8-2010

Development and Use of a Computer Program “Hyper-N” to Predict the Performance of Air Vehicles Traveling at Hypersonic Speeds

Younes Baalla
University of Tennessee - Knoxville, ybaalla@utsi.edu

To the Graduate Council:

I am submitting herewith a thesis written by Younes Baalla entitled "Development and Use of a Computer Program “Hyper-N” to Predict the Performance of Air Vehicles Traveling at Hypersonic Speeds." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Aviation Systems.

Peter U. Solies, Major Professor

We have read this thesis and recommend its acceptance:

Stephen Corda, Borja Martos

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Younes Baalla entitled "Development and Use of a Computer Program 'Hyper N' to Predict the Performance of Air Vehicles Traveling at Hypersonic Speeds." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Aviation Systems.

Uwe Peter Solies , Major Professor

We have read this thesis
and recommend its acceptance:

Stephen Corda

Borja Martos

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Development and Use of a Computer Program 'Hyper N' to Predict the Performance of Air Vehicles Traveling at Hypersonic Speeds

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Younes Baalla
August 2010

Dedication

This thesis is dedicated to my loving parents and lovely fiancé. This is only a small repayment of all that I owe them. It's because of their unconditional support that I was able to achieve many goals in my life.

Acknowledgements

I wish to acknowledge all of those who have guided me toward the achievement of my Master of Science degree. Whether it is the support of friends or the teachers that have broadened my horizons I am convinced that I wouldn't be here without you. A special thanks to Dr. Peter Solies who presented the idea and advised me through the project. I am also deeply grateful for the financial support I received from the University of Tennessee Space Institute.

I would like to extend my gratitude to my committee members: Dr. Stephen Corda, Mr. Borja Martos, and of course my advisor Dr. Peter Solies. I appreciate them for taking the time to mentor me through my degree.

Abstract

The main objective of this thesis was to develop a method than can be used to approximate the pressure forces on air vehicles traveling at hypersonic speed (Mach number > 5). The aerodynamic forces such as lift and drag were calculated from the pressure values on the surface of the airplane. Pitching moment was also tabulated.

This work was initiated based on the idea of developing a flow solver proficient and capable of providing aerodynamic data (lift and drag look-up tables) for hypersonic air vehicles that can be fed to a flight simulator (used by the Aviation Systems Department) at the University of Tennessee Space Institute. Several approximation methods are used to solve hypersonic such as shock expansion method. Based on different studies, Computational Fluid Dynamic (CFD) proved to produce very accurate results; however, it is a difficult technique to use.

In this thesis work Newtonian Method was adopted as a technique to approximate the aerodynamic forces and hence the performance of hypersonic airplanes, therefore, a computer program (Hyper-N) has been developed for aerodynamic analysis of three dimensional geometries airplane. The program is designed to read in a previously configured list of plates and compute the aerodynamic forces and moments for hypersonic free stream conditions. Programming was completed using MatLab language. The results obtained from the Hyper-N program were for the experimental airplane X-43A which were found to match the results when the shock expansion method is used for the same airplane, [1].

Because of the difficulties involve in using CFD or the complete Navier Stocks equation to obtain the aerodynamic forces on bodies traveling at hypersonic speeds, the Newtonian method is considered to be the most efficient technique to use for preliminary evaluation of the

performance of hypersonic airplanes. Modified Newtonian theory and the computational requirement of the code are described. A number of geometric configurations, including the X-43A (experimental hypersonic) airplane, are provided as examples of applications of the Hyper-N program.

Preface

At the onset of the thesis a brief overview of the Inclination Methods is introduced with the emphasis on Newtonian method. The focus on Newtonian method should initially seem immaterial since it is not take into account the characteristics associated with hypersonic flows. However, the Newtonian method is considered an appropriate method to use for approximating the preliminary aerodynamic performance of hypersonic airplanes.

.

Table of contents

I. Introduction …… 1

1.1. Background – Aerodynamic Modeling …….. 2

1.2. Newtonian Theory …….. 5

1.3. Modified Newtonian Theory …….. 8

II. Project Overview

2.1. Concept of Thesis…… 9

2.2. Overview of the Algorithm……..11

III. Description of the Algorithm

3.1. Plates Intersection Consideration ……17

3.2. Algorithm Stages……..23

3.3. Determination of Centroids……..26

3.4. Plates Projection……...26

3.5. Plates Intersections Projection……..29

3.6. Algorithm Capabilities……..32

3.7. Algorithm General Consideration……37

3.8. Computation of Forces and Moments……38

IV. Results for model of X-43A……. 41

V. Implementation of the Algorithm……66

5.1. General Considerations……66

5.3. Organization of the Program…….. 66

VI. Concluding Remarks……69

VII. Vita…… 76

LIST OF FIGURES

Figure 1: Drawing of the X43 Model

Figure 2: Drawing of the X43 Model (Normals Shown)

Figure 3: Drawing of the X43 Model (Stabilizer deflection)

Figure 4: X43 Model (Impacted Surface Highlighted in red)

Figure 5: X43 Model (Restriction of Surface after first stage)

Figure 6: X43 Model (2D projection in Airstream Direction)

Figure 7: X43 Model (Restriction of Surface After Final Stage)

Figure 8: Drawing of a Demonstration Model

Figure 9: Plates Lying in the Same Plane

Figure 10: Plates Lying in Parallel Planes

Figure 11: Plates Lying in intersecting Planes

Figure 12: Plates not Satisfying Criteria

Figure 13: Normal Vector of Plate

Figure 14: Plate Facing Airstream

Figure 15: Plate Facing away from Airstream

Figure 16: Collection of Plates Facing Airstream

Figure 17: (x,y,z) and (x',y',z') Coordinates

Figure 18: Demonstration Model (intersection of projections)

Figure 19: Two Regions with intersecting projections

Figure 20: After removal of obstructed surfaces

Figure 21: Example of three plates placed in airstream

Figure 22: 2D projection of boundaries in Airstream Direction

Figure 23: Distinction of intersected areas

Figure 24: Illustration of comparison points

Figure 25: Small plate entirely in front of a larger plate

Figure 26: 2D Projection

Figure 27: Resulting selected areas

Figure 28: 2D selected areas mapped back to plates in 3D

Figure 29: Direction of Force Vectors

Figure 30: Reference axes for moments

Figure 31: Drawing of the X43 Model (engine removed)

Figure 32: Lift Coefficien vs angle of incidence

Figure 33: Drag Coefficien vs angle of incidence

Figure 34: Drag Polar

Figure 35: Lift to Drag Ratio

Figure 36: Pitching Moment ac = 21.35 feet

Figure 37: Moment with respect to the y-axis at x = 40 feet

Figure 38: Moment with respect to the x-axis at x = 40 feet with zero deflection angle of the control surfaces

Figure 39: Moment with respect to the z-axis at x = 40 feet with zero deflection angle of the control surfaces

Figure 40: illistration of stabalizer deflection (exagerated)

Figure 41: Moment with respect to the y-axis with -0.04 radians deflection angle of horizontal stabalizers

Figure 42: Rudder deflection exaggerated by a factor of 10

Figure 43: Angle of inclination = 0 degrees

Figure 44: Angle of inclination = 2.5 degrees

Figure 45: Angle of inclination = 5 degrees

Figure 46: Moment with respect to the z-axis with -0.02 radians deflection angle of rudders

Figure 47: Stabilizer deflections exaggerated by a factor of 50

Figure 48: Moment with respect to the x-axis with 0.01, -0.01 radians deflection angle of horizontal stabalizers

Figure 49: Drawing of the X43 Model (engine included)

Figure 50: Lift Coefficient vs angle of incidence

Figure 51: Drag Coefficient vs angle of incidence

Figure 52: Drag Polar

Figure 53: Lift to Drag Ratio

Figure 54: Drag coefficient with and without engine

Figure 55: 10 degrees wedge angle

Figure 56: Lift Coefficient vs angle of incidence

Figure 57: Drag Coefficient vs angle of incidence

Figure 58: Drag Polar

Figure 59: 10 degrees deflection angle

Figure 60: Lift Coefficient vs angle of incidence

Figure 61: Drag Coefficient vs angle of incidence

Figure 62: Comparison for 10 degree wedge and fanned wedge

Figure 63: 3D rotation described by 2 perameters instead of 3

Nomenclature

C_P = Pressure Coefficient

C_{Pmax} = Maximum Pressure Coefficient

P = Static Pressure

P∞ = Free Stream Pressure

Q_inf = Dynamic Pressure

P_02 = Pressure Behind Shock Wave

Psurface = Pressure at The Surface

M = Mach number

P_i = i^{th} Vertex

x_i, y_i, z_i = Rectangular Coordinates

c = Bari-center

n = Normal Vector to the Plate's Surface

× = Cross Product

· = Dot Product

V = Velocity

A_d = Free Stream Direction

M_x, M_y, M_z = Linear Moment of Inertia.

I = Moment of Inertia Tensor

F = Force.

a = Acceleration.

m = Mass.

τ = Torque

ω = Angular Velocity

L = Angular Momentum

C_L = Lift Coefficient

C_D = Drag Coefficient

Greek

Θ = Deflection Angle.

γ = Specific Heat Ratio

Introduction

As stated earlier, different methods are used to either approximate or obtain exact results for the pressure forces on hypersonic airplanes, however, Local Surface Inclination methods, in particular Newtonian method, were considered because they are capable of modeling the inherently nonlinear hypersonic flows and presenting them as simple linear relationships. They are considered to allow rapid estimation of pressure distribution over hypersonic bodies, defined solely in terms of the local surface inclination angle. An overview of the Newtonian method and the way that can be applied to hypersonic flows is described in details in later section of this thesis. As a result, the inviscid flow solver (Hyper-N) developed in this thesis work uses an impact theory, to estimate the pressure distribution over three dimensional bodies in high Mach number flow regime. Hyper-N is based on Newtonian flow theory developed three hundred years ago by Isaac Newton for low speed hydrodynamic analysis. Ironically, Newton's model is more suitable for hypersonic flow fields, finding applications as an approximated scheme for determining aerodynamic performance of three-dimensional bodies travelling at hypersonic speeds.

The body geometry of the airplane in question consists of a set of flat plates; each single plate (defined by its vertices in three-dimensional coordinate system) is analyzed independently as either an impact or shadow flow region. The surface pressure over an impact region is calculated using the local surface inclination (relative to the free-stream direction) method. Shadow flow regions are considered to have zero pressure coefficient, where the surface pressure equal to the free-stream pressure. There are no routines within Hyper-N to analyze the complete flow processing for the integrated airframe and engine arrangement found in scramjet planes. For such an application, the program is limited to finding estimates for impact pressure on forward facing

surfaces of the body and other external surfaces over the length. Blunt body shapes such as reentry vehicles do not have this limitation and provide the most useful applications for Hyper-N in particular at the stagnation point where the result from Newton's method is exact.

A review of literature on aerodynamic modeling of hypersonic vehicles is provided as justification for the code. The theoretical basis of the program along with a detailed description of the surface modeling data required as input is provided. The structure of the code is then described, detailing the application of Modified Newtonian flow to arbitrary configurations.

1.1 Background – Aerodynamic Modeling

The review of relevant literature given here presents the idea that simple techniques can be useful for hypersonic aerodynamic analysis. Newtonian theory and other pressure methods have proven to be useful preliminary (conceptual) analysis tools, primarily due to speed of computation. Critical aerodynamic and control issues can be identified and examined extensively. With no intermediate techniques separating these methods and full Euler or Navier-Stokes approach, a Newtonian based analysis for arbitrary bodies may be suitable for control system design. When a more accurate knowledge of aircraft performance is required for complex geometries, the more sophisticated CFD techniques must be used but at the computational cost that may be much higher than the cost of running a Newtonian analysis.

A considerable number of flight/launch vehicles focusing on the hypersonic flow regime, for which ground-based testing is limited (in size, performance, and cost). With this in mind and through the benefit of modern computing power, computational methods in the aerodynamic design and analysis of aircraft performance are becoming frequently used.

There are a large range of computational techniques used for aerodynamic design and analysis, design analysis and cost are the key elements in determining which technique or approach is appropriate and efficient for use.

Stability and control studies for systems with a behavior that is not well understood, will normally involve several aerodynamic performance estimates around the actual design. As a result, compromises in computational techniques are essential, leading to simplified models of the system. Computational Fluid Dynamics (CFD), even though it provides the most in depth picture, is frequently left for detailing special problems in practical design.

Aerodynamic prediction techniques for hypersonic configuration have often been based on the *Hypersonic Arbitrary Body Program* (HABP), developed by Gentry [1] in the late 1960's. The HABP analysis procedure is based on "non-interfering constant pressure finite-element analysis". The compression-expansion methods are suitable for determining the pressure profile for a hypersonic analysis of a surface configuration. HABP is used for preliminary design and analysis of hypersonic vehicles.

In 1980 Divan [2] constructed an interactive program which enabled mesh configuration and viewing to be linked to the analysis routines. Divan's *Aerodynamic Preliminary Analysis System* used the force calculation modules of HABP, providing a comprehensive program covering speeds from subsonic to hypersonic.

During the late 1980's, Moore & William [3] provided a selection rational for the methods described in HABP. A range of hypersonic configurations were considered with the focus on vehicle control analysis. Solution techniques where allocated to three basic body parts, nose, body, and aerodynamic surfaces. For high hypersonic speeds, the Modified Newtonian method

was used for all impact surfaces, while all shadowed regions incorporated Prandtl-Meyer expansion. It was also noticed that there was no significant improvement in prediction accuracy by applying a real gas approximation through the use of an effective specific heat ratio, γ. There was also a minimal change in the overall vehicle pressure distribution when viscous methods were applied especially as the Mach number gets higher (Mach >> 5) [4], viscous considerations are more critical for estimating vehicle drag. The prediction of lateral-directional aerodynamics may not be as accurate using the same selection rational as that for the longitudinal aerodynamics when applying more complex viscous calculations.

Cruz & Wilhite used modified version of HABP known as the *Aerodynamic P*reliminary Analysis System (APAS) [5], presumably similar to that developed by Divan [6], though no indication is given in the report. The aerodynamic characteristics of six simple configurations and three complex configurations were investigated, with independent inviscid and viscous solutions applied to a non-interacting finite element model. Lift and drag coefficients were compared with experimental and computational fluid dynamics (CFD) results, with good overall agreement. Viscous contributions to vehicle drag proved most important for cone based bodies such as the winged-cone configuration. Vehicles with a lower slenderness ratio produce pressure distributions dominated by pressure drag rather than viscous effects.

Maughmer *et al* [7] reported on the accuracy and validity of the local surface inclination methods found in HABP, for predicting control forces and moments. Hypersonic configurations considered included the X-15, the Hypersonic Research Airplane (scramjet powered), and the Space Shuttle orbiter. Comparisons were made with both experimental and flight test data, with each vehicle assigned analysis methods in relation to nose, body, and aerodynamic lifting forces. Modified Newtonian theory was used on all blunt surfaces such as leading and trailing edges.

The other methods which were applied to the inviscid analysis were tangent-wedge, tangent-cone, and Prandtl-Meyer (shadow regions). The primary control derivatives for the lateral/directional case also provided reasonable results.

Other useful references of the application of Newtonian flow to the analysis of vehicles in hypersonic flow are Anderson [8] and Chavez & Schmidt [1, 7]. Anderson dedicates a chapter to local surface inclination methods, presenting them as simple linear relationships capable of modeling the inherently nonlinear hypersonic flows. They are considered to allow rapid estimation of pressure distribution over hypersonic bodies, defined solely in terms of the local surface inclination angle..

Schmidt has been extensively involved in investigating the integration between vehicle airframe and propulsion found in hypersonic vehicles. In reference [2, 4], the analysis procedure used in the dynamic analysis of a configuration known as X-30 (National Aerospace Plane), is presented. Newtonian theory was used for determining the pressure distribution over the fore-body, and coupled to a one dimensional model of the scramjet engine. Two-dimensional shock-expansion theory gave the pressure distribution on the after-body/nozzle.

1.2 Newtonian Theory

When a blunt body is immersed in a high-Mach-number (supersonic or hypersonic) stream, a bow shock surrounds the front portion of the body. The solution for supersonic *blunt-body problems* generally requires powerful computational fluid dynamic techniques. However, the use of a simpler method or technique, such as, the *local surface inclination method* may not be appropriate predict the pressure distribution on the body surface. This method was originated by

Isaac Newton long before compressible flow concepts such as shockwave were even imagined; this model is known as *Newtonian theory.*

Newton considered a uniform stream of particles that collides in-elastically with a surface. Newton hypothesized that when a particle hits a surface; it loses all of its momentum in the direction normal to the impacted surface, but retains all of its momentum in the direction along (tangent) to the surface. Newton had no success in using this concept to determine the hydrodynamic forces on ship hulls.

The method was not widely used until the 1950s, when it was realized that a gas traveling at extremely high speeds behaves in a manner not unlike that proposed by Newton. It was found that at high Mach numbers, the bow shock surrounding a blunt body (or oblique shock in the case of sharp leading-edge bodies) lies very close to the surface of the body. Accordingly, in a very high-speed flow, the gas particles are almost to reach the body without any change in their velocity. After passing through the shock, the gas if forced to rapidly turn and form a thin region between the shock and the body, which is called *shock layer.*

It is easy to demonstrate, from the oblique-shock theory, that a fixed wedge angle δ, as the free-stream Mach number increases, the shock angle θ approaches the deflection angle δ. [1,6]It can also be shown that for Mach = ∞ and specific heat ratio $\gamma = 1$ that $\delta = \theta$. Further justification of this theory comes from the fact that because of high temperatures and pressures of very hypersonic flows, real gas effects drive the value of γ towards unity. [1, 4, 8]

Consider a portion of the bow shock and shock layer surrounding a blunt body, from the momentum balance in the direction of the shock the equation below was obtained

$$p_1 - p_2 = \rho_2(V_{n2})^2 - \rho_2(V_{n1})^2. \qquad (1)$$

Because the flow upstream is the free-stream and because the gas downstream of the shock loses all of its normal momentum, the following row vector expression is obtained

$$[p_1\ V_{n1}\ \rho_1] = [\rho\square\ V\square sin\delta\ \rho\square] \qquad (2)$$

and

$$[p_2\ V_{n2}\ \rho_2] = [p_{surface}\ 0\ \rho_2] \qquad (3)$$

δ is the local inclination angle of the surface with respect to the airstream direction. This method is referred to as the *local surface inclination method.* Several other methods are also being used, such as, the shock-expansion theory, the tangent wedge, and tangent cone theories [4]. Inserting the foregoing conditions into the momentum equation above, produces

$$p_{surface} = p\square(V\square)^2(sin\delta)^2 \qquad (4)$$

the above expression is obtained directly by applying control-volume methods. The expression may be rewritten as a *pressure coefficient*

$$C_p = 2(p_{surface} - p\square)/(\rho\square V^2\square) = 2sin^2\delta \qquad (5)$$

This expression is often referred to as *Newton's sine-squared law*. From above, the pressure coefficient is dependent solely on the geometry of the body and not on the free-stream Mach number.

1.3 Modified Newtonian flow theory

The modified Newtonian method [8] is one of a number of impact/shadow flow analysis methods which are commonly used in determining the surface pressure distribution over hypersonic geometries. As a *local surface inclination method*, it neglects high temperature, viscous, and boundary layers effects. If the body is defined as a collection of flat surfaces or plates, Newtonian theory can be applied independently to these elements to provide a complete body pressure distribution. From this distribution, aerodynamic forces and moments can be integrated. The only geometric parameter required to obtain the surface pressure from Newtonian theory, is the angle the free-stream impacts with the element. To compute aerodynamic forces and moments, plates areas and centroids are also used.

Modified Newtonian theory follows the standard Newtonian sine-squared law, with an adjustment to give the correct pressure coefficient at the stagnation point. The modification to Newtonian flow is represented here, (as it is in Reference [2, 3]), in terms of the pressure coefficient,

$$C_P = C_{Pmax} \sin^2\theta \qquad (6)$$

where C_P is the local pressure coefficient,

$$C_P = (p - p_{\infty})/q_{\infty} \qquad (7)$$

p is the local pressure, q_∞ is the free-stream dynamic pressure and is equal to $\frac{1}{2}\rho_{\infty}(V_{\infty})^2$, with V being the free-stream velocity. The angle theta is the body deflection angle, the angle the surface makes with the free-stream direction. Theta takes into account the geometry as defined relative to the body-fixed axes and the angle of attack of the body in the flow. The local surface normal vector is the geometric parameter used. C_{pmax} is the maximum value of the pressure coefficient, evaluated at the stagnation point behind a normal shock wave,

$$C_{Pmax} = (p_{02}-p_{\infty})/q_{\infty}. \qquad (8)$$

The Rayleigh Pitot tube formula, derived from normal shock wave relations, enables the pressure at the stagnation point P_{02}, to be determined. This forces the modified Newtonian analysis to be "exact" at the stagnation point, following the ($\sin^2$) law for all other areas of the body. The stagnation pressure behind a normal shock wave, p_{02}, is defined by the formula,

$$P_{surface} = p_{\infty} + (p_0 - p_{\infty})\sin^2\theta = \left[\frac{1+\gamma M^2_{\infty}\sin^2\theta}{1+\gamma M^2_{\infty}}\right] \qquad (9)$$

with C_{Pmax} evaluated for given free-stream conditions, the pressure coefficient over the geometry can be determined,(theta now being the shock angle). Modified Newtonian theory is therefore only dependent on the free-stream conditions and the geometry of the body.

The reference angle is the angle between the velocity vector and the inward normal to the plate surface. Thus $\cos^2(\theta)$ replaces the $\sin^2$ term in equation 1. Once the geometry of the body surface is defined (by a set of flat plates) the application of equation (1) plate, is all that is needed to determine the complete pressure profile.

II. PROJECT OVERVIEW

2.1 Concept of Thesis

It is beyond the scope of this thesis to consider an aircraft that must be described by arbitrary surfaces. A number of aircraft geometries can be approximated by a series of flat plates.

In any case any aircraft could be closely approximated (Mathematically) by a model using a large number of flat plates that are small in size. The program, in this thesis, used to model aerodynamic forces acting on the aircraft is written in MatLab scripts and therefore is not able to handle a large number of plates.

MatLab programs run at enormously lower speeds than compiled software and have poor memory management, limiting their use for computationally intensive algorithms.

However, the algorithm used would work if it was rewritten in a programming language such as C^{++} or FORTRAN. The X-43A is the aircraft used as the example in this project. It has a geometry that can be closely approximated by a relatively small number of flat plates.

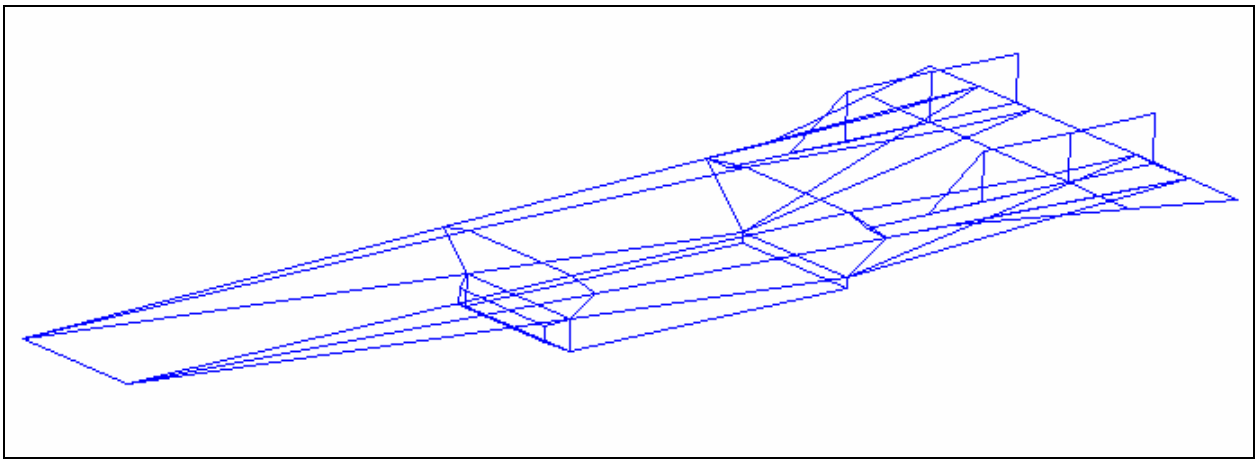

Figure 1: Drawing of the X43 Model

At the beginning of this project it was necessary to take into account the effect of airflow being obstructed from certain portions of the aircraft.

. The algorithm to calculate forces as predicted by the Newtonian model consists of several stages. Drawing the portions of the original collection of plates together with their normal vectors and centroids after each stage of the calculation was necessary to provide the intuition needed to develop the complete algorithm. Additionally, the visualization of the intermediate results helped find errors in the first attempt to define and program each stage of the algorithm.

When the project began it was only known that the Newtonian model would be used. Each stage of the procedure developed was decided on after pondering on drawings of the aircraft positioned in a variety of different angles between the center line of the aircraft and the direction of the free stream. Each time the plates were projected onto a 2 dimensional plane perpendicular to the free stream. By looking at the drawings of both the projected plates and the aircraft drawn in three dimensions it was observed that the projected areas intersected one another and that the studies of these areas of intersection were the key to understanding how to develop an appropriate algorithm.

2.2 OVERVIEW OF THE ALGORITHM

The plane is described as a finite collection of flat plates. Each “flat plate boundary” is defined by a convex polygon with its vertices given as 3 dimensional coordinates in the xyz Cartesian coordinate system. The edges of each plate are coincident with the edge of some adjacent plate. The plates combined form the outer surface of the aircraft. Since each plate in this collection represents part of the surface of the aircraft it has an associated outer normal vector. The outer normal points in the direction to the outside of the aircraft see Figure 2.

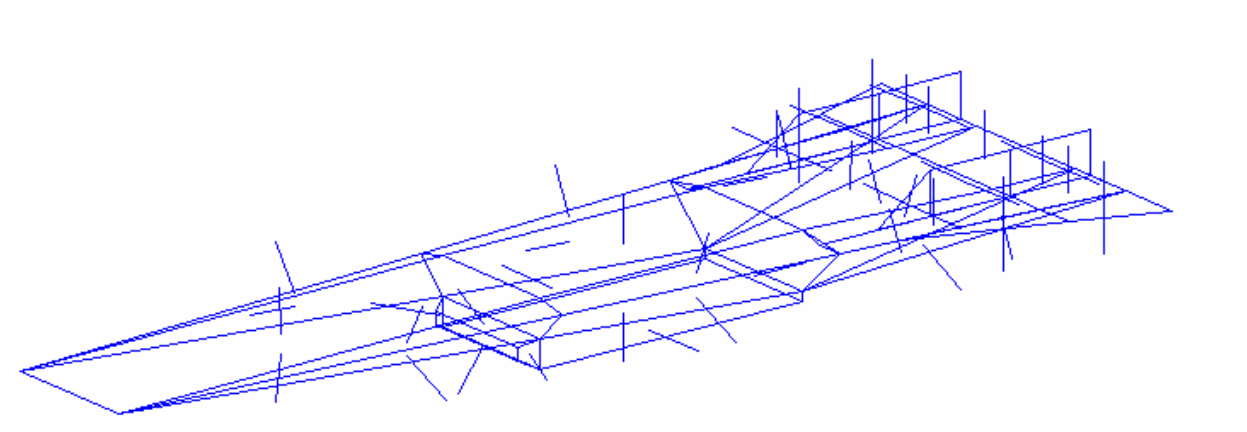

Figure 2: Drawing of the X43 Model (Surface-Normal Shown)

Control surfaces will be represented by plates that can be moved (rotated about an axis). Otherwise, the plates that the control surfaces consist of will be dealt with in the same way as the other plates since they will be held fixed in position during calculations of forces.

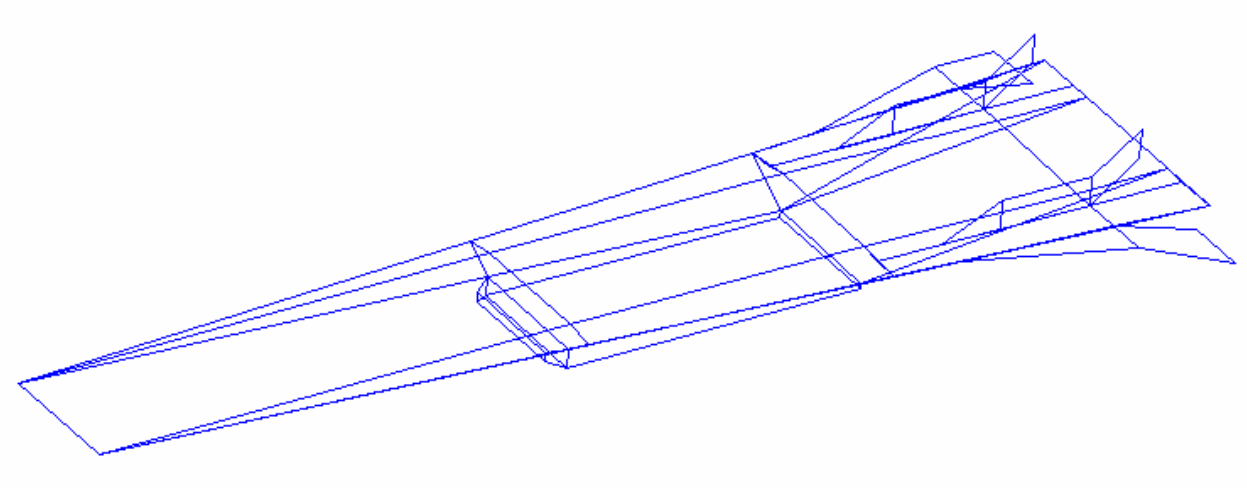

Figure 3: Drawing of the X43 Model (Stabilizer deflection)

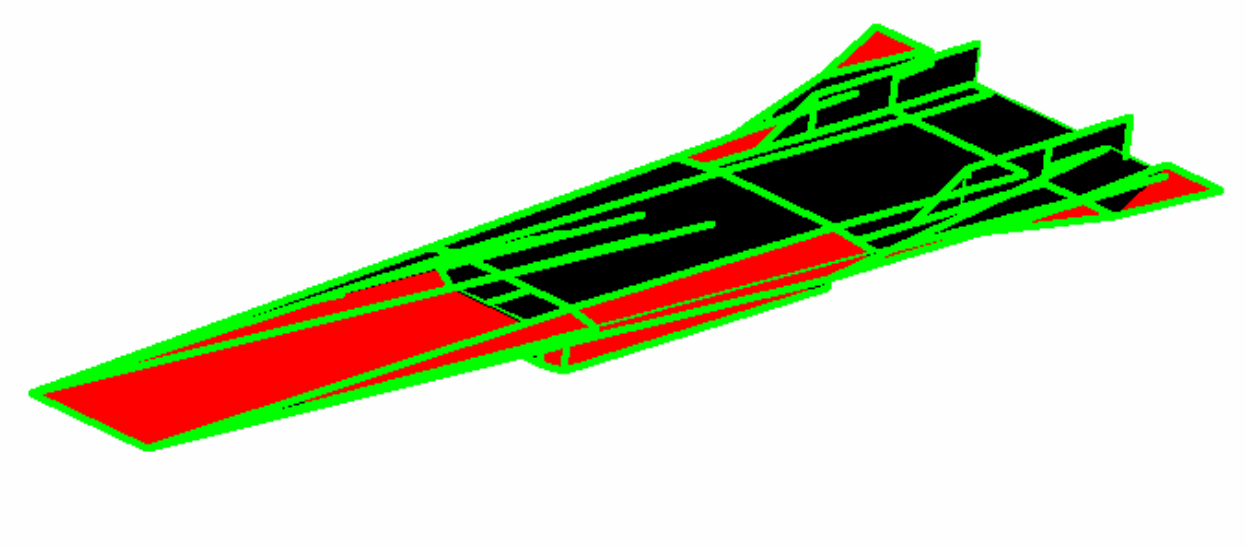

Figure 4: X43 Model (Impacted Surface Highlighted in red)

The air stream can impact a plate only if the angle between a line in the direction of the air stream and the outer normal, of the plate in question, is less than 90 degrees. The first stage of the algorithm is to determine the sub-collection of the plates for which these angles are less than 90 degrees. This is not enough to resolve the portion of the plane's outer surface that would impact the air

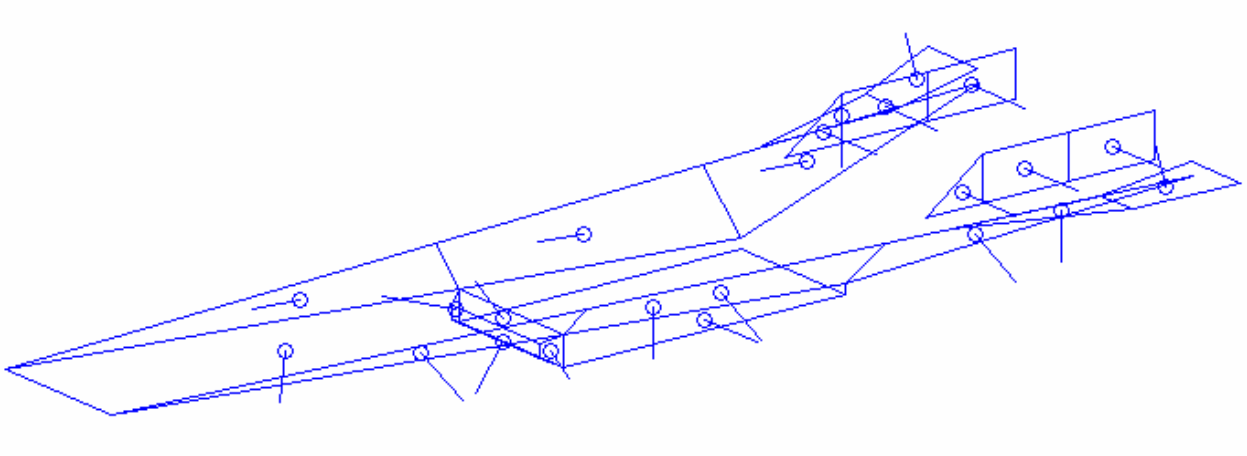

Figure 5: X43 Model (Restriction of Surface after first stage)

Some plates may be obstructed from the airstream either entirely or in part by plates that are in front of them with respect to the direction of the airstream. Therefore, we must further exclude or reduce the areas of the obstructed plates. Additionally, the unobstructed parts of a plate that is partially obstructed can occur as more than one disjoint piece. Other complexities occur such as the obstructed areas are interior to a plate forming holes. In these cases the remaining portions of the plates often are not bounded by convex polygons.

Returning to the place where the collection of plates has been restricted to just those that are facing the direction of the airstream, the algorithm continues by projecting this collection of plates onto a 2 dimensional plane perpendicular to the airstream direction.

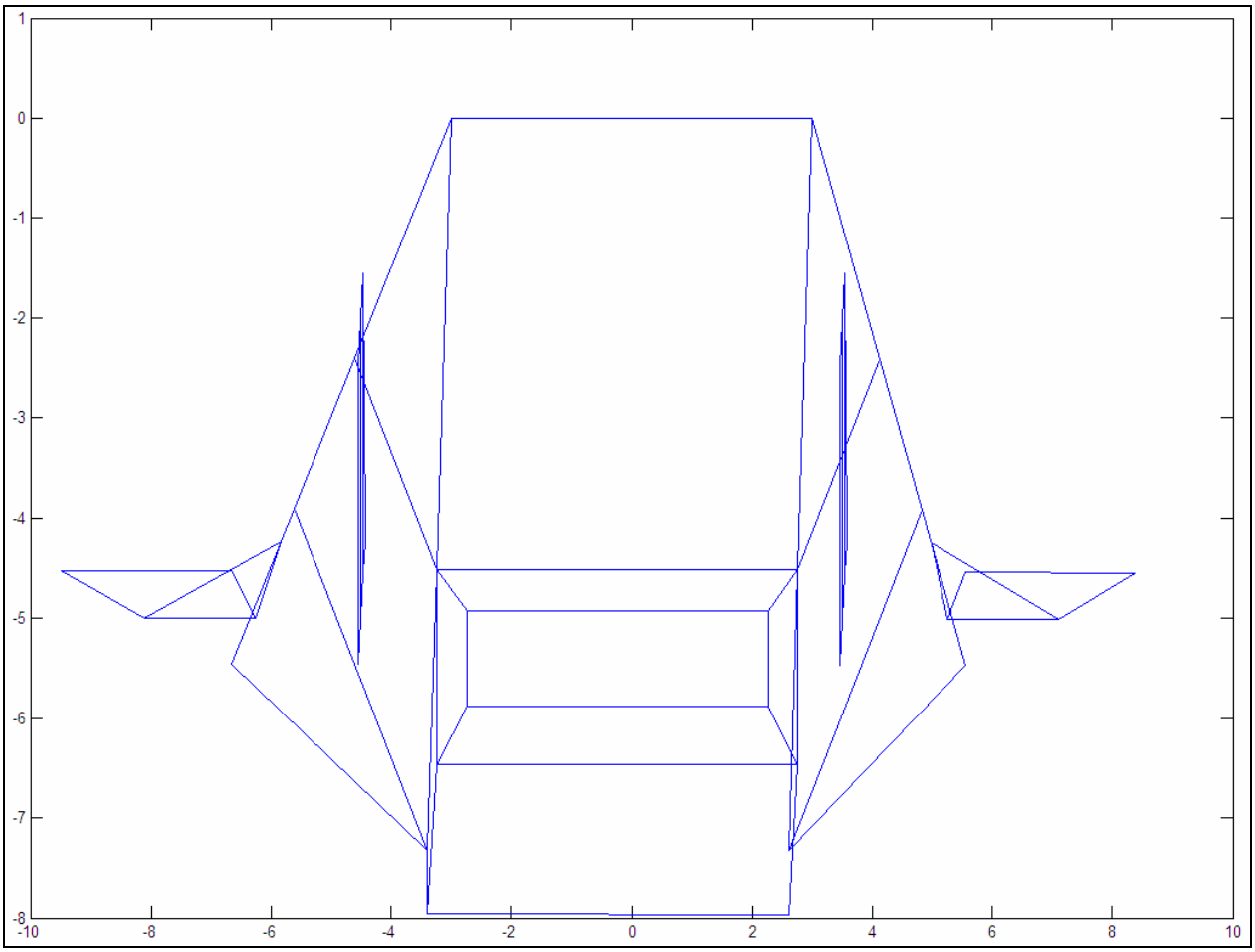

Figure 6: X43 Model (2D projection in Airstream Direction)

At this point the plates are projected onto the cross sectional area that the air impacts. Usually there are a number of intersections between the projected areas of two or more plates. For each of these areas of intersection the airstream will impact only the corresponding portions of the plate that is forward most with respect to the direction of the airstream. The projected plates and portions of projected plates that do not intersect any others are obviously included to find the net impacted areas plates.

The next stage of the algorithm is to find all of these non-intersected and intersected areas. The non-intersected areas are easily handled because we know which plates they represent and the areas concerned. For the intersected areas we must determine which plates they reference. In

order to do so, it was found that the original partitioning of the plane's surface into plates must be fine enough so that no ambiguities arise as to whether the portions of the different plates an intersected area corresponds to have one plate that is distinguished from the others by all of its intersected area being in front of the other plates intersected areas. This was accomplished by requiring when any two plates are considered each plate lies entirely on one side of the line of intersection between the planes for which the two plates lie on.

Now for each of these intersected areas the portions of the plates to which they relate are determined. Next, the coordinates of appropriate comparison points are calculated in a coordinate system using the airstream direction as one of the axis. The foremost area is the one for which the coordinate of its comparison point in the direction of the airstream is the least.

Finally, all of the areas in question are collected. This includes the areas where no intersection occurred and the areas on the plates distinguished as explained above where intersections occurred. After this is done the areas together with their projected areas, centroids, and normal vectors are listed. Total force on the aircraft surface is then calculated piece by piece considering each of areas one at a time.

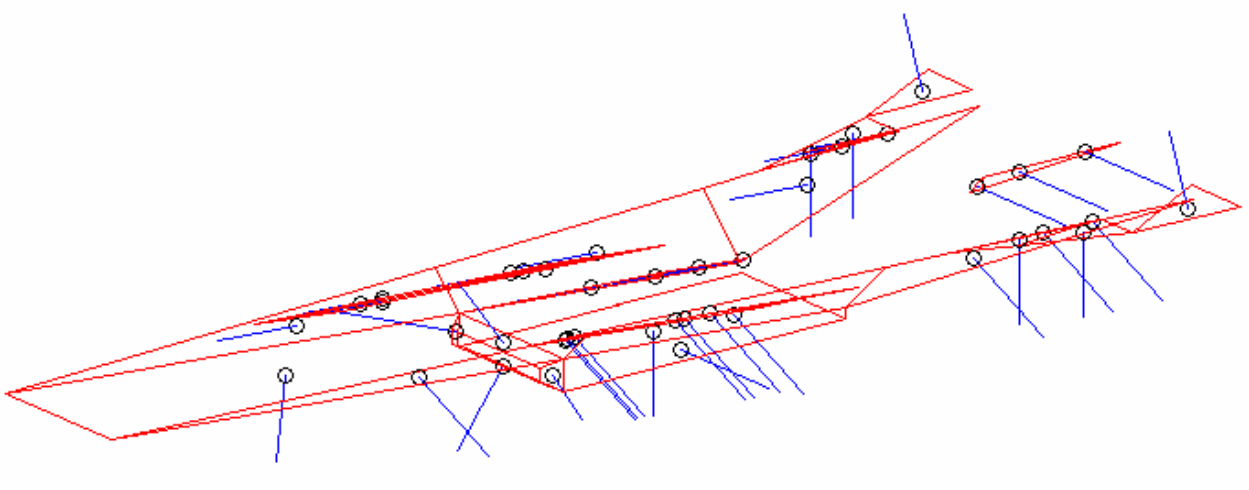

Figure 7: X43 Model (Restriction of Surface after Final Stage)

III. DESCRIPTION OF THE ALGORITHM

3.1 Plates Intersection Consideration

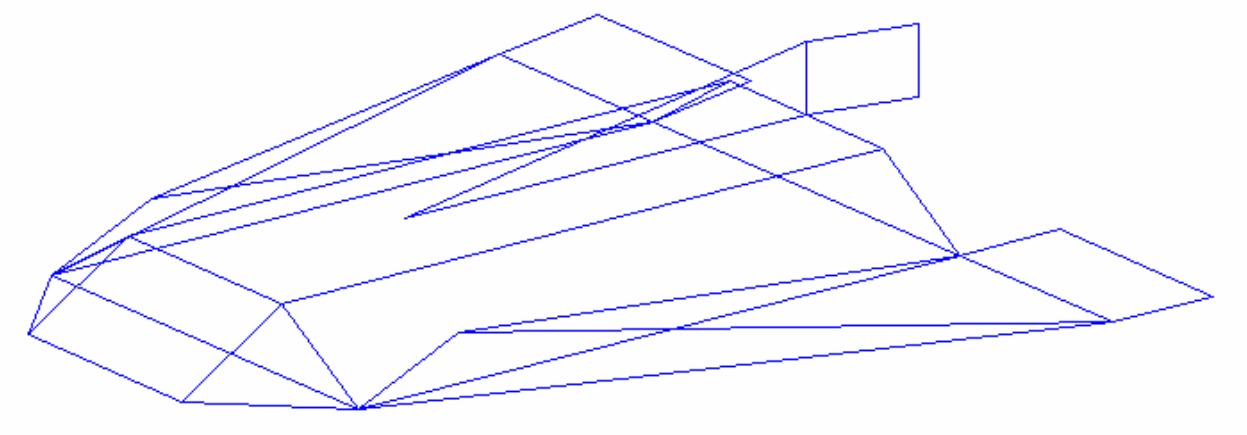

Figure 8: Drawing of a Demonstration Model

The algorithm begins with the plane's outer surface described as a collection of flat plates. The partitioning of the plane's surface into plates must be fine enough so that for any airstream direction when any two plates whose projections into a plane perpendicular to the airstream overlap the regions on each plate corresponding to the overlap are considered one region lies completely in front of the other with respect to the airstream. This is accomplished by requiring the following conditions. Each plate has a plane on which it resides. If two plates are chosen there are three possibilities. One of which is that they lie on the same plane. Another is they can be on parallel planes. The third case is that the two planes intersect.

The first two cases need no additional consideration as explained below. The third case each plate must lie entirely on one side of the line of intersection between the two planes.

In the first case one of the plates does not obstructed the airstream from the other.

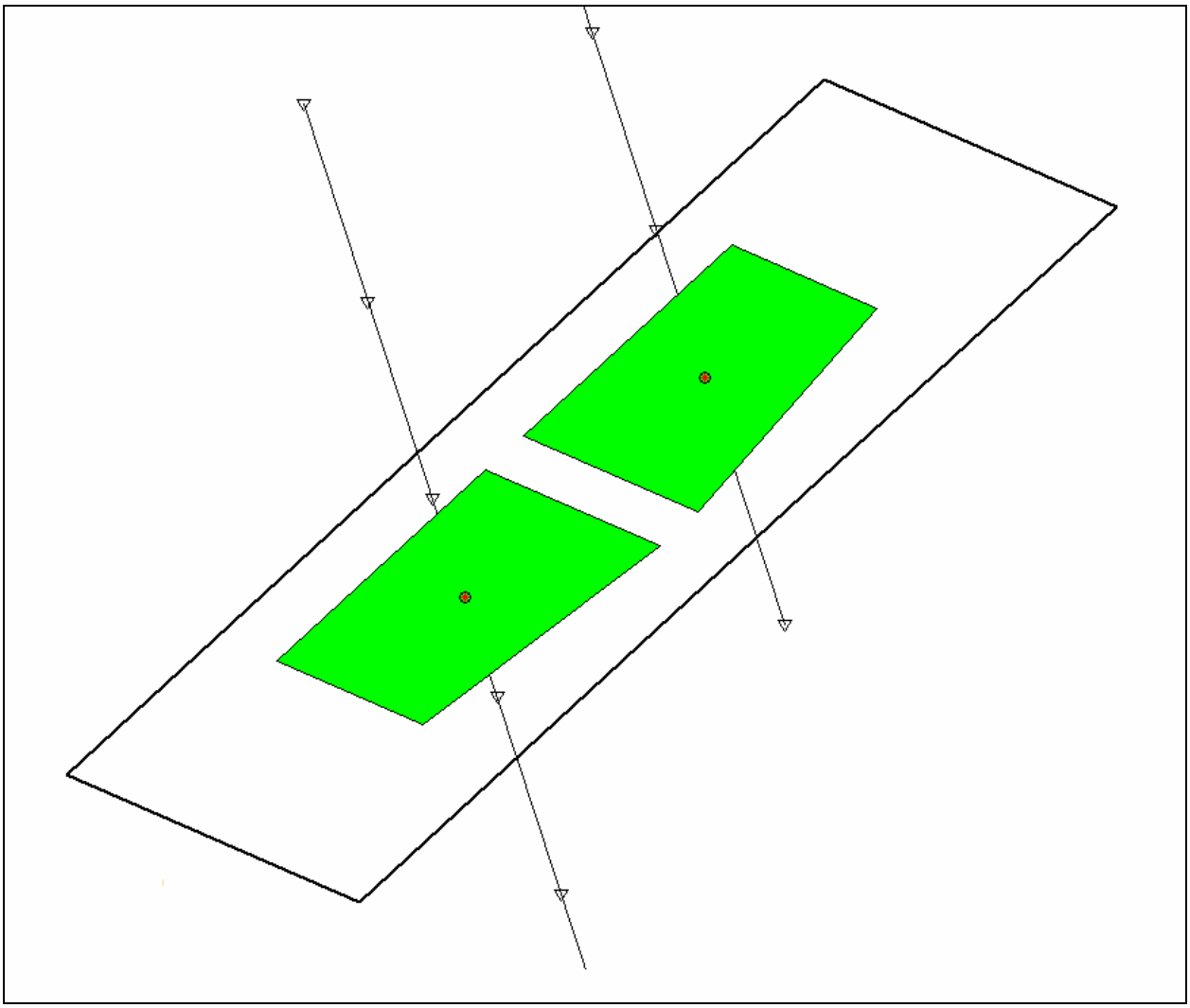

Figure 9: Plates Lying in the Same Plane

In the second case, even though one plate may obstruct the other, it will always lie completely before the other with respect to the airstream direction.

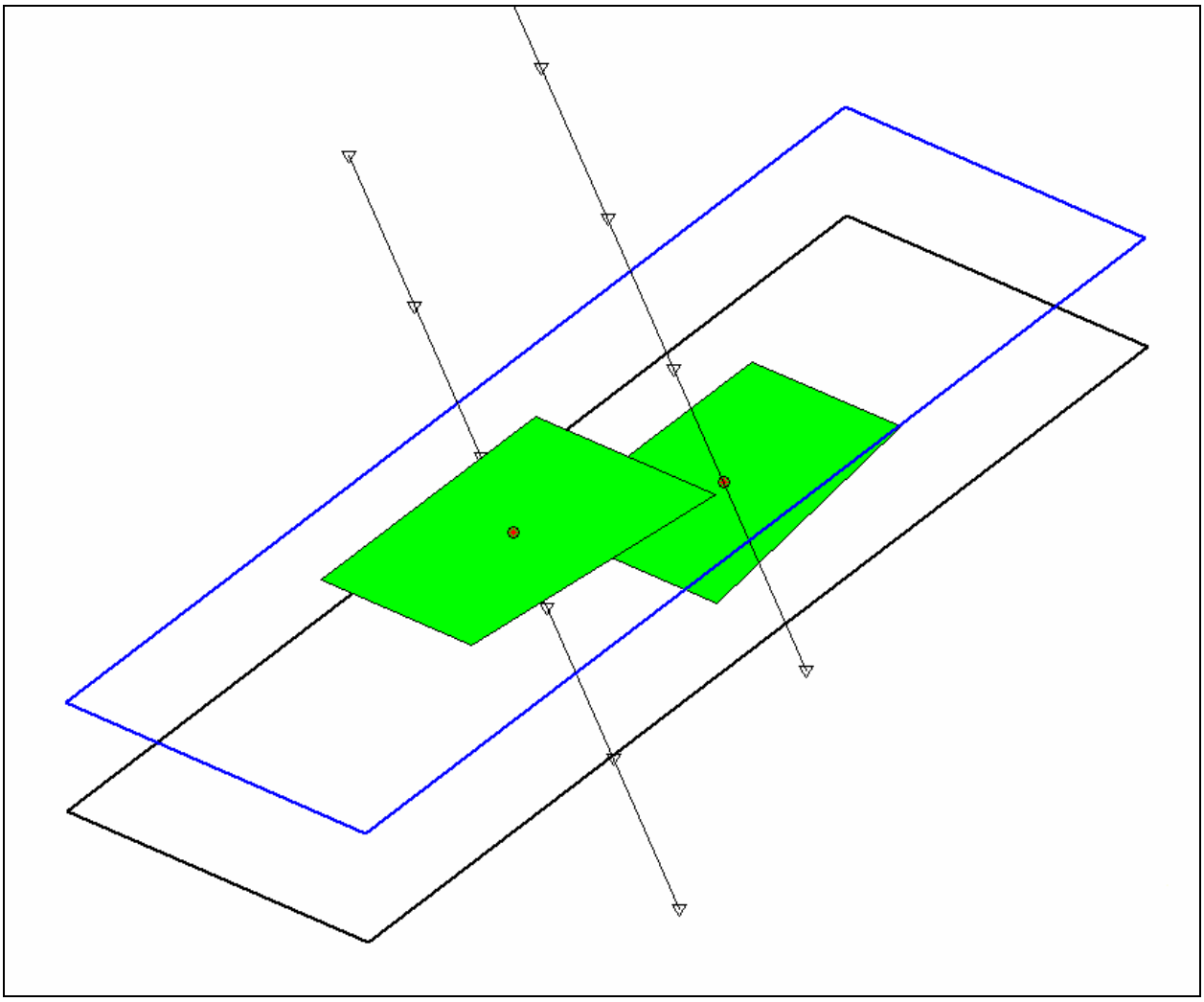

Figure 10: Plates Lying in Parallel Planes

In this case, the airstream will always impact the same half plane before the other. Which half plane, and thus which plate, is struck first can be decided by the positions of properly chosen comparison points one on each plate. The choice made was the following. Consider the overlap region of the projection of the two plates into a, as before mentioned, plane perpendicular to the airstream. (Shown as the hollow red polygon in the figure below, with an offset from the rest of the picture for clarity)

A line in the direction of the airstream passing through the centroid of this “hollow red” area intersects each plate in a point inside the region on that plate corresponding to the overlap. These two points are the points chosen for comparison. The details of this will be explained in the sections to follow.

A rectangular coordinate system with one of its axis in the direction of the airstream is used to determine which comparison point comes first. In particular the comparison point with the least coordinate on this axis is the one that comes first.

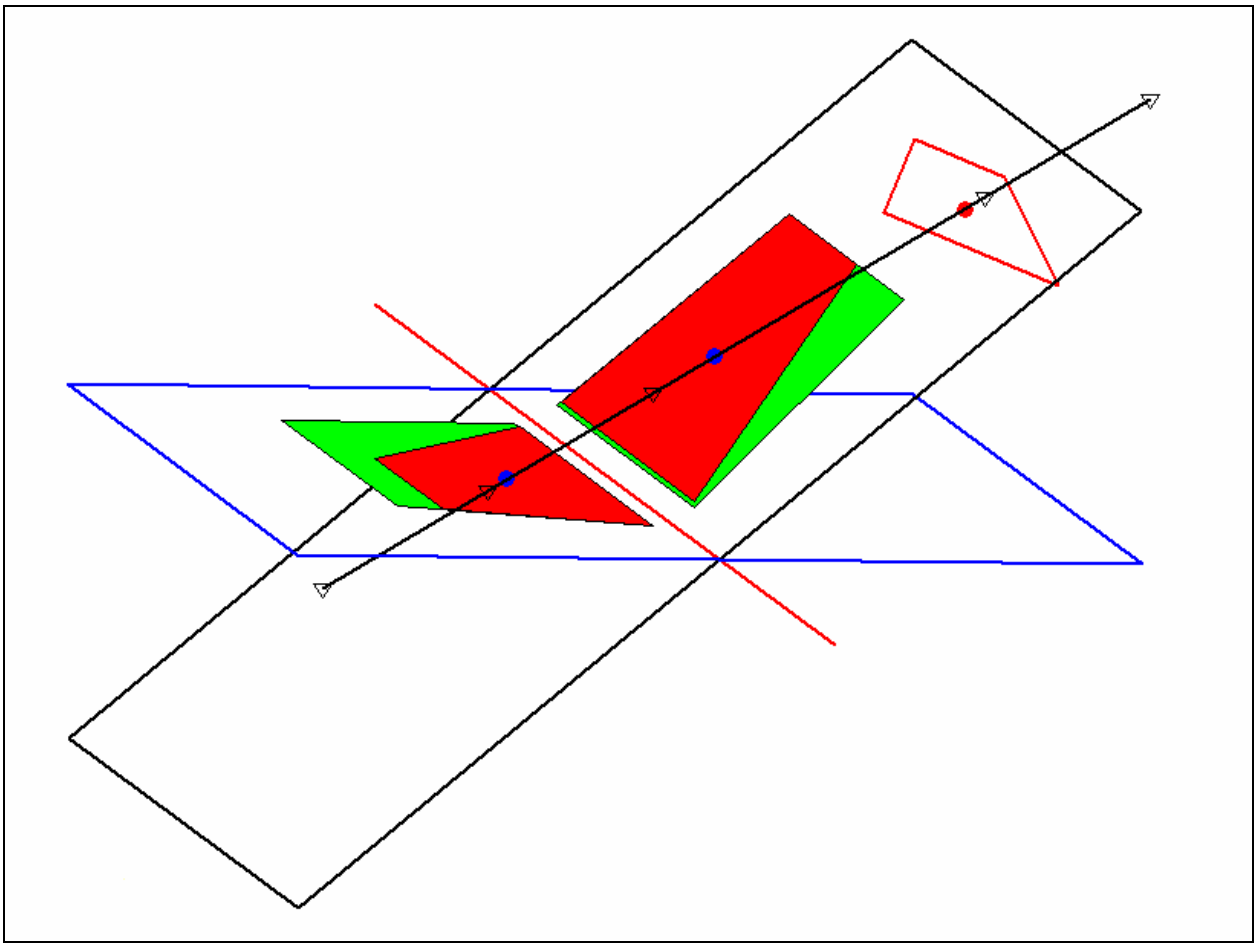

Figure 11: Plates lying in intersecting Planes

To see the need of this condition consider two plates which are in violation as illustrated below. The comparison point of the region in plane P_2 lies before the comparison point on the region in plane P_1. However, the region in plane P_2 (colored yellow) is obstructed from the airstream by a portion of the plate in plane P_1 colored red.

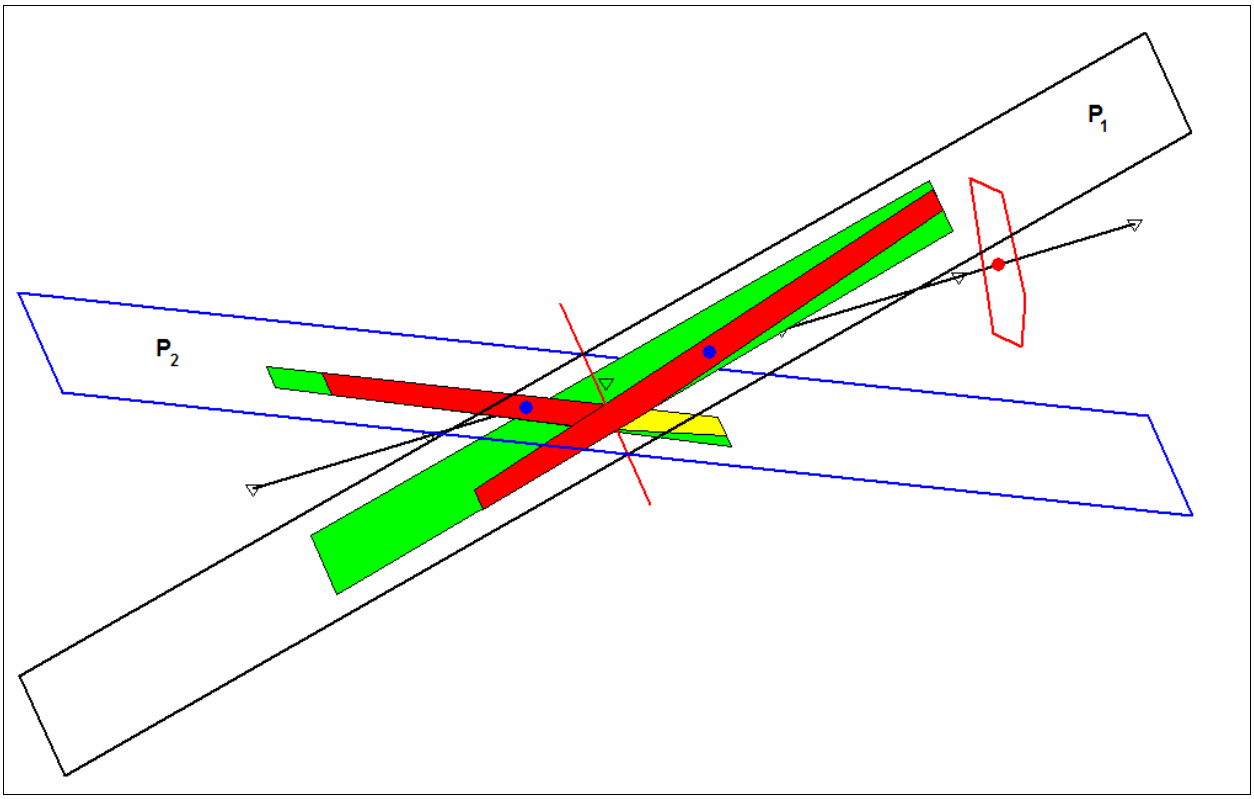

Figure 12: Plates not Satisfying Criteria

3.2 Algorithm Stages

The first stage of the algorithm is to calculate the outer normal vector of each plate. This was done by finding an interior point of the plate illustrated as **c** in the figure below. The most obvious choice for **c** is calculated by summing the vertices coordinate wise and dividing by their number. That is if the plate has vertices

$$P_i = (x_i, y_i, z_i)\ i = 1, 2, \ldots , n \qquad (10)$$

then

$$\mathbf{c} = \sum P_i / n. \qquad (11)$$

the point **c** is guaranteed to be in the interior of the plate since the plate is assumed to be convex.

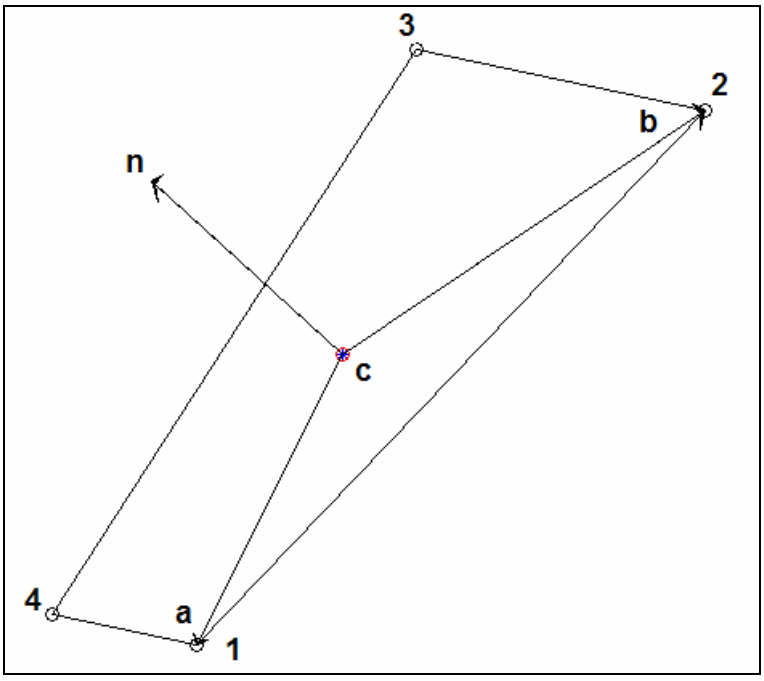

Figure 13: Normal Vector of Plate

Then two vectors are constructed each with its initial point at the point **c** and the terminal points at two different vertices of the polygon bounding the plate. The cross product of these two vectors is in the direction of the outer normal of the plate. If needed, an adjustment is made to the sign to make sure that the direction of the outer normal vector is pointing to the outside of the surface. The angle between a line in the direction of the airstream and the outer normal is calculated using the "dot-product".

When the angle between the airstream and the normal vector is less than $\pi/2$ the plate is impacted as the surface facing the airstream is on the outside of the aircraft.

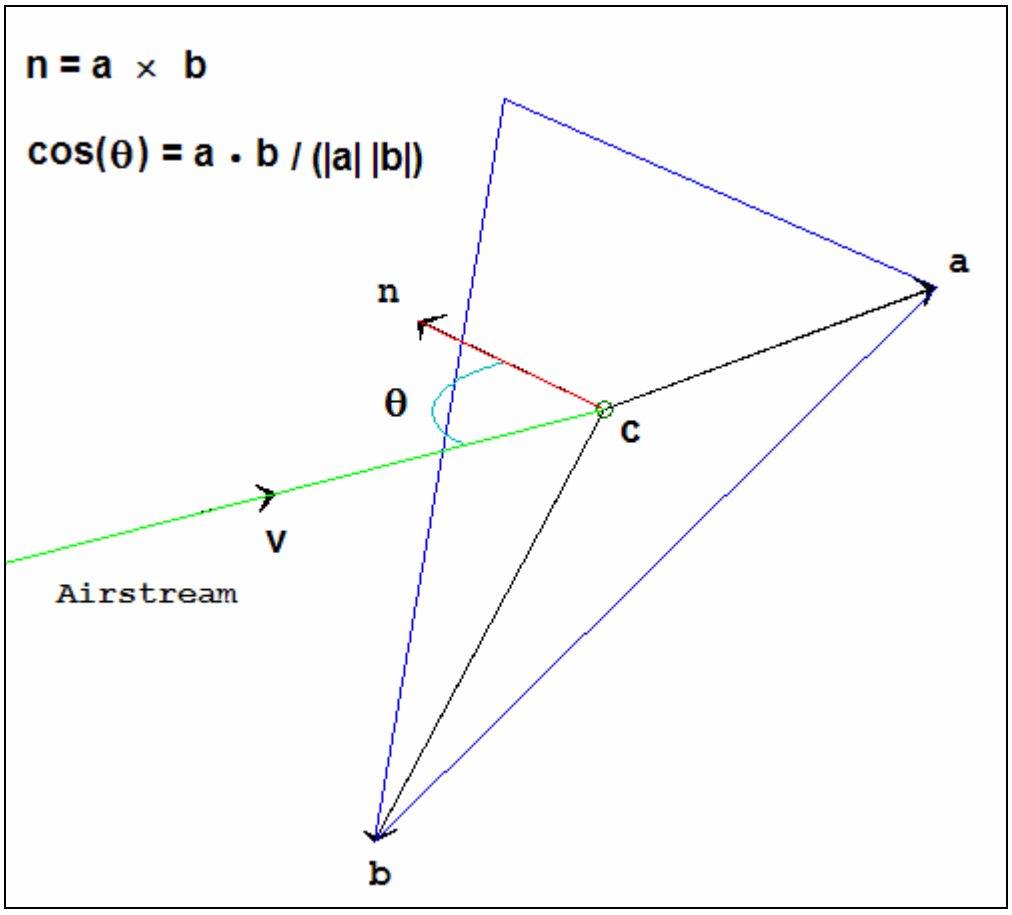

Figure 14: Plate Facing Airstream

When the angle between the airstream and the normal vector is greather than π/2 the plate is not impacted as the surface facing the airstream is on the inside of the aircraft. When the airstream is parallel to the plate, i.e. when the angle is π/2, we ignore this plate also even though it may contribute to some aerodynamic effects. At this point we restrict the original collection of plates to only the ones with $\boldsymbol{\theta} < \pi/2$.

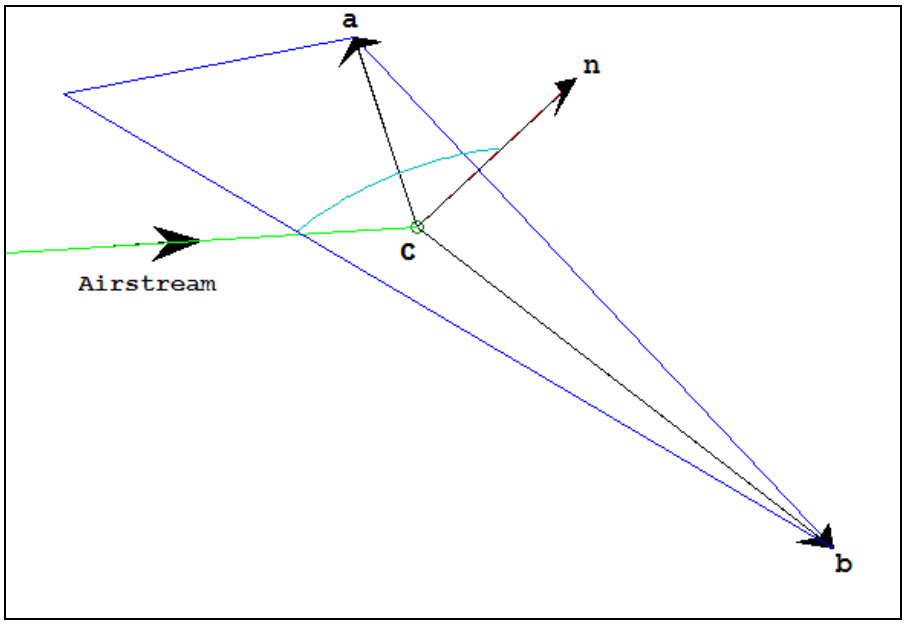

Figure 15: Plate Facing away from Airstream

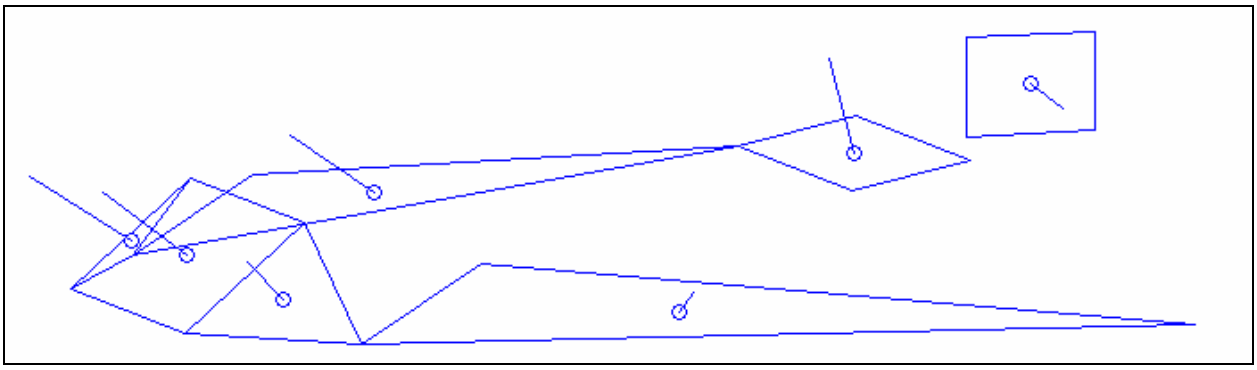

Figure 16: Collection of Plates Facing Airstream

3.3 Determination of centroids

Throughout the remaining discussion the centroid of a plate or other area will often be referred to. The centroid of an area represents it best, for the purposes of this algorithm, since it is the point at which the total force on an area can be applied to have the same effect as distributed force over the entire area. Even though many interior points of the plates can be used to determine the order of the plates with respect to the airstream direction, ones lying on the same line in the direction of the airstream are preferred points used in this discussion.

This formula is available for the centroid of a planer area bounded by a polygon.

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i\ y_{i+1} - x_{i+1}\ y_i)\ ,$$

$$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i\ y_{i+1} - x_{i+1}\ y_i)$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i\ y_{i+1} - x_{i+1}\ y_i).$$

The areas in question are defined in three dimensional coordinates not in two dimensional coordinates needed to apply the formula. Each area that arises during the algorithm is either an original plate or some subset of an original plate, so its normal vector will be known as explained in section **3.2.** This allows projecting the area in question into a two dimensional plane perpendicular to the normal vector, **n =** < n_1, n_2, n_3 >**.** The projection is defined as follows.

Let n_{k1}, n_{k2}, n_{k3} be $\mathbf{n_1}$, $\mathbf{n_2}$, $\mathbf{n_3}$ sorted from largest to least in absolute value. Let **a =** < a_1, a_2, a_3 >

where $a_{k1} = n_{k2}$, $a_{k2} = -n_{k1}$, and $a_{k3} = 0$. This assures that **a** is orthogonal to **n** since

$\mathbf{a \cdot n = a_1 n_1 + a_2 n_2 + a_3 n_3 = a_{k1} n_{k1} + a_{k2} n_{k2} + a_{k3} n_{k3} = n_{k2} n_{k1} - n_{k1} n_{k2} = 0.}$ Then for $\mathbf{b = n \times a}$

the projection given by $\mathbf{x_i} = < x_i, y_i, z_i > \cdot \mathbf{a}$ and $\mathbf{y_i} = < x_i, y_i, z_i > \cdot \mathbf{b}$ puts the polygon

$P_i = (x_i, y_i, z_i)$ i = 1, 2, ... , n in two dimensional coordinates. Now the formula above is applied to find the **x, y** coordinates of the centroid $\mathbf{C_x, C_y}$.

To transform back to the original coordinates let $\mathbf{z_i} = < x_i, y_i, z_i > \cdot \mathbf{n}$ and note that since the points (x_i, y_i, z_i) lie on the plane with the normal vector **n,** $\mathbf{z_i = z_1 =} x_1 n_1 + y_1 n_2 + z_1 n_3$ for each i.

The coordinates of the centroid in (x,y,z) the original coordinate system are given by

$$\begin{bmatrix} xc \\ yc \\ zc \end{bmatrix} = \begin{bmatrix} a1 & b1 & n1 \\ a2 & b2 & n2 \\ a3 & b3 & n3 \end{bmatrix} \begin{bmatrix} Cx \\ Cy \\ z1 \end{bmatrix}.$$

Hereafter is assumed that the centroid can be known for any planer area bounded by a polygon.

3.4 Plates Projection

Returning to the discussion where section **3.2** left off. After this is done, the area must be further restricted to take into account some plates may be obstructed from the airstream by others. The next stage of the algorithm is to project the plates into a two dimensional plane perpendicular to the airstream direction.

The vertices of the plates are described in the coordinate system (x, y, z). A new coordinate system (x', y', z') is determined as follows. The positive x' axis is chosen to be in the direction of the airstream $\mathbf{V} = < v_1, v_2, v_3 >$, with v_1, v_2, v_3 the components of v in (x, y, z) coordinates.

let $\mathbf{A_d = v / |v|}$, $\mathbf{A_y} = < -v_2, v_1, 0 > / |< -v_2, v_1, 0 >|$. And $\mathbf{A_z = A_y \times A_d / | A_y \times A_d |}$.

Then the new coordinates (x', y', z') are given by $x' = < x, y, z > \cdot \mathbf{A_d,}$ $y' = < x, y, z > \cdot \mathbf{A_y,}$ and

z’ = < x, y, z > · $\mathbf{A_z}$, (assuming v_1 and v_2 are not both zero)

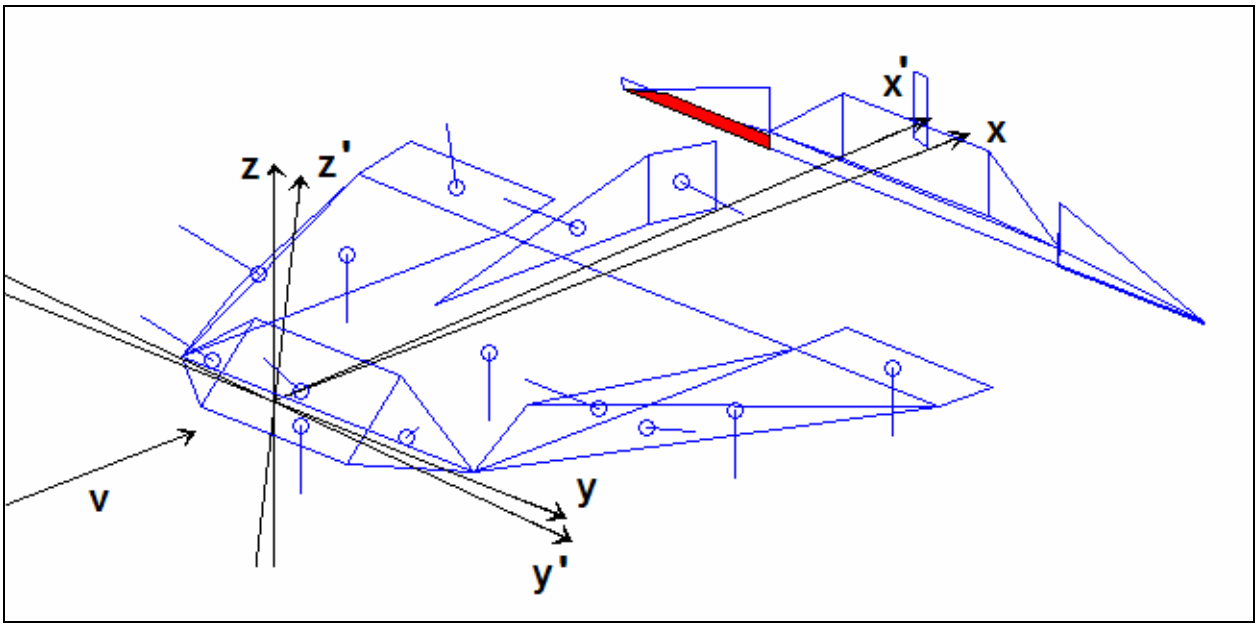

Figure 17: (x,y,z) and (x’,y’,z’) Coordinates

Now the two dimensional projection of the airplane into the y’-z’ coordinate plane is given by (0, y’, z’) or more simply (y’, z’).

3.5 Plates intersections Projection

After projection the plane appears as a collection of areas bounded by polygons. Some of the areas may correspond to where the projected images of two or more plates intersect. This relates to when the airstream is obstructed from one plate by another. The figure below shows one such intersected area for the case of the illustration above, colored in red.

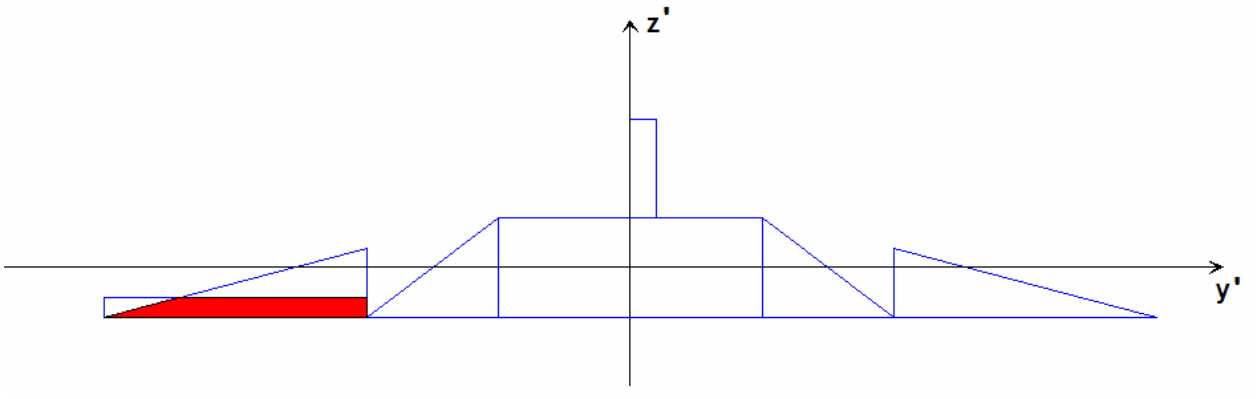

Figure 18: Demonstration Model (intersection of projections)

In the case illustrated above the intersected area corresponds to portions of two different plates.

In the figure below, the centroids of these two portions of the plates are highlighted in red.

The algorithm must decide which of these portions to include and which to exclude.

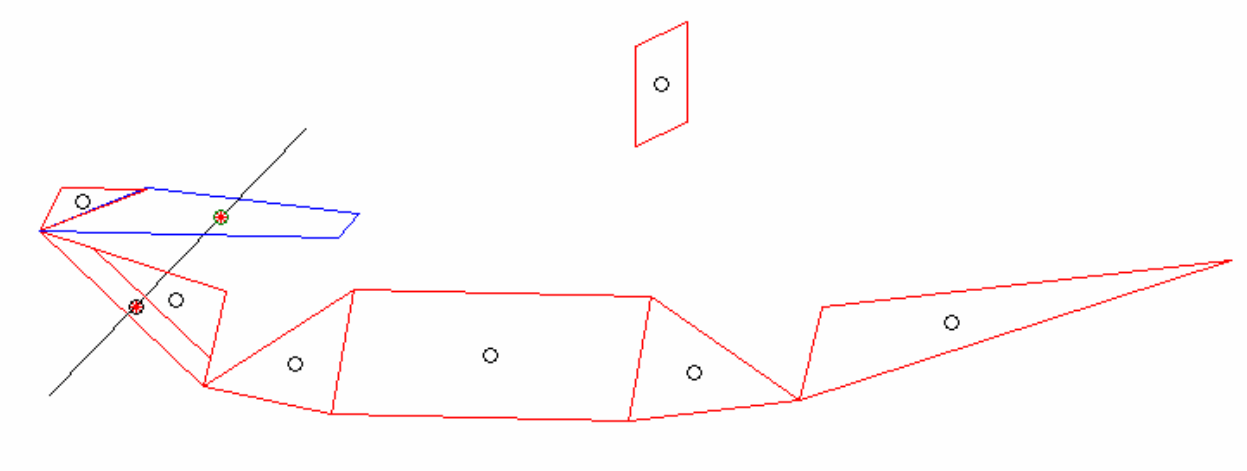

Figure 19: Two Regions with intersecting projections

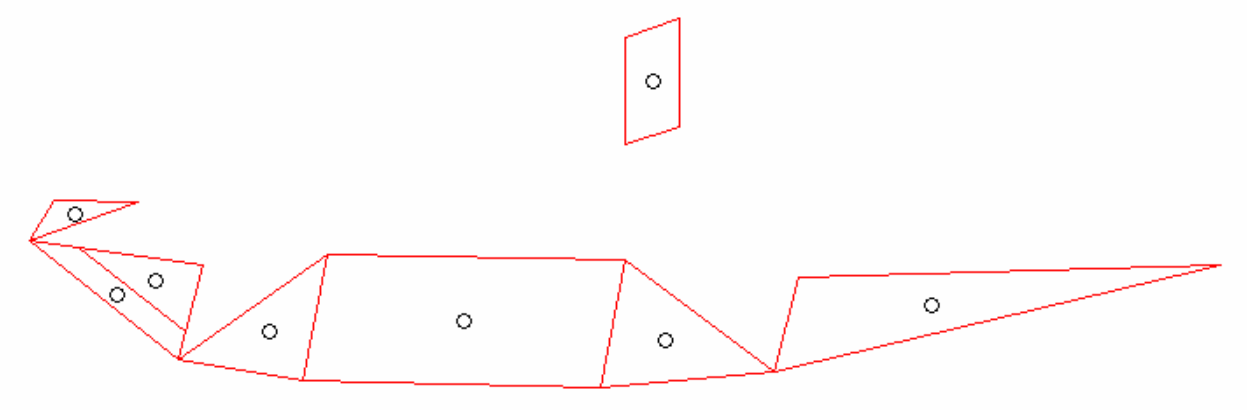

Figure 20: After removal of obstructed surfaces

3.6 Algorithm Capabilities

The algorithm can in fact handle intersections of arbitrary complexity. The discussion below illustrates the algorithm for three plates. Initially the three plates are listed (sorted in an arbitrary order) in positions 1', 2', and 3'. The first thing to do is to calculate the coordinates of the vertices of each of the plates in a coordinate system (x', y', z') defined by the airstream direction vector $\mathbf{A_d}$ as in section **3.4.**

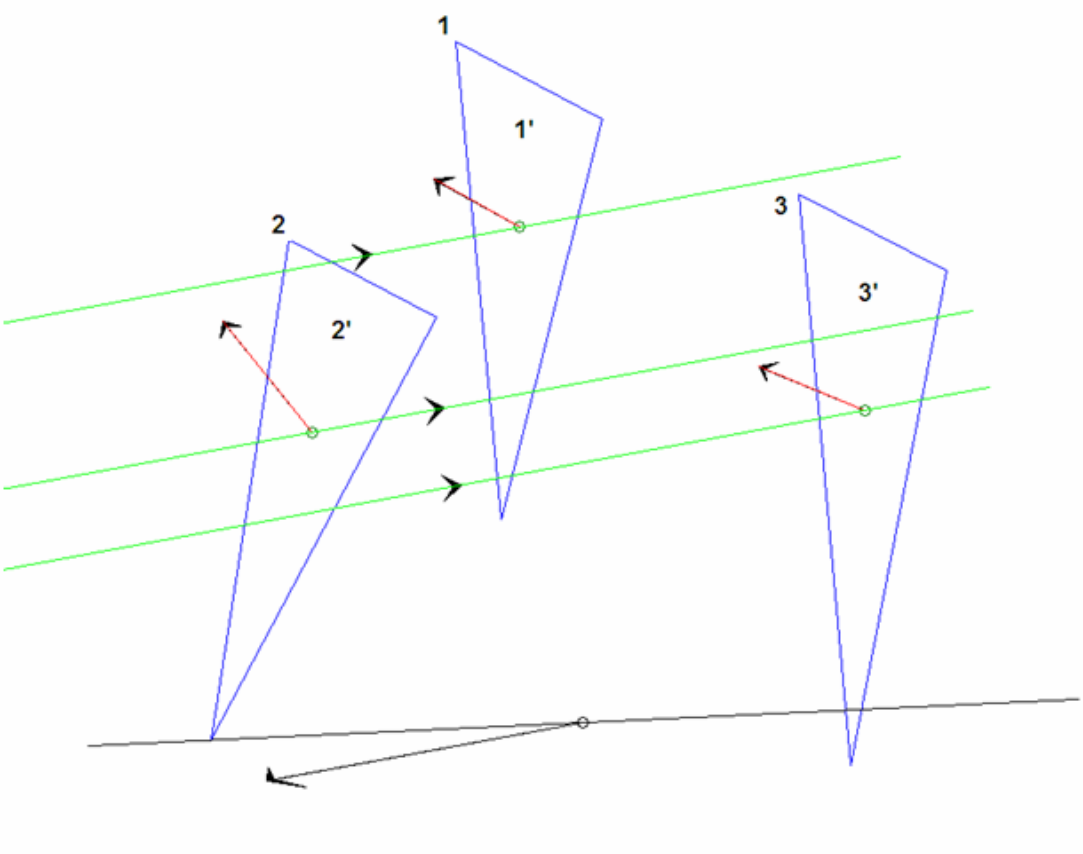

Figure 21: Example of three plates placed in airstream

Each of the vertices has a coordinate on the axis in the direction of the airstream given by

x' = < x, y, z > · $\mathbf{A_d}$, proceeding to calculate (as in **2.3.4**)

y' = < x, y, z > · $\mathbf{A_y}$, and z' = < x, y, z > · $\mathbf{A_z}$.

for the vertices of each plate , the corresponding two dimensional vertices of the plate's projection are the (y', z'). In this example plate 2 has vertices v_1, v_2, v_3 plate 1 has vertices v_4, v_5, v_6 and plate 3 vertices v_7, v_8, v_9. After projection the plates look like this.

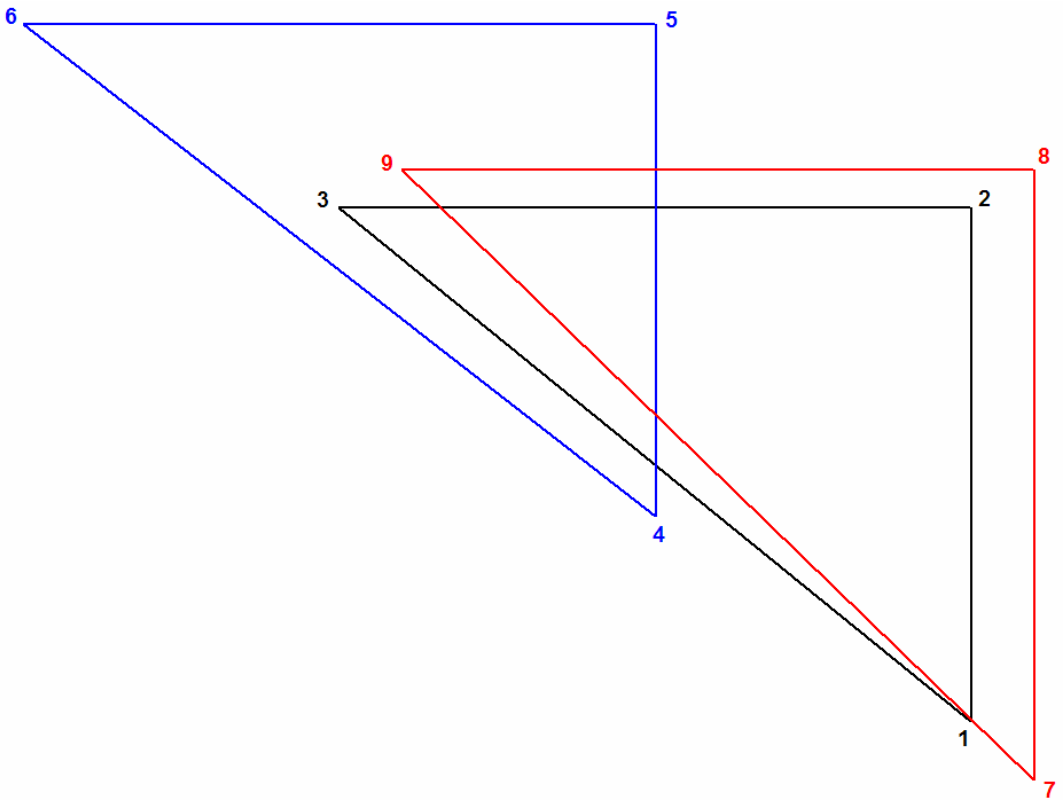

Figure 22: 2D projection of boundaries in Airstream Direction

This divides the total projected area into 7 regions each bounded by a polygon. The figure below shows these regions each with a different color. The legend indicates in which plate's projection a region lies. For example, the region colored green only lies in plate 2's projection. The region colored blue lies in both plate 2's and plate 3's projection. The region colored red lies in the projections of all three plates.

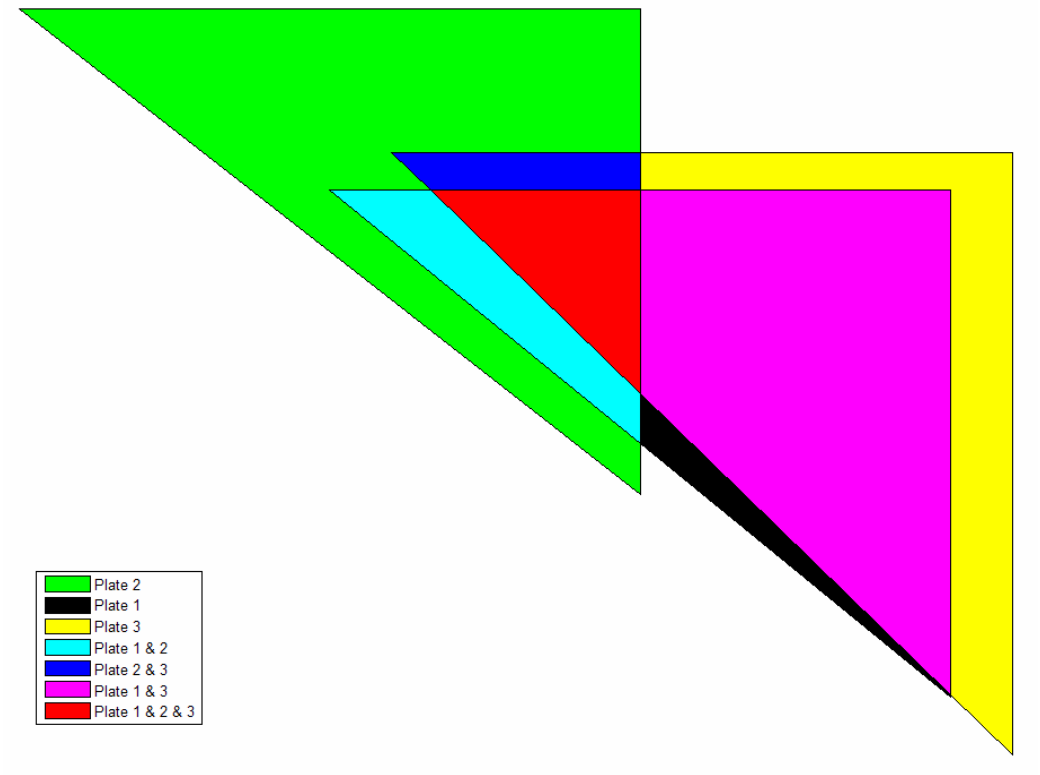

Figure 23: Distinction of intersected areas

To be precise about the example illustrated above the exact details of what is drawn is what follows. The plates are initially described in the (x,y,z) coordinate system as shown in Figure 21.

With the airstream direction of Ad = <0.9960, 0, -0.0899>, the plates are then described in the (x',y',z') coordinate system shown in Figure 21, hence in (y',z') coordinates as shown in Fig.22.

All intersections are calculated to find the 7 regions with the results below. The MatLab function **Polygons_intersection** was used to make the calculations. These Different regions are labeled with different numbers, see Figure 23.

Illustrating the results in three dimensions to demonstrate the additional capabilities of the **Polygons_intersection** function and the algorithm, this example is given to see that obstructions that cause holes are also handled, Fig.24 .

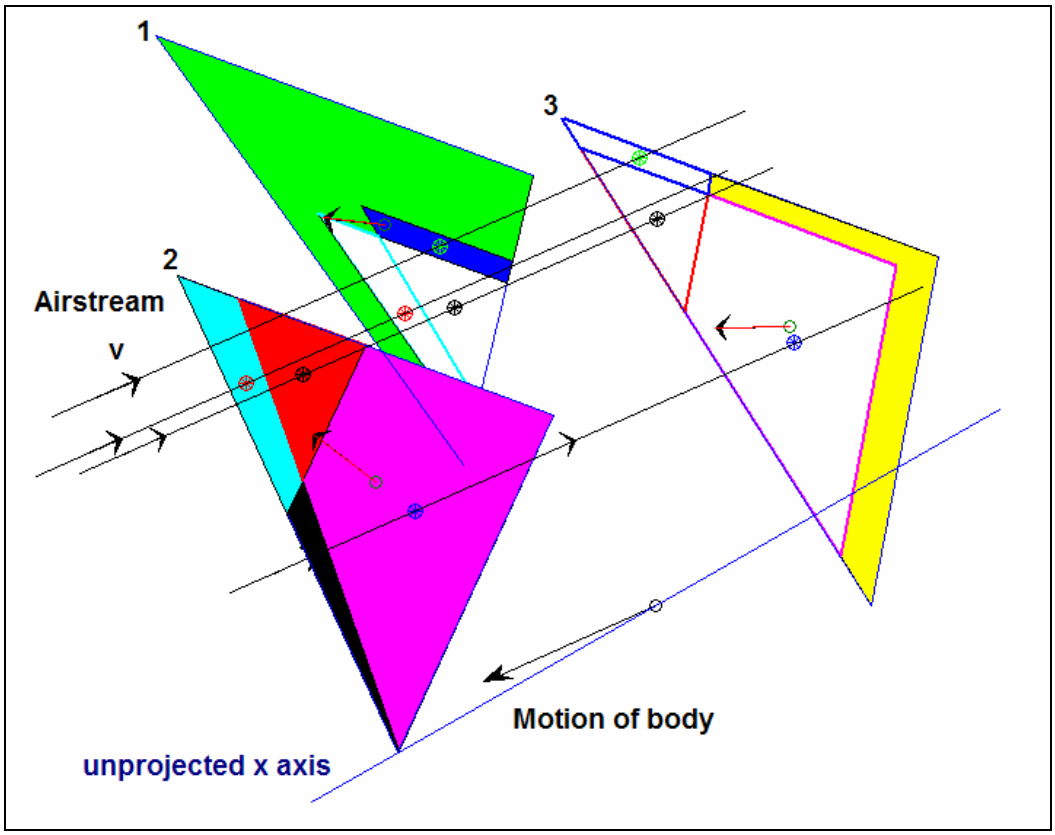

Figure 24: Illustration of comparison points

Passing the projections of the plates to a MatLab function with the call, **S** being a structure with the list of plates, **[Geo] = Polygons_intersection(S,0).**

3.7 Algorithm General Considerations

In this section this stage of the algorithm is described in generality. At this point the collection of plates have been restricted to a sub-collection

Plate_1**',** Plate_2**', ...,** Plate_n**' ,** based on the angle between the direction of the airstream **v** and the normal's, as in section 2.3.2 (angle $\boldsymbol{\theta} < \pi/2$). As in section **3.4** define x', y', z' coordinates for the vertices of each plate 1 through n. namely, $\mathbf{A_d} = \mathbf{v} / |\mathbf{v}|$, $\mathbf{A_y} = < -v_2, v_1, 0 > / |< -v_2, v_1, 0 >|$, and

$\mathbf{A_z} = \mathbf{A_y} \times \mathbf{A_d} / | \mathbf{A_y} \times \mathbf{A_d} |$. x' = < x, y, z > · $\mathbf{A_d}$, y' = < x, y, z > · $\mathbf{A_y}$, and z' = < x, y, z > · $\mathbf{A_z}$.

Now restrict the coordinates defining the bounding polygons of the plates to their (y', z') coordinates. This is the so called projections of the plates. This list of projected plates is passed to an appropriate procedure to find all of the intersected and non-intersected regions. This **Polygon Intersection** function should have the following features. First of all it returns the bounding polygons of each sub region. Second it lists the indexes of the plates intersecting on regions where more than one plate overlaps.

The MatLab function **PolygonIntersection** does this nicely returning additional information such as areas and handles complex intersections of many plates which may contain holes.

After this, a new list of plates and portions of plates, but only with the y', z' coordinates specified $Q_1, Q_2, ..., Q_m$ is generated together with a list of sets of indexes $I_1, I_2, ..., I_m$.

The indexes specify which plate or plates the region corresponded to. When there is only one plate projected onto a region this set has only one index being that of the plate in question. When there is more than one plate projected onto a region there is one index for each plate.

At this point for the regions with multiple indexes the index of the plate that is struck by the airstream must be determined. This is done by calculating appropriate comparison points on each of the plates involved. The region projected into y', z' coordinate plane has a two dimensional centroid (C_y', C_z'), the line in x', y', z' coordinates that passes through (0, C_y', C_z')

intersects each of the plates when described in x', y', z'. These points are the comparison points chosen, with the x' coordinates indicating the positions of the plates' sub-regions, corresponding to the intersected area, relative to the airstream direction. The index of the plate whose comparison point has the least x' coordinate is chosen. This allows constructing a list of the associated normal vectors n_1**'**, n_2**'**, **...,** n_m**'**, $\mathbf{n}_i$**'** being the normal vector of plate i. These being calculated in the (x', y', z') coordinates. From the equation of a plane

$\mathbf{n} \cdot \langle x' - x_0', y' - y_0', z' - z_0' \rangle = 0$, the remaining x' coordinates for each of the regions in the list Q_1**,** Q_2**, ...,** Q_m are calculated. This results in another list of the regions with their bounding polygons defined by x', y', z' coordinates R_1**',** R_2**', ...,** R_m**'.** Reasoning similarly as in section **2.3.3** a final list of the regions with their bounding polygons defined in x, y, z coordinates

R_1**,** R_2**, ...,** R_m**,**together with their normal vectors n_1**,** n_2**, ...,** n_m are known

3.8 Computation of Forces and Moments

Returning to the demonstration aircraft figures the portion of the plane which the airstream impacts is partitioned into pieces with centroids and normal vectors known see figure below.

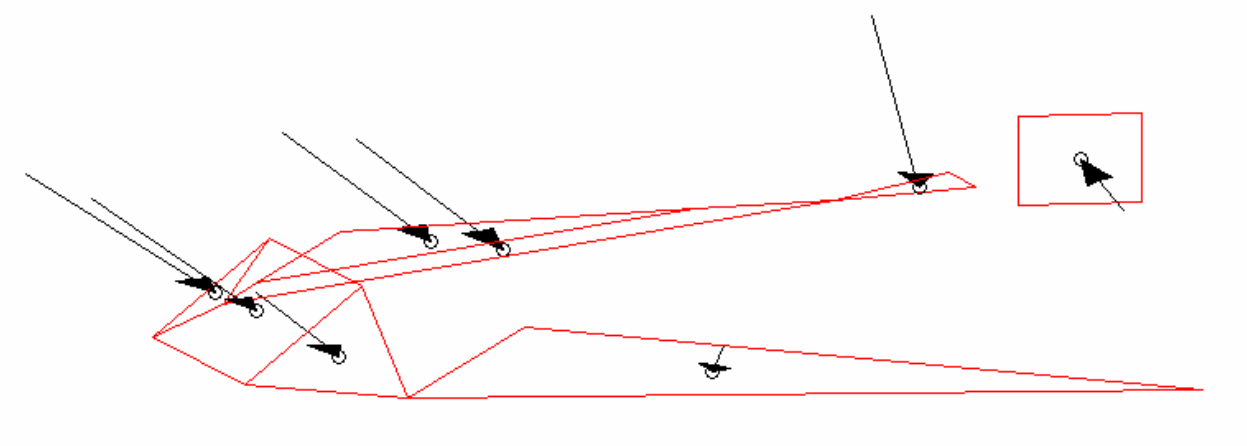

Figure 29: Direction of Force Vectors

At this point some aerodynamic model is used to predict the force to apply to each sub-region, as discussed in the preceding sections, of the aircraft surface. Call the magnitudes of these forces

$F_1, F_2, ..., F_n$. then the total effect of these forces is the sum of the local forces

$-F_1n_1, -F_2n_2, ..., -F_nn_n$. When it can be expected that the rotational velocities are small the interactions between translational and rotational motion can be neglected, for the most part.

Let the plane have linear moments of inertia M_x, M_y, M_z and moment of inertia tensor

$$\mathbf{I} = \begin{bmatrix} I_{11} & -I_{12} & -I_{13} \\ -I_{21} & I_{22} & -I_{23} \\ -I_{31} & -I_{32} & I_{33} \end{bmatrix}$$

$\mathbf{I_x = <1, 0, 0> I<1, 0, 0>^T}$, $\mathbf{I_y = <0\ 1, 0> I<0, 1, 0>^T}$, and $\mathbf{I_z = <0, 0, 1> I<0, 0, 1>^T}$

The translational force on the airplane is given by

$$\mathbf{F} = \sum(-F_i n_i) \qquad (\ \mathbf{11}\ )$$

The translational acceleration

$$\mathbf{a = F/}m \quad (\ 12\ )$$

where m is the total mass of the airplane. The torque on the aircraft is given by

$$\boldsymbol{\tau} = \sum \mathbf{r_i} \times (-F_i n_i) \quad (\ \mathbf{13})$$

where $\mathbf{r_i} = \langle x_i, y_i, z_i \rangle$ is the position of region it's centroid. The angular momentum is given by

$$L = I\boldsymbol{\omega} \qquad (\ \mathbf{14}\ )$$

where $\boldsymbol{\omega}$ is the angular velocity. The torque on a body determines the rate of change of the body's angular momentum $\boldsymbol{\tau} =$ dL/dt while the angular accelerations are found by $dL/dt = Id\boldsymbol{\omega}/dt = \mathbf{t}$ and $d\omega/dt = I^{-1}\ t$.

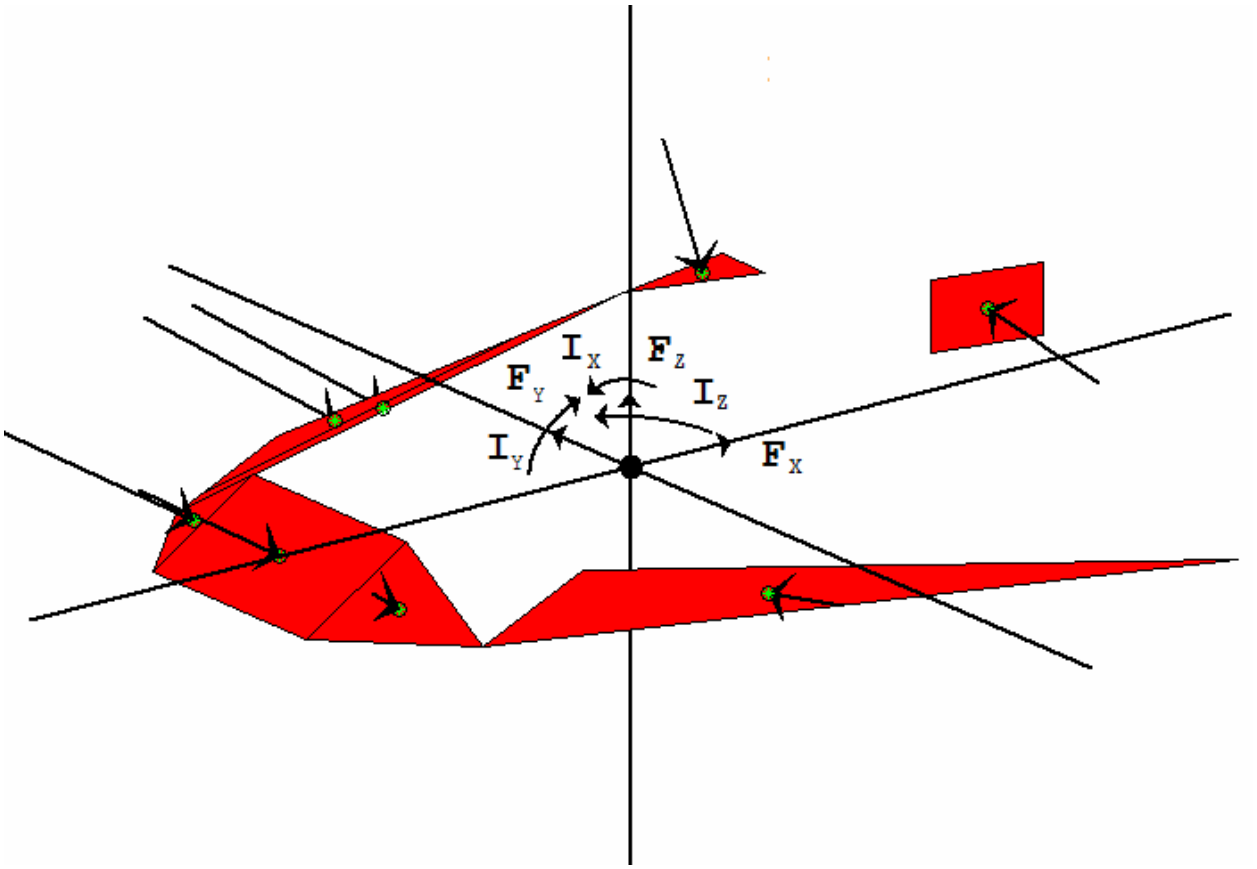

Figure 30: Reference axes for moments

The accelerations determined both linear and angular are with respect to the fixed x, y, z coordinates system (body axis) of the aircraft. Since the airplane may not be moving in a direction coinciding with its center line, these accelerations must be converted to an absolute coordinate system. This is easily accomplished since the difference between the plane's coordinate system and an absolute depend on the planes direction of motion (disregarding the effect of wind). A rotation from the one direction to the other accomplishes this.

VI. Results for model of X-43A

The calculation of the force on a single plate when using the inclination method only depends on the angle of inclination, the area of the plate, and the dynamic pressure. In fact it depends on the dynamic pressure directly, as does the sum of the forces over all of the impacted plates. This allows reporting lift, drag, and pitching moment as dimensionless quantities depending on the plane's angle of incidence with the airstream. For the plots generated using the algorithm below the ordinate (vertical axis) is multiplied by the dynamic pressure to obtain the actual lift, drag, or pitching moment for whatever airspeed and density.

The algorithm described in chapter IV returns a list of areas together with their centroids and normal vectors. From the normal vectors the angle of inclination (with respect to the airstream) is easily calculated for each of these flat regions. This together with their areas is all that is needed to calculate these dimensionless forces and their sums.

The first set of plots used this model of the plane (engine removed).

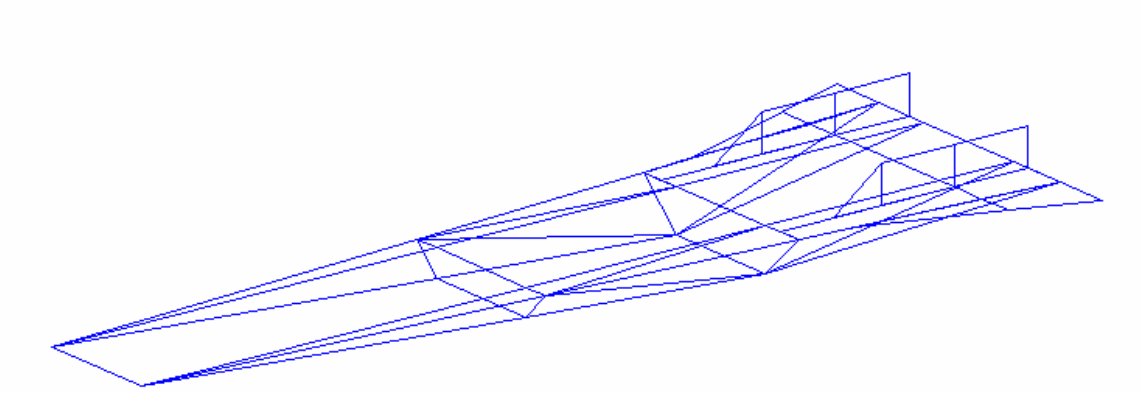

Figure 31: Drawing of the X43 Model (engine removed)

The center line is taken to be the x axis in the coordinate system in which plane's plates are defined. It does not necessarily represent a center line with respect to the balance of aerodynamic forces.

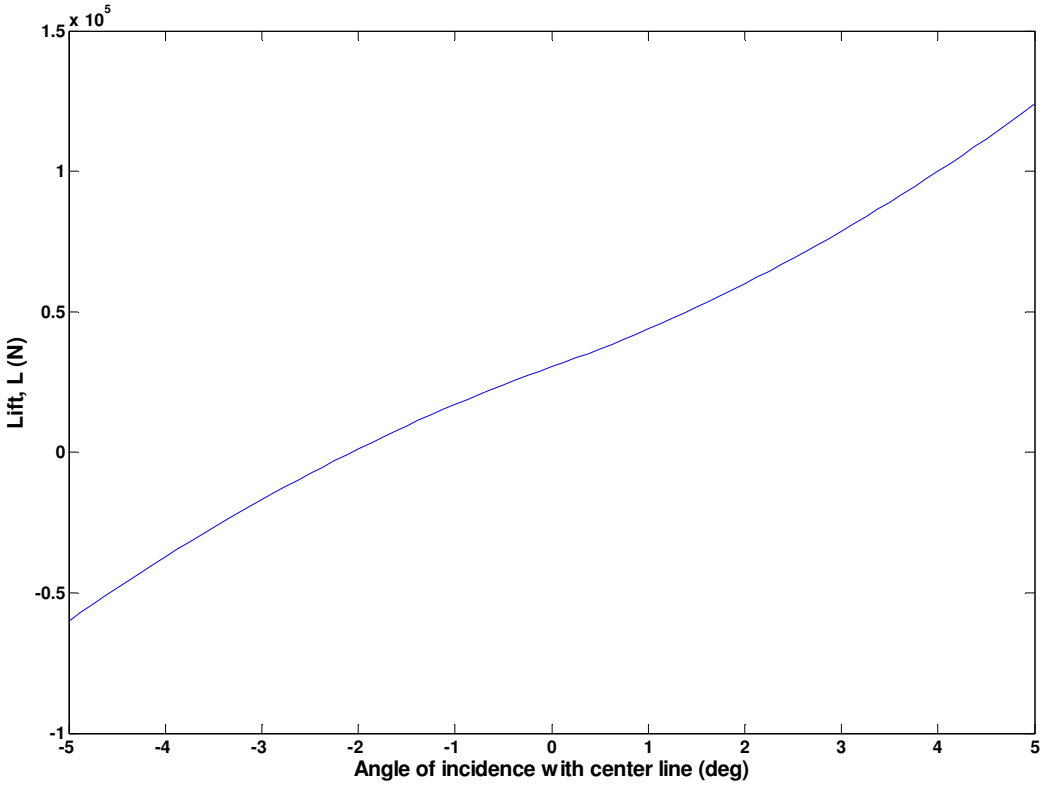

Figure 32a: Lift vs angle of incidence

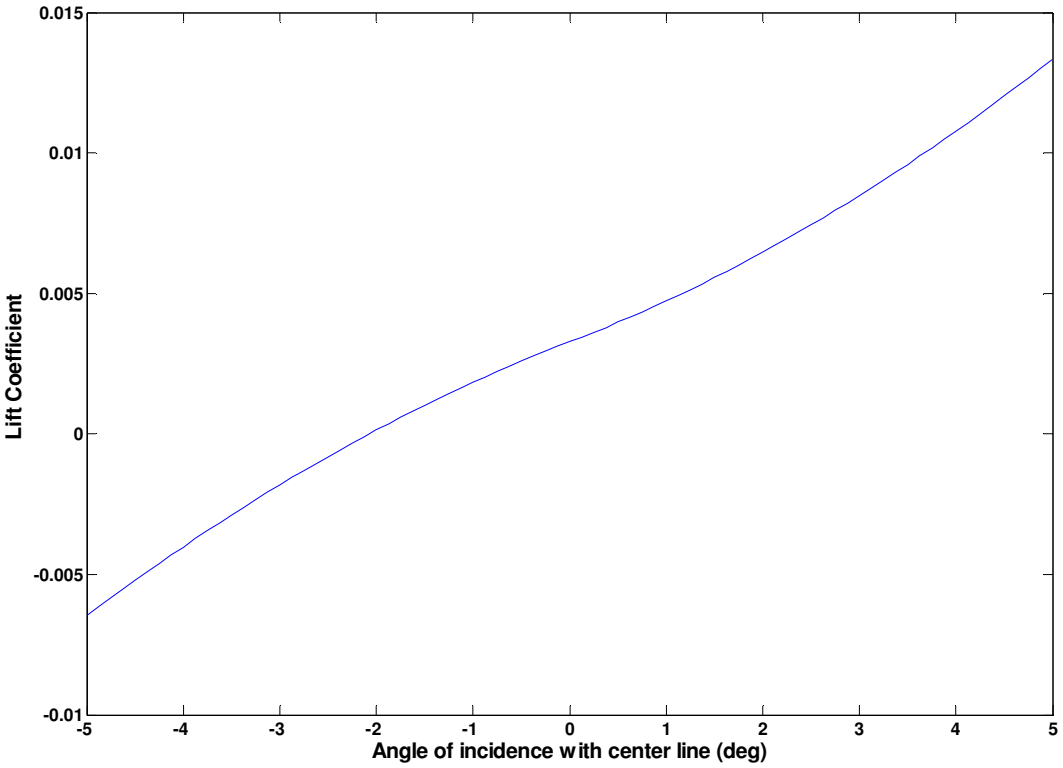

Figure 32b: Lift Coefficient vs angle of incidence

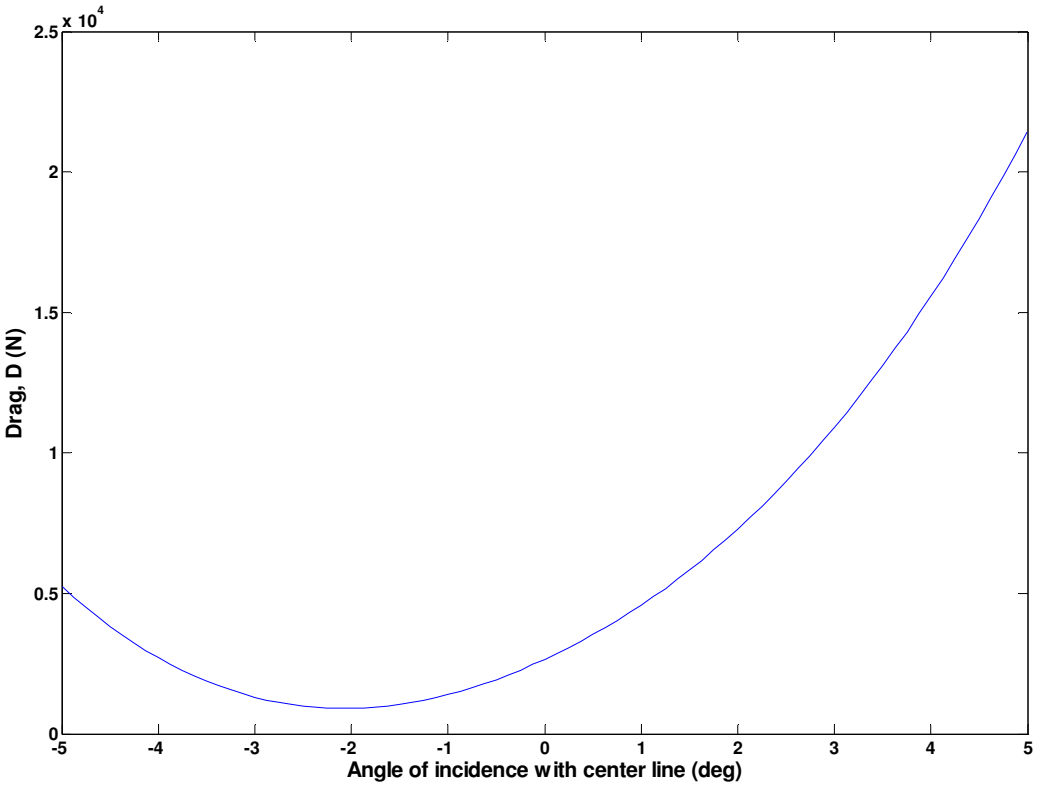

Figure 33a: Drag vs angle of incidence

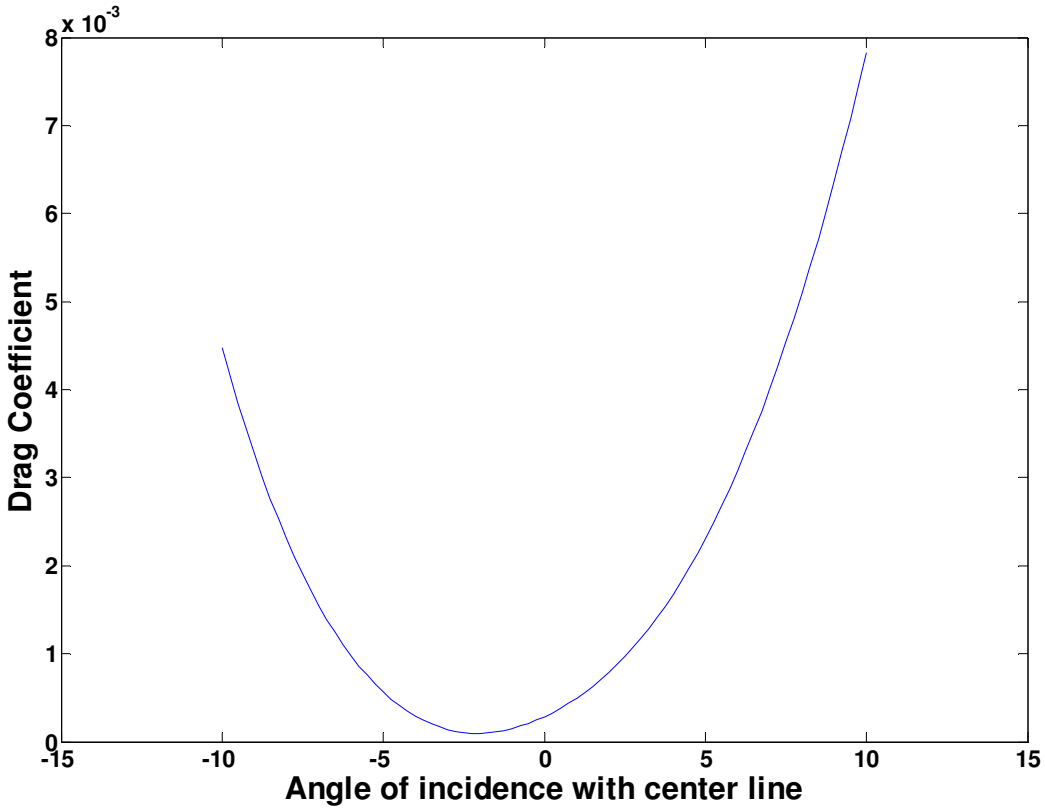

Figure 33b: Drag Coefficient vs angle of incidence

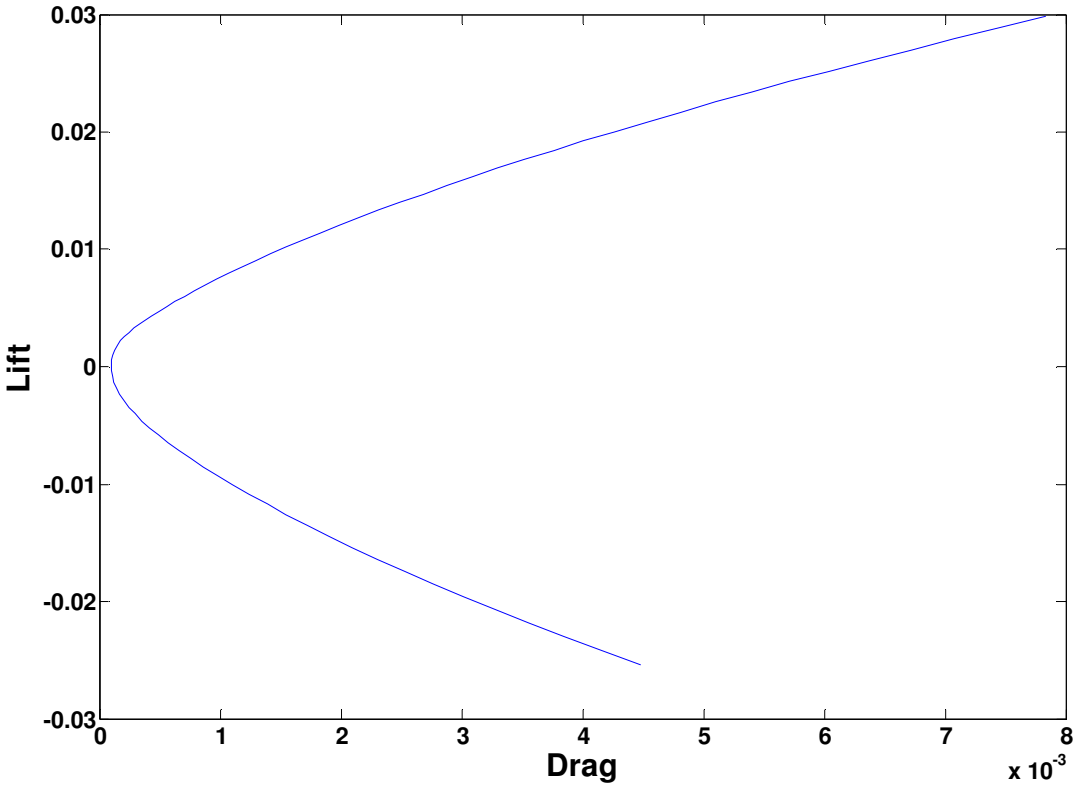

Figure 34: Drag Polar

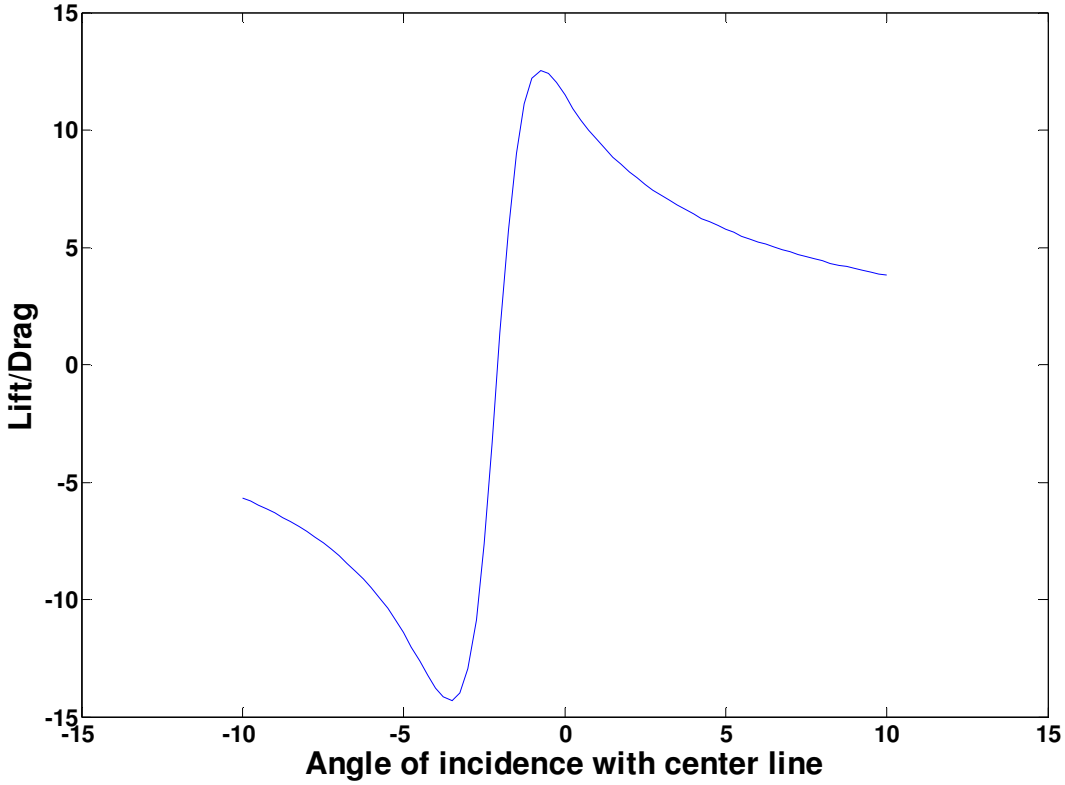

Figure 35: Lift to Drag Ratio

To find the aerodynamic center of the plane the center point from which the moments of inertia are measured with respect to was moved to find where the torque with respect to the y axis was minimal. Moving the point in the x direction will cause an increase in this torque. The best point for minimum pitching moment is shown in Figure 36.

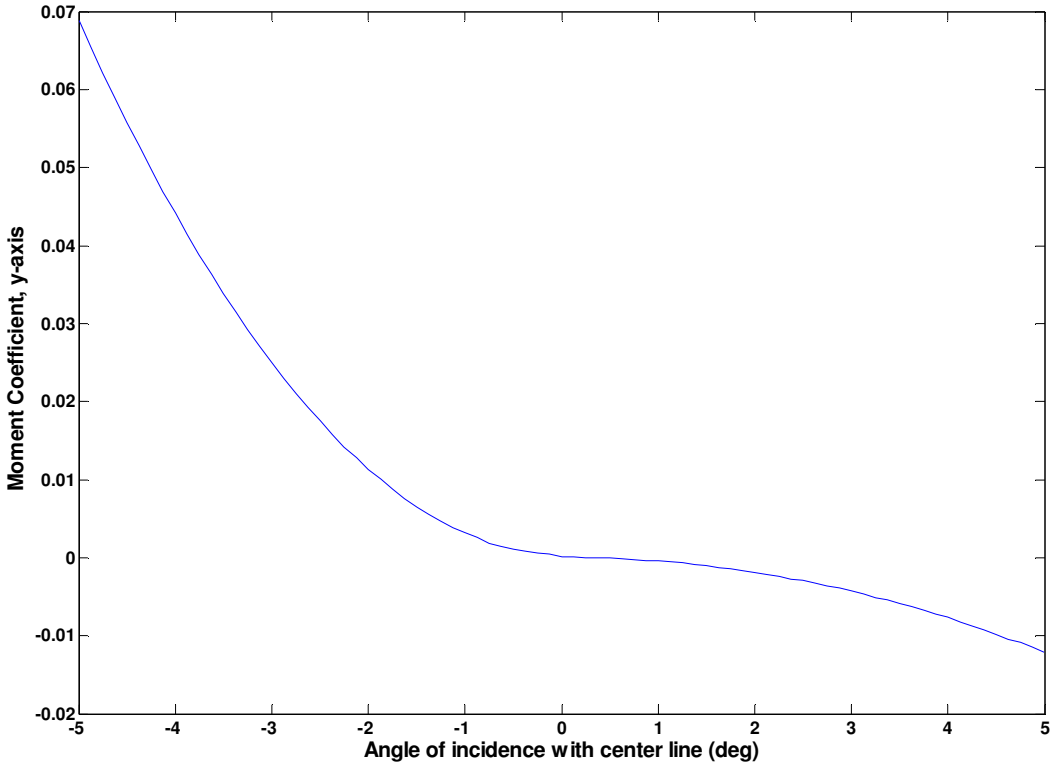

Figure 36a: Pitching Moment Coefficient ac = 21.35 feet

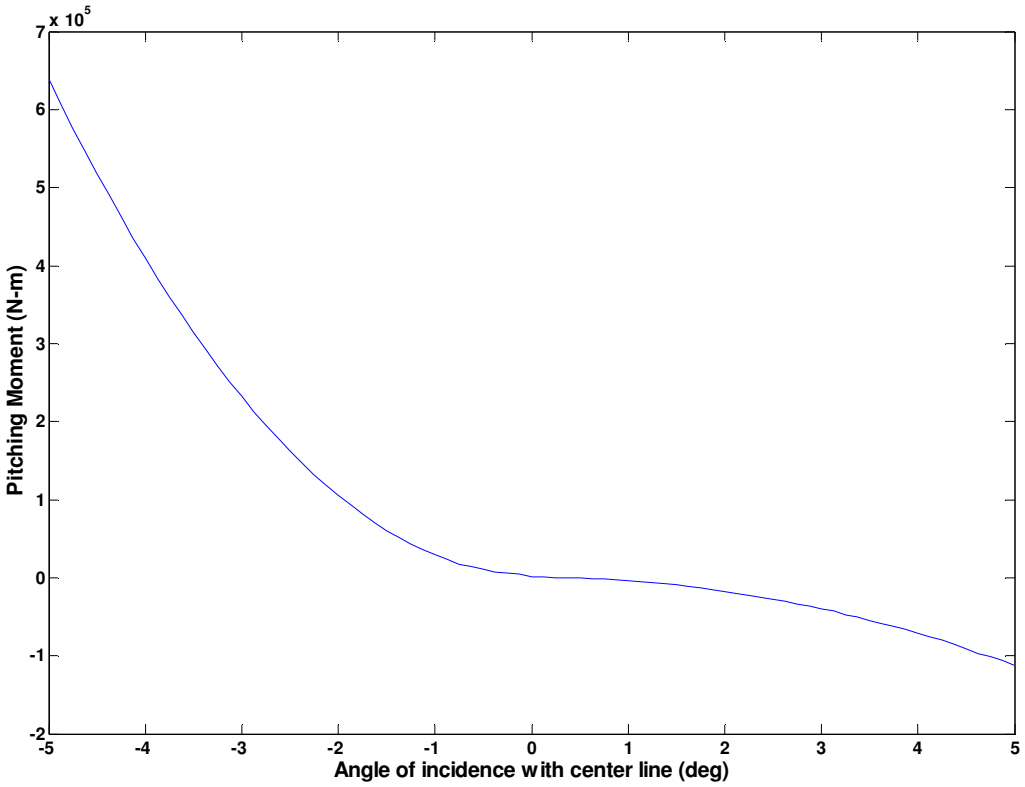

Figure 36b: Pitching Moment ac = 21.35 feet

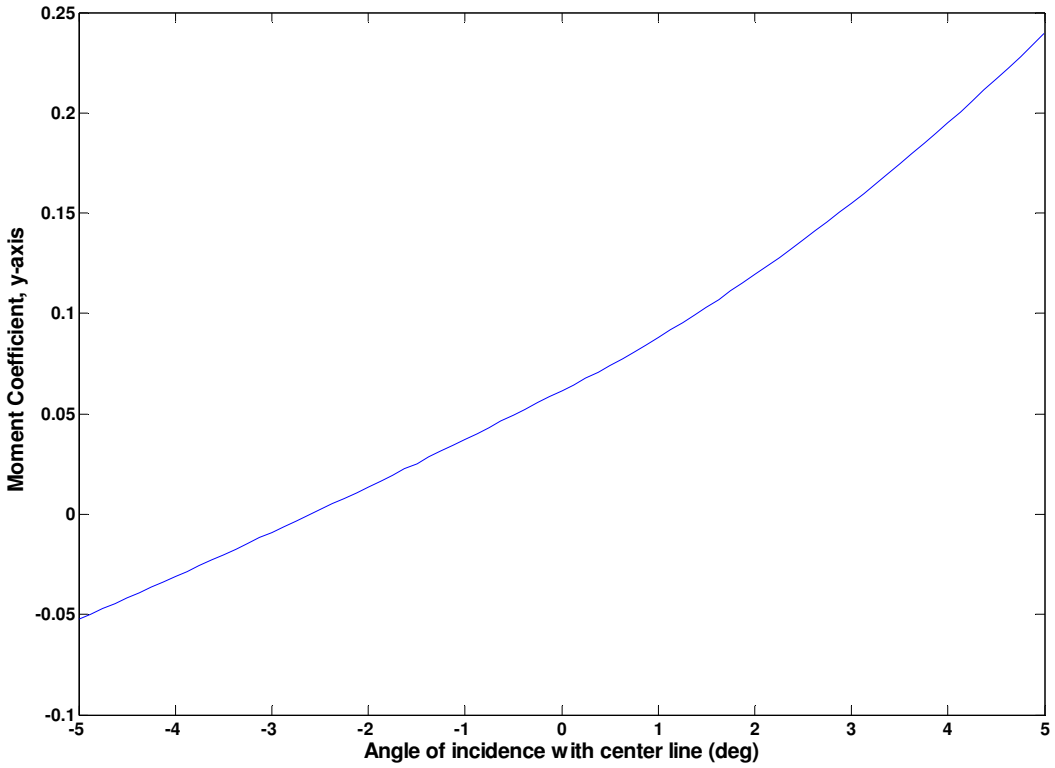

Figure 37: Moment Coefficient with respect to the y-axis at x = 40 feet

The moment with respect to the y-axis at x = 40 feet is tremendously greater than that at x = 21.35. This would imply that if the aircraft's center of gravity is at x = 40 ft (for example) the plane then would have stability problems.

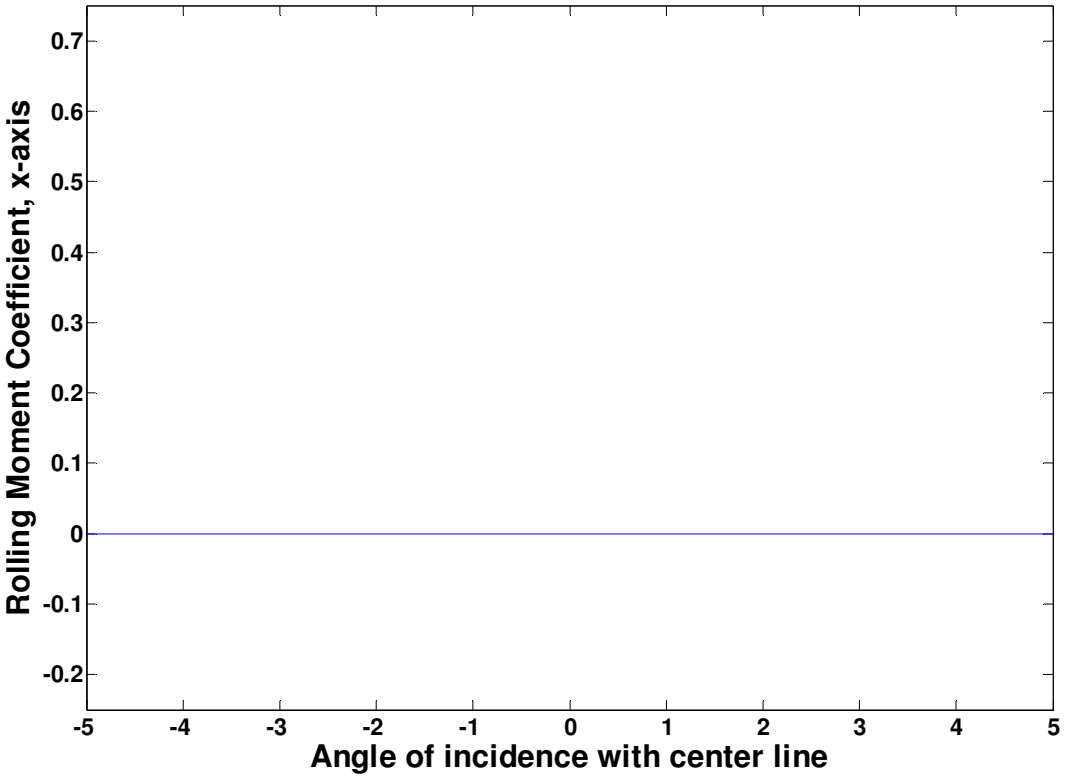

Figure 38: Moment with respect to the x-axis at x = 40 feet with zero deflection angle of the control surfaces

Due to symmetry, the moments with respect to the x and z axis are expected to be sezo as shown in Figures 38 and 39.

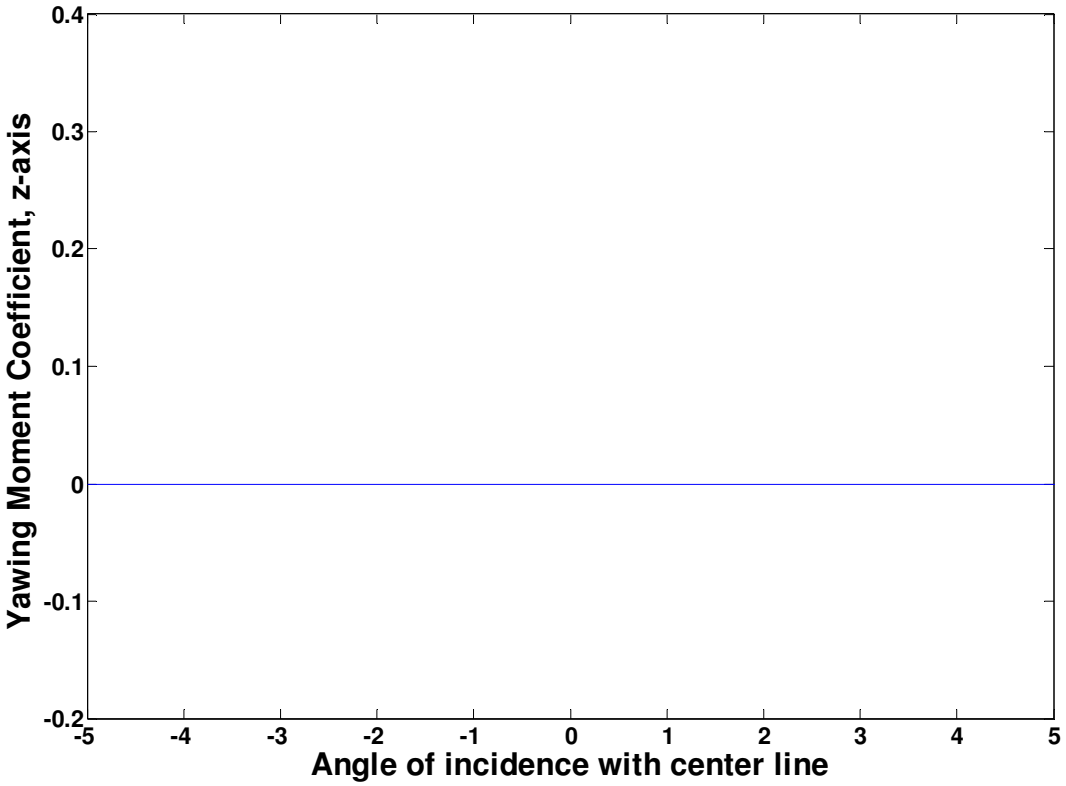

Figure 39: Moment with respect to the z-axis at x = 40 feet with zero deflection angle of the control surfaces

Making a stabilizer adjustment can eliminate this torque for a certain angle of inclination.

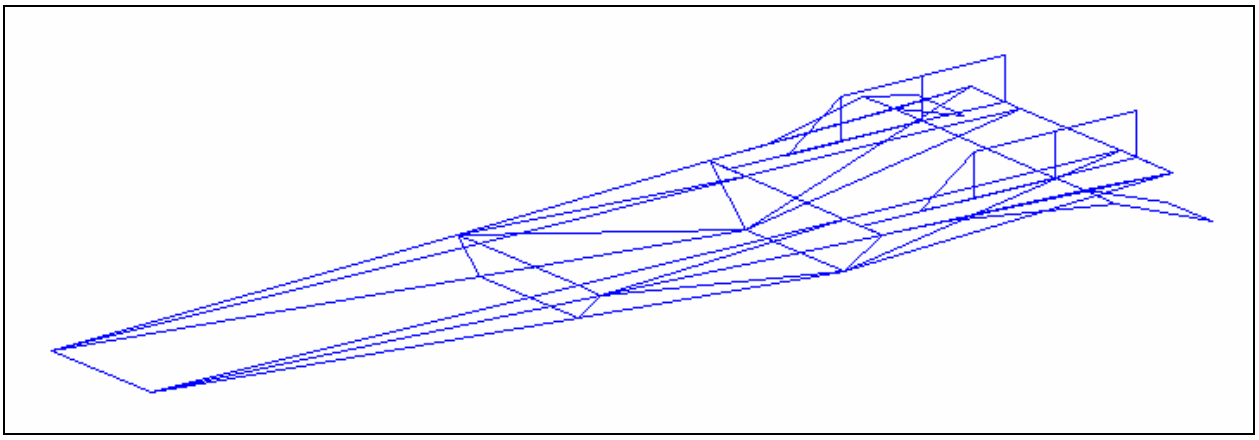

Figure 40: illistration of stabalizer deflection

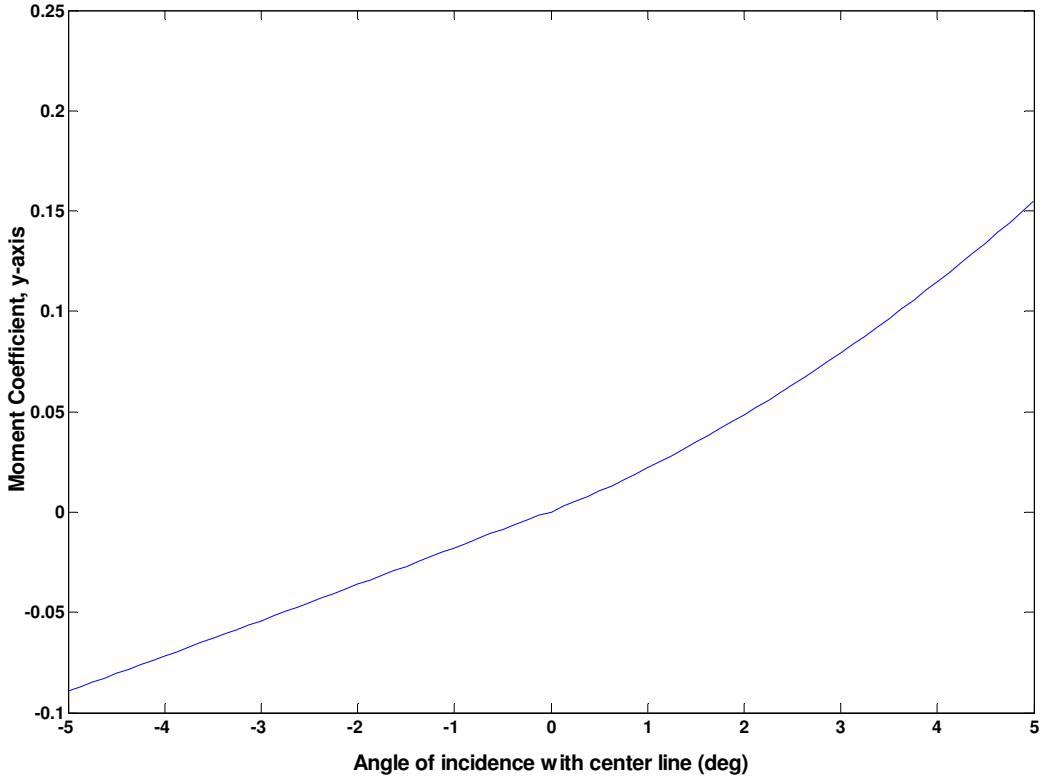

Figure 41: Moment Coefficient with respect to the y-axis with -0.4 radians deflection angle of horizontal stabalizer

In figure 37 the moment with respect to the y-axis about x = 40 ft is reduced from about 0.06 to zero by deflecting the airplane's stabilizers as shown in figures 40 and 41. It was also of interest to study the torque with respect to the z axis as it depends on rudder deflection.

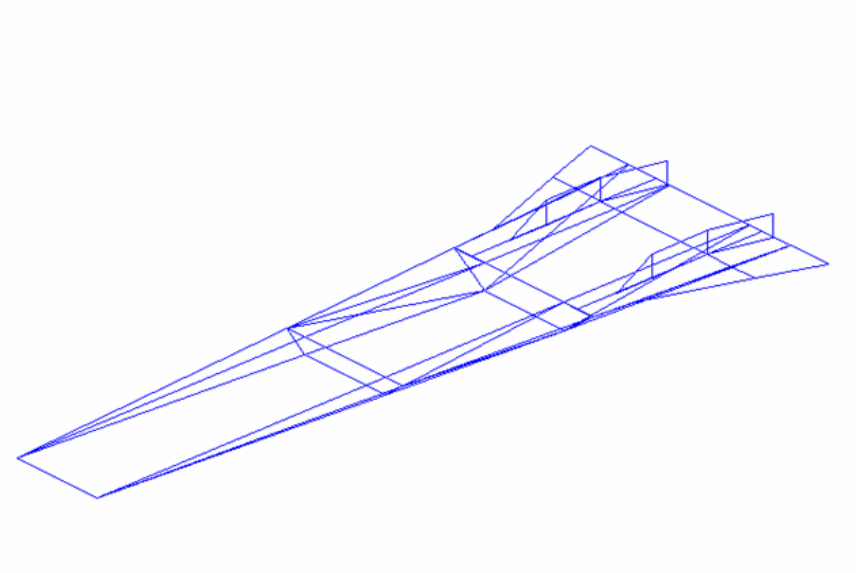

Figure 42: Rudder deflection exaggerated by a factor of 10

The following three figures show the effect the angle of inclination makes on the amount of rudder surface impacted by the air.

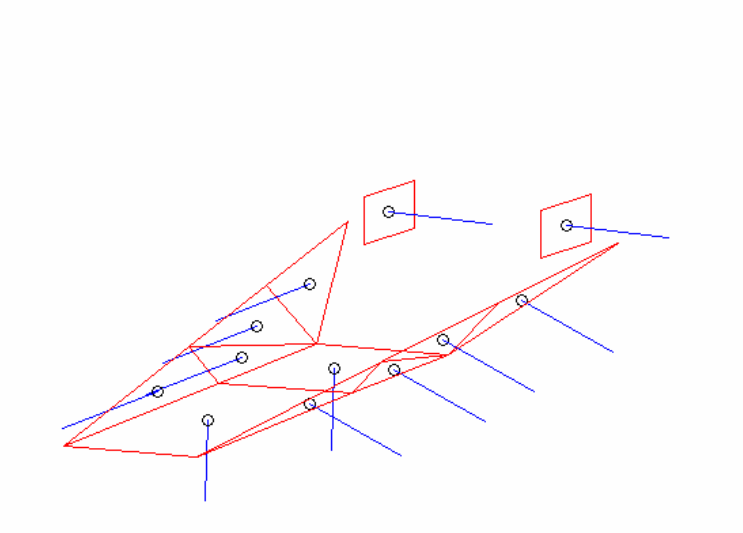

Figure 43: Angle of inclination = 0 degrees

At an angle of inclination of zero degrees, the rudders are fully impacted by the airstream whenever they have any deflection as shown in figure 43.

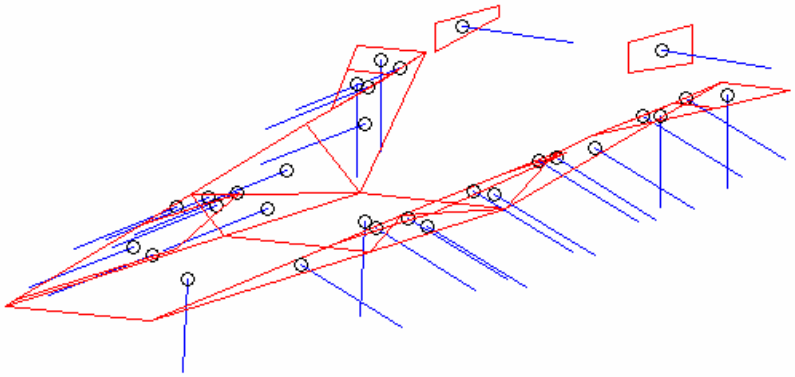

Figure 44: Angle of inclination = 2.5 degrees

At an inclination of 2.5 degrees the rudders impacted areas are redused to about half of their original area. The airstream is partially blocked from impacting the rudders total area by the front portion of the aircraft. At five degrees the rudders are completely obscured by the shadow resulted for the airstram impacting the front of the airplane. This has the implication that when the aircraft is pitched up there is a lost in streering capabilities.

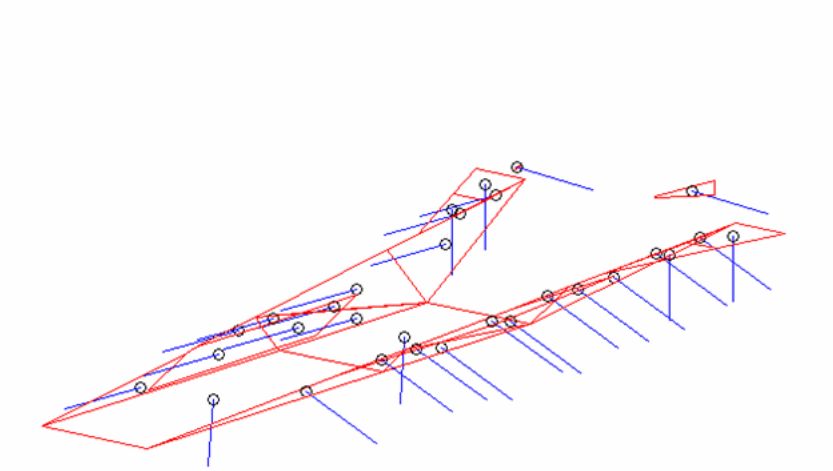

Figure 45: Angle of inclination = 5 degrees

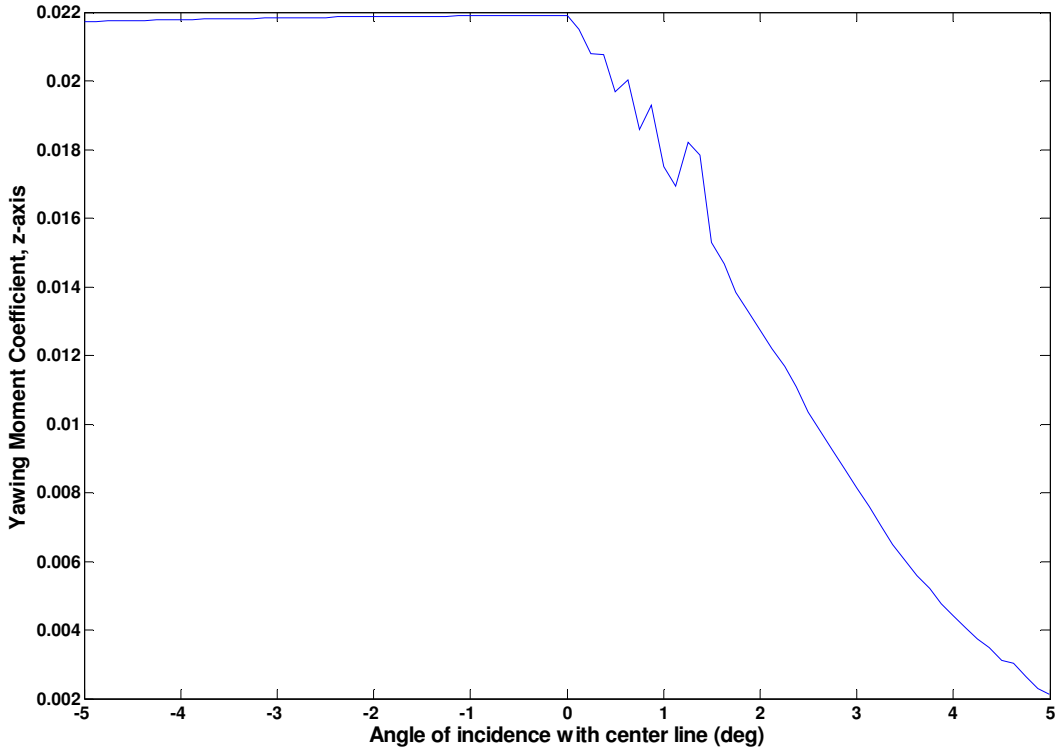

Figure 46: Yawing Moment Coefficient with respect to the z-axis with -0.2 radians deflection angle of rudders

The figure above shows the strength of the yawing moment at -0.2 rad. It also indicates that the airplane is stable when the pitch angle is negative, and becomes unstable as the pitch angle increases.

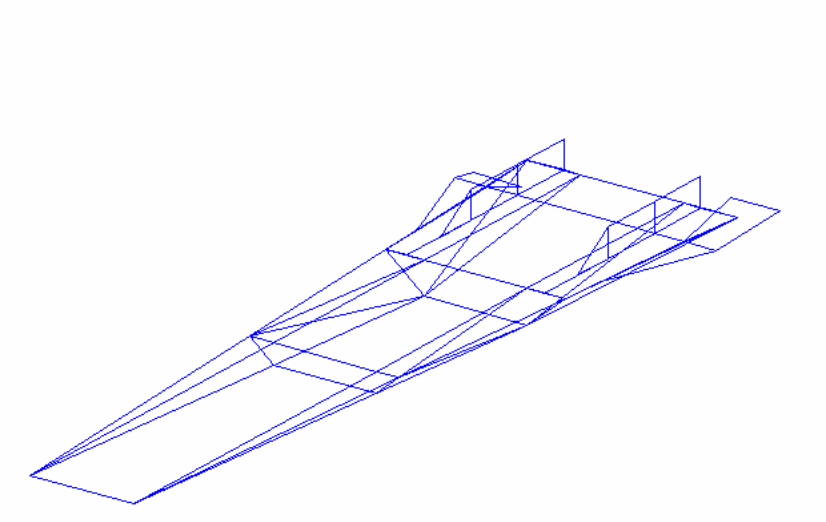

Figure 47: Stabilizer deflections exaggerated by a factor of 50

The next 4 plots show the effect of adding the engine. What is noticed is that when the angle of inclination is such that the engine is impacted by the air this results in an increase in the drag, figure 48 was generated by the program and shows the engine added

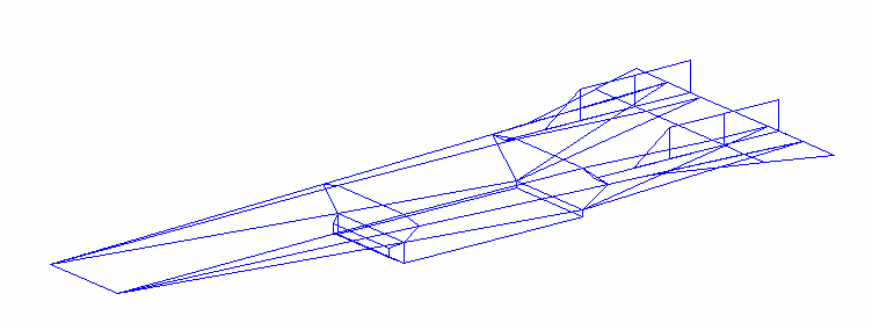

Figure 48: Drawing of the X43 Model (engine included)

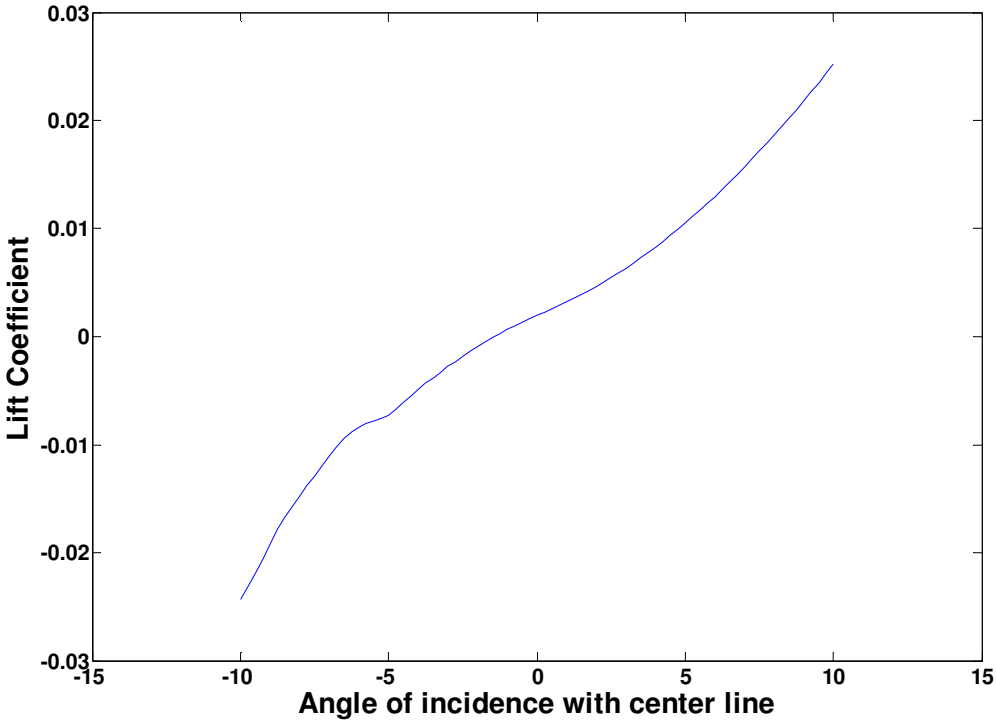

Figure 50: Lift Coefficien vs angle of incidence

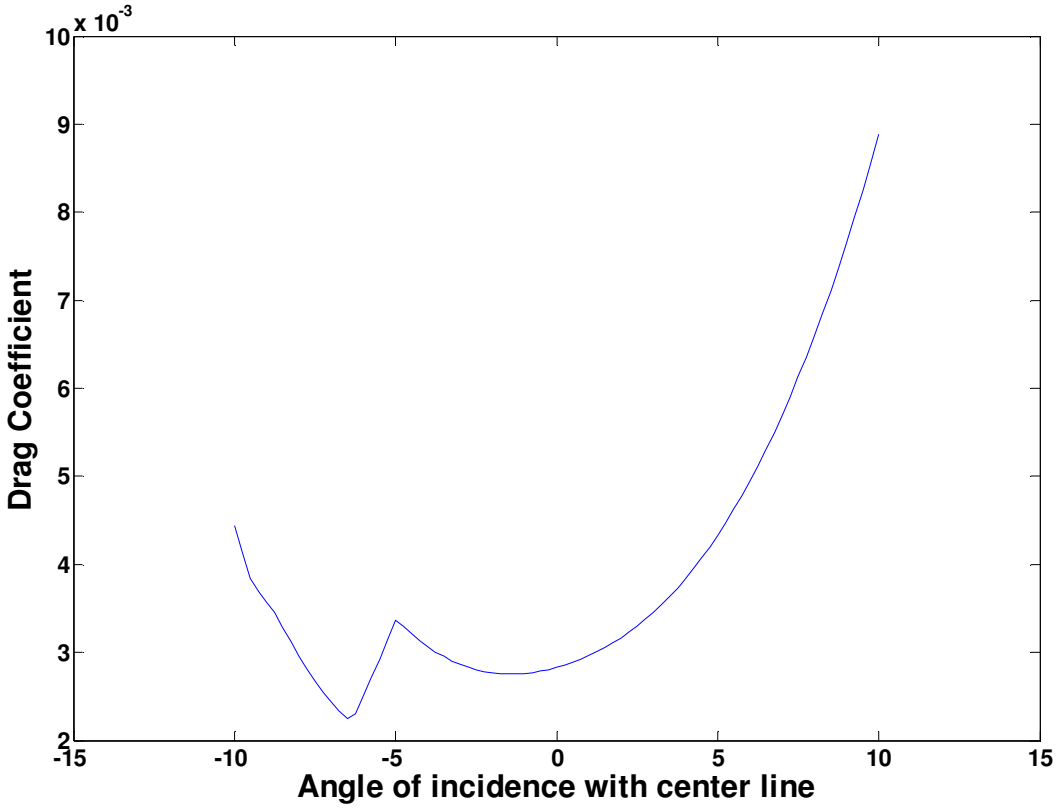

Figure 51: Drag Coefficien vs angle of incidence

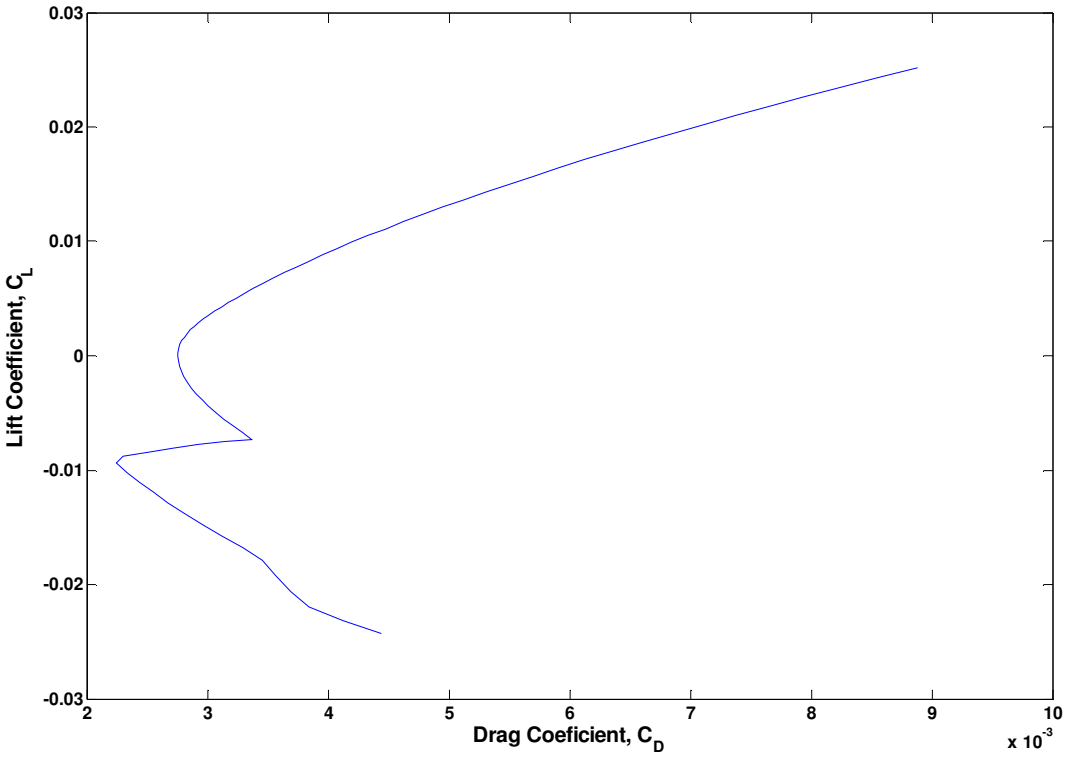

Figure 52: Drag Polar

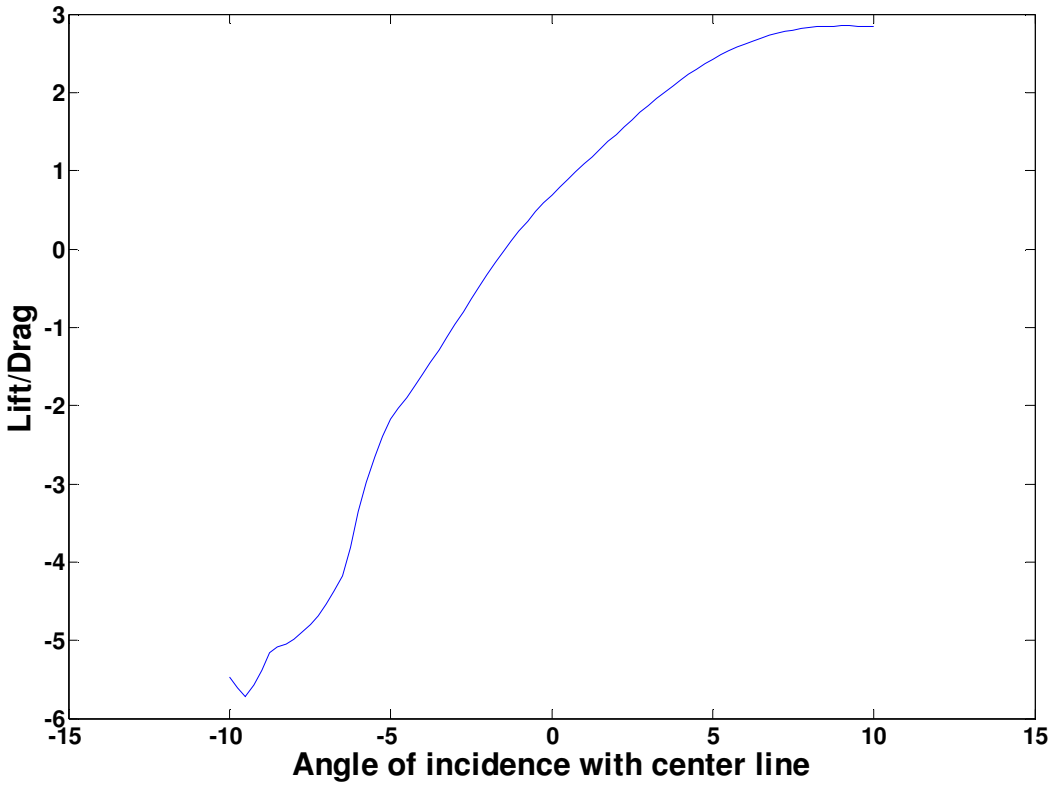

Figure 53: Lift to Drag Ratio

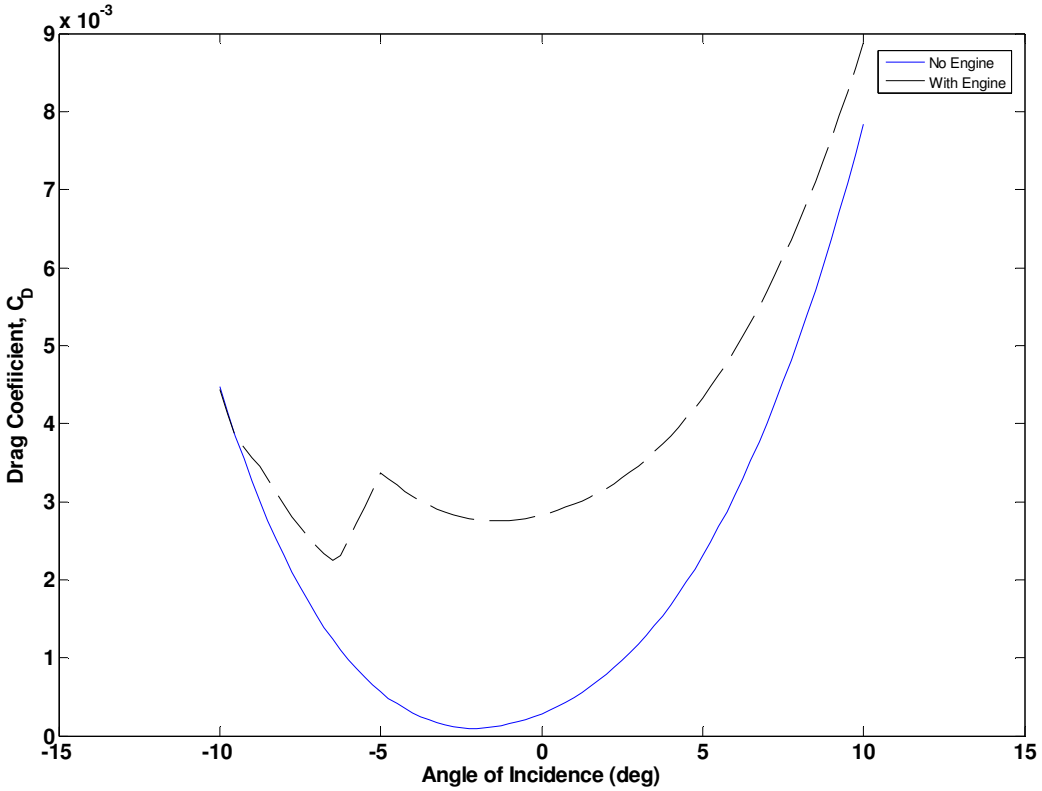

Figure 54: Drag coefficient with and without engine

V. IMPLEMANTATION OF THE ALGORITHM

5.1 General Considerations

The description of the airplane should be intuitive and easy to input into the program. Additionally, small adjustments to the design, such as the exact positions of the vertices of the plates, are easily made. This was accomplished by placing the details of the plane in several different text files segregating the information as vertices, plates, controls, and moments.

The program, as designed, allows adjusting the coordinates of a vertex or adding or removing a plate without making any changes except in the text input files, see appendix.

5.2 Organization of the program

The program is written in a number of MatLab script files. For each input file a separate script reads the file and places its content in variables, arrays, and associative cell arrays. A script is written to read all of the input files with a single command. Each of the different stages of the algorithm, together with their sub-procedures, is placed in separate function scripts. One of these functions calculates the force per unit area, normalized by dynamic pressure, and is in its own MatLab function file so that the program can be adjusted to different aerodynamic models by only editing this one function (aero_params.m).

There are many mathematical functions called to perform the algorithm each with its own script file. One module draws the aircraft in three dimensional space illustrating the stages of the algorithm. The illustrations of the plane in the proceeding chapters were produced with this module (plot_plates_x43).

Even though directions and rotations are more easily dealt with using three parameters when tabulating output that depends on angles in three dimensions it is much more concise to use two parameters instead. Given the direction shown in color blue below, the angle with the x-axis is one parameter and the angle between the x-z plane and a line perpendicular to the x-axis and intersecting the line passing through the origin in the given direction is the second parameter used in this program.

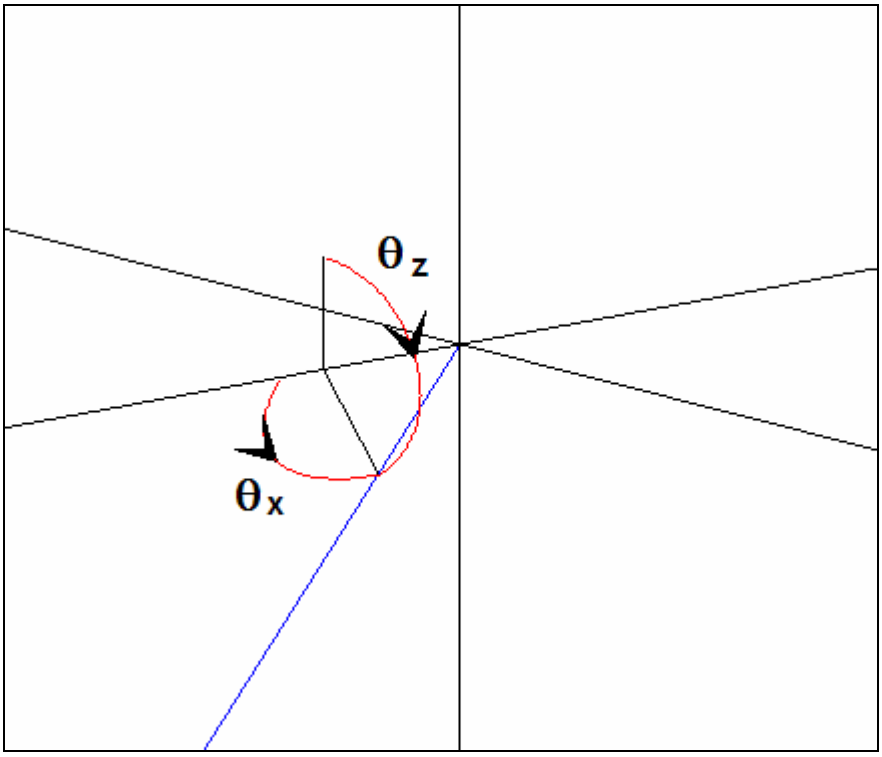

Figure 63: 3D rotation described by 2 perameters instead of 3

A typical cycle of the procedure runs as follows. At this point the coordinates of all fixed plates are known and in a list. The first step is to read control adjustments and rotate all control plates. Then, the list of fixed and the list of control plates are combined.

For a given inclination of the aircraft with respect to the airstream the parameters $\boldsymbol{\theta_x}$ and $\boldsymbol{\theta_z}$ are calculated.

Then, the projection axes $A_{d,}$ A_y and A_z are computed.

Plots may be done in the body plane coordinates and the projected coordinates.

The function call [cp_n_w, cn_w] = plates_normal_1(plate_plane_verts,orientation_plane_plate); or [Fp_w,Ort_w,cp_n_w,cn_w]= aero_normal_1(plate_plane_verts,orientation_plane_plate,Ad); returns a list of the plates restricted to just those that the airstream may impact, as determined by the angle between the airstream and the normal vector.

The call [. . .] = project_plates_x43(plate_plane_verts, . . .); (arguments omitted) returns all of the impacted regions together with associated normal vectors, centroids, areas, and normal components of forces. The function [S_F,I_F] = sum_forces(plates_centroid_impacted_xyz, plates_normals_impacted_xyz, Cn_w, E_area, C_G); returns forces and moments

Most of the program's calculations are done in project_plates_x43, which calls a variety of sub-procedures. For the production of plots the program cycle is executed for a list of input arguments and saved in a file containing a table of values. A script that reads one of these output files and generates plots is the final step.

VI. Concluding Remarks

An arbitrary body analysis program, Hyper-N has been developed using modified Newtonian flow theory. The flow solver has provided data confirming experimental results obtained for hypersonic flow past a double wedge. However, due to the unavailability of experimental data on the X-43A airplane, no comparison could be made to justify the degree of accuracy of the program. To improve the performance of the code, future versions may offer a little more satisfaction by replacing the shadow flow analysis with Prandtl-Meyer expansion and by the inclusion of a viscous model. It is predicted that these adoptions will make the code more applicable to a wider range of configurations. The coupling of an engine model could also enable the investigation of controller design for a three dimensional scramjet. For this task it is necessary to have a fast routine representing the flow processing so that the simulation of the dynamics can be performed.

References

[1] J. D. Anderson. *Hypersonic and High Temperature Gas Dynamics.* McGraw-Hill, 1989.

[2] M. R. Mendenhall, S. C. Lesieutre, S. C. Caruso, F. E. Dillenius, and G. D. Kuhn. Aerodynamic design of Pegasus: Concept flight with computational fluid dynamics. A.I.A.A. Journal of Spacecraft and Rockets, 31(6):1007-1015, 1994.

[3] A. E. Gentry. Hypersonic arbitrary-body aerodynamic computer program (mark iii version). Vol. 1-user,s manual. Rep. DAC 61552, McDonnell-Douglas Corp., April 1968.

[4] P. E. Divan. Aerodynamic analysis system for conceptual and preliminary analysis from subsonic to hypersonic speeds. A.I.A.A. Paper 80-1897, 1980.

[5] M. Moore and J. Williams. Aerodynamic prediction rational for analysis of hypersonic configurations. A.I.A.A. Paper 89-0525, 1989.

[6] C .I. Cruz and A. W. Wilhite. Prediction of high-speed aerodynamic characteristics using the aerodynamic preliminary analysis system. A.I.A.A Paper 89-2173, 1989.

[7] M. Maughmer, L. Ozroski, D. Straussfogel, and L. Long. Validation of engineering methods for predicting hypersonic vehicle control forces and moments. A.I.A.A Journal of Guidance, control, and Dynamics, 16(4):762-769, 1993.

[8] F. R. Chavez and D. K. Schmidt. An integrated analytical Aero-propulsive / Aero-elastic model for the dynamic analysis of hypersonic vehicles. A.I.A.A. Paper 92-4567, 1992.

Appendix

Input Files

Vertices are listed in a file with the simple format

<name> <x y z>

for example

v15 4.25 -1 0

Lines beginning with % are comments in all the files. There are two files containing a list of plates. One of which lists all the plates that remain fixed in position relative to the body coordinate system.

Plates are listed in a file with the simple format

<name> <orientation>

<vertices list>

vertices list::v_1 v_2 ... v_n

orientation is 1 or -1 depending on whether the vertices are listed in counterclockwise or clockwise order, respectively, with respect to the outer normal direction of the plate.

for example

plate1 1

v1 v2 v3

or

plate2 -1

v2 v3 v4 v5

Controls are described as sets of plates that rotate together in the same rigid frame. Control plates are listed in another file.

For example

%plate50 right horizontal stab

plate50 1

v25 v4 v26

%plate51 right horizontal stab

plate51 -1

v25 v4 v26

Another file lists controls in the format

<name> <x y z> <a b c>

<plate list>

Where <x y z> is the initial point of an axis of rotation and <a b c> is the direction of the axis of rotation.

For example

%stabilizer

right_flap 55 -6.105 0 0 -1 0

plate50 plate51

%left flap

left_flap 55.6 6.105 0 0 1 0

plate52 plate53

The last of the input files lists the information needed to calculate moments. I_x, I_y , I_z is the reference position used to calculate moments. The scale factor term adjust from the linear dimensions in the vertices file to the actual dimension of the airplane. The thrust vector is the direction to apply engine thrust. For example,

% mass

3000

% CG

% I_x

40

% I_y

0

% I_z

-1

% principal moment of inertia

% I_xx

1.6667

% I_yy

8.333

% I_zz

-0.333

% products of inertia

% I_xy

0

% I_xz

0

% I_yz

0

%scale factor

0.6

%thrust vector

1

0

0

.

Mass and thrust inputs are used in the simulation of flight, but due to time limitations no results were included in this thesis.

VI. Vita

Younes Baalla was born on June 20th, 1971 to Mohammed Baalla and Fatima Akakas of Casablanca, Morocco. In 2003 He began his undergraduate studies at the University of Tennessee, Knoxville in aerospace engineering. In 2007 he completed his tenure at UTK and began pursuing his graduate education at the University of Tennessee Space Institute. Younes Baalla received his Master of Science in Aviation Systems from the University of Tennessee, Knoxville in August of 2010. His focus was on the aerodynamics of hypersonic flows.

His academic accomplishments included the completion of a computer program that can approximate the performance of hypersonic airplanes. Also, he had two conference publications with Dr. Zei Wi Lin in the field space radiation protection.