tr>
<th>Convergence of</th>
<th>Num. of Iter</th>
<th>Approx. Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hrs.</td>
<td></td>
</tr>
<tr>
<td>(C_t) and (C_d)</td>
<td>3000-5000</td>
<td>36</td>
</tr>
<tr>
<td>Dens. Norm</td>
<td>4000-10000</td>
<td>48-65</td>
</tr>
<tr>
<td>Surf. Profiles</td>
<td>4000-15000</td>
<td>48-130</td>
</tr>
</tbody>
</table>

Table 5.1: Approximate Number of Iterations and Run-Time Hours Needed for Different Levels of Convergence

The caret wing has one of the largest discrepancies between different convergence times. Yet, the largest solution difference between the first convergence time and the last is only 1.5% and occurs in the drag coefficient. This implies that for first-order design, the definition of convergence is not terribly important and the lowest level will still give good results while potentially decreasing design turn-around time considerably.

\(^1\)A Sun Ultra 2 (200 MHz) and a PC (Pentium II - 233 MHz) were used for these simulations.
5.3 Computed Solution on a Yawed Cone Compared to the Expected Solution

A flow simulation was performed on a 10° half-angle cone placed in a Mach 4 free-stream yawed 10° to the cone axis. The solution is presented after the density and pressure profiles achieved self-similarity. Figure 5.1 is a plot of the computed pressure contours for the yawed cone. The symmetric shock that would be formed in an axis-symmetric flow is pushed off-center in this case. The shock is stronger than an axis-symmetric case on the bottom and becomes non-existent on top as the flow relaxes toward free-stream. The contour lines are smooth and symmetric except

![Pressure Line Contours](image)

Figure 5.1: Pressure Contours of a 10° Cone Yawed 10° in Mach 4 Flow

where the grid was recently refined or unrefined by the automated process.

Figure 5.2 compares the computed density solution (the curve) and the AGAR-
Figure 5.2: Density On the Surface and Within the Shock of a $10^\circ$ Cone Yawed $10^\circ$ to the Free-stream Mach 4 Flow

DOGRAPH [37] tabulated values (the squares). The plot shows the density on the surface of the cone and on a horizontal line through the center of the shock. When pressure is plotted against the tabulated values the same shape and error magnitudes are obtained. Small oscillations occur over the entire body due to the way the grid approximates the conical cross-section at each x-station. The horizontal line across the bottom of Figure 5.2 shows the free-stream density value. We can see that the density on the body as well as within the shocked region is accurately computed. The largest difference between the tabulated and the computed values is on the expansion surface and is 3.5%. Figure 5.2 also shows that the shock position is well captured. The shock thickness is 3 cells.

Figure 5.2 shows that oscillations appear on the compression surface of the cone.
For this case the amplitude of these oscillations is less than 1.0%. However, when the number of cells on the body was reduced by a factor of 4 the variation in the oscillations on the compression surface was 2.5% and the percent difference from the tabulated value was 1.5%. This same phenomenon was seen when the cone was yawed at only 5 degrees and was refined to the same (coarser) level. The oscillations appear as a slight wobble in the surface profile at one interaction but then appear as a large bump within 500 iterations. The bump then essentially disappears but the cycle continues to repeat without a noticeable decrease in the amplitude of the error wave. The origin of the initial error is unknown. Since the amplitude of the oscillations increase with decreased refinement of the grid, it seems that geometry errors are a likely source. It may be that the geometry errors from early in the simulation remain in the solution. If this is the case adding more damping (increasing the artificial dissipation and/or using boundary procedures) should eliminate these errors. Most steady state calculations include more damping than is being used in this project. However, the accuracy of the results is very good and while increasing the damping would produce better surface plots it would not significantly improve the accuracy.

The yawed cone solution looks simple, but it is actually quite difficult to compute. Solutions to cone flow have been the subject of debate and controversy for decades. Many of the conical solution features are subtle, second-order effects. Most of the difficulty stems from the fact that the streamlines on yawed cones wrap around the body and create a very thin vortical layer as seen in Figure 5.3. Anderson in [2]
Figure 5.3: Entropy Contours for a 10° Yawed Cone in a Mach 4 Flow in Conical Coordinates

Figure 5.4: Entropy on the Surface of a 10° Yawed Cone in a Mach 4 Flow
gives a good explanation of this. Other references are [75] and [18]. To summarize, a streamline which approaches the vertex and then moves to the bottom (compression side) of the cone passes through the shock and acquires an entropy value, say s1. This streamline proceeds to wet the surface of the cone. This means that the entire surface of the cone should have a constant entropy value of s1. However, a streamline which approaches the vertex and then moves to the top of the cone does not, in this case, pass through a shock at all and retains the free-stream value of the entropy. Thus, on the top of the cone a singularity exists where entropy and vorticity are multi-valued. Entropy follows the streamlines and Figure 5.3 shows the wrapping of the surface by the streamlines creating an entropy layer with a singularity at the top of the cone.

The very strong entropy gradients in this layer lead to high velocity gradients and strong vorticity. These large gradients result in more errors either through truncation errors or by triggering the scheme to produce more numerical viscosity and hence entropy$^2$. These errors will increase as the gradients increase due to either an increase in flow speed, flow deflection angle or body curvature.

Entropy is a very sensitive measure of the accuracy of a solution method [66]. In the case of a yawed cone it is especially interesting and difficult. Figure 5.4 shows the entropy on the surface of the cone. The variation on the surface of the cone is 0.013 when the free-stream value is 0.624. This neglects the region between the asterisks. The entropy value expected at the stagnation point on the surface of the cone is 0.663. The computed value at the stagnation point is 0.6709. That is about

$^2$Entropy production is proportional to some power of the cell size over the curvature [72].
a 1.2% error. The entropy difference generated on the surface of the cone (ignoring the singularity) is very close to the same when the cone has four times the number of cells on the body. The entropy layer over a yawed cone is very thin and not likely to be resolved even on the finest grid.

Two cases where entropy should be constant on the surface are a cylinder and a symmetric cone. The difference in entropy on the surface of a simulated free-stream cylinder is less than $10^{-4}$. But the entropy on a symmetric cone begins to show significant differences on the surface. Rough experiments showed that the entropy difference is exaggerated as the flow conditions become more severe. Entropy variations develops partly due to the incorrect initial conditions supplied (free-stream) and partly due to the shock capturing method. The problem of entropy production in Euler codes is well known [61] [25] [66] [53].

The average yawed cone solution can easily be computed using a coarse grid. However, I found spurious surface features may develop. When the same $10^\circ$ cone solution was computed using a quarter of the cells on the body surface, the same solution through the shock and average solution on the body surface were obtained. However small oscillations seen on the expansion side of the density profile in Figure 5.2 were greatly exaggerated. Of note is that when the $10^\circ$ cone was yawed at only $5^\circ$ no oscillations appeared even for the coarser refinement.

The accuracy of the solution on average are still very good in all cases. However, in extreme flow conditions such as highly yawed cones the flow solution details on the surface of the bodies are a little suspect. Marquina proposed a hybrid scheme using
Roe’s FDS and a Lax-Friedrichs scheme. This method uses an artificial viscosity which acts like a heat conduction mechanism to help reduce spurious inviscid flow features in unsteady flow [25]. His method supplies the entropy fix to FDS yet still provides good resolution of shocks and other difficult flow features. This method could be tested to see if it reduced the entropy errors and spurious flow features on coarser grids in steady flow.

5.4 Theoretical Analysis of On-Design Caret Wings

Before I present the simulation results of flow over a caret wing let us look at some caret wing flow features and the development of an on-design caret wing analytic solution.

The shock over a waverider is assumed to be globally weak\(^3\) so that the flow remains supersonic. However, with respect to the leading edge (in the plane perpendicular to the leading edge) this shock may be either strong or weak depending on the body’s sweep. Whether to design a body to have a strong or weak shock with respect to the leading edge is important. The decision not only affects the undersurface pressure obtainable but also the character of the flow when off-design. As will be seen below, a change in the angle of attack, \(\alpha\), or the Mach number affects the flow differently depending on whether the starting point was a design having a strong or weak shock normal to the leading edge.

If we are given a caret wing geometry (sweep (\(\Lambda\)) and body angle (\(\omega\))\(^4\)) the curve,
D, in Figure 5.5 along which a planar shock would be attached to the wing’s leading edges can be drawn in the Mach-δ plane. The curve along which an attached shock no longer exists can also be drawn. This curve is labeled T in Figure 5.5.

The Mach-angle of attack \((M - \alpha)\) plane in Figure 5.5 is divided into four regions

![Diagram](image)

Figure 5.5: On-Design and Detachment Curves for a Caret Wing With a Given Sweep, \(\Lambda\), and Body Angle, \(\omega\)

by the design and detachment curves. The design curve, D, can be divided into two parts S and W at the point where the detachment curve, T, is tangent to it. The lower part of D, W, represents the conditions which will produce a weak planar, leading edge shock. The, S, portion is the set of conditions where a strong attached, planar, leading edge shock would result.

### 5.4.1 On-Design Caret Wing

The leading edge streamlines of a caret wing at design flight conditions will encounter the shock at the leading edge and remain parallel to the centerline of the body in the post-shock flow. From Figure 5.6 notice that the interacting characteris-

See Appendix D for angle definitions.
tics (the projection of the Mach cone onto this planform view) on the bottom surface of the body are of opposite families, $R^+$ and $R^-$. Thus, they do not interfere with each other and uniform flow is maintained under the entire surface. In this particular example the leading edge shock must be a weak one since the characteristics lie inside the leading edges of the wing. For the same flight conditions increasing the sweep of the wing by a sufficient amount forces the characteristics to lie outside the leading edges of the wing, resulting in a strong shock at the leading edges. Also note that the uniform flow already satisfies the symmetry boundary condition that exists at the body centerline.

![Figure 5.6: Characteristics and Streamlines on Under-Surface of Caret Wing at Design Conditions](image)

It is also sometimes useful to look at conical waverider flows in conical coordinates. This system projects the streamlines onto a sphere which is centered at the body’s apex\(^5\). When looking at a cross-section of a wing with uniform flow in coni-

---

\(^5\)Refer to Smith [75] for further discussion of conical flow and conical streamlines.
cal coordinates the streamlines point radially inward. Figure 5.7 shows a caret wing cross-section at design flight conditions and its velocity vectors in conical coordinates. The flows appear to head toward the apex of the body. The shocked under-surface flow is parallel to the lower ridge line. The magnitude of the velocity will diminish to zero in this projection as the center points are approached.

![Figure 5.7: Velocity Vectors in Conical Coordinate on a Caret Wing at Design Flight Conditions](image)

Since on-design caret wings are designed from 2-D wedge flow (see Appendix A) the uniform state under it is simply computed as the 2-D solution over a wedge with the same flow deflection angle and Mach number. The pressure and density profiles under the caret wing are constant values. The cross-flow velocity is zero.

### 5.5 Computed Caret Wing On-Design Solution Compared with The Expected Solution

To test the accuracy of the code, the results of computed solutions over caret wings having different strength shocks were compared to the known analytical solu-
tions. The computed results compared very well with the expected solution in all cases. Below is a typical example of the accuracy obtained.

A caret wing with a planform angle of 10° and a lengthwise body angle, ω, equal to 6.733° was placed in a Mach 7.199 free-stream flow at zero angle of attack. The simulations used a second-order HLLE method. The results in table 5.2 show the errors are less than 2% when compared to the ideal solution. Figure 5.8 shows that the expected planar shock is obtained from the computation. The leading edge radius was blunted to 75% of the span in the initial cross-plane and 5% in the final plane where solutions are presented. The blunting of the tips leads to an expansion around the leading edges (too small to be seen in Figure 5.8). The pressure and density profiles on the surface of the body can be seen in Figures 5.9 and 5.10. The leakage of the flow around the blunted leading edges can be seen in the dip of density at the leading edges. Thirteen percent of the body is affected by this spillage by more than 2% of the expected solution. The total variation of the density over the lower surface of the body is 6.2% of the average value as a result of the leakage. The

<table>
<thead>
<tr>
<th></th>
<th>Expected</th>
<th>Computed</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_l$</td>
<td>0.0290</td>
<td>0.0293</td>
<td>0.2</td>
</tr>
<tr>
<td>$C_d$</td>
<td>0.00426</td>
<td>0.00422</td>
<td>1.0</td>
</tr>
<tr>
<td>$L/D$</td>
<td>6.800</td>
<td>6.879</td>
<td>1.2</td>
</tr>
<tr>
<td>Avg. $\frac{P_l}{P_1}$</td>
<td>2.051</td>
<td>2.053</td>
<td>0.1</td>
</tr>
<tr>
<td>Avg. $\frac{\rho_l}{\rho_1}$</td>
<td>1.653</td>
<td>1.633</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 5.2: A Comparison of Expected and Computed Solution Over a Caret Wing: $\omega=6.733$, $\Lambda=80.0$, Mach=7.199.

The method and effects of blunting will be discussed in Section 5.5.1.
pressure on the surface of the body is not significantly affected by the blunting in this case.

To test how much of the loss in density and pressure (very slight dip inboard of the leading edges in Figure 5.9) could be attributed to specifying the wrong initial conditions (free-stream), I ran the same case specifying the exact, expected solution as the initial conditions. The same pressure and density profiles were obtained. Therefore, initial condition errors can be eliminated as a major source of leading edge errors. Since the pressure profile remains constant while the total pressure and density dip at the leading edge, entropy seems to be the major error component at the leading edge.

When free-stream initial conditions were specified self-similarity was achieved in
Figure 5.9: The Computed Pressure Profile on the Upper and Lower Surface of a Caret Wing: \( \omega=6.733, \Lambda=80.0, \text{Mach}=7.199 \)

about 16,000 iterations. The lift and drag converged after about 4000 iterations. For this simulation the pressure and density profiles on the body surface took much longer to converge. At 10000 iterations, when the density norm converged, there were still slight dips close to the leading edges in the pressure profile. The difference in the lift and drag between 4000 iterations and 16000 iterations was less than 1.3%.

Oscillations can be seen on the free-stream surface in the density profile, Figure 5.10. These oscillations likely originate from the leading edge truncation errors. As discussed in Section 3.3, the faceted approximation of the leading edges, even when blunted, leads to oscillations in the location of the leading edge. These in turn cause errors in the solution which will be carried along the streamlines. When the leading edge is blunted the high pressure leakage results in the leading edge streamlines being
pushed to the top surface. Thus, all the leading edge resolution errors are seen on the top surface. These errors are seen more in the density than the pressure since the pressure field tends to show more local effects while density retains more global errors [72]. The density errors aren’t meaningful for inviscid design since only lift and drag are considered. The errors in pressure, as will be seen, remain local at the leading edges and don’t affect the majority of the solution.

The next sections will further explore the effects of blunting the leading edges. It will also discuss the extent of the errors due to the Cartesian grid approximation of the sharp corners at the tips.
5.5.1 Solution Details at the Leading Edges

This work is only concerned with attached shocks. So, ideally, bodies would have perfectly sharp leading edges and would not allow any leakage around the tip; the upper and lower flow regions would be completely separated. However, as discussed in Section 3.3, the grid merging routine requires that the tips be rounded. Figures 5.11 show a close-up of the leading edge solution of a caret wing that has been blunted at two different values. They show that more blunting of the leading edge will allow more high pressure flow to leak around to the top surface. On one hand,

![Pressure Line Contours](image)

(a) The Pressure Contours on a Constant Radius Leading Edge - Blunting is .04% of Body Span at this Cross-Section

(b) The Pressure Contours on a Constant Ratio Leading Edge - Blunting is .25% of the Body Span

Figure 5.11: A Comparison of the Spillage Around the Leading Edge of a Caret Wing Blunted to Different Radii. (Both Pictures Display approx. 30.0% of the Body)

the spillage could be seen as an error since the computed solution will not match the ideal body being simulated. However, in reality the tips of a sharp edged body would have to be blunted to reduce leading edge temperatures. When the leading edges are blunted an entropy layer will develop due to the detachment of the shock.
A genuine error is produced because of the way the grid resolves sharp corners. This error results in high spiked values at the leading edges in the surface profiles.

5.5.1.1 How the Leading Edge Radius Affects the Solution

The bodies simulated in this work were for the most part self-similar. The exception is at the leading edges. To ensure that in later design studies we are comparing like with like it is desirable to have the leading edges be as sharp as possible to contain the compressed air. To satisfy the grid criteria for body resolution while still having a reasonable number of cells, the radius of the tip was kept fixed as the body span increased in each cross-plane. This allowed the bluntness of the leading edge relative to the span to approach zero, as in the ideal case. This method, however, makes it difficult to determine the extent of the flow changes caused by blunting since the solution will never completely converge. But in practice, it is obvious when the solution away from the tips converges. After this point, the forces, density norm and solution profiles change by an insignificant\(^7\) amount over hundreds of iterations. When the radius of the leading edge is kept constant, the effects due to blunting are too small to be measured in the pressure profile. Therefore, the lift and drag on the bodies is not significantly different than the ideal, zero radius case. What is affected is the density and cross-flow velocity profiles. These profiles are affected over the entire body. When the radius is kept constant the amount of leakage at the leading edge is kept constant, reducing the gradients to the centerline as the span increases.

\(^7\)Significant will be defined as any amount greater than 2.0\%. 

Thus, the errors are continuously reduced.

To be able to quantify how bluntness affects the solutions, two caret wing cases were run which were truly self-similar. These cases did not keep the same leading edge radius, but rather maintained a constant leading edge radius to span ratio. The results of these cases give an idea of the maximum errors that will be generated by blunting since the radius to span ratio is always larger than in the constant radius simulations. Figure 5.11 compares a constant radius case with a truly self-similar body which keeps a fixed leading edge to span ratio. The constant ratio case ends up with a larger radius and thus more spillage. The constant ratio case results are compared to the theoretical caret wing solution and to the constant radius cases in table 5.3.

The amount of body affected by the blunting increases considerably as the radius of the tip is increased. About 80.0% of the body is affected by more than 2.0% when the leading edge is blunted by 1.0%. Less than 52.0% of the body is affected by the same amount when the radius is decreased by a factor of 4. The effect of the pressure spillage (Figure 5.11) extends much further in the 0.25% case than in the constant radius 0.05% case. The increase in leakage is also seen in the magnitude of the cross-flow velocity in table 5.3. The magnitude of the cross flow velocity decreases by almost half when the radius is four times smaller. But it decreases by more than five times when the radius is further decreased by a factor of five using the constant radius method. When the blunting is large the effects appear in the surface pressure profile as well as the density profile. But even for the 1.0% blunted
<table>
<thead>
<tr>
<th>LE Radius ( % of Span )</th>
<th>Ideal</th>
<th>Init.-Final</th>
<th>Constant $\frac{LE}{Span}$ Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. $\frac{p_2}{p_1}$</td>
<td>2.051</td>
<td>2.053</td>
<td>2.061</td>
</tr>
<tr>
<td>Avg. $\frac{p_2}{p_1}$</td>
<td>1.653</td>
<td>1.633</td>
<td>1.597</td>
</tr>
<tr>
<td>$C_l$</td>
<td>.0290</td>
<td>.0293</td>
<td>.0293</td>
</tr>
<tr>
<td>$C_d$</td>
<td>.00426</td>
<td>.00422</td>
<td>.00446</td>
</tr>
<tr>
<td>$\frac{L}{D}$</td>
<td>6.800</td>
<td>6.879</td>
<td>6.569</td>
</tr>
<tr>
<td>% Diff. $\frac{L}{D}$ from Ideal</td>
<td>0.0</td>
<td>1.2</td>
<td>3.4</td>
</tr>
<tr>
<td>% of Body Where Density less Than 2% of Average Value</td>
<td>0.0</td>
<td>12.64</td>
<td>51.2</td>
</tr>
<tr>
<td>Max. Cross-Flow Vel. at LE</td>
<td>0.0</td>
<td>.022</td>
<td>.26</td>
</tr>
<tr>
<td>Num. Iterations to Profile Convergence</td>
<td>-</td>
<td>&gt; 16000</td>
<td>&gt; 16000</td>
</tr>
<tr>
<td>% Body Truncated</td>
<td>0.</td>
<td>.23</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Table 5.3: The Effects of Blunting the Leading Edges on a Caret Wing
case, the pressure, density, and lift are within 5.0% of the ideal solution. However, the drag is increased by 13.6%.

The constant ratio case was expected to converge faster than the constant radius case, but it didn’t. It took over 16000 iterations for both of them to attain “self-similar” surface pressure and density profiles away from the immediate vicinity of the leading edges.

5.5.1.2 Leading Edge Errors Due to Sharp Corner Resolution

![Pressure Cross-Sections.](image)

Figure 5.12: An Example of Grid Truncation Errors at the Leading Edge

In addition to the errors due to leakage there are errors due to the way the grid resolves sharp curves. As seen in Figure 3.4 the amount of truncation will vary between cross-planes. The more the tip is resolved by the grid the less severe
these differences will be. But each increase in the refinement level increases the computation time by at least a factor of two. So, there must be a balance between how much to resolve the body and how many errors are generated.

The errors generated at the leading edges are confined to the merged groups of cells next to the body where the tip is truncated and their neighboring groups. The errors generated are quickly dissipated away from this region as shown in Figures 5.12 and 5.13. An average of 14 cells are affected by truncation errors of more than 2% of the expected value. For the average refinement levels used in this work the local error magnitude is large but the width of the region affected by this error is less than 2% of the body. For the purposes of this work, this error level was acceptable. By refining the body more the error magnitude and the width of its influence on the
body could be reduced.

5.6 Off-Design Performance of Caret Wings

Calculation of the post-shock streamlines and flow characteristics is important in understanding and predicting the flow patterns on waveriders at different flight conditions. For this reason I will briefly discuss the ideas and a couple of examples, relegating the mathematical details to appendices D and E. These calculations are more helpful in understanding the flow near the leading edges than the interior flow. A more detailed method-of-characteristics computation would be required to understand details of the interior flow.

When a caret wing is flown at off-design conditions the shock will move either inside or outside the plane of the leading edges. The streamlines of these off-design flows can be calculated using oblique shock theory, where small portions of the leading edge can be viewed locally as a yawed wedge. The free-stream velocity can be decomposed into components parallel and perpendicular to the leading edge. These components are then used in conjunction with oblique shock theory to compute the post-shock flow locally. By calculating the leading edge post-shock state, the angle of the streamline with the centerline and the Mach cone about the streamlines can be found. For cases where uniform flow exists inside the leading edge this method gives the exact flow state for this region. These calculations can also help estimate the width of the uniform region and predict where an embedded shock may exist under the body.
Whether the leading edge shock at design flight conditions is strong or weak depends not only on the free-stream Mach number and the angle of attack but also on the sweep of the body. The Mach-angle of attack ($M - \alpha$) plane in Figure 5.14 is divided into four regions by the design and detachment curves. The area of the flow regions change as the body sweep ($\Lambda$) changes. As an example, Figure 5.15 shows the flow regions for a wing with $\omega = 6^\circ$ and $\Lambda = 80^\circ$ are much different than the regions for a wing with $\omega = 6^\circ$ and $\Lambda = 60^\circ$. Also shown is a sample of how $\omega$, affects the regions.

Thought experiments originally done by Squire and Roe in [76] along with some scattered wind tunnel tests suggest the flow patterns that will exist in the various regions of the plane. My own numerical simulations agree that the following flow patterns can occur in the given regions, though they may not be the only possible solutions.

---

8Refer to Kuchemann [41] for an in-depth discussion of this.
9see also [82]
Figure 5.15: An Example of How the Flow Regimes Change for Caret Wings of Different Geometries.

5.6.0.3 Region I

When a waverider designed for a point on the strong branch of the design curve is flown at a decreased angle of attack \( \alpha \), region I, the shock globally becomes weaker but the post-shock flow viewed from the leading edge still has enough momentum in its approach to the body center line that a second, weaker shock develops inside the leading edges. This condition can be predicted using 2-D oblique shock theory. This same condition can be reached if the wing was designed for a weak leading edge shock but then flown at an increased Mach number.

Figure 5.16 shows an example of post-shock, off-design flow characteristics that may exist in this region. The characteristics at the leading edges have a large enough
inward angle that they eventually cross (interact) with the characteristics from the other side of the wing. This interaction causes a change to the flow state that, if too rapid, may result in a weak secondary embedded shock inboard of the leading edge. Prior to reaching this coalescence of characteristics, or weak shock, the characteristics and streamlines remain straight resulting in uniform flow over a region inside the leading edges. Once the interaction zone is encountered, the flow may be shocked to a near uniform state or continue compressing until the symmetry condition is reached.

If we look at Figure 5.17, a cross-sectional view of the same velocity vectors in conical coordinates, the over turning of the flow at the leading edges becomes more obvious. The secondary embedded compression zone slows the cross-flow so that it
Figure 5.17: Right Side of Body Velocity Vectors in Conical Coordinates on a Caret Wing at Off-Design Flight Conditions - Region I

approaches near uniform conditions. The surface pressure on a cross-section of a wing in region I looks, theoretically like Figure 5.18. The non-uniform center region may be either a further compression of the flow or an expansion of the flow. These alternatives are indicated by the dashed curve in Figure 5.18. A numerical simulation produced the profile in Figure 5.19.

The Mach number for the simulation was 10. The on-design case is the one discussed in Section 5.5 where the Mach number was 7.199. The sharpness of the embedded shock was about the same when FDS and HLLE were used. The profile in Figure 5.19 was computed using extra dissipation in the HLLE scheme. For reasons still unknown, this substantially improved the sharpness of the embedded shock's pressure profile.

As with the yawed cone case, this case exhibited long period oscillations in the
Figure 5.18: Theoretical Pressure Profile Over a Caret Wing Cross-Section For A Weak Design Wing Flown at Off-Design (Higher) Mach Number. Region I

Figure 5.19: Computed Pressure Profile Over a Caret Wing Cross-Section For A Weak Design Wing Flown at Off-Design (Higher) Mach Number. Region I
center (most compressive portion) of the body. These take an extremely long time to be removed from the solution. Using a first-order scheme to compute the same case gives a smoother profile and damps these errors faster but the shock is again smeared.

5.6.0.4 Region II

Figure 5.20: Computational Pressure Profile Over a Caret Wing Cross-Section For a Strong Design Wing Flown at Off-Design (Lower) Mach Number.

Region II

The flow pattern expected in region II has not been completely decided [76] [82]. It has not been proven one way or another whether the strong leading edge solution will actually exist in practice. If it does, a caret wing designed for a strong leading edge solution and then flown at a higher angle of attack may detach. If the strong
leading edge solution does not manifest, then the weak leading edge shock will move further inside the leading edges. In this case a second embedded shock may develop as in Region I [82].

Figure 5.21: One Example of an Off-Design Caret Wing in Region II

A computed solution for this region is shown in Figure 5.20. The shock is attached at the leading edges and is manifested as the strong leading edge shock solution. I obtained this solution by starting from free-stream initial conditions and iterating until self-similarity was achieved. At these high pressure ratios errors made in splin- ing the body surface become very apparent using second-order calculations. These errors manifest in oscillations in the surface pressure and density profiles. The profile shown is a first-order calculation. The body and the shock wave shape are shown in Figure 5.21.
5.6.0.5 Region III

For a weak leading edge design that is flown at a lower Mach number the shock becomes weaker and moves outside the leading edge plane. The flow, in this case, must expand along a streamline so that the flow is parallel to the body axis by the time the streamline reaches the centerline. Like the pressure profile over a delta wing, the pressure for this case will show an expansion region in the center. And there may or may not be uniform regions just inside the leading edges as seen in, Figure 5.22. The computed pressure profile for this case is shown in Figure 5.23.

5.6.0.6 Region IV

The shock is detached above and to the left of detachment curve. Detachment can be predicted where the oblique solution at the leading edge does not exist. This occurs when the wing is flown at too large of an angle of attack for the free-stream
Figure 5.23: A Computed Pressure Profile Over a Caret Wing Cross-Section For a Weak Design Wing Flown at Off-Design (Lower) Mach Number. Region III
Figure 5.24: Pressure Contours on a Caret Wing Cross-Section For A Weak Design Wing Flown at Off-Design (Lower) Mach Number. Region IV

Mach number or at a too low a Mach number for a given angle of attack. A case having a detached shock and the body cross-section are shown in Figure 5.24. The computed pressure profile for this case is shown in Figure 5.25. The on-design case is again the weak caret wing solution discussed in Section 5.5. The case shown here has a free-stream Mach number of 4.0, for the same wing geometry.

5.7 Other Numerical Experiments with Caret Wings

The use of fine grids to numerically simulate caret wings has resulted in some interesting solution features that would be interesting to explore further. Simulation of a caret wing at a strong design condition showed some unexpected behavior. When the solution was sought by beginning the simulation with free-stream initial
conditions, the weak leading edge shock solution was obtained\textsuperscript{10} and an embedded shock pattern was obtained instead of the expected planar solution. It is unclear whether this solution was an extremely long transient solution or was a final steady solution. When the exact strong leading edge solution was imposed, it remained. Both solutions were stable to perturbations in Mach number. Though it hasn’t been proven, it is suspected that the strong leading edge solution will not exist in practice [76] [82].

The appearance of both of these solutions raises the question of the existence of a region about the design curve somewhere close to this case where the flow solutions

\textsuperscript{10}The leading edge solution satisfied the jump conditions for the weak leading edge shock while the central portion of the shock satisfied the jump conditions for the 2-D wedge.
Figure 5.26: A Close-up of Density Contours on a Caret Wing with an Embedded Shock

may not be unique. These non-unique solutions may not have been seen in wind tunnel testing due to viscous effects, and not in numerical simulations due to the coarseness of the grids. The secondary solution can only be obtained when a fine, adaptive grid is used. The possibility of non-unique solutions for this inviscid flow would not be out of line with recent works like [19] [6] [59] and [36] which find non-unique solutions in flows analogous to this.

The second interesting flow feature is that of the spreading of the secondary, embedded shocks near the body. This is not a result of a coarsening of the grid. On the contrary it only appears when the grid is sufficiently refined. A picture of the phenomenon is shown in Figures 5.26 and 5.27. This feature does not appear to be due to boundary conditions as the rest of the body’s boundary conditions are correct.
Figure 5.27: A Close-up of Pressure Contours on a Caret Wing with an Embedded Shock

It also does not appear to be a manifestation of the merging of cut cells on the body because the area covered by the spread shock is much greater than would be covered by a couple of groups of merged cells. The rest of the jump quantities are correct so it doesn’t seem to be a problem in the equations or their solution. The spreading only appears in stronger flow conditions and only for second-order calculations.

Entropy is generated at the leading edges due to the blunting of the tip and the detachment of the shock. The entropy production from the leading edge results in an entropy layer. The embedded shock wave then interacts with this layer as a shock would interact with a boundary layer. As shown in Figure 5.26, the density is reduced at the wall. Note that the entropy layer from the leading edge does not extend to the center line but only exists for a portion of the body inside of the leading edges.
By increasing the distance between the wave speeds in the HLLE scheme the width of the embedded shock at the wall can be significantly reduced. I found entropy production increases by about 30% when the radius was increased by four\textsuperscript{11}.

Since shock thickening persists even when the grid is refined, only for second-order calculations and when the entropy layer does not exist where the embedded shock meets the wall, I believe another mechanism other than the entropy layer is contributing to the shock thickening problem at the wall. In unsteady problems it is known that reflecting shock waves can exhibit a shock thickening problem at the wall [25] [61] [66]. The cause is attributed to the incorrect generation of entropy around the reflecting shock. This problem is countered by adding more viscosity to the scheme or, as previously discussed, modifying the viscosity to include a heat conduction like behavior [25].

Rather late in this work I realized the problems being simulated have most of the difficult flow features known to exist for shock capturing methods: Slowly moving shocks, shock reflections and bodies with high curvature (blunted leading edges). In the process of generating initial conditions, the solutions on caret wings (and the parameterized bodies to be discussed) contain embedded shocks which may or may not remain in the final solution. These shocks move quite slowly toward their final position and can sometimes have noise behind them just as in unsteady problems. In strong flow conditions these shocks also exhibit the infamous kinking. All these erroneous features occurred with both HLLE and Roe’s FDS and can be seen in

\textsuperscript{11}However, the case with the larger leading edge radius had a less refined grid and the effects of cell size is not yet clear.
Figure 5.26. But after the recent experiments with decreasing the embedded shock width at the wall. I believe that increasing the viscosity in the HLLE method or implementation of Marquina’s hybrid method will reduce most of these errors.

5.8 Computed Delta Wing Solution Compared to the Expected Solution

The average pressure over a delta wing having an attached shock can be approximated as the pressure over a 2-D wedge of the same deflection angle placed in the same free-stream. This is the exact solution according to linear theory and is a good approximation by Thin Shock Layer Theory. Since these are the two extremes of strong and weak shocks, approximating the pressure in this manner is probably within 5% error [71]. The flow was computed over a delta wing with a planform half-angle of 34.285°, a deflection angle equal to 9.2646° and placed in a free-stream with Mach number of 6.0.

The average pressure ratio over a wedge of angle 9.2646° placed in a free-stream of Mach number 6.0 is 3.3784 and the density ratio is 2.268. These are the expected values listed in table 5.4. This table shows that the average solution over the delta wing agrees well with what was expected. Density has a larger error as was the case with the caret wing. I believe this is due to the blunted leading edges, as discussed previously.

The flow conditions on the delta wing are such that the Mach cone at the apex of the body does not encompass the whole body. As a result, there will be a uniform
flow region along the leading edges. The flow in this region is only influenced by
the shock at the leading edge and not the flow in the center of the body. In such
cases, the solution just inside the leading edge can be computed as if it were a yawed
wedge. The mathematical details to this problem can be found in appendix F.

<table>
<thead>
<tr>
<th>Average Solution</th>
<th>Expected</th>
<th>Computed</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{p_2}{p_1} )</td>
<td>3.378</td>
<td>3.377</td>
<td>.056</td>
</tr>
<tr>
<td>( \frac{p_2}{p_1} )</td>
<td>2.268</td>
<td>2.162</td>
<td>4.6</td>
</tr>
</tbody>
</table>

Table 5.4: A Comparison of Expected and Computed Average Solution Over a Delta Wing: \( \Lambda=55.715 \), \( \delta=9.265 \), Mach=6.0

Figure 5.28: TheComputed Pressure Contours on a Delta Wing: \( \Lambda=55.715 \), 
\( \delta=9.265 \), Mach=6.0
Figure 5.29: The Computed Pressure on the Surface of a Delta Wing: \( \Lambda = 55.715 \), \( \delta = 9.265 \), Mach=6.0

The pressure contours shown in figure 5.28 show the planar shock from the leading edges meeting the non-planar shock of the non-uniform region inside the leading edges. A smooth expansion occurs between the uniform flow at the leading edges and the centerline of the body. The flow must meet the symmetry condition at the centerline and run parallel to it.

The uniform flow inside the leading edges is contained within an angle equal to the planform angle minus the Mach cone angle about the shocked state velocity vector minus the angle that this vector makes with the centerline. Again, the details of this can be found in appendix F. The two regions of the flow can be clearly seen in figure 5.29, which shows the pressure on the surface of the body in a cross-sectional
Figure 5.30: The Computed Density on the Under-Surface of a Delta Wing: $\Lambda=55.715, \delta=9.265, \text{Mach}=6.0$

The magnitude of the leading edge pressure matches the expected delta wing value very well (see table 5.5). The density profile is also expected to show a uniform region at the leading edges and an expansion region in the center. But the computed density solution in figure 5.30 only hints at the expected solution. Starting with incorrect initial conditions results in an error on the compression centerline of the body. And as seen in the previous section, the effects of blunting the tips is seen much more in the density than in the pressure. In this case, the gradient caused by the leakage around the blunted tips interacts with the uniform region causing a density gradient instead of a constant value. Entropy production may also contribute to the low leading edge density value.

\footnote{Note that the pressure over the windward or compression side of the wing is plotted on top.
A close-up of the leading edge region density profile on the lower surface is shown in figure 5.31. I assumed the region between the marks to be the region trying to be uniform. The region to the left is the region most affected by blunting the tips. Between the marks the value of the density differs from the expected uniform value by 1.9% at the right mark to 4.4% at the left mark. The region from the tip to the first mark represents 19.0% of the body. The flow spillage results in an average density that is in error by about 5.0%.

The uniform region should make an angle of $20.02^\circ$ to the leading edge in the planform view. The lift and drag converged in about 5000 iterations. The density norm converged at around 6000 iterations. But the density profiles and the angle of the uniform region continued to adjust until about 11000 iterations. After this point the surface profiles are self-similar. A blow-up of the transition region between the uniform and non-uniform flow is shown in figure 5.32. The uniform analytic solution in this region is shown by the horizontal line which is intersected by a slanted line.
Figure 5.31: A Close-up of the Leading Edge Density Profile on the Surface of a Delta Wing

The intersection point marks the end of the uniform region in this cross-section based on analytic computations given in appendix F. We can see that the length of the computed uniform region is slightly smaller than the analytic solution. The slope of the pressure curve at the intersection point can be determined from equations given by Roe in [70] (refer to appendix F). This slope is shown in figure 5.32 by the slanted line. The non-uniform region should intersect the uniform region in a sharp corner. The shock capturing method used in this work results in a blended region rather than a sharp intersection. A summary of the comparison between the leading edge expected solution and the computed solution are given in table 5.5.

Figure 5.30 shows small oscillations in density on the free-stream surface. These originate from the resolution errors of the blunted leading edges as discussed in
Figure 5.32: A Zoom of the Transition Region Between the Uniform and Non-uniform Regions on the Surface of a Delta Wing

The angle of the leading edge shock should be equivalent to that over a yawed wedge at the same conditions. The angle in this case is expected to be 30.519° in the plane normal to the leading edge or 12.39° as seen from the cross-plane\textsuperscript{13}. The computed leading edge shock angle is captured well with only 2.0% error.

5.9 Summary

As in unsteady problems, it is difficult to compute strong shock problems. From my experience it appears that pressure ratios across the shock less than four can be

\textsuperscript{13}To find the angle that the shock makes with the lower delta wing surface in the cross-plane, $\beta_{\text{unit}}$, we must use equations D.28, D.30, D.38, D.37 and figure D.3 in appendix D.2.
easily computed without erroneous flow features, except for extra entropy production. For pressure ratios in excess of about four spurious flow features may develop such as kinked shocks, smeared shocks on the body wall, noise behind embedded shocks and oscillations in the surface profiles. In unsteady flow increasing the viscosity or making corrections to its form reduces these problems. I believe this will also work here.

While these extraneous flow features are unpleasant, the average solutions on the bodies are still very accurate. For airframe design which relies on integral quantities, the presence of these erroneous flow features is not too much of a hindrance. However, blunting of the leading edges does significantly affect the design process due to the reduction of lift, drag and the creation of entropy.

It is hard to say what part of the decrease in $\frac{L}{D}$ is due to the spillage and how much is due to the decrease of span. The 1.0% blunting leads to a 10.0% drop in $\frac{L}{D}$ and 10.0% of the body is truncated. When approximately 2.0% of the body is truncated, the performance drop is 3.4%.

The rate of convergence of the density norm was about the same in the 1.0% and the .25% cases. In fact, the convergence rate was not worse for the constant radius case even though the body is not perfectly self-similar. Therefore, using a constant radius is better for capturing the features of a sharp edged body and does not require more time for convergence. The most important factor determining running time is the level of body and flow refinement. Having too many cells increases the computation time for no gain in solution accuracy.
The errors due to the resolution of the tip only affect 2.0% of body in the tip region. The magnitude of these errors doesn’t result in any significant error in average density, pressure, lift or drag. The error magnitudes and region of the body they affect can be reduced by greatly increasing the refinement levels on the body. But this doesn’t seem to be warranted in first-order design type problems, especially when it is desirable to have fairly quick turn-around time.

Overall the solutions obtained using this code are very good. They may in fact be more accurate than is necessary to find optimal bodies using lift and drag criteria. First-order flow solution results may be sufficient.
CHAPTER VI

A COMPARISON OF Waverider Compression Surfaces

As a demonstration of the direct design method, a case study is presented where flow solutions over a family of body shapes are computed and compared in order to find the best compression surface shape for a Mach 7.199 flight.

The shapes used in this study stem from a base (or parent) caret wing which when flown at the same free-stream conditions would support a planar leading edge shock. For Mach 7.199 flight a caret wing having a sweep(\(\Lambda\))=80^o and angle of attack(\(\delta\))=4.31^o supports a moderate strength, planar leading edge shock. The bodies compared in this work, therefore, all have sweep(\(\Lambda\))=80^o but the effective angle of attack varies for each under-surface shape. The comparison is made easier by having self-similar\(^1\) bodies and free-stream upper surfaces. The combined requirements of a free-stream upper surface and self-similarity require the top to be a caret shape.

\(^1\)The bodies are self-similar except at the leading edges. The radii of the leading edges is held at a constant value so that as the body span increases the leading edge becomes relatively more sharp. This was discussed in Sections 3.3 and 5.5.1
Since this is an academic study and not aimed at specific mission requirements, the coefficient of lift and the lift-to-drag ratio were used as figures of merit with an eye toward improving the volume over that of a caret wing. Thin shock layer theory and caret-wing theory are used as aids in comparing the lift-to-drag ratios of the test bodies.

6.1 Preliminaries

6.1.1 Choosing the Body Shapes

The compression surface of the base caret wing is replaced by a series of quartic curves. The quartic definition of the under-surface allows shapes ranging from a low angle-of-incidence caret wing with a rounded lower ridge (Figure 6.1(a)), to one resembling a delta wing with some relief of the compression surface (Figure 6.1(b)). In addition, a quartic shape can resemble what others have found to be optimum waverider shapes for a variety of mission requirements [68] [12] [45].

The under-surface of the base caret wing, shown in Figure 6.2, is replaced by a quartic curve defined by $z = ay^4 + by^2 + c$. (The cubic and linear terms are set to zero to enforce left-right symmetry.) $c$ is found by matching the top surface with the bottom. This leaves two free parameters, $a$ and $b$. These coefficients are constrained first by the upper surface. The under-surface must not cross the upper surface and, in fact, due to limitations of the grid, must form a reasonable angle with it. Second, the angle that the flow would see at the leading edge must not be so large that it results in a detached shock. The calculations are given in Appendix
Figure 6.1: Examples of the Range of Shapes Achievable With a Quartic Definition
Figure 6.2: Base Caret Wing Compared to Quartic Under-Surface Test Shape

G. The parameter space and its restrictions are shown in Figure 6.3. I choose lines within this allowable parameter space and called their intercepts \( b_j \). This provides a convenient method for choosing test bodies. Once \( b_j \) is set and an \( a \) is chosen, \( b \) is calculated from equation G.11,

\[
b = -2y_{bc}^2a + b_j. \tag{6.1}
\]

After the shape restrictions are met, there is still considerable latitude in choosing an under-surface shape. The location of the case numbers in Figure 6.3 shows the under-surface parameters used in each case. Bodies get thinner as \( b_j \) decreases and \( a \) increases. The thinnest body is case 23 and thickest is case 11.

All the bodies simulated in this study, have a leading edge radius that is initially 1.0% of the span. The absolute separation is held constant for all cross-sections, so that at the final cross-section (where results are presented) the leading edge radius
Figure 6.3: Quartic Under-Surface Parametric Constraints Shown in Parameter Space
Figure 6.4: An Example of a Truncated Leading Edge, The Initial Points and The Body After Splination

is closer to 0.5% of the span. Figure 6.4 shows that after the top and bottom surface separation is specified, points are inserted outboard and between these in such a way that after splination a nice rounding is achieved. Figure 6.2 shows that truncating the leading edge results in a shortening of the initial span. At successive cross-sections the span will approach that of its sharp edged counter-part, also shown in Figure 6.2. At the initial cross-section, about 6.0-10.0% of the span is truncated. At the cross-plane where solutions are presented 2.5-4.0% of the span is lost.

6.1.2 Initial Conditions

On bodies which would generate strong shocks the code was not robust when free-stream values were used as initial conditions. This is a result of the oscillations
in the body's surface due to the grid generation. In such cases, supplying initial conditions more in the neighborhood of the expected solution yields good results. The test bodies in this chapter had medium strength shocks so that free-stream initial conditions could be specified.

The use of self-similar bodies eliminates a step in the solution process. This is because the converged solution is the correct initial condition for the body as well as the final solution sought.

6.1.3 Convergence

The density norm used in this work was discussed at the end of Section 4.5. This norm along with the convergence history of lift and drag were used to judge when a solution had reached an acceptable level of convergence. After the density norm and lift and drag values converged, visually comparing the pressure and density contours at two different marching planes showed the flow was often still adjusting and had not reached a steady state. A typical density norm convergence history is shown in Figure 6.5. Figure 6.6(a) shows the density on the surface of the body at 4500 iterations after the density norm, lift and drag values converged. Figure 6.6(b) shows the density on the surface at 10350 iterations. Neither the density norm nor the integral quantities (lift and drag) consistently converged before the other. I accepted a case as converged when all three values converged. It takes about 36 hours to run a case to this convergence level on a computer with at least a 200 MHz processor. The difference between the integral quantities at this definition of convergence and at three times the number of iterations, when the surface profiles converged, was less
Figure 6.5: Typical Convergence History on Quartic Test Bodies

than .05%.

6.1.4 Comparison Criteria: Figure of Merit

Whether or not a body shape is “better” than another depends on many factors, the most important of which is the application. For most applications, however, lift and drag are useful quantities in choosing the best shape. Other important factors may be the amount of volume required and its distribution, controllability, heating characteristics and ease of incorporating control surfaces and propulsion systems.

For this project lift, drag and an increase in cross-sectional area over that of a caret-wing were used as performance criteria upon which to compare different airframe shapes. No attempt was made to account for viscous effects or to include the base drag. If the body shapes used in this study were used as is, the base drag
Figure 6.6: Comparison of Norm-Converged Solution and Steady-State Solution
would be a significant part of the total drag. However, they would not be used exactly as is, but would be modified to lessen the base drag, probably by tapering the body and relaxing the flow toward free-stream before the tail is reached.

6.1.4.1 Computation of Lift and Drag Forces

The numerical computation of the inviscid body forces is straight-forward. Lift, the force on the body perpendicular to the free-stream velocity, is simply the pressure at the body surface times the planform-projected area. The drag is the body surface pressure times the exposed body area projected into a plane perpendicular to the marching direction (\( \mathbf{\hat{x}} \)). Using the notation of Figure 6.7 the coefficients of lift and drag are computed for a cross-section of depth \( \Delta x \) from

\[
C_l = \frac{\sum_{k=1}^{N} P_{wall_k} \Delta y \Delta x}{S \Delta x q_{\infty}} \tag{6.2a}
\]
where \( N \) is the number of body segments in the cross-section, \( S \) is the body span at the current cross-section and \( q_\infty = \frac{1}{2} \rho \infty V_\infty^2 \). In practice the cells next to the body are cut. So, \( \Delta y \) and \( \Delta z \) in the above equations are the actual wall cell face lengths for each individual cut cell.

### 6.1.4.2 Lift-to-Drag Ratio as a Figure of Merit

A caret wing generated from a wedge with angle \( \delta \) in a flow having an angle of attack \( \alpha \) has a lift-to-drag ratio which is inversely proportional to \( \delta \). Thus, more slender bodies have higher values of \( L/D \) than less slender bodies for inviscid flow [64]. The dependence of lift and drag on body slenderness can be removed in order to better compare different body shapes. For this work, the ratio \( \frac{C_l^2}{C_d} \) is used as a slenderness-independent measure of the lifting efficiency. The hypersonic limit of this ratio can be used to help compare the different test shapes. The value of the ratio at the hypersonic limit, \( K_\delta = \infty \), for a caret-wing is

\[
\frac{C_l^2}{C_d} = \sqrt{\gamma + 1}
\]  

[64]. This happens when the upper surface is free-stream, as in the standard caret-wing on-design case. Note that this is not the absolute maximum value this ratio can obtain for an arbitrary shape.

The exact solutions for various on-design caret wings at a given \( M_\infty \) will have coefficients of lift and drag which lie on a curve in the \( C_l - C_d \) plane. This curve along with the curve generated from equation 6.3 will be used in Section 6.2 to compare
different test shapes.

6.2 Summary of Case Study Results

A uniform distribution of under-surface test cases (cases 6 through 19) was run within the allowable parameter space. The parameters for these cases are shown in figure 6.3 and Table 6.1. After the base set of solutions were obtained a few more cases (cases 21, 22 and 23) were run to try to improve upon the previous results.

The simulation of cases where the lead coefficient, $a$, was too negative for the given $b_j$ resulted in a strong expansion around the blunted leading edges and the code crashed. These cases are shown in Table 6.1 with asterisks in the results fields. These cases should run without trouble when the leading edges are truly sharp. An attempt was made to decrease the leading edge radius but the level of refinement necessary to capture a sharp enough leading edge made the run times impractical.

The coefficients of lift and drag in Table 6.1 are the converged values for the test cases. The $L/D$ is then the ratio of those coefficients. The “Area Diff.” is the amount of cross-sectional area gained for the test case over the original caret wing under-surface. The cross-sectional area change is a measure of the volume increase for each case. The last column, $C_l^{3/2}/C_d$, is a measure of the lifting efficiency. This ratio removes the effects of the body thickness on performance. This was discussed in section 6.1.4.1.

The dashed curve in figure 6.8 is equation 6.3, the hypersonic limit in TSL of the ratio of $C_l^{3/2}/C_d$. The solid curve represents the family of caret wings with
<table>
<thead>
<tr>
<th>Case</th>
<th>$b_j$</th>
<th>$a$</th>
<th>$C_l$</th>
<th>$C_d$</th>
<th>$L/D$</th>
<th>Area Diff.</th>
<th>$\frac{C_l}{C_d}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theor. Caret</td>
<td>-</td>
<td>-</td>
<td>0.02898</td>
<td>0.04262</td>
<td>6.7995</td>
<td>0</td>
<td>1.1575</td>
</tr>
<tr>
<td>Comp. Caret</td>
<td>-</td>
<td>-</td>
<td>0.02903</td>
<td>0.04220</td>
<td>6.8724</td>
<td>0</td>
<td>1.1721</td>
</tr>
<tr>
<td>9</td>
<td>-0.646</td>
<td>-2</td>
<td>0.08575</td>
<td>0.0178</td>
<td>4.8148</td>
<td>0.12570</td>
<td>1.4107</td>
</tr>
<tr>
<td>10</td>
<td>-0.646</td>
<td>-3</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>11</td>
<td>-0.646</td>
<td>-4</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>15</td>
<td>-0.754</td>
<td>-2</td>
<td>0.07832</td>
<td>0.01518</td>
<td>5.1589</td>
<td>0.10770</td>
<td>1.4439</td>
</tr>
<tr>
<td>16</td>
<td>-0.754</td>
<td>-3</td>
<td>0.09197</td>
<td>0.01998</td>
<td>4.6045</td>
<td>0.14103</td>
<td>1.3960</td>
</tr>
<tr>
<td>17</td>
<td>-0.754</td>
<td>-4</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>18</td>
<td>-0.97</td>
<td>-3</td>
<td>0.07752</td>
<td>0.01514</td>
<td>5.1202</td>
<td>0.10503</td>
<td>1.4256</td>
</tr>
<tr>
<td>19</td>
<td>-0.97</td>
<td>-4</td>
<td>0.08944</td>
<td>0.01972</td>
<td>4.5346</td>
<td>0.13837</td>
<td>1.3564</td>
</tr>
<tr>
<td>20</td>
<td>-0.97</td>
<td>-5</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>12</td>
<td>-0.862</td>
<td>-2</td>
<td>0.07145</td>
<td>0.01336</td>
<td>5.3475</td>
<td>0.08970</td>
<td>1.4295</td>
</tr>
<tr>
<td>13</td>
<td>-0.862</td>
<td>-3</td>
<td>0.08548</td>
<td>0.01756</td>
<td>4.8978</td>
<td>0.12303</td>
<td>1.4232</td>
</tr>
<tr>
<td>14</td>
<td>-0.862</td>
<td>-4</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>6</td>
<td>-1.078</td>
<td>-3</td>
<td>0.06984</td>
<td>0.01340</td>
<td>5.2135</td>
<td>0.08703</td>
<td>1.3774</td>
</tr>
<tr>
<td>7</td>
<td>-1.078</td>
<td>-4</td>
<td>0.08500</td>
<td>0.01776</td>
<td>4.7853</td>
<td>0.12037</td>
<td>1.3954</td>
</tr>
<tr>
<td>8</td>
<td>-1.078</td>
<td>-5</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>21</td>
<td>-0.754</td>
<td>-1</td>
<td>0.06299</td>
<td>0.01150</td>
<td>5.4774</td>
<td>0.07437</td>
<td>1.3747</td>
</tr>
<tr>
<td>22</td>
<td>-0.862</td>
<td>-3.5</td>
<td>0.09077</td>
<td>0.01960</td>
<td>4.6307</td>
<td>0.13970</td>
<td>1.3953</td>
</tr>
<tr>
<td>23</td>
<td>-0.646</td>
<td>0</td>
<td>0.05535</td>
<td>0.00980</td>
<td>5.6474</td>
<td>0.05903</td>
<td>1.3288</td>
</tr>
</tbody>
</table>

Table 6.1: Results of Parameterized Under-surface Test Cases
Figure 6.8: Comparison of Test Cases to the Thin Shock Layer Hypersonic Limit and Exact On-Design Caret Wings for Mach$_\infty=7.199$
increasing deflection angle flown in a Mach 7.199 free-stream. The performance
difference between a caret wing and the test cases can be seen in the distance the
test cases are above the caret wing curve. The caret wing I was trying to improve
upon is marked with a C (computed solution) and an E (exact solution) in figure
6.8.

![Graph showing increase in Volume and $C_l$ Over Computed Original Caret Wing](image)

**Figure 6.9: Increase in Volume and $C_l$ Over Computed Original Caret Wing**

If we look at the increase of lift and volume for each case over that of the original
caret wing (figure 6.9), we see that all the cases have a higher coefficient of lift than
the caret wing, which would be represented by a point at the origin. The vertical
axis of figure 6.9 shows the increase in lift of the case over that of the original caret
wing. The horizontal axis shows the increase in cross-sectional area (proportional to
volume) over that of the original caret wing. Figure 6.11 shows the shapes having
the highest $C_l$. These are the thickest bodies. It can easily be shown that lift is a function of volume for caret and delta wings. However, for an arbitrarily shaped under-surface a clean expression cannot be obtained. Yet, figure 6.9 seems to imply that a linear relation may still exist. A more mathematical discussion is given in appendix H.

All test bodies had higher drag than the original caret wing flown at the same conditions. The increase in drag is understandable since all the bodies are thicker, causing a greater disturbance to the flow. The increase in body thickness means a larger effective angle of attack. This results in a higher post-shock pressure and higher lift than the original caret wing. As expected, the more slender bodies (cases 23, 21, 12) had a higher $L/D$ because they generated less drag while still maintaining
Figure 6.11: A Comparison of the Under-surface Shapes for Cases 16, 19 and 22 - The Highest $C_l$ Cases

good compression. These bodies are shown in figure 6.12. Figure 6.10 displays the change in $L/D$ versus the increase in cross-sectional area over the original caret wing. For bodies which were too thick the small increase in lift was overcome by a larger increase in drag.

We can notice from figures 6.9, 6.10, 6.11 and 6.12 that a small change in the under-surface shape can significantly improve performance, primarily by increasing the lift while keeping the drag about the same. As an example, the $C_l$ in case 12 is 2.6% higher and has 3.7% more area than case 6. It also has 3.8% more lifting efficiency. In a preliminary study this amount of increase may not be exciting, but in a final design process a 3.8% increase in performance would be welcomed.

The effects of body thickness on performance can be eliminated by comparing
Figure 6.12: A Comparison of the Under-surface Shapes for Cases 6, 12, 21 and 23 - The Highest L/D Cases

the ratio $C_l^3/C_d$. Table 6.1 gives the value of this ratio for the test cases. Cases 15, 12 and 13, shown in figure 6.13, have the highest lifting efficiency along with case 18. These are intermediate thickness bodies which balance lift and drag generation. More lifting efficiency is obtained by shaving some area off the center and adding more toward the tips. The increase in lifting efficiency and cross-sectional area over that of the caret wing are shown in figure 6.14.

6.2.1 Evaluation of the Process

This method allows more freedom in body design than other waverider design methods. Except for the thickness of the body and corners, there are no restrictions on body shapes. The bodies do not have to be conical, or even self-similar and we
Figure 6.13: A Comparison of the Under-surface Shapes for Cases 15, 12 and 13 -
The Cases with the Highest Lifting Efficiency

don’t have to be able to compute the streamlines or stream-surfaces over the bodies a priori. Without any formal search for an optimum body it was possible to create bodies which had more volume and better volume distribution than a caret wing while still increasing the $C_l$ and lifting efficiency. This implies that any attempt to optimize the compression surface will reap great benefits.

The draw back to this approach is mainly the amount of time it takes to simulate one case. this time can be reduced considerably. Also, the accuracy level of this study was probably more than is necessary at this level of a design project. Allowing a lower level of accuracy would permit quicker design turn-around.
Figure 6.14: Increase in Volume and Lifting Efficiency Over the Original Computed Caret Wing
CHAPTER VII

BEYOND EULER FLOWS AND EULER CODES

The use of an Euler code in a definitively viscous regime may permit quick preliminary design turn-around but poses the question of how valuable the design will be in real viscous flow. Experimental and numerical performance analysis on waveriders show that even inviscidly optimized bodies perform reasonably well, though maybe not optimally, at their design condition in close to real flow fields. For thicker bodies at lower speeds, inviscid analysis can provide a good estimate of the surface pressure [46]. However, at higher Mach numbers and altitudes, viscous effects begin to dominate the flow field.

Existing ground-based test facilities such as wind tunnels and arc jet heaters cannot duplicate the exact flight conditions of waverider vehicles [51]. The design of practical hypersonic vehicles will thus rely heavily on numerical simulations. Unfortunately, most of the work to improve methods so that “real” conditions can be simulated have been aimed at the unsteady equations. Unsteady methods which in-
clude non-equilibrium real-gas effects\(^1\), first-order turbulence, and viscous methods\(^2,3\) can now predict flow solutions with good accuracy. But it is computationally more efficient to analyze the flow over waveriders using steady space-marching methods. As such, the waverider design and optimization process would greatly benefit from faster and more accurate ways of integrating “real” effects into space-marching methods.

The remainder of this chapter will explore what “real” flow means for waverider flight regimes. There are a few methods available for including some real flow effects in space-marching methods. I will talk about these, but also include a discussion of how real flow effects are included when the unsteady equations are used. The methods used in unsteady flow may provide a starting point for developing analogous methods for the steady equations.

### 7.1 Important Second-Order Effects In Waverider Flow

#### 7.1.1 Viscous Effects

It’s obvious that the inclusion of viscous effects in the design, and certainly in the analysis, of waveriders is necessary to accurately predict performance. Hypersonic speeds at high altitudes produce low Reynolds numbers which exacerbate the effects of viscosity. Under some conditions viscous effects can dominate the flow field and drastically alter the pressure distribution. A significant amount of work has been done to incorporate different viscous effects one at a time (and a few at a time) in an

\(^1\) A few of the many references for computing with real-gas effects are: [51, 84, 78]

\(^2\) Some references about the effects of viscosity and turbulence on waveriders: [52, 12, 23, 77, 20, 21]

\(^3\) A few of the many references for PNS methods are: [42, 17, 74, 83, 85, 29, 5]
attempt to capture the most important viscous effects while still maintaining code efficiency.

The use of parabolized Navier-Stokes (PNS) may be preferable to coupling an inviscid code with a boundary layer code since the assumptions of an inner viscous layer and an outer inviscid layer are not valid over a large part of the hypersonic regime. An advantage of using parabolized Navier-Stokes (PNS) is that, barring separation, all viscous interactions are taken care of automatically.

Early hypersonic research seemed to have a good handle on the various viscous effects and methods for determining when each may be important. However, current work seems to include different effects at random, without regard to which has the greatest impact on the current flight regime. Chang, in his thesis work at the University of Maryland [16], has made a good contribution toward prioritizing the viscous effects. He showed (and Mundy and Hasen experimentally verified [52]) that at higher Mach numbers and higher altitudes a significant part of drag (20-40%) comes from self-induced pressure interactions over and above skin friction drag\(^4\). Chang's work includes velocity-altitude plots delineating where viscous interaction effects are significant for modern hypersonic waveriders. In particular, he finds that pressure-induced viscous interaction effects become important for Mach numbers greater than 16 and altitudes greater than 140,000 ft. [46].

The importance of vorticity and shock wave/boundary layer interactions on the performance of hypersonic vehicles are still not fully understood. This knowledge would be helpful in deciding what effects to include at what level of the design

\(^4\)These effects increase the lift by 12%
process and what the minimum design tool requirements are to sufficiently predict performance of candidate designs for a given mission.

7.1.2 Effects of Turbulence

Corda and Anderson [23] suggest that most practical waveriders will have boundary layers that are mostly turbulent. Laminar aerodynamic predictions will therefore significantly over-predict the lift-to-drag ratio over these bodies. Also, shapes optimized using only laminar boundary layers may be quite different from those taking into account transition and turbulence.

7.1.3 Real Gas Effects

Equilibrium real-gas behaves much differently than ideal gas and thus should be recognized when trying to accurately predict heat transfer rates in hypersonic flow. As with viscous effects there are various levels of “real-gas” effects. Deviations of the thermodynamic properties of real air from an ideal gas with constant specific heats occur at high temperatures in the form of molecular vibrational excitation, chemical reactions and ionization. These processes occur at finite rates which can be characterized by relaxation lengths. When these lengths are much smaller than the characteristic dimension of the vehicle, equilibrium thermo-chemical properties can be assumed. Equilibrium real-gas effects vary with temperature and can be included fairly easily using table look up and interpolation techniques, or fitting methods based on the known, tabulated behavior. Pfitzner and Weiland [58] find that including these effects increases the computational time by about 30%.
One manifestation of real-gas effects is the swallowing of compression energy by the internal degrees of freedom (vibration, and dissociation). This leads to a decrease of the temperature and to an increase of density behind the bow shock in comparison to the ideal gas case [58]. However, the downstream forces (i.e. \( \frac{P_2}{P_1} \)) are only significantly affected in high speed expansion flows [64]. In non-equilibrium flow, the leading edge shock or bow shock will become curved and a resulting pressure gradient will occur in the streamwise direction, decreasing the pressure from the nose and approaching the equilibrium pressure toward the tail. This non-equilibrium pressure distribution results in a slight nose-up pitching which is under-predicted using equilibrium methods [13] [4].

Equilibrium fluid dynamic equations of motion are closed using thermal and caloric equations of state to relate the thermodynamic variables. Following [58], the most general equations of state are

\[
\text{Thermal: } P = \rho RT Z(\rho, T) \tag{7.1}
\]

\[
\text{Caloric: } e = f(T, \rho), \tag{7.2}
\]

or combined,

\[
P(\rho, e) = \rho RT(\rho, e)Z(\rho, e), \tag{7.3}
\]

where \( e \) is the internal energy, \( R \) is the specific gas constant \( (R = C_p - C_v) \) and \( Z \) is the compressibility factor. For non-dissociating and non-ionizing gases at low enough pressures that intermolecular forces are not important, and temperatures generally lower than \( 500^\circ C \), the compressibility factor is roughly \( 1 \) (\( Z \approx 1 \) for thermally ideal
gases) and the energy does not depend on density so that these can be written as

\[
\text{Thermal: } P = \rho RT \quad (7.4)
\]

\[
\text{Caloric: } \epsilon(T) = R \int C_v(T) dT. \quad (7.5)
\]

For thermally ideal gases the specific heats \((C_p\text{ and } C_v)\) are only functions of temperature; for calorically perfect gases they are constants independent of temperature [64].

In the usual waverider flight regime of \(T=500-2000^\circ C\), the vibrational degrees of freedom become excited and \(C_v = f(T)\). Thermally ideal gas may still be assumed as long as the relaxation lengths remain much shorter than the smallest flow field dimension.

As the relaxation lengths and flow field dimensions become comparable non-equilibrium effects become increasingly important [13]. When temperatures become hot enough for dissociation, such as in reentry, chemically non-equilibrium real-gas should be used as the flow medium [58]. When non-equilibrium effects are included, species conservation equations must be solved along with the gas dynamic equations. This increases the computational time.

McLaughlin [47] found that adding equilibrium effects to an inviscid design procedure was not beneficial since the temperature rises are not significant enough to cause chemical reactions. Viscous flows, however, do produce high enough temperatures, in many practical applications, to necessitate the inclusion of chemically reacting boundary layers [3].
7.1.4 The Continuum Assumption

Attention must also be paid to the assumption of continuum flow when the Navier-Stokes or Euler Equations are used. The Euler equations is an acceptable model for flows where the Knudsen\(^5\) \((Kn)\rightarrow 0\) and the Navier-Stokes for \(Kn \ll 1\). At high altitudes the density becomes low and the mean free path large, which implies large Knudsen number. When \(Kn > .1\) the assumption of continuum gas dynamics should be abandoned and replaced with kinetic-theory based methods [46]. Even for \(Kn > .01\) Navier-Stokes begins to lose its applicability [31]. Work done at the University of Michigan with Extended Hydrodynamics and moment closure methods may prove useful in this regard. Their methods use the 10 and 35 moment equations, thus increasing the range of applicability to \(.1 < Kn < \) between 5 and 10. However, no work has been done to modify these methods for space marching [31].

7.1.5 Inclusion of Viscous and Real Gas Effects

The University of Maryland’s state of the art inverse design methodology uses an inviscid analytical flow field and incorporates analytical approximations to account for the effects of laminar [12] and turbulent boundary layer skin friction, pressure induced viscous interactions [16], and real-gas effects into the optimization procedures. But, as in most inverse design methods, it does not include all possible viscous effects. For example, shock wave/boundary layer interactions and vortical viscous interactions have not yet been incorporated.

For analysis, as opposed to design, the combination of space-marching the PNS

\(^5Kn = \lambda/L, \lambda=\)the mean free path, \(L=\)characteristic length of vehicle
in wholly supersonic regions and time-marching in the remainder of the flow, as done by Molvik and Merkle in [51], appears to be a very efficient and effective method. This would eliminate the need for any viscous layer approximations or restrictions. Their method also allows for chemical non-equilibrium.

For design, however, speed and efficiency are important and we want to stick with space-marching. Some very accurate and efficient methods have been developed recently which allow finite-volume space-marching of the steady PNS equations.

PNS algorithms are limited to entirely supersonic flow fields with the exception of the subsonic viscous layer. PNS are obtained by 1) neglecting unsteady terms and streamwise viscous diffusion terms within the Navier-Stokes equations and 2) modifying the streamwise convective flux vector to permit stable time-like marching of the equations downstream of initial data.

The basic PNS have been integrated using a variety of finite-difference schemes. Currently, refinements of the non-iterative, implicit, approximate-factorization schemes developed by Vigneron et al. [83] and Schiff and Steger [74] represent the state-of-the-art. The primary difference between these two methods is in the treatment of the streamwise pressure gradient, which if not properly handled will allow information to propagate upstream through the subsonic viscous layer. Schiff and Steger suppress this backward traveling of information via the "sub-layer approximation" which assumes constant pressure through the boundary layer. Vigneron's technique involves splitting the streamwise flux vector into two parts: one part is treated as a source term which is either neglected or evaluated in the supersonic region. According to
Lawrence et al. [42], “Beam-Warming type algorithms”, such as Vigneron, Schiff and Steger’s, “suffer because the central differencing of fluxes across flow field discontinuities tends to introduce errors into the solution in the form of local flow property oscillations.” These must be controlled by some type of user-specified smoothing.

In [65] Rieger develops a finite-volume version of these methods which still requires the input of some artificial dissipation. Lawrence and Tannehill [42] have developed an upwind implicit, finite-volume, space-marching method which is non-iterative and, using Vigneron’s technique, requires no smoothing. The implicit nature of this method eliminates the marching step-size restriction and allows free-stream conditions to be specified ahead of a sharp-nosed body. However, computations over blunt-nosed bodies still require initial conditions from an outside code. A continuation of this work [78] adds the capability of including equilibrium real-gas effects.

Surveying the many different ways of implementing the PNS equations can make your head spin. There are implicit, explicit, iterative, and non-iterative methods, as well as permutations of the former with the latter; there are techniques for including first-order turbulence, upwind information propagation, real-gas effects, strong viscous interactions and separation. Including additional effects in the simulation comes at a price. To include turbulence or shock wave/boundary layer interactions (e.g. separation) an iterative approach is necessary, and hence design time is increased. As far as I can tell, a single sweep method can account for induced pressure and vortical interactions but not shock wave/boundary layer interactions, turbulence or separation. Once more information is known about which effects need to be included
when, there should be a reasonably fast and accurate method available.

It doesn’t appear that any one method for incorporating non-equilibrium effects out-shines the others. The method that seems to have the most in common with this work is that of Molvik and Merkle [51] Their upwind finite-volume, combined time-space-marching algorithm includes, along with PNS, strongly coupled reaction rate equations.

7.2 Benefits of Direct-Design Approach

Recent work done at the University of Maryland to design optimal waveriders accounting for viscous effects using both skin friction and, more recently, induced pressure viscous interactions [16], has concluded that waverider shapes optimized in the three cases of inviscid, skin friction and skin friction plus viscous interactions are significantly different. However, their inverse design method only allows body surfaces which lie along inviscid streamlines (boundary layer displacements cannot be accounted for). One study, in fact, stated that better performance may be obtained by carving out the body to negate the effects of the boundary layer displacement [16]. Thus, even viscous-optimized waveriders, designed by inverse methods, require adjustments to be truly optimal in the actual flow field.

By using an inviscid, direct design method, one can incorporate knowledge of the true physics of the flow (viscous and real-gas effects) into the preliminary design. Additional features such as thicker wing tips and concave compression surfaces [79] to reduce local heat transfer rates can also be incorporated. More detailed viscous
effects can be accounted for later, after an inviscid optimum design has been found. For example, it would be a simple modification to an inviscid design to displace the body surfaces in order to negate the effects of boundary layer displacement.

The direct design approach can, if desired, include turbulent boundary layer and viscous and real gas effects in the design/optimization process. Viscous components can be added through Parabolized Navier-Stokes equations. Techniques for including various levels of non-equilibrium real-gas effects and turbulence are within the realm of today’s space-marching methodologies. Including all these at once may result in an impractical design/optimization code. But as knowledge increases about which effects are the most important to include in each flight regime, intelligent choices can be made about which ones to include in preliminary design analysis so that there is good aerodynamic prediction.
CHAPTER VIII

FUTURE WORK AND CONCLUSIONS

This project is intended as a first-look into the feasibility of direct design methods for waveriders. An accurate space-marching CFD code with good resolution of non-simple bodies and flow features is used to simulate the flow over waverider shapes that may not be obtainable using inverse-design methods. The current code also has the advantage over inverse methods of being able to analyze the performance of waveriders away from design conditions.

However, as with any “first-look” work, there are places for improvement. In Section 8.1 I discuss the major areas where design flexibility and code efficiency can be enhanced. There are still many studies which can be done to advance the power and usefulness of direct design methods. I will discuss a few and suggest how direct design methods may be used in real design problems.
8.1 Suggested Improvements to Current Methods

8.1.1 Initial Condition Specification

Design turn-around time will be improved by finding a faster method of specifying the initial conditions. It may seem odd to use a space marching code in an iterate-until-convergence manner. But this method affords flexibility in airframe design. I also expected it to require less work and to yield a better quality initial condition than approximating the initial condition using an analytic method like thin shock layer theory. However, this method of iteration severely slows the design process. The convergence of surface profiles for strong shocks (the worst case) takes about 15,000 iterations and the body width increases 32 times. This means it takes at least the same amount of computational time, probably twice as much, to compute the initial condition as it does to run a design simulation. In order to obtain an initial condition solution in these strong shock cases it is necessary to slowly increase the deflection angle of the body from zero to the desired angle. Even so, extreme flow changes often exist which require manual changes to code parameters like the CFL number and allowable apparent body motion in order to continue the simulation. Bayyuk doesn’t seem to have as much trouble running similar high pressure shock cases using the unsteady equations [9]. Thus, the use of an unsteady Euler code may produce initial conditions faster, simply due to the increased stability and resulting lack of human intervention needed.

To help eliminate errors dissipation is often added to 2-D steady computations. In addition, the outer boundary is placed far from the body and used to attenuate
errors generated in the solution process. When shocks are involved the errors bounce around between the shock and the body, never reaching the outer boundary. The errors, therefore, remain in the solution a long time. A shorter convergence time may result if damping is added by modifying the body boundary procedures in this space-marching code.

Convergence time is not significantly reduced by starting with initial conditions closer to the correct initial state (whether generated from an approximate method or using the solution for a similar body shape). It does, however, eliminate the need for fiddling with the marching step size and ramping of the angle of incidence in strong shock cases. The reason may be that the long wavelength errors are still present and limit the convergence rate. As I understand it, the same slow convergence can be expected from 2-D unsteady equations due to the same mechanism [11]. However, as mentioned above, the unsteady equations may prove more stable for strong shock initial condition generation and hence slightly faster in human time. A 3-D unsteady code may produce initial conditions faster due to the inclusion of the 3-D flow effects and thus faster elimination of the long wavelength errors. Any initial condition method which requires interpolation of the results onto the initial space-marching plane will generate errors which will again take time to eliminate.

Another alternative would be to switch to an implicit method which would allow starting with free-stream initial conditions ahead of the body. An implicit method may also be preferable if viscous effects are included, as discussed previously.
8.1.2 Body Shapes

For this project body shapes were restricted to those having self-similarity, partly due to the difficulty in generating initial conditions; and to those having a free-stream upper surface, so that the performance of shapes could be easily compared. It turned out that using an iterative method for initial condition generation was not an efficient method (see above). In addition, when both self-similarity and a free-stream upper-surface is required the upper surface is limited to a caret top. Thus, the family of shapes accessible for comparison in this work was limited. These shape restrictions can be eliminated by finding a more efficient method of initial condition generation. Shapes having non-free-stream tops can be easily compared with a few code modifications. The beauty of the grid generation method is that once the initial conditions are given, the flow around any body shape can be computed.

8.1.3 Grid Code Improvements

The grid generation code is still evolving and as such the limits imposed by it (leading edge bluntness and truncation, over-refinement of flat surfaces, decreased spatial step size, and inefficient methods of flux-area computations) can be eliminated in the future.

Chapter III discusses the possibility of improving the merging algorithm to allow merging around sharp corners and using sub-cell resolution to enable perfectly sharp leading edges to be modeled. Bayyuk also plans to implement multiple level refinement on the body so that the sharp (yet still blunted) leading edges don’t drive
geometric refinement for the entire body.

As discussed in Section 4.3, the flux computation and area calculations in the code are so intimately linked that computing the area of a face requires two flux computations for the same face. Even without looping over the groups, this could be made more efficient by computing the side areas and fluxes for the group geometry. Since the code loops over the cells, it would be necessary to ensure that this computation is only added to the residual once. Better yet, the geometric and flux calculations should be separated so that the flux computation is only done once for any face.

8.1.4 Optimization

Until many of the previous improvements are made there isn’t much to be gained by implementing an automated optimization procedure. Once the code is made more efficient a simple directional search on the body parameterizations may be used to aid in the search for an optimum shape.

8.1.5 Unexpected Flow Features

The study of hypersonic waveriders will necessarily include very strong shocks. In this work strong shocks often resulted in spurious flow features like surface oscillations on yawed cones, kinked, embedded shocks which spread at the wall and long wave length noise. Many of these features aren’t apparent in first-order computations. This could be because they don’t exist. But more likely it is because they have been smeared out. These features may appear for first-order computations if the grid is
sufficiently refined since the features only appear for a sufficiently refined grid in second-order computations. The errors may be a result of the flow solver. Many of the flow features seen have been documented for first-order unsteady problems by Quirk [61] and were fixed by adding more viscosity to the scheme. The variety of unexpected features will undoubtedly lead to a variety of causes and more work needs to be done to determine the source and reason for each.

8.2 Advantages of Direct-Design Approach

After discussing the specific problems of the current implementation I am still enthusiastic about the direct design approach. Most practical vehicles will be only waverider inspired; they will have non-trivial modifications to the waverider body. After these changes, even if the flow field was previously known, it no longer will be and a full-simulation will be required.

When the direct design method is used more of the final body modifications and additions can be included in the optimization process. Waveriders designed by the inverse method can be optimized given one generating body shape. But there are an infinite number of body shapes which can be used to generate the flow field. With the direct design method we can compare a variety of shapes without being restricted to one type of flowfield.

It is of course important to be able to simulate the off-design behavior of these vehicles. For steady flow, off-design conditions can easily be simulated with this method.
8.3 Enhancing the Usefulness of Direct-Design

This study was intended as a first-look into the direct design of waveriders. As is usual with investigatory studies, this study raises many questions. To proceed with the direct design method there needs to be a prioritization of the important real flow effects (viscous effects, shock-boundary layer interactions, turbulent boundary layers, non-equilibrium effects, real gas effects, etc.) to include in the design code. In addition an optimization procedure should be implemented. There is also a lot of work that can be done to correlate the effects of body shape changes and performance. There also needs to be more research done to explore the regions of the flight regime where the expected solutions are still not obvious.

8.4 Conclusions

The work herein constitutes a good beginning to a practical direct design code. The automatic, adaptive mesh makes it as easy as possible to use arbitrary body shapes which may develop very complicated flows. The code is also fairly easily adapted to include more real flow characteristics.

Even after improvements to the code are made, this design process may not be as fast the inverse methods. But used in conjunction with inverse methods, perhaps after inverse methods have bracketed a space or when real flow effects and control surfaces are needed, a direct design method can greatly expand the design space and produce more practical, realistic results.
APPENDICES
APPENDIX A

Inverse Method of Waverider Design

Traditionally a waverider wing was designed FROM a given analytical flow. This is termed an inverse design approach. Kuchemann [41], Nonweiler [55], Rasmussen [64] and Roe [69] are good references for basic waverider design. Figure A.1 shows how a caret wing waverider may be designed\(^1\). Similarly, waveriders may be generated out of other known flowfields such as cones, seen in Figure A.2, ellipses, and combination bodies such as seen in Figure A.3. References [38][62][63] discuss these more complicated waverider design methods.

\(^1\)This figure has been adapted from one produced by Phil Roe.
Figure A.1: The Use of Inverse Design Methods to Create A Caret Wing Waverider
Figure A.2: A Waverider generated from Cone Flow, Shown With its Attached Shock.
Figure A.3: Examples of Combined Body Shapes From Which Waveriders Can be Generated.
APPENDIX B

Using Parameter Vector in Derivation of

Eigenvalues and Eigenvectors

The non-conservative form of the Euler equations written in terms of the parameter vector are much easier to work with when deriving the Jacobian matrices, eigenvectors and eigenvalues.

\( CW_x + EW_y + JW_z = 0 \)  \hspace{1cm} (B.1)

\( W \) is the parameter vector consisting of

\[
W = \begin{pmatrix}
\sqrt{\rho} \\
\sqrt{\rho u} \\
\sqrt{\rho v} \\
\sqrt{\rho w}
\end{pmatrix}
\]  \hspace{1cm} (B.2)

\( C \) is the matrix defined as:

\[
C = \begin{bmatrix}
w_2 & w_1 & 0 & 0 \\
2h_o \frac{2-\gamma}{\gamma} w_1 & w_2 \frac{\gamma+1}{\gamma} & -w_3 \frac{2-1}{\gamma} & -w_4 \frac{2-1}{\gamma} \\
0 & w_3 & w_2 & 0 \\
0 & w_4 & 0 & w_2
\end{bmatrix}
\]  \hspace{1cm} (B.3)
Matrix $J$ was not needed since the flux function was written for the two-dimensional problem.

To convert the Jacobians and eigenvectors back to the conserved variables the Jacobian matrix for the mapping of the parametric variables to the conserved variables is needed and is given below.

\[
E = \begin{bmatrix}
    w_3 & 0 & w_1 & 0 \\
    0 & w_3 & w_2 & 0 \\
    2h_o (\frac{2-1}{\gamma} w_1 - w_2 \frac{2-1}{\gamma} w_3 \frac{\gamma+1}{\gamma} - w_4 \frac{2-1}{\gamma}) & 0 & 0 & w_4 & w_3
\end{bmatrix}
\]  \tag{B.4}

\[
P = \mathcal{J} \mathcal{U} = \begin{bmatrix}
    2w_1 & 0 & 0 & 0 \\
    w_2 & w_1 & 0 & 0 \\
    w_3 & 0 & w_1 & 0 \\
    w_4 & 0 & 0 & w_1
\end{bmatrix}
\]  \tag{B.5}
APPENDIX C

Body Parametrization

In 3-D space marching the body surface is developed as the cross-sectional planes are collected. To get a smooth body I chose to make each cross-section a function of the previous one. Two functions, \( f_{xz} \) and \( f_{yx} \), act on the body points of the previous cross-section like scale factors, expanding and distorting the previous body cross-section to give the new cross-sectional shape. These functions will control how the body looks in the x-z plane (the body’s sideview) and x-y plane (the body’s planform). Refer to Figure C.1. Aside from symmetry of the vehicle about the x-z plane we would like to make the body shape as general as possible, so the functions \( f_{xz} \) and \( f_{yx} \) can be specified to be any shape as long as they are consistent with the initial cross-section.

For example, a full caret-wing body is obtained by defining the body cross-section to be four line segments, two for the top surface and two for the bottom. These line segments are then ”expanded” according to the two shape functions \( f_{xz} \) and \( f_{yx} \). For the caret wing these functions are both linear \( f_{xz} = z_{le} = z_0 + \ell (x - x_0) \) and \( f_{yx} = x = x_0 + k (y_{le} - y_0) \) as seen in the ”Angle Definition for Finding Relations”
Body Cross Section View
(Plane Normal to Marching Direction)

Angle of Attack

Planform

Figure C.1: Projections of Body
figure in Appendix D. The line segments of the body expand leaving \((y_0, z_0)\) a fixed point as viewed in the \(y\)-\(z\) plane.

A parabolic planform implies that \(f_{xz} = z_{le} = z_0 + \ell (x - x_0)\) and \(f_{yx} = x = x_0 + k(y_{le} - y_0)^2\). However, note that, a parabolic planform and a caret top will not maintain a streamwise top surface if the ridgeline is made to follow the \((x_0, y_0, z_0)\) line. A streamwise surface could be maintained if the top is allowed to be an expansion surface, \((y_0, z_0) = f(x)\).

Specifying the initial cross-sectional shape sets \(z_{le}, y_{le}, z_0, y_0\) (for a streamwise top). The free parameters are \(k\) or \(\ell\) and \(x_0\) or \(x(1)\). A linear angle of attack is usually preferred therefore \(f_{xz}\) is linear, \(\ell\) is a constant and \(k\) will be found by matching the three shapes at the leading edge. Either \(x_0\), the location of the body nose, or \(x(1)\), the location of the specified (first) cross-section, can be chosen. The other is obtained from matching shapes at the leading edges.

Assuming the new cross-plane’s \(x\) location is known and satisfies all stability conditions, the process of creating a new body cross-section out of the previous one is as follows: Note: 'new' refers to cross-section being calculated, 'old' refers to the previous cross-section.

- choose the form for \(f_{xz}, f_{yx}, \ell\), either \(x_0\) or \(x(1)\) and an initial cross-section shape, which implies \(z_0, y_0, z_{le}, y_{le}\).

- from the specified cross-section and the above chosen parameters find the remaining two parameters

- find the new leading edge’s maximum \(z\) coordinate: \(z_{le,new} = f_{xz}\)
• find the new leading edge’s $y$ coordinate: $y_{le_{new}} = f_{y_{p}}$

• find the scale factor for the $\hat{z}$ direction: $scale\ factor_{z} = \frac{z_{le_{new}} - z_{0}}{z_{le_{old}} - z_{0}}$

• find the scale factor for the $\hat{y}$ direction: $scale\ factor_{y} = \frac{y_{le_{new}} - y_{0}}{y_{le_{old}} - y_{0}}$

• the old $y$ coordinates are then scaled to give the new $y$ coordinates: $y_{le_{new}} = (y_{le_{old}} - y_{0}) \times scale\ factor_{y} + y_{0}$

• the old $z$ coordinates are then scaled to give new $z$ coordinates: $z_{le_{old}} = (z_{le_{new}} - z_{0}) \times scale\ factor_{z} + z_{0}$

The last two equations perform a shift to the origin, a scaling and then shifts the shape back to the fixed point $(y_{0}, z_{0})$.

The code implementation of the above requires knowing how much each point in a cross-section will move in order to form the new cross-sectional shape. The equations become:

$$\delta y = y_{le_{new}} - y_{le_{old}} \quad (C.1)$$

$$= (y_{le_{old}} - y_{0}) \times scale\ factor_{y} + y_{0} - y_{le_{old}} \quad (C.2)$$

$$= (y_{le_{old}} - y_{0}) \times (scale\ factor_{y} - 1.) \quad (C.3)$$

$$\delta z = z_{le_{new}} - z_{le_{old}} \quad (C.4)$$

$$= (z_{le_{old}} - z_{0}) \times scale\ factor_{z} + z_{0} - y_{le_{old}} \quad (C.5)$$

$$= (z_{le_{old}} - z_{0}) \times (scale\ factor_{z} - 1.) \quad (C.6)$$
Building the 3-D body out of cross-sections allows for a change in $f_{xz}$ and $f_{yx}$ between any two sections. Thus, bodies with discontinuous planforms and angles of attack can be created.
APPENDIX D

Calculation of Leading Edge Flow on Waverider Bodies Using Oblique Shock Theory

First we need to define the coordinate systems used in this work.

D.1 Body Angle Definitions for Caret and Delta Wings

The following can be obtained from looking at Figure D.1 which uses a coordinate system having the \( x \)-axis along the top ridge line of the body.
Figure D.1: Angle Definitions for Finding Relations
\[
\tan(\beta) = \frac{P}{R} \quad \cos(\beta) = \frac{R}{\ell}
\]

\[
\tan(\delta) = \frac{t}{R} \quad \tan(\Omega + \zeta) = \frac{P}{q}
\]

\[
P - t = s \quad \tan(\Omega) = \frac{t}{q}
\]

\[
\tan(\theta) = \frac{q}{\ell}
\]

These relations can be combined to give

\[
\tan(\Omega + \zeta) = \frac{\sin(\beta)}{\tan(\theta)} \tag{D.1}
\]

\[
\tan(\Omega) = \frac{\cos(\beta)}{\tan(\theta)}(\tan(\beta) - \tan(\delta)) \tag{D.2}
\]

\(\omega\) is a classic angle used in caret wing theory. It is the angle between the plane of the leading edges and the centerline of the compression surface. With \(\omega = \beta - \delta\), the second equation can be written as

\[
\tan \Omega = \frac{\sin \omega}{\tan \theta \cos \delta}. \tag{D.3}
\]

These angles and relations are used to define the cross-sectional shapes and in the 2-D flow approximation at the leading edges of caret and delta wings.
D.2 Approximating The Flow State on a Caret Wing Using 2-D Oblique Shock Theory

To get an analytical approximation of the flow state over a caret or delta wing it is common to view the wings as swept, infinite wedges. If we view the wing from a plane perpendicular to the leading edge line, represented by the unit vector $\ell$, (see Figures D.2 and D.3) the body looks like a wedge. Thus, the free stream flow can be projected into this plane and the equivalent wedge inflow and flow deflection angle can be found. Using these quantities the shocked flow state is found using 2-D oblique shock theory.

To find the velocity vector $\mathbf{V}_n$ and flow deflection angle $a_{nc}$ in a plane perpendicular to the leading edge line we need to find $\ell$ in the $x$-$y$-$z$ coordinate system. Letting $\ell = \ell_x \mathbf{\hat{x}} + \ell_y \mathbf{\hat{y}} + \ell_z \mathbf{\hat{z}}$ we solve for the components from the following four facts:

\begin{align*}
\ell \cdot \mathbf{\hat{y}} &= \cos(\Lambda) \quad (D.4) \\
\ell \cdot \mathbf{\hat{n}} &= 0 \quad (D.5) \\
|\ell| &= 1 \quad (D.6)
\end{align*}

\[\mathbf{\hat{n}} = \sin(\beta) \mathbf{\hat{x}} + \cos(\beta) \mathbf{\hat{z}}. \quad (D.7)\]

Refer to Figure D.2 for angle and vector definitions. The resulting $\ell$ is

\[\ell = \sin(\Lambda) \cos(\beta) \mathbf{\hat{x}} + \cos(\Lambda) \mathbf{\hat{y}} - \sin(\beta) \sin(\Lambda) \mathbf{\hat{z}}. \quad (D.8)\]
Figure D.2: Coordinate System Definition

Other facts, useful later in the analysis are $\hat{x}''$ and the direction cosine matrix. $\hat{x}''$ can be obtained from

$$\hat{x}'' \cdot \hat{y} = \sin(\Lambda),$$
$$\hat{x}'' \cdot \hat{n} = 0,$$
$$|\hat{x}''| = 1$$

And the direction cosine matrix for the conversion from the $x$-$y$-$z$ system to $x''$-$y''$-$z''$ system is:

$$\hat{x}'' = \cos(\Lambda) \cos(\beta) \hat{x} - \sin(\Lambda) \hat{y} - \sin(\beta) \cos(\Lambda) \hat{z} \quad (D.9)$$
$$\hat{y}'' = \sin(\Lambda) \cos(\beta) \hat{x} + \cos(\Lambda) \hat{y} - \sin(\beta) \sin(\Lambda) \hat{z} \quad (D.10)$$
$$\hat{z}'' = \sin(\beta) \hat{x} + \cos(\beta) \hat{z} \quad (D.11)$$

Delta Wing − Body
Lower Surface
or Caret Wing
Shock Plane

normal to body surface

$\beta$

$\Lambda$

$V$ velocity vector

vector in direction of leading edge
Plane Normal to Leading Edge Line Vector (l)

- $\alpha_{nc}$ is flow deflection angle for caret wing
- $\alpha_{n\delta}$ is flow deflection angle for delta wing
- $\Omega'$ is angle to lower body surface in the plane normal to the leading edge
  \[ \alpha_{nc} + \Omega' = \alpha_{n\delta} \]
- $V_n$ is velocity in plane normal to leading edge
- $b'$ is a unit vector along the caret body lower surface in this plane
- $s'$ is a unit vector along the shock in this plane
- $\beta' = (\alpha_{nc} + \Omega')$ is the shock angle as a result of $V_n$ and $\alpha_{nc}$

Figure D.3: Angle Definitions in Plane Normal to the Leading Edge
The free-stream velocity vector, $\mathbf{V}$, is assumed to be along $\mathbf{\hat{x}}$ so that

$$\mathbf{V} = |\mathbf{V}| \mathbf{\hat{x}}.$$

Projection into the perpendicular plane is achieved using

$$\mathbf{V}_n = \mathbf{V} - (\mathbf{V} \cdot \ell) \ell.$$  \hspace{1cm} (D.13)

If the body has a planar bottom (as in a delta wing) then the flow deflection angle would be $\alpha_{n\delta}$ which can be found from

$$\mathbf{V}_n \cdot \mathbf{\hat{n}} = |\mathbf{V}_n| \cos(90. - \alpha_{n\delta})$$  \hspace{1cm} (D.14)

From equations D.8 and D.12-D.14 $\mathbf{V}_n$ and $\alpha_{n\delta}$ are found to be

$$\mathbf{V}_n = |\mathbf{V}|((1 - \sin^2(\Lambda) \cos^2(\beta)) \mathbf{\hat{x}} - \sin(\Lambda) \cos(\Lambda) \cos(\beta) \mathbf{\hat{y}} + \sin^2(\Lambda) \sin(\beta) \cos(\beta) \mathbf{\hat{z}}$$

$$\alpha_{n\delta} = 90. - \arccos\left(\frac{|\mathbf{V}|}{|\mathbf{V}_n|} \sin(\beta)\right).$$  \hspace{1cm} (D.15)

We now know $\alpha_{n\delta}$. But if the body does not have a planar bottom the flow deflection is not $\alpha_{n\delta}$, but is a function of it. If the body is a caret wing the flow deflection angle seen by $\mathbf{V}_n$ is $\alpha_{nc}$. This angle is

$$\alpha_{nc} = \alpha_{n\delta} - \Omega'.$$  \hspace{1cm} (D.17)

From the body definition we know $\Omega$. $\Omega'$ is obtained from projecting the body’s lower surface vector $\mathbf{\hat{b}}$ into the plane perpendicular to the leading edge and finding

$$\Omega' = \arctan\left(\frac{|b_{x''}|}{|b_{x'}|}\right)$$  \hspace{1cm} (D.18)
From figure D.3 we see

\[ \mathbf{\hat{b}} = b_x \mathbf{\hat{x}} + b_y \mathbf{\hat{y}} + b_z \mathbf{\hat{z}} = -\cos(\Omega) \mathbf{\hat{y}} + \sin(\Omega) \mathbf{\hat{z}}. \]  

(D.19)

Or it can be rewritten in the "coordinates as

\[ \mathbf{\hat{b}} = b_{x''} \mathbf{\hat{x}''} + b_{y''} \mathbf{\hat{y}''} + b_{z''} \mathbf{\hat{z}''}. \]  

(D.20)

The direction cosine matrix shows that

\[ b_{x''} = \cos(\Lambda) \cos(\beta) b_x - \sin(\Lambda) b_y - \sin(\beta) \cos(\Lambda) b_z \]  

(D.21)

\[ b_{y''} = \sin(\Lambda) \cos(\beta) b_x + \cos(\Lambda) b_y - \sin(\beta) \sin(\Lambda) b_z \]  

(D.22)

\[ b_{z''} = \sin(\beta) b_x + \cos(\beta) b_z. \]  

(D.23)

Substituting in \( b_x, b_y \) and \( b_z \)

\[ \mathbf{\hat{b}} = (\cos(\Omega) \sin(\Lambda) - \sin(\beta) \cos(\Lambda) \sin(\Omega)) \mathbf{\hat{x}''} \]  

(D.24)

\[ + (-\sin(\Omega) \sin(\beta) \sin(\Lambda) - \cos(\Omega) \cos(\Lambda)) \mathbf{\hat{y}''} \]  

(D.25)

\[ + (\sin(\Omega) \cos(\beta)) \mathbf{\hat{z}''} \]  

(D.26)

Therefore equation D.18 becomes

\[ \Omega' = \arctan\left(\frac{\left|\cos(\beta) \sin(\Omega)\right|}{\left|\sin(\Lambda) \cos(\Omega) - \sin(\beta) \cos(\Lambda) \sin(\Omega)\right|}\right). \]  

(D.27)

Now evaluating D.27 and D.17 we can find \( \alpha_{nc} \) and thus proceed to find approximate flow characteristics from 2-D oblique shock theory including the shock angle, \( \beta' \), as seen in the perpendicular leading edge plane.
To test the correctness of the shock angle we need to know the angle in the cross-sectional plane. Let \( s \) be a unit vector representing the shock viewed in the cross-sectional plane. Then

\[
s = s_x \hat{x} + s_y \hat{y} + s_z \hat{z} = s_x'' \hat{x}'' + s_y'' \hat{y}'' + s_z'' \hat{z}''
\]

(D.28)

and

\[
s_x = 0
\]

(D.29)

\[|s| = 1.
\]

(D.30)

Then there are two equations and two unknowns for finding the coefficients of \( s \). These can be used to find the angle of the shock above the \( y \) axis in the cross-sectional view. The result is

\[
\tan(\Omega' - \rho') = \frac{s_z''}{s_x''}
\]

(D.31)

where

\[
\rho' = \beta' - \alpha_{ne}.
\]

(D.32)

In the double prime system, with \( s_x = 0 \),

\[
s_x'' = -\sin(\Lambda)s_y - \sin(\beta)\cos(\Lambda)s_z
\]

(D.33)

\[
s_y'' = \cos(\Lambda)s_y - \sin(\beta)\sin(\Lambda)s_z
\]

(D.34)

\[
s_z'' = \cos(\beta)s_z.
\]

(D.35)

Then

\[
\tan(\Omega' - \rho') = \frac{\cos(\beta)s_z}{-\sin(\Lambda)s_y - \sin(\beta)\cos(\Lambda)s_z}
\]

(D.36)
and $s_z$ is found from

$$s_z^2 = \frac{1}{1 + \left(\frac{\cos(\beta)}{q \sin(\lambda)}\right)^2 + \left(\frac{2 \cos(\beta) \sin(\beta)}{q \sin(\lambda) \tan(\lambda)} + \frac{\sin(\beta)}{\tan(\lambda)}\right)^2}, \quad (D.37)$$

where

$$q = \tan(\Omega' - \rho') \quad (D.38)$$

and the root which makes $|s_z| < 1$ and $s_y < 0$ is chosen.

### D.2.1 Computing Using a Coordinate System with X-axis Along Bottom Ridge Line

The cross-sectional plane is different in the two coordinates systems. Thus, the angle between horizontal and the body’s lower surface and the angle between the upper and lower body surfaces will be different in the two coordinate systems. In the previous system these angles were defined as $\Omega$ and $\Omega + \zeta$, in this system they will be called $\Omega_r$ and $(\Omega + \zeta)_r$. Apart from these difference the pictures are the same as seen previously.

If we use the same methods as above, the body angle equations can be derived for this new coordinate system. The resulting body geometry relations are:

$$\tan(\Omega_r) = \frac{\sin(\omega)}{\tan(\theta)} \quad (D.39)$$

and

$$\tan((\Omega + \zeta)_r) = \frac{\tan(\omega) + \tan(\delta) \cos(\omega)}{\tan(\theta)}. \quad (D.40)$$

The normal vector to the leading edge remains the same:

$$\mathbf{n} = \sin(\beta) \mathbf{\hat{x}} + \cos(\beta) \mathbf{\hat{z}}. \quad (D.41)$$
The free-stream velocity vector becomes

\[ \mathbf{V} = |\mathbf{V}|(\cos(\delta)\hat{x} + \sin(\delta))\hat{y} \]  

(D.42)

The unit vector along the leading edge changes to

\[ \ell = \sin(\Lambda)\cos(\omega)\hat{x} + \cos(\Lambda)\hat{y} - \sin(\omega)\sin(\Lambda)\hat{z}. \]  

(D.43)

The direction cosine matrix for the conversion from the x-y-z system to x”-y”-z” system is:

\[ \hat{x}” = \cos(\Lambda)\cos(\omega)\hat{x} - \sin(\Lambda)\hat{y} - \sin(\omega)\cos(\Lambda)\hat{z} \]  

(D.44)

\[ \hat{y}” = \sin(\Lambda)\cos(\omega)\hat{x} + \cos(\Lambda)\hat{y} - \sin(\omega)\sin(\Lambda)\hat{z} \]  

(D.45)

\[ \hat{z}” = \sin(\omega)\hat{x} + \cos(\omega)\hat{z}. \]  

(D.46)

The angles in the plane normal to the leading edge are the same since this plane is the same in both coordinate systems. The expressions for obtaining the quantities in this plane are different. Following are some useful expressions.

\[ \beta = \delta + \omega \]  

(D.47)

\[ \tan(\alpha_{ns}) = \frac{\tan(\delta + \omega)}{\cos(\Lambda)} \]  

(D.48)

\[ \sin(\Omega’’) = \frac{\tan(\omega)}{[\tan^2(\omega) + \cos^2(\Lambda)]^{\frac{1}{2}}} \]  

(D.49)

\[ \sin(\alpha_{nc}) = \frac{(\tan(\beta) - \tan(\omega)\cos(\Lambda))}{[(\tan^2(\omega) + \cos^2(\Lambda))(\tan^2(\beta) + \cos^2(\Lambda))]^{\frac{1}{2}}} \]  

(D.50)

\[ \mathbf{V}_n = \mathbf{V}[1 - \sin^2(\Lambda)\cos^2(\delta + \omega)]^{\frac{1}{4}} \]  

(D.51)
D.2.2 Calculating The Shocked State

For an ideal gas the state after the shock can be calculated easily using textbook equations. The following were taken from [1]. Letting \( \delta \) be the relevant flow deflection angle the shock wave angle, \( \eta \) is found from the implicit equation

\[
\cot(\delta) = \tan(\eta)\left[\frac{6M_1^2}{5(M_1^2 \sin^2(\eta) - 1)} - 1\right]. \tag{D.52}
\]

The shock wave angle can then be used to calculate the shocked flow state as follows:

\[
\frac{P_2}{P_1} = \frac{7M_1^2 \sin^2(\eta) - 1}{6} \tag{D.53}
\]

\[
\frac{\rho_2}{\rho_1} = \frac{6M_1^2 \sin^2(\eta)}{M_1^2 \sin^2(\eta) + 5} \tag{D.54}
\]

\[
a_2 = \sqrt{\frac{\gamma P_2}{\rho_2}} \tag{D.55}
\]

\[
u_2 = |V_1||1 - \frac{5(M_1^2 \sin^2(\eta) - 1)}{6M_1^2}| \tag{D.56}
\]

\[
v_2 = |V_1|[\cot(\eta)\frac{5(M_1^2 \sin^2(\eta) - 1)}{6M_1^2}] \tag{D.57}
\]

\[
\frac{|V_{2p}|}{|V_{n2}|} = 1 - \frac{4(M_1^2 \sin^2(\eta) - 1)(\gamma M_1^2 \sin^2(\eta) + 1)}{(\gamma + 1)^2 M_1^4 \sin^2(\eta)}. \tag{D.58}
\]
APPENDIX E

Range of Influence of Leading Edge in Post-Shock Flow

In this work I found it very useful to compute the angle of the streamline just after
the leading edge shock and the corresponding flow characteristics. This information
provided me with a good prediction to the type of flow I would see for a given
case. Following are the equations I needed to compute the streamline angle, the
characteristic angles, and the state of the flow under caret wings.

Equations will be defined using a coordinate system with \( \hat{y}'' \) pointing downstream
along the leading edge and \( \hat{x}'' \) and \( \hat{z}'' \) in a plane perpendicular to the leading. This
is the double prime system as discussed in D.2.

A vector along the lower ridge line, \( \hat{x} \), in the '' system is

\[
\hat{x} = \cos(\Lambda)\cos(\omega)\hat{x}'' + \sin(\Lambda)\cos(\omega)\hat{y}'' + \sin(\omega)\hat{z}'
\]  

(E.1)

Let \( \hat{b} \), defined as

\[
\hat{b} = \cos(\Omega')\hat{x}'' + \sin(\Omega')\hat{z}'', 
\]  

(E.2)
be a vector in both the plane perpendicular to the leading edge and in the wing’s lower surface, see Figure D.3 in Appendix D. Then the post-shock velocity vector which must lie parallel to the bottom of the wing can be defined as

$$V_2 = V_{2\parallel} \hat{y} + V_{2\perp} \hat{b}. \tag{E.3}$$

$V_{\perp}$ and $V_{\parallel}$ are the components of the post-shock velocity vector along and perpendicular to the leading edge vector $\ell$, as discussed in D.2. The subscripts 1 and 2 denote conditions before and after the flow encounters the shock.

$$V_{2\parallel} = (V_1 \cdot \ell) \ell = V_{1\parallel} \tag{E.4}$$

$$V_{1\perp} = V_1 - V_{1\parallel} = V_n \tag{E.5}$$

$$V_1 = |V_1| (\cos (\delta) \hat{x} + \sin (\delta) \hat{y}) \tag{E.6}$$

$$\ell = \sin (\Lambda) \cos (\omega) \hat{x} + \cos (\Lambda) \hat{y} - \sin (\omega) \sin (\Lambda) \hat{z} \tag{E.7}$$

$V_{2\perp}$ is calculated from the shock jump equations. $V_{2n}$ is found from

$$\frac{|V_{2n}^2|}{|V_{n}^2|} = 1 - \frac{4(\gamma M_1^2 \sin^2 (\alpha_n) - 1)(\gamma M_1^2 \sin^2 (\alpha_n) + 1)}{(\gamma + 1)^2 M_1^2 \sin^2 (\alpha_n)} . \tag{E.8}$$

When E.2 is substituted into E.3 then

$$V_2 = V_{2\perp} \cos (\Omega') \hat{x} + V_{2\parallel} \hat{y} + V_{2\perp} \sin (\Omega') \hat{z} \tag{E.9}.$$  

If we define the angle $\zeta$ as the angle of the post-shock streamline to the lower ridge line then

$$V_2 \cdot \hat{x} = |V_2| \cos (\zeta). \tag{E.10}$$
Which means $\zeta$ is found from

$$|V_2| \cos(\zeta) = V_{2\perp} \cos(\Omega') \cos(\Lambda) \cos(\nu) + V_{2\parallel} \sin(\Lambda) \cos(\nu) + V_{2\perp} \sin(\nu) \sin(\Omega').$$

(E.11)

The Mach cone angle, $\mu$, is found from equation E.12 and determines the range of influence of a point on the leading edge.

$$\sin(\mu) = \frac{1}{M_2}.$$  

(E.12)

$M_2$ is the post-shock Mach number found from the jump equations.

If $V_2 \cdot \ell > \hat{\ell} \cdot \ell$ then the streamline originating at the nose would cross the centerline if there was not another wing surface to interfere with its original direction.
APPENDIX F

Calculation of Expected Uniform Flow Region on Delta Wing

When equations D.15 and D.16 are reformulated in the double prime coordinate system equation F.1 is obtained. This is a simpler form for the normal velocity component.

\[ V_n = |V|(\cos(\Lambda)\cos(\beta)\hat{\alpha}'' + \sin(\beta)\hat{\beta}'') \]  \hspace{1cm} (F.1)

The angle of attack of this velocity component onto the leading edge is

\[ \alpha_n = \frac{\Pi}{2} - \cos^{-1}\left(\frac{|V|}{|V_n|}\sin(\beta)\right). \] \hspace{1cm} (F.2)

This normal velocity component and flow deflection angle are used to compute the flow state in the uniform region of a delta wing flow. The normal velocity component \( V_n \) becomes the incoming Mach number, \( M_1 \), on the equivalent yawed wedge. The equivalent flow deflection angle is \( \alpha_n \). These values are used in equations D.52 through D.58 to compute the shocked state within the uniform region under a delta wing.
Remember that the shocked state velocity obtained from the 2-D wedge equations is only the component normal to the shock, \( V_n = V_{2\perp} \). The parallel component, \( V_{2\parallel} \) is equal to the pre-shock parallel component, \( V_{1\parallel} \). This component is found from

\[
V_{1\parallel} = \sqrt{|V|^2 - |V_n|^2}
\]  

(F.3)
and

\[
V_{2\perp} = \sqrt{u^2 + v^2},
\]  

(F.4)

where \( u \) and \( v \) are the velocity components found in equations D.58. Then

\[
|V_2| = \sqrt{V_{2\parallel}^2 + V_{2\perp}^2}.
\]  

(F.5)

The angle of the velocity vector in the uniform region with respect to the leading edge is

\[
\chi = \arctan\left(\frac{V_{2\perp}}{V_{2\parallel}}\right).
\]  

(F.6)

The Mach cone angle about this velocity vector is

\[
\mu = \arcsin\left(\frac{a_2}{|V_2|}\right).
\]  

(F.7)

So, the angle from the center line to the edge of the uniform region is \( \mu + \zeta \), where, from Figure F.1, \( \zeta = \theta - \chi \).

Another measure of the accuracy of the solution is the slope of the pressure curve at the transition between the uniform and non-uniform flows. This equation was formulated by Roe in \([70]\).

\[
\frac{dp}{d\theta} = \frac{\rho_\infty u_\infty^2}{\gamma + 1} \frac{\sqrt{M_\infty^2 - 1}}{M_\infty^2}
\]  

(F.8)
Figure F.1: Planform View of the Under-Surface of A Delta Wing Showing Angle Definitions
APPENDIX G

Calculation of the Limits Within Parameter Space

For Body Definition

This Appendix will compute the domain in parameter space from which we can choose quartic shapes for the waverider bottom surface.

The parameters in the quartic equation are constrained by the angle of the underside surface leading edge (to maintain shock attachment) and by the angle of the caret top (so that the bottom surface does not cross the top). The quartic bottom is defined as

\[ z = ay^4 + by^2 + c. \quad (G.1) \]

The slope at the leading edge of the bottom surface \((y_{le} < 0)\) is

\[ z' = 4ay^3 + 2by. \quad (G.2) \]

The two free parameters are \(a\) and \(b\). \(c\) is fixed by matching the top and bottom surfaces at the leading edge.

The angle of the upper surface with respect to the horizontal is the angle \((\Omega + \zeta)\) as defined in the body geometry in Appendix D. So, one limit on the parameters of
the quartic \((a\) and \(b)\) is
\[
z'_{le} > \tan (\Omega + \zeta) \quad (G.3)
\]
or
\[
(4ay^3 + 2by)_{le} > \tan (\Omega + \zeta). \quad (G.4)
\]
This leads to an expression for \(b\) given \(a\):
\[
b > \frac{\tan (\Omega + \zeta)}{2y_{le}} - 2ay_{le}^2. \quad (G.5)
\]

The second limit on the lower surface equation is the shock detachment limit. This limit requires that the deflection angle seen by the flow (again see Appendix D) be less than the maximum deflection angle, \(\delta_{\text{max}}\),
\[
\alpha_n\delta - z'_{le} \leq \delta_{\text{max}}. \quad (G.6)
\]
Approximating \(z'_{le}\) in a plane perpendicular to the leading edge as \(z'_{le}\) in a plane perpendicular to the marching direction this becomes
\[
z'_{le} > \alpha_n\delta - \delta_{\text{max}}. \quad (G.7)
\]
Substituting the above expressions and simplifying this becomes
\[
b < \frac{\alpha_n\delta - \delta_{\text{max}} - 4a y_{le}^3}{2y_{le}}. \quad (G.8)
\]
Remember that \(y_{le} < 0\) in this case and that to have a concave slope at the leading edge \(a\) must be less than zero.

The bounded region for the design caret wing used in this work is shown in Figure 6.3. The equations of the bounding lines are
\[
b < -.5a - .538 \quad (G.9)
\]
and

\[ b > -0.5a - 1.0863 \]  \hspace{1cm} (G.10)

To have a systematic way of choosing test cases for this work I choose a series of lines parallel to those derived as boundaries of the allowable domain. I then used the \( y \)-intercepts of these lines as a free parameter. Then, after choosing \( a \), and a \( y \)-intercept, \( b_j \), \( b \) is found from

\[ b = -2a y^2 + b_j. \]  \hspace{1cm} (G.11)
A Relation Between Lift and Volume

It can easily be shown that lift is proportional to volume for caret and delta wings. However, for an arbitrarily shaped under-surface a clean expression cannot be obtained. Yet, a graph of the test cases seems to imply that a linear relation still exists.

To derive a relation for a caret or delta wing let \( \phi \) be the compliment to the angle \((\Omega + \zeta)\) in Figure D.1. Then the lift, \( L \), for a caret and delta wing can be expressed as

\[
L = q_\infty C_{pw} S_p, \tag{H.1}
\]

\( S_p \) is the planform area,

\[
S_p = l^2 \tan \beta_w \tan \phi, \tag{H.2}
\]

\( V \) is the volume

\[
V = \frac{l}{3} A_b, \tag{H.3}
\]

\( C_{pw} \) is the coefficient of pressure at the wall, \( q_\infty \) is the dynamic pressure at free-stream conditions, \( l \) is the body length and the angles are defined in Figure D.1.
The base areas are found from

\[ A_{\text{caret}} = S_p \tan \delta_{\text{caret}} \quad (H.4) \]

and

\[ A_{\delta} = S_p l^2 \tan^2 \delta_{\delta} \tan \phi. \quad (H.5) \]

\( \delta_{\delta} \) and \( \delta_{\text{caret}} \) are the flow deflection angles seen at the center-line of the bodies.

If we let \( \delta \) represent the relevant deflection angle of the body (caret or delta wing), lift can be expressed in terms of the volume as

\[ L = \frac{3q_\infty C_{pw} V}{l \tan \delta}. \quad (H.6) \]

If we assume that the flow deflection angle remains constant for a family of bodies, then \( C_{pw} \) and \( \beta \), the shock wave angle, are constants. Thus \( L \) is proportional to \( V \).

However, the case study of this work involved changing the under-surface shapes in such a way that \( \delta \) is constant at the leading edges but not at the center-line. If \( \delta \) does not remain constant then \( C_{pw} = f(\delta, M_\infty) \) and \( V = f(l, \delta, \phi) \). Because of this we cannot discern from equation H.6 a simple relation between \( L \) and \( V \).

It is very interesting, however, that the quartic-bottom cases in this study show a proportional relation between \( L \) and \( V \) as seen in Figure 6.9. Notice, the line does not go through zero since a flat plate still generates lift.

\[ ^1 \text{Roe suggests [72] that using Thin Shock Layer Theory a relation, or at least an approximation, may be obtained.} \]
BIBLIOGRAPHY


[65] H. Rieger, “Solution of some 3-d viscous and inviscid supersonic flow problems by finite-volume space-marching schemes.”


ABSTRACT

TOWARD THE DIRECT DESIGN OF WAVERIDER AIRFRAMES

by

Dawn Danielle Kinsey

Chairperson: Philip L. Roe

A waverider is a hypersonic vehicle that keeps the bow shock attached to its leading edges. This class of vehicles originated in the late 1950’s as a theoretical construct. Since the late 1980’s there has been a resurgence of waverider research and an explosion of waverider applications.

Waveriders have traditionally been and are still primarily designed using inverse methods, wherein the flowfield is known and the body surfaces are formed by collecting streamlines or streamsurfaces out of the flowfield. In the work presented here, the advantages of using a more conventional direct-design method for waveriders are investigated. The tools necessary to construct a practical Computational Fluid Dynamics design code are evaluated. As a start, a 3-D steady Euler code is built. The code uses an adaptive Cartesian mesh which allows arbitrary body shapes. The strengths and weaknesses of the code and direct-design method will be discussed.

Several test cases and a simple design study are performed. These show that off-design conditions are easily simulated. A case study reveals that small changes in the body shape can significantly impact the lift-to-drag ratio. This points to an
advantage of the direct-design method. Because the direct method does not constrain
the design space, it can be used to look for optimum shapes between those found
using inverse methods.