DEVELOPMENT OF AN INTERACTIVE WAVE DRAG CAPABILITY FOR THE OPENVSP PARAMETRIC GEOMETRY TOOL

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Aerospace Engineering

by
Michael J. Waddington
July 2015

COMMITTEE MEMBERSHIP

TITLE:	Development of an Interactive Wave Drag Capability for the OpenVSP Parametric Geometry Tool
AUTHOR:	Michael J. Waddington
DATE SUBMITTED:	July 2015
COMMITTEE CHAIR:	Robert McDonald, Ph.D. Professor of Aerospace Engineering
COMMITTEE MEMBER:	David Marshall, Ph.D. Professor of Aerospace Engineering
COMMITTEE MEMBER:	Sriram Rallabhandi, Ph.D. Associate Research Fellow, NIA
COMMITTEE MEMBER:	Benjamin Schiltgen Aerospace Engineer, Empirical Systems Aerospace

ABSTRACT

Development of an Interactive Wave Drag Capability for the OpenVSP Parametric Geometry Tool

Michael J. Waddington

Minimizing wave drag is critical to successful and efficient transonic and supersonic flight. Area-ruling is the process of managing the cross-sectional area of an aircraft to lessen the wave drag experienced in flight. Effectively calculating the necessary areas for a given aircraft can be difficult, and existing tools for conducting a wave drag analysis often carry limitations in both functionality and availability.

In this work, the author utilized an existing parametric geometry tool named OpenVSP to create an interactive design tool for approximating zero-lift wave drag. Here, the wave drag calculation methodology used in industry for decades is combined with the powerful geometry engine of OpenVSP, which was recently heavily upgraded at the start of 2015. Various visual aids allow users of this OpenVSP wave drag tool to interact with area and wave drag results and develop intuition for supersonic aircraft design using the area rule approach. OpenVSP allows geometry changes to be made quickly, enabling rapid reanalysis by the wave drag tool for expeditious comparison of results across the design space.

This wave drag tool will be readily available once it is added to the public distribution of OpenVSP. Developed by NASA, OpenVSP is an open source tool released under a NASA Open Source Agreement and is employed in industries both public and private. Exposure of the wave drag tool to the OpenVSP community will be achieved via version update.

ACKNOWLEDGMENTS

I would like to thank my numerous AERO friends, so uniquely bonded by the trials and tribulations of our chosen major, for being a source of support, advice, and occasional comic relief. And to my fellow Engineering Student Council people: thank you for the years of unforgettable experiences and opportunities. It's been epic.

Of course, I would also like to thank my committee. Dr. Marshall, it has been an honor learning from you, in topics both aeronautical and ethical. Your graduate CFD class remains one of my favorites, and your experience with OpenVSP has been a valuable contribution to my graduate work. Dr. Rallabhandi, your experience with the subject matter to which my topic applies, along with your coded implementation of the wave drag evaluation, has been undoubtedly valuable to me, and I thank you. Mr. Schiltgen, my time with you at ESAero provided me with some of the greatest experiences and memories I've had. Thank you for helping me become a business-oriented engineer. And Dr. McDonald, I can't thank you enough. You took it upon yourself to take me up as your grad student, survive LENR with me, work with me through various topic ideas, and be immensely supportive throughout my graduate work. You have always been there with just the right question, just the right answer, just the right anecdote, and occasionally tri-tip. Thank you for the journey.

This would not be complete without thanking my parents, as well as my grandparents. I may never know the full extent to which you made sacrifices for me, but I do know that it was your support (financial and otherwise) that got me through seven (yes, I know) years of college. Thank you so very much.

To my fiancée: Thank you for 7.5 years of unwavering love and support, and for quite possibly being more excited about my graduation than me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

NOMENCLATURE

θ	=	Mach plane rotation about aircraft axis, in radians
κ	=	$\cos^{-1}(1-2x)$
λ	=	Constant multiplier
β	=	Function of Mach number
ρ	=	Density
A	=	Cross-sectional area, $S(x_i)$
a	=	Fourier series coefficient
API	=	Application program interface
B	=	$S(1)$
C	=	Coefficient
CFD	=	Computational fluid dynamics
D	=	Wave drag
I	=	Wave drag integral
L	=	Length
M	=	Mach number
N	=	$S(0)$
n	=	Number of points $x_i, 0 < x_i < 1$
$NACA$	=	National Advisory Committee for Aeronautics
q	=	Dynamic pressure
r	=	Radius
S	=	Cross-sectional area distribution function
U	=	Velocity
V	=	Volume
VSP	=	Vehicle Sketch Pad
x	=	Integral value, or Cartesian coordinate
y	=	Integral value, or Cartesian coordinate

Subscripts & Superscripts

∞	=	Freestream
0	=	Zero-lift condition
b	=	Base
D	=	Drag
i	=	Inner
max	=	Maximum
o	=	Outer
$wave$	=	Pertaining to wave drag

1 INTRODUCTION

Once man took to the skies with heavier-than-air technology, the natural need and desire for increased capabilities fueled the eventual accomplishment of manned supersonic flight. On October 14, 1947, a pilot named Chuck Yeager piloted an aircraft across the so-called "sound barrier" and ushered in the era in which man could travel faster than the speed of sound. Despite this historic success, a lack of understanding and test capabilities for transonic aerodynamics continued to pose serious problems for aeronautical engineers of the time. This can be evidenced by the shape of Yeager's Bell X-1 aircraft, which resembled that of a .50 caliber machine-gun bullet— ballisticians already had a grasp on the aerodynamics of these bullets at supersonic speeds, and Bell Aircraft capitalized on this by lending their X-1 a similar shape [1].

While this approach to shape management was sufficient for the rocket-powered X-1, the drag rise at high transonic speeds was problematic for the design of supersonic jet aircraft desired by the U.S. military [2]. This rise can be attributed to the presence of wave drag, which results from the production of shock waves as airflow over the body reaches supersonic speeds [3]. The graphic in Figure 1.1 illustrates the dramatic increase in total drag caused by the presence of wave drag. The location at which this rise begins is known as the drag divergence Mach number [4].

In the years following Yeager's historic flight, a NACA Langley Aeronautical Laboratory engineer by the name of Richard Whitcomb produced a set of revolutionary work that provided the necessary solution for overcoming wave drag. Results from tests Whitcomb ran in the Langley transonic Eight-Foot High-Speed Tunnel in 1951 revealed stronger

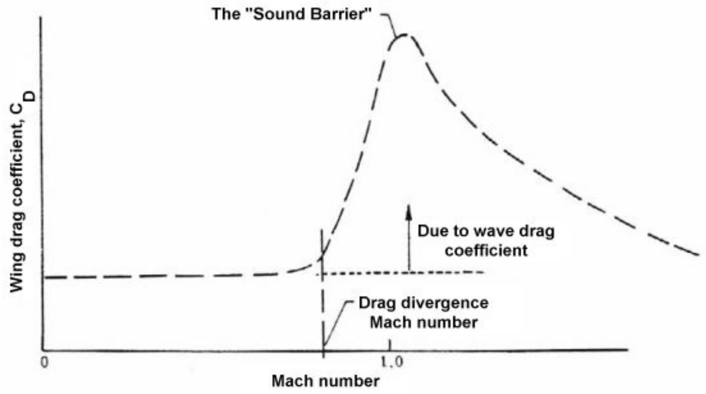
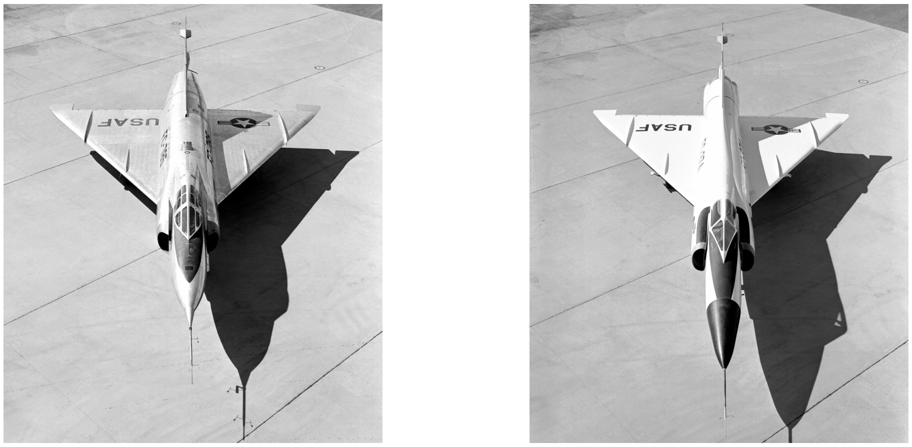


Figure 1.1: Illustration of transonic drag rise [5].

shocks than were expected. Shortly thereafter, Whitcomb attended a presentation by Adolf Busemann in which the famed aeronautical engineer likened the process of accommodating sonic flow to “pipefitting”. Here, the metaphorical pipes describe how compressibility effects lead to the stream tubes of air maintaining constant diameter. In the subsonic region, contrarily, stream tube diameter varies with velocity, becoming thinner as the freestream velocity increases [2].

Whitcomb’s profound discovery was that this behavior of the stream tubes meant changes in the cross-sectional area of any *part* of an aircraft would affect the *entire* aircraft, omnidirectionally. Instead of managing the shape of each aircraft component separately, Whitcomb noticed that it was the combined cross-sectional area of all the aircraft components that would drive the wave drag. Mitigating wave drag would be achieved by ensuring a smooth distribution of total cross-sectional area from nose to tail. This management of cross-sectional area became known as the Whitcomb area rule, and it earned him the Collier Trophy in 1954 [2].

An early example of the area rule in use was during the faltering Convair F-102 program. In 1952, the YF-102, intended to be supersonic, remained unable to exceed Mach 0.9. Facing potential cancellation of their contract, Convair engineers met with Langley researchers and ran tests in the transonic tunnel. These tests revealed the YF-102 was violating the newly-discovered area rule’s stipulation that the total cross-sectional area dis-

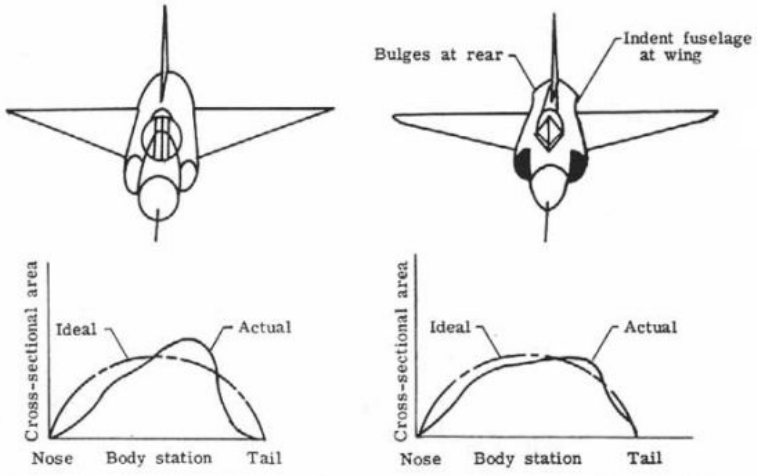


(a) The F-102 with straight fuselage.

(b) The F-102A with area-ruled fuselage, necked down in the region of the wing.

Figure 1.2: Comparing the YF-102 and the Whitcomb-modified YF-102A [7].

tribution be smooth for successful supersonic flight. With Whitcomb assisting in redrawing the lines, the layout of the YF-102A was completed in July 1953. With a fuselage that was thinner where the wing was thickest, among various other adjustments, the YF-102A possessed a cross-sectional area distribution much more in line with the area rule. When flight tested, the YF-102A crossed Mach 1 while in a climb [6]. Photographs of the YF-102 and Whitcomb-modified YF-102A are shown in Figure 1.2. Their respective cross-sectional area distributions can be seen in Figure 1.3.



(a) YF-102A before area ruling.

(b) F-102A after area ruling.

Figure 1.3: Cross-sectional area distributions of the YF-102 and YF-102A [8].

Advancements in technology soon permitted area ruling and wave drag analyses to be conducted by computers. Roy Harris of NASA Langley Research Center published such a computer tool, and it became known simply as the Harris Wave Drag tool [9]. This routine, described in Harris's 1964 paper (Ref. [9]), accepted columnized geometry data from punched cards and read it onto magnetic tape. The Fortran source code, published in the appendix of Ref. [9], has since been modernized for use with text files and can be executed by any knowledgeable user with a Fortran compiler and a command line [10]. AWAVE is a streamlined implementation of the Harris Wave Drag tool from L.A. McCullers that has also been used at NASA Langely Research Center [11].

In these tools, limitations on input geometry introduce inaccuracies. As examples, wings may be defined by no more than ten airfoil sections, which must be aligned streamwise and are then connected linearly; the vertical and horizontal tails must have constant, symmetric airfoils. Additionally, component intersections are not handled by these routines, meaning such intersections as the fuselage and wing are not properly represented unless both components are straight-sided and in the streamwise direction. Fuselages that are in any way curved in the span of the wing root chord will result in extra or missing cross-sectional area when the tool is executed. An $X-Z$ plane symmetry assumption also prohibits such concepts as oblique wings.

OpenVSP provides vast flexibility in defining the geometry of an aircraft. Its powerful geometry engine lends ease to calculating various areas and volumes, earning praise from the OpenVSP user community. The author sought to take advantage of OpenVSP by utilizing its existing ability to create and analyze geometry. By introducing to OpenVSP the means to calculate zero-lift wave drag using geometry, the author implemented a more elegant approach to cross-sectional area analysis and subsequent wave drag calculation.

In this work, the geometry-based wave drag solution is conducted in the same tool that generated the geometry in the first place. Whether the geometry was built from components in OpenVSP or created from an imported geometry file, OpenVSP contains the structure and user interfaces to fully manage and manipulate the geometry. Housing the wave drag capability in the same tool as the geometry engine provided a unique opportunity for greater visual understanding and speed of user interaction in a wave drag analysis. Joining rapid modeling with interactive wave drag mitigation was the goal that guided this work, of which the discussion and results are the purpose of this document.

2 OPENVSP ESSENTIALS

OpenVSP is a parametric geometry tool developed for NASA that permits designers to quickly create three-dimensional aircraft models that may be used in a variety of analyses [12]. It was released as open source under a NASA Open Source Agreement (Version 1.3) in 2012. OpenVSP is cross-platform and is available as a downloadable executable for Windows or Macintosh systems from the OpenVSP website and as source files from the OpenVSP GitHub website for those opting to compile OpenVSP themselves [13, 14].

A selection of pre-defined components in OpenVSP, such as wings and fuselages, permit rapid assembly of geometries defined by aircraft parameters. This parametric definition of geometry is an asset to OpenVSP and its usefulness to aerospace applications. Here, dimensions and shape definitions exist in parameters familiar to aircraft designers. For example, wing components may be described directly by wingspan, chord, thickness to chord ratio, sweep, and other such quantities. Such intuitiveness lends speed and ease-of-use to modeling aircraft. OpenVSP also operates in a "unitless" environment: users build in whichever unit of measure they choose. If a length is set to "10", this can be 10 inches, 10 centimeters, 10 yards, etc.— the only implied requirement being that the user is consistent with their units of measure.

Recent expansions of OpenVSP now allow users to script their own custom components [15]. Using a broad selection of exposed functions, users can create components and build the GUIs with which to interact with them. This added feature adds to the flexibility of OpenVSP and further facilitates rapid construction of evermore creative and representative models.

2.1 Structure of OpenVSP

OpenVSP employs a useful approach to its code structure in which the underlying geometry and analysis tools do not depend on the existence of graphics. This independence is imperceptible to the user when using the full OpenVSP package. The independence of the geometry and analysis tools from the graphics allows OpenVSP's capabilities to be used for instances in which graphics are undesired or prohibited, such as in batch analyses or on supercomputers with strict usage limitations. For these cases, OpenVSP may be compiled in a "no-graphics" format and accessed via an API, through which models may be created, opened, modified, and analyzed from the command line. This no-graphics version of OpenVSP is distributed as an executable along with the standard OpenVSP executable, and may also be compiled from the OpenVSP source.

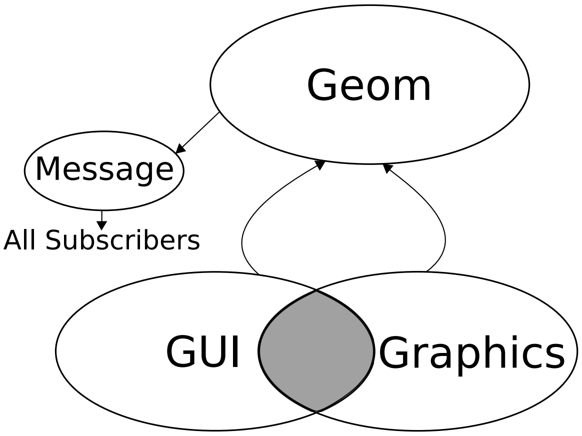


Figure 2.1: Simplified OpenVSP internal structure diagram.

The structure of OpenVSP is vastly simplified for purposes of demonstration in Figure 2.1. As noted above, graphics and user interfaces work together and depend on the geometry engine, but the geometry engine stands alone. Here, a unidirectional message passing system allows the geometry engine to pass information to graphics and user interfaces without depending on knowledge of their existence. The message system can be likened to a car radio in which the system passively listens for a radio signal, but does not require one in order to function. Likewise, the radio station broadcasts the radio signal without knowledge of any listeners.

2.2 Scripting with OpenVSP

OpenVSP employs AngelScript as the scripting language with which to communicate with OpenVSP. AngelScript is cross-platform and open source with a C/C++ syntax [16].

Using scripts, users may interact with OpenVSP by creating and editing geometries, as well as executing a number of functions. Any ensuing results from analyses are pushed to a results manager which may be accessed in the script. Thus, results may be retrieved and formatted however the user desires [17].

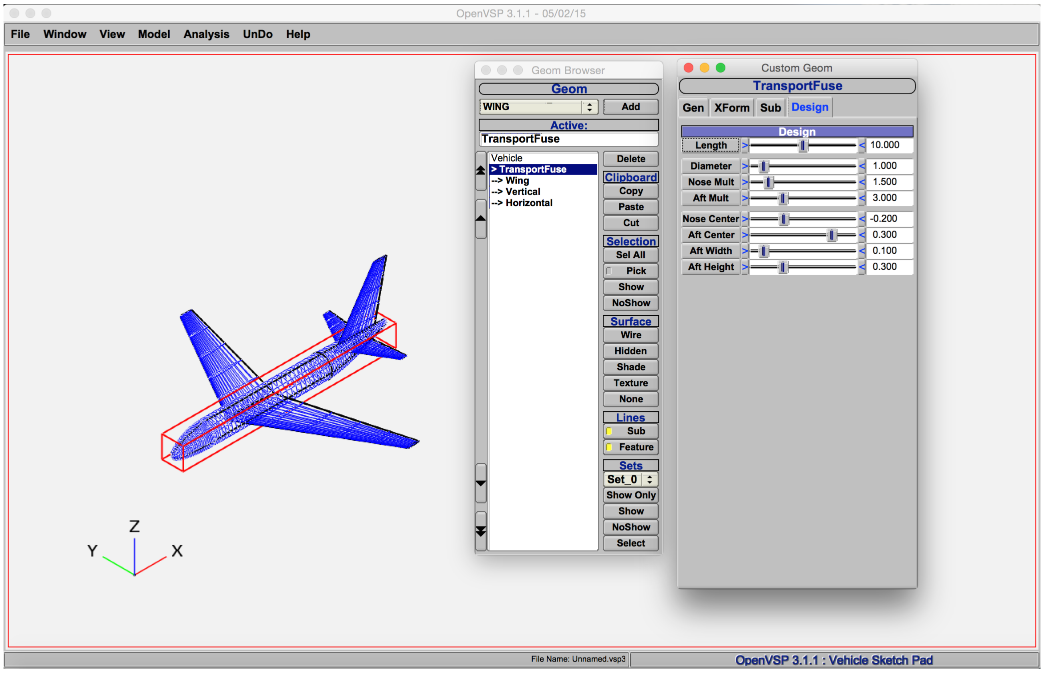
Interaction with OpenVSP through the API via scripting takes advantage of the OpenVSP structure outlined above by foregoing the GUI and graphics. Scripts can be set up to conduct a batch mode analysis for optimization or design space exploration.

Development of the wave drag tool remained conscious of the division between geometry engine and graphics, as well as the foundation for API access. As such, the wave drag tool may be accessible from the API in the same manner as other OpenVSP analyses.

2.3 Interacting with OpenVSP

OpenVSP aids the conceptual design process by enabling mid-fidelity analysis of three-dimensional aircraft models. Higher fidelity analyses, such as CFD, are valuable to the design process, but inappropriate at the conceptual level when the outer mold lines may still be significantly altered. The level of fidelity in the analysis should be comparable to the level of fidelity in the model. As such, designers may be relegated to investing time and effort in a detailed model without sufficient confidence in the geometry in order to run a high-fidelity analysis— one that may very well alter major aspects of the initial design.

OpenVSP fills a niche in the conceptual design process by providing designers with tools to conduct useful analyses that are appropriately correct for the conceptual level.



Length	10.000
Diameter	1.000
Nose Mult	1.500
Aft Mult	3.000
Nose Center	-0.200
Aft Center	0.300
Aft Width	0.100
Aft Height	0.300

Figure 2.2: OpenVSP home screen, showing example transport model with geometry user interfaces.

For example, OpenVSP can conduct various area, volume, and mass properties analyses, work in conjunction with the NASA-developed vortex lattice code VSPAERO for aero-propulsive analysis [18], and export data in numerous formats for use in other modeling and design tools. Here, these mid-fidelity analyses can inform design modifications before vast efforts have been used on detailed modeling. Major designs flaws can be uncovered and solved in rapid conceptual-level design iterations.

In Figure 2.2, a basic transport aircraft model is shown in the OpenVSP main screen. From the Geom Browser interface, one can see the component tree, or list of parent and child components of which the model is comprised. Here, the child components wing ("Wing"), vertical stabilizer ("Vertical"), and horizontal stabilizer ("Horizontal") belong to the parent component, the fuselage ("TransportFuse"). A red bounding box is visible around the component currently selected in the Geom Browser, which is TransportFuse in the given example. The geometry editor for TransportFuse is shown on the right of

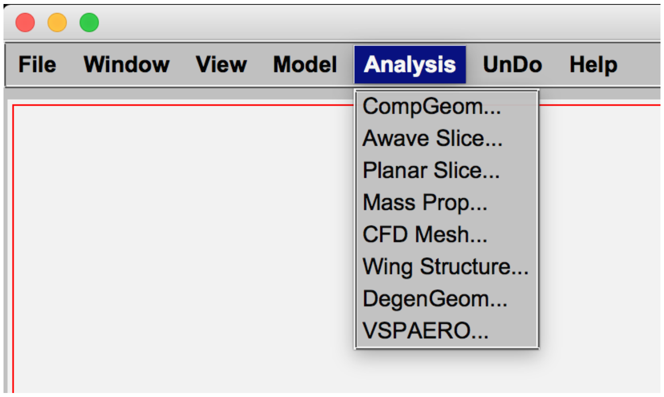


Figure 2.3: The options contained within the Analysis tab of OpenVSP.

the screen capture, exemplifying the design parameters available to the user for a given component.

The various analysis tools provided in OpenVSP can be accessed via the "Analysis" tab in the menu of the main screen, shown in Figure 2.3. From here, users may access various meshing, slicing, geometry, and aerodynamic tools. It is this collection of tools to which the wave drag tool would be added.

One will note that the "Awave Slice" tool is, indeed, related to the AWAVE wave drag tool mentioned in Chapter 1. Here, the Awave Slice tool was intended to provide merely the cross-sectional area calculations neccessary for an AWAVE analysis, rather than the AWAVE analysis itself. Additionally, bugs and insufficient capabilities further hindered the usefulness of the Awave Slicing tool. In the course of this work, the Awave Slicing tool was scrapped and its intended functionality made available as a byproduct of the more capable wave drag tool.

3 WAVE DRAG METHODOLOGY

The evaluation of wave drag begins with a momentum theory analysis in which a body is said to be contained in a control surface, as illustrated in Figure 3.1. Here, let the upstream circular surface normal to the flow be $Surf_1$, the cylinder surface parallel to the flow be $Surf_2$, and the downstream circular surface normal to the flow be $Surf_3$. Per Ref. [19], the pressure drag on the body obtained via momentum theory is then given by Eq. 3.1:

$$D = -\rho_\infty U_\infty^2 \iint_{Surf_2} \phi_x \phi_r \, dSurf_2 + \frac{1}{2}\rho_\infty U_\infty^2 \iint_{Surf_3} \left(\phi_y^2 + \phi_z^2\right) dSurf_3. \tag{3.1}$$

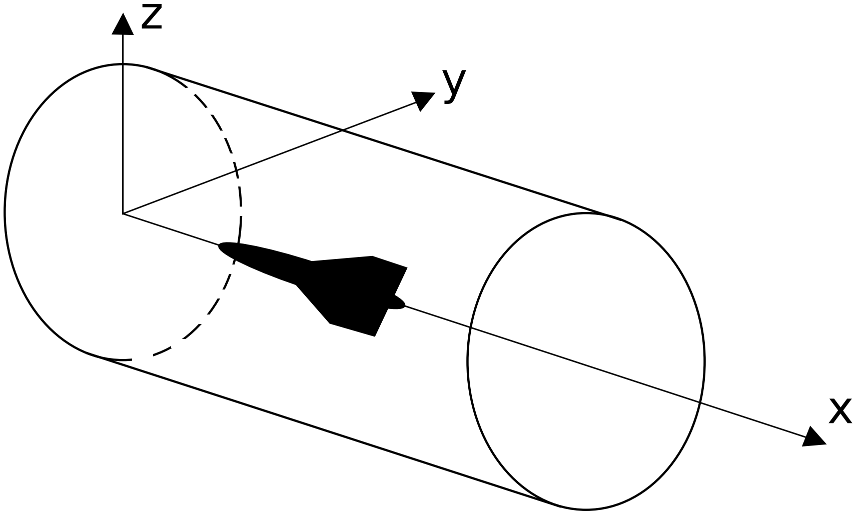


Figure 3.1: Control surface for momentum theory approach.

Here, the second term accounts for vortex drag while the first term represents the the zero-lift wave drag. From this first term, Ward gave the wave drag integral when linearized theory is applied as Eq. 3.2 in Ref. [20].

$$\frac{D}{q} = I = -\frac{1}{2\pi}\int_0^1 \int_0^1 S''(x)\, S''(y)\, \log|x-y|\, dx\, dy \tag{3.2}$$

Of particular note in Eq. 3.2 is the second derivative of cross-sectional area terms, given by S''. The cross-sectional areas to which this equation refer are the projected areas cast on the plane normal to the flow by Mach plane intersections with the body.

From a far field perspective, a Mach cone can be approximated as a set of flat Mach planes rotated about the body axis, as illustrated in Figure 3.2. The intersection of these planes with the body along its axis produce cuts like the ones illustrated in Figure 3.3. The projected area of the shaded region corresponds to S, from which the S'' in Eq. 3.2 is derived.

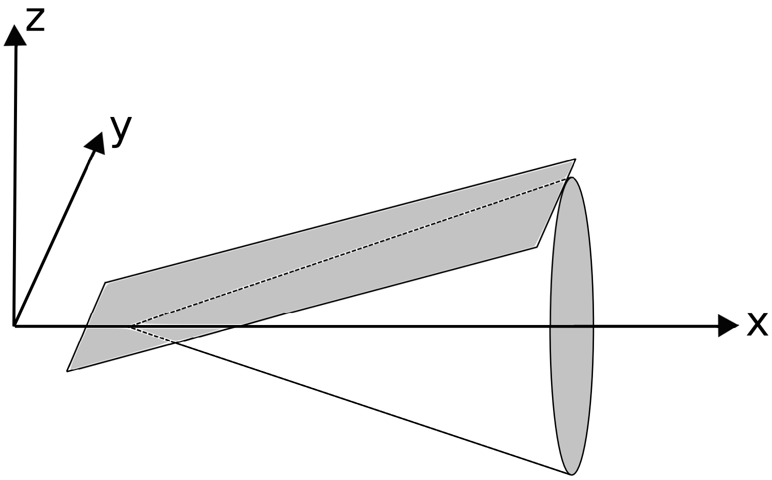


Figure 3.2: A Mach plane corresponding to a given Mach cone.

Concerning about this requirement for the second derivative of S is the ease, or lack thereof, with which the differentiation is achieved. In order for Eq. 3.2 to remain true, S must be such that S' is continuous over the length of the body, which here has been normalized to unity. Additionally, $S'(0) = S'(1) = 0$ must be satisfied. Computing the projected areas of planes intersecting an aircraft, a body with curves and protruding components like wings, at various locations and rotations is not trivial— the requirements on Eq. 3.2

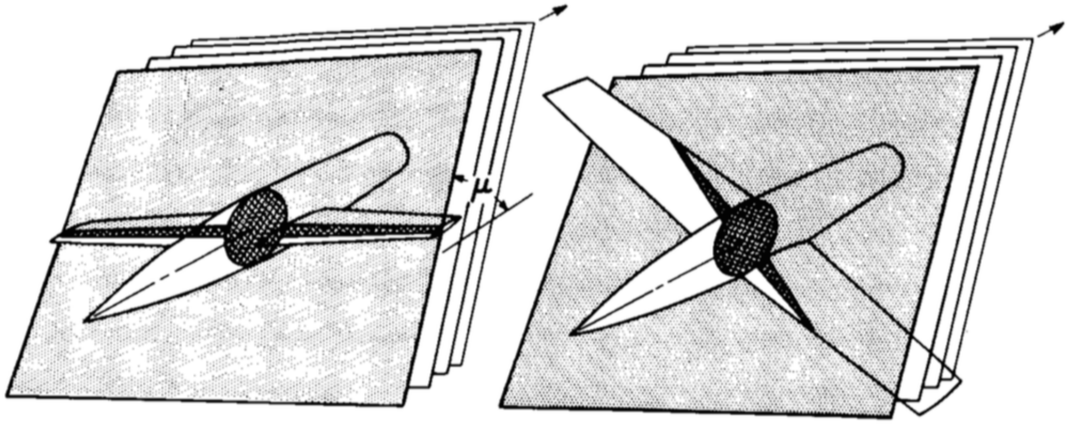


Figure 3.3: Projected areas of Mach angle cutting planes intersecting a body [21].

mandate at least several evaluations of these intersections be made for some semblance of continuity in S.

In 1956, Eminton and Lord published their treatment of the cross-sectional area distribution for evaluating the wave drag integral over slender bodies [22]. In their work, they give a Fourier sine series of the discrete values of $A(x_i)$ over n finite points between $x = 0$ and $x = 1$ such that the aforementioned conditions on Eq. 3.2 are met [23]. Here, the Fourier series offers the approximated curve that passes through each of the discrete points. Additionally, and not insignificantly, this curve also represents the minimum wave drag shape that accommodates each of the given points.

This Fourier sine series is the minimal function for which $S(x_i) = A_i$ for all i and is represented by

$$
\begin{aligned}
S(x) = a &+ \frac{1}{4}a_1(\kappa - \sin\kappa\cos\kappa) \\
&+ \sum_{r=2}^{\infty}\frac{1}{4}\left[\frac{\sin(r-1)\kappa}{r-1} - \frac{\sin(r+1)\kappa}{r+1}\right]\frac{1}{r}\sum_{i=1}^{n}\lambda_i\left[\frac{\sin(r-1)\kappa_i}{r-1} - \frac{\sin(r+1)\kappa_i}{r+1}\right]. \quad (3.3)
\end{aligned}
$$

Here,

$$a = N \tag{3.4}$$

and

$$a_1 = \frac{4}{\pi}(B - N). \tag{3.5}$$

The terms a_r for all $r > 1$ may be given by

$$a_r = \sum_{r=2}^{\infty} \frac{1}{r} \sum_{i=1}^{n} \lambda_i \left[\frac{\sin(r-1)\kappa_i}{r-1} - \frac{\sin(r+1)\kappa_i}{r+1} \right] \tag{3.6}$$

such that Eq. 3.3 may be written as

$$S(x) = a + \frac{1}{4} a_1 (\kappa - \sin\kappa \cos\kappa) + \sum_{r=2}^{\infty} \frac{1}{4} \left[\frac{\sin(r-1)\kappa}{r-1} - \frac{\sin(r+1)\kappa}{r+1} \right] a_r. \tag{3.7}$$

The κ term, a transformation of x, is obtained from

$$\kappa = \cos^{-1}(1 - 2x). \tag{3.8}$$

The coordinate system employed here is such that the x-axis is along the roll axis, with the positive direction defined as from nose to tail; the y-axis is along the pitch axis, with positive direction extending out the right wing; the z-axis is along the yaw axis, with positive direction extending upward. This is represented by the diagram in Figure 3.4. In this work, it is taken that the flow direction is along the x-axis and in the positive direction.

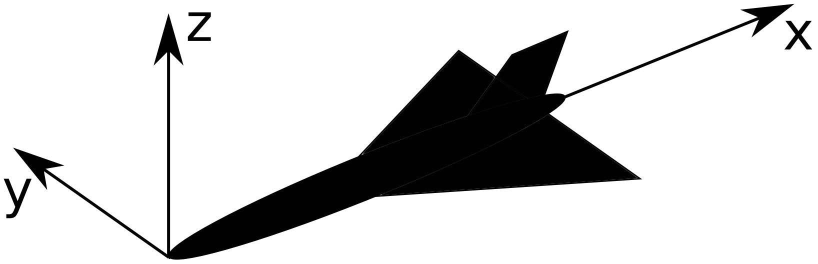


Figure 3.4: Depiction of aircraft axis system used by OpenVSP.

In this approach, a finite set of Mach cutting planes at a prescribed Mach angle, as depicted in Figure 3.2, are intersected with the aircraft to calculate the requisite cross-sectional areas. These areas are used in assuming an equivalent body of revolution. Similar sets of Mach cutting planes are generated at various θ rotations about the aircraft axis. By doing so, a wave drag evaluation is permitted at each θ rotation. Integrating over all θ values by means of Eq. 3.9 permits the overall aircraft wave drag to be evaluated [9].

$$D = \frac{1}{2\pi} \int_0^{2\pi} D(\theta)\, d\theta \tag{3.9}$$

The evaluation of the wave drag integral for each θ rotation highlights the independence of each rotation's cross-sectional area distribution from the others. The overall wave drag is calculated by the integrated average of the drag values at each θ rotation, not by an average of cross-sectional area distributions around all θ values. This precludes assigning any importance to averaging the cross-sectional areas of each θ into an overall area distribution. Not only would this value be unused in the wave drag evaluation, but would be nonsensical to compute— the bounds of the area distribution for each θ vary, thus there is no consistent range over which to average the collection of cross-sectional areas.

Further details regarding handling the Mach cutting planes, Fourier analysis, and subsequent wave drag evaluation in OpenVSP are handled in the chapters that follow.

4 ANALYSIS OF THE WAVE DRAG INTEGRAL

Various analyses were undertaken of the wave drag approach previously introduced. Such analyses informed design decisions for the wave drag tool and introduced further tool capabilities.

4.1 Calculation of Maximum Drag Locations

Of interest to the author at the start of this work was having the ability to locate the aircraft features most responsible for contributing zero-lift wave drag. While plotting the cross-sectional area provides visual indication of any departures from a smooth distribution, analytically determining the axis location with the highest contribution to wave drag would lend confidence. To do so, seeking the conditions of the terms in Eq. 3.2 under which D is maximized became necessary.

First to be considered was the logarithmic term. As the focus of this analysis was to determine conditions for maximum D of Eq. 3.2, the immediate interest with this natural logarithmic term became minimizing the argument to which it is passed, as this maximizes the resulting magnitude. One should note the negative multiplier of Eq. 3.2 combines with the negative result of the logarithmic term to produce a positive value of D.

On the normalized values x and y of the wave drag integral, it is clear that the $|x-y|$ argument yields a minimum value of zero and maximum value of 1. Under these conditions, the case for which $|x-y|$ equals zero will yield the most magnitude from the logarithmic term. Thus, for the purposes of maximizing D, only the cases in which x equals y must be considered. A reference plot of the natural logarithm on these bounds is shown in

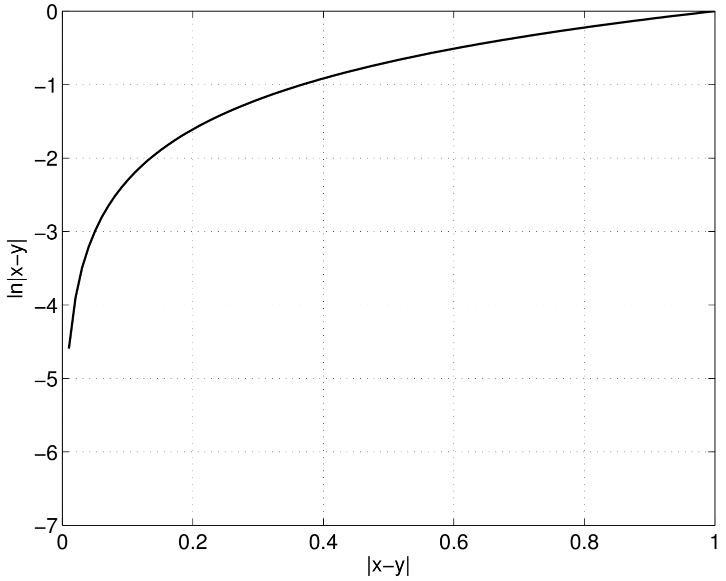


Figure 4.1: Plot of the natural log of $|x-y|$ over the bounds $0 \leq x, y \leq 1$.

Figure 4.1. The singularity that occurs at $log(0)$ is accounted for by Eminton and Lord in Ref. [22], to where the reader is referred if greater explanation is sought. There is no concern that aversion of the singularity is inconsistent with physical behavior, as drag is known to be a finite value.

It is clearly seen that the left portion of the drag integral in Eq. 3.2 is maximized when the S'' terms are largest and of the same sign. Thus, if x and y are taken to be equal per the analysis of the logarithmic term, one may rewrite this left portion as

$$S''(x)S''(y) = S''(x)S''(x) = \left(S''(x)\right)^2.$$

With the square, it can then be taken that the x-value providing maximum magnitude of S'' is the value of interest for maximizing $((S''(x))^2$. Using the magnitude of S'' also ensures that negative values are considered.

Thus, from this inspection, it has been shown that: 1) the logarithmic portion of the wave drag integral is maximized in magnitude when x and y are of equal value, and 2) the remaining portion can be brought to maximum by letting $x = y$ from the prior understanding and then seeking the x-location that produces the greatest magnitude of S''. Given this understanding, the location of the highest wave drag contribution may be found by inspecting the absolute value of S'' over the bounds of $0 \leq x \leq 1$, and the x-value producing the largest result is the x-value of interest.

4.2 Impact of Engine Flow-Through on Wave Drag Evaluation

Aircraft cross-sectional area distribution is affected by one's approach to handling the inlet and exit area of propulsive faces. As the flow-through portion of an engine does not block the air, the projected areas of engine inlets and exits do not contribute to wave drag. Recall from Chapter 1 the constant diameter of the stream tubes and the metaphor of "pipefitting"— the stream tubes entering and exiting the engine do not have to move out of the way, and thus do not contribute to the area calculations considered in wave drag evaluation. For this work, it was necessary that the author explore how a given approach to accounting for these propulsive faces would affect the wave drag results.

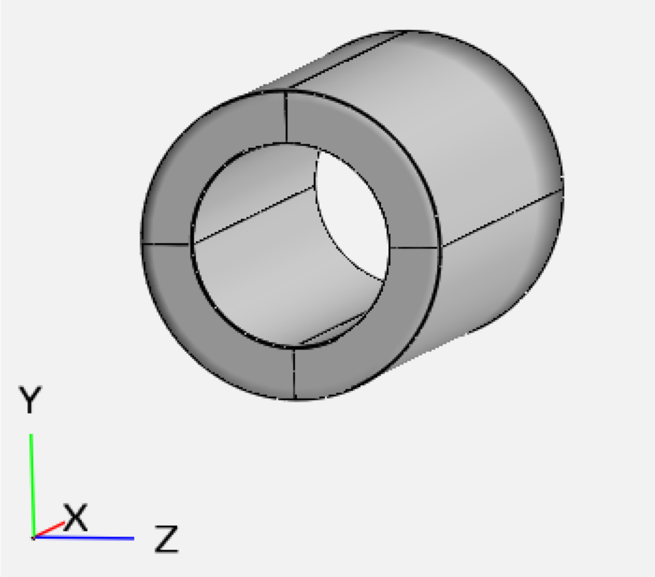


Figure 4.2: Engine component with a hollow center.

Accounting for the flow of air through a propulsive device can be handled prior to the cross-sectional area calculation by intentionally hollowing out the propulsive component in the model, as in Figure 4.2. If this is done, the management of the projected area of the propulsive faces is no longer necessary— having hollowed out the flow-through portion of the component, this area is not included in the cross-sectional area calculation in the first place. Methods to account for this propulsive face area become necessary in the cases in which the flow-through holes are *not* modeled and the propulsive components remain solid.

In OpenVSP, it is not always convenient or possible to model these flow-through holes, leaving solid propulsive components as common occurrences. Historical cross-sectional area plots also demonstrate a similar propensity for propulsive devices that are modeled without flow-through holes [9, 24, 25]. In these cases, the projected area of the propulsive faces is considered in the total cross-sectional area of the aircraft. An example is given to better illustrate this concept.

Consider Figure 4.3, which is a top view of a fighter model with visible engine inlets on the sides of the fuselage and an exit at the rear. The cross-sectional area distribution for this model could be represented by Figure 4.4. Here, it is clear that the cross-sectional area curve jumps upward when the engine inlets occur, then back downward when the engine exit is located.

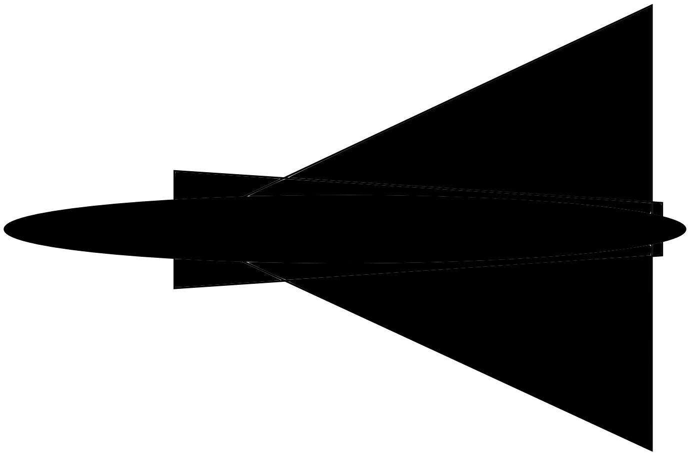


Figure 4.3: Top view of an example fighter model, with engine inlets visible on the sides of the fuselage.

When one now considers the projected areas of engine inlets and exits being accounted for in order to manage flow-through components, consider the dashed lines in Figure 4.5.

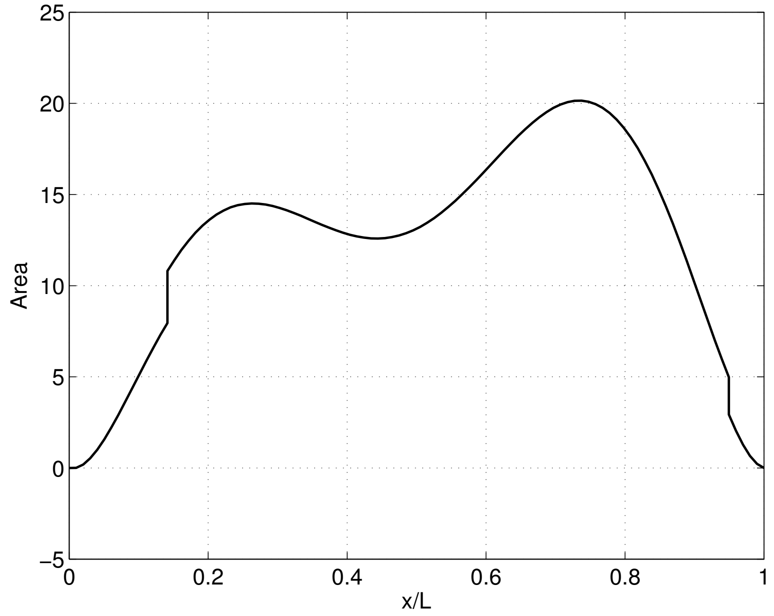


Figure 4.4: Representative cross-sectional area distribution for the example in Figure 4.3.

These lines indicate the area occupied by the inlets and exit of the example model if they were projected forward and aft. Properly accounting for these projected areas allows one to successfully manage flow-through area in a wave drag analysis. However, the means by which these projected areas are incorporated into the results differ, with two common approaches described below.

Arguably the most straightforward approach, Approach A, is to simply incorporate the projected area of propulsive faces into the total cross-sectional area distribution. Here, the area value at each station along the aircraft axis receives an increase corresponding to the value of projected area from the propulsive faces. Locations in front of the propulsive components receive the projected inlet areas while locations aft of the propulsive components receive projected exit areas. The visual result is an upward shift of the cross-sectional area curve; the beginning and end points are positive values equating to the total projected inlet and exit area of the propulsive faces.

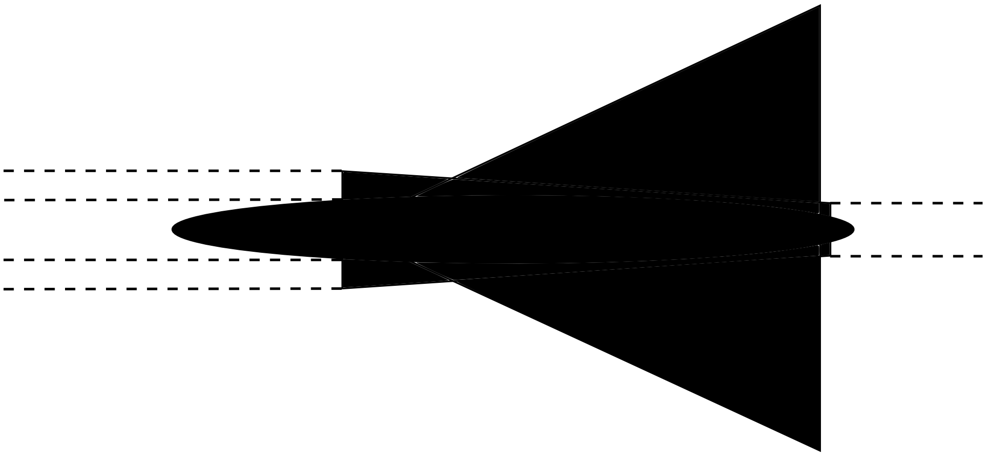


Figure 4.5: Top view of an example fighter model with dashed lines representing projected engine faces.

Continuing the previous example, this Approach A can be seen utilized in Figure 4.6. As shown, the dashed line projected engine areas of Figure 4.5 have been accounted for, and the cross-sectional area distribution of Figure 4.6 begins and ends at positive values.

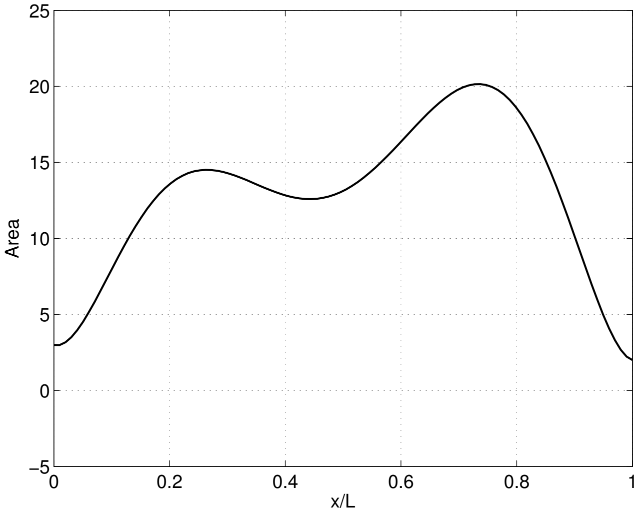


Figure 4.6: Example area distribution utilizing flow-through Approach A.

Another approach, Approach B, is to subtract the inlet area from each station's total cross-sectional area. Doing so removes the projected area of the inlet from the area calculated at each station and effectively artificially creates the flow-through holes in the propulsive components by removing the quantity of area attributed to the projected inlet

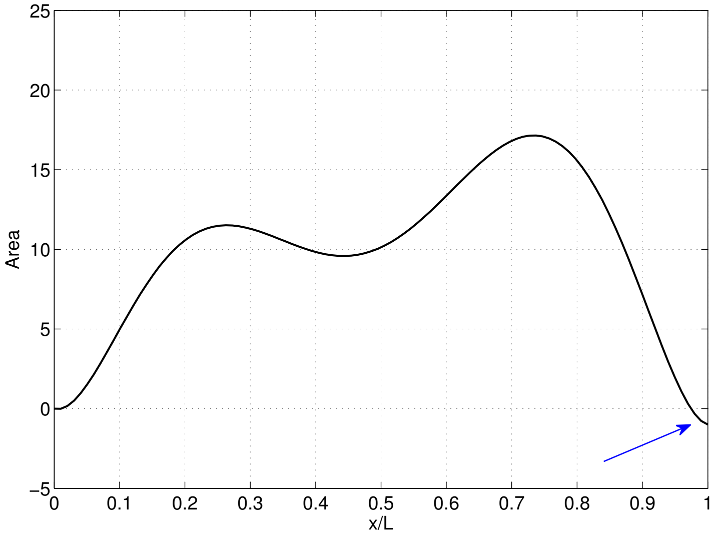


Figure 4.7: Example area distribution utilizing flow-through Approach B.

area. This shifts downward the cross-sectional area curve discussed above, with the starting area equal to zero. However, any difference in the inlet area from the exit area will result in a non-zero end value in the cross-sectional area curve; often times this final value is negative, occurring when the exit area of the propulsive faces is smaller than that of the inlet faces.

Figure 4.7 continues the example from Figure 4.6 with Approach B applied. Note the right hand side of the curve reaches negative values when the inlet area has been subtracted. The false notion that the rear of the aircraft contains negative area is merely a result of indiscriminately subtracting the value of inlet area from nose to tail.

Another example can be provided by Ref. [24], in which Feagin presents the cross-sectional area distribution for the F-106A (a top view shown in Figure 4.8) using this Approach B, which can be seen in Figure 4.9.

In building the wave drag tool, the author explored the impact on the results if flow-through components were implemented by subtracting the inlet area from the first x-location, the exit area from the last x-location, and a linear combination of the two for all locations in between. Thus, the cross-sectional area curve would begin and end at zero area value.

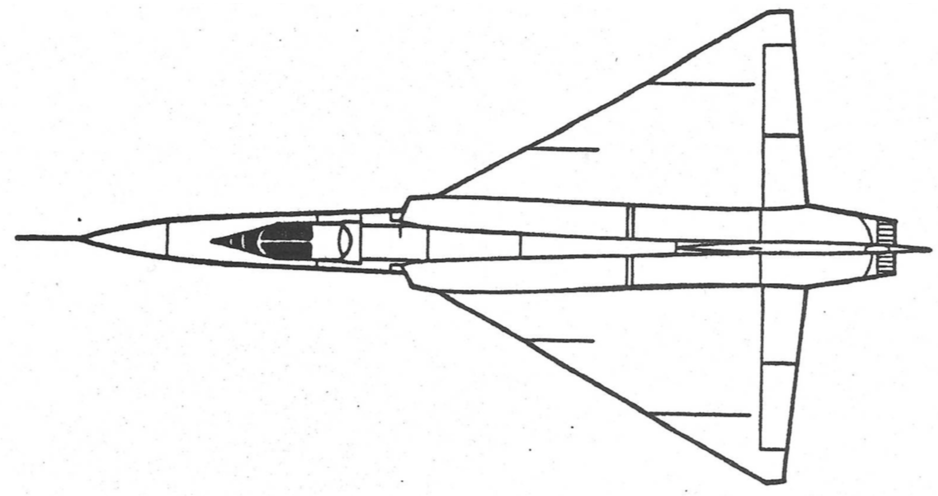


Figure 4.8: Top view drawing of an F-106A [24].

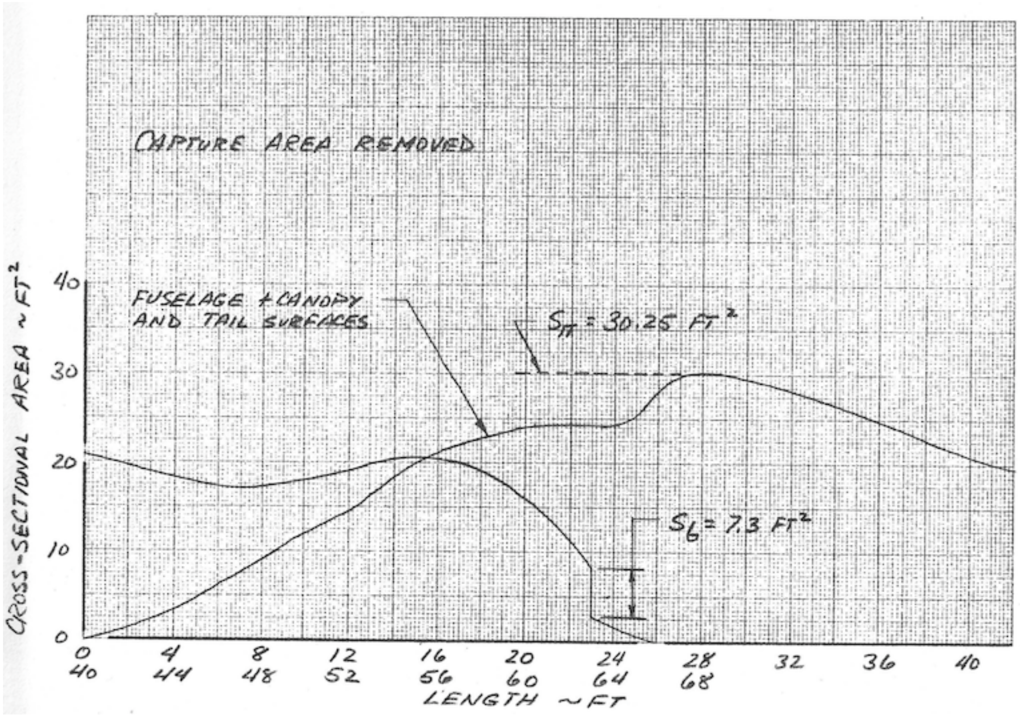


Figure 4.9: Cross-sectional area distribution for the F-106A [24]. Here, capture area has been removed.

This approach, Approach C, poses two distinct advantages over Approaches A and B: 1) the cross-sectional area curve starting and ending at zero is consistent physically with the area contributed by an aircraft of finite length to a set of x-locations spanning a length greater than the aircraft; 2) the curve ending at zero eliminates the physical impossibility, and ensuing non-intuitiveness, of negative area in the rear of the aircraft when smaller exit areas occur in Approach B.

Approach C applied to the example case is shown in Figure 4.10. Note that the curve begins and ends at zero, despite unequal inlet and exit areas as shown in Figure 4.6.

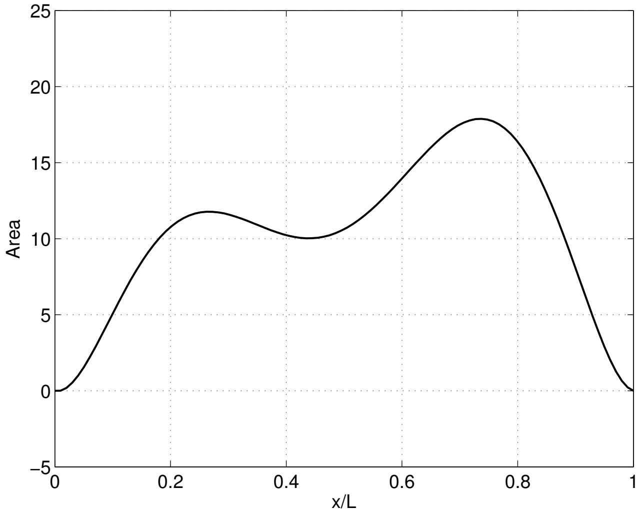


Figure 4.10: Example area distribution utilizing flow-through Approach C.

The question that inevitably arises when any of the above approaches is utilized is how the changes made to the cross-sectional area distribution affect the wave drag calculation. Clearly, it would be unacceptable if any changes made to the area distribution altered the wave drag result.

Here, consider the mathematical impact each of the three approaches above has on the cross-sectional area distribution equation from Eq. 3.7. Approach A entails adding either the value of projected inlet area or projected exit area; while, strictly speaking, the switchover between which value to add implies a piecewise function, the important part to note is the value being added to Eq. 3.7 is only a constant. In Approach B, the simple subtraction of the inlet projected area at all x-locations also adds only a constant to Eq. 3.7. Approach C subtracts a linear combination of inlet and exit projection areas, making this approach dependent on x. The equation for Approach C is shown in Eq. 4.1.

$$S_C(x) = S(x) - \left[S(0) - (S(0) - S(L))\left(\frac{x}{L}\right)\right] \tag{4.1}$$

The critical understanding here is gained by recalling that, as shown in Eq. 3.2, wave drag is a function of the *second derivative* of the cross-sectional area, or S''. Thus, to have an impact on the second derivative, changes to the cross-sectional area distribution equation, S, must be of second-order in x or higher. Any terms of first- or zeroth-order will differentiate to zero by the second derivative. Therefore, the mathematical contributions to the cross-sectional area distribution of Approaches A, B, and C (constant, constant, and first-order x, respectively) differentiate to zero in the equation for S''. With zero contribution to S'', Approaches A, B, and C have no impact on the results of the wave drag equation.

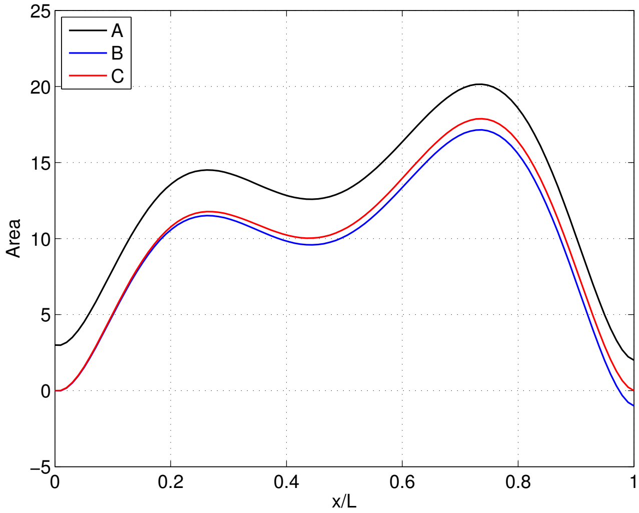


Figure 4.11: Comparing the area distributions of Approaches A, B, and C for the example case.

For visual verification of this conclusion, the area distributions of Figures 4.6, 4.7, and 4.10 are shown together in Figure 4.11; by comparison, the second derivatives of area for all three approaches are shown in Figure 4.12. As seen, the S equations are clearly different for all three approaches, but the terms applied to handle flow-through components do not manifest any changes in the S'' equations and, thus, have no affect on the wave drag evaluation.

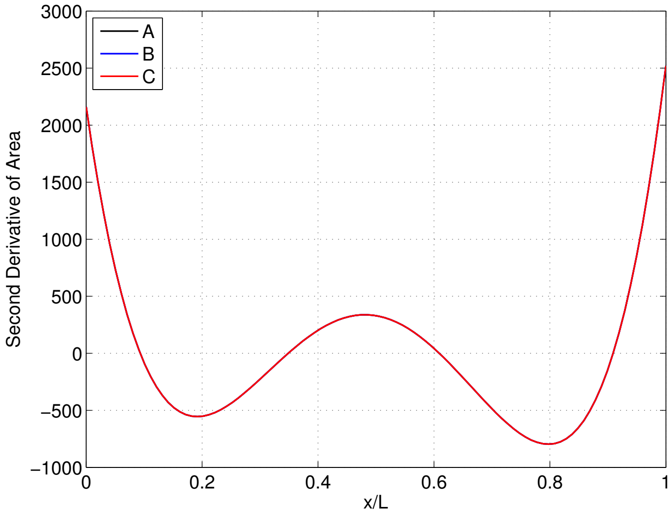


Figure 4.12: Comparing the second derivatives of area of Approaches A, B, and C for the example case.

It may be helpful to think of the wave drag depending not on the cross-sectional area itself, but on the behavior of its second derivative over bounds extending before and after the aircraft. Whether the cross-sectional area curve starts and ends at non-zero values, as in Approaches A and B, or at zero in Approach C, the value of the second derivative of area at the start and end locations is always zero. Of course, the same holds true for any case in which mathematically accounting for flow-through is unnecessary and the cross-sectional area distribution is not modified. Adjusting the cross-sectional area distribution by a constant or first-order equation will never affect the value of the second derivative and, thus, never affect the outcome of the wave drag equation.

This understanding allowed the author to progress forward with Approach C, the most conceptually and visually attractive approach of the three outlined above.

5 INCORPORATING THE WAVE DRAG TOOL INTO OPENVSP

Both the geometry engine and user interface of OpenVSP received additions and modifications in the construction of the wave drag tool. Completely new classes were created to handle the new wave drag capability: `WaveDragScreen` and `WaveDragMgr`. Additionally, the newly created `WaveDragEL` contains the functions for specifically the wave drag evaluation. These are complemented by modifications made to existing classes, primarily `Vehicle` and `MeshGeom`, to facilitate the new functionality. A general flow diagram of the wave drag tool information is shown in Figure 5.1.

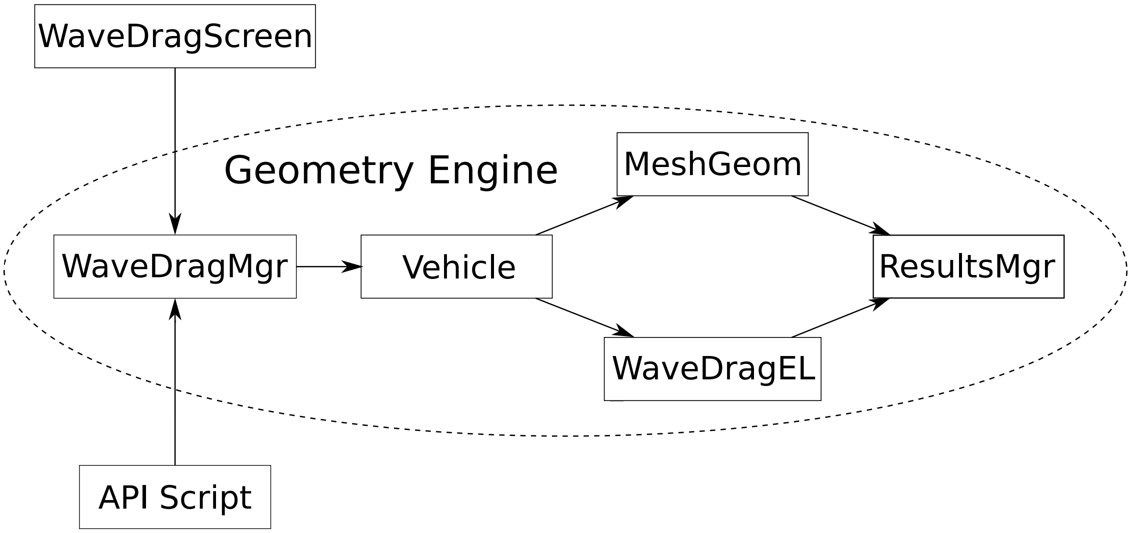


Figure 5.1: Flow of wave drag tool information between OpenVSP classes.

The `WaveDragScreen` class was created to handle the user interface portion of the wave drag tool and is housed, naturally, in the GUI engine of the source code. In keeping with OpenVSP's separation of geometry management and graphics, no methods or parameters upon which the wave drag evaluation is dependent are housed in `WaveDragScreen`—

WaveDragScreen will not exist in implementations of OpenVSP that are compiled in the no-graphics format. As such, the WaveDragMgr class exists as the parameter and method manager for the wave drag tool. The WaveDragMgr class exists in the geometry engine and is always included in OpenVSP builds. When graphics are present, WaveDragScreen makes use of the necessary public members of WaveDragMgr.

Vehicle manages all of the geometries and is the primary class of the geometry engine. Through Vehicle, information about the geometry can be used with methods of other classes to accomplish various tasks. As WaveDragMgr does not inherently contain knowledge of the geometries, its parameters are used in calls to the methods of Vehicle to handle the execution of area calculations and wave drag evaluations.

The MeshGeom class permits working with the triangular meshes of which the geometries are comprised: triangles, nodes, edges, and all of their associated information are managed through MeshGeom. The methods for conducting the Mach cutting plane procedure and subsequent area calculations are handled by MeshGeom and called from Vehicle.

WaveDragEL exists to execute the wave drag evaluation per the approach of Eminton and Lord in Ref. [22], hence the "EL" in the class name. As with the area routine, Vehicle uses the WaveDragEL class to handle the wave drag calculations.

In both cases of MeshGeom area calculations and WaveDragEL wave drag evaluations, the results are pushed to ResultsMgr to handle the formatting and creation of output files. Results pushed to ResultsMgr are accessed by WaveDragMgr in intermediate calculations, accessed by WaveDragScreen for displaying results to the user interface, and are accessible to the API when scripts are being used to run OpenVSP in no-graphics builds.

5.1 Mach Cutting Plane Procedure

The ability of OpenVSP to intersect components with one another was used in creating the existing tools "Awave Slice" and "Planar Slice", as visible in Figure 2.3. Development of the wave drag tool rested on further development of this ability.

When creating the Mach cutting planes for a wave drag analysis, of critical importance is determining the locations at which to start and end these slices along the x-axis. Improper determination of these locations would result in the slicing bounds starting too late or ending too early– thus, missing part of the aircraft– or starting too early or ending too late– thus, slicing useless empty space beyond the scope of the aircraft.

For clearer understanding of this problem, consider Figure 5.2. Here, an aircraft of length 10 is shown; the black line corresponds to a Mach cutting plane at Mach 1.5 that is located at the same starting place along the *x*-axis as the aircraft. As can be seen, the Mach cutting plane clips part of the wing when the slicing routine is given this starting point. The cross-sectional area distribution that would be returned for this slicing routine would then incompletely represent the aircraft. Simply letting the bounds of the slicing routine equal the bounds of the aircraft along the *x*-axis, then, was insufficient for the development of the wave drag tool. Thus, a more logical approach was required.

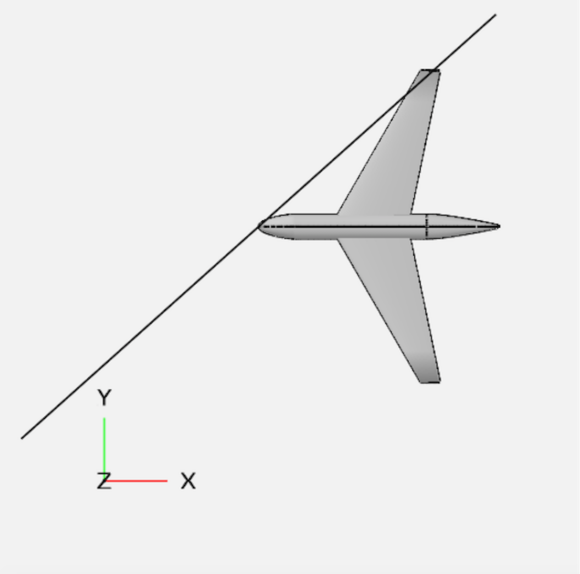


Figure 5.2: Aircraft model with an example Mach cutting plane located at the same x-axis location as the aircraft. The cutting plane clips the wing, demonstrating the need for tailored cutting plane bounds.

In Ref. [9], a trigonometric approach is outlined for properly accounting for the shift in the cutting plane bounds necessary to perfectly capture the aircraft for a given Mach angle and θ rotation. This equation is reproduced here

$$X = x - (\beta\cos\theta)(y + r\cos\theta) - (\beta\sin\theta)(z + r + \sin\theta) \tag{5.1}$$

where

$$\beta = \sqrt{M^2 - 1}. \tag{5.2}$$

This equation is present in the Harris Wave Drag code to determine proper cutting plane bounds for a wing-mounted pod where x, y, and z are the coordinates for the centerline of the pod at the leading edge and r is the pod radius. The result of Eq. 5.1, X, is the x-distance from the front of the pod at which the first cutting plane must be located to be tangent to the circle that defines the front face of the pod.

In this work, the usefulness of Eq. 5.1 was expanded to permit a sweep over all points of the aircraft to determine the slicing bounds that accommodate more than just wing-mounted pods. Here, the (x, y, z) coordinates of each node in the aircraft mesh are used in conjunction with Eq. 5.1— the result is the x-distance from the node at which the given Mach cutting plane should be located for tangency. Sweeping over all aircraft nodes yields the foremost and aftmost x-axis locations calculated by the nodal analysis.

The same aircraft in the example shown in Figure 5.2 is shown again in Figures 5.3 and 5.4 with the black lines representing the Mach 1.5 cutting planes at the start and end locations calculated by the nodal analysis outlined above. As seen, with with aircraft of length 10 spanning $0 \leq x \leq 10$ along the x-axis, the Mach cutting plane bounds must be located -0.514 units before the aircraft and 4.820 units after the aircraft to ensure appropriate capture of area.

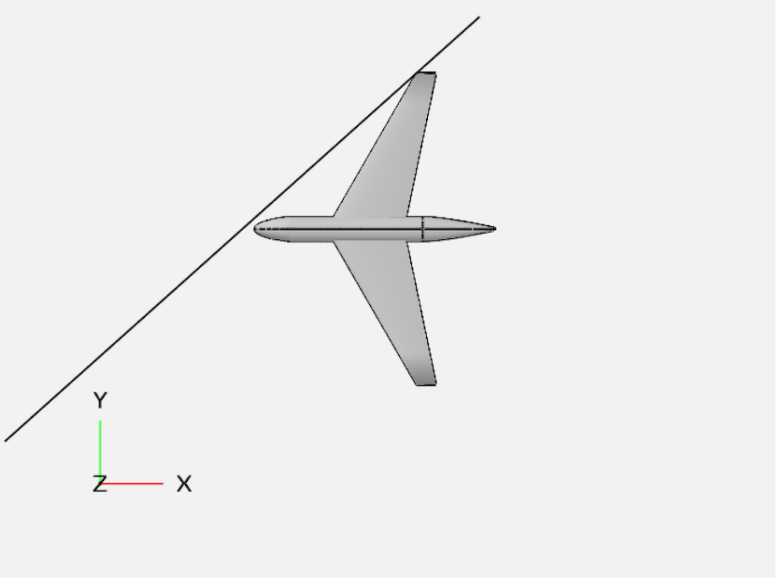


Figure 5.3: Tangent cutting plane at $x = -0.514$.

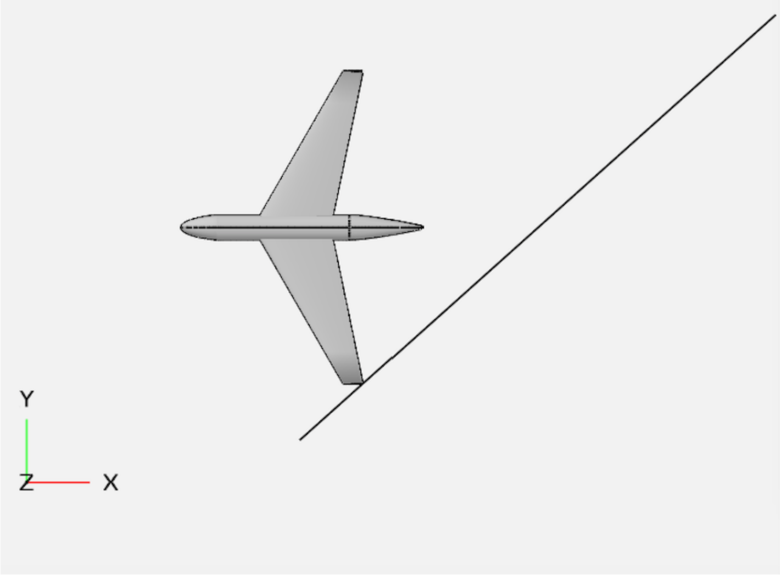


Figure 5.4: Tangent cutting plane at $x = 14.820$.

This nodal analysis is repeated for each value of θ in the wave drag tool run. Conducted in `Vehicle` with the Mach number and rotation information from `WaveDragMgr`, these slicing bounds are used as additional inputs when the slicing routine in `MeshGeom` is then called to conduct the cross-sectional area calculations. There, two-dimensional cutting planes are generated at desired locations and orientations, after which the edges formed by the planes intersecting the aircraft are determined. The parts of the cutting planes that fall outside this intersection edge are discarded and the area of the remaining portion of the plane is calculated. This area corresponds to the shaded region of Figure 5.5.

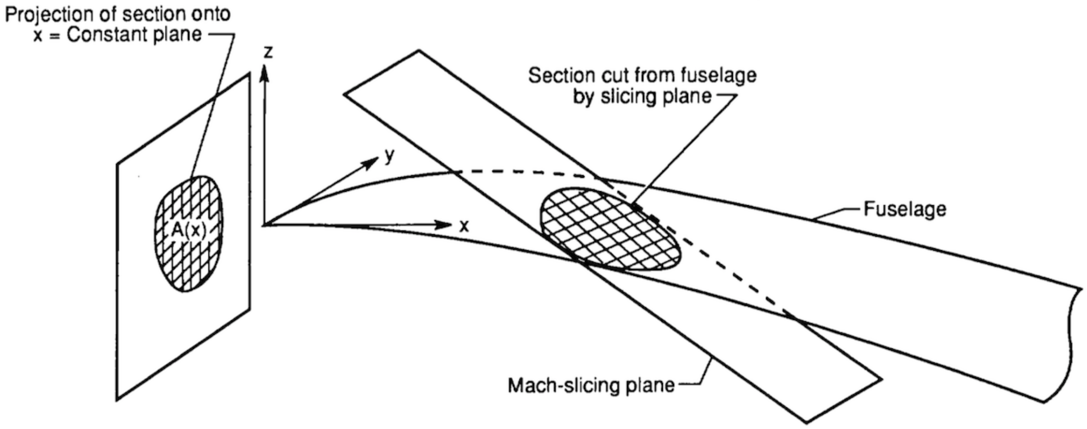


Figure 5.5: Depiction of the projected area of a Mach cutting plane when intersected with a body [26].

The meshes of both the aircraft and the Mach cutting plane slices are made visible to the user in the main screen upon running the wave drag tool. Modifications were made to the MeshGeom GUI to allow the user to toggle mesh and slices on or off for further visual flexibility.

5.2 Wave Drag Evaluation

As prescribed by Chapter 3, the Eminton-Lord approach is utilized in this work for the evaluation of the wave drag integral. This approach has been the standard for wave drag tools, being the routine of choice in the Harris Wave Drag code and, subsequently, the AWAVE tool [9].

In the Harris Wave Drag code, a subroutine appropriately named `emlord` implements the Fourier sine series method of Eminton and Lord to evaluate the wave drag integral. A similar routine was implemented in C as a standalone executable by Sriram Rallabhandi of NASA Langley Research Center; this code was made available to the author. Modifications were necessary to compile the Eminton-Lord routine as an OpenVSP class, rather than an executable used with an input data file and command line.

The OpenVSP implementation of the modified C routine exists in `WaveDragEL` and uses information regarding the number and area of Mach cutting plane slices to evaluate wave drag. Per the methodology discussed in Chapter 3 and introduced in Eq. 3.9, `WaveDragEL` is executed once for each set of sliced areas at a given rotation.

5.3 Accommodating Flow-Through Components

As discussed in Section 4.2, handling flow-through components can be achieved directly by building a component with a hole or indirectly by building a solid component and using a form of area subtractions. In this work, accommodating solid flow-through components was accomplished by extending flow faces into stream tubes that are intersected by each Mach cutting plane. This approach was modeled after the Harris Wave Drag code.

In Ref. [9], Harris describes accommodating engine faces by extending them into infinitely long cylinders and subtracting the resulting area; in practice, the Harris Wave Drag code subtracts the capture area of the first slice through an engine from the total area, as in Approach B of Section 4.2 rather than actually extending the faces to “infinity” [9]. In the wave drag tool, the flow faces are extended just far enough to be captured by each Mach cutting plane slice. This allows for more complex inlet and exit faces than the Harris code’s strict requirements of circles distributed x-wise on the $Y - Z$ plane.

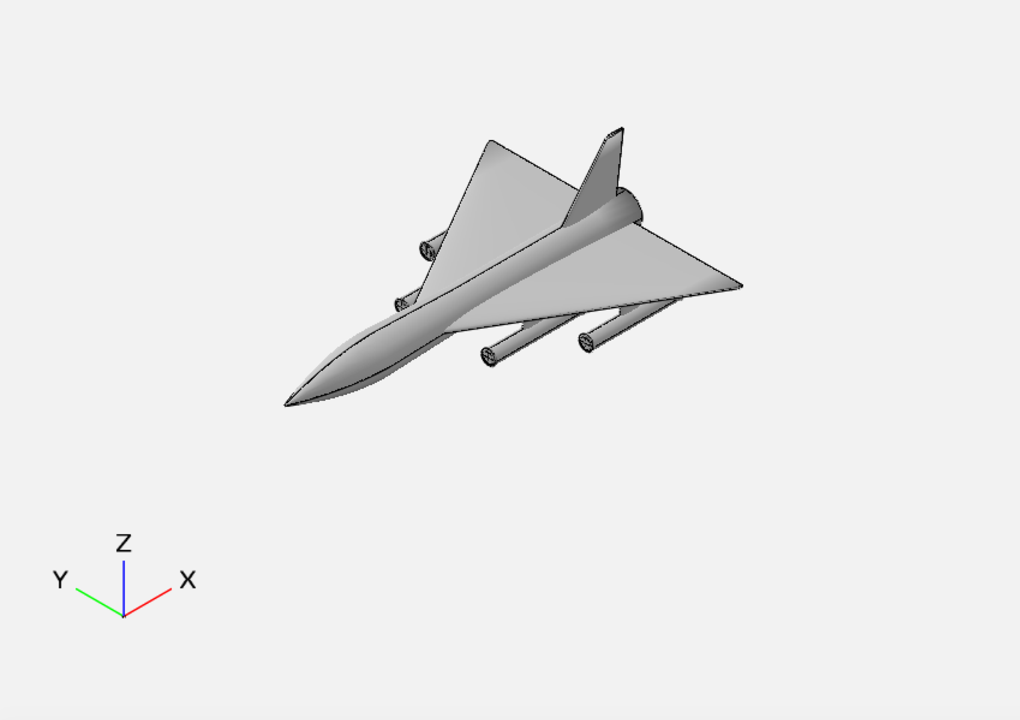


Figure 5.6: Example fighter model.

Here, subsurfaces are used for creating flow faces, with the wave drag tool providing the functionality to designate subsurfaces as flow-through. Creating subsurfaces is an existing capability of OpenVSP that allows users to draw lines and select areas of components for special treatment [27]. For the purpose of demonstration, an example fighter model is used in this section to aid the description of the subsurface and flow-through process. This model is shown in Figure 5.6.

Figure 5.7 shows one of the engine components of the fighter example before any subsurfaces were added. Figure 5.8 shows the same engine component with an inlet subsurface added; the area enclosed by this line becomes the subsurface that may be used by the wave drag tool to account for the flow-through area.

With a subsurface, the mesh-generating features of OpenVSP first intersect subsurfaces with their parent component before the component intersection. Doing so ensures not only that subsurfaces are meshed with smooth lines, but that that triangles inside the subsurface all receive a special tag indicating its location inside a specific subsurface [27]. This tagging can be used in such ways as in conjunction with a CFD analysis to tag certain areas as control surfaces, or in the wave drag tool to tag flow faces.

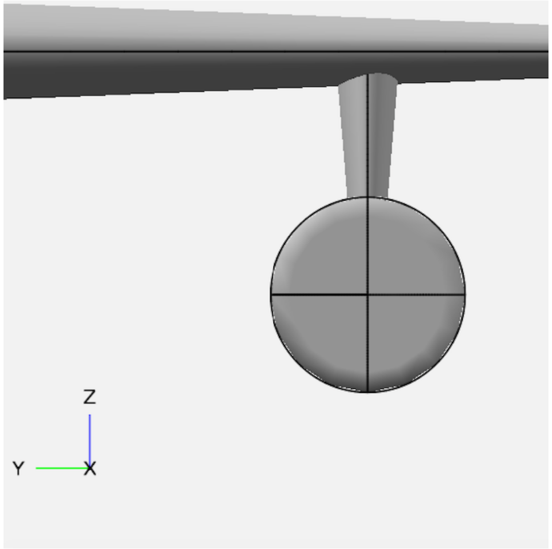


Figure 5.7: Example fighter model engine component before inlet flow face subsurface has been added.

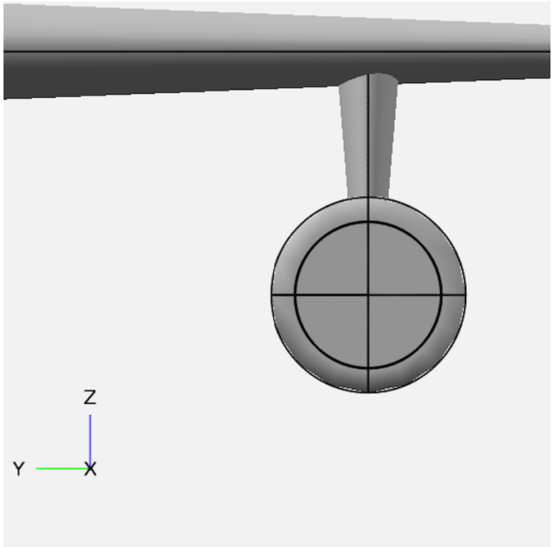


Figure 5.8: Example fighter model engine component after inlet flow face subsurface has been added.

When the wave drag tool is executed, the identifiers for the subsurfaces indicated as flow-through are passed to `MeshGeom` when the slicing routine is called from `Vehicle`. In `MeshGeom`, the subsurface intersection routine loops over the subsurfaces in each triangular

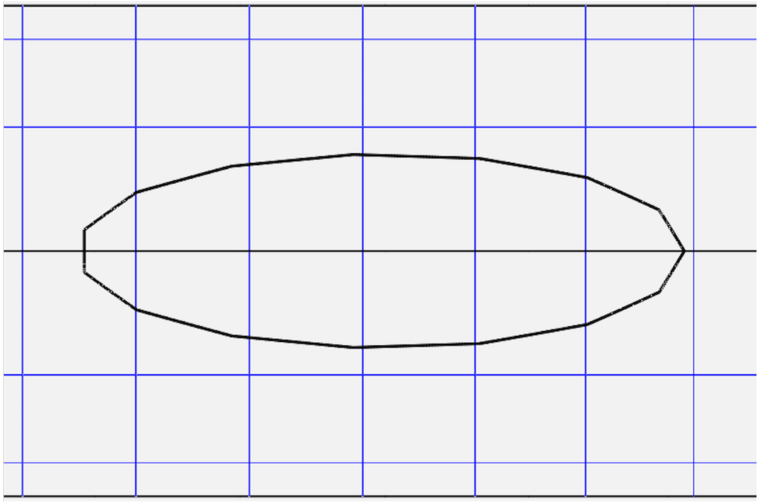


Figure 5.9: Subsurface before intersection with parent component.

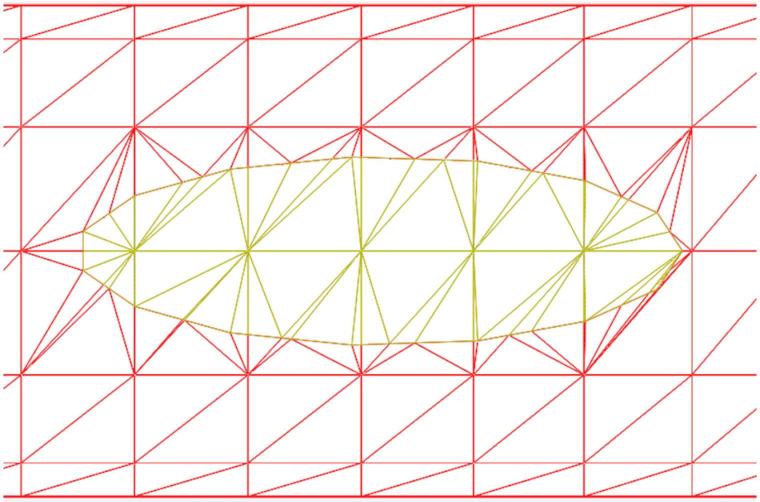


Figure 5.10: Subsurface after intersection with parent component.

mesh (`TMesh`) in the model; the intersection and extension routines are executed for all subsurfaces dictated as flow-through.

Each subsurface is first intersected with the parent component just as in the `CompGeom` routine. In each `TMesh`, new edges and nodes are created along the subsurface boundaries and the resulting triangles are split. An example subsurface before and after intersection with the parent component is shown in Figures 5.9 and 5.10.

After the intersection, the norms of each subsurface mesh are then examined for the signs of their *x*-components. Negative *x*-values in the norms indicate that the subsurface

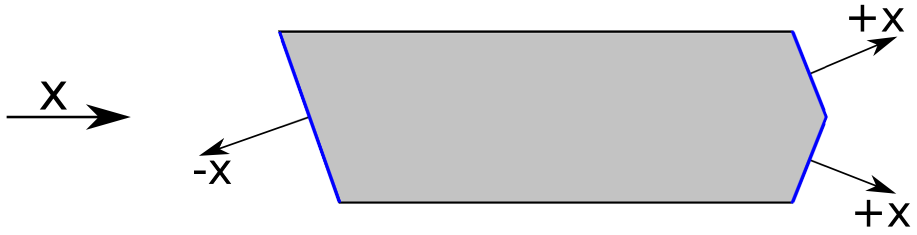


Figure 5.11: Depiction of subsurface norm *x*-components. The two blue surfaces indicate the inlet and exit faces.

is facing forward and is an inlet; positive norm *x*-values indicate the subsurface is facing aft and is an exit. This determination dictates whether the following routine extends the subsurface forward or aft.

For robustness, the number of positive and negative norm *x*-values are counted when each subsurface mesh is run through the routine. Nominally, the triangles all return norm *x*-values of the same sign, indicating the subsurface at least generally faces exclusively forward or exclusively aft. An example of this is shown in Figure 5.11. However, if the user has dictated as flow-through a subsurface that faces both forward *and* aft, both positive and negative norm *x*-values will occur. This case, shown in the example in Figure 5.12, represents a flow-through surface that is neither an inlet nor an exit as it faces both upstream and downstream simultaneously. Such a case is not a useful nor viable flow-through surface and triggers an “ambiguous subsurface” error to the console and the subsurface will not be extended.

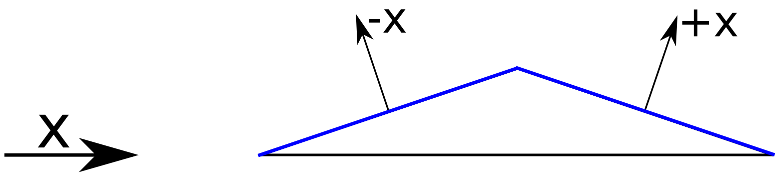


Figure 5.12: Depiction of ambiguous subsurface norm *x*-components. The blue surface indicates the single face that points in two *x*-directions.

From here, the edges of intersection between the subsurfaces and parent components are looped over to construct the open cylinder of the extension. Each intersection edge contains two nodes, one on either side. Each of these nodes is duplicated and the copies are

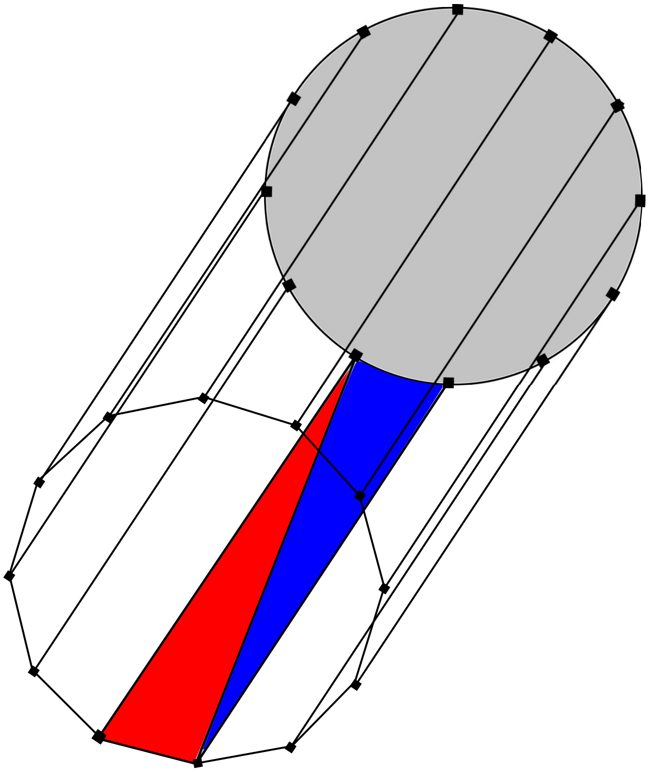


Figure 5.13: Depiction of triangles added after translating copies of subsurface intersection nodes. The red and blue triangles highlight how each rectangle of four nodes becomes two meshed triangles.

translated to the extension location determined by the wave drag routine in `Vehicle`. From these nodes, two triangles are created that form a rectangle: one containing the two existing nodes and the first of the new translated nodes, and one containing the two new translated nodes and the second of the existing nodes. This is depicted in Figure 5.13; this process is repeated for each of the intersection edges. The result is a hollow cylinder extending from the subsurface to the predetermined extension point. Looping over all subsurfaces in all `TMeshes` ensures that every flow-through subsurface receives its extension tube.

When the loops have been completed, all triangles in the geometry are looped over to search for those containing subsurface tags. If a triangle is tagged as one of the inlet subsurfaces, the triangle is translated to the inlet extension location. Likewise, if a triangle is tagged as one of exit subsurfaces, the triangle is translated to the exit extension location.

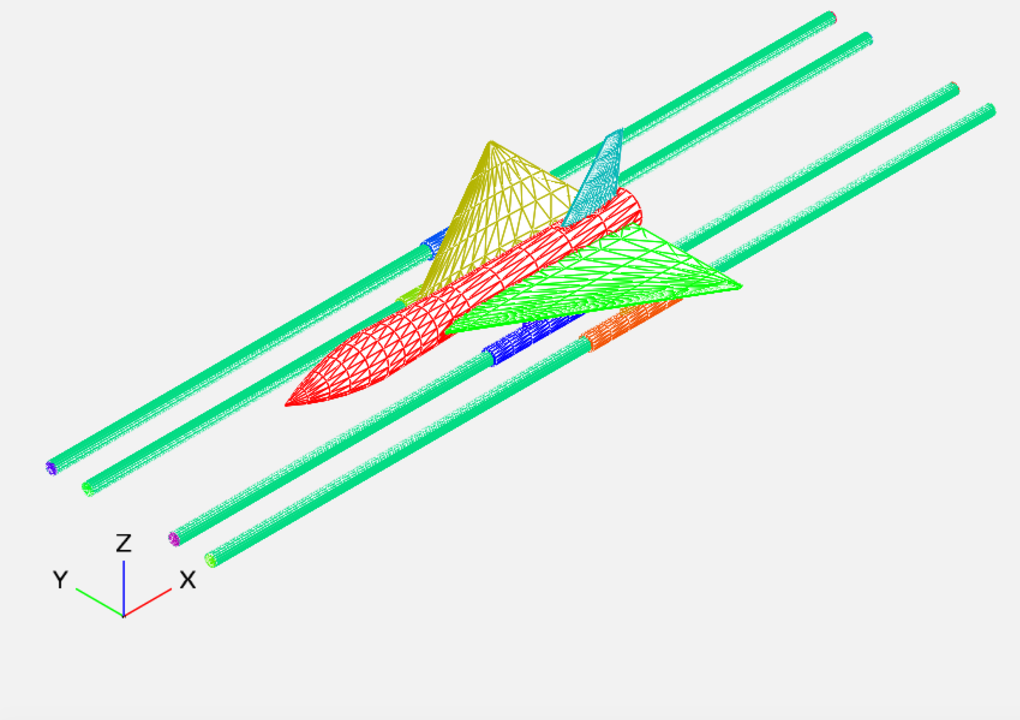


Figure 5.14: Fighter example, showing extension of flow-through tubes after running the wave drag tool.

When the triangles of a subsurface are translated, the `TMesh` from which the subsurface originated is no longer watertight. The extension tube, however, is the mesh that perfectly connects the `TMesh` to its translated subsurface triangles. Thus, the extension tube meshes are then merged with their corresponding `TMeshes`, where duplicate nodes and edges are removed. The result is each `TMesh` containing flow-through subsurfaces returns to a single watertight mesh, extension tubes included. Figure 5.14 shows the flow-through subsurfaces of the fighter example model from Figure 5.6 after execution of the wave drag tool. Subsurfaces, in the manner shown in Figure 5.8, were added as inlet and exits on all four engine components.

The watertight translation of subsurfaces with flow-through designation allows the projected area of the flow-through faces to be accounted for when the cutting planes are intersected with the aircraft. Two extra slices are used in the `MeshGeom` slicing routine explicitly to slice any inlet extension tubes and any exit extension tubes. The model from Figures 5.6

and 5.14 is shown in Figures 5.15 and 5.16 with the black line demonstrating that the extension tubes are positioned just beyond the bounds of the slice region to ensure proper results.

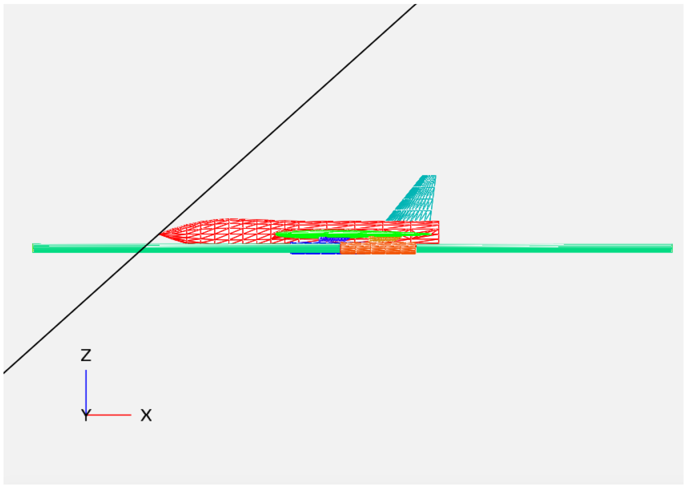


Figure 5.15: Black line depicts Mach 1.5 cutting plane at slicing start location, with engine component inlet extension tubes extending forward of the slicing bounds.

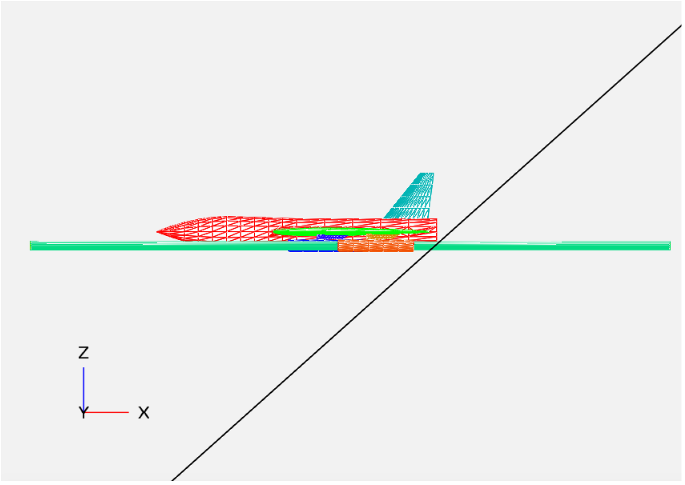


Figure 5.16: Black line depicts Mach 1.5 cutting plane at slicing end location, with engine component inlet extension tubes extending aft of the slicing bound.

In implementing of the wave drag tool, these flow-through areas are handled as in Approach C of Section 4.2 to permit wave drag evaluation and visualization of the results.

5.4 Utilizing the Fourier Sine Series

The Mach cutting plane slicing routine provides area data not only for the wave drag evaluation, but affords the opportunity to present itself visually to the user as well. This data is in the form of the discrete area values along the aircraft axis for each θ rotation. However, the Fourier terms of the Eminton-Lord approach have been made available through `WaveDragEL` such that they may be used in conjunction with the area data. Recall from Chapter 3 that these Fourier terms allow a curve to be constructed that approximates the location of the discrete data and produces the minimum wave drag shape through those points, as well.

By modifying the Eminton-Lord routine in `WaveDragEL`, access was granted to the λ terms of Eq. 3.6 such that the area distribution of Eq. 3.7 may be calculated in the wave drag tool. Sufficient sample points are used to ensure a smooth curve. Had the Fourier terms not been utilized, plotting area data would be limited to connecting discrete data linearly, as seen in Figure 5.17. The same data with the Fourier series curve in seen in Figure 5.18.

Having calculated the a_r terms for Eq. 3.7, Equation 7.2 was implemented as well in order to pursue the maximum wave drag contribution location elaborated upon in Section 4.1. For each θ rotation, the x-location of the maximum S'' magnitude is saved, along with the θ index of the global maximum S'' value. Use of this second derivative information is further described in Chapter 7.

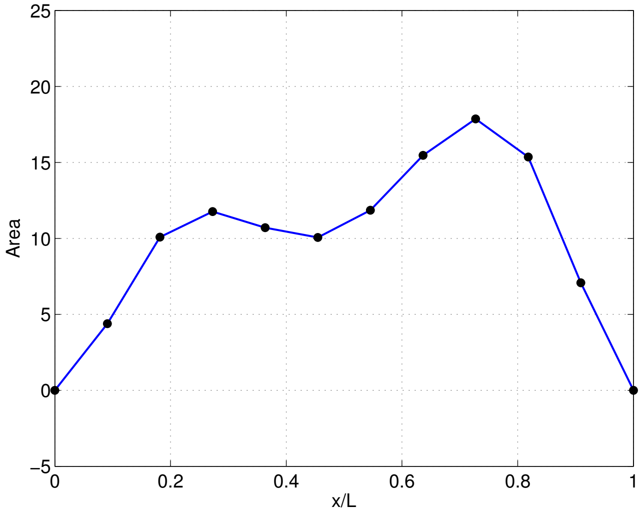


Figure 5.17: Cross-sectional area plot with discrete data connected linearly.

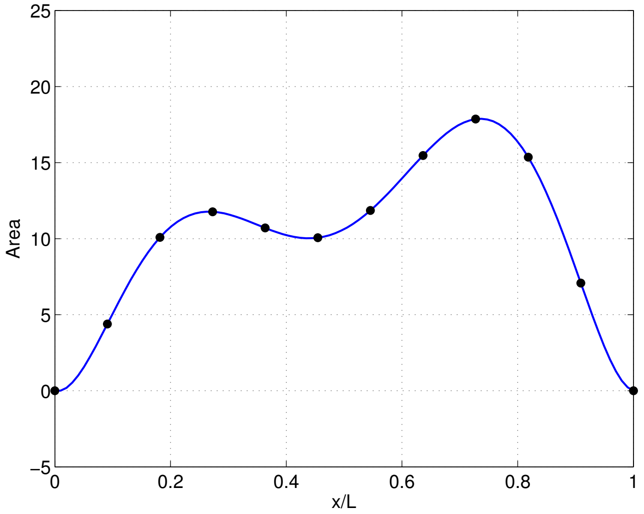


Figure 5.18: Cross-sectional area plot with discrete data connected by Fourier approximation curve.

6 USER INTERFACE DESIGN

Designing the user interface is a critical step in creating an interactive tool. Poor design decisions can hinder usefulness, mask capabilities behind a veil of confusion, and limit potential for users to gain valuable aircraft design intuition from the tool. Careful consideration was given to the design of the wave drag tool GUI. Improvements to the GUI were made throughout the tool's construction as extensive interaction with the tool constantly shaped the author's understanding of how best to shape the user experience.

Interactions with the wave drag tool were separated into three general categories: inputs necessary for the tool run, interfacing with subsurfaces for any desired flow-through designations, and controls for the visualization options.

Among the inputs are all the parameters the wave drag tool requires to run: number of desired Mach cutting planes, number of θ rotations, and Mach number are needed for the slicing routine; a reference area parameter allows for the drag coefficient to be calculated from the wave drag evaluation.

With subsurfaces being used for implementing flow faces on solid components, an interface for designating these flow-through subsurfaces was necessary. Not all subsurfaces in a model may be intended as flow faces (i.e., some may be control surfaces) and the functionality to select specific subsurfaces in a list of all existing subsurfaces was required. Whereas the input information of the previous group is necessary for any implementation of a wave drag tool, the required interface for subsurface designation is purely a result of this tool's implementation in OpenVSP.

Interactiveness was highly valued in the development of this tool. Visualization of the cross-sectional area results grants a unique understanding about the aircraft model and is a stellar complement to a simple table of numerical results. Much attention was given to ensure that a useful display of the cross-sectional data was available with the wave drag tool. For continuity, and the simplicity allowed by housing the entire functionality of the wave drag tool in one interface, the cross-sectional area plot was granted space in the same GUI as the rest of the tool controls— this decision was the primary driver behind the overall size of the wave drag tool GUI, which is on par with some of the larger tool GUIs in OpenVSP, such as the GUI for advanced parameter linking.

As each value of θ for which the tool is run produces a different cross-sectional area distribution, flexibility in the plotting process was necessary to allow the user to switch between which θ rotation's results are visible in the plot. Other desired functions of the visualization, such as ideal body of revolution curves and a Mach cutting plane visualizer, also required controls that were best associated with the plot controls under the umbrella of visualization.

Lastly, space was required to provide both the control to execute the tool and display of the wave drag evaluation results.

OpenVSP's user interface library, Fast Light Toolkit (FLTK), provided the necessary tools to construct the wave drag tool GUI [28] while two-dimensional plotting library Cartesian permitted extra visualization ability [29].

7 TOOL IMPLEMENTATION

The sections that follow describe the wave drag tool as it was implemented using the methods, analyses, and design considerations elaborated upon in the previous chapters. A working example is employed to demonstrate use of the tool.

7.1 Accessing the Wave Drag Tool

Upon running OpenVSP, the wave drag tool may be accessed from the "Analysis" pull-down menu at the top of the main screen, as illustrated in Figure 7.1. As stated in Chapter 5, the introduction of the wave drag tool made obsolete the "Awave Slice" tool— Figure 7.1 reflects the removal of the Awave Slice tool from the OpenVSP source and, hence, the "Analysis" menu. The basic supersonic aircraft model of Figure 5.6 is used for clarifying examples in this chapter.

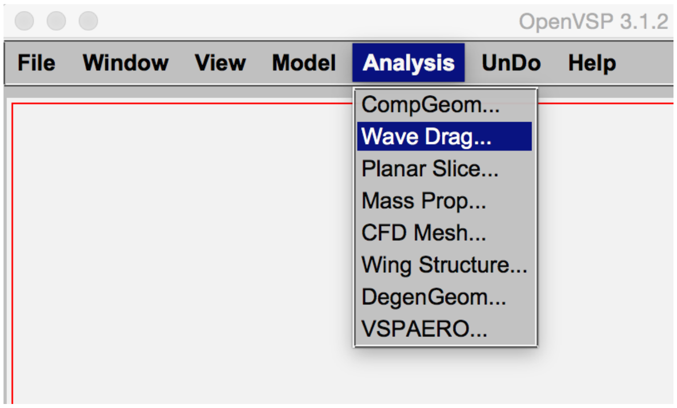
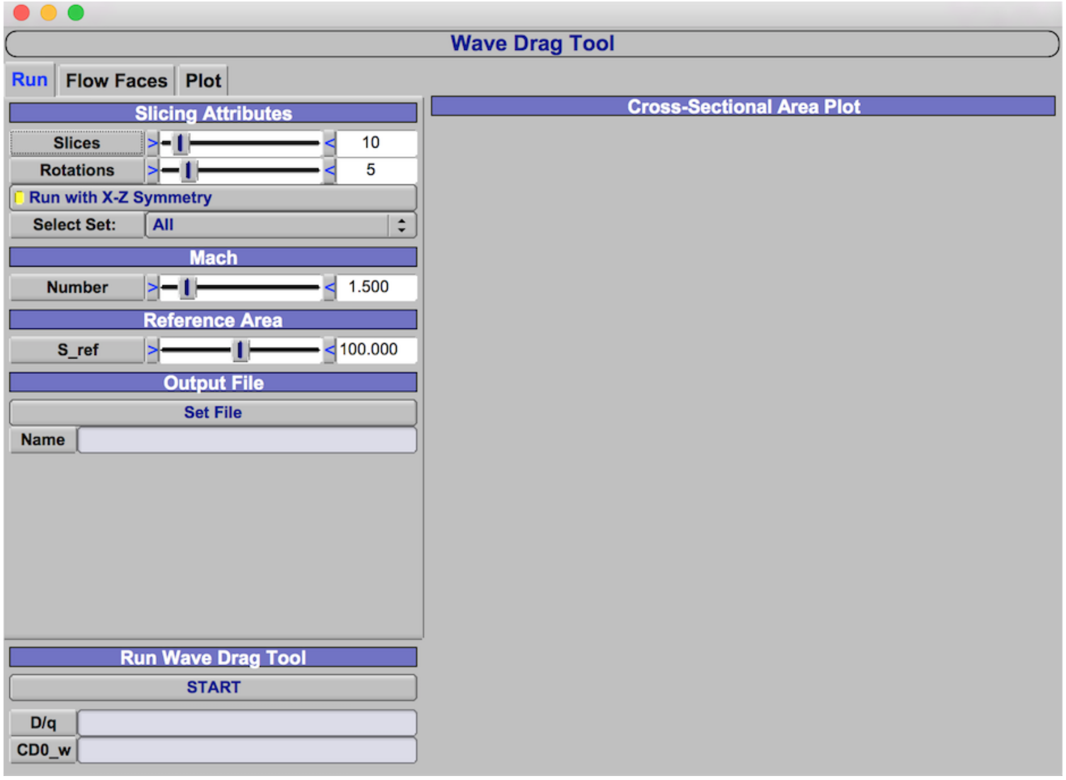


Figure 7.1: Highlighting the Wave Drag option from the OpenVSP Analysis pull-down menu.



Slices 10
Rotations 5
Select Set: All
Number 1.500
S_ref 100.000
Name
D/q
CD0_w

Figure 7.2: Default wave drag tool GUI.

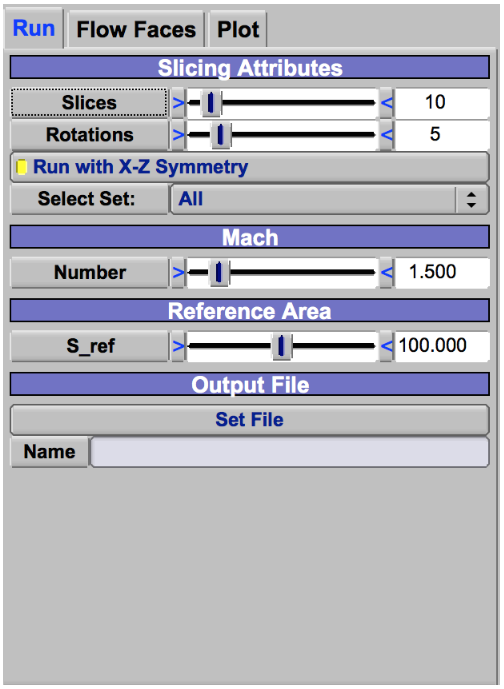
After selecting “Wave Drag Tool” from the list of options in the “Analysis” pull-down, the wave drag tool GUI appears in its default opening state, as shown in Figure 7.2. As the tool has not yet been executed in this image, the data does not yet exist to populate the results fields nor the cross-sectional area plot. Once the wave drag calculation has run once, results from the tool exist and the cross-sectional area plot will be available for the remainder of the OpenVSP session.

The sections that follow detail the elements of the GUI.

7.2 “Run” Tab

The interface for the necessary inputs to the wave drag tool was created in the “Run” tab of the wave drag tool GUI, as seen in Figure 7.3.

User controls for the number of slices per θ rotation and the number of θ rotations are given under the “Slicing Attributes” header. Also shown is a toggle button permitting the



Slices 10

Rotations 5

Run with X-Z Symmetry

Select Set: All

Number 1.500

S_ref 100.000

Name

Figure 7.3: “Run” tab of the wave drag tool GUI.

tool to be run with or without $X-Z$ symmetry. When this symmetry option is turned on, as is the default, the wave drag tool rescales the distribution of the θ rotations on a $0-180°$ basis rather than $0-360°$ as the two halves of the available axis rotation contain identical geometry. The advantage of applying this option in cases of $X-Z$ symmetry is achieving the same fidelity with fewer rotations.

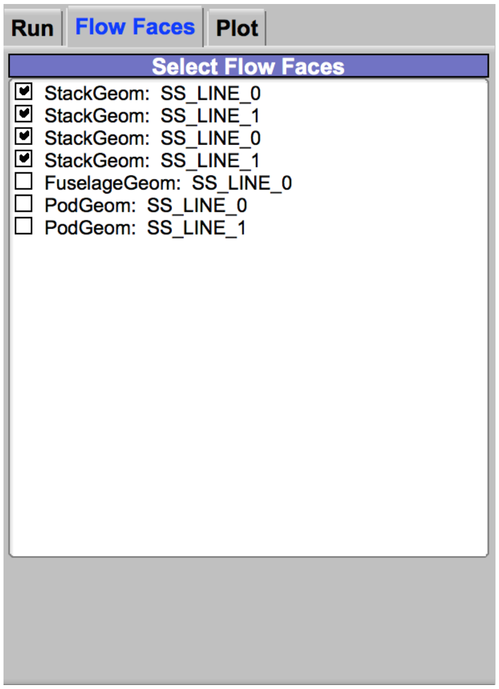
With fewer rotations, the symmetry option can achieve the same result without the symmetry option with an even number n of rotations; the impossibility of having a fraction of a rotation precludes equal results when the non-symmetry number of rotations n is odd.

OpenVSP allows users to place components of their model in different component collections, or “sets”— the wave drag tool GUI allows the user to access these sets via a pull-down menu at the bottom of the “Slicing Attributes” section.

The middle portion of the “Run” tab is dedicated to setting the Mach number and reference area. Mach angle is computed internally as Mach number is more intuitive for a drag analysis. Using the reference area, the zero-lift wave drag coefficient, $C_{D0,wave}$, can be reported in addition to the D/q value.

Lastly, the option to save the resulting cross-sectional areas as a text output is given with a file navigator. The “Set File” button opens a file browser and file naming window from which the user dictates a *.txt file in which the cross-sectional area data will be saved. The file string is displayed in the “Name” box. Note that selecting a file is an optional process; not selecting a file simply means the area data will not be saved to a file. This text output is a more complete and correct version of the output from the old “Awave Slice” tool.

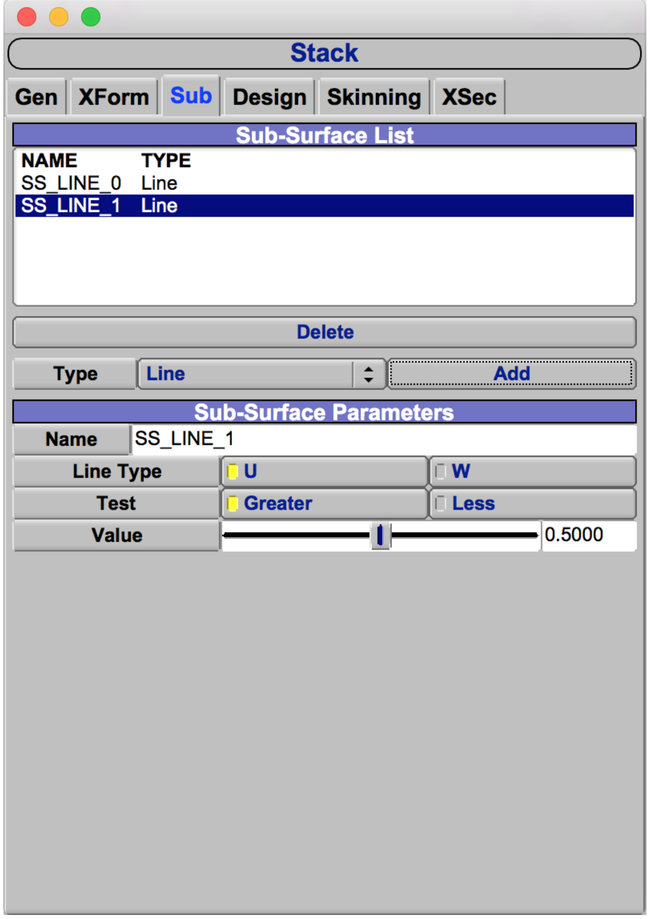
7.3 “Flow Faces” Tab



☑ StackGeom: SS_LINE_0
☑ StackGeom: SS_LINE_1
☑ StackGeom: SS_LINE_0
☑ StackGeom: SS_LINE_1
□ FuselageGeom: SS_LINE_0
□ PodGeom: SS_LINE_0
□ PodGeom: SS_LINE_1

Figure 7.4: “Flow Faces” tab of the wave drag tool GUI, with example subsurfaces shown as a demonstration.

An example of the “Flow Faces” tab in use is shown in Figure 7.4. Using flow faces on solid components in the wave drag tool begins first with placing subsurfaces on the components intended to be flow-through. This is done using the subsurface interface for the component, which is shown in Figure 7.5. The user must dictate whether the subsurface lies outside or inside the subsurface line by selecting “Greater” or “Less” from the subsurface menu. With the Line subsurface, these options are with respect to *x*-location— with the "Greater" option, the portion of the component located in the positive *x*-direction from the subsurface line will be designated as the subsurface, while the opposite is true for the "Less" option. For example, to create inlet and exit subsurfaces on an engine component, the user would add a subsurface Line with "Less" at the inlet location and a subsurface Line with "Greater" at the exit location.



NAME	TYPE
SS_LINE_0	Line
SS_LINE_1	Line

Type: Line

Name: SS_LINE_1

Line Type: ● U ○ W

Test: ● Greater ○ Less

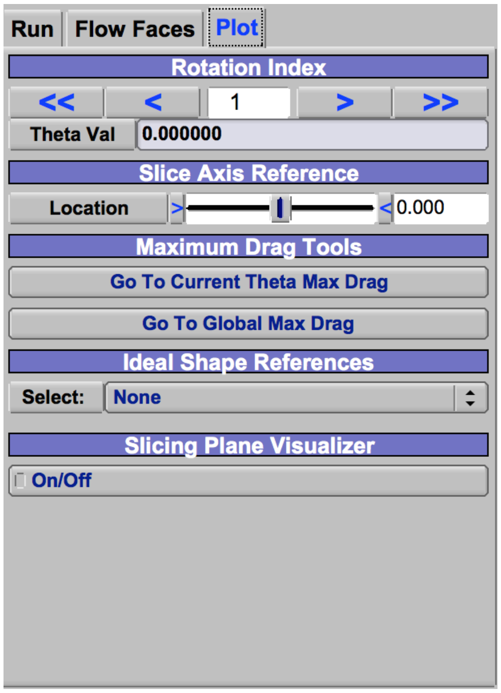
Value: 0.5000

Figure 7.5: Subsurface tab on a stack component.

The user communicates to the wave drag tool which subsurfaces are to be considered flow-through by using the "Flow Faces" tab of the wave drag tool GUI. This tab contains a checklist window of all subsurfaces in the geometry in which the default condition of the checkboxes is unchecked. Checking the boxes next to the subsurfaces to be used as flow faces is the only action required by the user. As described in Chapter 5, the wave drag tool determines whether the subsurface is an inlet or an exit. In Figure 7.4, four subsurfaces have been selected as flow faces.

7.4 “Plot” Tab

Controls for managing the visual interaction tools were segregated into the “Plot” tab of the wave drag tool GUI, shown in Figure 7.6.



Rotation Index: 1

Theta Val 0.000000

Location 0.000

Select: None

Figure 7.6: “Plot” tab of the wave drag tool GUI.

A rotation index selector allows the user to select which of the available θ rotation cross-sectional area plots to visualize. Internally, the θ values are in radians and their quantity is stored as an index of integers, neither of which are intuitive to the user for where the cutting plane is located. Thus, an additional window is provided to display the value, in degrees, of the currently selected θ. This result is displayed to the “Theta Val” box.

A visual indicator of the current *x*-location of interest is shown on the cross-sectional area plot. The *x*-location of this indicator is controlled by the slider under Slice Axis Reference header.

The Maximum Drag Tools section contains two trigger buttons that enact the abilities of Section 4.1 to locate the maximum drag contribution locations across all θ values. The "Go to Current Theta Max Drag" button permits the user to instantly relocate the visual indicator bar to the x-location of maximum wave drag contribution on the current θ value. The "Go To Global Max Drag" button first changes the current θ to the previously determined θ index containing the global maximum wave drag, then relocates the visual indicator bar to the appropriate x-location.

The ability behind these maximum drag buttons follows from the analysis presented in Chapter 4. In Ref. [22], the equation for $S'(x)$ is given as

$$S'(x) = \sum_{r=1}^{\infty} a_r \sin r\theta, \quad 0 \leq x \leq 1. \tag{7.1}$$

Thus, differentiation gives

$$S''(x) = \sum_{r=1}^{\infty} a_r r \cos(r\theta) \frac{1}{\sqrt{-(x-1)x}} \tag{7.2}$$

where the rightmost term accounts for differentiating the transformation in Eq. 3.8.

Eq. 7.2 is utilized in the manner prescribed in Chapter 4 in which the x-value corresponding to the largest magnitude of S'' is the location of maximum wave drag contribution. In the wave drag tool, these x-values are determined for each set of Mach cutting planes over the dictated θ rotations as to permit the use of both maximum drag buttons in the "Plot" tab.

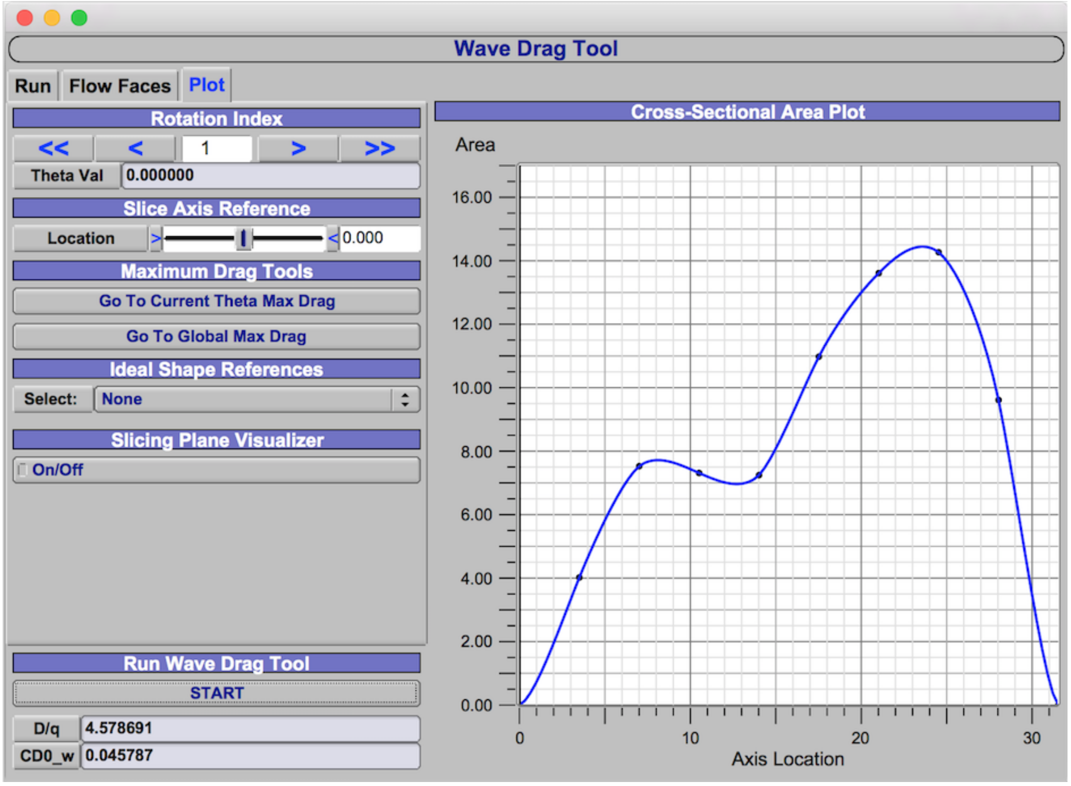
The "Ideal Shape References" pull-down menu allows the user to select from a given list of available bodies of revolution whose cross-sectional area distributions will appear on the cross-sectional area plot along with the distribution for the existing aircraft model. The default is to display none of these curves, but the list contains: 1) Sears-Haack body; 2) von

Kármán ogive; and 3) Lighthill's body. All three are bodies that use length and one other parameter to distribute area from nose to tail [19]. These curves are useful for comparing to the area distribution of the aircraft model and are discussed further in the following section.

The last option of the "Plot" tab is a toggle button to turn on or off a Mach cutting plane visualizer on the main screen. As shown, the default condition of this toggle is off. The cutting plane visualizer is discussed in the final section of this chapter.

7.5 Cross-Sectional Area Visualization

A two-dimensional graphics library was incorporated into the OpenVSP source for use in visualizing the cross-sectional area data [29]. Each time the wave drag tool GUI is updated, the plot is redrawn to reflect any changes.



Rotation Index: 1

Theta Val: 0.000000

Location: 0.000

Select: None

□ On/Off

D/q: 4.578691

CD0_w: 0.045787

Figure 7.7: Wave drag tool GUI as it appears after execution with the fighter model working example.

Once the tool has been executed, the wave drag tool GUI retrieves the axis location and cross-sectional area results for the current θ selected in the "Plot" tab. The minimum and maximum values of the x-axis are determined by the minimum and maximum values of the axis locations used in the slicing of the commanded θ value. The minimum of the y-axis is zero, for zero cross-sectional area, and the maximum value is the global maximum value of cross-sectional area for all θ, plus a buffer to disallow any data from being plotted at the very top of the plot.

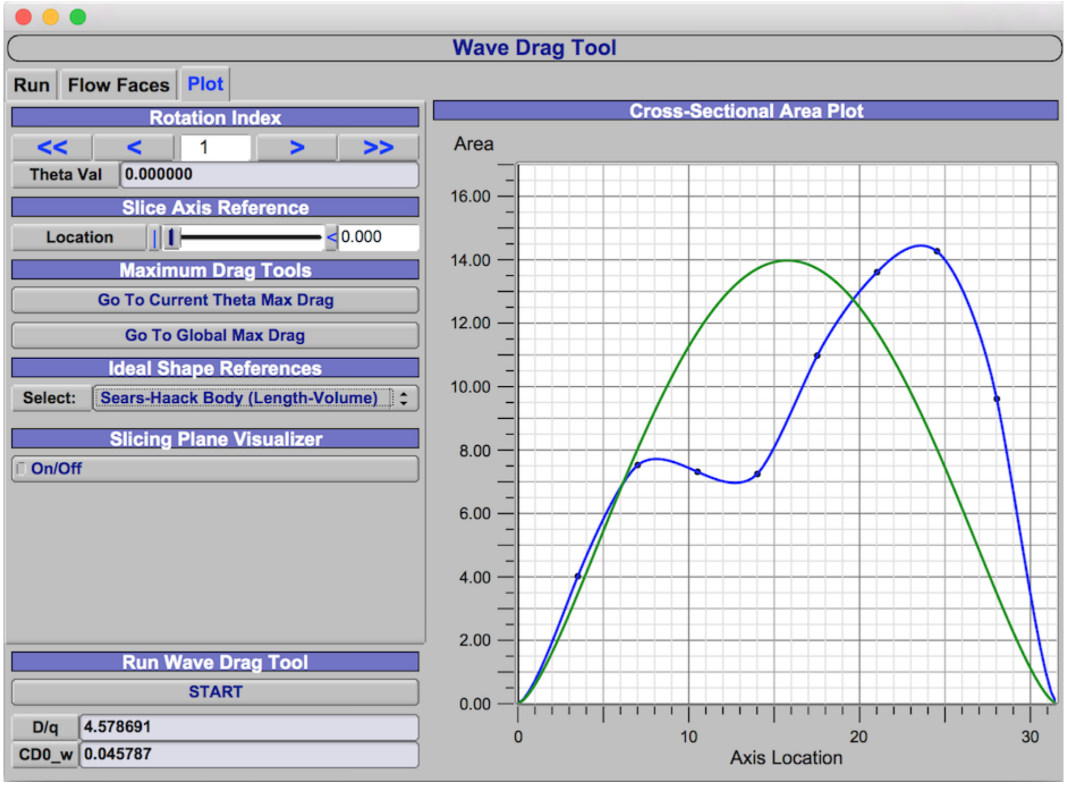
To illustrate the interaction with the cross-sectional area plot, the fighter model example shown in Figure 5.6 is run through the wave drag tool. Here, the tool was commanded to run with 10 slices on five θ rotations at a Mach number of 1.5; as the aircraft is $X-Z$ symmetric, the symmetry option was utilized.

The selected plot for viewing in Figure 7.7 is that of the first θ index, which is shown to correspond to 0°. Using the axis location and cross-sectional area as (x,y) coordinates, the data for the current θ is then plotted as black points on the canvas. As discussed in Chapter 5, the Fourier terms of the Eminton-Lord approach are used to create the smooth curve that approximates the discrete values from the area calculation, shown in blue on the cross-sectional area plot.

Selecting a body of revolution curve from the "Ideal Shape References" menu plots the selected curve in green on top of the existing data.

The Sears-Haack body is the minimum wave drag shape for a given length and volume and is given by Eq. 7.3 [19]. In the wave drag tool, the length is obtained from the x-wise span of the current θ value; the volume is calculated as the integral of the area results on the current θ. The result is an equivalent Sears-Haack body created to match the geometry data from the aircraft model. The Sears-Haack selection is shown plotted against the fighter model example in Figure 7.8.

$$S(x) = \frac{16V}{3L\pi}\left[4\frac{x}{L}(1-\frac{x}{L})\right]^{\frac{3}{2}} \tag{7.3}$$



Rotation Index: 1

Theta Val: 0.000000

Location: 0.000

Select: Sears-Haack Body (Length-Volume)

D/q: 4.578691

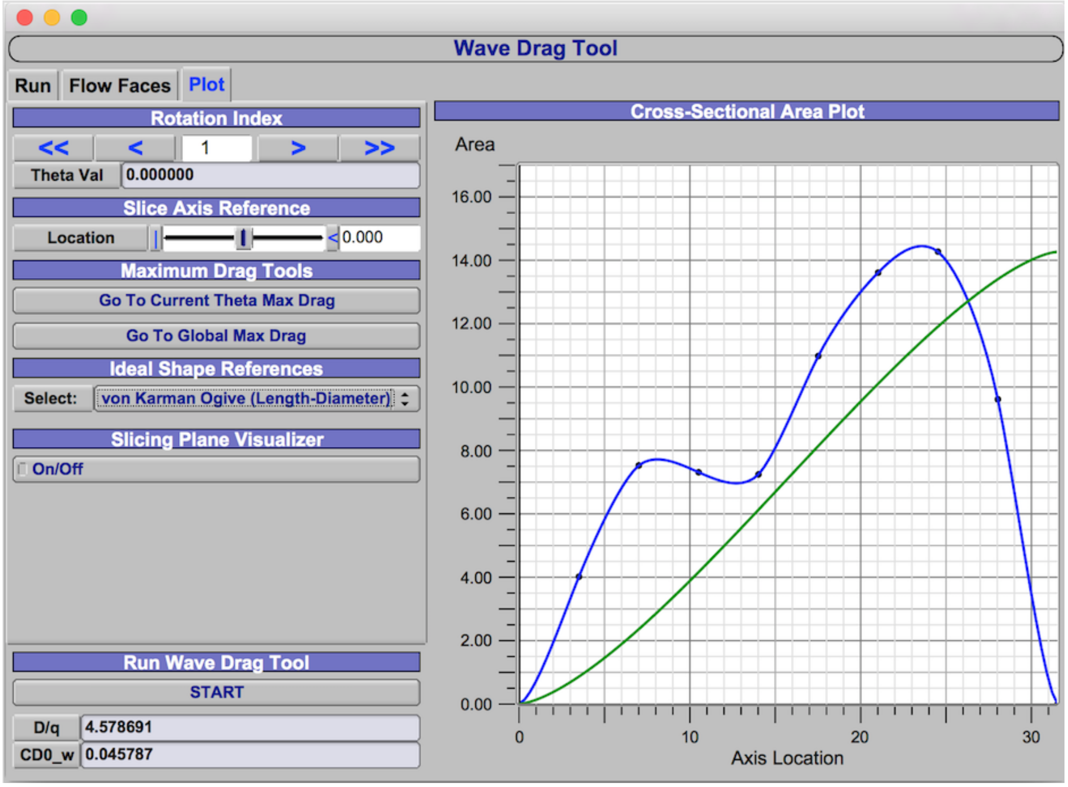
CD0_w: 0.045787

Figure 7.8: Overlay of the equivalent Sears-Haack body cross-sectional area curve selected in the fight model working example.

The von Kármán ogive is most commonly referenced in nose cone design. This body uses a given length and maximum diameter to produce the minimum wave drag shape for when the maximum diameter is located at the base. The equation for the von Kármán ogive is given in Eq. 7.4 [19]; again, the length parameter used is the x-wise span of the current θ value. The fighter model working example is shown with the von Kármán ogive in the wave drag tool run in Figure 7.9.

$$\left(\frac{r}{r_b}\right)^2 = \frac{2}{\pi}\left[\arcsin\sqrt{\frac{x}{L}} - \left(1-2\frac{x}{L}\right)\sqrt{\frac{x}{L}\left(1-\frac{x}{L}\right)}\right] \tag{7.4}$$

Lighthill's body is very similar to the von Kármán ogive, with the length and diameter being specified. However, the maximum diameter location is moved to the midpoint. Certain assumptions are made about the slenderness of the body and sufficiently low super-



Rotation Index: 1

Theta Val: 0.000000

Location: 0.000

Select: von Karman Ogive (Length-Diameter)

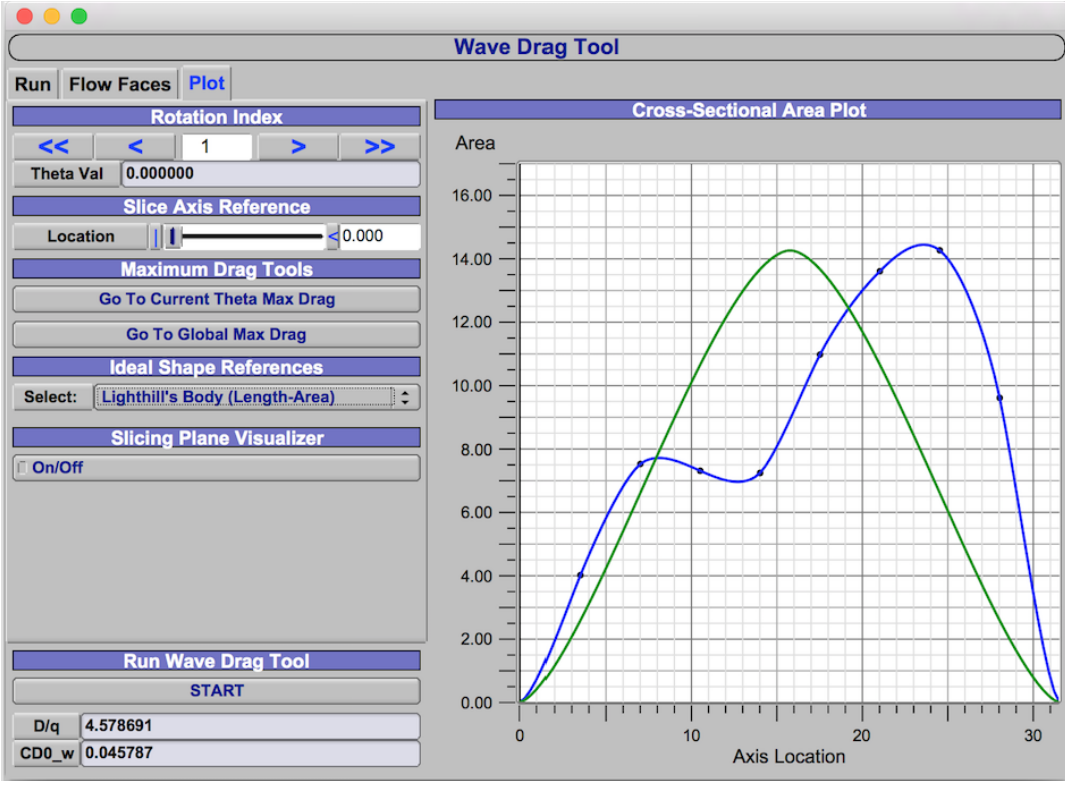
D/q: 4.578691

CD0_w: 0.045787

Figure 7.9: Overlay of the equivalent von Kármán body cross-sectional area curve selected in the fight model working example.

sonic Mach numbers in generating Lighthill's body equation and are covered in Ref. [19]; the equation is given in Eq. 7.5. The Lighthill's body curve versus that of the example fighter model is shown in Figure 7.10.

$$\left(\frac{r}{r_b}\right)^2 = 2\sqrt{\frac{x}{L}\left(1-\frac{x}{L}\right)} - \left(1-2\frac{x}{L}\right)^2 \cosh^{-1}\left|\frac{1}{\left(1-2\frac{x}{L}\right)}\right| \tag{7.5}$$



Rotation Index: 1

Theta Val: 0.000000

Location: 0.000

Select: Lighthill's Body (Length-Area)

D/q: 4.578691

CD0_w: 0.045787

Figure 7.10: Overlay of the equivalent Lighthill's body cross-sectional area curve selected in the fight model working example.

7.6 Cutting Plane Visualizer

The author relied on OpenVSP's OpenGL graphics library for the creation of the cutting plane visualizer. A simple rectangular plane was created and given various coloring and lighting parameters such that the cutting plane visualizer takes on a gray, translucent nature, not unlike that of tinted glass. This rectangle is outlined in black using the OpenGL line loop method.

Three orientation parameters are applied to the plane coordinates before the OpenGL objects are created: x-axis location, Mach angle, and θ value. The values for these orientation parameters are obtained from corresponding values in the wave drag tool GUI and passed to the cutting plane visualizer generator in `WaveDragMgr`. Figure 7.11 shows this cutting plane visualizer at the $x = 19.777$ location for the fighter model working example.

The Slice Axis Reference slider previously mentioned controls the location of the cutting plane visualizer, as well. The black vertical reference bar is always located at the same x-location as the cutting plane visualizer, as can be seen in Figure 7.12 for the $x = 19.777$ case aforementioned.

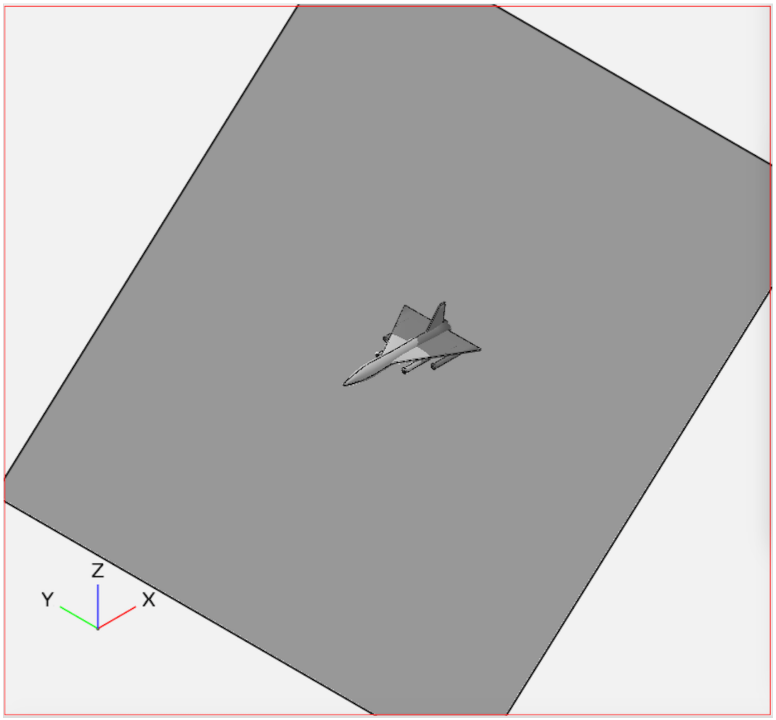
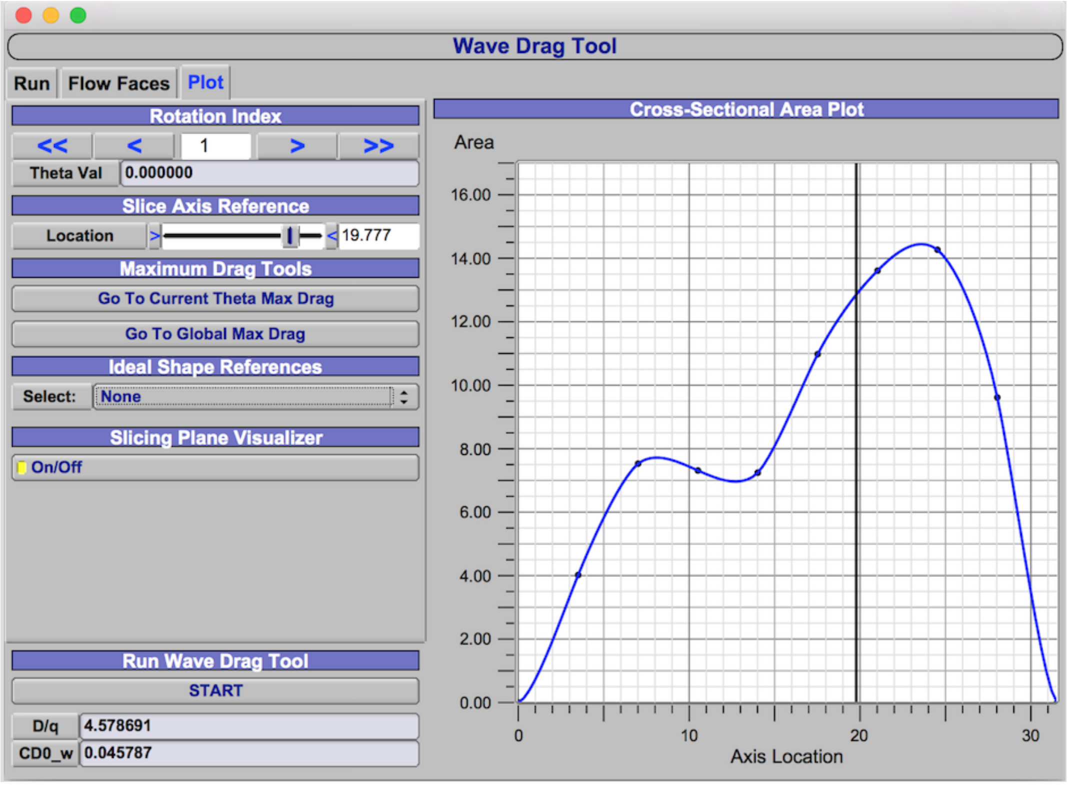


Figure 7.11: Fighter model working example with Mach cutting plane visualizer activated. The visualizer has been arbitrarily relocated to $x = 19.777$ for demonstrative purposes.



Rotation Index: 1

Theta Val: 0.000000

Location: 19.777

Select: None

D/q: 4.578691

CD0_w: 0.045787

Figure 7.12: Wave drag tool GUI with Slice Axis Reference at $x = 19.777$. The black vertical reference bar and cutting plane visualizer are translated to this x-location.

8 COMPARISON TO ANALYTICAL SOLUTIONS

Analytical solutions to the wave drag integral served as important verification checkpoints for implementation of the wave drag tool. Such solutions allowed the results of this tool to be compared to their analytical solutions and determine the convergence of tool's solution upon the analytical solution when resolution of the Mach cutting plane routine is increased.

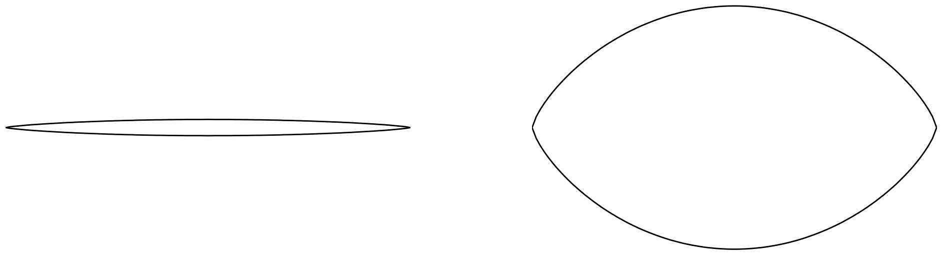
8.1 Sears-Haack Body

The analytical definition of the Sears-Haack body area distribution makes obtainment of an analytical solution for zero-lift wave drag simple. The equation for the wave drag on a Sears-Haack body is given in Eq. 8.1.

$$\frac{D}{q} = \frac{9\pi^3 r_{max}^4}{2L^2} \tag{8.1}$$

Sears-Haack body meshes were generated in MATLAB for r/L ratios 0.02 to 0.3. The side profiles of these two bounding Sears-Haack bodies are shown in Figure 8.1.

These Sears-Haack meshes, saved as Hermite files and imported into OpenVSP, were run through the wave drag tool and the resulting D/q values recorded. These results are plotted along with their corresponding analytical solutions from Eq. 8.1 in Figure 8.2. In all cases, the wave drag tool was executed with 50 Mach cutting plane slices and a Mach number of 1. The axis-symmetric nature of the body precludes dependency on the number of rotation sections.



(a) $r/L = 0.02$

(b) $r/L = 0.3$

Figure 8.1: Smallest and largest r/L ratios over which meshes were generated for use in OpenVSP.

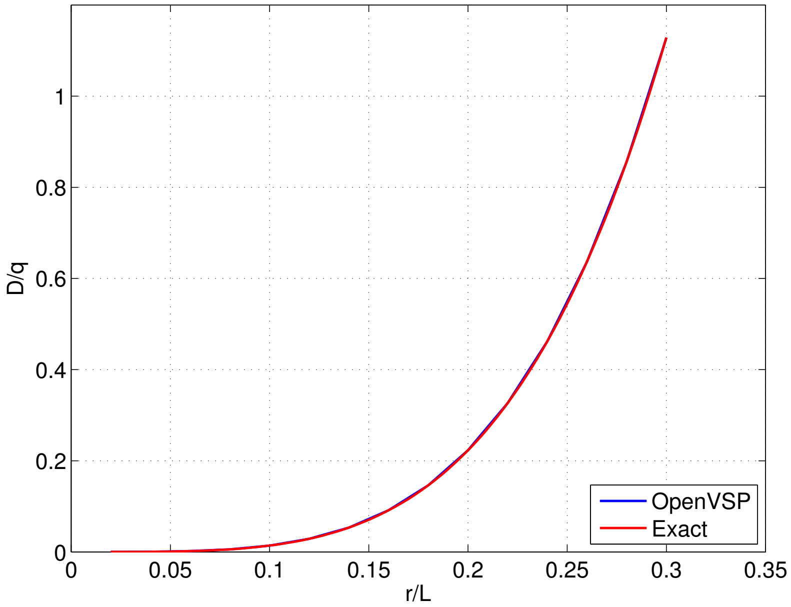


Figure 8.2: Drag results from a sweep of Sears-Haack bodies through the wave drag tool compared to the analytical solution.

As can be seen in Figure 8.2, the results are essentially identical. The error between the wave drag tool and analytical solution is shown on a logarithmic scale in Figure 8.3. Error on the Sears-Haack body drag results remains consistently less than 0.04%. This error is likely contributed by rounding error in the area calculation routine and wave drag

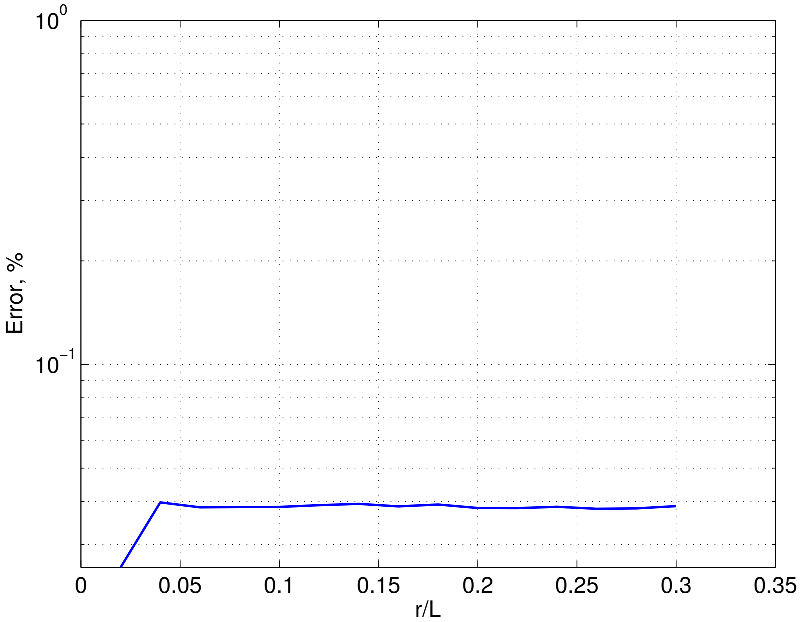


Figure 8.3: Error on the drag results from the sweep of Sears-Haack bodies through the wave drag tool.

evaluation, as well as by the linear discretization of the continuous Sears-Haack function in the Hermite mesh.

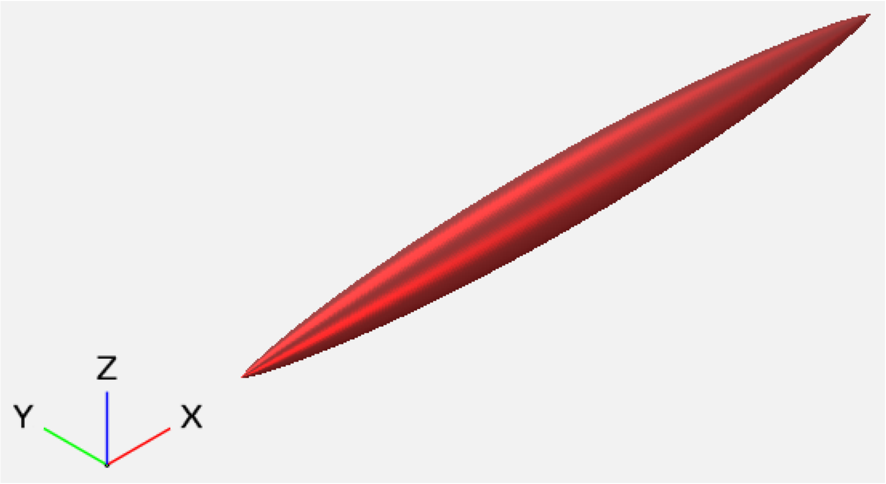


Figure 8.4: OpenVSP model of a Sears-Haack body ($r/L = 0.04$) after being imported from a Hermite file.

A convergence study was conducted on a Sears-Haack body of $r/L = 0.04$ to explore the effect of the number of slices on the resulting drag. This body in OpenVSP is shown

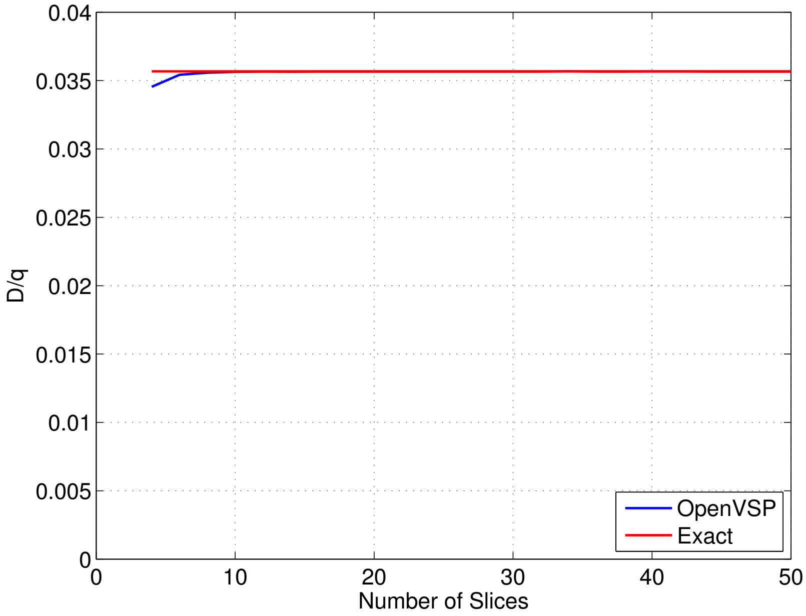


Figure 8.5: Convergence study on the number of Mach cutting plane slices versus the wave drag result compared to the analytical solution for the Sears-Haack body.

in Figure 8.4. The number of slices was varied from four to 50, with the resulting D/q values recorded for comparison to the analytical solution— this comparison is shown in Figure 8.5. Here, the error went below 1% after six slices, and below 0.1% after 10, which can be seen clearer in Figure 8.6.

The results from running a variety of Sears-Haack bodies demonstrated that the wave drag tool functions as expected. The cross-sectional areas were obtained correctly and the wave drag integral was evaluated properly, enabling the results from the tool to match so closely to the analytical solutions. Missteps in either would have become obvious. However, the Sears-Haack body presents a simple case in which the geometry is very smooth and evenly distributed. One would expect to achieve favorable results from a wave drag tool for body that was designed for minimum wave drag. In the following section, a non-ideal body of revolution is considered.

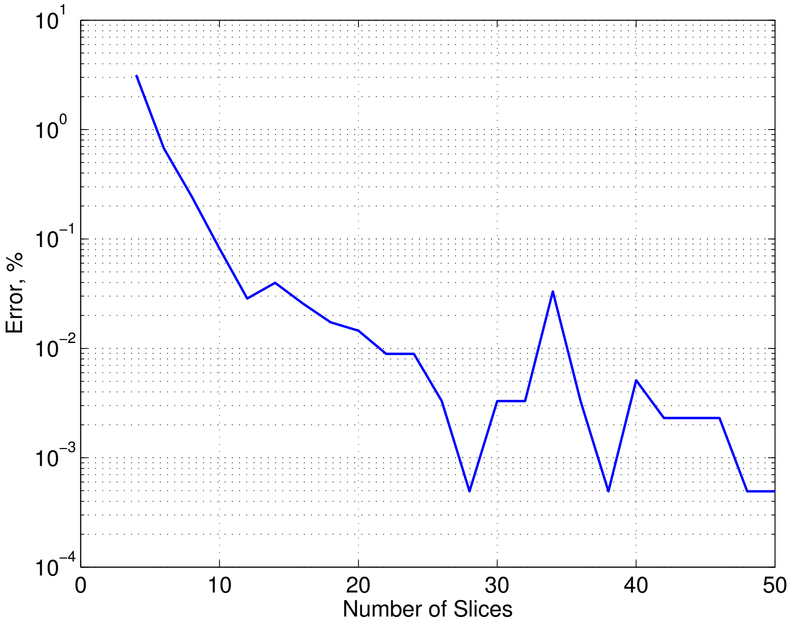


Figure 8.6: Convergence study error for the Sears-Haack body ($r/L = 0.04$).

8.2 Non-Ideal Body Examination

In Ref. [22], Eminton and Lord present an equation for a body of revolution that is similar to that of a fuselage with a swept wing. This equation, in terms of cross-sectional area, is given in Eq. 8.2.

$$S\left(\frac{x}{L}\right) = 10\left(\frac{x}{L}\right)^{2}\left[400\left(\frac{x}{L}\right)^{4} - 1176\left(\frac{x}{L}\right)^{3} + 1257\left(\frac{x}{L}\right)^{2} - 588\left(\frac{x}{L}\right) + 108\right] \tag{8.2}$$

This cross-sectional area distribution is such that the zero-lift wave drag on this body can be found using Eq. 8.3. Eminton and Lord prescribe a body of length 40 units, such that Eq. 8.3 resolves to very nearly eight (7.997535890367740) at Mach 1.

$$\frac{D}{q} = \frac{1}{\pi L^{2}}40200 \tag{8.3}$$

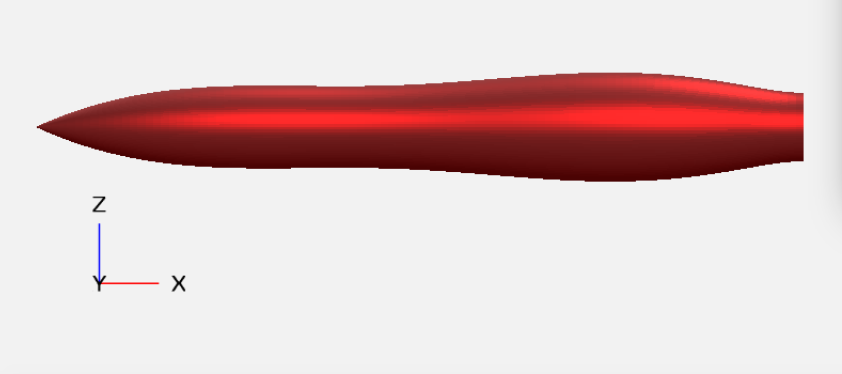


Figure 8.7: OpenVSP model of an Eminton and Lord body ($L = 40$) **after being imported from a Hermite file.**

Using MATLAB to generate the Hermite file for this body, it was imported into OpenVSP. The resulting model is shown in Figure 8.7.

A convergence study similar to that of the Sears-Haack bodies was conducted on the Eminton and Lord body, the results of which are shown in Figure 8.8. As expected, without the smooth and optimized shape of the Sears-Haack body, the Eminton and Lord body required the use of more Mach cutting plane slices before convergence occured. Here, 34 slices were required to reduce the error to less than 1%. A logarithmic plot of the error is shown in Figure 8.9.

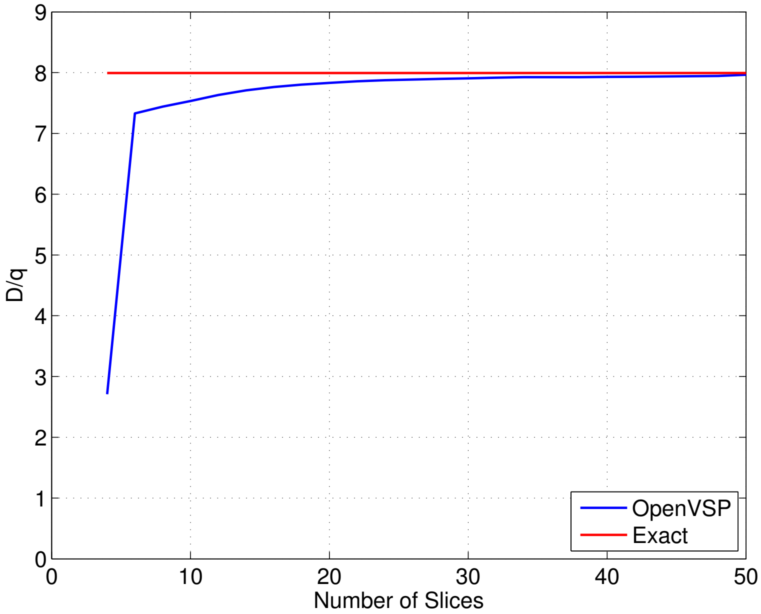


Figure 8.8: Convergence study on the number of Mach cutting plane slices versus the wave drag result compared to the analytical solution for the Eminton and Lord body.

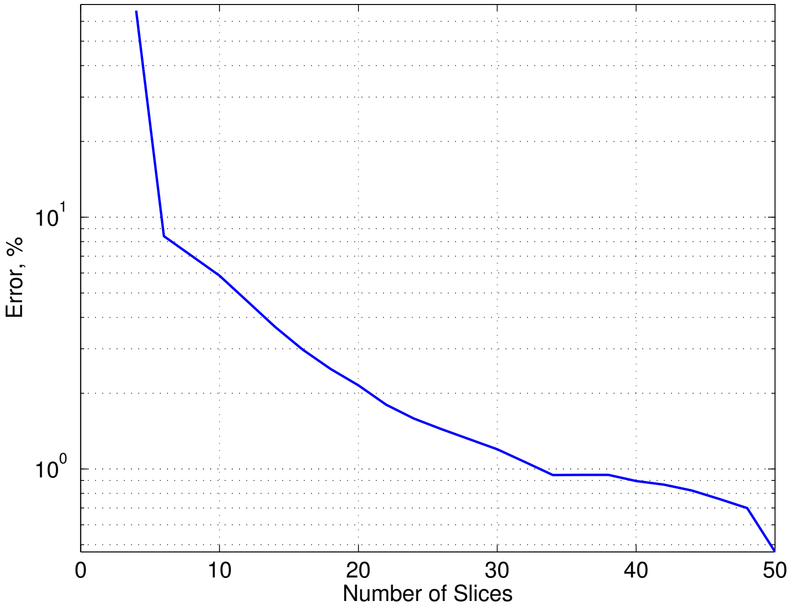


Figure 8.9: Convergence study error for the Eminton and Lord body.

8.3 Note on Analytical Flow-Through Solutions

An analytical solution to the zero-lift wave drag on a body with a flow-through portion can be attained with the understanding that the wave drag evaluation will remain unchanged so long as the second derivative of the cross-sectional area distribution stays the same. Beginning with a body with a known continuous function for area distribution, it is simple to apply modifications to said equation that create a body of revolution with a hollow center without changing the area distribution.

The equation for a Sears-Haack body in terms of radius is given in Eq. 8.4. With the constraint that the area at each value of x remain unchanged, an equation for a new definition of the outer radius can be created, as shown in Eq. 8.5.

$$r = r_{max}\left(4\frac{x}{L}\left(1-\frac{x}{L}\right)\right)^{\frac{3}{4}} \tag{8.4}$$

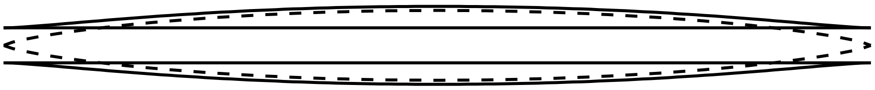


Figure 8.10: Shape of two Sears-Haack bodies of equal cross-sectional area distribution with one containing a flow-through section. Flow-through is of $r/L = 0.02$.

$$r_o = \sqrt{r^2 + r_i^2} \tag{8.5}$$

As the cross-sectional area distribution has remained unchanged, Sears-Haack bodies of the same length and equivalent r/L will produce the same second derivative of cross-sectional area. As per the discussion in Chapter 4, this equivalency also results in equal wave drag evaluations. The $r/L = 0.04$ case from Section 8.1 is compared to its equivalent flow-through Sears-Haack body with a flow-through portion of $r/L = 0.02$ in Figure 8.10.

The unmodified Sears-Haack body is shown in dashes for improved clarity. These Sears-Haack body shapes, despite having dissimilar outer radii, return the same results from area calculations and wave drag evaluations. The wave drag tool results for the flow-through Sears-Haack body are within 0.016% of the analytical wave drag solution for an unmodified Sears-Haack body of the same equivalent r/L.

9 CONCLUSION

Computer codes for estimating wave drag have existed for decades, but the inclusion of such a code in an active, open source geometry tool provides an opportunity to make visually-interactive wave drag mitigation available to more designers.

By delegating area calculations to the geometry engine of OpenVSP, the component intersection limitations experienced with tools such as the Harris code are no longer present. Integration of the wave drag calculation into the geometry tool creates a seamless hand-off of the geometry data. While minor geometry changes could be accommodated by the Harris code and AWAVE by "stacking" the inputs and using control terms to hold constant the unchanged inputs [11], the OpenVSP user has full, visual control over their model at all times; geometry can be changed in any fashion and the wave drag tool can be rerun in the click of a button.

Rapid visualization of results bolsters the user experience in OpenVSP, while the exporting of area results allows users to work with the data in whichever external manner they please. Comparisons to known analytical wave drag solutions provide validity to the wave drag tool, which produced less than 1% error for analytical bodies of revolution for only a modest number of cutting plane slices.

In the future, this wave drag tool can be expanded to include some form of optimization. As the Eminton-Lord procedure produces the minimum wave drag shape for a discrete set of area values, the resulting smooth Fourier series curve could be used in conjunction with OpenVSP's cross-section definition of components. By defining the area values and their corresponding locations, the `WaveDragEL` routine could be used to obtain the minimum wave drag curve; the area values available from the curve could then be specified to OpenVSP to build or modify component cross-sections to match the area values that lie on the smooth minimum wave drag curve.

This application of the tool could be used to design minimum wave drag fuselages when a handful of important points are defined: starting location, maximum cockpit cross-section and its location, etc. While the wave drag tool would not require many analytical changes, the GUI would need a new element through which the user could define the set of areas and locations for which the curve will be calculated. The largest changes, per the author's current opinion, would be to use the results of the minimum wave drag curve to dictate geometry changes to the model.

In this work, the same linearized theory for zero-lift wave drag from legacy codes has been made accessible to any OpenVSP user, coupled with modern geometry analyses. This interactive wave drag tool is an advancement of OpenVSP's ability to provide geometry-based analysis capabilities in an effort to further the success of the OpenVSP mission.

BIBLIOGRAPHY

[1] John D. Anderson Jr. *A History of Aerodynamics and Its Impact on Flying Machines*. Vol. 8. Cambridge Aerospace Series. New York, NY: Cambridge University Press, 1997.

[2] Lane E. Wallace. "The Whitcomb Area Rule: NACA Aerodynamics Research and Innovation". In: *From Engineering Science to Big Science: The NACA and NASA Collier Trophy Research Project Winners*. Ed. by Pamela E. Mack. Washington, D.C.: NASA History Office, 1998, pp. 135–148.

[3] John D. Anderson Jr. *Introduction to Flight*. Third. New York, NY: McGraw-Hill, 1989.

[4] John D. Anderson Jr. *Fundamentals of Aerodynamics*. 5e. New York, NY: McGraw-Hill, 2011.

[5] *Area Rule and Transonic Flight*. Aerospaceweb. 2012. URL: http://www.aerospaceweb.org/question/aerodynamics/q0104.shtml.

[6] David A. Anderton. "NACA Formula Eases Supersonic Flight". In: *Aviation Week* 63.11 (1955), pp. 12–16.

[7] *Convair YF-102 Delta Dagger Image Gallery*. NASA Armstrong Flight Research Center. 2015.

[8] *The Convair F-102 Delta Dagger*. 456th Fighter Interceptor Squadron. 2015. URL: http://www.456fis.org/F-102.htm.

[9] Roy V. Harris Jr. *An Analysis and Correlation of Aircraft Wave Drag*. NASA TM X-947. NASA, 1964.

[10] *Wave Drag by the Supersonic Area Rule*. Ralph Carmichael, Public Domain Aeronautical Software (PDAS). www.pdas.com/wavedrag.html. 2013.

[11] *AWAVE User's Guide*. L. A. McCullers, NASA Langley Research Center. 1992.

[12] Andrew Hahn. "Vehicle Sketch Pad: Parametric Geometry for Conceptual Aircraft Design". In: *48th AIAA Aerospace Sciences Meeting*. AIAA-2010-657. Orlando, FL: American Institute of Aeronautics and Astronautics, January 4-7, 2010.

[13] *OpenVSP*. 2015. URL: `http://openvsp.org/`.

[14] *OpenVSP Project*. GitHub. 2015. URL: `https://github.com/OpenVSP/OpenVSP`.

[15] James R. Gloudemans and Robert A. McDonald. "User Defined Components in the OpenVSP Parametric Geometry Tool". In: *15th AIAA Aviation Technology, Integration, and Operations Conference*. AIAA-2015-2547. Dallas, TX: American Institute of Aeronautics and Astronautics, June 22-26, 2015.

[16] *AngelScript Manual*. AngelCode. 2015. URL: `http://www.angelcode.com/angelscript/sdk/docs/manual/index.html`.

[17] J.R. Gloudemans. *Scripting and API*. OpenVSP. 2014. URL: `http://www.openvsp.org/wiki/lib/exe/fetch.php?media=workshopv3:scripting_api.pdf`.

[18] *Using VSPAREO*. OpenVSP. 2015. URL: `http://www.openvsp.org/wiki/doku.php?id=vspaerotutorial`.

[19] Holt Ashley and Marten Landahl. *Aerodynamics of Wings and Bodies*. New York, NY: Dover Publications, 1985.

[20] G.N. Ward. "Supersonic Flow Past Slender Pointed Bodies". In: *Journal of Mechanics and Applied Mathematics* 2 (1949), pp. 75–97.

[21] Robert T. Jones. *Theory of Wing-Body Drag at Supersonic Speeds*. Research Memorandum A53H18a. NACA, 1953.

[22] E. Eminton and W. T. Lord. “Note on the Numerical Evaluation of the Wave Drag of Smooth Slender Bodies Using Optimum Area Distributions for Minimum Wave Drag”. In: *Journal of the Royal Aeronautical Society* 60 (1956), pp. 61–63.

[23] Evelyn Eminton. *On the Numerical Evaluation of the Drag Integral*. Report and Memoranda No. 3341. Ministry of Aviation Aeronautical Research Council, 1961.

[24] R.C. Feagin. *Transonic Aircraft Drag Technique Applied to Military Aircraft, Volume II*. Anderson series. Burbank, CA: Lockheed-California Company, 1978.

[25] C.H. Vance. *Standard Aircraft Characteristics Performance of the Northrop F-5E Air Superiority Fighter with Two J85-GE-21 Engines*. Hawthorne, CA: Northrop Corporation Aircraft Division, 1976.

[26] Raymod L. Barger and Mary S. Adams. *Fuselage Design for a Specified Mach-Sliced Area Distribution*. Technical Paper 2975. NASA, 1990.

[27] Alex M. Gary and Robert A. McDonald. “Parametric Identification of Surface Regions in OpenVSP for Improved Engineering Analysis”. In: *53rd AIAA Aerospace Sciences Meeting*. Kissimmee, FL: American Institute of Aeronautics and Astronautics, January 5-9, 2015.

[28] *Fast, Light Toolkit (FLTK), Ver. 2.0*. Bill Spitzak, et. al. 2011.

[29] *Cartesian, Ver. 1.1*. Roman Kantor. 2010.