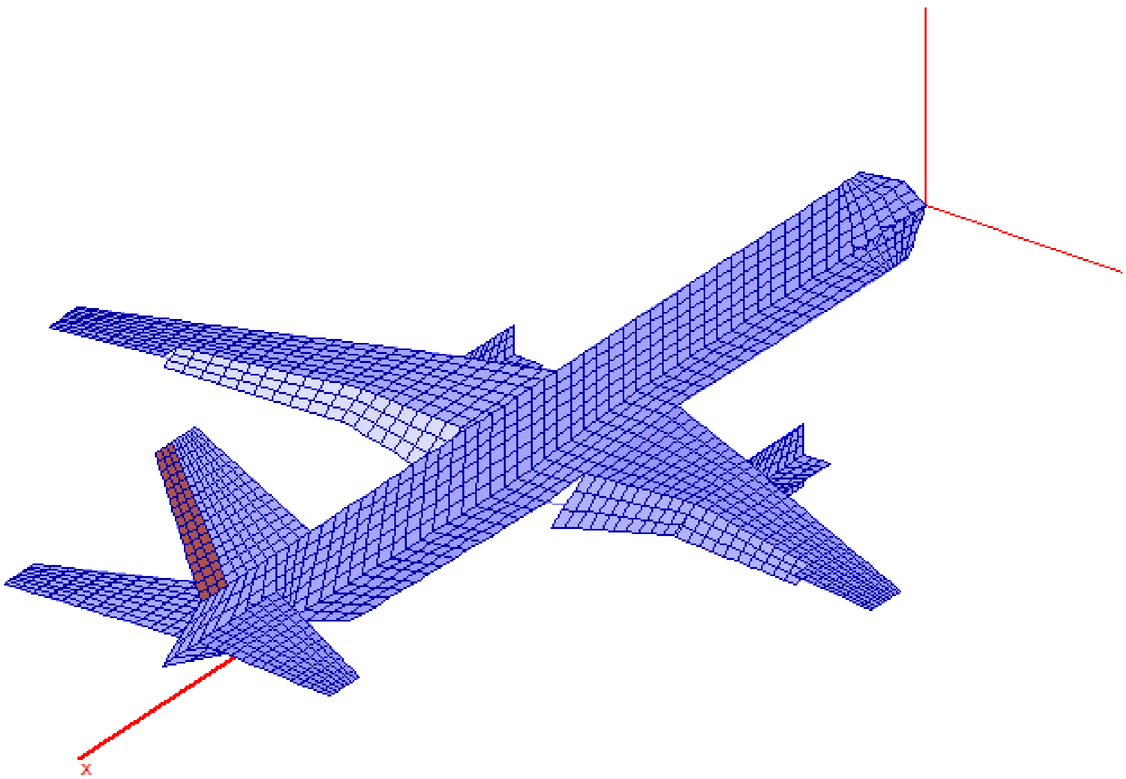
Great OWL Publishing
Engineering Software

SURFACES

VORTEX LATTICE MODULE

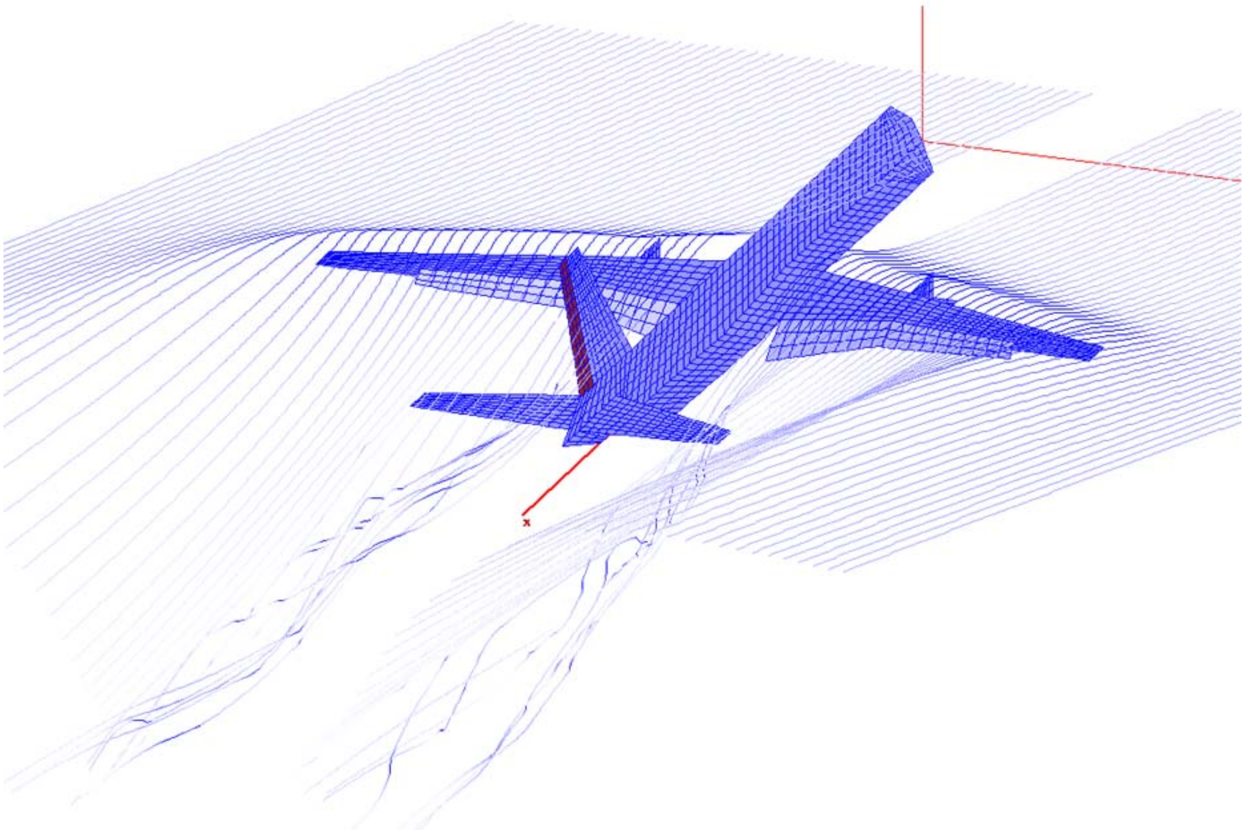


User Manual

August 2009

SURFACES

Vortex-Lattice Module



SURFACES – Vortex-Lattice Module

INTRODUCTION 5
"Vortex Lattice Methods" Why Should You Care? 6
Current Status 7
1. Orientation of Forces and Moments 9
2. Force and Moment Nomenclature 10
3. Project Task List 11
4. Creating a Simple Model with SURFACES 14
5. Accomplishing Special Projects with SURFACES 44
5.1 Tailoring Wings to Improve Stall Characteristics 44
5.2 Determine Shear, Moment, and Torsion 45
5.3 How to Manage Airfoils in SURFACES 45
6. Transformation of Load Vectors from a Global to a Local Coordinate System 52
6.1 Establishment of a Local Coordinate System 52
6.2 Transformation of Force Vector in Coordinate System A-B_N-C 54
6.3 Determination of Moment Vector in Coordinate System A-B_N-C 55
6.4 Determination of Shear and Moment Distribution 57
6.5 Presentation of Data in SURFACES 61
7. Using the Virtual Wind Tunnel 64
8. Determination of a Trimmed Flight Condition 65
9. Determination of Drag in SURFACES 67
9.1 Introduction 67
9.2 Basic Drag Coefficient, C_{Do} 73
9.3 Skin Friction Drag Coefficient, C_{Df} 75
9.4 Induced Drag Coefficient, C_{Di} 84
9.5 Total Drag Coefficient, C_D 85
9.6 Compressibility Modeling 86
9.8 How SURFACES Calculates Do, Df, Di, and D. 87
9.9 Limitations of Drag Estimation Methodologies 87
9.10 Setting up Drag Modeling on Example Aircraft 88
9.11 Summary of SURFACES Drag Analysis Methodology 101
10. Validation Samples 102
Validation 1: 2-D Flat Plate Airfoil 103
V1.1 Model 103
V1.2 Expected Result 103
V1.3 Results from SURFACES 104
Validation 2: 3-D Properties of Two Wings 106
V2.1 Models 106
V2.2 Expected Result 106
V2.3 Results from SURFACES 107
Validation 3: Warren 12 Wing 109
V3.1 Model 109
V3.2 Expected Result 110
V3.3 Results from SURFACES 110
Validation 4 : Bertin-Smith 2-D Wing 111
V4.1 Model 111

V4.2 Expected Result *111*
V4.3 Results from SURFACES *112*
Validation 5: Cessna 172 113
V5.1 Model *113*
V5.2 Expected Result *113*
V5.3 Results from SURFACES *115*
V5.4 Comparison of Codes *117*
Validation 6: 2-D C_L, C_D, C_M for NACA 23012 119
V6.1 Model *119*
V6.2 Expected Result *119*
V6.3 Results from SURFACES *120*
Validation 7: F-104 Starfighter 122
V7.1 Model *122*
V7.2 Results from SURFACES *122*
Validation 8: Ryan Navion 124
V8.1 Model *124*
V8.2 Expected Result *125*
V8.3 Results from SURFACES *127*
Validation 9: Comparison to NACA R-1208 130
V9.1 Introduction *130*
V9.2 Expected Result *131*
V9.3 Results from SURFACES *131*
Validation 10: Comparison to NACA TN-1422 133
V10.1 Introduction *133*
V10.2 Results from SURFACES *133*

INTRODUCTION

Thank you for purchasing **SURFACES**.

We are certain you will find **SURFACES** priceless for your aircraft design projects. **SURFACES** was developed in real aircraft design environment and is loaded with highly developed tools that give you answers quickly. We consider the program analogous to an extremely sophisticated airplane calculator. Create a model of your aircraft and then use **SURFACES** to extract hard-to-get information about it. Stability derivatives, loads, performance parameters are just the beginning of your discoveries. You can extract in a matter of seconds some super complicated parameters that would take a trained aerospace engineer weeks to calculate using classical methods. Use the extra time to study variations of your design to make it even better for its intended mission. Whatever the design task, **SURFACES** will save you weeks if not months of work.

SURFACES is the ultimate tool for anyone designing subsonic aircraft, whether it be a professional aerospace engineer or the designer of homebuilt aircraft. **SURFACES** is not just user friendly, it provides you with very powerful features to help design your aircraft.

SURFACES uses a Three-Dimensional Vortex Lattice Method (VLM) to solve the airflow around an aircraft and extract an incredible amount of information from the solution. Plot the flow solution to better understand how the flow behaves around the airplane.

SURFACES is the perfect solution in any preliminary design environment, or to reverse engineer existing airplanes. It allows you to quickly extract loads and stability and control data.

SURFACES allows you to swiftly model any aircraft. Do you have a three-view drawing of your favorite aircraft? Simply import it in to the environment and scale it up. No pencils, rulers, or calculators are needed for scaling up the model. You do it all from within SURFACES. It’s as easy as clicking a mouse button.

SURFACES determines most stability derivatives and, when used with the built-in Aircraft Datasheet feature, allows you to perform very sophisticated dynamic stability analyses. Import stability derivatives directly from your Vortex-Lattice analyses into an Aircraft Datasheet and plot the aircraft’s Short Period, Phugoid, Spiral Stability, Rolling Convergence, and Dutch Roll modes. You can even simulate the dynamic response of the aircraft in real time!

SURFACES allows you to incorporate all the details of your design, such as airfoil properties, wing twist, dihedral, multiple lifting surfaces, asymmetric geometries, winglets, deflection of control surfaces and high lift devices. **SURFACES** even allows you to account for engine forces as functions of angle-of-attack, airspeed and altitude, whose properties are taken into account when determining trim or stability derivatives.

SURFACES allows you to extract surface pressures, forces and moments, force and moment coefficients, distributed loads, section lift coefficients, and create shear, moment and torsion diagrams on the model.

SURFACES comes with video tutorials. You will be working on your own airplane in 30 minutes or less.

"Vortex Lattice Methods" Why Should You Care?

By Mike Garton

Some of the latest glider designs are advertised as having computer optimized wings. For instance ads for the Saphire, Psyko, Laser, and Edge all list it as a design feature. NSP's ad mentions the "LinAir" program, which uses a form of computational fluid dynamics that we aerospace engineers call "vortex lattice methods" or "simple panel codes". There is not space here to discuss how these codes work (and perhaps not interest either) but I will briefly describe what can be done with these programs and what it means for the pilot. If I lose you in technical jargon, just skip to the last two paragraphs.

A vortex lattice program takes a wing planform, wing-twist, and angle of attack as inputs. Using this information it calculates the induced velocity field surrounding the wing including the effect of tip vortices. It is somewhat non-intuitive, but the angle of attack of a wing is not simply the angle between your root chord and your tailboom. The wing "induces" some vertical components of velocity that change the "effective" angle of attack. Generally the induced angle is smaller at the root of the wing and larger at the wing tips. A tip vortex will add a downward component to the air above the wing tip. This causes the "effective angle of attack" of most wing tips to be reduced. This is one form of aerodynamic wash out. A vortex lattice program allows a designer to quantify these effects, before the plane is built and without the need for a wind tunnel. The use of this tool does not guarantee a good wing. Like any tool, it still takes wisdom and proper application to get good results. This particular tool is usually reserved for graduate degreed aerospace engineers with specialization in computational fluid dynamics.

Some of the things a glider designer can do with this program are to: 1. Minimize induced drag (drag do to tip vortices), 2. Manage which part of the wing will stall first, 3. Given a planform, refine its twist distribution, and 4. Calculate the local flow direction on the stab including downwash from the wing. In general, the refined wings have nearly elliptical chord distributions with finite tip chords (no big surprise here). Aerospace Engineers will assert that elliptical lift distributions DO result in the minimum possible induced drag for low speed wings. At our low Reynold's numbers, a truly elliptical chord distribution does NOT result in an elliptical lift distribution. At low speeds on a truly elliptical winged model, the air flow will separate near the wing tip, leading to too little lift in that region and tip stall. This is why the refined sailplanes tend to have finite tip chords. The nearly elliptical wing has another beneficial quality. The downwash angle is relatively constant along the span. This means the whole wing is flying at the same "effective" angle of attack. A constant angle of attack is good because no part of the wing will stall early and the wing can achieve a high average lift coefficient. When any section of the wing stalls, it will usually propagate sideways and stall the entire wing. As an example, a straight taper wing with its uneven effective angle of attack will stall at an average lift coefficient roughly 20% lower than the computer refined four taper wing. I am assuming that the designer of the four taper wing used the vortex lattice code properly.

So what might a pilot notice in flight when flying one of these planes refined with a vortex lattice code? Most pilots won't notice the differences. After trimming the plane, an expert pilot should notice that the launch is steeper because the wing can pull a higher lift coefficient before stalling. The sink rate and glide ratio should be a tweak better as well. We are only talking a couple percent decrease in drag over the "eye balled" planforms, but every little bit helps. The plane should be able to fly slower than other planes with the same airfoil and wing loading, again because of the higher available lift coefficient.

Will the computer refined planes always win? In general, no. In most weather conditions a thermal duration contest is still 90% pilot 10% airplane. The contest placings usually sort the pilots by skill regardless of what they are flying. If anyone wants to play with a vortex lattice program, contact me and I can email you directions on how to obtain a public domain program.

Reprinted from: http://eiss.cnde.iastate.edu/articles/VortexLattice.htm

NOTE: This article available online from the above link and is therefore assumed public (in the public domain). It was not written with SURFACES specifically in mind, but is reprinted here as the editor of this manual considered it well written and pertinent to anyone using CFD methods. Great OWL Publishing reprints it here for your convenience, but assumes no responsibilities for it.

Current Status

Currently, the latest version of **SURFACES** is 2.8.10.

The following changes have been made to the program since Version 2.86 (or 2.8.6):

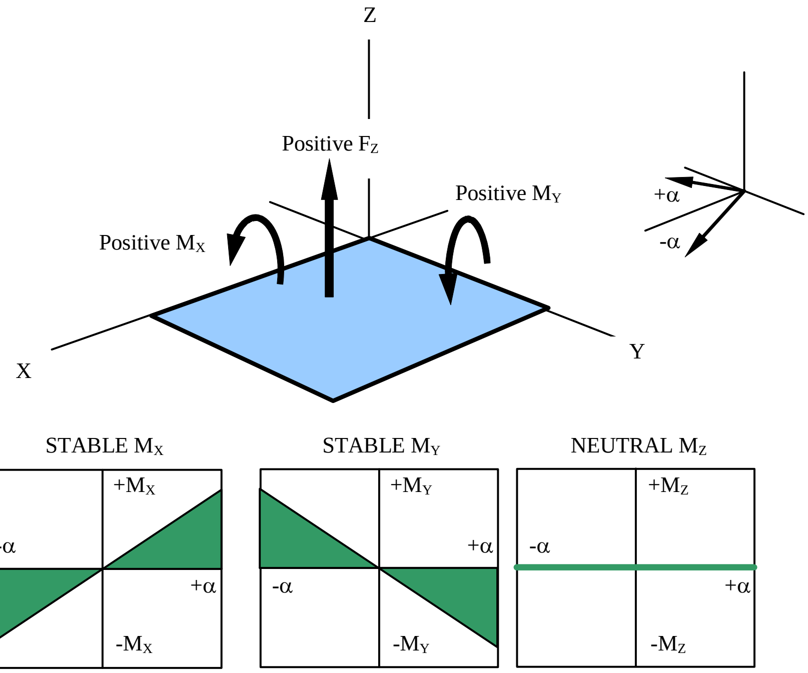
REPAIR LOG

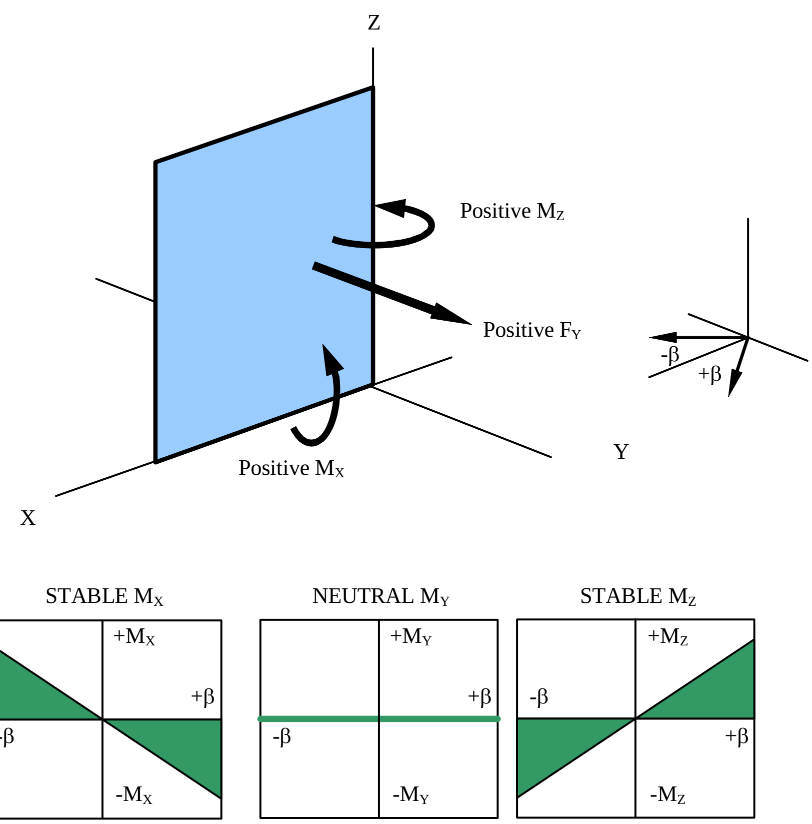
ID	Date	Version	Description	Status
1	6/29/09	2.8.7	Selected surfaces deselected when VLM console icon on MDIForm clicked.	Fixed
2	6/29/09	2.8.7	Pitch/Yaw coupled surfaces (e.g. V-tail) reset elevator deflection in the VWT. Subroutine **DOC_Surface_ModifyDeflection** not originally designed to handle coupled surfaces. Revised it to handle such surfaces correctly.	Fixed
3	6/29/09	2.8.7	Controllers tab on VLM console: Pressing the Reset button would not change numbers in the textboxes. This has been changed.	Fixed
4	6/29/09	2.8.7	Controllers tab on VLM console: Subroutine **DOC_Surface_ModifyDeflection** is used when the user presses the Set buttons. The modification in ID2 now allows the user to enter a elevator+rudder deflection for V-tails	Fixed
5	6/29/09	2.8.7	New functions added: [SDfwd(i)] and [SDaft(i)], which retrieve forward and aft deflection angles of the selected surface i.	Added
6	7/2/09	2.8.8	Bug in subroutine **VLM_PlotStreamlines** which would cause a crash if number of streamlines was 1.	Fixed
7	7/2/09	2.8.8	Improved user information for usage of control deflections in form FormVLM17 (stab ders).	Added
8	7/3/09	2.8.8	Overflow message generated when zoom in too far	Fixed
9	7/3/09	2.8.8	Recent projects list added	Added
10	7/4/09	2.8.8	Data Analyzer multi-variable regression states the following in the text output “Analysis assumes X is in *Col. 1*” and it should say “*last column*” to match equation template.	Fixed
11	7/4/09	2.8.8	VLM Solution Seeker tool repaired and made visible to user.	Fixed
12	7/5/09	2.8.8	Math object list is now synchronized with the list that appears when the user presses the “Press to Select Objects for Legend…” button.	Fixed
13	7/5/09	2.8.8	Rotate about vector operations use a left-hand coordinate system (should be right-hand)	Fixed
14	7/5/09	2.8.8	Math object list does not recalculate upon opening file	Fixed
15	7/13/09	2.8.9	Pressing Browse… in VWT form and navigating the directory form could crash the program if the selected drive was inop.	Fixed
16	7/20/09	2.8.9	Drag calculations have been completely scrubbed. Now the user can associated skin friction drag with both surfaces and vectors (airfoils). Usage of drag has been improved, simplified and made far more user friendly, but yet more powerful. Function [CDf], [CDi], [CD], and [CL] were added to allow user to directly extract drag and lift coefficients from the model and VL solution. User can specify CDf directly for surfaces or specify transition location on airfoils for mixed laminar-turbulent boundary layers. Four new features have been added to the VLM Console. These help the user to view the extent of the prescribed laminar flow on surfaces and the magnitude of skin friction drag on each surface.	Added
17	7/20/09	2.8.9	A large section on Drag Analysis has been added to VLM.PDF. This is Section 9.	Added

18	7/27/09	2.8.10	Panel orientation has been made independent of orientation as the program will now reassign panel corner IDs based on a special algorithm. This means that the user can use Curves A1 and A2 for surfaces that are no longer parallel to the X-axis. The panels still have to be aligned to the X-axis, as this is a requirement of the VL method. However, the user can model circular shapes like an engine nacelle or round fuselage more easily.	Added
19	7/31/09	2.8.10	User can press F2 to copy viewport info (such as state of zoom) and paste into another viewport using F3.	Added
20	7/31/09	2.8.10	User can investigate panel orientation in addition to surface A1/B1 curve orientation (by pressing Ctrl+T).	Added
21	7/31/09	2.8.10	A bug that allowed any number of categories in the Project Properties form was fixed.	Fixed
22	7/31/09	2.8.10	Function [Swet(surf1, surf2, …)] added to extract wetted area.	Added
23	7/31/09	2.8.10	Expanded geometry recognition when user selects a math object referring to the geometry,	Fixed
24	8/15/09	2.8.10	User can turn AutoCalc on or off by double-clicking a panel on the status bar. This is handy for slower computers, as it will prevent math objects from being solved after each change, which is what happens when AutoCalc is on. It is intended to allow the user to temporarily turn the feature off, but user must know that while off, the math objects will not update correctly.	Added

Great OWL Publishing
Engineering Software

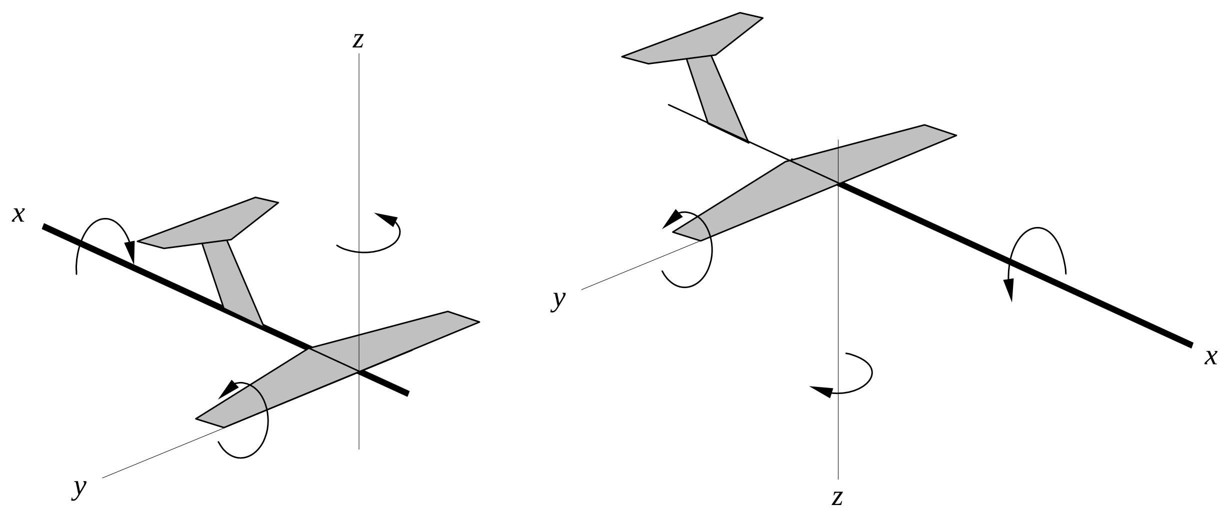
1. Orientation of Forces and Moments





2. Force and Moment Nomenclature

Name	SURFACES Symbol	Other names
Axial force (along X-axis)	FX	X
Side force (along Y-axis)	FY	Y
Normal force (along Z-axis)	FZ	Z
Rolling moment (about X-axis)	MX	L
Pitching moment (about Y-axis)	MY	M
Yawing moment (about Z-axis)	MZ	N
Coefficient of axial force (along X-axis)	Cx	C_x
Coefficient of side force (along Y-axis)	Cy	C_y
Coefficient of normal force (along Z-axis)	Cz	C_z
Coefficient of rolling moment (about X-axis)	Cl	C_l
Coefficient of pitching moment (about Y-axis)	Cm	C_m
Coefficient of yawing moment (about Z-axis)	Cn	C_n



Standard right-handed Aerodynamic Coordinate System (ACS).

Typical right-handed Stability Coordinate System (SCS).

Note 1: Positive rotation about an axis is always in the direction of the thumb of the right hand, as can be seen in the above figure.

Note 2: **SURFACES** uses a standard right handed Aerodynamic Coordinate System (ACS), which is conventionally used for other aspects of aircraft aerodynamic analyses. In this coordinate system, the sign of the lift is positive, when pointing upwards (i.e. towards positive Z), and the sign of the drag is positive, when pointing backwards (i.e. towards positive X). The user must be cognizant of the orientation of the axes when interpreting results.

Note 3: **SURFACES** comes with a routine that will convert stability derivatives to a standard body axes Stability Coordinate System (SCS). This is typically the default for stability and control related tools.

3. Project Task List

A typical project in **SURFACES** is conducted per the following list:

Model Creation

Task	Description	Remark
1	Define Points	Drop points as required to represent the extremes of the aircraft.
2	Define Vectors	Draw vectors, parametric curves, or Bezier curves as needed, using the points. Use parametric or Bezier curves to represent cambered airfoils.
3	Create Surfaces	Define surfaces by selecting the opposite curves A1 and A2, and B1 and B2. Only use curves A1 and A2 for curved surfaces.

Model Preparation

Task	Description	Remark
4	Determine the Trapezoidal Mean Aerodynamic Chord	Select **Tools->Trapezoidal Mean Aerodynamic Chord…** from the VLM Console. This tool will determine several important geometric reference parameters to use with your model, including the MAC, its location, the wing area, and wing span. It also allows you to specify the CG location in terms of %MAC. You must use the *Transfer* tab on the form to transfer the calculated values to your model. While not necessary, it's recommended you copy the analysis report and paste as a Remark with your model. Do this by selecting **Edit->Remark…** from the Surfaces Worksheet window.
5	Determine the Horizontal and Vertical Tail Volumes	Select **Tools->Horizontal/Vertical Tail Volume…** from the VLM Console. Although not necessary for analysis, it is a good idea to tail volume and compare to other airplanes. Copy and paste the analysis report into the remark.

6	Other model properties	Select **Edit->Model Properties…** from the VLM Console. Try to fill in as many properties as possible. All entries marked with an asterisk (*) are required for any Vortex-Lattice analyses.

Once your model runs, you can initiate a large number of specific investigations.

Basic Investigations

Task	Description	Remark
7	Determine Neutral Point	Select **Tools->Determine Neutral Point…** from the VLM Console. This is a necessary step as it will determine your aft CG limit. Always consult the CG location of your design with a qualified Aerospace Engineer. The CG is typically at least 8-10% MAC forward of the neutral point.
8	Trim Analysis	Select **Tools->Determine Neutral Point…** from the VLM Console. This tool is helpful to determine required surface deflections for given weights, airspeed, and yaw angles. Note that before you can use this tool, you must define control surfaces using edge deflections and proper references under the Edit Surface dialogbox (*Edge Deflections* and *Reference* tabs).
9	Panel Results	Select the *Panel Results* tab on the VLM Console. Here you can extract various information pertaining to panels, such as areas, normals, vortex strengths, velocity over a panel, force generated by a panel, pressure coefficients, panel lift coefficients, as well as the center of pressure.
10	Body Results	Select the *Body Results* tab on the VLM Console. Here you can extract information about forces and moments acting on your model.

11	Stip Results	Select the *Panel Results* tab on the VLM Console. Here you can extract a number of information about strips of panels (chordwise), for instance, forces, moments and coefficients. Display strip CL (section lift coefficients) to help you design for delayed tip stall.

You can conduct even more sophisticated analysis per the following task list.

Advanced Investigations

Task	Description	Remark
12	Determine Stability Derivatives	Select **Tools->Determine Stability Derivatives…** from the VLM Console.
13	Determine Control Response	Select **Tools->Determine Control Response…** from the VLM Console.
14	Determine Loads	Select **Results->Force Integrator…** from the VLM Console.
15	Determine Specific Conditions	Select **Tools->Goal Seek…** from the VLM Console. With this tool you can calculate AOA, AOY, or Vinf required to generate a specific load, lift, or even lift coefficient. Note the result don't necessarily result in an aerodynamically balanced model (i.e. MX, MY, or MZ will be non-zero).
16	Modify Geometry to Satisfy Specific Conditions	Select **Tools->Geometric Goal Seek…** from the VLM Console. This tool can be used to move points so that specific conditions are satisfied. The best example of its use is to move the leading points on a stabilator in the Z-directions at a specific flight condition so the MY is zero. In other word, determine an ideal angle of incidence of a stabilator.
17	Virtual Wind Tunnel	Select **Virtual WT->Setup and Execute WT Run…** from the VLM Console.

4. Creating a Simple Model with SURFACES

The following model is designed to allow the novice user to quickly become familiar with **SURFACES**. Pay close attention to which options and checks are made in each form below before proceeding to the next step.

STEP 1: Start a new project by selecting **File->New Project…**

This will open a small form on which you need to specify the type of project to create. Press the button labeled 'Surfaces Worksheet' to open a blank worksheet. Maximize the window for added convenience. The move on to create surfaces to represent the wing.

STEP 2: Select **Insert->Trapezoidal Surface…**

STEP 3: Create the WING using the numbers in the dialog in Figure 4-1a through 4-1d.

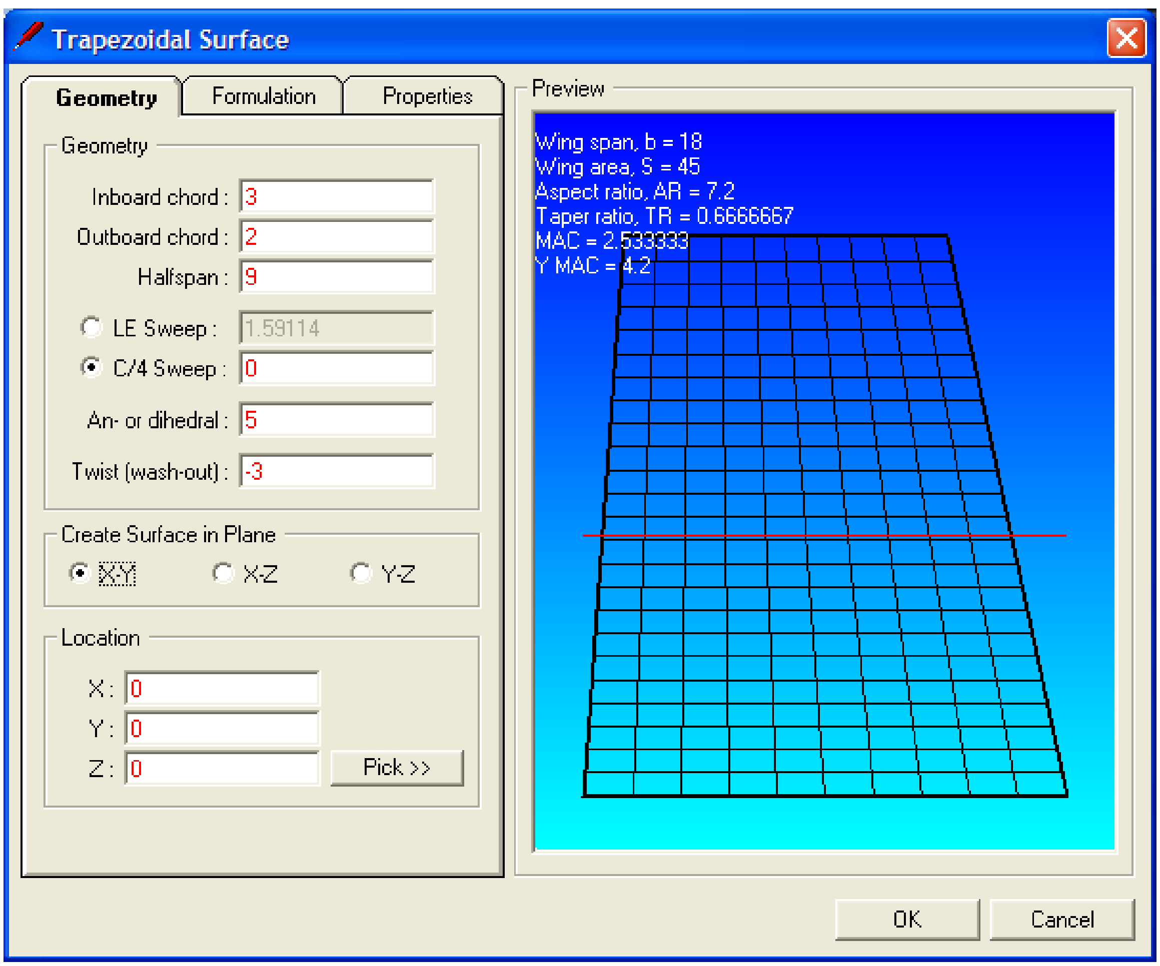


Figure 4-1a: Creating the wing – Entering geometry (Step 3).

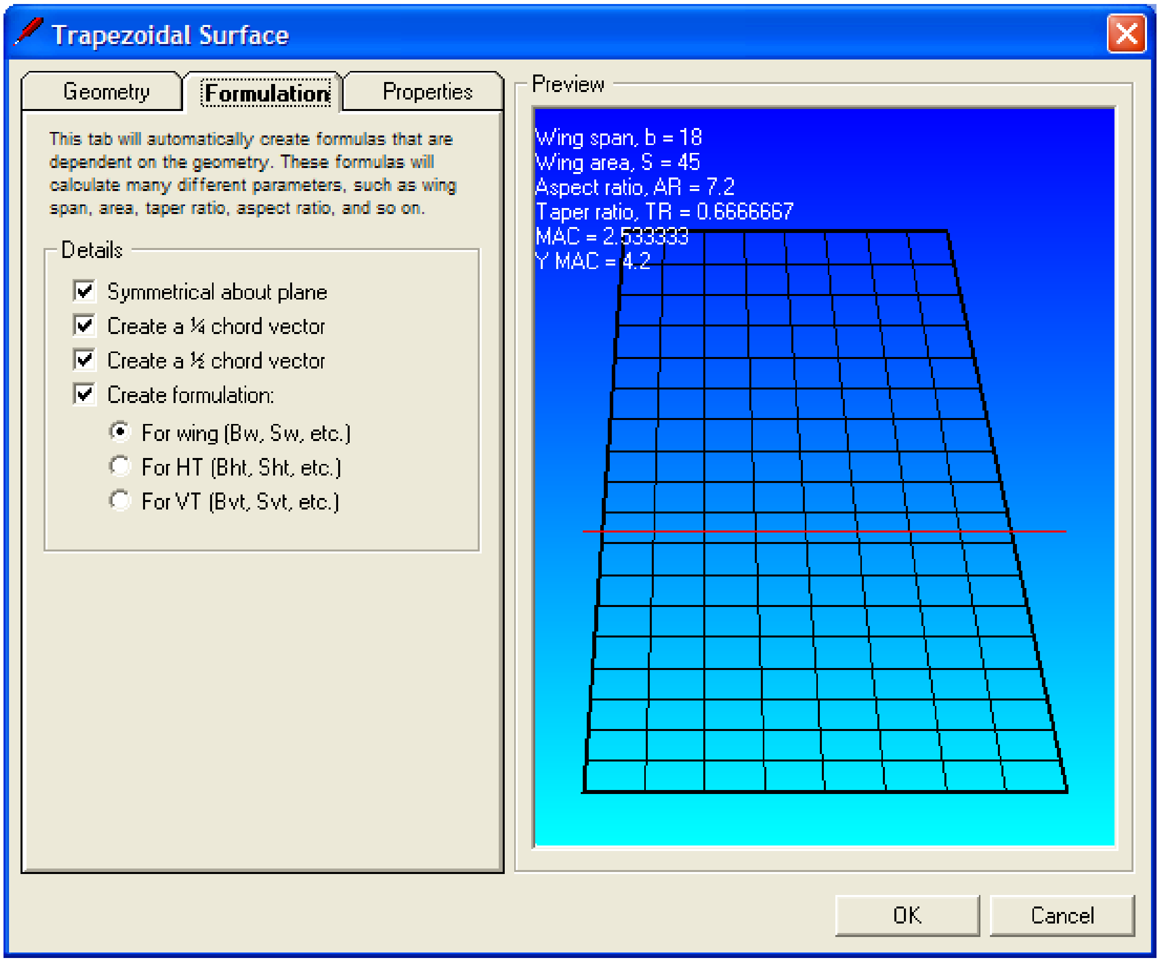


Figure 4-1b: Creating the wing – This tab will help you create geometrically dependent formulas. Note the selected checkboxes and options (Step 3).

The purpose of the options in Figure 4-1b is to automatically create formulation that calculates wing span, aspect ratio, wing area, taper ratio, and other for your convenience. There are other ways to create such formulas, but you will learn these at later time.

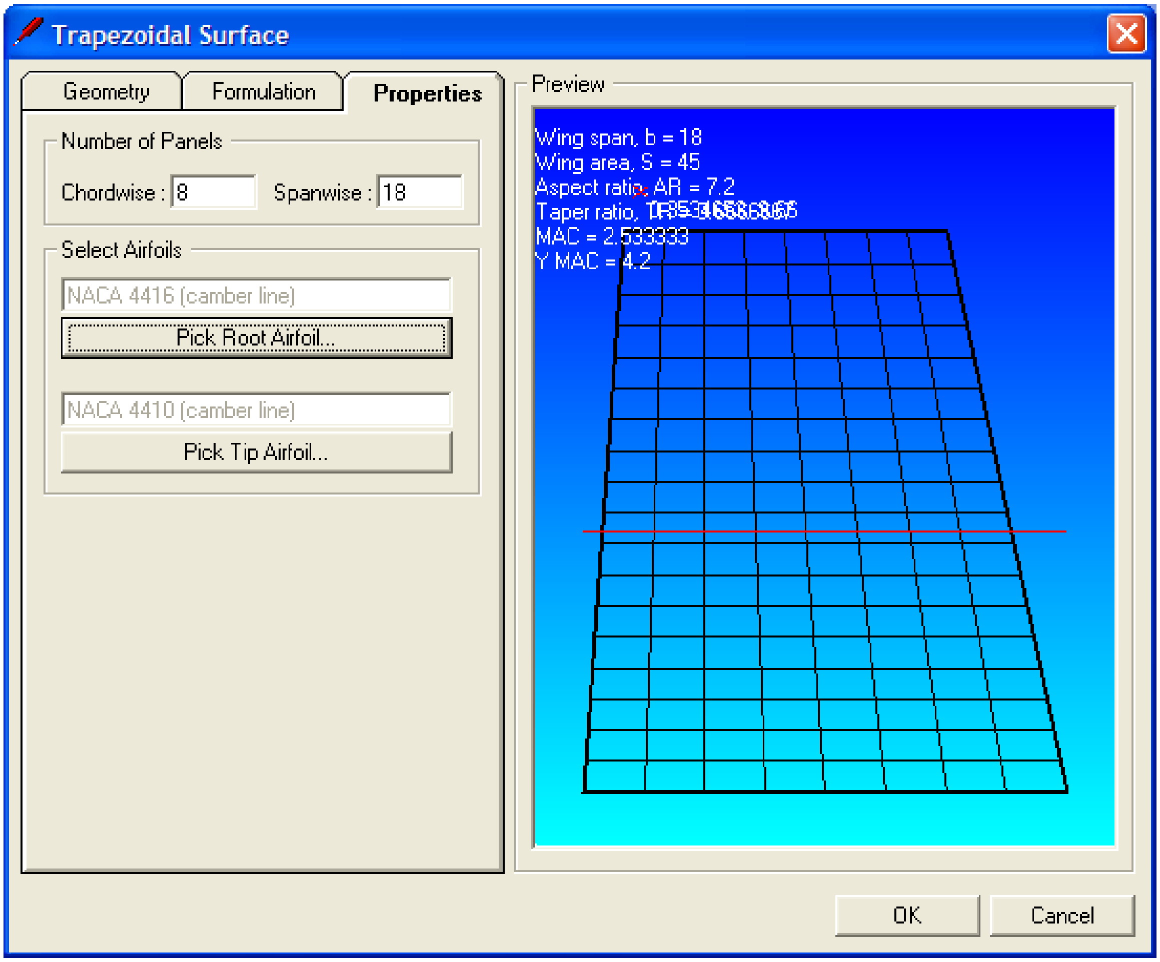


Figure 4-1c: Creating the wing – Setting panel density and picking airfoils for root and tip. Note that pressing the [Pick Root Airfoil…] or [Pick Tip Airfoil…] buttons will open the Camber Creator form in Figure 4-1d (Step 3).

You must press each of the buttons in Figure 4-1c to create your airfoils. If an airfoil is not recognized, a flat plat is assumed. You can also create your own airfoils, but these are stored as text files that are called shape files. They have the extension .SHP. You can navigate to the /Surfaces/Shape Files folder and double-click on one such file to open it in Windows Notepad and investigate how simple they are.

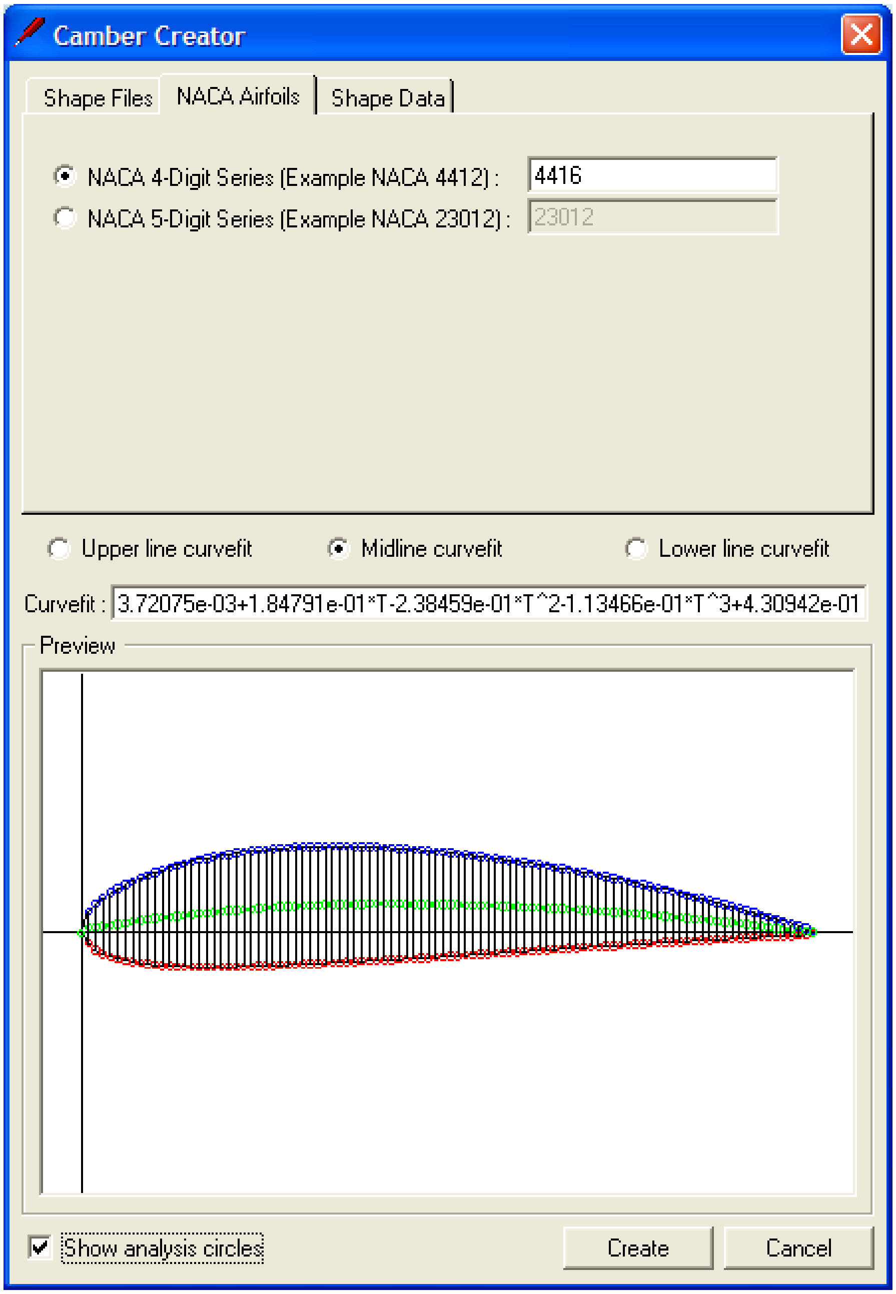


Figure 4-1d: Creating the wing – Picking airfoil. Here select NACA 4416 for the root airfoil and NACA 4410 for the tip (Step 3).

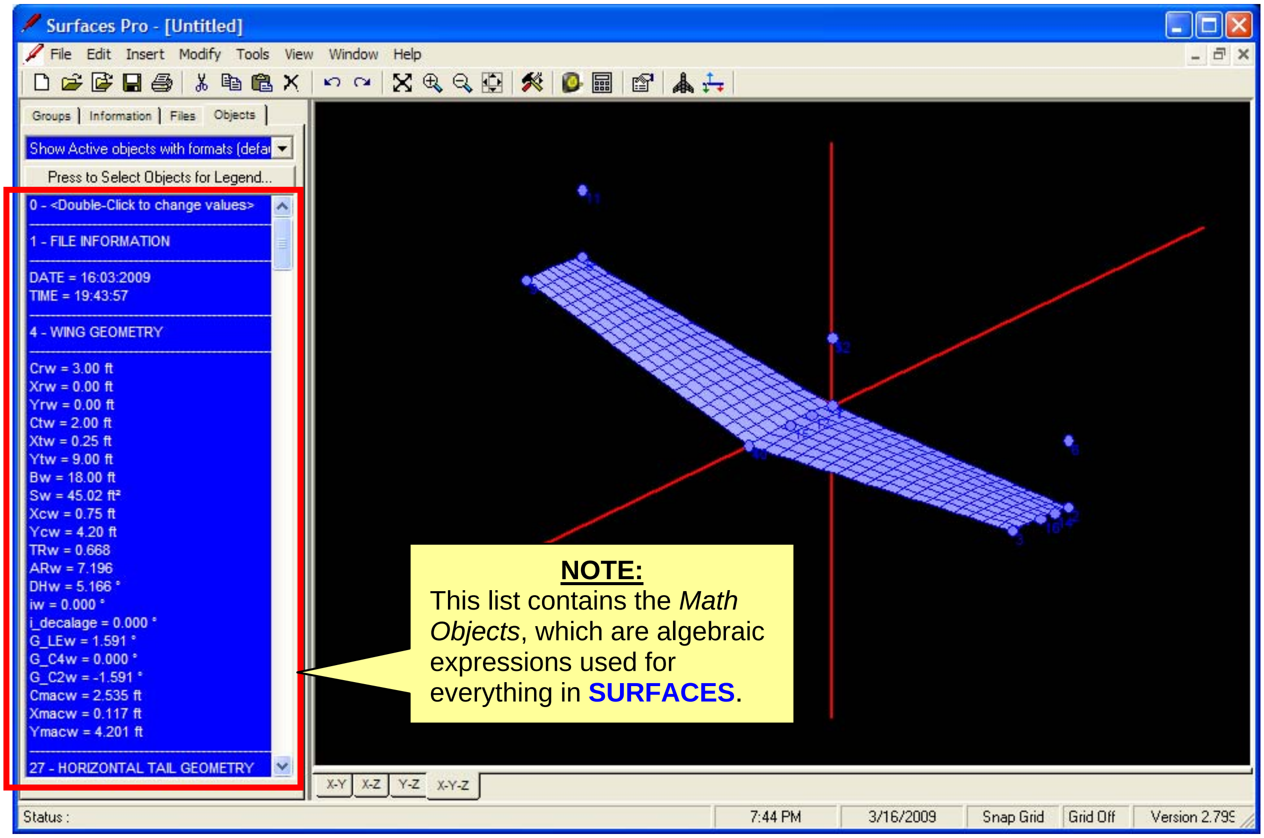


Figure 4-2: If you followed Steps 1 through 3 correctly, the wing will appear as shown, containing the selected airfoils, twist, and dihedral (Step 3).

STEP 4: Select **Insert->Trapezoidal Surface…** to create the HORIZONTAL TAIL (HT). Fill in the form using the numbers in the dialog in Figures 3a through 3c.

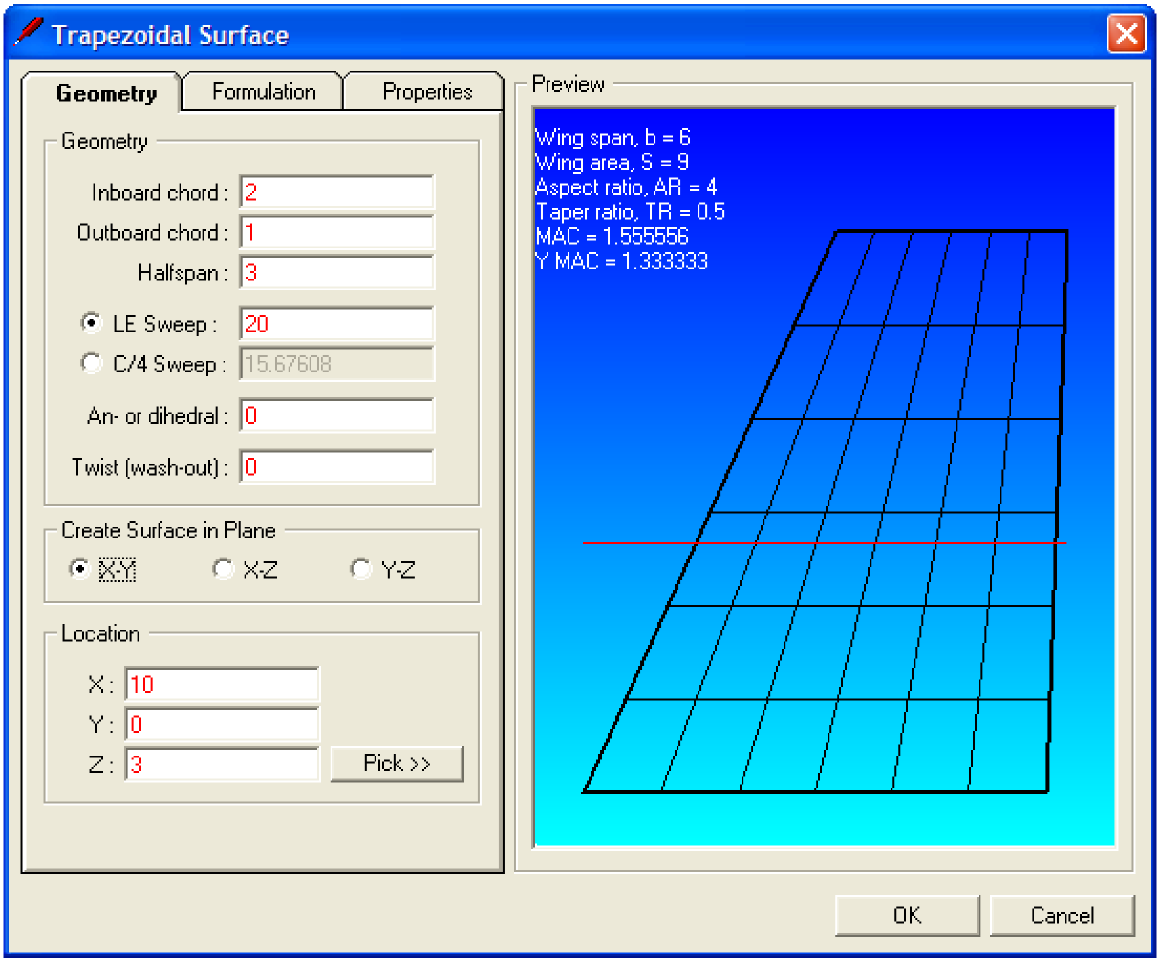


Figure 4-3a: Creating the HT – Entering geometry (Step 4).

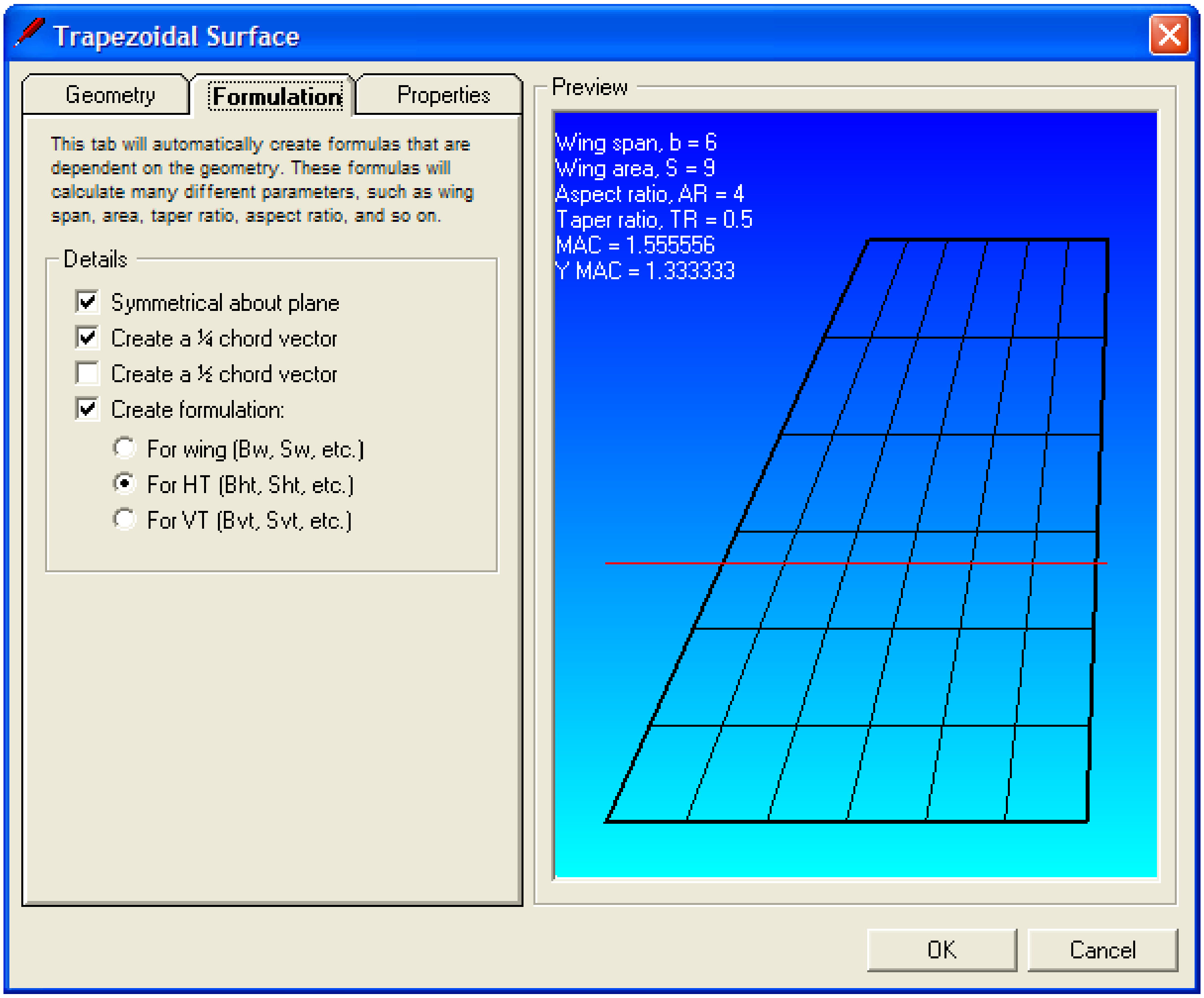


Figure 4-3b: Creating the HT – This tab will help you create geometrically dependent formulas. Note the selected checkboxes and options (Step 4).

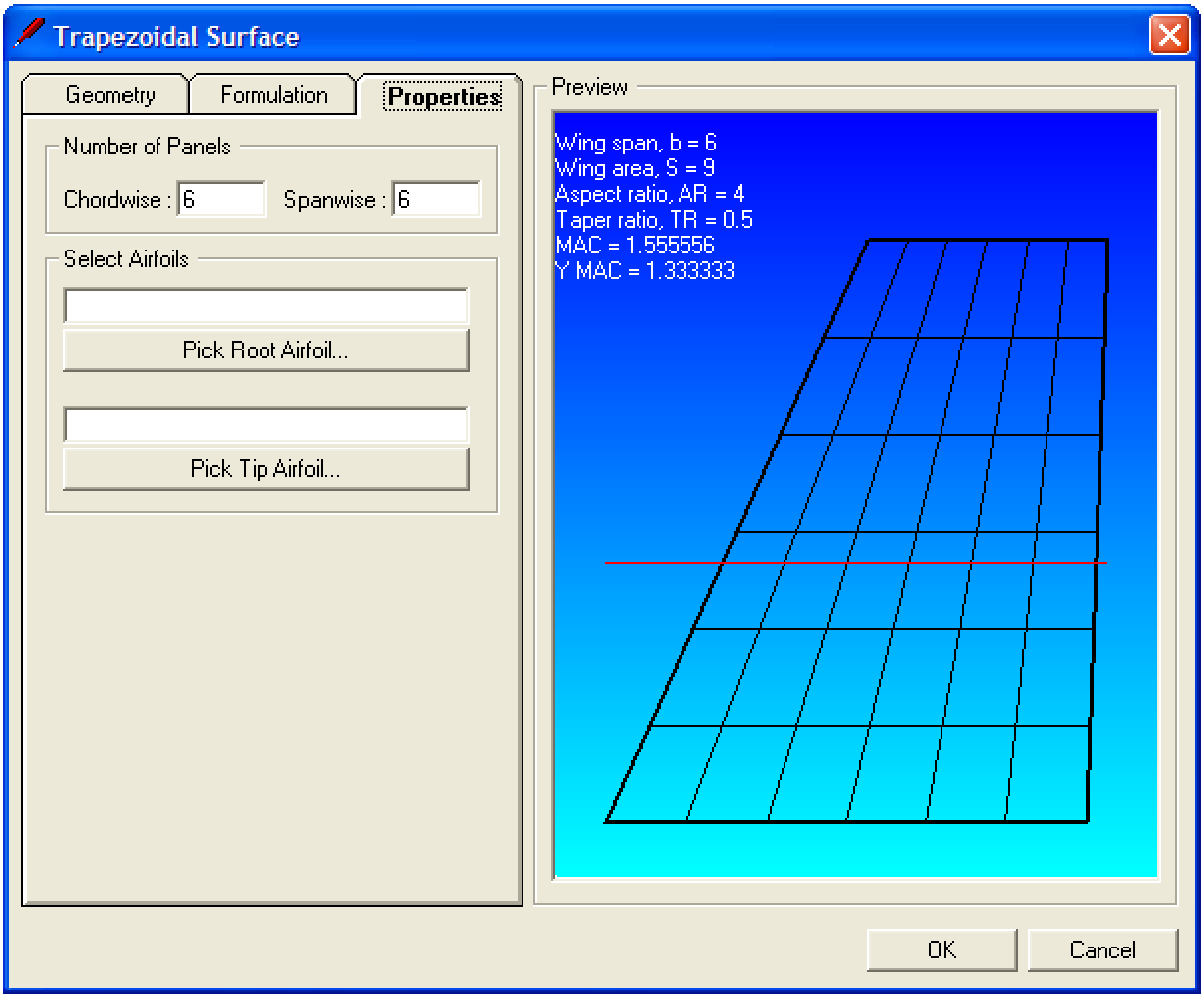


Figure 4-3c: Creating the HT – Setting panel density. Note that no airfoils are picked here, so the resulting airfoil is a flat plate (symmetrical airfoil) (Step 4).

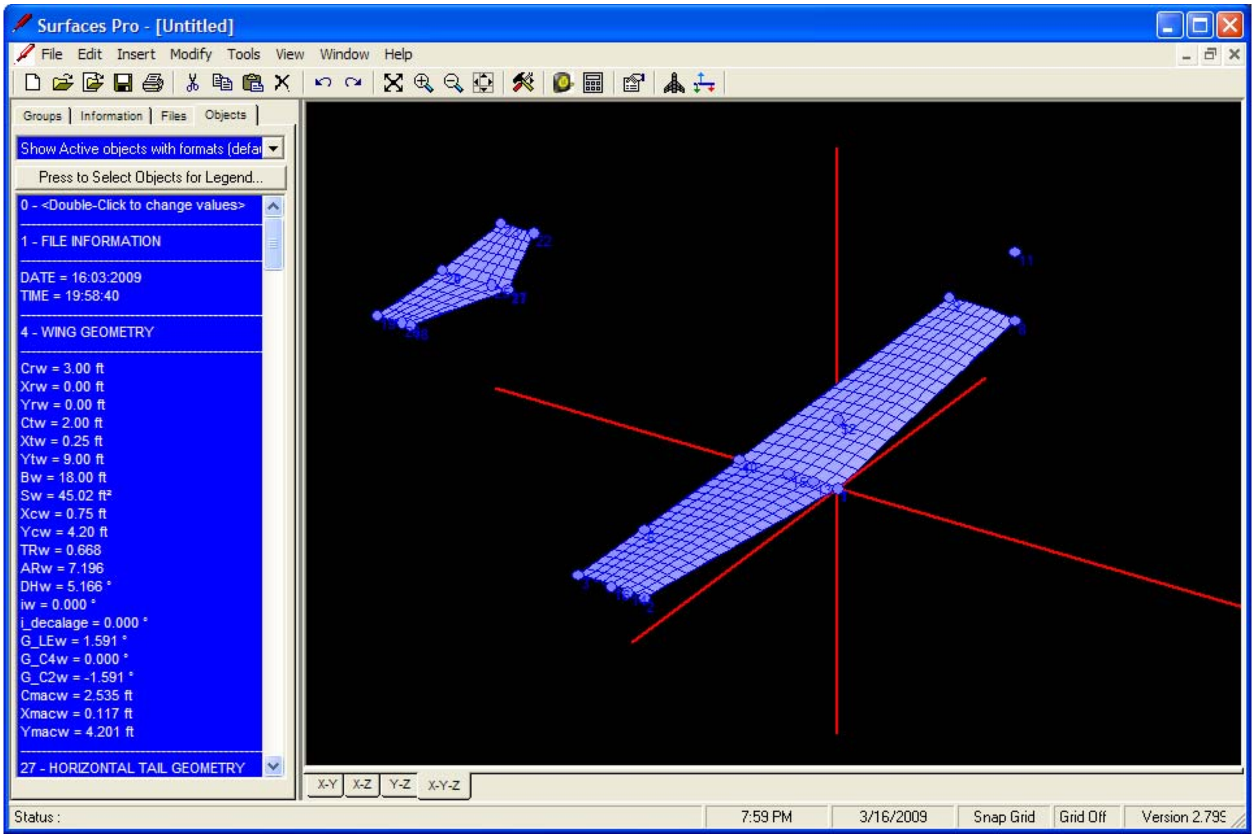


Figure 4-4: If you followed Step 4 correctly, the wing and HT will appear as shown (Step 4).

STEP 5: Select **Insert->Trapezoidal Surface…** one more time and create the VERTICAL TAIL (VT) by filling the form using the numbers in the dialog in Figures 5a through 5c.

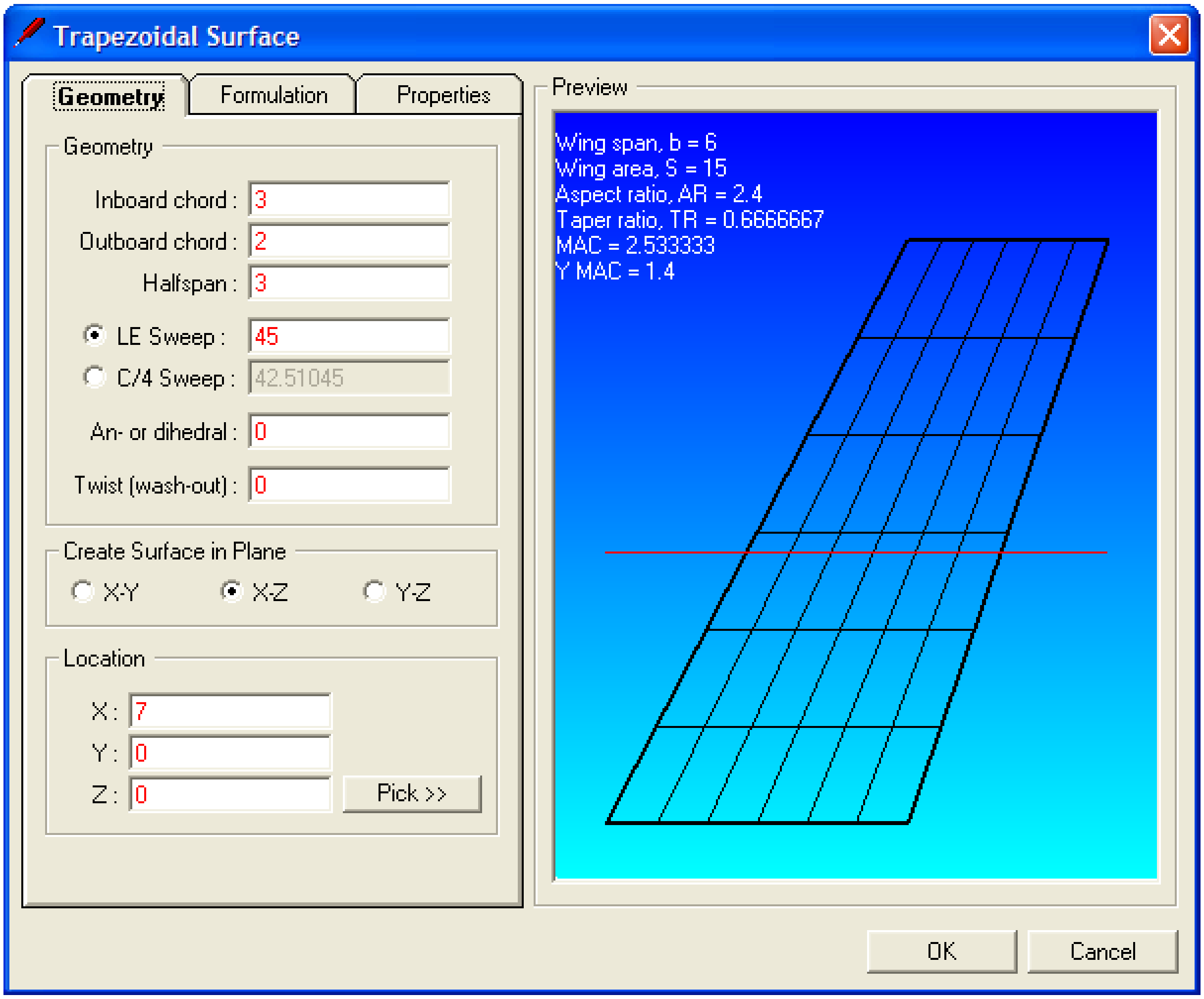


Figure 4-5a: Creating the VT – Entering geometry. Note the option selected in the "Create Surface in Plane" frame is now the X-Z plane, rather than the X-Y plane used for the wing and HT (Step 5).

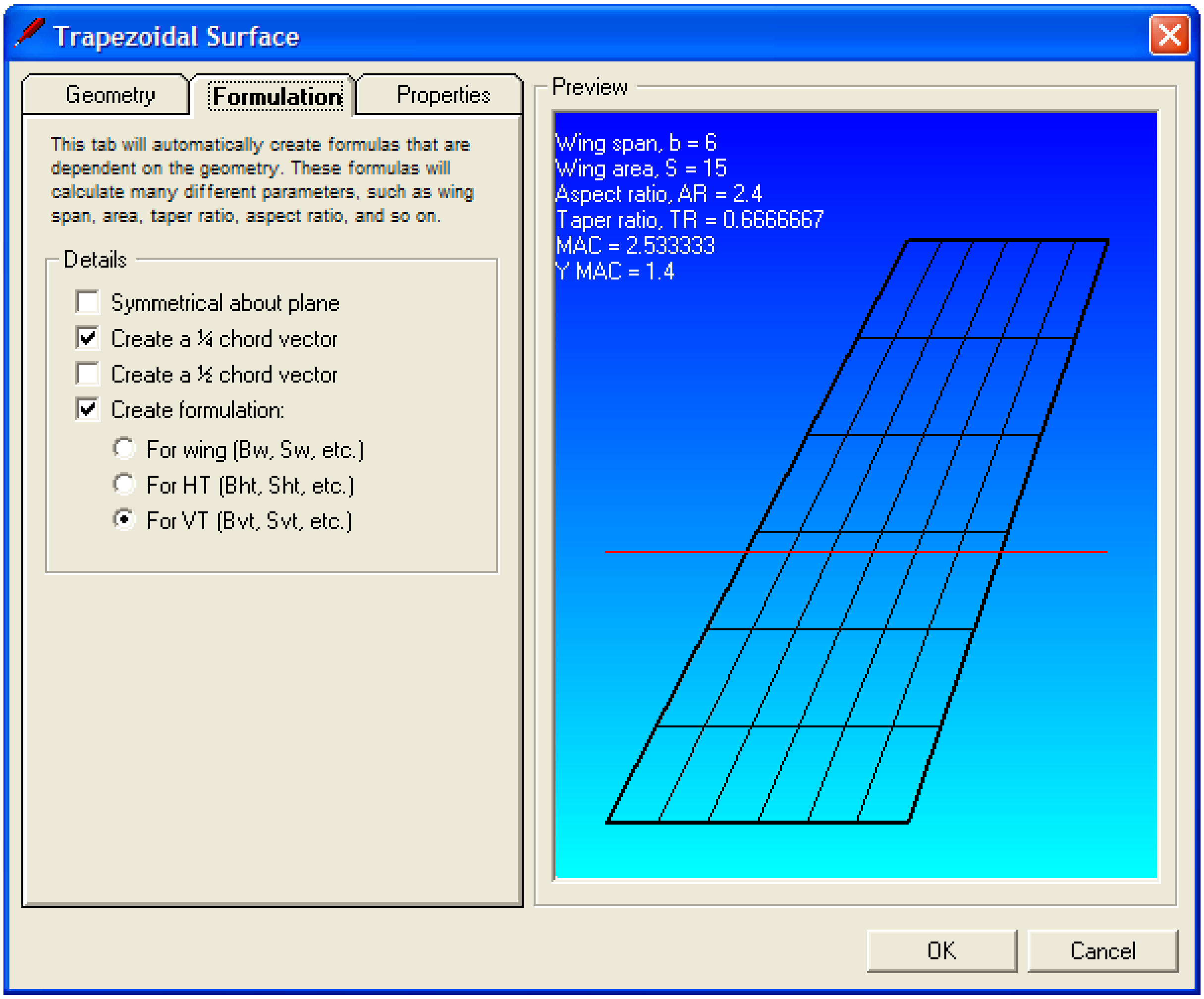


Figure 4-5b: Creating the VT – This tab will help you create geometrically dependent formulas. Note the selected checkboxes and options (Step 5).

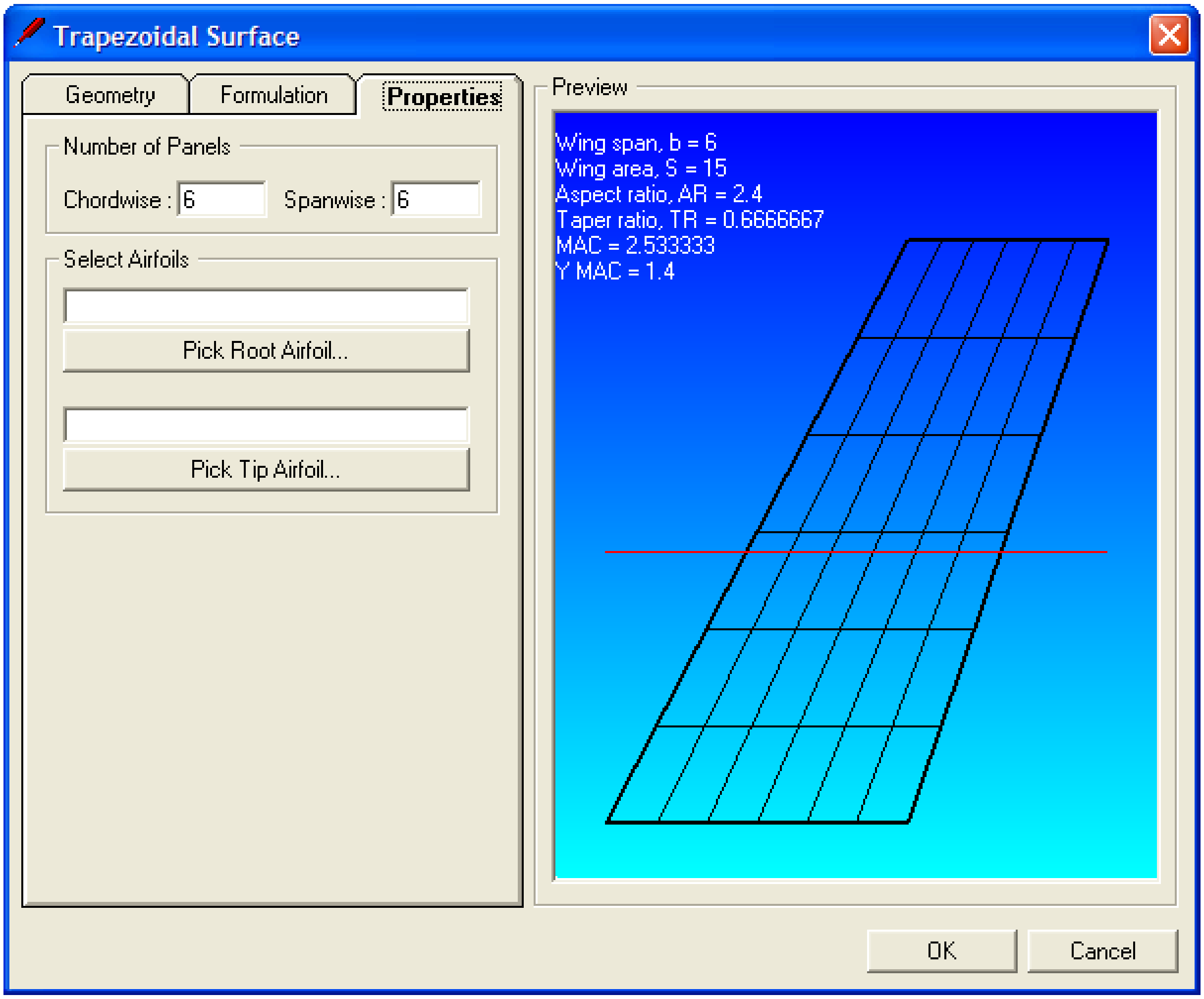


Figure 4-5c: Creating the VT – Setting panel density. Note that no airfoils are picked here, so the resulting airfoil is a flat plate (symmetrical airfoil) (Step 5).

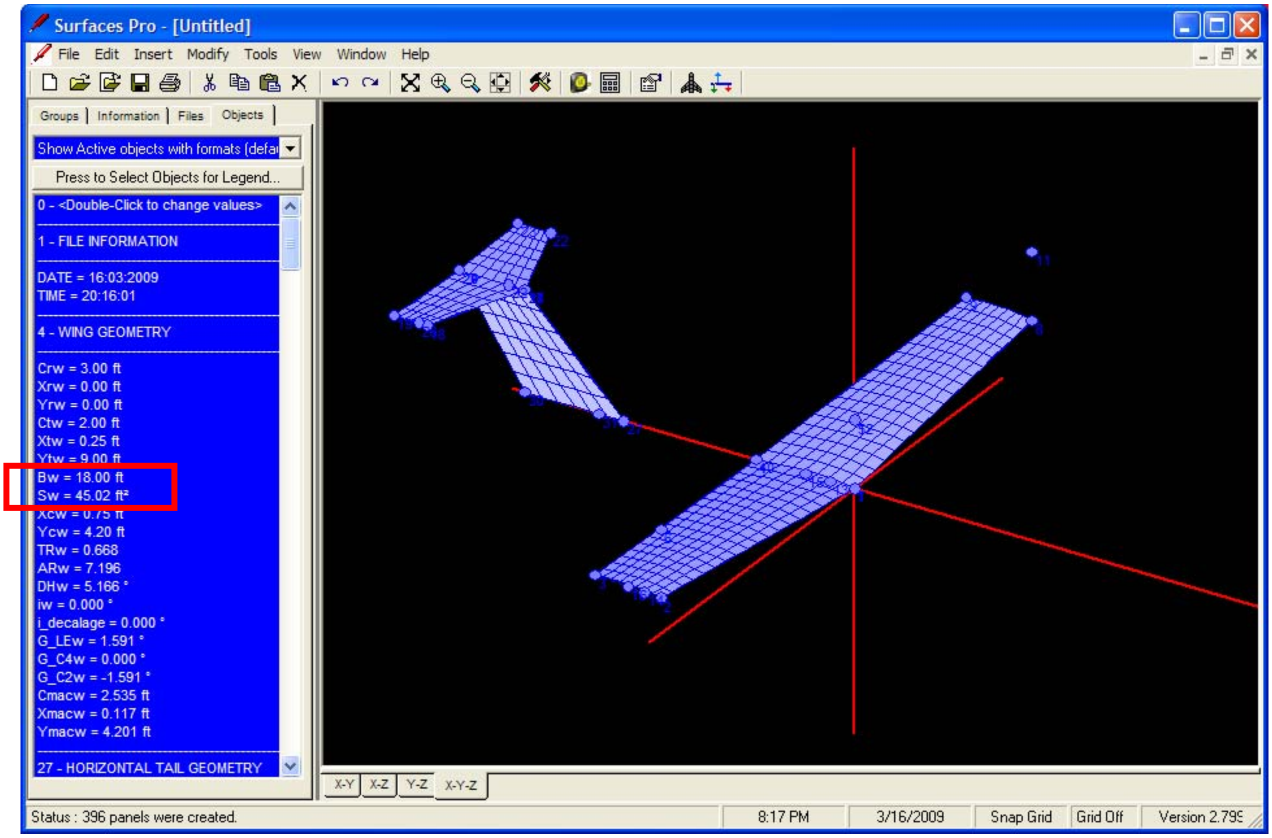


Figure 4-6: If you followed Step 5 correctly, the wing, HT, and VT will appear as shown in the completed basic model (Step 5).

When complete your model should look like the one in Figure 4-6; a T-tail design with a straight tapered wing. You should be aware of that you can also create the surfaces directly by dropping points, stretching vectors, and inserting surfaces. However, in the interest of time and simplicity, the user can create trapezoidal surfaces more easily using this tool.

Note that you can hide points, vectors, and surfaces. While this is not necessary, it may clean up the view. Here let's hide the points. Do this by clicking somewhere on the black background. This ensures the workspace (image) has the focus. Then, simultaneously press Shift and P (for Points). This selects all the points. Then simultaneously press Ctrl and H (for Hide). The resulting image appears in Figure 4-7.

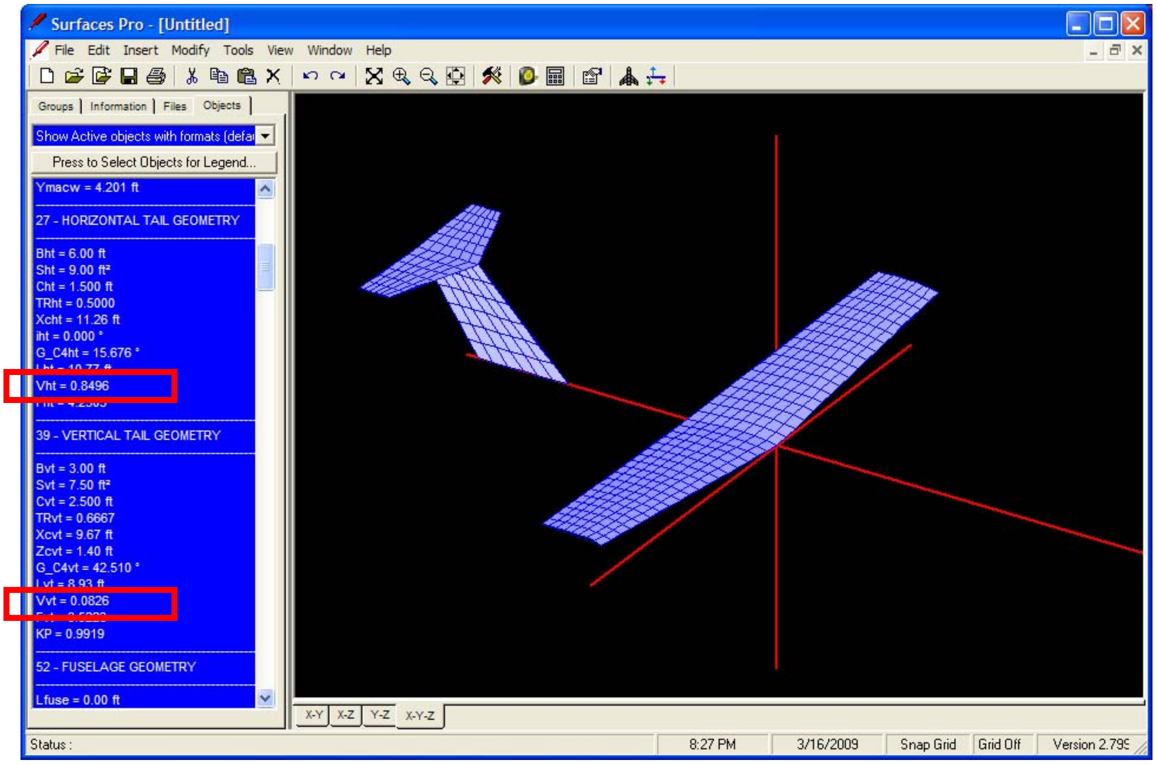


Figure 4-7: The basic model after the points have been hidden.

As you can see identified by the red box in Figure 4-6, the wing span (Bw) is 18 ft and wing area (Sw) is 45 ft². Similarly, you can see identified by red boxes in Figure 4-7 the horizontal and vertical tail volumes should be 0.8496 and 0.0826, respectively. Now let's add weight to the model using the specialized tools in **SURFACES**.

STEP 6: Select **Edit->Select Surfaces...** Press the [Select All] button and then the [OK] button (see Figure 4-8).

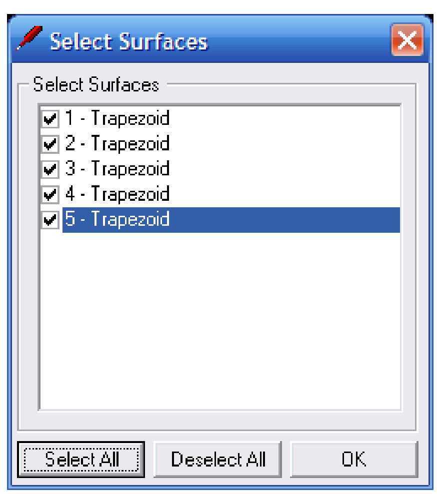


Figure 4-8: Selecting all surfaces simultaneously (Step 6).

STEP 7: Select **Tools->Distribute Weight on Selected Surfaces and Nodes...** Enter 400 in the entry box and press the [OK] button (see Figure 4-9).

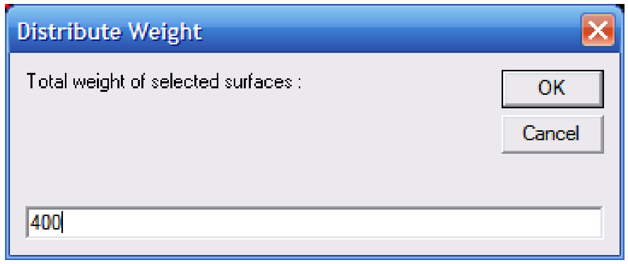


Figure 4-9: Enter weight of the selected surfaces, here as 400 lbs (Step 7).

This will distribute a total weight of 400 lbs onto the model based on the area. That is, **SURFACES** calculates the total area of the selected surfaces and then computes weight per total area. The weight property of each surface will then be assigned a number, which is calculated as (weight per total area of the selected surfaces) x (the area of the surface). As a consequence, the total weight of the wings turns out to be 293.3 lbs, the HT weighs 58.2 lbs, and the VT weighs 48.5 lbs. Clearly, this adds up to 400 lbs. You can check weight by selecting surfaces and pressing the F6 button (or by selecting **Tools->Properties of Selected Surfaces**). The results will be displayed on the Status bar on the bottom of the main window.

STEP 8: Make sure the CG is visible. Select **Tools->Options...** Check the 'Show CG, Neutral Point, Aerodynamic Center' checkbox and press the [OK] button (see Figure 4-10).

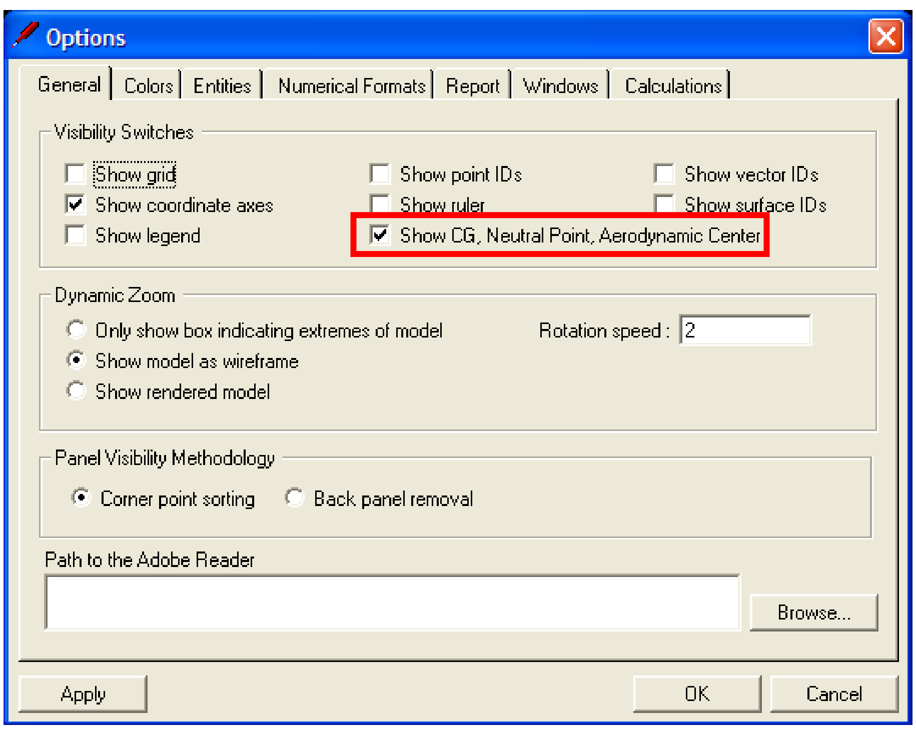


Figure 4-10: Confirm the CG checkbox is marked so you can see the CG in the workspace (Step 8).

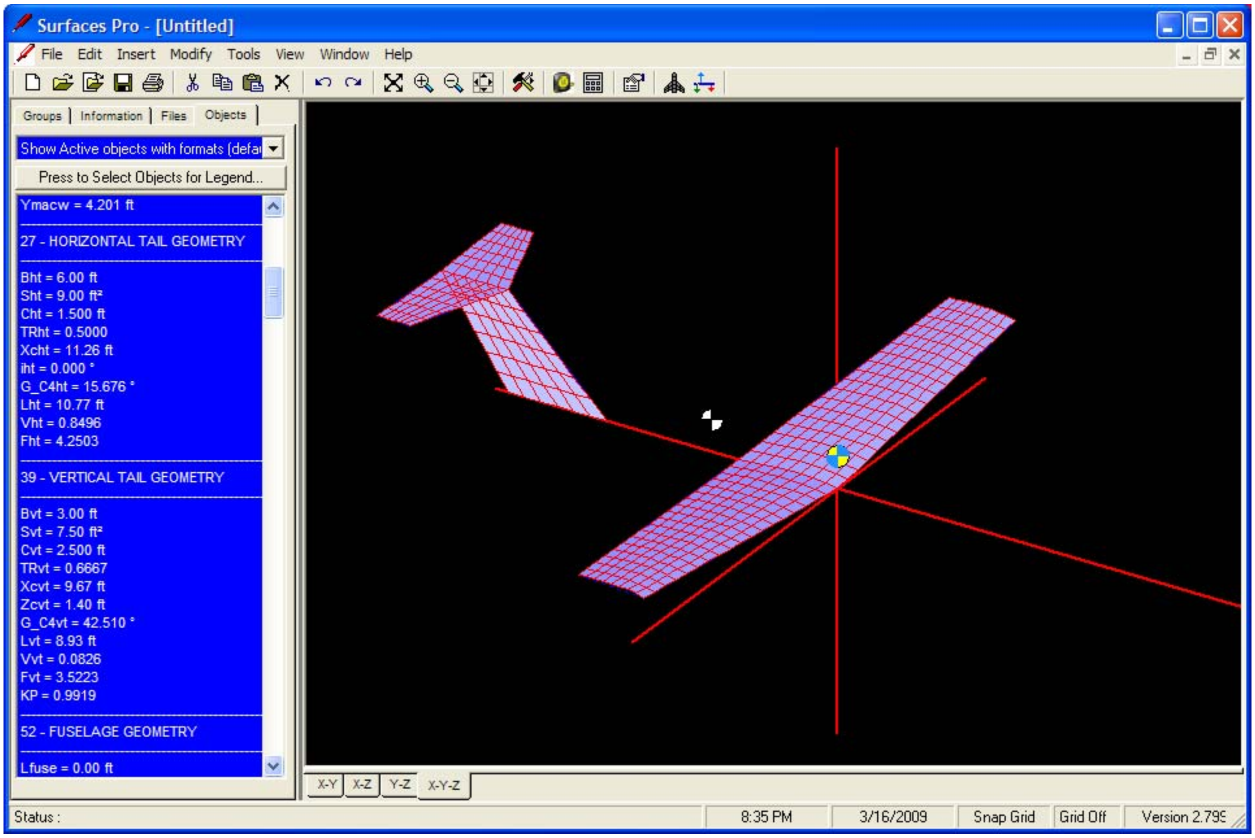
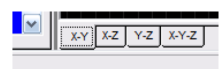


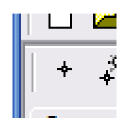
Figure 4-11: We can see the CG location (black-white circle) is too far aft.

When completed, your model should look like the one in Figure 4-11. It is immediately evident that the CG is too far aft. To fix this and to allow us to control the location of the CG, let's create a ballast point.



STEP 9: Press the X-Y tab on the bottom of the workspace. This will display the model projected onto the X-Y plane.

STEP 10: Press the sketch-mode icon to display the sketch toolbar.



STEP 11: Press the *Insert a point* icon and drop a point somewhere in front of the wing, near the X-axis, similar to what is shown in Figure 4-12.

STEP 12: Select the point by clicking on it and press the *Insert a node point* icon to convert it to a node. This will open a dialog box to allow user to enter additional data. Enter the information shown in Figure 4-13. Once completed, press the [OK] button.

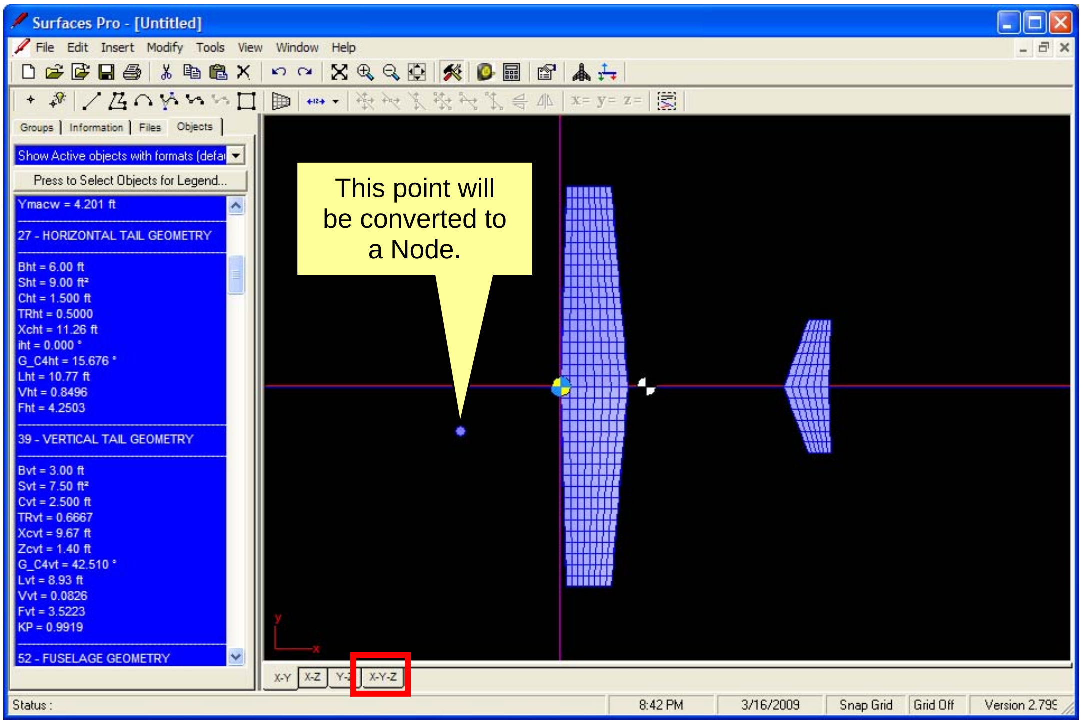


Figure 4-12: Drop the point (to be converted to a node) in a location similar as shown (Step 11).

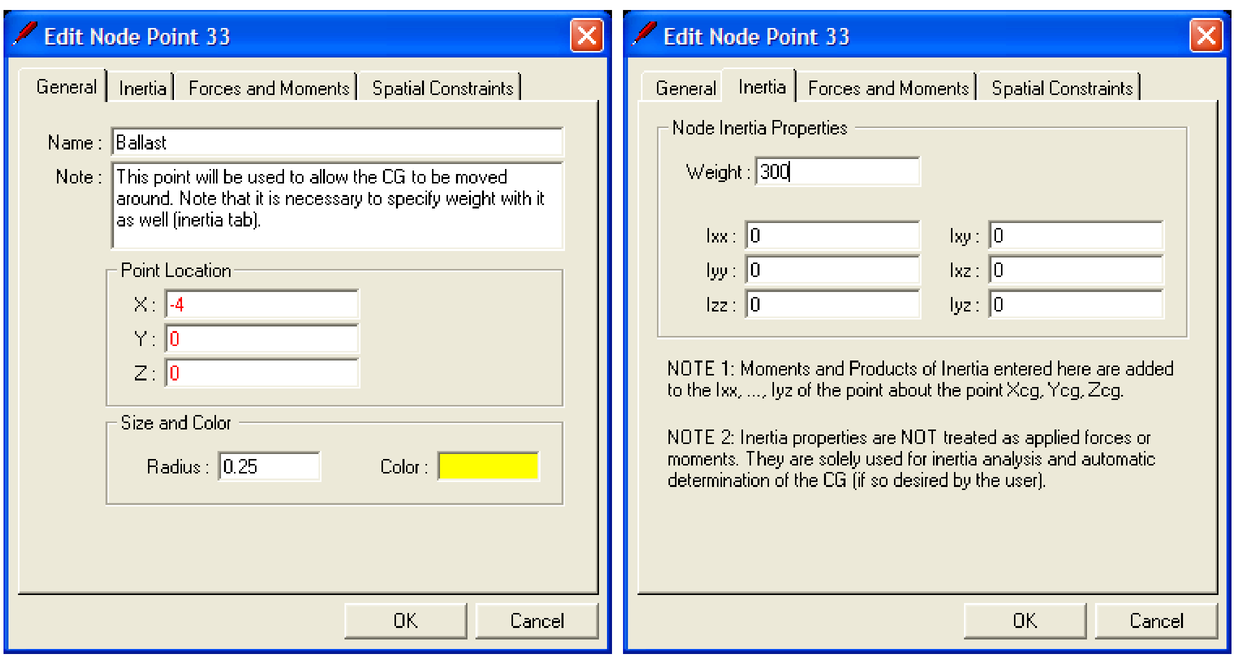


Figure 4-13: Information entered with Step 12.

Return to the 3-D view by pressing the X-Y-Z tab (see the bottom of Figure 4-12). When completed your model should look like the one in Figure 4-14. To see what the true location of the CG is at this point, locate the math objects `Pmac` and `Xcg` in the object list on the left hand side (`Pmac` is highlighted in

Figure 4-14). The variable `Pmac` stores the CG location as a percentage of the Mean Aerodynamic Chord (`Cref`, found under the REFERENCE PARAMETERS block in the Math Object list). We see the CG is located at 13.967% MAC or at 0.47 ft. Often it is necessary to specify directly the location of the CG. **SURFACES** comes with a tool to help you accomplish that. The following steps show how to move the CG to 25% MAC.

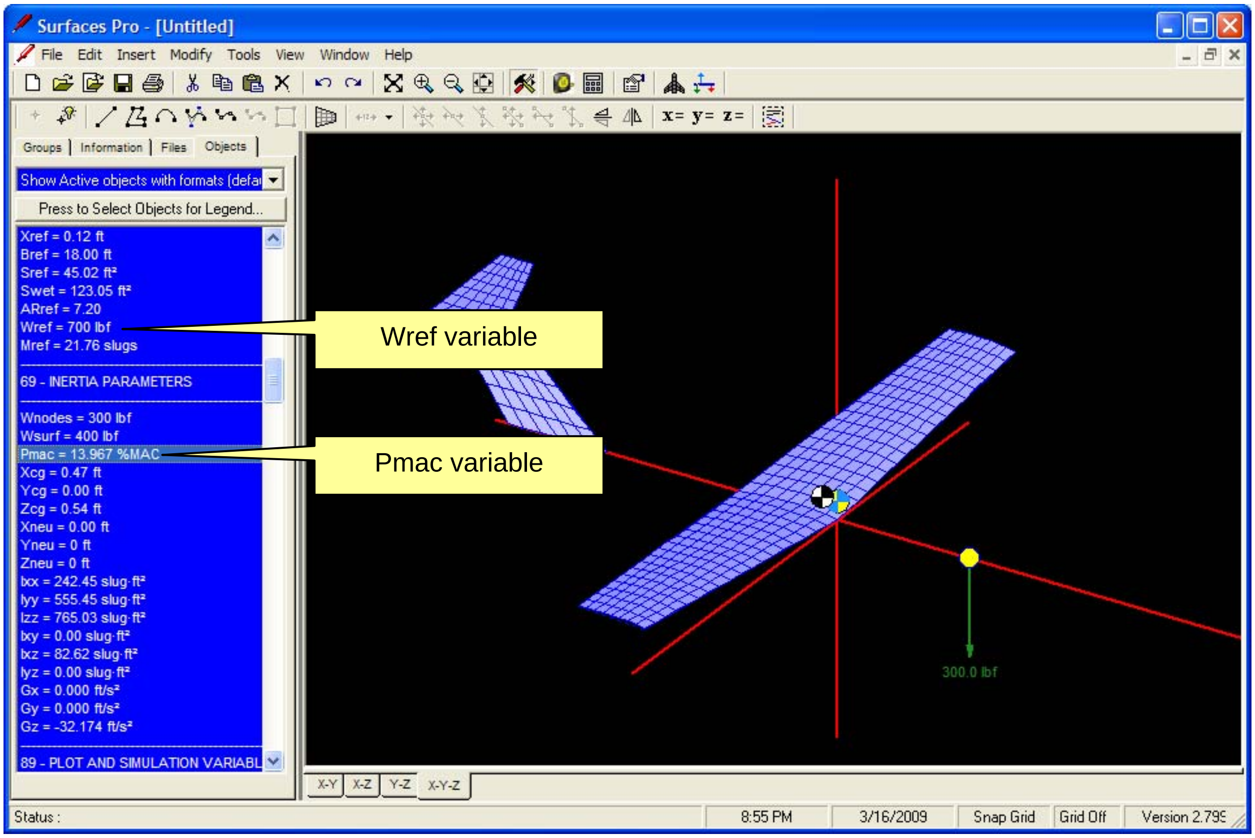


Figure 4-14: The model with ballast point defined.

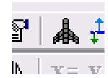
STEP 13: Click once on the Ballast node to select it. We will move it with a special tool. Note that **SURFACES** will only move the selected node or nodes, when adjusting the CG location. If none are selected a warning message appears.

STEP 14: Select **Tools->Specify a CG Location…** Select the option and enter the value shown in Figure 4-15.

STEP 15: Press the [Adjust] button. Respond to the warning that appears by pressing [Yes]. Then, press [Close] button to exit the form.

When completed, your node will appear closer to the wing than before, or but **SURFACES** has automatically changed its X location from -4 to -3.347556 ft, moving the CG in the process (i.e. to the 25% MAC). Now let's learn some more details about the model. Let's determine the neutral point per the following steps.

STEP 16: Press the VLM Console icon. This will open the Vortex-Lattice Method Console shown in Figure 4-16.



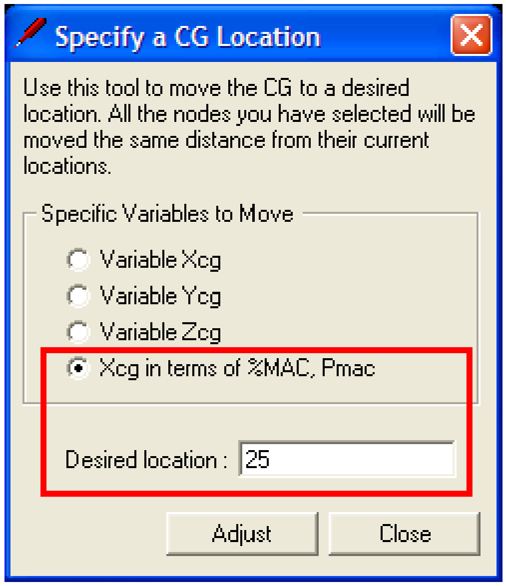


Figure 4-15: Specifying a CG location (Step 14).

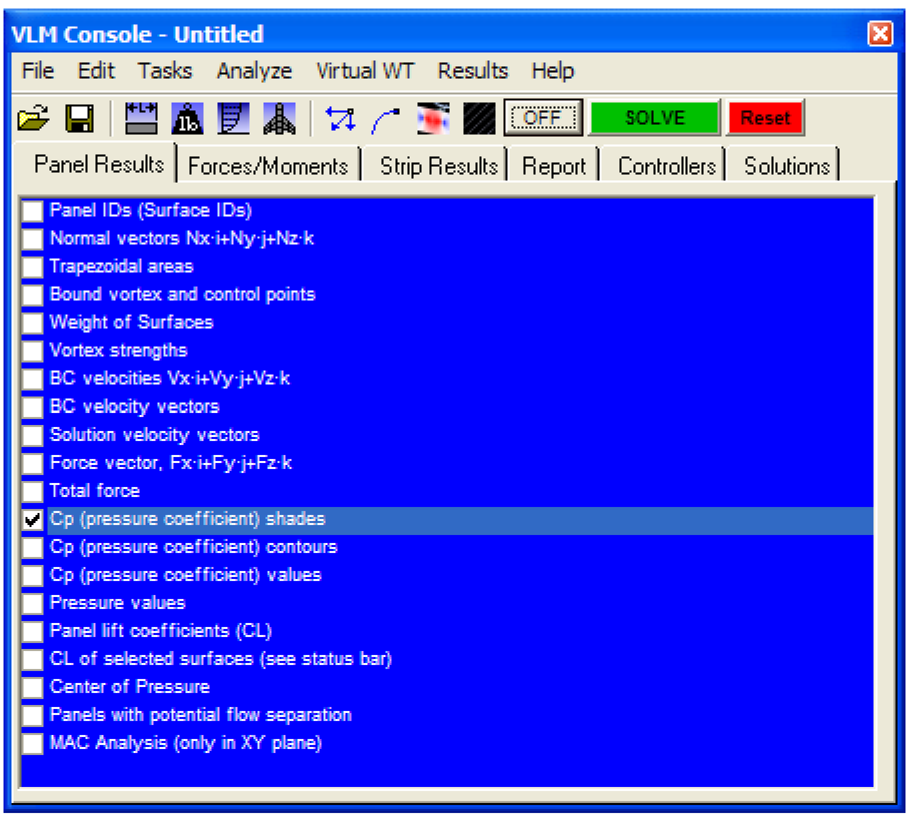


Figure 4-16: The VLM Console (Step 16).

Note that when you create a new project, **SURFACES** has preset values for a multitude of variables. Among those are the airspeed variables (`Vcas`, `Vtas`, `Vinf`), altitude (`Href`), and angle-of-attack (AOA). Naturally, you can change these with ease, but currently `Vcas`=100 knots, `Href`=0 ft, and AOA=2°. In interest of saving time for this demo, let's assume these will suffice for our analysis.

STEP 17: Select **Tasks->Determine Neutral Point…** Press the [Analyze] button to begin, and after a few seconds, once done, review the results in Figure 4-17.

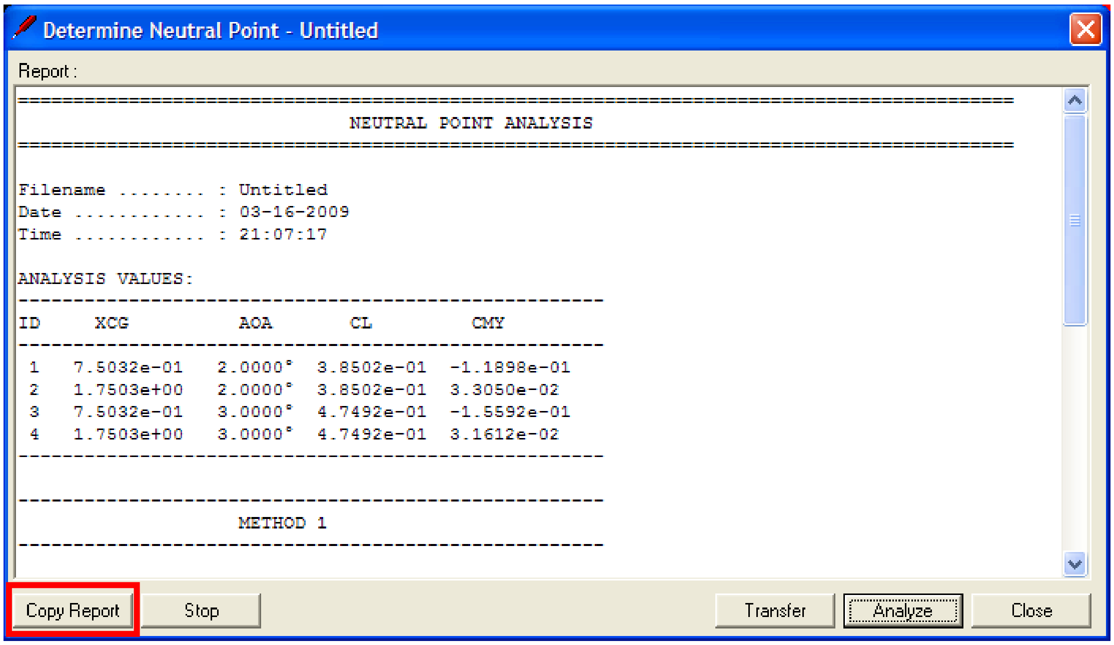


Figure 4-17: Determining neutral point (Step 17).

The full report is displayed below. Note that **SURFACES** uses two methods to compute the neutral point. Generally you should pick the neutral point with the lower value of `Xneu`, here this implies Method 2. Let's transfer the resulting value to the variable `Xneu` in the model, which currently has the initial value 0.

```
==========================================================================================
                              NEUTRAL POINT ANALYSIS
==========================================================================================

Filename ........ : SimplePlane(03162009).SRF
Date ............ : 03-16-2009
Time ............ : 21:17:00

ANALYSIS VALUES:
-----------------------------------------------------
ID     XCG          AOA       CL         CMY
-----------------------------------------------------
 1   7.5032e-01   2.0000°  3.8059e-01  -1.1761e-01
 2   1.7503e+00   2.0000°  3.8059e-01  3.2670e-02
 3   7.5032e-01   3.0000°  4.6946e-01  -1.5413e-01
 4   1.7503e+00   3.0000°  4.6946e-01  3.1248e-02
-----------------------------------------------------

-----------------------------------------------------
                    METHOD 1
-----------------------------------------------------
Calculates Xneu from the expression:
```

```
          Xneu = Xcg - Cref·dCm/dCL
               = 0.75032 - (2.534505)·(-0.036512)/(0.088875)
               = 1.791557 (66.08247% MAC)

-----------------------------------------------------
                    METHOD 2
-----------------------------------------------------
Calculates Xneu by evaluating changes of CG and AOA on Cm:

Function 1 (degrees): -0.036512·AOA - 0.044590
Function 1 (radians): -2.091979·AOA - 0.044590

Function 2 (degrees): -0.001422·AOA - 0.154126
Function 2 (radians): -0.081474·AOA - 0.154126

          Xneu = 1.790844 (66.05433% MAC)

Total time:0h:00m:05s
```

STEP 18: Press the [Transfer] button and select the option 'Neutral point using Method 2'. This displays a notification. Press the [OK] button to close it. Press the [Close] button on the form to close it as well.

Note the [Copy Report] button in the form in Figure 4-17. It allows you to copy the entire text in the form to the clipboard. We consider it a good practice to copy and paste it as a comment under **Edit->Remark…** in the main worksheet for future reference.

Now let's trim the aircraft for a level flight. First we must define which surfaces serve as the elevators. To do that, return back to the worksheet where the model is.

STEP 19: Double-click on one of the two surfaces that serve as the horizontal tail. This opens the dialog box shown in Figure 4-18. Select the 'Edge Deflections' tab. Set number of chordwise panels on the aft edge to deflect to 2.

STEP 20: In the same dialog select the 'Reference' tab. Check the 'Surface is used for Pitch Control'. Press the [OK] button. If a warning appears stating there's already a VLM solution in memory, just press the [Yes] button.

STEP 21: Repeat Steps 19 and 20 for the other horizontal surface.

Also, by now, it would be a good idea to save the work. Here, we select **File->Save As…** and call it SIMPLE DEMO.SRF. You should do the same.

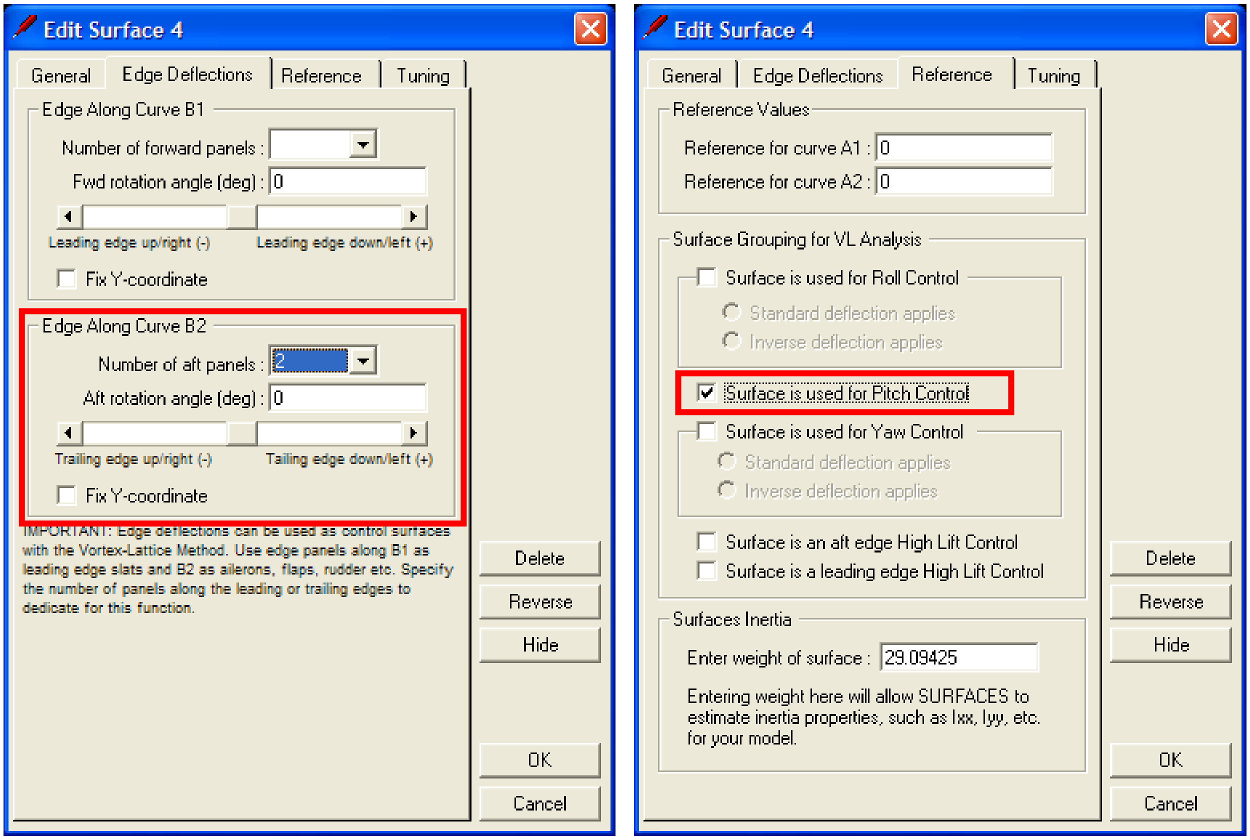


Figure 4-18: Setting up elevator functionality (Steps 19-21).

You have now given **SURFACES** information it can use to automatically deflect the elevators to trim the model for level flight. You can try the functionality out by displaying the VLM Console and select the 'Controllers' tab. For instance, enter -20 in the Pitch control textbox and press the [Set] button to see the model regenerate with that deflection, as shown in Figure 4-19. Once done, press the [Reset] button to return the elevators to a neutral deflection (0°) and get ready to trim the model.

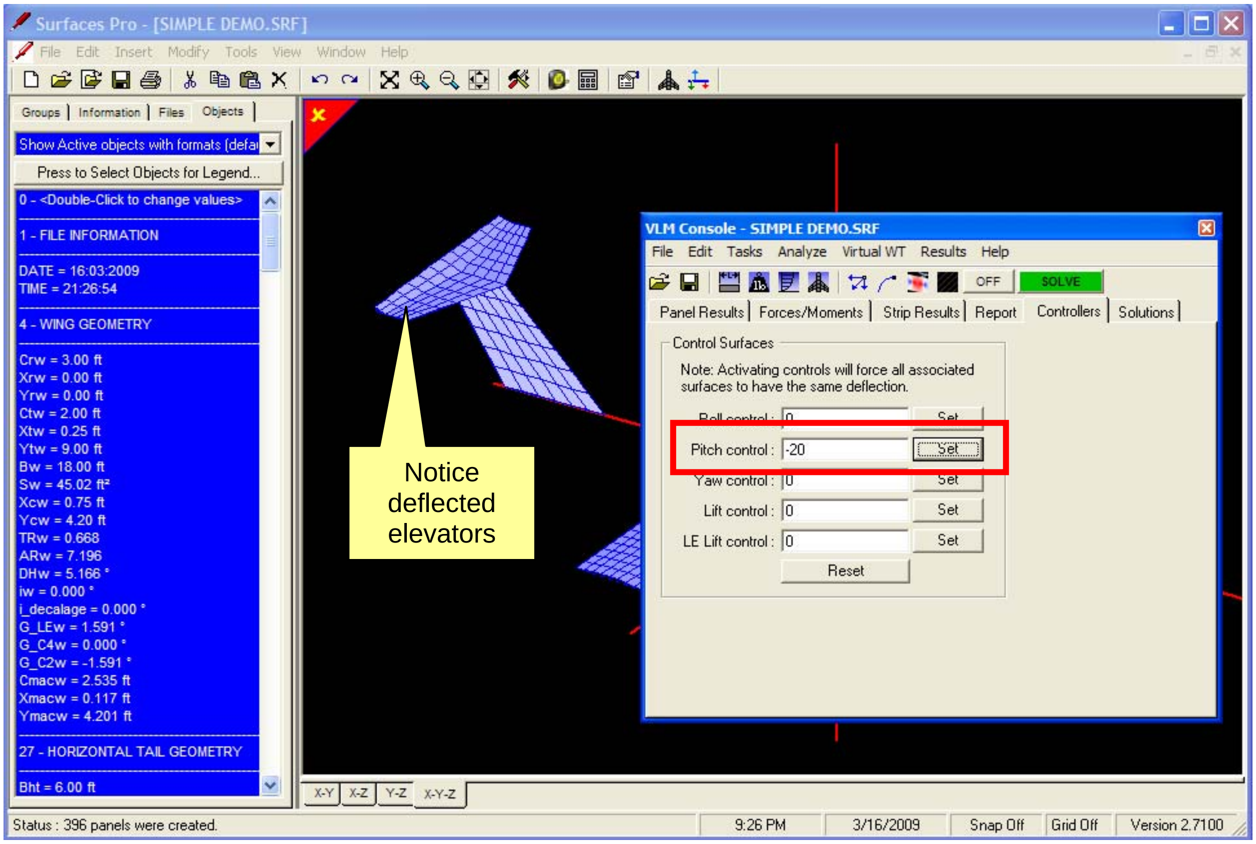
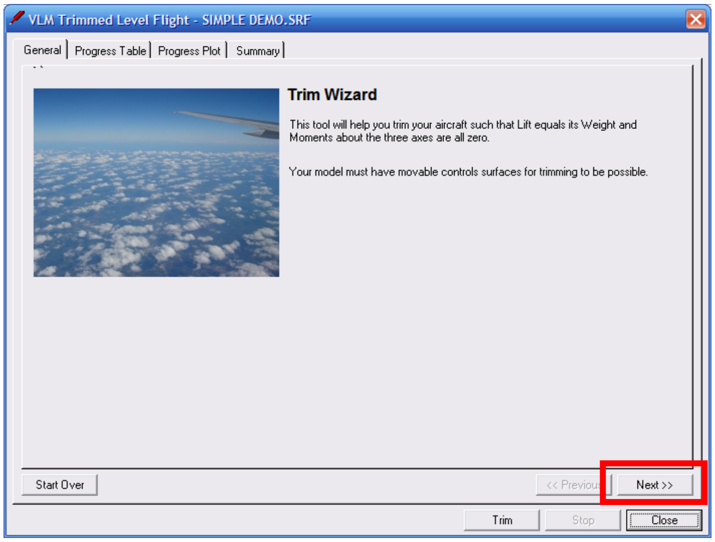
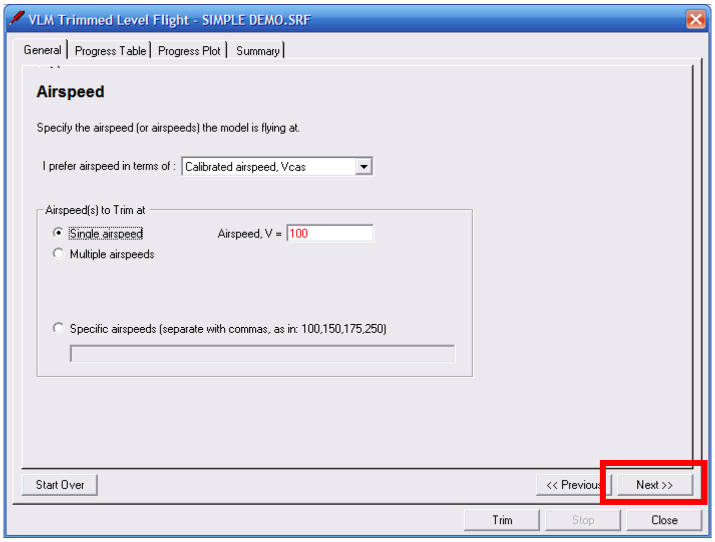


Figure 4-19: Demonstrating elevator functionality.

STEP 22: Select **Tasks->Trimmed Level Flight...** to display the Trim wizard. Follow the steps shown in the subsequent list of images.

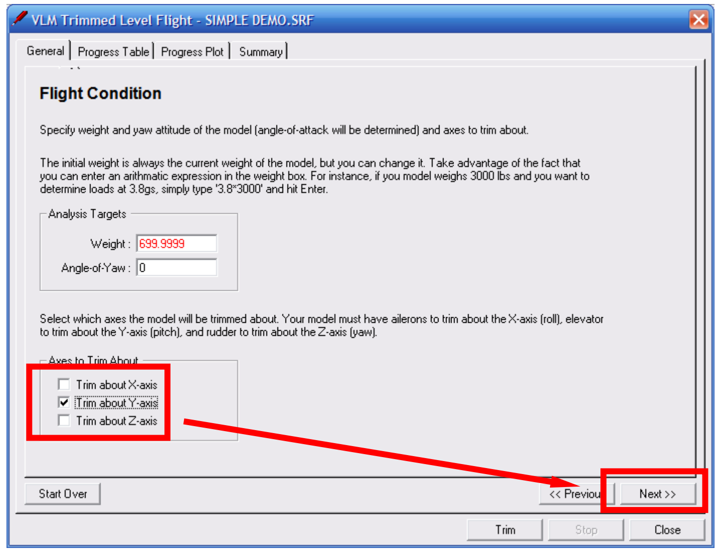


STEP 22a:
Press the [Next >>] button.



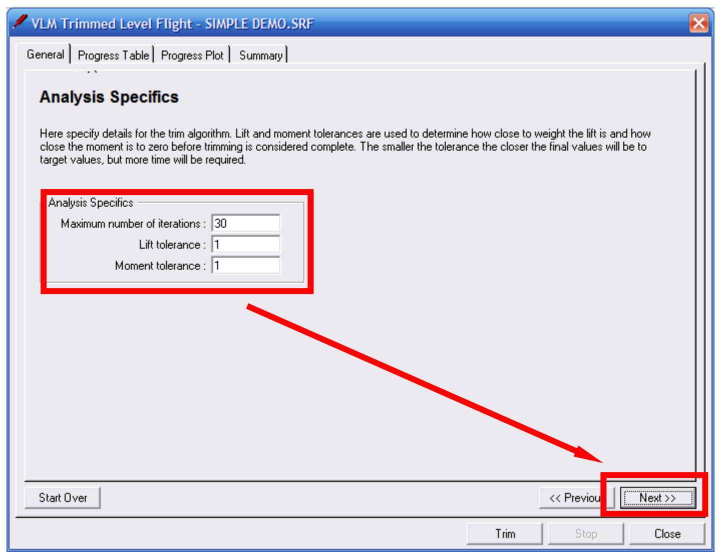
STEP 22b:
Ensure the selection shown.
Press the [Next >>] button.

We will just trim to a single airspeed, but multiple airspeeds can also be analyzed.



STEP 22c:
Ensure the selection shown.
Press the [Next >>] button.

Once complete, the lift generated will be 700 lbs at the airspeed specified in Step 22b.

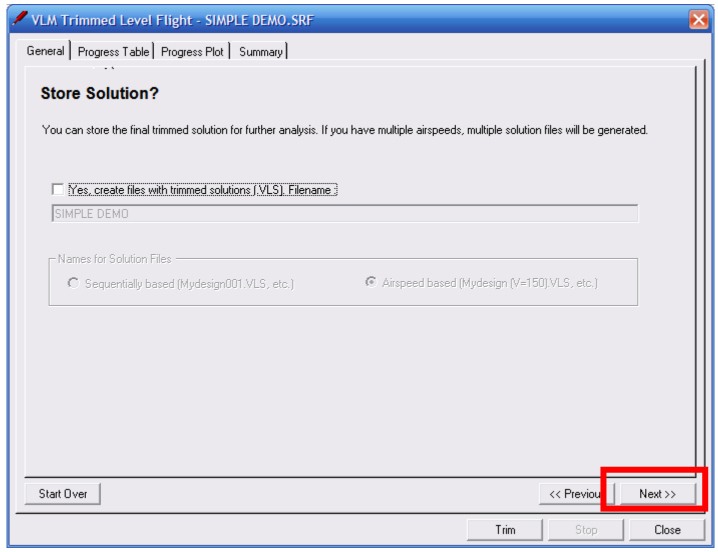


STEP 22d:
Ensure the selection shown.
Press the [Next >>] button.

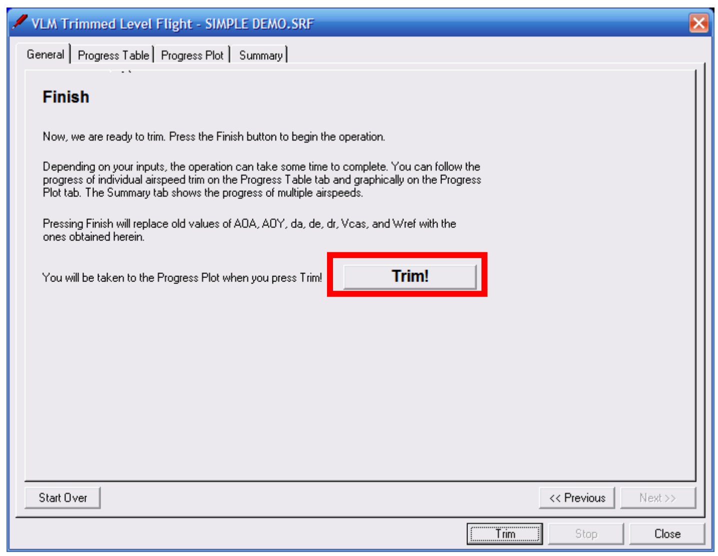
Here we allow 30 iterations before a solution will be declared as unachievable. If solution is found, the resulting lift will be 700 ±1 lb_f and the moment 0 ±1 $ft\cdot lb_f$. As a rule of thumb, acceptable accuracy is provided by specifying 1% of the weight. Here, the accuracy is closer to 0.14%.

STEP 22e:
Ensure the selection shown.
Press the [Next >>] button.

Note that solution files can be created and saved using the file name entered as a seed.



STEP 22f:
Press the [Trim!] button.



Once **SURFACES** begins to trim, you can follow the progress on the 'Progress Table' or 'Progress Plot' tabs (see Figure 4-20). The time to trim largely depends on the number of panels in the model and accuracy desired. The model presented here took 16 iterations and 31 seconds to trim. Press 'Summary' tab to read the results for each completed trim speed. In this case, the model will fly level at an AOA of 3.3449° and will require an elevator deflection of -4.3966° (trailing edge up) to balance. The lift generated is 699.587 lbf and moment about the y-axis (located at the CG) is 0.287698 ft·lbf. The model is automatically set to the resulting AOA and elevator deflection. Press the [Close] button to exit the form.

Next let's determine stability derivatives for the model in this particular configuration.

STEP 23: Select **Tasks->Determine Stability Derivatives...** to display the Stability Derivatives form. Check and uncheck the boxes shown in Figure 4-21 and press the [Analyze] button.

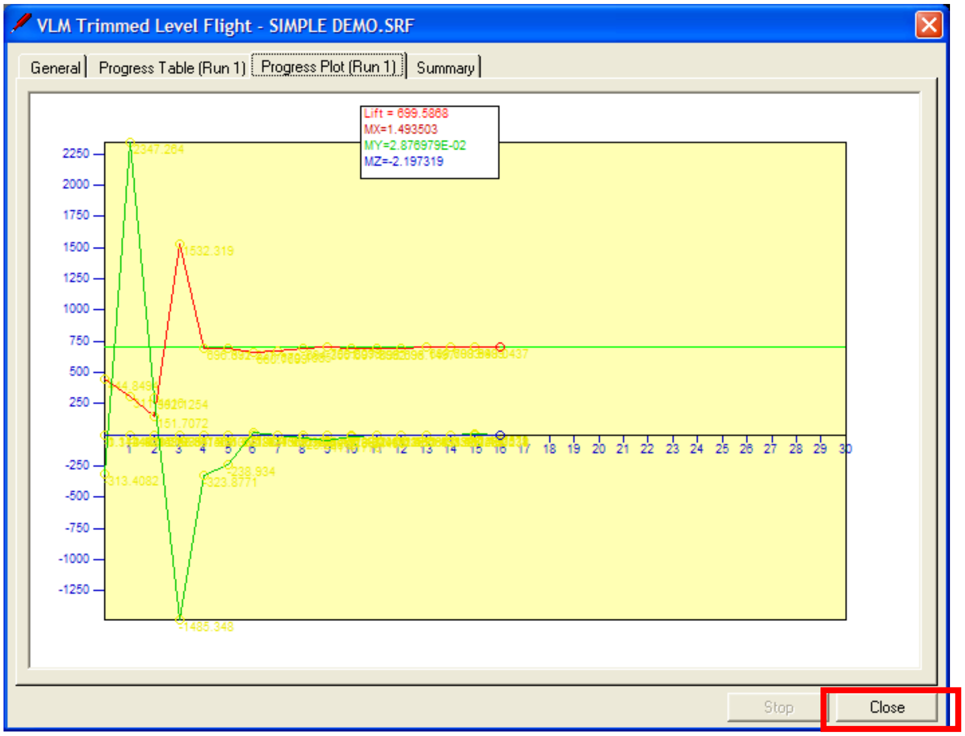


Figure 4-20: Trim progress is displayed on the 'Progress Plot' tab.

Upon completion you will see the results as shown in Figure 4-22. Without going into too many details, we can see from values for Cma (-2.119), Clb (-0.105), and Cnb (0.172) that our airplane is statically stable about all three axes. What we don't know at this time are its dynamic stability properties. And this is what we intend to investigate next. First, however, we must transfer these results to the airplane model.

STEP 24: Select the 'Transfer' tab. Follow the remaining steps closely.

STEP 25: Press the [Select All] button to select all the derivatives in the list.

STEP 26: Press the [Deselect Nonrequested] button to deselect the derivatives that were not calculated.

STEP 27: In addition, uncheck the following variables: CL, CDi, CD, CDa, hcg, and hn (see Figure 4-23). This will prevent them from being overwritten, but they already contain algebraic expressions that we don't want to be deleted.

STEP 28: Press the [Transfer] button. Press [Yes] (in this example) if prompted to overwrite formulas. Press the [OK] button on the form that appears to notify you of a successful transfer. Then press the [Close] button to close the Stability Derivatives form.

Now let's proceed to the dynamic stability analysis.

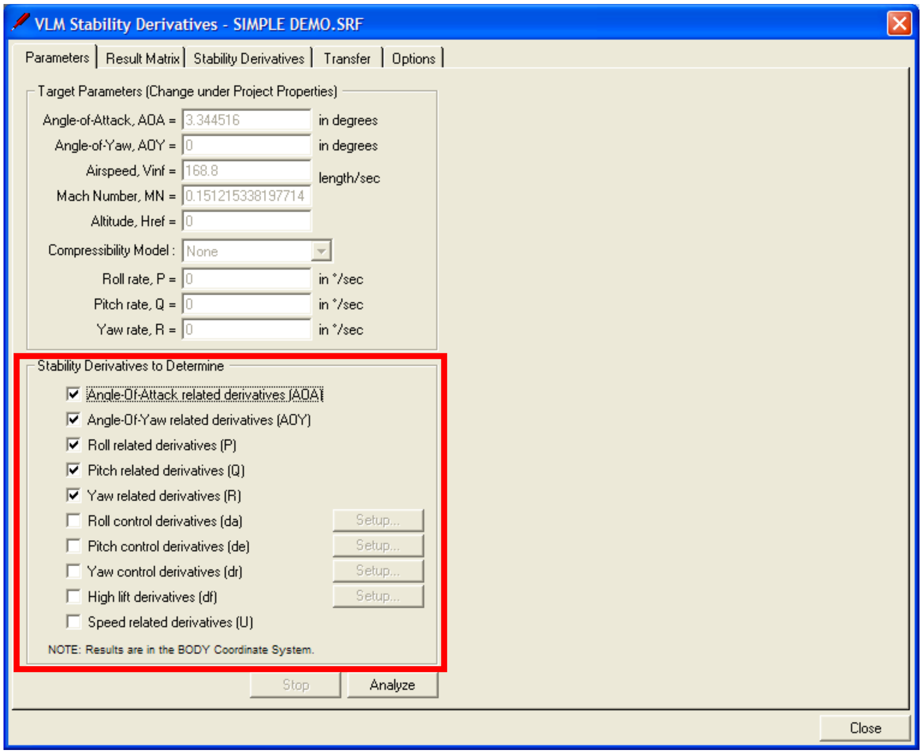
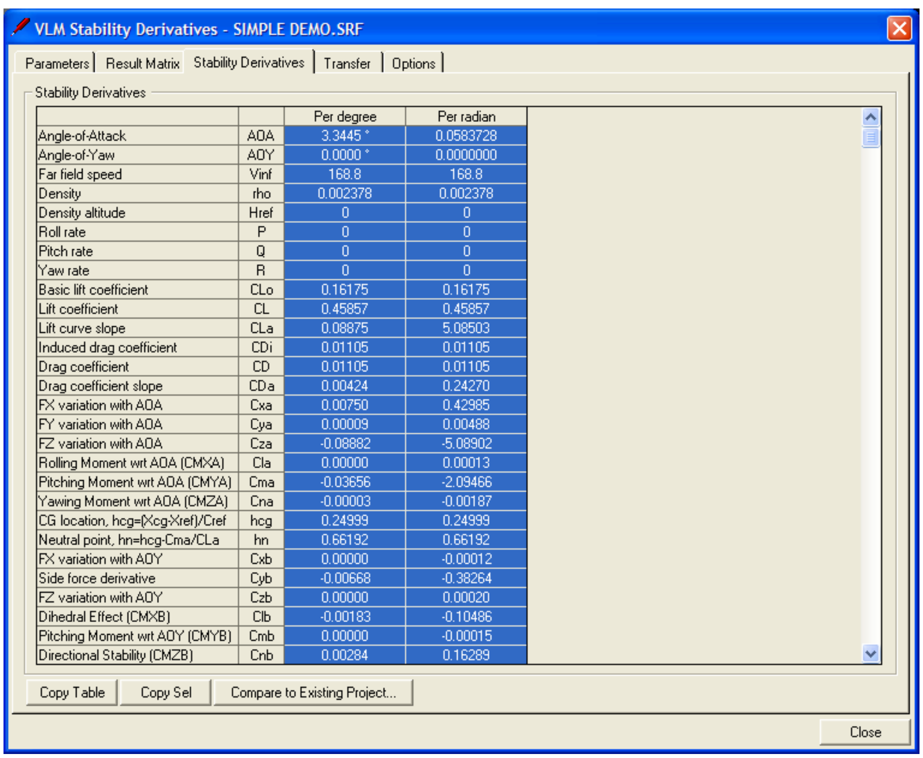


Figure 4-21: Preparing to determine stability derivatives (Step 23).



		Per degree	Per radian
Angle-of-Attack	AOA	3.3445 °	0.0583728
Angle-of-Yaw	AOY	0.0000 °	0.0000000
Far field speed	Vinf	168.8	168.8
Density	rho	0.002378	0.002378
Density altitude	Href	0	0
Roll rate	P	0	0
Pitch rate	Q	0	0
Yaw rate	R	0	0
Basic lift coefficient	CLo	0.16175	0.16175
Lift coefficient	CL	0.45857	0.45857
Lift curve slope	CLa	0.08875	5.08503
Induced drag coefficient	CDi	0.01105	0.01105
Drag coefficient	CD	0.01105	0.01105
Drag coefficient slope	CDa	0.00424	0.24270
FX variation with AOA	Cxa	0.00750	0.42985
FY variation with AOA	Cya	0.00009	0.00488
FZ variation with AOA	Cza	-0.08882	-5.08902
Rolling Moment wrt AOA (CMXA)	Cla	0.00000	0.00013
Pitching Moment wrt AOA (CMYA)	Cma	-0.03656	-2.09466
Yawing Moment wrt AOA (CMZA)	Cna	-0.00003	-0.00187
CG location, hcg=(Xcg-Xref)/Cref	hcg	0.24999	0.24999
Neutral point, hn=hcg-Cma/CLa	hn	0.66192	0.66192
FX variation with AOY	Cxb	0.00000	-0.00012
Side force derivative	Cyb	-0.00668	-0.38264
FZ variation with AOY	Czb	0.00000	0.00020
Dihedral Effect (CMXB)	Clb	-0.00183	-0.10486
Pitching Moment wrt AOY (CMYB)	Cmb	0.00000	-0.00015
Directional Stability (CMZB)	Cnb	0.00284	0.16289

Figure 4-22: Stability derivatives for the model (Step 23).

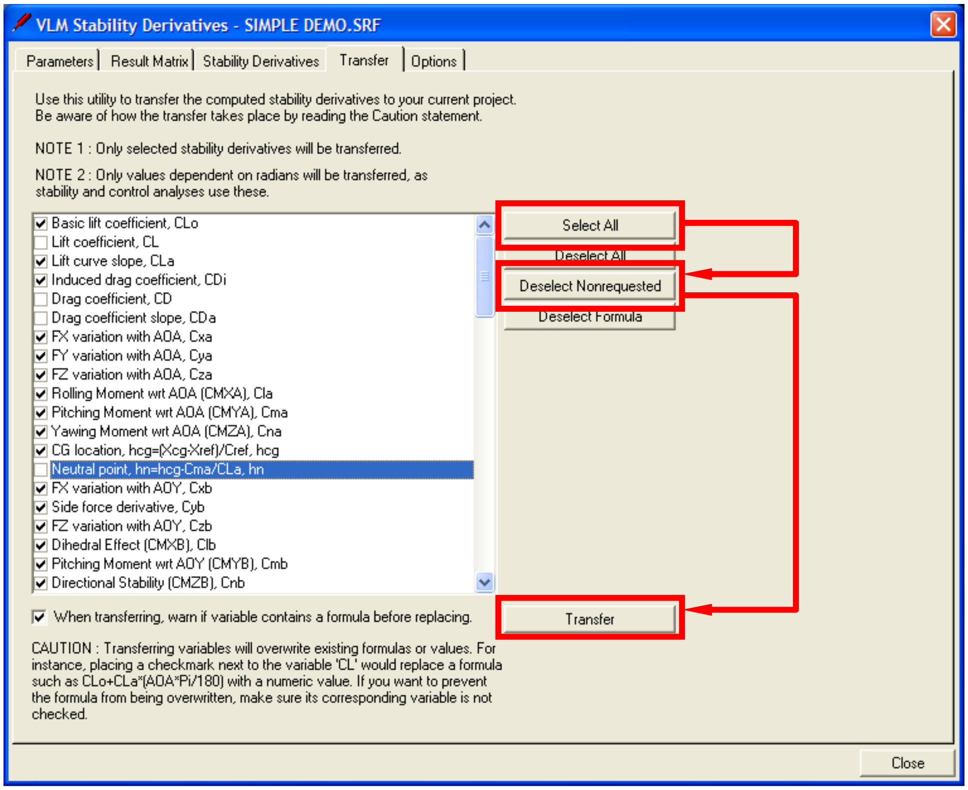
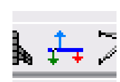
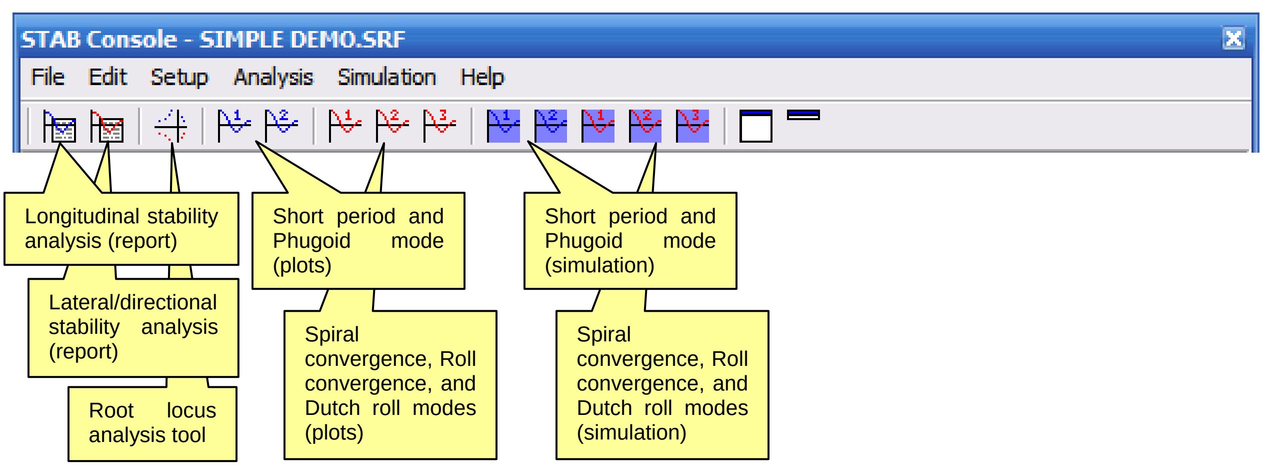


Figure 4-23: Stability derivatives for the model (Steps 24-28).

STEP 28: Press the STAB Console icon. This will open the Stability Analysis Console shown in Figure 4-29.



It is left as an exercise for the user to press the various icons to experience functionality. The simulation icons will display the motion of the aircraft in real time.



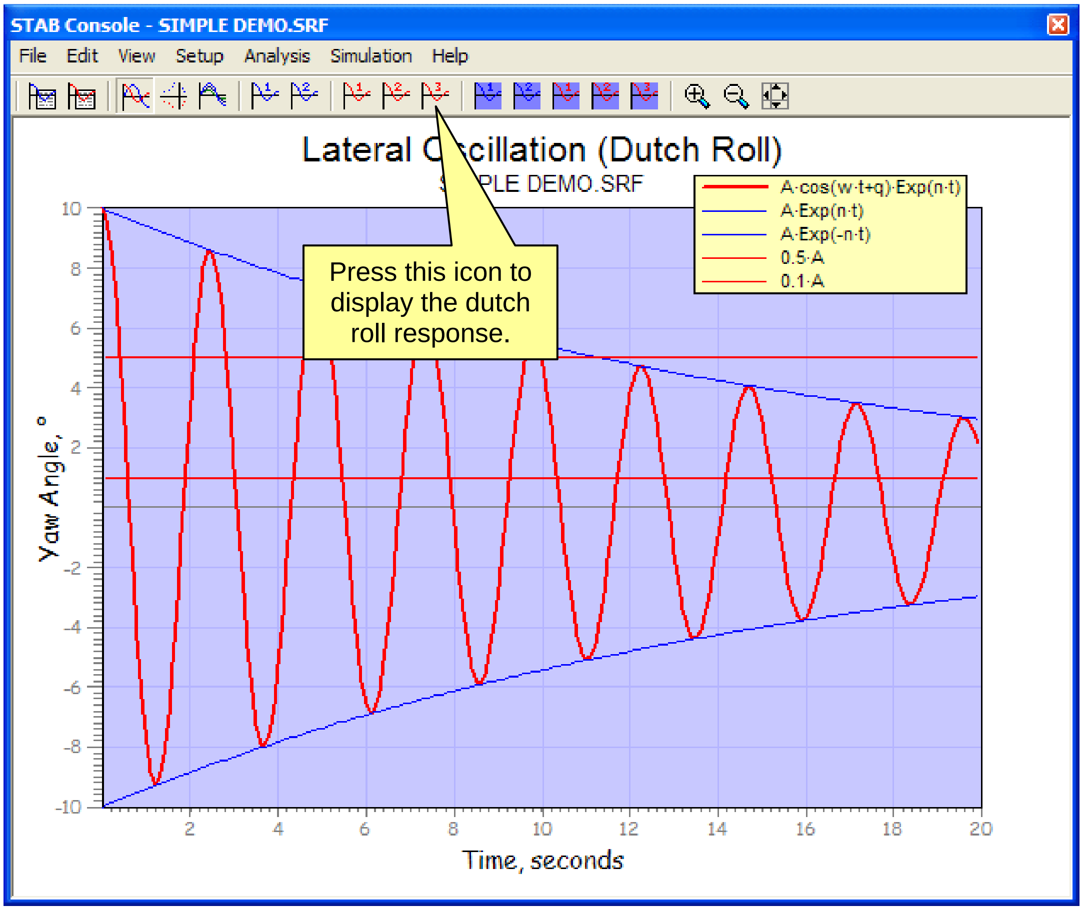


Figure 4-24: Stability analysis module (Step 29.

You can get a report detailing the properties of the response by selecting **View->Show Comparison Table**. The resulting table is shown below. This is but one of many ways to extract information from the STAB module. Also try **Analysis->Create Analysis Report…** to get a more detailed dynamic stability report.

Description	Symbol	Unit	SIMPLE DEMO.SRF
Airspeed	Vtas	KTAS	100
Altitude	Href	ft	0
Period of oscillation	T	sec/cycle	2.450
Damping coefficient	n	1/sec	-0.0612
Natural frequency	Wn	cycles/sec	2.5648
Damped frequency	Wd	cycles/sec	2.5641
Damping Ratio	Zeta		0.0238
Time to 0.5 Amplitude	t½	sec	11.3324
Cycles to 0.5 Amplitude	N½	cycles	4.6246
Time to 0.1 Amplitude	t0.1	sec	37.6454
Cycles to 0.1 Amplitude	N0.1	cycles	15.3626

This concludes the introductory example. This model is also used for a skin friction drag demo in Section 9, so it will be convenient to save it.

5. Accomplishing Special Projects with SURFACES

5.1 Tailoring Wings to Improve Stall Characteristics

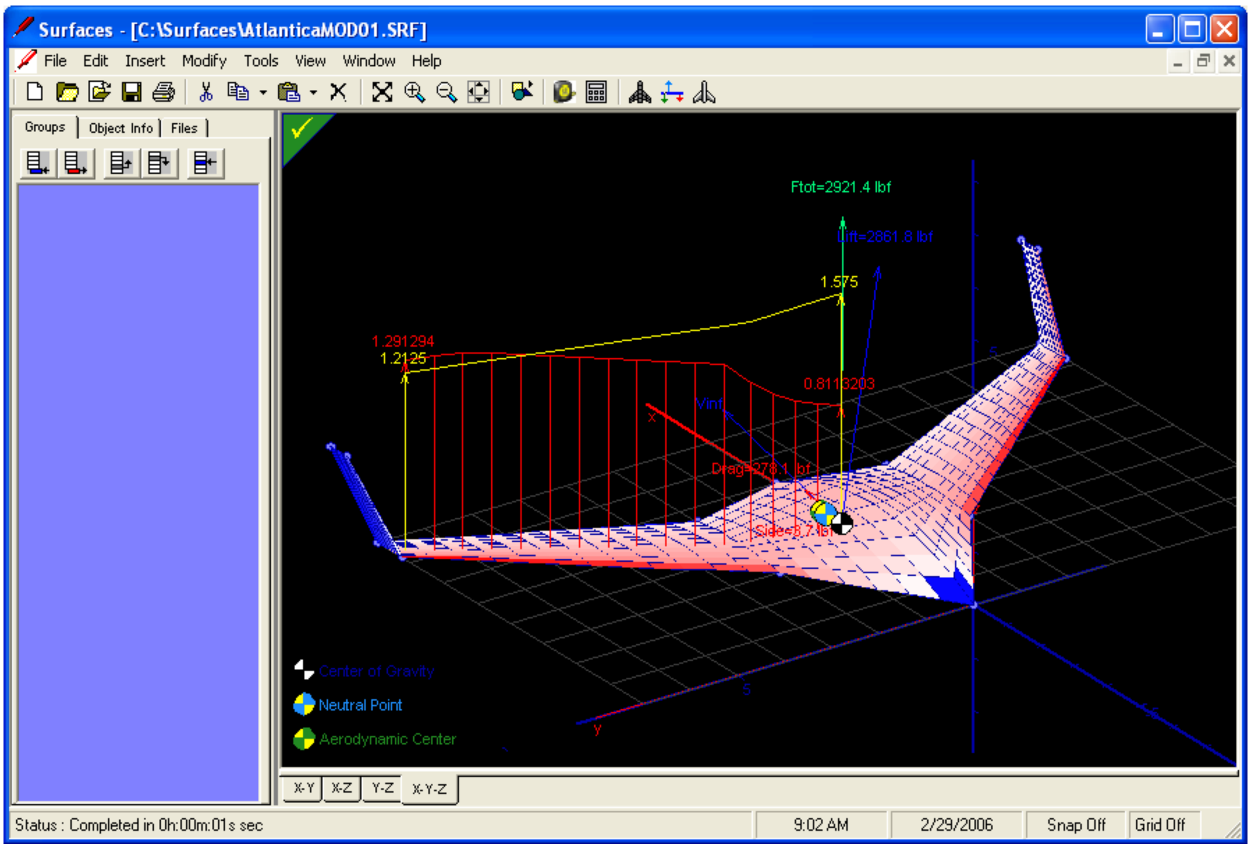


Figure 5-1: A model whose section lift coefficients near the tip are very high ("tip-loaded).

Figure 5-1 shows how **SURFACES** can be used to help optimize stall characteristics. The yellow line represents section lift coefficients at stall. These are entered as reference values for curves A1 and A2 for each surface. The red lines represent section lift coefficients at the flight condition. The image shows the wing tip stalls long before the inboard part of the wing. Not only would this cause the airplane to a roll at stall (as one wing tip is prone to stalling before the other one), but more seriously, would result in an uncontrollable nose pitch-up moment. This situation can be remedied by modifying the wing geometry, for instance by adding wing washout, increase tip chord, reduce sweep, or using airfoils with a higher max lift coefficient.

5.2 Determine Shear, Moment, and Torsion

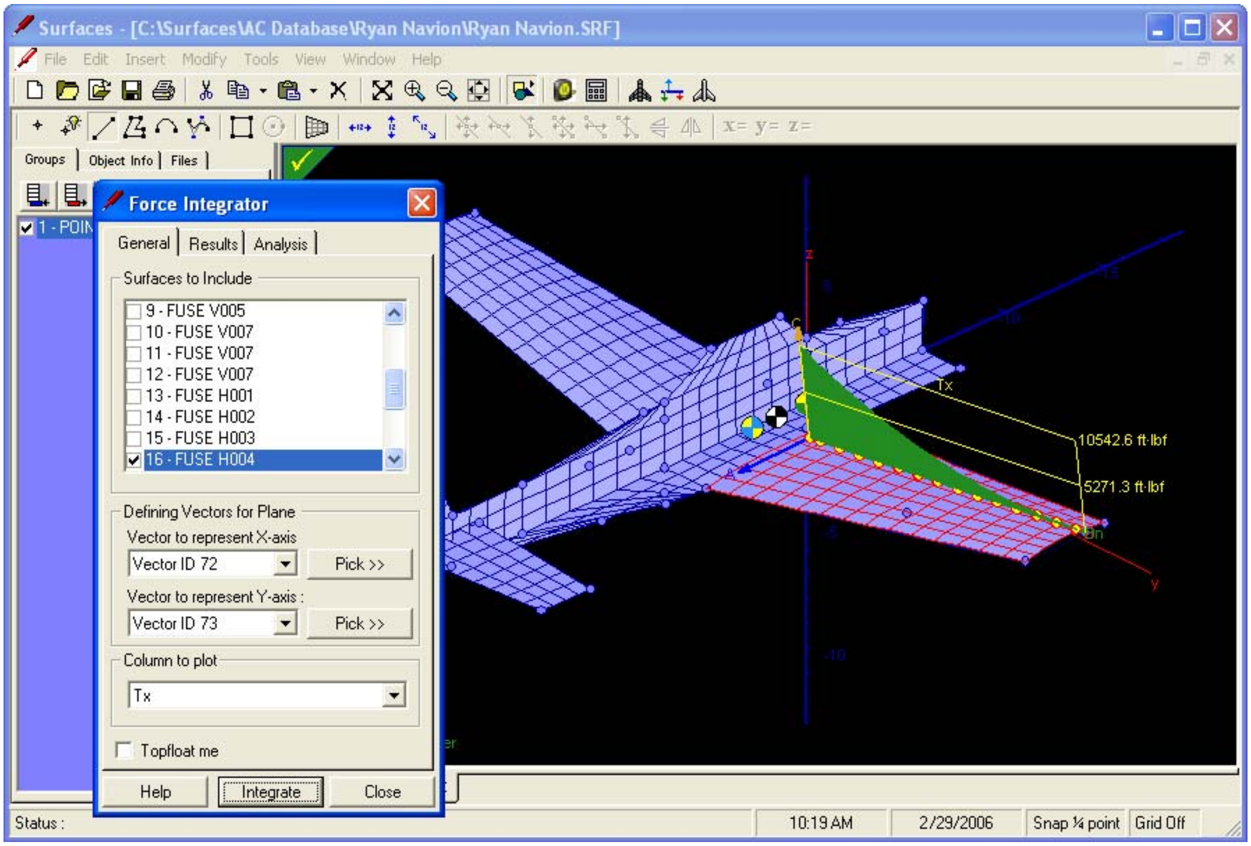


Figure 5-2: Obtaining shear and moment distribution for a lifting surface.

SURFACES comes equipped with a tool that allows you to analyze cantilevered shear and moment acting on any surface. Figure 5-2 shows the Force Integrator tool as applied to the right wing on the Ryan Navion model. The bending moments along the right wing are plotted. Note the wing curvature represents the camber line of the aircraft's airfoils.

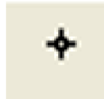
5.3 How to Manage Airfoils in SURFACES

SURFACES allows the user to study the influence of airfoils on flight characteristics. This is done by specifying the camber line of the airfoil. The program comes with a tool that helps the user to do this more easily (see Figure 4-1d). The user can define camber lines using four different curves; a parametric, a 4-point Bezier curve, a list of points, or a B-spline. In order to do this effectively, the user must keep the some rules in mind when manipulating or managing curves. The following example, in which a parametric curve is created, gives an insight into how this is done.

`STEP 1:` `Start a new project. Select` **`File->New…`**

`STEP 2:` `Go into sketch mode by pressing the`  `icon.`

Great OWL Publishing
Engineering Software

STEP 3: Select the point icon  and drop two points. One at 1,3 (point A) and the other at 9,3 (point B. See Figure 5-3).

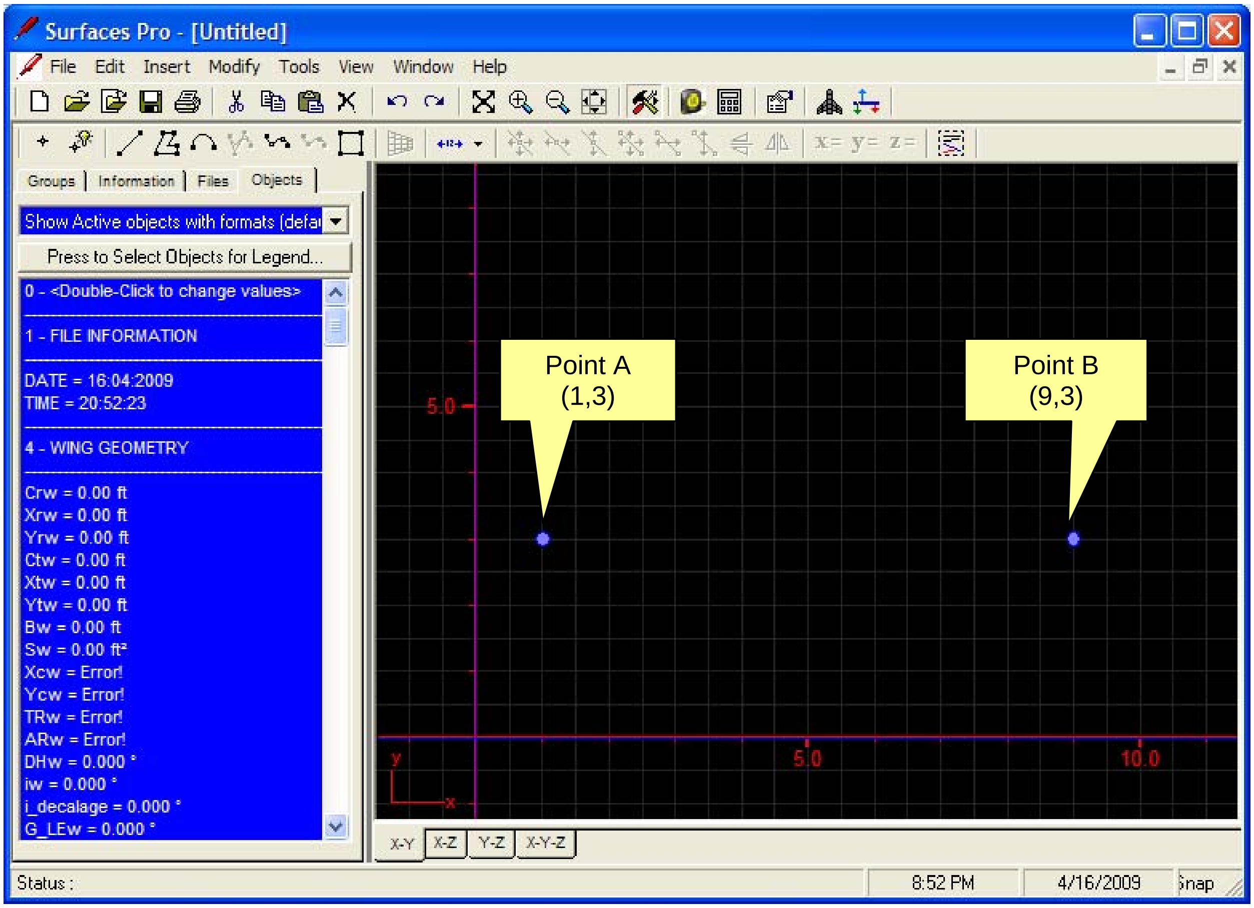
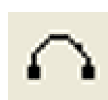


Figure 5-3: Defining start and end points for a vector in the X-Y plane.

STEP 4: Select the parametric curve icon  and stretch a curve from point A to point B. Right click to stop (see Figure 5-4).

If you select the X-Y-Z view, you can see that **SURFACES** has created a third point (see Figure 5-5). This point is called an **alignment point**. If you select the vector you'll see that **SURFACES** highlights the vector, but also a line extending from the start point to this third point (see Figure 5-6).

The purpose of this point is to allow you to orient the parametric curve in 3D space. Let us create a simple parametric curve to demonstrate this better.

STEP 5: Double-click the parametric curve to open the Edit Parametric Curve. Ignore the form that pops up first by pressing the OK button. See Figure 5-7.

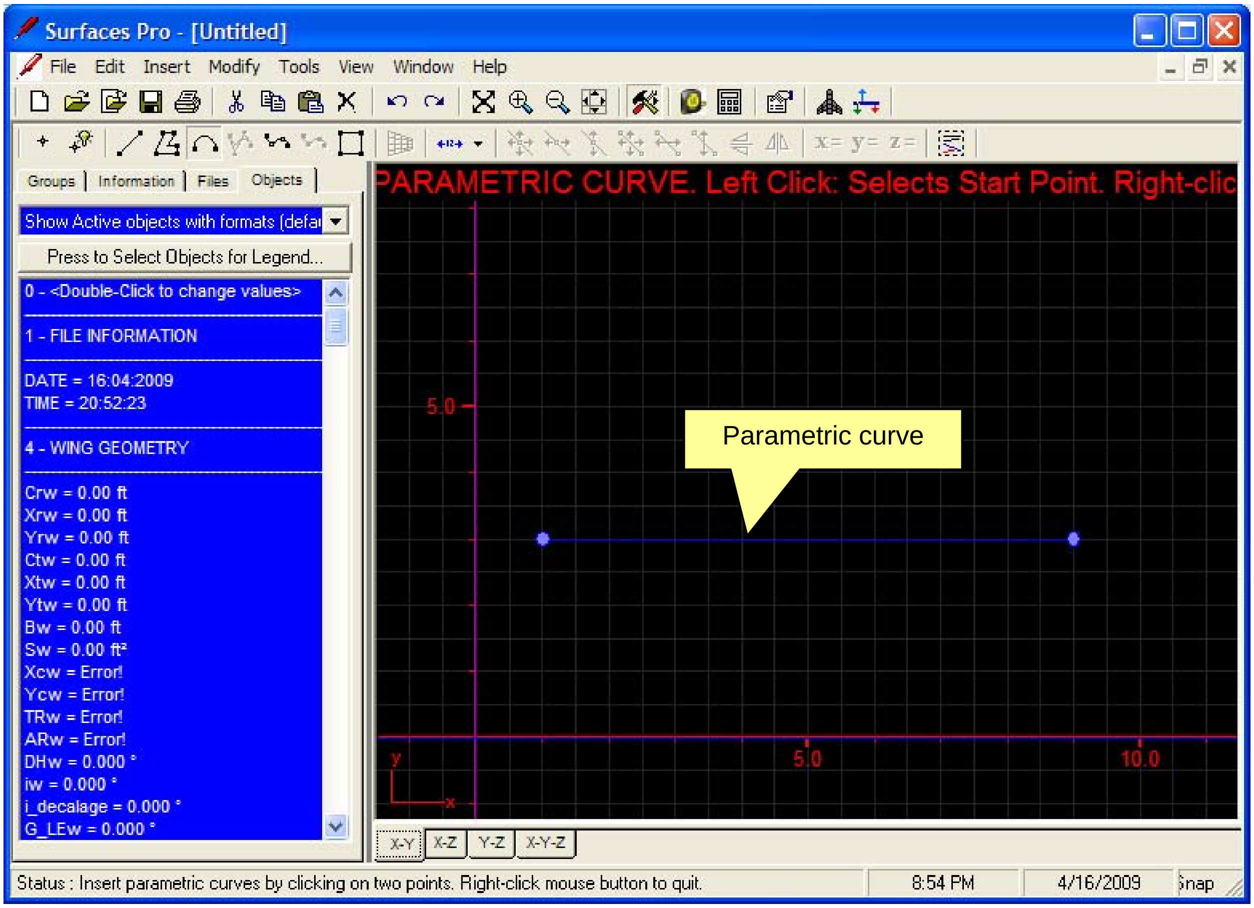


Figure 5-4: Creating a parametric curve.

Pay attention to the data in the form in Figure 5-7. You can see that the start point ID is 1 (point A), the end point ID is 2 (point B), and the alignment point ID is 3 (point C).

STEP 6: In the textbox under the "Parametric Functions" frame labeled P(t), enter the function: t-t*t. Note you must use the variable 't'. This is the parametric function SURFACES will use to compute the shape of the parametric curve. See Figure 5-7.

Press the Preview button to see what the curve looks like in 2-dimensions (see Figure 5-7). Note that the curve should consist of 30 points.

STEP 7: Press the OK button.

If you did everything correctly, you should see a curve identical to the one of Figure 5-8. Note how the curve has been drawn, aligned to a plane formed by two vectors; one extending from point A to B and the other from point A to C.

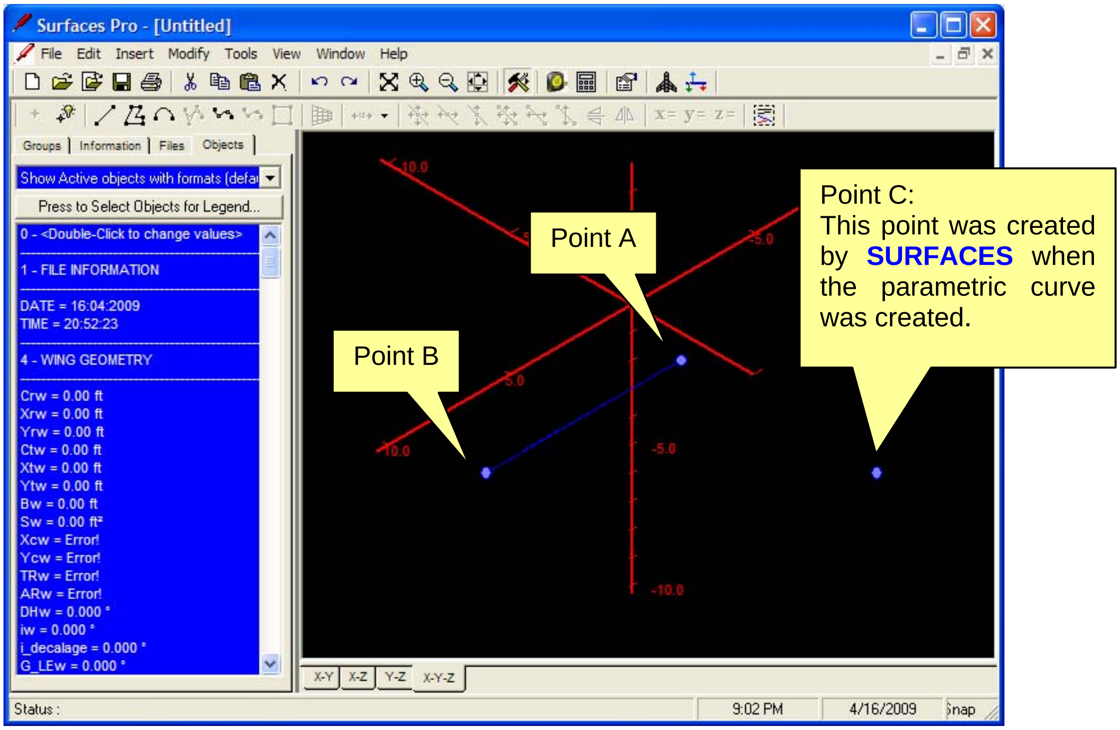


Figure 5-5: Points A, B, and C define the parametric curve.

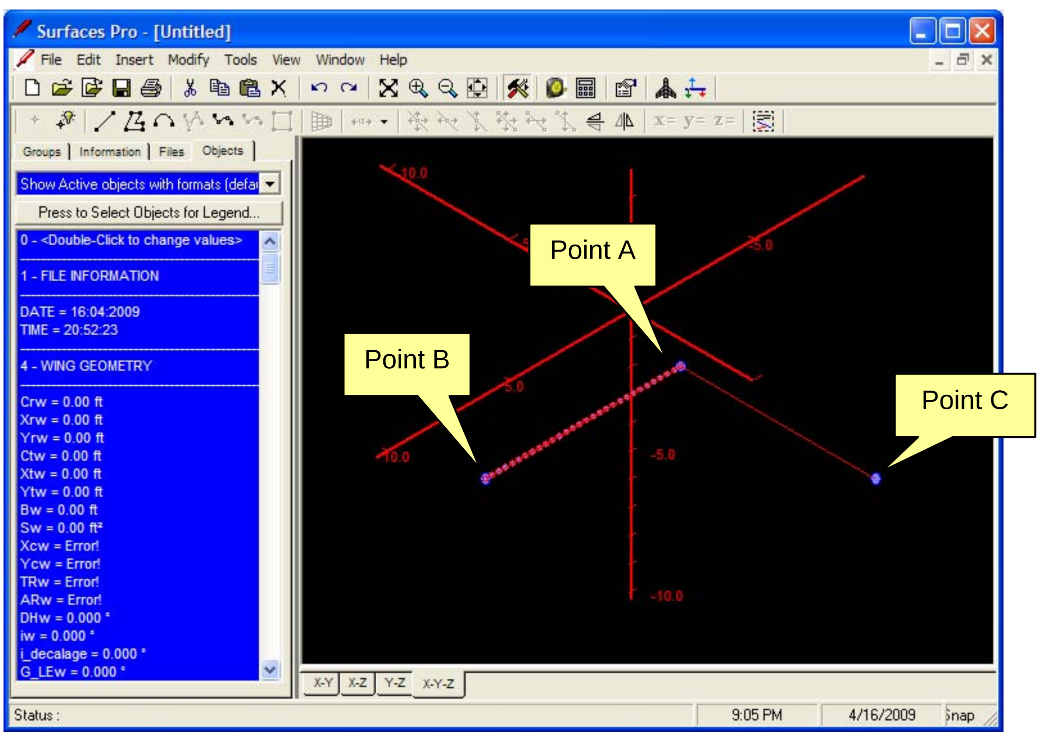


Figure 5-6: Selecting the parametric curve displays how SURFACES uses points to define a plane.

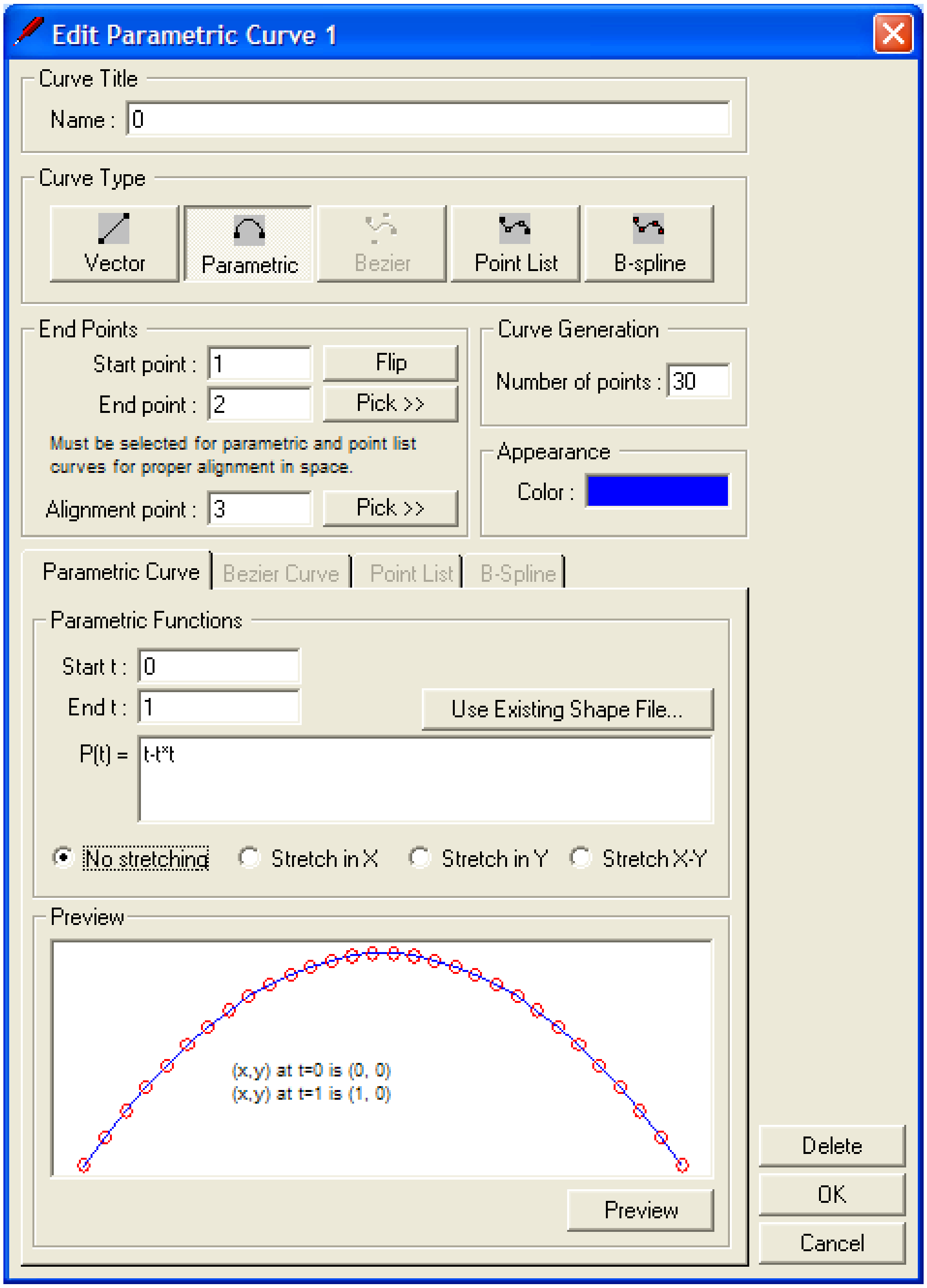


Figure 5-7: Creating a parametric curve.

STEP 8: Double-click on point C and change its Z-value from 0 to 6. Press the Apply button. The resulting orientation can be seen in Figure 5-8.

Re-orient the image (CTRL+ mouse center button) to see how the airfoil is still being drawn in the plane formed by the three points. Now, let us align the curve so it is parallel to the X-Z plane. This is done in Step 9:

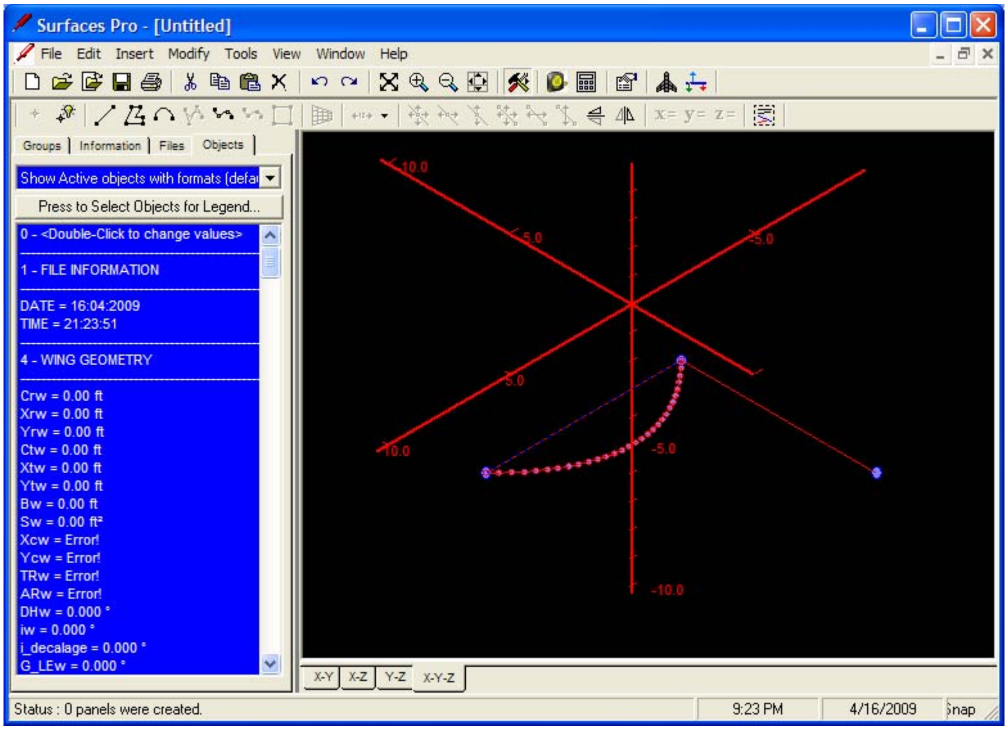


Figure 5-8: The parametric curve t-t² shown as originally created in the X-Y plane.

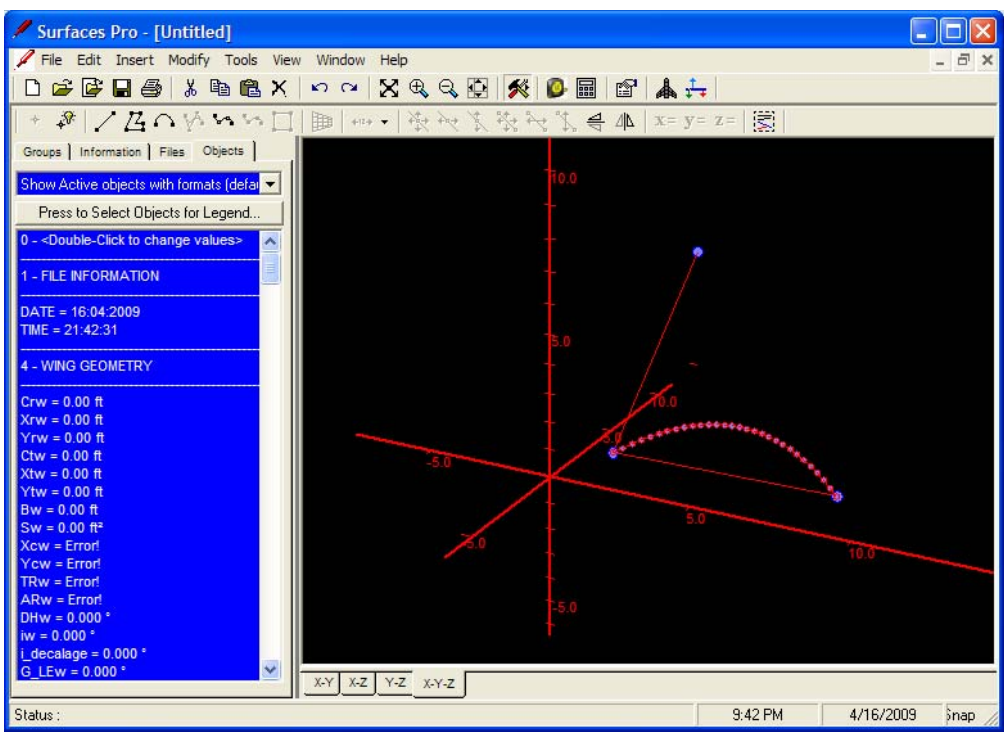


Figure 5-9: The parametric curve t-t² shown at an angle.

STEP 9: Double-click on point C and change its Y-value to 3. Press the Apply button. The resulting orientation can be seen in Figure 5-9.

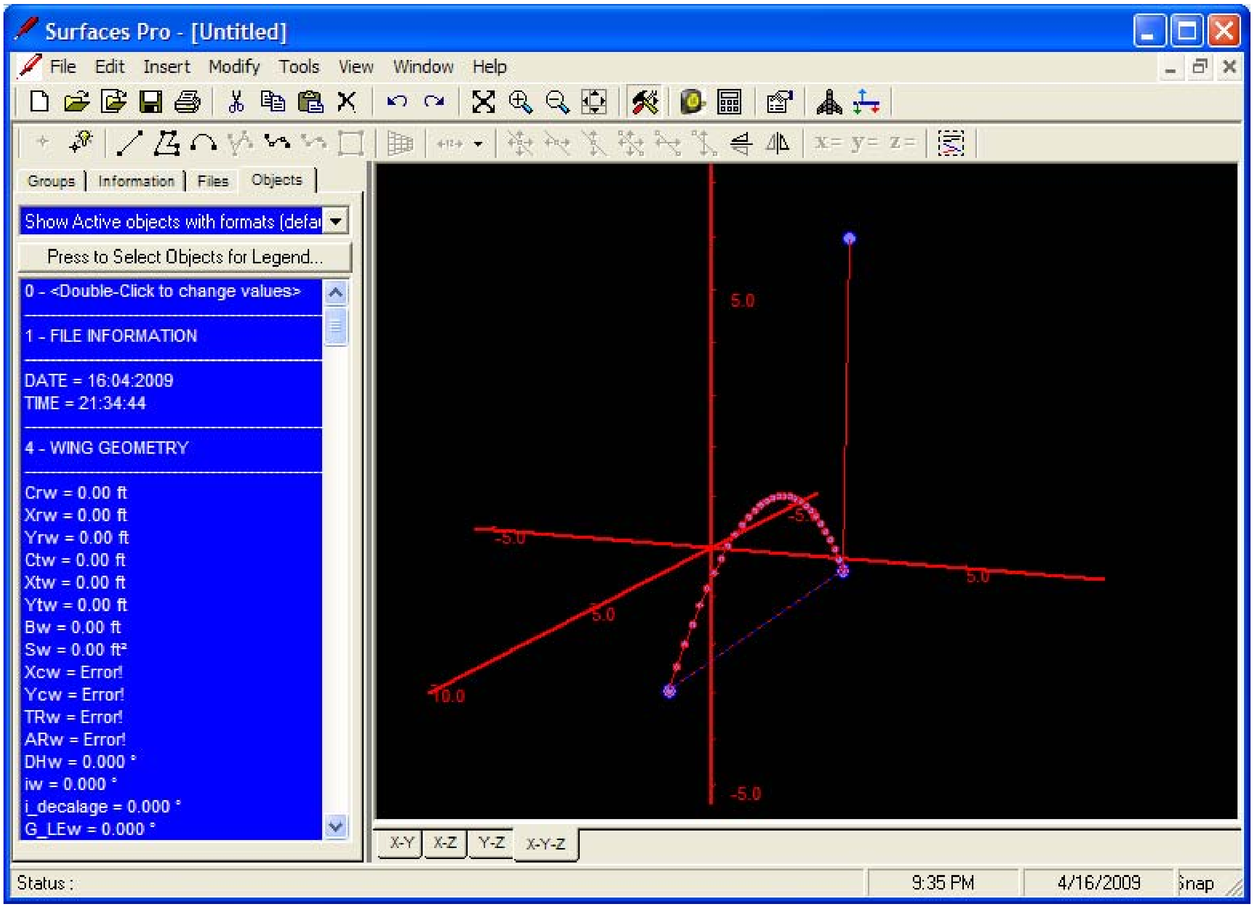


Figure 5-10: The parametric curve t-t² shown parallel to the X-Z plane.

Note how the curve is always drawn, as if on an imaginary 2D plane that is oriented in 3D space.

6. Transformation of Load Vectors from a Global to a Local Coordinate System

The following derives mathematical formulation to determine shear forces and moments about an arbitrary axis. The goal is to provide **SURFACES** with a tool that helps the structural analyst retrieve aerodynamic loads. However, the formulation is in fact applicable to any load analysis involving a discrete distribution of elemental loads.

Consider a lifting surface in a 3D coordinate system (from now on referred to as the **global coordinate system**). For structural purposes it is desired to determine the shear and moments about an axis, called the *quarter chord*. **SURFACES** allows this to be done quickly and effectively. The analysis requires a coordinate system to be constructed, which from now on referred to as the **local coordinate system**.

A more descriptive example of this is shown with the typical Vortex-Lattice model in Figure 1. A vector on the leading edge and along the fuselage have been highlighted (in red). Additionally, the right wing has been highlighted. With this information, it is now possible to determine the 3D shear and moment distribution along either vector, due to the discrete elemental forces generated by the right wing. The two vectors are necessary to create the **local coordinate system** about which the shear and moments are resolved. Consequently, they are referred to as the *basis* of the local coordinate system. This way, one can analyze loads along vectors of arbitrary orientation.

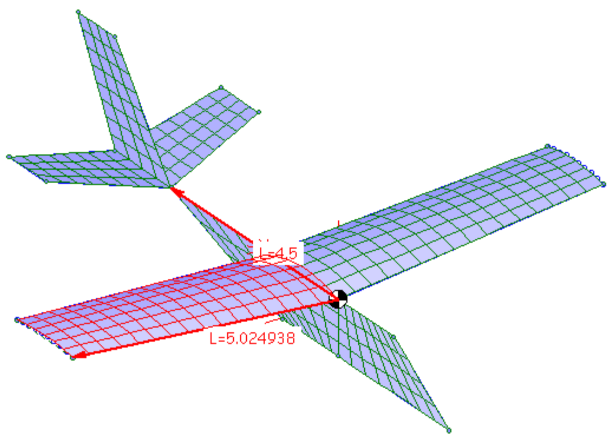


Figure 1: A typical Vortex-Lattice model.

6.1 Establishment of a Local Coordinate System

Consider the force F generated by an arbitrary panel in the global coordinate system X-Y-Z as shown in Figure 2.

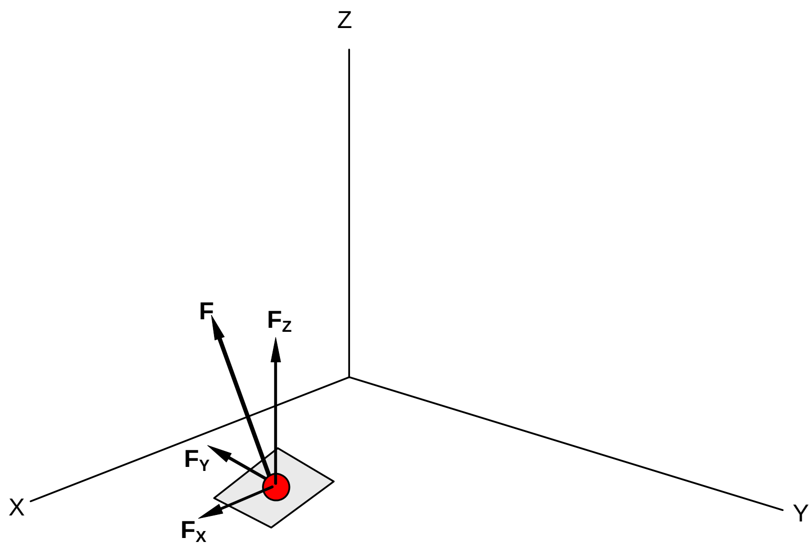


Figure 2: A force in the global coordinate system.

Consider a local coordinate system, identified by the selection of two vectors, **A** and **B**, such that **A** is not parallel to **B** (see Figure 3). These vectors uniquely define a plane (and are thus the basis of the coordinate system), whose normal is given by the vector **C**, such that:

$$\mathbf{C} = \mathbf{A} \times \mathbf{B} \tag{1}$$

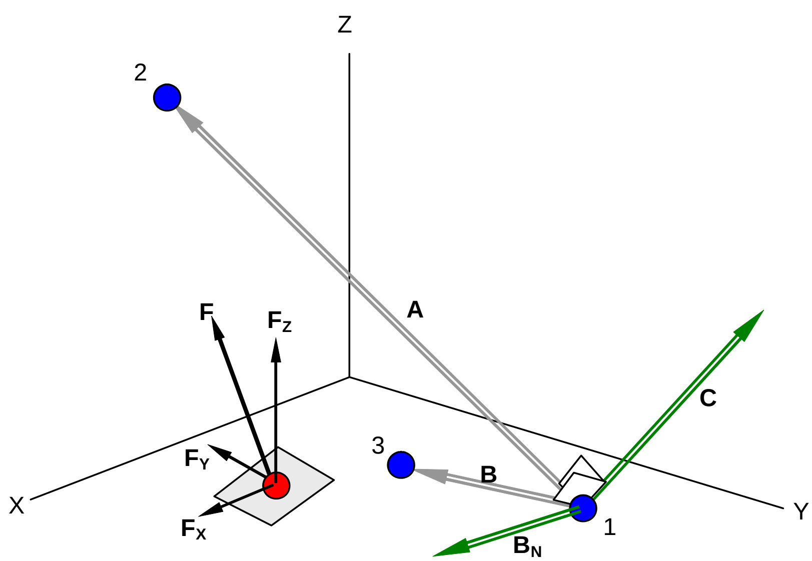


Figure 3: Defining the local coordinate system.

We can now create a local coordinate system, denoted by the vectors **A**, $\mathbf{B_N}$, and **C**, where $\mathbf{B_N}$ is given by

$$\mathbf{B}_\mathrm{N} = \mathbf{A} \times \mathbf{C} \tag{2}$$

Note that the three vectors form a mutually perpendicular coordinate system. The determination of $\mathbf{B_N}$ is necessary as **B** may or may not be perpendicular to the vector **A**.

Also note that according to convention, the vector **A** represents the *X*-axis of the local coordinate system, here denoted by the lower case letters *x-y-z*. The vectors $\mathbf{B_N}$ and **C** correspond to the *Y* and *Z* axes, respectively.

Finally, note that the unit vectors for the local coordinate system are denoted as follows:

Unit vector for A: $\begin{bmatrix} u_{AX} & u_{AY} & u_{AZ} \end{bmatrix}$

Unit vector for B: $\begin{bmatrix} u_{BX} & u_{BY} & u_{BZ} \end{bmatrix}$

Unit vector for C: $\begin{bmatrix} u_{CX} & u_{CY} & u_{CZ} \end{bmatrix}$

6.2 Transformation of Force Vector in Coordinate System A-B_N-C

The force vector, **F**, represented as {F_X, F_Y, F_Z} or $\mathrm{F_X} \cdot i + \mathrm{F_Y} \cdot j + \mathrm{F_Z} \cdot k$ in the global coordinate system can now be represented as a force in the local one as {F_x, F_y, F_z} (see Figure 4). This is accomplished with a simple transformation of the vector **F** onto the three vectors **A**, $\mathbf{B_N}$, and **C** using the matrix notation of Equation (3).

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \begin{bmatrix} u_{AX} & u_{AY} & u_{AZ} \\ u_{BX} & u_{BY} & u_{AZ} \\ u_{CX} & u_{CY} & u_{AZ} \end{bmatrix} \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix} \tag{3}$$

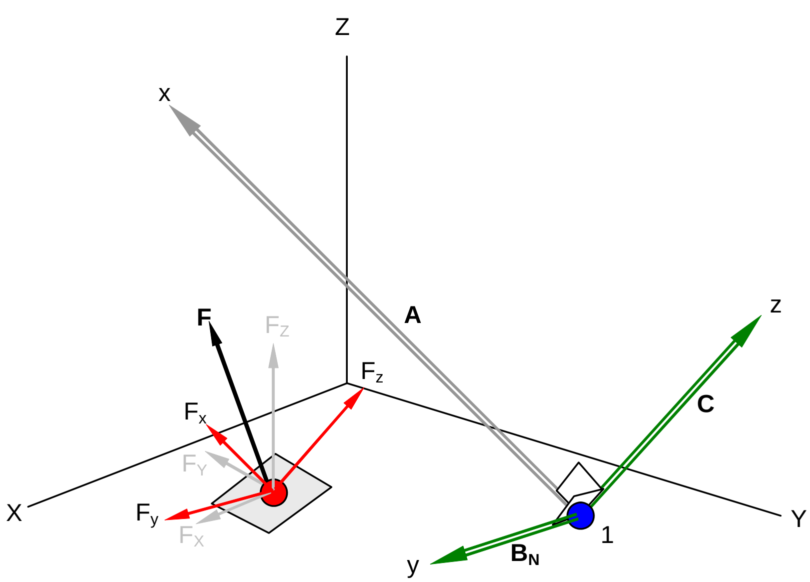


Figure 4: Transformation of vector **F**.

Example:

The force vector $\mathbf{F} = \mathrm{F_x} \cdot i + \mathrm{F_y} \cdot j + \mathrm{F_z} \cdot k = 10 \cdot i - 5 \cdot j + 10 \cdot k$ is given in a global coordinate system. Two vectors **A** and **B** are given as the basis for our local coordinate system as follows:

$$\mathbf{A} = -i - j + k$$
$$\mathbf{B} = 0.5i - j + 0.5k$$

Determine the components of F in the local coordinate system created by the vectors **A** and **B**.

Solution:

Step 1: Determine the vector **C** from $\mathbf{C} = \mathbf{A} \times \mathbf{B}$.

$$\mathbf{C} = \mathbf{A} \times \mathbf{B} = \begin{vmatrix} i & j & k \\ -1 & -1 & 1 \\ 0.5 & -1 & 0.5 \end{vmatrix} = 0.5i + j + 1.5k$$

Step 2: Determine the vector $\mathbf{B_N}$ from $\mathbf{B_N} = \mathbf{A} \times \mathbf{C}$.

$$\mathbf{B_N} = \mathbf{A} \times \mathbf{C} = \begin{vmatrix} i & j & k \\ -1 & -1 & 1 \\ 0.5 & 1 & 1.5 \end{vmatrix} = -2.5i + 2j - 0.5k$$

Step 3: Determine force component per Equation (5). Start by determining the unit vectors and assemble into the transformation matrix:

$$\begin{bmatrix} u_{AX} & u_{AY} & u_{AZ} \\ u_{BX} & u_{BY} & u_{AZ} \\ u_{CX} & u_{CY} & u_{AZ} \end{bmatrix} = \begin{bmatrix} -0.57735 & -0.57735 & 0.57735 \\ -0.77152 & 0.61721 & -0.15430 \\ 0.26726 & 0.53452 & 0.80178 \end{bmatrix}$$

This yields the following force components using Equation (3):

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \begin{bmatrix} -0.57735 & -0.57735 & 0.57735 \\ -0.77152 & 0.61721 & -0.15430 \\ 0.26726 & 0.53452 & 0.80178 \end{bmatrix} \begin{Bmatrix} 10 \\ -5 \\ 10 \end{Bmatrix} = \begin{Bmatrix} 2.8868 \\ -12.3443 \\ 8.0178 \end{Bmatrix}$$

6.3 Determination of Moment Vector in Coordinate System A-B_N-C

As stated in the introduction, ultimately, the goal of the analysis presented herein is the determination of shear forces and moments about an axis due to the cumulative effects of multiple discrete forces. It was demonstrated in Section 2 how shear forces are transformed to a local coordinate system. The same methodology can be applied to the generation of moments, but it involves a tranformation about a point, P, through which the vector **A** goes. This point will be called the *projection point* from now on. It is the projection of the point (x_F, y_F, z_F) on to the vector **A** (see Figure 6). It is denoted by the point (x_P, y_P, z_P).

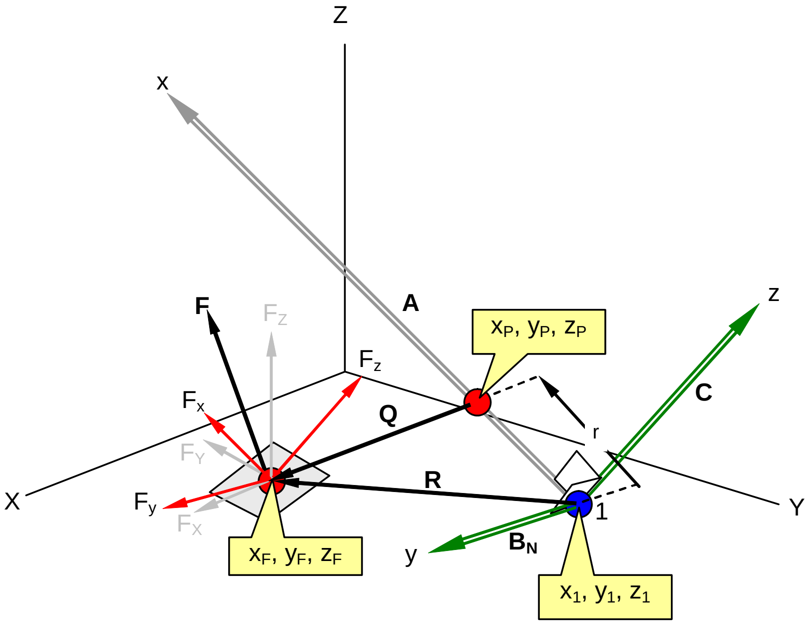


Figure 6: Determination of moment vector **M**.

The location of this point is obtained using standard vector algebra. The reader is referred to the one presented on page 31 in *Introduction to Vector Analysis*, by Davis and Snyder. The method can be explained using Figure 7, which defines the arbitrary vectors V and W.

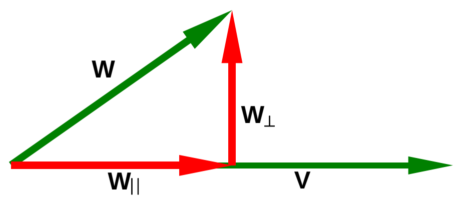


Figure 7: Projection of vector **W** onto vector **V**.

Then, the parallel projection of W onto V is given by:

$$\mathbf{W}_{\parallel} = \left(\frac{\mathbf{V} \cdot \mathbf{W}}{\mathbf{V} \cdot \mathbf{V}} \right) \mathbf{V} \qquad (4)$$

The perpendicular projection is simply found from:

$$\mathbf{W}_{\perp} = \mathbf{W} - \mathbf{W}_{\parallel} \qquad (5)$$

Using this, we first determine the vector **R** from the start point of the vector **A** to the force point, i.e.:

$$\mathbf{R} = \begin{Bmatrix} x_F - x_1 \\ y_F - y_1 \\ z_F - z_1 \end{Bmatrix} \tag{6}$$

The location is then found by referencing Figure 7 and Equation (4) and by writing:

$$\begin{Bmatrix} x_P \\ y_P \\ z_P \end{Bmatrix} = \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix} + \left(\frac{\mathbf{A} \cdot \mathbf{R}}{\mathbf{A} \cdot \mathbf{A}} \right) \mathbf{A} \tag{7}$$

The length of the parallel projection (the rightmost term of Equation (7)) is denoted by the letter *r*. It will be used in Section 4 to sort the discrete loads and moments along the vector **A**. Now, one must determine the vector from the projection point to the force point, denoted by **Q**. This vector is given by Equation (8):

$$\mathbf{Q} = \begin{Bmatrix} x_F - x_p \\ y_F - y_p \\ z_F - z_p \end{Bmatrix} \tag{8}$$

Then, calculate the disrete moment about the projection point from

$$\mathbf{M} = \mathbf{F} \times \mathbf{Q} = \begin{vmatrix} i & j & k \\ \mathrm{F_x} & \mathrm{F_y} & \mathrm{F_z} \\ \mathrm{Q_x} & \mathrm{Q_y} & \mathrm{Q_z} \end{vmatrix} \tag{9}$$

The moment vector, **M**, represented as $\{M_X, M_Y, M_Z\}$ is still in the global coordinate system. It can now be treated as the force in the local one, i.e. as $\{\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z\}$ using the same transformation as for the force vector.

$$\begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix} = \begin{bmatrix} u_{AX} & u_{AY} & u_{AZ} \\ u_{BX} & u_{BY} & u_{AZ} \\ u_{CX} & u_{CY} & u_{AZ} \end{bmatrix} \begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix} \tag{10}$$

6.4 Determination of Shear and Moment Distribution

Figure 7 shows several loads whose components have been transformed to the local coordinate system specified by **A**, $\mathbf{B_N}$, and **C**. Each has associated force and moment components and the parameter r, which is simply the distance of the projection point from the starting point of vector A (point 1). The purpose of the parameter r is to allow sorting to take place (say from start towards the end of the vector A). The sorted components are then used to construct shear and moment diagrams in a standard fashion.

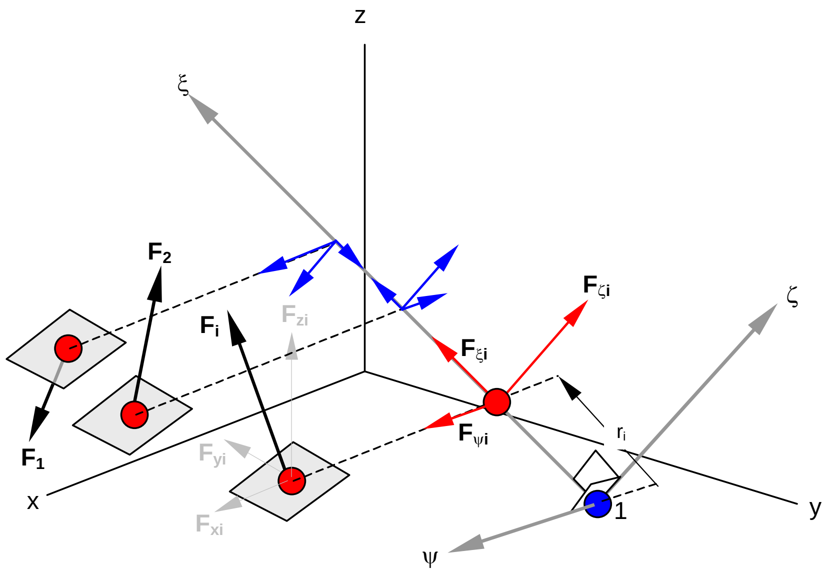


Figure 7: Methodology for construction shear and moment diagrams.

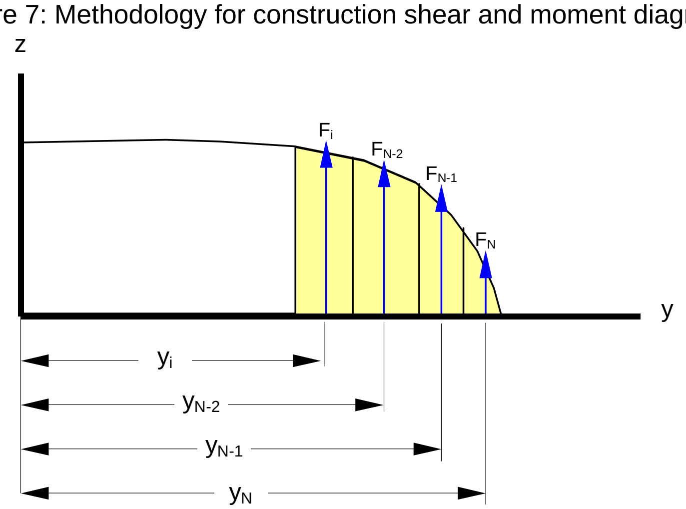


Figure 8: Creation of shear and moment diagrams from discrete forces.

6.4.1 Approximation for Shear in the Z-direction Along the Y-axis Vector

Approximating shear forces is simple, just apply Equation (11),

$$V_{zi} \approx \sum_{i=j}^{N} F_{zi} \tag{11}$$

6.4.2 Approximation for Moment about X-axis Along the Y-axis Vector

The approximation for the moments is implemented as follows. The moment at *N*-1 is due to the force F_{zN} acting at a distance $y_N - y_{N-1}$. Similarly, the moment at *N*-2 is due to the force F_{zN} acting at a distance $y_N - y_{N-2}$ and the force F_{zN-1} acting at a distance $y_{N-1} - y_{N-2}$. Writing this in a general form leads to:

$$M_{xi} \approx F_{zi+1} \cdot (y_{i+1} - y_i) + \ldots + F_{zN-2} \cdot (y_{N-2} - y_i) + F_{zN-1} \cdot (y_{N-1} - y_i) + F_{zN} \cdot (y_N - y_i)$$
$$= \sum_{j=i+1}^{N} F_{zj} \cdot (y_j - y_i) \tag{12}$$

6.4.3 Approximation for Torsion About the Y-axis Vector

The approximation for the torsion is implemented as follows. The torsion at *N*-1 is due to the force F_N acting at an offset distance of $x_N - x_{pN}$, where x_p denotes the x value of the projection point. Similarly, the moment at point *N*-1 is due to the force F_{zN} acting at a distance $x_N - x_{pN}$ and the force F_{zN-1} acting at a distance $x_{N-1} - x_{pN-1}$. Writing this in a general form leads to:

$$M_{yi} \approx F_{zi} \cdot (x_i - x_{pi}) + \ldots + F_{zN-2} \cdot (x_{N-2} - x_{pN-2}) + F_{zN-1} \cdot (x_{N-1} - x_{pN-1}) + F_{zN} \cdot (x_N - x_{pN})$$
$$= \sum_{j=i}^{N} F_{zj} \cdot (x_j - x_{pj}) \tag{13}$$

Example:

A lifting surface is 10 ft long (span) and 2 ft wide (chord). It carries a uniform pressure load of 1 lb_f/ft^2. Determine the shear in the z-direction, moment about the x-axis, and torsion about the y-axis at y=0.5 ft, assuming the span to be partitioned into 10, 1 ft wide strips. Note that each strip will carry 2 lb_f of load.

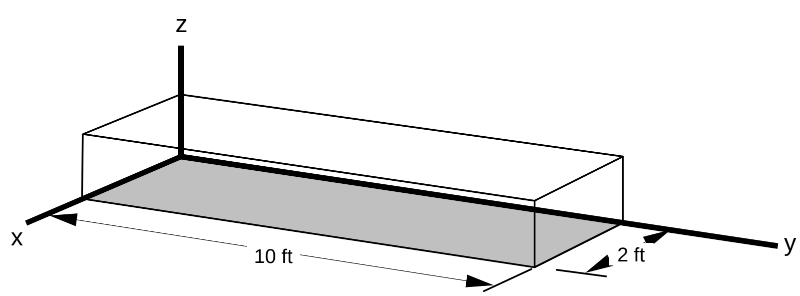


Figure 9: Lifting surface with a uniform pressure distribution.

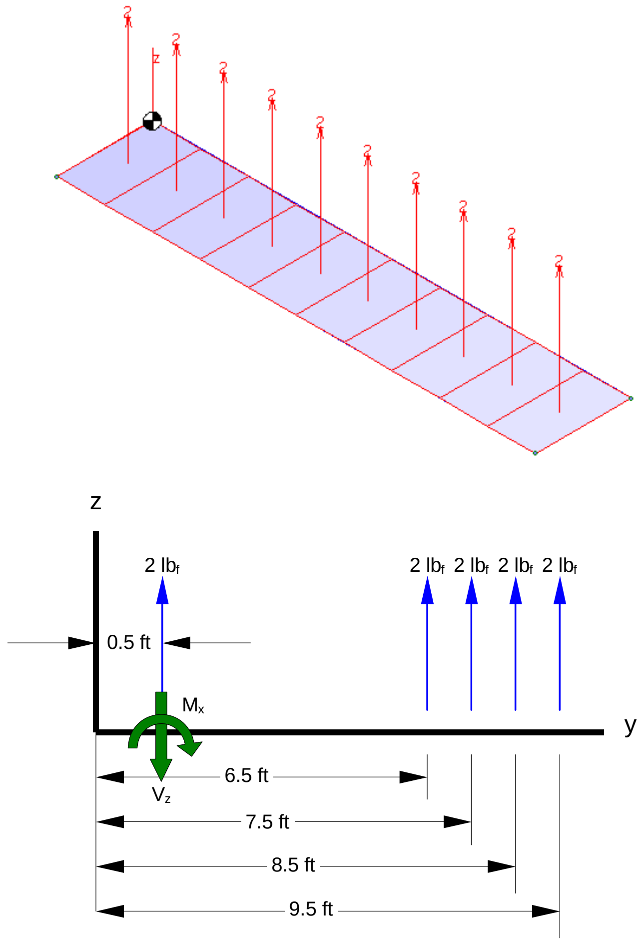


Figure 10: Discrete forces replace the uniform distribution. Reaction forces are shown in green.

Solution:

Shear is determined from Equation (11): $$V_i \approx \sum_{i=1}^{N} F_i = 2+2+...+2 = 20 \quad lb_f$$

Moment is determined from Equation (12):

$$\begin{aligned} M_x &\approx F_2 \cdot (y_2 - y_1) + ... + F_8 \cdot (y_8 - y_1) + F_9 \cdot (y_9 - y_1) + + F_{10} \cdot (y_{10} - y_1) \\ &= 2 \cdot (1) + ... + 2 \cdot (7) + 2 \cdot (8) + 2 \cdot (9) \\ &= 90 \;\; ft \cdot lb_f \end{aligned}$$

Torsion is determined from Equation (13):

$$
\begin{aligned}
M_y &\approx F_1 \cdot (x_1 - x_{p1}) + \ldots + F_8 \cdot (x_8 - x_{p8}) + F_9 \cdot (x_9 - x_{p9}) + + F_{10} \cdot (x_{10} - x_{p10}) \\
&= 2 \cdot (1) + \ldots + 2 \cdot (1) + 2 \cdot (1) + 2 \cdot (1) \\
&= 20 \;\; ft \cdot lb_f
\end{aligned}
$$

The exact value for the shear is determined from V = w·A = (1 lb_f/ft^2)·(10 ft x 2 ft) = 20 lb_f.

Similarly (noticing that the centroid of the force V is at y=5 ft), the moment about a point y = 0.5 ft (necessitated by the discreteness of the strip solution) is M_x = V·Δy = (20 lb_f)·(5 ft – 0.5 ft) = 90 ft·lb_f.

Finally, noticing the the centroid of the force V is at x=1 ft, we find that M_y = V·Δx = (20 lb_f)·(1 ft) = 20 ft·lb_f.

6.5 Presentation of Data in SURFACES

The user selects **Results->Force Integrator…** from the VLM Console in **SURFACES** as shown in Figure 11 below.

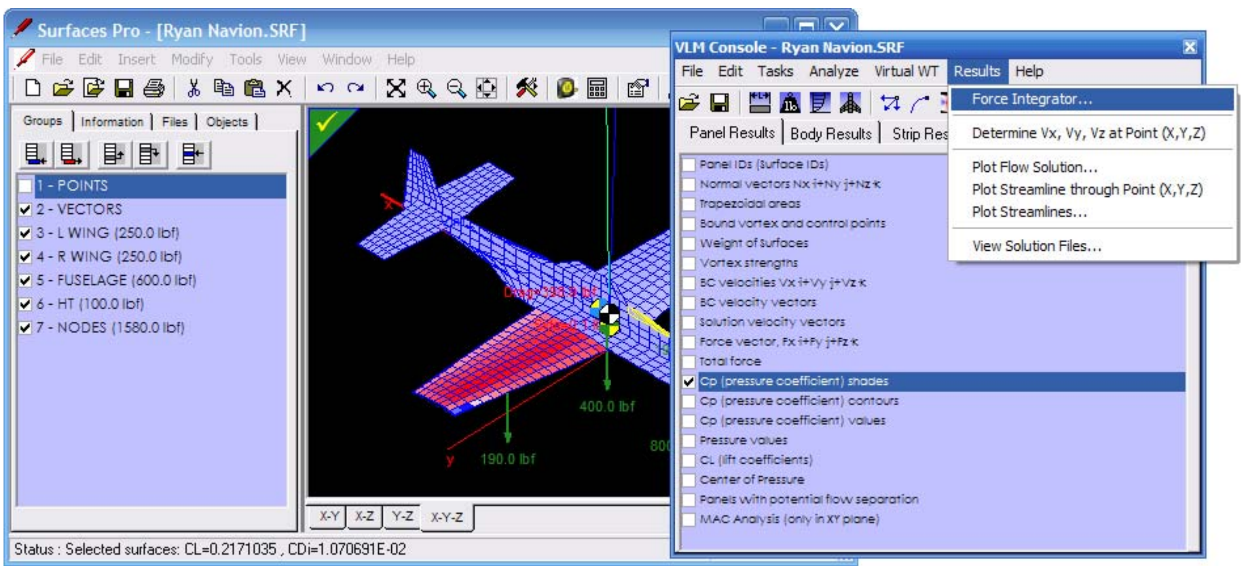


Figure 11: Selecting the Force Integrator tool.

Once the pertinent surfaces and vectors (corresponding to vectors **A** and **B**) have been selected, the user can press the *Integrate* button as shown in Figure 12. Selecting the Results tab will display a table with analysis results. Table 2 details the heading names.

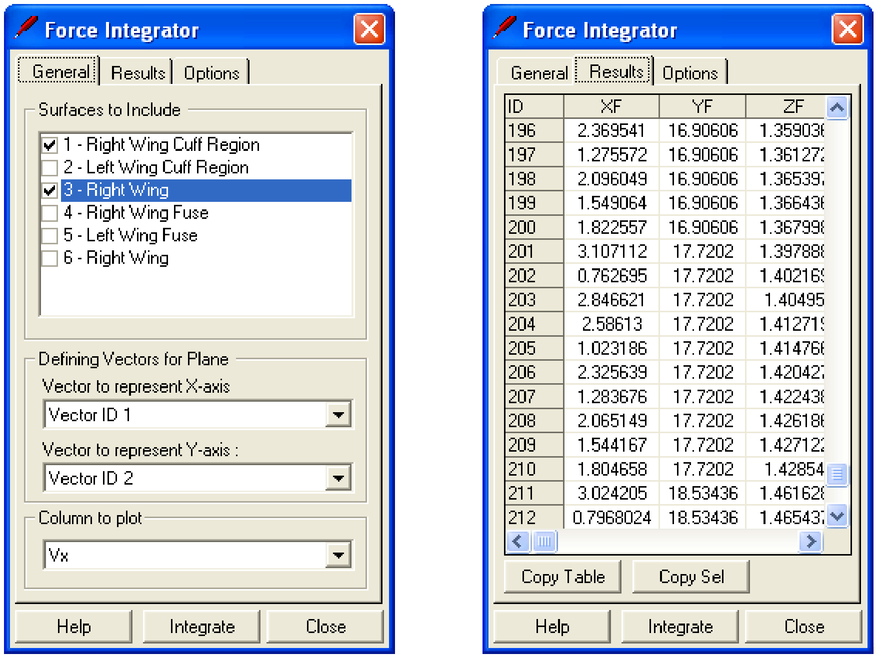


Figure 12: Force Integrator tool.

Table 2: Heading Names

Heading	Description
XF, YF, ZF	X, Y, and Z-coordinates of the panel force, which is its centroid.
XP, YP, ZP	X, Y, and Z-coordinates of the panel force panel force projection onto vector A.
r	Distance from Point 1 of vector A to XP,YP,ZP.
Rx, Ry, Rz	Components of the vector R, from Point 1 of vector A.
|R|	The length of vector R.
Qx, Qy, Qz	Components of the vector Q, from XP, YP, ZP to XF, YF, ZF.
|Q|	The length of vector Q.

```
'Panel force (body system) in global coordinate system
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Fbx"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Fby"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Fbz"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "|Fb|"
'Panel force (airspeed system) in global coordinate system
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Fx"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Fy"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Fz"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "|F|"
'Panel moment in global coordinate system
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Mx"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "My"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Mz"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "|M|"
'Panel force in global coordinate system
```

```
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Ftx"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Fty"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Ftz"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "|Ft|"
'Panel moment in global coordinate system
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Mtx"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Mty"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Mtz"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "|Mt|"
'Panel force in local coordinate system
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Vx"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Vy"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Vz"
'Panel moment in local coordinate system
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Tx"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Ty"
i = i + 1: gridCntrl.Col = i: gridCntrl.Text = "Tz"
```

7. Using the Virtual Wind Tunnel

The Virtual Wind Tunnel (VWT) allows you to analyze you model exactly as if you were to run it in a real wind tunnel. You can vary several parameters from an initial value to a final value in prescribed steps. For instance, you can perform an *alpha*- or a *beta*-sweep, exactly as you would do it in a real tunnel, but without the hassle. Before you use the tunnel, you must understand its limitations.

Any good airplane design operates most of its lifetime at airspeeds at which the airflow is relatively smooth and at a low angles-of-attack (AOA) and yaw (AOY). The lifting surfaces are always sized such that this is achievable. The primary advantage is that drag is minimum at such conditions and, therefore, the airplane is the most efficient. Under these circumstances, forces and moments change linearly with these angles. However, when the airplane slows down before it lands, or for some other extreme maneuvering, it begins to operate at larger AOAs and AOYs, causing the flow to separate. This will introduce a nonlinearity into forces and moments. Linear codes, including **SURFACES**, do not account for this phenomena.

At this point you may be asking yourself, why then resort to linear analysis if it has this shortcoming? The answer is as simple as it is resounding. Speed! Accuracy is an additional benefit if your model is well created. But the primary reason is speed. Linear analysis is extremely fast when compared to nonlinear analysis. At the time of this writing, using **SURFACES** one can create and analyze an aircraft in the linear range with an incredible accuracy in a matter of minutes. The same model may take 4-6 weeks to prepare for a nonlinear Navier-Stokes solver, and would give one (yes one) AOA, say every 24 hours, if one's computer network holds up. And, you should ask the question; But isn't the Navier-Stokes (N-S) method more accurate? The answer is yes and no. In fact, in the linear range, it will give a similar answer as the Vortex-Lattice Method (VLM), it will just take much, much longer to get those answers. The person writing these words has experienced many times that the VLM has been closer to actual wind tunnel data than N-S. The strength of N-S solvers is separated flow, but at this time, such tools are better at giving the aerodynamicist an idea of what the flow field looks like than trustworthy coefficients.

Naturally, it must be emphasized that **SURFACES** is performing a mathematical simulation when you use its wind tunnel test tool. The same rule applied to all computer codes that emulate wind tunnels; a real wind tunnel test always overrides any such calculations (assuming the data was obtained by reliable means). However, assuming you are using **SURFACES** to create a mathematical model of your design, the VWT is a great tool to help you understand the following issues:

(1) The AOA and AOY, the airspeeds, and the rotation rates (P, Q, R) where your math model breaks down. You will want to know at which AOA the linear assumption breaks down.
(2) Features of your model that, well, still need to be improved before an accurate comparison can be made of existing wind tunnel. The concept of tuning is well known in the world of finite elements, flutter, and linear modeling and the like. Tuning is done by making minor changes to the model un

8. Determination of a Trimmed Flight Condition

The following derivation details the requirements for a trimmed flight condition. A trimmed flight is defined as a flight in which the moment about all three axes is zero. For instance, when determining longitudinal trim (assuming a solution can be found) the following must hold:

$$\begin{aligned} C_L &= C_{L0} + C_{L\alpha} \cdot \alpha + C_{L\delta_e} \cdot \delta_e \\ C_m &= C_{m0} + C_{m\alpha} \cdot \alpha + C_{m\delta_e} \cdot \delta_e \end{aligned} \qquad (1)$$

Where; C_L = lift coefficient
C_{L0} = lift coefficient for zero angle-of-attack and zero elevator deflection
$C_{L0\delta e}$ = lift coefficient for zero angle-of-attack
$C_{Ltarget}$ = lift coefficient obtained from the lift equation
$C_{L\alpha}$ = lift coefficient change with angle-of-attack
$C_{L\delta e}$ = lift coefficient change with elevator deflection
C_m = pitching moment coefficient
C_{m0} = pitching moment coefficient for zero angle-of-attack and zero elevator deflection
$C_{m0\delta e}$ = pitching coefficient for zero angle-of-attack
$C_{m\alpha}$ = pitching moment change with angle-of-attack
$C_{m\delta e}$ = pitching moment change with elevator deflection
α = angle-of-attack
α_{CURR} = current angle-of-attack
$\Delta\alpha$ = deviation from current angle-of-attack
δ_e = elevator deflection
$\Delta\delta_e$ = deviation elevator deflection

If the coefficient are known, we can write Equation (1) as follows:

$$\begin{aligned} C_{L\alpha} \cdot \alpha + C_{L\delta_e} \cdot \delta_e &= C_L - C_{L0} \\ C_{m\alpha} \cdot \alpha + C_{m\delta_e} \cdot \delta_e &= C_m - C_{m0} \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} C_{L\alpha} & C_{L\delta_e} \\ C_{m\alpha} & C_{m\delta_e} \end{bmatrix} \begin{Bmatrix} \alpha \\ \delta_e \end{Bmatrix} = \begin{Bmatrix} C_L - C_{L0} \\ C_m - C_{m0} \end{Bmatrix} \qquad (2)$$

The solution protocol is a follows:

STEP 1: Compute: $$C_{\mathrm{Ltarget}} = \frac{2W}{\rho V^2 S}$$

STEP 2: Establish a value for $\Delta\alpha$ and $\Delta\delta_e$. Set $\alpha = \alpha_{CURR} - \Delta\alpha$ and $\delta_e = 0$, to determine C_{L1} and C_{m1}.

STEP 3: Set $\alpha = \alpha_{CURR} + \Delta\alpha$ and $\delta_e = 0$, to determine C_{L2} and C_{m2}.

STEP 4: Compute $C_{L\alpha}$ and $C_{m\alpha}$ from:

$$\left.\begin{aligned} C_{L1} &= C_{L0} + C_{L\alpha} \cdot \alpha_1 \\ C_{L2} &= C_{L0} + C_{L\alpha} \cdot \alpha_2 \end{aligned}\right\} \quad \Rightarrow \quad C_{L\alpha} = \frac{C_{L2} - C_{L1}}{\alpha_2 - \alpha_1} \quad and \quad C_{L0} = C_{L2} - C_{L\alpha} \cdot \alpha_2$$

and $$\left.\begin{aligned} C_{m1} &= C_{m0} + C_{m\alpha}\cdot\alpha_1 \\ C_{m2} &= C_{m0} + C_{m\alpha}\cdot\alpha_2 \end{aligned}\right\} \Rightarrow \quad C_{m\alpha} = \frac{C_{m2}-C_{m1}}{\alpha_2-\alpha_1} \quad \text{and} \quad C_{m0} = C_{m2} - C_{m\alpha}\cdot\alpha_2$$

STEP 5: Compute: $$\alpha_{\text{target}} = \frac{C_{L\text{target}} - C_{L0}}{C_{L_\alpha}}$$

STEP 6: Set $\alpha = \alpha_{\text{target}}$ and $\delta_e = \delta_{e\ \text{CURR}} - \Delta\delta_e$, to determine C_{L3} and C_{m3}.

STEP 7: Set $\alpha = \alpha_{\text{target}}$ and $\delta_e = \delta_{e\ \text{CURR}} + \Delta\delta_e$, to determine C_{L4} and C_{m4}.

STEP 8: Compute $C_{L\delta e}$ and $C_{m\delta e}$ from: $C_{L\delta_e} = \dfrac{C_{L3}-C_{L0}}{\delta_e}$ and $C_{m\delta_e} = \dfrac{C_{m3}-C_{m0}}{\delta_e}$

STEP 9: Compute the required C_L to support the desired lift and knowing that C_m=0 for a balanced condition we populate the matrix of Equation (2) as follows:

$$\begin{bmatrix} C_{L\alpha} & C_{L\delta_e} \\ C_{m\alpha} & C_{m\delta_e} \end{bmatrix} \begin{Bmatrix} \alpha \\ \delta_e \end{Bmatrix} = \begin{Bmatrix} C_L - C_{L0} \\ -C_{m0} \end{Bmatrix} \tag{3}$$

And solve for the α and δ_e, which define the trimmed condition.

SURFACES solves this using an iterative algorithm and can do so about each of the airplane's three axes. This is necessary because the deflection of a control surface modifies the geometry which, in turn, requires a new flow solution. The program comes with an easy to use Trim Wizard that makes this a breeze. Additionally, you can trim for multiple airspeeds, creating an individual flow solution for each trimmed condition. This is handy when you want multiple solutions for the same CG location. You can leave your computer overnight running the trim solutions, and study the solution files the next day.

9. Determination of Drag in SURFACES

9.1 Introduction

One of the primary advantages of using the Vortex-Lattice Method is speed and accuracy in the estimation of aerodynamic forces and moments. A prominent of those is drag. Since so many other factors rely on drag (performance, engine requirements, etc) any tool that allows for a quick and reliable estimation is priceless. Unfortunately, drag estimation is wrought with challenges.

There are several things that make drag remarkable as an aerodynamic force. Among those is how hard it is to accurately estimate its magnitude. Drag is a rapidly changing variable, making its estimation harder and harder as the angle-of-attack increases and air begins to separate and form "separation bubbles". Another challenge is the fact that when airspeed increases, compressibility effects contribute more and more to the total drag.

The shape of a properly designed airplane flying at a low angle-of-attack (high speed) is such that air flows over it smoothly and its drag is relatively low when compared to other flight conditions. Reducing the airspeed requires an increase in angle-of-attack, which eventually causes airflow to separate in various areas (e.g. along trailing edge of wings, fuselage wing juncture, etc.), increasing its drag. Such flow adds a considerable complexity to analysis work. In fact, it is so complex in nature that even state of the art Navier-Stokes solvers have a hard time predicting it accurately. Extracting drag from wind tunnel testing presents challenges as well and requires great expertise, especially for scaled wind tunnel models. This is so, because the angle-of-attack at which flow separation begins differs from that of the full scale airplane. These difficulties must always kept in mind when predicting drag using any computer code. The calculation of drag is estimation only, and, as such, must be taken with a grain of salt. It is the purpose of this section to explain how **SURFACES** computes drag and, that way, help you make drag predictions that are as useful as possible.

> **IMPORTANT!**
>
> SURFACES is a symbolic vortex-lattice solver. It allows the user to create mathematical expressions, called *Math Objects* or *Variables* (which are used inter-changeably), that allow the designer to define own parameters that may be of importance to the airplane involved. This adds an incredible power to the analysis work. The math objects can use information directly from the geometry of your model. For instance, to calculate wing area you could enter a constant or you could use a function like [Saxy(surf1, surf2, …)]. So if you modify your wing area for some reason, the program will automatically update this value.

As is revealed in the famous Navier-Stokes equations, drag really has only two causes; *pressure* and *friction*, although the multitude of specialty drags that abound in aerospace engineering literature imply otherwise. The **SURFACES** development team uses these two drag sources to simplify drag estimation in the program. Drag estimation involves several parameters; the geometry of the exposed area (known as the wetted area), aircraft orientation (e.g. angle-of-attack and angle-of-yaw), and flow physics (density, airspeed, Reynolds Number and Mach Number). Mathematically, this is represented in the formula:

$$D = f(geometry, \alpha, \beta, \rho, V_\infty, \mathrm{Re}, M) \quad (1)$$

Where: *geometry* refers to reference and wetted area[1].
M = Mach Number. Stored in the variable `MN`.
Re = Reynolds Number. Stored in the variable `Re`.
V_∞ = Far-field airspeed. Stored in the variable `Vinf`.
α = Angle-of-attack. Stored in the variable `AOA`.

[1] In **SURFACES** geometry terms are stored in variables such as `ARref`, `Eref`, `Sref`, and `Swet`.

β = Angle-of-yaw. Stored in the variable `AOY`.
ρ = Air density. Stored in the variable `rho`.

The word geometry is somewhat nebulous here, but it is so on purpose; the user may use geometry in own drag estimation beyond the variables cited. Also, while most texts on the subject tend to neglect the contribution of the angle-of-yaw, β, this is not done here for two reasons: First, the user must be made aware of the impact asymmetric flight has on aircraft performance, especially when designing multi-engine aircraft for engine-out situations. Second, by using **SURFACES** this is simply no more complicated than accounting for angle-of-attack.

So, let's begin by writing a standard definition of the total drag force:

$$D = \tfrac{1}{2}\rho V_\infty^2 S_{ref} C_D \tag{2}$$

Where: C_D = Total drag coefficient, dimensionless. Stored in the variable `CD`.
D = Drag force in lb_f (UK system) or N (SI system).
S_{ref} = Reference area, typically in ft² or m².
V_∞ = Far-field airspeed, typically in ft/s or m/s.
ρ = Air density, typically in slugs/ft^3 or kg/m^3. Stored in the variable `rho`.

Equation (2) explicitly contains three of the variables mentioned for Equation (1), namely; *geometry*, ρ, and V_∞. Dependency on α, β, M, and Re is usually handled in the expression for drag coefficient, C_D. In aircraft design, aerodynamicists typically regard the drag coefficient as a function of the lift coefficient, C_L and plot the two on a graph called the *drag polar*. A typical representation of airfoil data is shown in Figure 9.1-1[2]. This shows a lift curve, drag polar, and pitching moment curves for several 2D airfoils and shows two graphs. The left graph shows how the lift coefficient varies with angle-of-attack. The right one shows how the drag coefficient varies with the lift coefficient. Note that the pitching moment coefficients are not important in this discussion.

The shape of the drag polar depends on several factors. The first is lift, which depends on the angle-of-attack (and yaw) of the geometry. It is also evident that the C_D is always larger than zero, achieving a certain minimum value at relatively low values of C_L. It follows it makes sense to consider the drag as the sum of some minimum drag, call it C_{Dmin}, and additional drag, caused in part by the change in C_L. This additional drag is caused by an increase in flow separation, which increases the pressure drag.

The dip in the drag polar around a C_L of 0.2 to 0.5 is referred to as a *drag bucket* and is typically associated with laminar flow airfoils. For instance, note how all but two of the airfoils in Figure 9.1-1 (64_2-415 and 23012) display this phenomenon. Exceeding this band of lift coefficients on either side, will result in a notable change in airflow behavior. First, the location where laminar boundary layer transitions into a turbulent one on the upper surface moves closer to the leading edge of the airfoil. Second, as the angle-of-attack increases more, flow begins to separate near the trailing edges of the wing. This change affects the distribution of pressure around the airfoil and, therefore, causes a rise in the pressure drag. By the same token, the transition point on the lower surface will move closer to the trailing edge. This changes the extent of laminar versus turbulent boundary layer and, therefore, changes the skin friction drag. This is the second factor to be considered.

The third factor is compressibility effects. This is a high speed phenomenon, but a simple explanation is that compressibility causes streamlines to align closer together and farther into the flow field than they do in an incompressible flow. This results in a higher speed over the airfoil than indicated by incompressibility, which increases the low-pressure on the airfoil and, thus, the rate at which both lift and

[2] Reproduced from NACA R-824.

drag change with angle-of-attack. Compressibility drag is exclusively a pressure drag effect[3] and eventually, if the airspeed increases further, a shockwave will form. **SURFACES** does not predict shockwave formation, so results in which shock would have formed in real flow are unreliable. Typically, shockwaves begin to form when airplanes fly at airspeeds faster than Mach 0.85, but may happen at a far lower airspeed, for instance if the airplane has thick wings. The theory of compressible flow is beyond the scope of this discussion, but the interested reader can refer to engineering texts such as References 2, 3, and 6 for further information. The user must be cognizant of such high speed effects. **SURFACES** has been designed to automatically include compressibility corrections if the user chooses to apply them. In **SURFACES**, you should apply compressibility corrections for cases when the airspeed exceeds Mach numbers of the order of 0.3 to 0.5.

SURFACES provides four different methods to model compressibility effects and, if selected, automatically computes their effects for the user. This will be talked about in greater detail shortly.

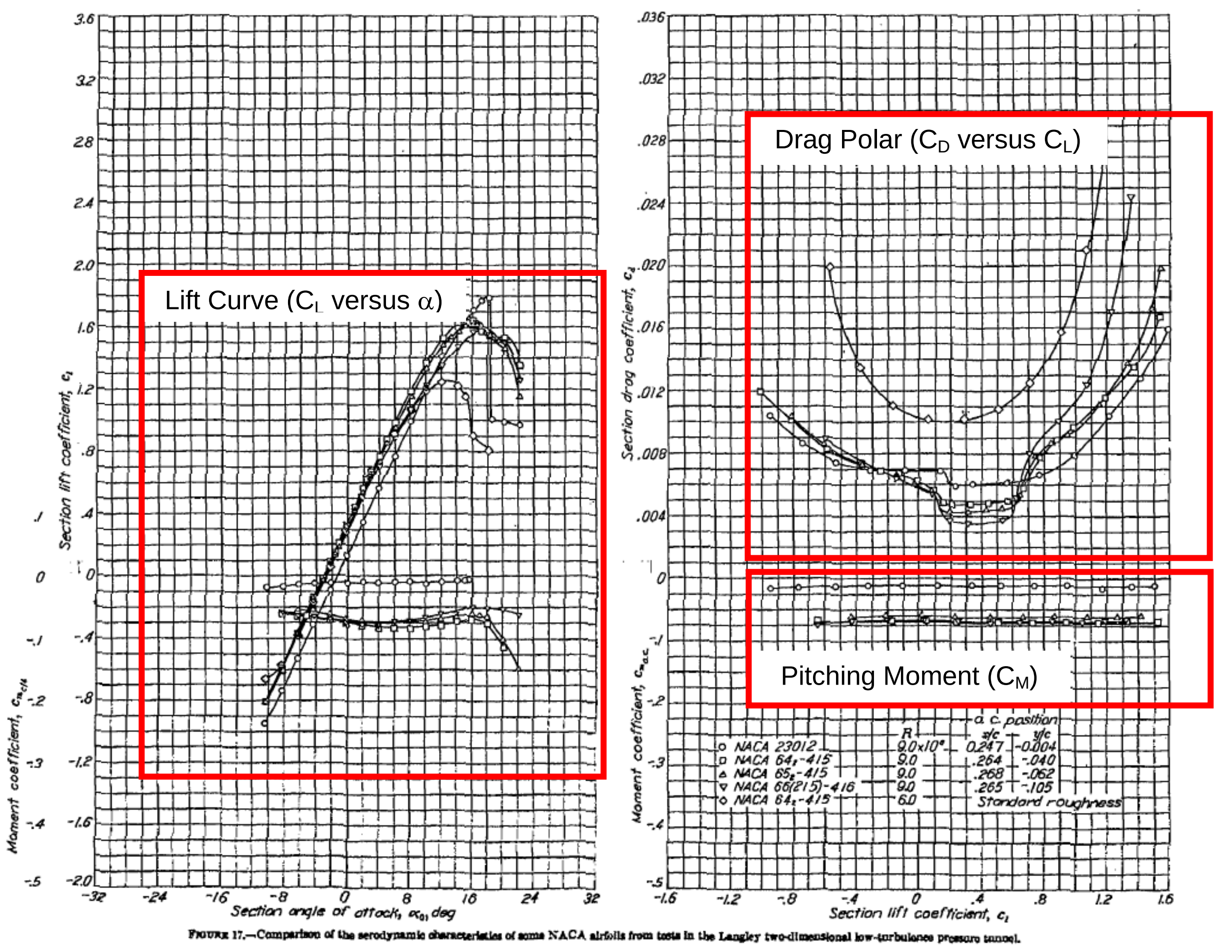


Figure 9.1-1: Drag polar for several 2D airfoils.

From this discussion it makes sense to define the drag coefficient as follows:

[3] For instance, see discussion in Aircraft Performance and Design, John D. Anderson, pages 115-116.

$$C_D = C_{Do} + C_{Df} + C_{Di} \qquad (3)$$

Where: C_D = Total drag coefficient, dimensionless. Stored in the variable `CD`.
C_{Do} = Basic drag coefficient, dimensionless. Stored in the variable `CDo`.
C_{Df} = Skin friction drag coefficient, dimensionless. Stored in the variable `CDf`.
C_{Di} = Induced drag coefficient, dimensionless. Stored in the variable `CDi`.

Note 1: The form of Equation (3) preserves the idea expressed in most texts on aircraft design.

Note 2: Since **SURFACES** is symbolic code, the user can enter complicated expressions for each component. However, **SURFACES** also provides the user with several tools to help and these will be discussed in greater detail in this section.

Note 3: Although many aerodynamic texts treat C_{Do} and C_{Df} as if they were constant with respect to α and β there is no guarantee this is true in reality. For instance, a change in α will move the laminar to turbulent flow transition point and reshape flow separation regions. Additionally, compressible skin friction coefficient reduces slightly with Mach Number, whereas the basic drag coefficient increases.

Note 4: Sometimes the basic drag coefficient is lumped together with the skin friction coefficient and called *profile drag*. This will not be done here for the simple reason that it adds complexity to keep track of yet another drag coefficient and hides the contribution of wetted area on the overall airplane drag.

Note 5: The effect of compressibility is accounted for by modifying C_{Do} and C_{Di} using corrections that pertain to pressure drag only, and using a correction only applicable to skin friction for C_{Df}.

Note 6: **SURFACES** has internal functions that calculate most of these coefficients for the user. The user must supply C_{Do} only, but the other coefficients can be calculated internally if the user so wishes. All can be displayed as math objects, using the functions `[CDf]` for skin friction, `[CDi]` for induced drag, `[CD]` for total drag (calculated per Equation (3)), and `[CL]` for lift coefficient. This is already set up in this fashion in the standard *Math Object template*. So, when a new project is created, the formulation is already correctly set up by default. Note that if these built-in functions are used, a C_{Di} and C_L of 0 will be reported when there is no Vortex-Lattice solution in memory, or if the user resets the solution (clears it out of memory). Also, a CDf of 0 will be reported until skin friction coefficient has been assigned to any of the surfaces.

IMPORTANT!

When you start a new project in SURFACES (by selecting **File->New…** and then Surfaces Worksheet), the program will load a standard list of math objects, so that you won't have to create commonly used variables each time. The program loads this from a template file called OBJECTTEMPLATE.INI, which is stored in the /SURFACES/TEMP/ folder. If you suspect this template file is corrupt or accidentally delete it, you can download a new one from:

www.greatowlpublishing.com.

Note 7: As said earlier, actual change in AOA or AOY <u>will</u> change C_{Do}, but this change is not to be confused with the change in induced drag, C_{Di}, whose magnitude depends on the lift coefficient, C_L. The change in C_{Do} is solely due to a change in pressure over the airplane, which is not used directly for lift generation (although those lines are blurred at times). It depends on the attitude of the airplane (i.e. angular orientation) in the air[4], but this affects the shape and size of flow separation regions. The C_{Di}, on the other hand, depends on the C_L. Induced drag can be defined as the drag created by a wing in excess of what it would create in an inviscid flow at the same C_L. One way the aerodynamicist can estimate a variation in C_{Do} with AOA and AOY is to wind tunnel test an aircraft with the lifting surfaces removed. See Note 9 for additional information.

[4] For instance, see page 186 of Reference 5.

Note 8: Figure 9.1-2 shows a schematic of how **SURFACES** handles drag calculations. First, incompressible drag coefficients are computed. Second, if compressibility correction is to be included the coefficients are modified. Third, the coefficients are added to return the total drag coefficient.

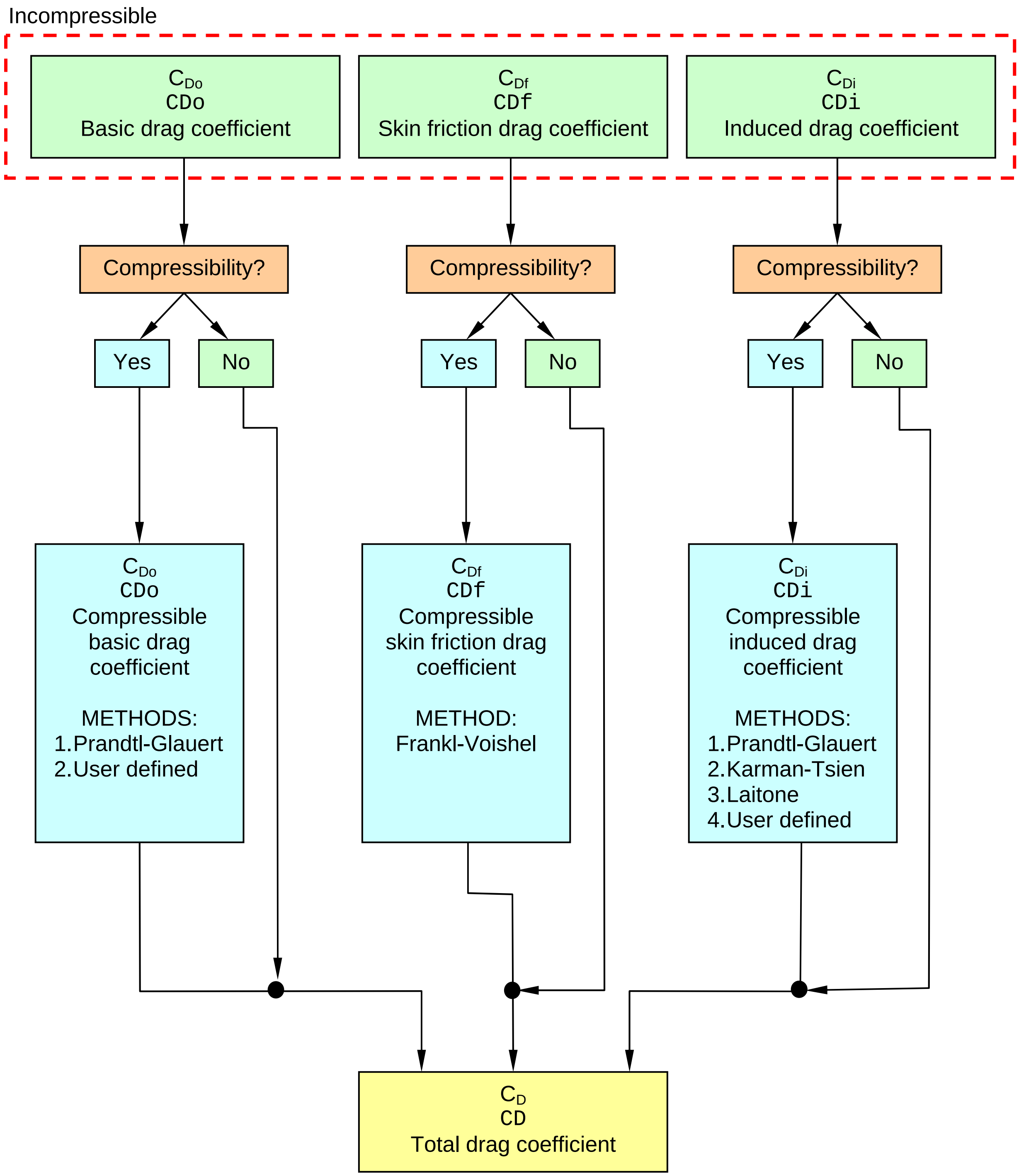


Figure 9.1-2: A schematic showing how SURFACES determines drag coefficients.

Note 9: Consider Figure 9.1-3, which shows a simplified example of how C_{Do}, C_{Df}, and C_{Di} might vary with angle of attack only (constant airspeed and altitude). In reality, C_{Do} might show a larger increase with AOA than displayed, especially at very low and very high AOA, and C_{Df} will likely change as well as the laminar and turbulent flow regions change, but one should be careful in assuming C_{Do} and C_{Df} remain constant. Figure 9.1-4 show how the same coefficients build up to form C_D.

Note 10: Aerospace engineering literature introduces the casual reader to an assortment of drag types. There is transonic drag, nacelle drag, external store drag, protuberance drag, interference drag, parasitic drag, leakage drag, just to name a few. At times it's not clear whether one is reading about aerospace or medical science. With that in mind, there are two points that must be emphasized: (A) Textbook authors are prolific inventors of terms for things that either increase pressure drag or skin friction drag, or a combination thereof. This leaves the impression that there exist imaginary drag types that only affect certain airplane features. Only airplanes with nacelles get nacelle drag, only airplanes with protuberances suffer from protuberance drag, and so on, when in fact these features are simply changing the pressure field or modifying the boundary layer. While there are probably many who consider this advantageous, this can also confuse the issue. The confusion does not stem from the names these specialty drags receive, but a difference in definition between authors, when one author creates a name for a specialty drag another author doesn't even mention. (B) **SURFACES** handles this assortment of drag types in a simple manner; it ignores them. It only uses the three terms in Equation (3) and leaves it to the user's to define as many drag terms as desired, naturally limited by computer resources only.

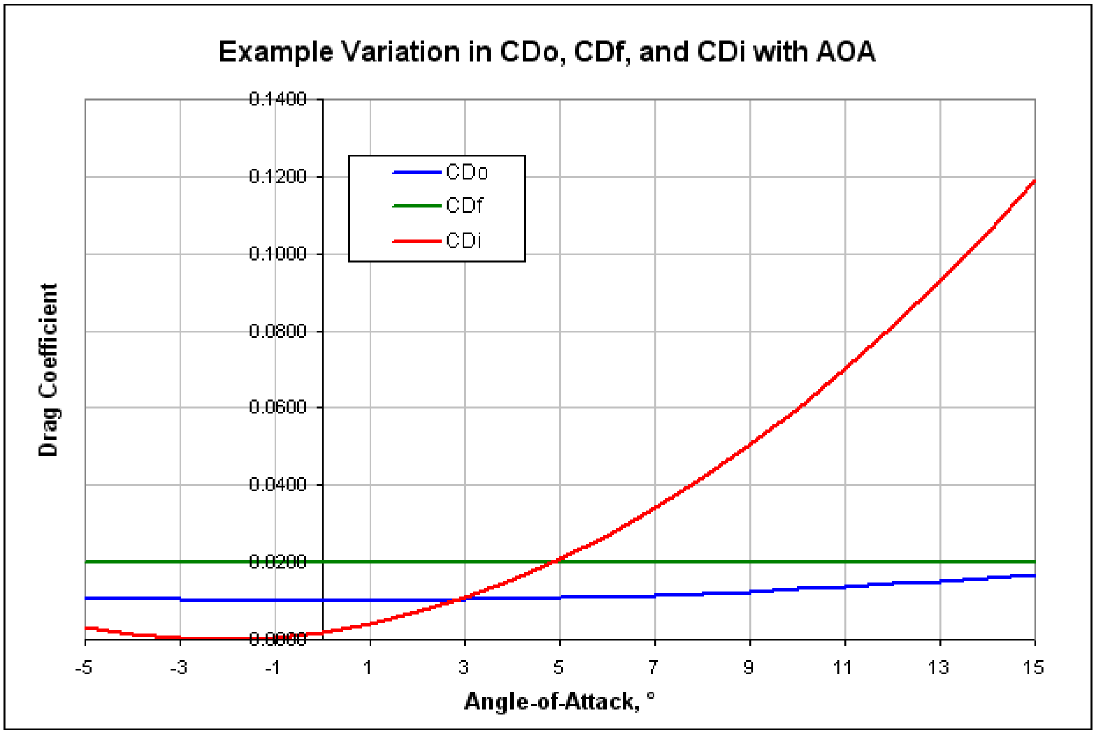


Figure 9.1-3: Basic drag coefficient plotted for AOA and AOY.

[NOTE THAT THIS APPLIES TO QUADRATIC DRAG MODEL ONLY]

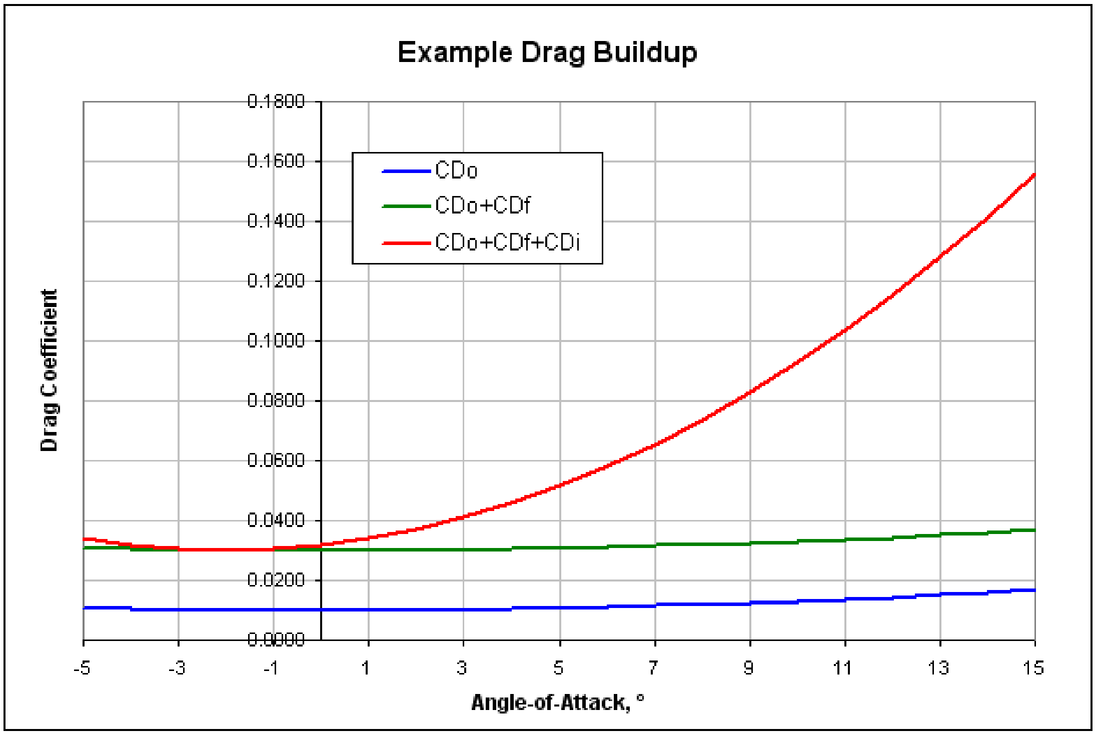


Figure 9.1-4: Basic drag coefficient plotted for AOA and AOY.

Now let's look at the three constituent drag coefficients in greater detail.

9.2 Basic Drag Coefficient, C_{Do}

Basic drag is caused by pressure differences integrated over the entire external surface of the aircraft and always results in a force that impedes its motion. It includes the effects of interference of major parts, such as fuselage and wing. It gets larger with increase in flow separation and, therefore, generally should not be considered constant, although many do so in interest of convenience, especially during early concept studies of new aircraft. **SURFACES** assumes this coefficient is supplied by the user and, therefore, the default value for every new project is 0. The coefficient is stored in the math object `CDo`. Table 9.2-1 shows some examples of possible user entries for C_{Do}. If compressibility modeling has been selected, the returned value is the compressible basic drag coefficient.

IMPORTANT!

Entry is accomplished through the math object editor, shown in Figure 9.2-1. This is opened by double-clicking on the variable in the math object list, in the pane on the left hand side of the worksheet. Remember that you can enter an algebraic expression to account for changes with respect to any other variable in the program.

Table 9.2-1: Examples of User Entries for CDo.

Example	Formula (entered in the Formula box of Figure 2)	Comment
1	`0.001`	A constant value, which might be the result of a prior drag breakdown analysis for an single engine piston aircraft.
2	`0.001+0.05*(AOA*Pi/180)^2`	An example of how one could account for changes in the pressure drag with angle-of-attack.

3	`0.001+0.05*(AOA*Pi/180)^2+0.02*(AOY*Pi/180)^2`	An example of how one could account for changes in the pressure drag with angle-of-attack and angle-of-yaw. This is the formula of a surface and is plotted in Figure 9.2-2 for AOA ranging from -2° to 12° and AOY ranging from -15° to 15°. Also see Figure 9.2-1.
4	`0.001+0.05*(AOA*Pi/180)^2+0.02*(AOY*Pi/180)^2+0.0009*[SDaft(3)]`	An example of how to account for changes in angle-of-attack and angle-of-yaw, as well as the deflection of a flap, here assumed to be surface number 3. When [SDaft(3)] is 35°, a value of 0.0315 is added to the CDo.
5	`CDwing+CDfuse+CDldg+CDcool+CDtail+Cdnacelle+Cdprotruberance+CDmisc`	Here, the user has independently defined the extra math objects describing the drag buildup and is summing them up to return the basic drag coefficient.

*Note that these are just examples of how one might set such formulation up. Your formulation is likely to be different.

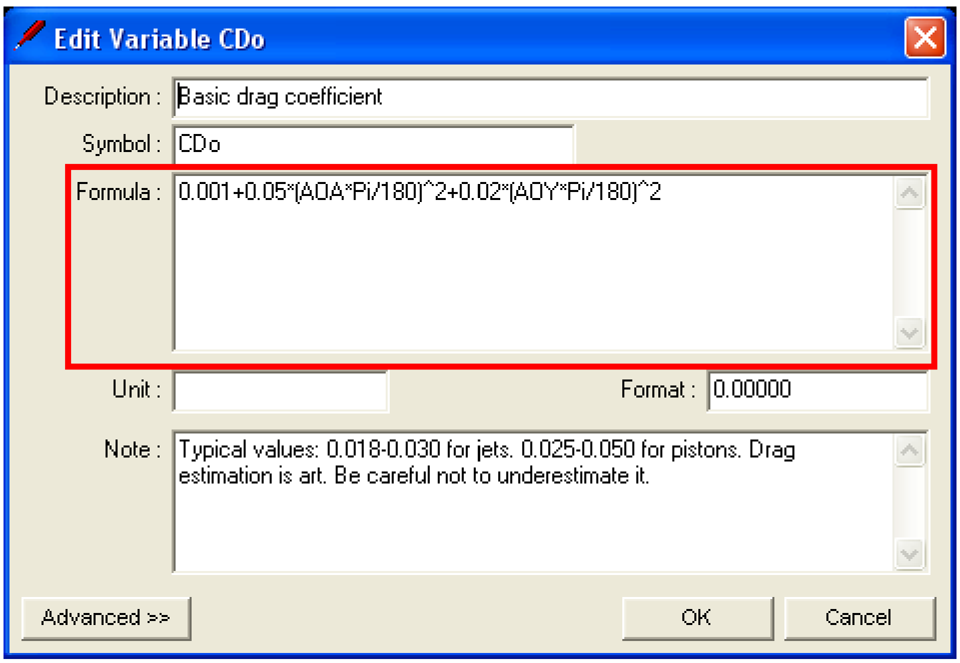


Figure 9.2-1: Entering the formula for Example 3 in Table 9.2-1 for the math object `CDo`.

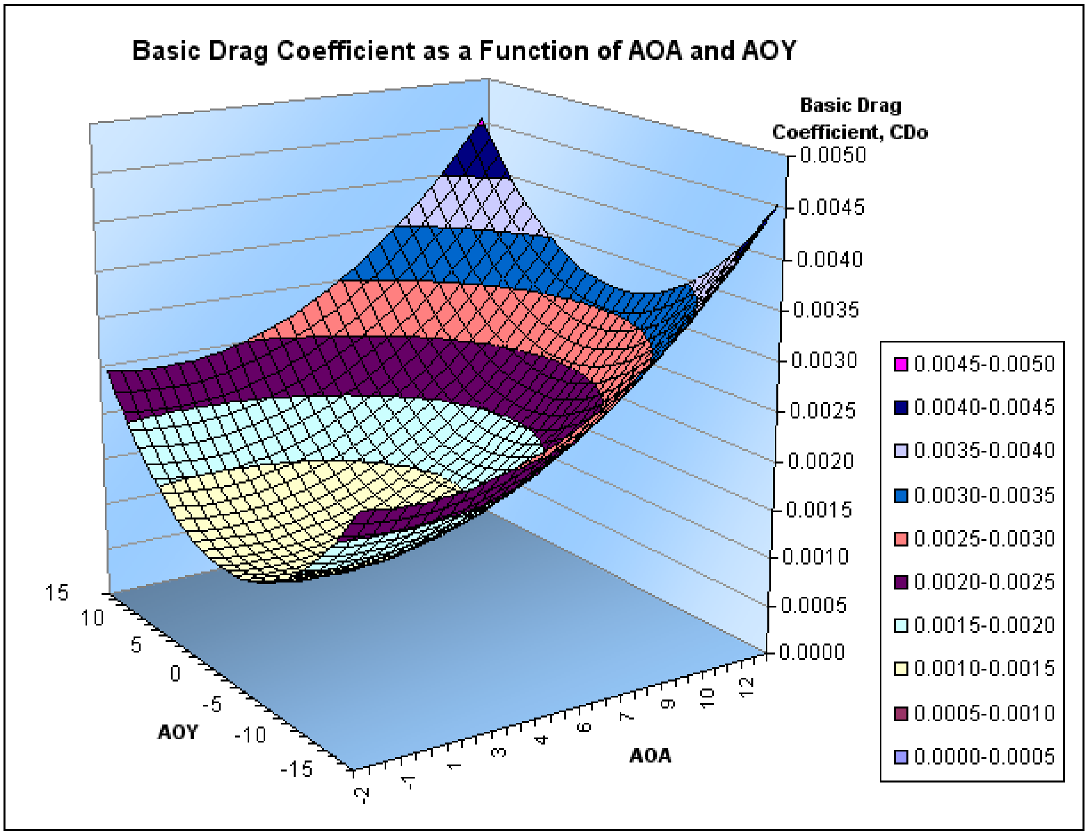


Figure 9.2-2: Basic drag coefficient of Example 3 plotted for AOA and AOY.

9.3 Skin Friction Drag Coefficient, C_{Df}

Skin friction is caused by the fluid viscosity as it flows over a surface. Its magnitude depends on the viscosity of air and the wetted (or total) surface area in contact with it. The coefficient is stored in the math object `CDf`. If compressibility modeling has been selected, the returned value is the compressible skin friction drag coefficient.

> **IMPORTANT!**
>
> Note that in this text, the *skin friction coefficient* is denoted by C_f and *skin friction drag coefficient* by C_{Df}. These are not interchangeable. C_f is determined for a laminar or turbulent boundary layer and is related to the wetted area, S_{wet}. The coefficient C_{Df} is the equivalent skin friction drag coefficient for the entire airplane and is related to the reference area, S_{ref}. For this reason, the distinction of the two terms must be kept in mind. The two are related, as shown in Equation (4).

The analysis of skin friction drag is complicated by a process called transition, when laminar boundary layer becomes turbulent (see Figure 9.3-1)[5]. This results in a mixed boundary layer, each with own skin friction coefficient. The nature of this behavior on airfoils is shown in Figure 9.3-2. Airfoils have two transition points; one on the upper and one on the lower surface. Each transition point moves forward or aft, as shown in the figure, when the angle-of-attack of the airfoil changes. Naturally, the travel is entirely dependent on the geometry and surface roughness of the airfoil.

[5] Note that it would be more correct to talk about a transition region. The line indicates a location beyond which transition has been completed.

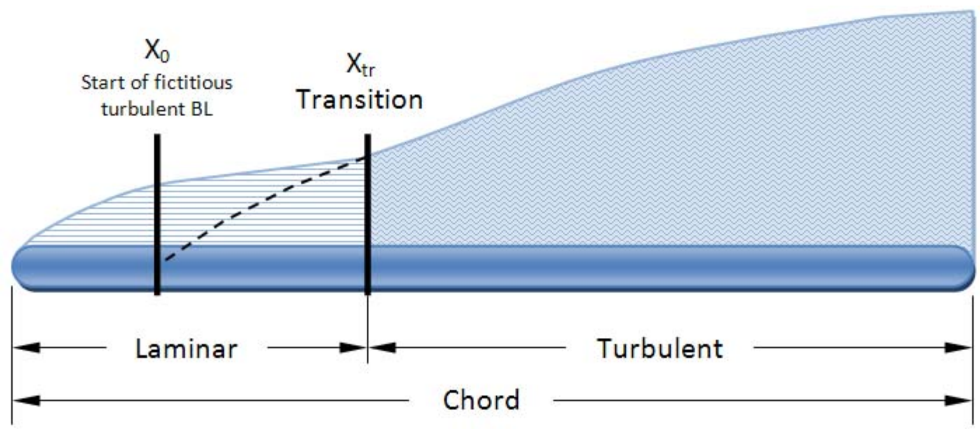


Figure 9.3-1: Mixed Boundary Layer conditions complicate skin friction drag analysis. This image is discussed in greater detail later.

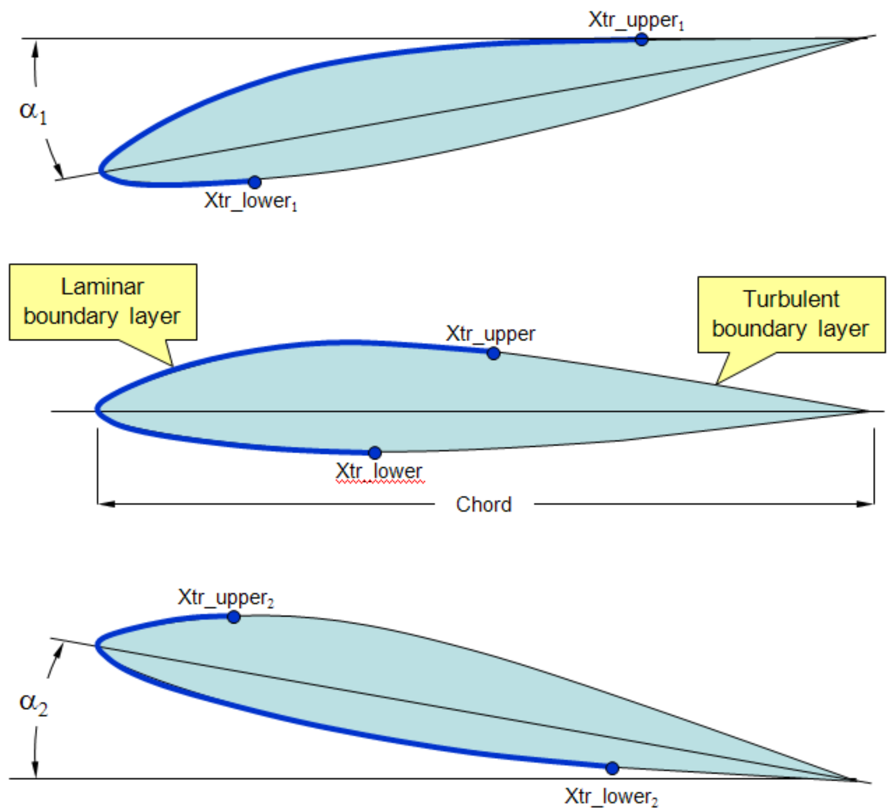


Figure 9.3-2: The laminar-to-turbulent transition points move around depending on angle-of-attack, airfoil shape, and surface roughness.

SURFACES employs a standard presentation of skin friction, for instance as presented in Reference 1. The skin friction drag coefficient is defined as follows:

$$C_{Df} = \frac{2D_f}{\rho V_\infty^2 S_{ref}} = C_f \left(\frac{S_{wet}}{S_{ref}} \right) \quad (4)$$

Where: D_f = Skin friction drag force in lb_f (UK system) or N (SI system).
ρ = Air density, typically in slugs/ft^3 or kg/m^3.
V_∞ = Far-field airspeed, typically in ft/s or m/s.
S_{wet} = Wetted area, typically in ft² or m².
C_{Df} = Skin friction drag coefficient, dimensionless.
C_f = Skin friction coefficient, dimensionless. See Equation (15) for more details.

If known, the user can enter an expression for the skin friction drag coefficient or use a combination of built-in functions in the two following ways:

1. Use any of the built-in functions that extract surface areas or wetted area of surfaces in your own formulation.
2. Use the built-in function `[CDf]` directly, but this requires skin friction coefficients to be defined for the surfaces to be used.

Either method (or a combination thereof) is very handy if you modify the geometry, as they will instantly update the skin friction drag coefficient. However, the `[CDf]` method is handier when you are estimating the skin friction drag of a new design. If you choose to use the built-in function `[CDf]` you should follow these steps to properly prepare the formulation (see Section 9.11 for an example setup):

STEP 1: Specify wetted area. Use the math object "`Swet`" for this purpose. The formula for "`Swet`" can be as simple as a number (if you know the value) to an algebraic representation using functions such as `[SA(surf1, surf2,…)]`or `[Swet(surf1, surf2,…)]`, which computes the total and wetted area of the selected surfaces `surf1`, `surf2`, and so on, respectively. At computation time the value of "`Swet`" is used internally with Equation (4).

STEP 2: Specify skin friction coefficients for each surface. You can do this in two ways. You can estimate a skin friction drag coefficient using your preferred method and enter for each surface[6]. Or you can use **SURFACES'** own internal estimation based on a laminar-to-turbulent boundary layer transition points that you provide. The latter method is probably far easier, but a numerical example of how **SURFACES** estimates this is presented later in this section to help clarify the method.

Since **SURFACES** models are made from infinitely thin surface panels, the program estimates wetted area by determining the surface area and then doubles the value to get wetted area. Table 9.3-1 shows some examples of possible user entries for `CDf`. If a function, such as `[Swet(surf1, surf2,…)]`, is used to estimate the wetted area, the user can multiply it by a factor to account for surface curvature (for instance as shown Example 3 in Table 9.3-1).

Table 9.3-1: Example user entries for CDf.

Example	Formula	Comment
1	`0.025`	A constant value, which might be the result of a prior drag breakdown analysis for an single engine piston aircraft.

[6] It is possible to enter this for multiple surfaces at a time.

2	`0.025-0.0001*Re^0.25`	This user accounted for changes in skin friction with Reynolds Number using this formula. It returns 0.0208 for Re = 3 000 000 and 0.01968 for Re = 8 000 000.
3	`0.025+0.000018*1.05*[Swet(5,6)]`	This user is adding the contribution of the additional wetted area of winglets (surfaces 5 and 6), multiplying the result by a 1.05 to correct for their curvature. For winglets with 50 ft² additional area, this formula returns 0.0260.
4	`0.01*(Cf_lam*Plam+Cf_turb*(100-Plam))*Swet/Sref`	Here the user is accounting for partial laminar flow in this estimation. The expression assumes the `Sref` will be divided out, leaving `Swet` remaining, when incorporated in standard drag calculations. The variable `Plam` means the percentage of laminar flow. `Plam` = 50 for laminar flow of up to 50% of wing wetted area. Note that `Swet` here is not the same as [`Swet()`]. See the discussion to follow for more information.
5	`[CDf]`	This formula returns the result of an internal calculation, in which all surfaces, to which a skin friction coefficient has been defined, are summed up using Equation (15).

*Note that these are just examples of how one might set such formulation up. Your formulation is likely to be different.

Other handy formulations are cited below for the convenience of the user.

Sutherland's Formula for Viscosity:
When using the UK system the temperature is in °R. In that case the viscosity can be found from[7]:

$$\mu = 3.170 \times 10^{-11} T^{1.5} \left(\frac{734.7}{T + 216} \right) \qquad \text{lb}_f\text{·s/ft}^2 \qquad (5)$$

When using the SI system the temperature is in K. In that case the viscosity can be found from[8]:

$$\mu = 1.458 \times 10^{-6} T^{1.5} \left(\frac{1}{T + 110.4} \right) \qquad \text{N·s/m}^2 \qquad (6)$$

Where; T = Outside Air Temperature, in °R or K.
μ = Air viscosity, in lb_f·s/ft² or N·s/m².

Reynolds Number:

$$Re = \frac{\rho V L}{\mu} \qquad (7)$$

[7] See Equation (2.90) of Reference 7.
[8] See Equation (2.91) of Reference 7.

Where; L = Reference length (for instance mean aerodynamic chord), in ft or m.
V = Reference airspeed, in ft/s or m/s.
ρ = Air density, in slugs/ft^3 or kg/m^3.
μ = Air viscosity, in lb$_f$·s/ft² or N·s/m².

A simple expression, valid for UK system at sea-level conditions only is (V and L are in ft/s and ft, respectively):

$$\mathrm{Re} \approx 6400VL \tag{8a}$$

A simple expression, valid for SI system at sea-level conditions only is (V and L are in m/s and m, respectively):

$$\mathrm{Re} \approx 68500VL \tag{8b}$$

Laminar Flow Skin Friction Coefficient[9]

This is the classical Blasius solution for a laminar boundary layer on a solid surface.

$$C_{flam} = \frac{1.328}{\sqrt{\mathrm{Re}}} \tag{9}$$

Turbulent Flow Skin Friction Coefficient[10]

This is the so-called Schlichting relation, which is found to be in good agreement with experiment.

$$C_{fturb} = \frac{0.455}{(\log_{10}(\mathrm{Re}))^{2.58}} \tag{10}$$

Turbulent Flow Skin Friction Coefficient – Compressible[11]

$$C_{fturb} = \frac{0.455}{(\log_{10}(\mathrm{Re}))^{2.58}(1+0.144M^2)^{0.65}} \tag{11}$$

Where; M = Mach Number.

Equation (10) and not (11) is the preferred form in **SURFACES** as the program will apply correction for compressibility effect using the Frankl-Voishel scheme. Using Equation (11) could result in the correction applied twice.

Mixed Laminar-Turbulent Flow Skin Friction Coefficient[12]

The method below is taken from Reference 8. Also refer to Figure 9.3-1 for the location of the points X_0 and X_{tr}. Of these, the user must specify the location of the transition point, which is used to calculate the start point of the fictitious turbulent laminar flow. This is required to ensure the boundary layer thickness is a continuous function. The user is referred to Reference 8 about methods on how to estimate transition location; however, often drag analysis in **SURFACES** involves estimating the impact of 25% or 50%

[9] See Equation (3.11) in Reference 8.
[10] See Equation (6.53) in Reference 8.
[11] See Equation (12.28) in Reference 4.
[12] See Section 6.8, pages 162-164 in Reference 8.

transition on the total airplane drag. In other words, the designer is attempting to answer a question like: “What is the benefit of achieving a partial laminar boundary layer on my design?” The answer may help direct the designer towards an appropriate airfoil.

$$\left(\frac{X_0}{C}\right) = 36.9 \times \left(\frac{X_{tr}}{C}\right)^{0.625} \left(\frac{1}{\text{Re}}\right)^{0.375} \tag{12}$$

Then, the skin friction coefficient is determined as follows:

$$C_f = \frac{0.074}{\text{Re}^{0.2}} \left(1 - \left(\frac{X_{tr} - X_0}{C}\right)\right)^{0.8} \tag{13}$$

Where; C = Reference length (e.g. wing chord).
X_0 = Location of the fictitious turbulent boundary layer.
X_{tr} = Location of where laminar boundary layer becomes turbulent.

Turbulent Flow Skin Friction Coefficient – Compressible[13]

Note that surface roughness affects C_{fturb}, but this is typically accounted for through the use of a so called “cutoff Reynolds Number.” If the actual Reynolds Number exceed the cutoff Reynolds Number, it is used instead. For more information on the topic, the reader is directed towards texts, such as Reference 4.

$$\text{Re}_{cutoff} = 38.21 \left(\frac{C}{k}\right)^{1.053} \tag{14}$$

Where; C = Reference length.
k = Skin roughness value.

The roughness value is based on the values in the following table, which is taken from Reference 4. If these are not acceptable, the user can also enter own Re_{cutoff} value.

Surface Type	**k**
Camouflage paint on aluminum	0.00040
Smooth paint	0.00025
Production sheet metal	0.00016
Polished sheet metal	0.00006
Smooth molded composite	0.00002

When using the built-in function [CDf], **SURFACES** uses Equation (15) to calculate the coefficient using all surfaces for which $(C_f)_i$ has been defined:

$$C_{Df} = \left(\frac{S_{wet}}{S_{ref}}\right) C_f = \left(\frac{S_{wet}}{S_{ref}}\right) \left(\frac{\sum_{i=1}^{N} (C_f)_i \times S_i}{S_{wet}}\right) \tag{15}$$

[13] See Equation (12.28) in Reference 4.

Where $(C_f)_i$ = Skin friction coefficient of surface i
N = Number of surfaces
S_i = wetted area of surface i (in ft² or m²)
S_{wet} = Wetted area (in ft² or m²)

Of these, the skin friction coefficient of each surface, $(C_f)_i$, needs further explanation. The user must estimate this value for each surface to be included in the analysis. This brings up an additional question: How does one handle laminar flow over a surface consisting of two distinct defining airfoils? In order to shed light on this, the demo aircraft model built in Section 4 will be used.

Consider the wing of the demo aircraft shown in Figure 9.3-2, which consists of two dissimilar airfoils on a tapered wing planform. The wing span is 18 ft, the root chord (Curve A1) is 3 ft and tip chord (Curve A2) is 2 ft (see Figure 4-1a). Also, the reference area is 45 ft² (as you will know if you created the model per the instructions in Section 4). Assume that at the given condition, the airfoil of curve A1 is a true laminar airfoil which is capable of sustaining 55% laminar flow on upper surface and 35% on the lower. The airfoil of curve A2 is a turbulent flow airfoil, but still sustains laminar flow to 15% on the upper surface and 15% on the lower. This airplane is cruising at 100 KTAS (168.8 ft/s) near sea-level, where the air density is 0.002378 slugs/ft^3. Determine the skin friction drag coefficient and force acting on the wing due to the mixed laminar and turbulent regions.

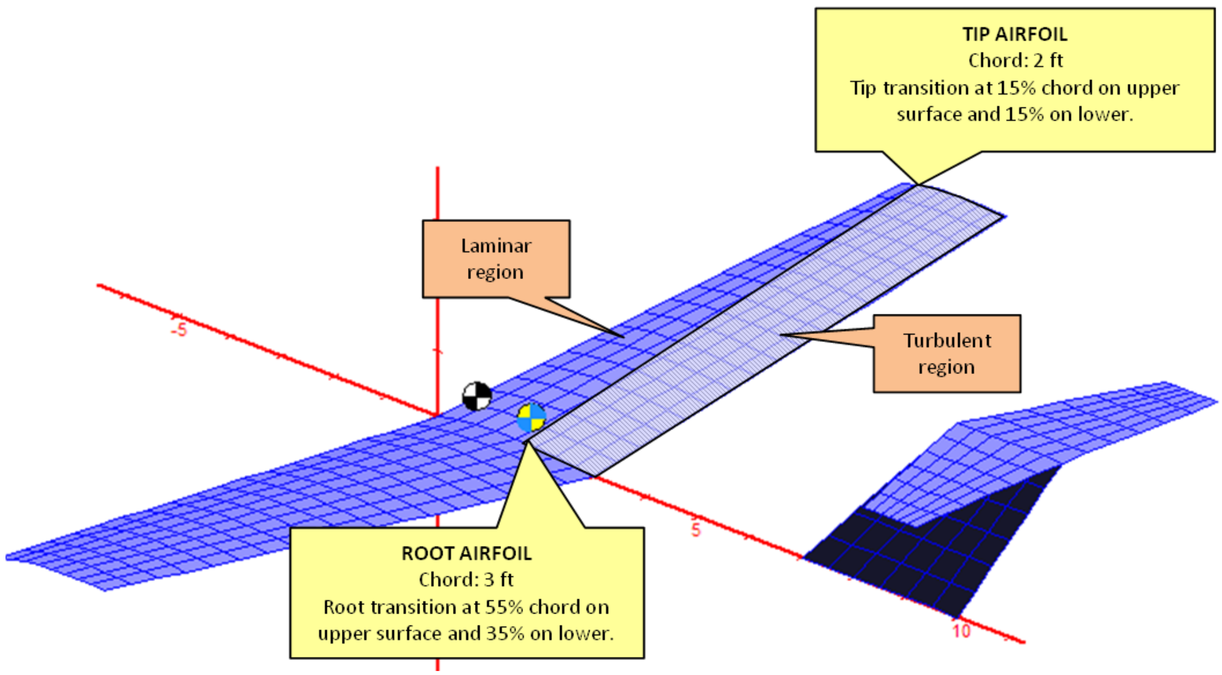


Figure 9.3-2: Example aircraft from Section 4.

One way to tackle this problem is to assume a linear change in laminar transition from A1 to A2. We'll calculate the skin friction, using the mixed boundary-layer formulation, as follows:

STEP 1: Start by using Equation (5) to compute the viscosity assuming an atmospheric temperature of 518.67 °R (15 °C):

$$\mu = 3.170\times10^{-11}(518.67)^{1.5}\left(\frac{734.7}{518.67+216}\right) = 3.745\times10^{-7} \qquad \text{lb}_f\cdot\text{s/ft}^2$$

STEP 2: Using Equation (7) we compute the Reynolds Number for airfoil 1, using a standard day air density of 0.002378 slugs/ft^3.

$$\text{Re}_1 = \frac{\rho VL}{\mu} = \frac{(0.002378)(168.8)(3)}{3.745\times10^{-7}} = 3215539$$

STEP 3: Then compute the Reynolds Number for airfoil 2

$$\text{Re}_2 = \frac{\rho VL}{\mu} = \frac{(0.002378)(168.8)(2)}{3.745\times10^{-7}} = 2143692$$

STEP 4: Using Equation (12) we compute the location of the fictitious turbulent boundary layer on the upper and lower surfaces of airfoil 1 (noting the different locations of the X_{tr} on each surface).

Lower: $$\left(\frac{X_0}{C}\right) = 36.9\times\left(\frac{X_{tr}}{C}\right)^{0.625}\left(\frac{1}{\text{Re}}\right)^{0.375} = 36.9\times(0.35)^{0.625}\left(\frac{1}{3215539}\right)^{0.375} = 0.06948$$

Upper: $$\left(\frac{X_0}{C}\right) = 36.9\times(0.55)^{0.625}\left(\frac{1}{3215539}\right)^{0.375} = 0.09216$$

STEP 5: Repeat for airfoil 2 (noting an equal value for each surface).

Lower: $$\left(\frac{X_0}{C}\right) = 36.9\times(0.15)^{0.625}\left(\frac{1}{2143692}\right)^{0.375} = 0.04763$$

Upper: $$\left(\frac{X_0}{C}\right) = 0.04763$$

STEP 6: The skin friction coefficient for upper and lower surface of airfoil 1 is determined using Equation (13) as follows:

Lower: $$(C_f)_{lower\,1} = \frac{0.074}{\text{Re}^{0.2}}\left(1-\left(\frac{X_{tr}-X_0}{C}\right)\right)^{0.8} = \frac{0.074}{3215539^{0.2}}(1-(0.35-0.06948))^{0.8} = 0.002841$$

Upper: $$(C_f)_{upper\,1} = \frac{0.074}{3215539^{0.2}}(1-(0.55-0.09216))^{0.8} = 0.002265$$

Call the average of the two the representative skin friction coefficient for airfoil 1, i.e.

$$C_{f_1} = \frac{0.002841+0.002265}{2} = 0.002553$$

STEP 7: Repeat for airfoil 2 (noting an equal value for each side).

Lower: $\left(C_f\right)_{lower\ 2} = \frac{0.074}{\mathrm{Re}^{0.2}}\left(1-\left(\frac{X_{tr}-X_0}{C}\right)\right)^{0.8} = \frac{0.074}{2143692^{0.2}}\left(1-\left(0.15-0.04763\right)\right)^{0.8} = 0.003677$

Upper: $\left(C_f\right)_{upper\ 2} = 0.00367$

The average of the two is of course:

$$C_{f_2} = 0.003677$$

STEP 8: The representative skin friction coefficient for the total wetted surface is simply the average of the coefficient for both airfoils, i.e:

$$C_f = \frac{0.002553+0.003677}{2} = 0.003115$$

STEP 9: Determine wetted area of the wing:

$$S_{wet} = 2\times\left[\tfrac{1}{2}\left(3+2\right)\times 18\right] = 90.0\ ft^2$$

STEP 10: Estimate skin friction drag due to the laminar flow.

$$D_{f\ lam} = \tfrac{1}{2}\rho V^2 \times S_{wet} \times C_f = \tfrac{1}{2}\left(0.002378\right)\left(168.8\right)^2 \times 90 \times \left(0.003115\right) = 9.5\ lbf$$

Note that an equivalent skin friction drag coefficient, which is based on S_{ref}, would be found from Equation (15):

$$C_{Df} = \left(\frac{S_{wet}}{S_{ref}}\right)C_f = \left(\frac{90}{45}\right)\left(0.003115\right) = 0.006230$$

Also note that the value, 0.003115 (and not 0.006230), is what one could enter as Cf_i for the wing surface when using the internal generation of C_{Df} in **SURFACES** (see the red box for each method below in Figures 9.3-2 and -3). This can be done by one of the two following methods.

Method 1: Surface-by-surface basis

Double-click on a surface to open its properties form. Click on the 'Tuning' tab. Enter the skin friction coefficient for the surface in the textbox in the red frame.

Method 2: Multiple surface entry

The user can select any number of surfaces (by holding Shift while clicking on surfaces) and then select **Modify->Surface Properties…**. Enter the desired value, which will be applied to all selected surfaces.

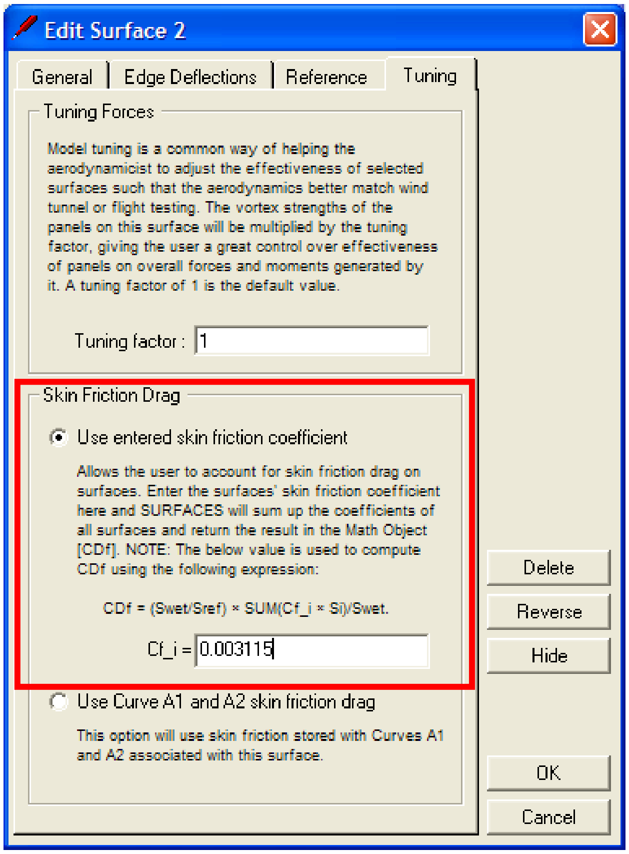


Figure 9.3-2: Method 1

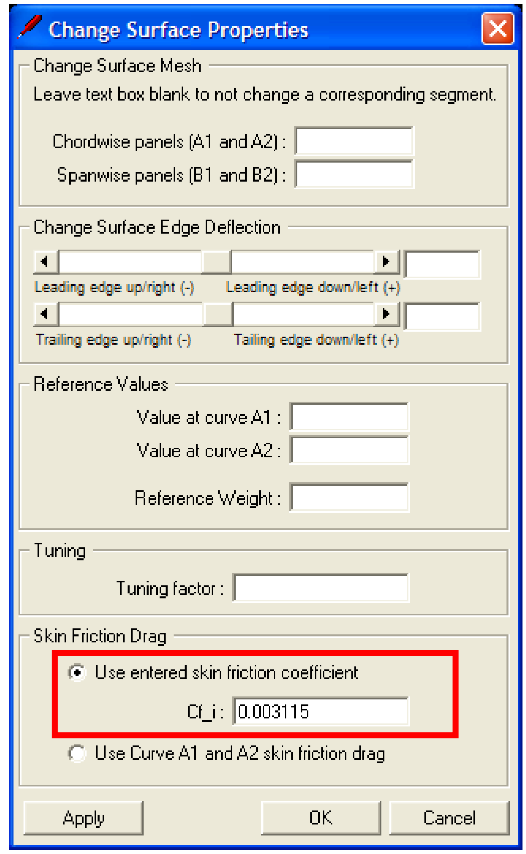


Figure 9.3-3: Method 2

9.4 Induced Drag Coefficient, C_{Di}

The induced drag is caused when the airflow perturbs the flow field as it makes its way around the wingtip (generating the wingtip vortices) of a 3D wing (see Figure 9.4-1), compared to what would happen to an infinitely long wing[14]. An integration of the pressure field over the wing yields a higher drag than would be obtained if this tip flow did not occur. In other words: the generation of the wingtip vortices induces the extra drag and the higher the lift, the higher is this additional drag.

The coefficient is stored in the math object `CDi`. If compressibility modeling has be selected, the returned value is the compressible induced drag coefficient. **SURFACES** allows the user to determine the induced drag using three different methods:

METHOD 1: Surface integration sums the pressure forces acting on each panel and resolves it into a three orthogonal components and rotates this to the wind axis coordinate system. Using the wing axis coordinate system, the force in the X-direction is by definition the drag, the force in the Y-direction is the side force, and the force in the Z direction is the lift.

[14] The astute student will recognize that D'Alembert's 2D paradox that a body in inviscid flow produces no drag does not apply in 3D flow, due to the downwash created by the trailing wake.

METHOD 2: $(C_L - C_{L\ CDmin})^2/(\pi \cdot ARref \cdot Eref)$ method computes the induced drag based on the current lift coefficient, the CL where minimum drag occurs ($C_{L\ CDmin}$), reference Aspect Ratio (ARref), and reference span efficiency (Eref).

METHOD 3: Trefftz plane integration uses flow perturbations in an imaginary plane infinitely far behind the model to determine the induced drag. The location of the plane is a mathematical simplification that allows one to neglect the x-perturbation from the flow field formulation, as it is theoretically zero that far from the model. This way, a 3D relationship (volume) can be considered as 2D (plane).

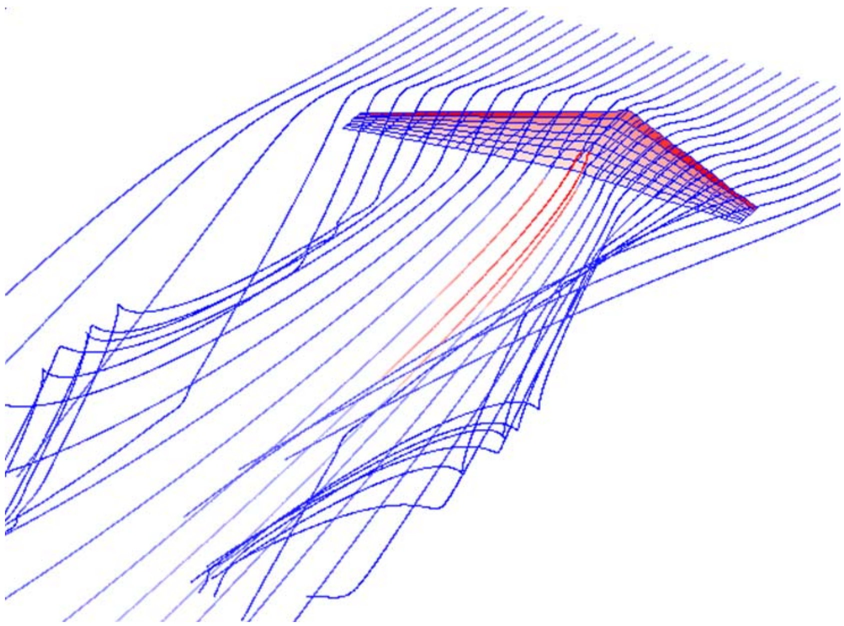


Figure 9.4-1: A 3D wing in airflow.

9.5 Total Drag Coefficient, C_D

Once **SURFACES** has determined the basic, skin friction, and induced drag coefficients, it computes the total drag coefficient using Equation (3), repeated here for convenience. The coefficient is stored in the math object `CD`.

$$C_D = C_{Do} + C_{Df} + C_{Di} \qquad (3)$$

It should be noted that the coefficients are based on S_{ref}. forces Equation (3) can be rewritten as follows:

$$C_D = \frac{2D_o}{\rho V^2 S_{ref}} + \frac{2D_f}{\rho V^2 S_{wet}} + \frac{2D_i}{\rho V^2 S_{ref}} \qquad (16)$$

For internal consistency, we could thus write;

$$C_D = \frac{2D_o}{\rho V^2 S_{ref}} + \frac{2D_f}{\rho V^2 S_{wet}}\left(\frac{S_{ref}}{S_{ref}}\right) + \frac{2D_i}{\rho V^2 S_{ref}} = \left(\frac{2}{\rho V^2 S_{ref}}\right)\left(D_o + D_f\left(\frac{S_{ref}}{S_{wet}}\right) + D_i\right) \quad (17)$$

Which, is how **SURFACES** returns the total drag coefficient.

Table 9.5-1: Example user entries for CD.

Example	Formula	Comment
1	`CDo+CDf+CDi`	Here, the math objects CDo, CDf, and CDi have already been defined (as it is in the standard template).
2	`[CDo]+[CDf]+[CDi]`	This could be a way to account for changes in skin friction with Reynolds Number.
3	`0.0045+0.000023*1.05*[Swet(5,6)]`	Here a user is adding contribution of the wetted area of surfaces 5 and 6, multiplying the result by a 1.05 to account for curvature.

9.6 Compressibility Modeling

SURFACES allows the user several options in compressibility modeling. Figure 9.6-1 shows the form used to select compressibility modeling. If no modeling is selected, **SURFACES** will return the incompressible coefficients C_L, C_D, C_{Do}, C_{Df}, and C_{Di}. Otherwise, the values returned will include the compressibility corrections. The following corrections are included:

Table 9.6-1: Compressibility formulation in SURFACES.

Name	Formulation	Remarks	Reference
Prandtl-Glauert	$C_P = \frac{C_{Po}}{\sqrt{1-M^2}}$	Typically under-predicts experimental values. Simple enough to be applicable to most of the coefficients.	Ref. 6, Equation (9.36)
Karman-Tsien	$C_P = \frac{C_{Po}}{\sqrt{1-M^2} + \left(\frac{M^2}{1+\sqrt{1-M^2}}\right)\frac{C_{Po}}{2}}$	Is applied directly to panel pressure coefficients inside **SURFACES** and is thus not applied to C_{Do}. Approaches Prandtl-Glauert for low Mach Numbers.	Ref. 6, Equation (9.40)
Laitone	$C_P = \frac{C_{Po}}{\sqrt{1-M^2} + \left(\frac{M^2(1+0.2M^2)}{2\sqrt{1-M^2}}\right)C_{Po}}$	Is applied directly to panel pressure coefficients inside **SURFACES** and is thus not applied to C_{Do}.	Ref. 6, Equation (9.39)
User defined	-	Is applied directly to panel pressure coefficient inside **SURFACES** and also to C_{Do}.	-
Frankl-Voishel	$C_{Df} = C_{Dfo}(0.000162\mathrm{M}^5 - 0.00383\mathrm{M}^4 + 0.0332\mathrm{M}^3 - 0.118\mathrm{M}^2 + 0.0204\mathrm{M} + 0.996)$	Based on Frankl-Voishel. The polynomial is obtained by interpolating the data in the graph on that page.	Ref. 3, 5.1.5.1-15.

Table 9.6-2: Compressibility Modeling in SURFACES.

When user selects…	C_L	C_D	C_{Do}	C_{Df}	C_{Di}
…these compressibility models are applied:					
None	None	None	None	None	None
Prandtl-Glauert	Prandtl-Glauert	Prandtl-Glauert	Prandtl-Glauert	Frankl-Voishel	Prandtl-Glauert
Karman-Tsien	Karman-Tsien	Karman-Tsien	Prandtl-Glauert	Frankl-Voishel	Karman-Tsien
Laitone	Laitone	Laitone	Prandtl-Glauert	Frankl-Voishel	Laitone
User defined	User defined	User defined	User defined	Frankl-Voishel	User defined

As can be seen from Table 9.6-2, the compressible C_{Do} always uses the Prandtl-Glauert correction when Karman-Tsien or Laitone are selected for C_{Di}. Frankl-Voishel is always used to correct C_{Df}.

9.8 How SURFACES Calculates Do, Df, Di, and D.

Once **SURFACES** has determined the constituent drag coefficients is computes the basic drag, skin friction drag, induced drag, and total drag forces using the following formulation:

Basic Drag Force:
$$D_o = \tfrac{1}{2}\rho V_\infty^2 S_{ref} C_{Do} \tag{18}$$

Skin Friction Drag Force:
$$D_f = \tfrac{1}{2}\rho V_\infty^2 S_{wet} C_{Df} \tag{19}$$

Induced Drag Force:
$$D_i = \tfrac{1}{2}\rho V_\infty^2 S_{ref} C_{Di} \tag{20}$$

Total Drag Force:
$$D = \tfrac{1}{2}\rho V_\infty^2 S_{ref} C_D \tag{21}$$

9.9 Limitations of Drag Estimation Methodologies

Figure 9.9-1 shows what a true drag polar might look like for a real airplane. This data might have been collected in flight or wind tunnel testing. The figure also shows a "simulated" drag polar, using a standard second order polynomial representation (also known CL-squared method). This is represented by an equation such as:

$$C_D = C_{Do} + C_{Df} + (C_L - C_{L\,CDmin})^2/\pi \cdot AR \cdot e$$

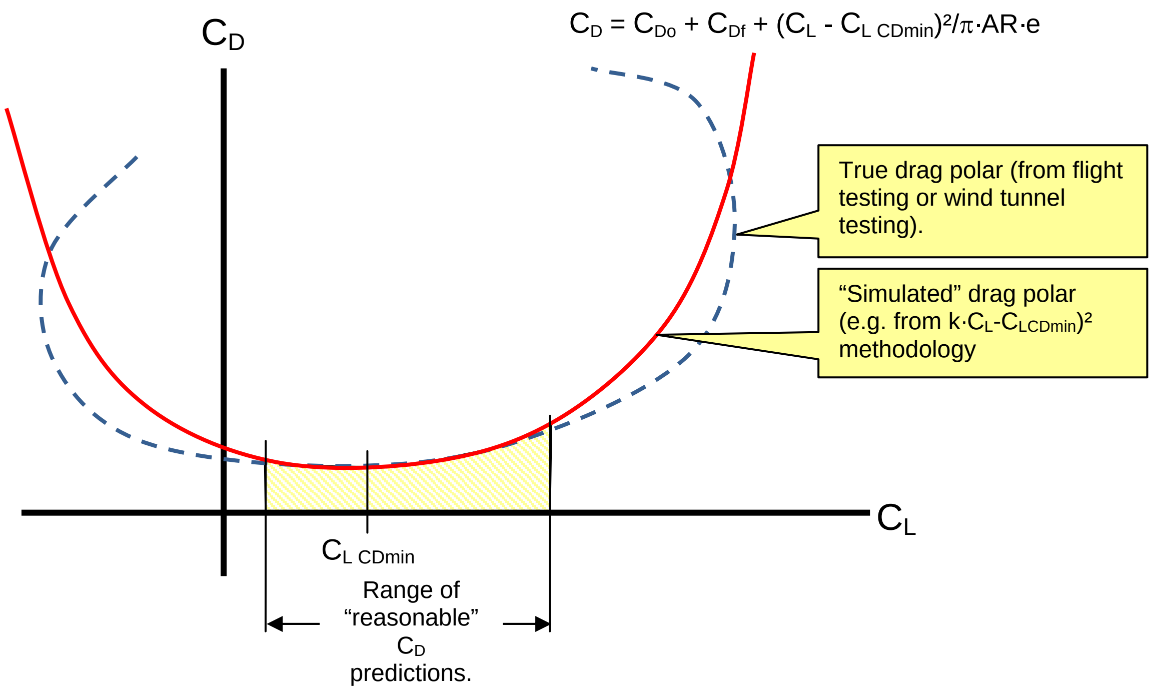


Figure 9.9-1: Typical drag polar.

The user should realize that the CL² method is only a representation that works over a range of C_Ls. It becomes increasingly inaccurate if C_L is too low or too high. Analysis done using that drag model will only be reliable within that range. For instance, predictions based on the red curve in Figure 9.9-1 would indicate less performance at higher AOA than the airplane would display in reality. However, there might also be a scenario in which the simulated curve indicated less drag, and therefore better performance than the real airplane would be capable. The point is that the user must understand the limitations of any prediction made.

9.10 Setting up Drag Modeling on Example Aircraft

One of the advantages in using **SURFACES** is the geometric information can be utilized directly when determining aerodynamic parameters. For instance, consider the balance a designer must find between lift and drag. A large wing area results in a lower stalling speed, but greater drag and structural weight. Being able to evaluate such parameters on the fly, as one modifies the wing (and thus its area) is priceless to the aircraft designer. This section will show how to use geometric relations in drag modeling. The model created in Section 4 will be used in a Step-by-Step procedure.

Generally, the user should prepare models for geometric relations after they have been constructed, in order to prevent relations to become corrupt as a consequence of adding and deleting geometric entities during the construction phase. At any rate, it is a good practice to check for errors in the assignment of geometric references before solving.

STEP 1: Open the demo airplane project from Section 4. Select **File->Open…** and navigate to find the file SIMPLE DEMO.SRF. Double-click to open.

STEP 2: Select the X-Y-Z view and orient the airplane similar to what is shown in Figure 9.10-1.

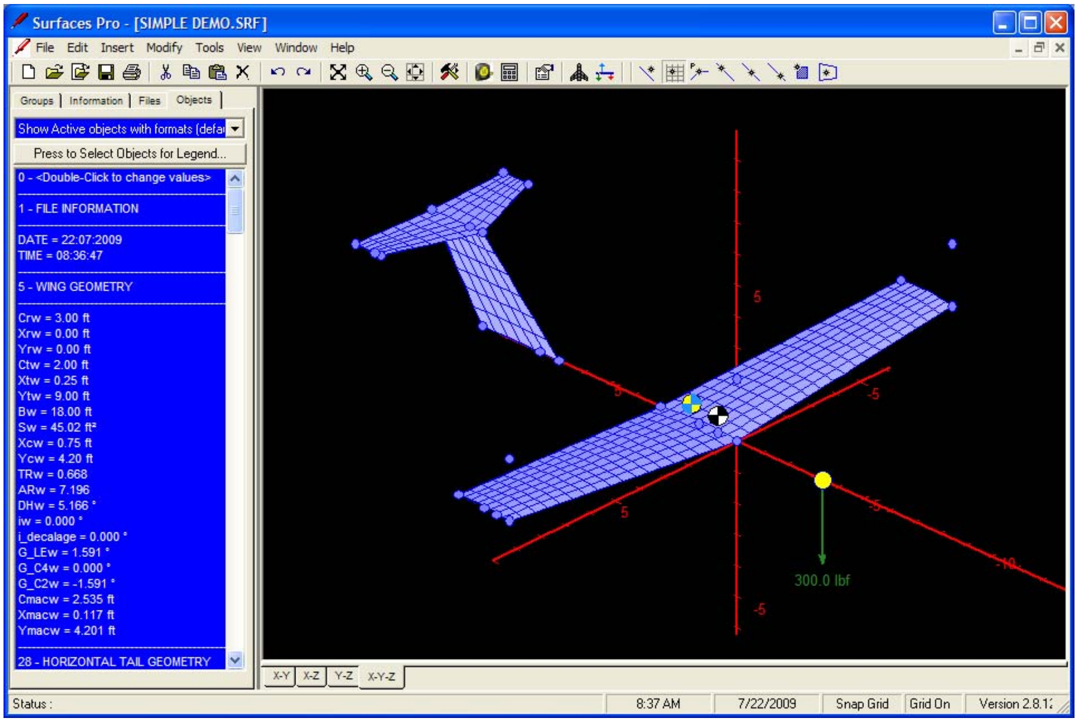


Figure 9.10-1: The model if Step 2 was followed.

Let's define the basic drag coefficient as follows:

```
CDo = 0.001+0.05*(AOA*Pi/180)^2+0.02*(AOY*Pi/180)^2
```

Let's define the skin friction drag coefficient as follows:

```
CDf = [CDf]
```

And let's define the induced drag coefficient as follows:

```
CDi = [CDi]
```

Now, let's enter these:

STEP 3: Open the VLM Console. From the **Edit** menu select **Reference Drag Modeling…** (See Figure 9.10-2). This opens the dialog box shown in Figure 9.10-3. Enter the above drag coefficients and other information as shown in the figure. When done, press the [OK] button to store the entered information and close the form.

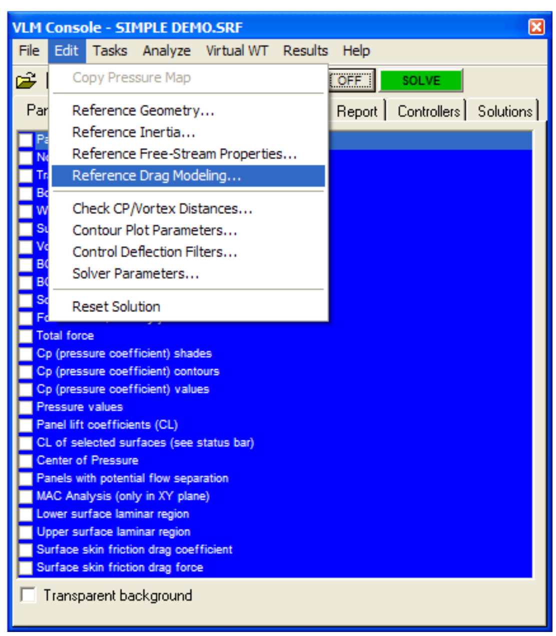


Figure 9.10-2: Select Reference Drag

Modelling… from the VLM Console.

This step tells **SURFACES** how to compute our three crucial drag coefficients. We will now set up the skin friction modeling for the surfaces and tell **SURFACES** how exactly to compute the skin friction drag.

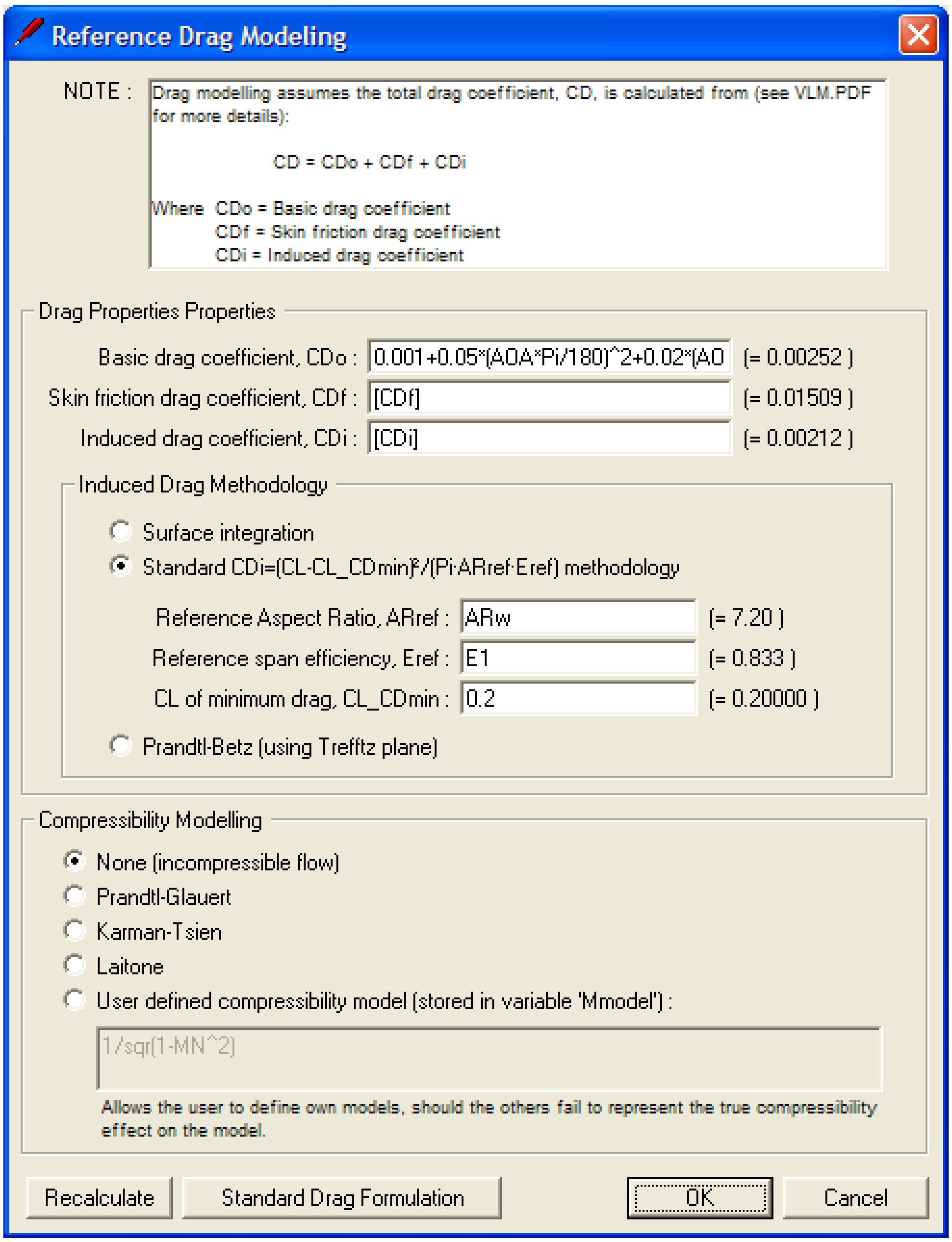


Figure 9.10-3: Step 3 calls for this form to be filled out as shown.

STEP 4: Go back to the worksheet and select **Edit->Select Surfaces…** The press the [Select All] button in the form that opens up and then press the [OK] button (see Figure 9.10-4). Now all the surfaces are selected.

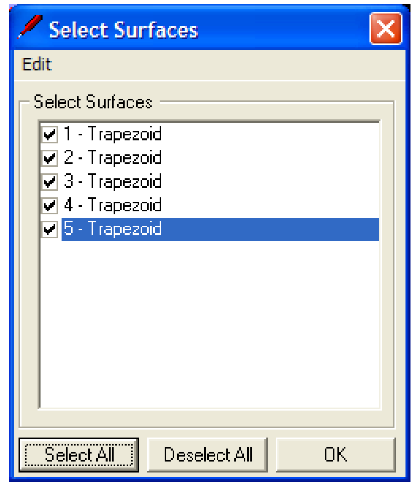


Figure 9.10-4: A quick selection of all surfaces.

STEP 5: Then select **Modify->Surface Properties…** Select the option 'Use Curve A1 and A2 skin friction drag' as shown in Figure 9.10-5. Press [OK].

This step tells **SURFACES** to calculate the skin friction drag using information we have yet to enter for the A1 and A2 curves of the surfaces.

First, let's assume the HT and VT are to be designed using laminar flow airfoils capable of sustaining 50% laminar flow. Let's also assume the wing will sustain laminar flow as discussed in the example of Section 9.3.

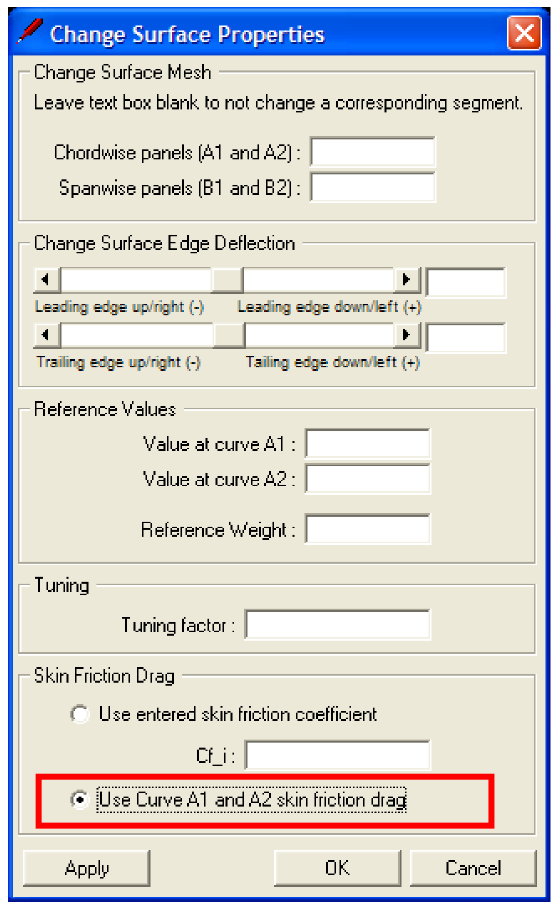


Figure 9.10-5: A quick selection of all surfaces.

STEP 6: Select **all the vectors** as shown in Figure 9.10-6. Make sure you use the rubberband when selecting the centerline vectors as there are really three vectors (or airfoils) there; two belonging to the HT root and one to the VT tip. The following assumes you did this correctly.

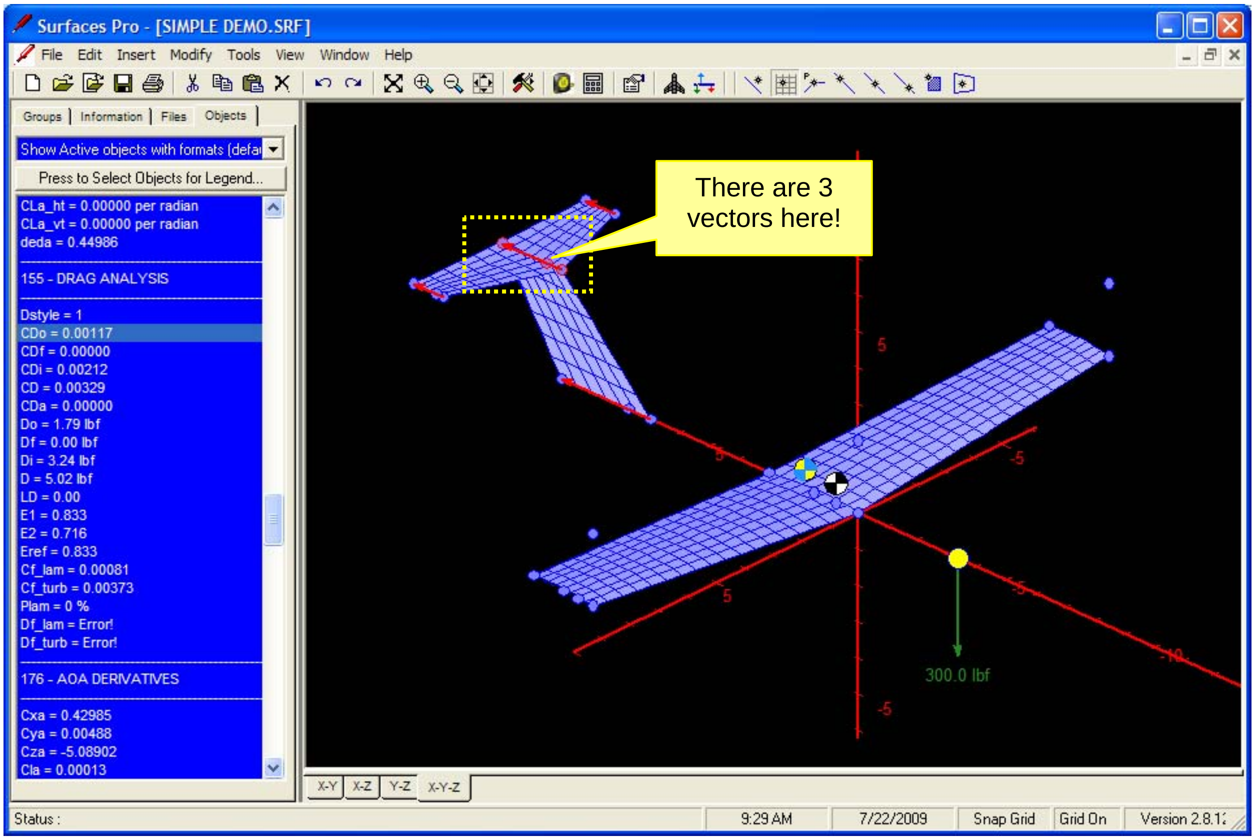


Figure 9.10-6: A1 and A2 curves have been selected for all surfaces to be included.

STEP 7: Select **Tools->Distribute Laminar Transition for Selected Vectors…** from the worksheet.

This opens the form shown in Figure 9.10-7. As said earlier, we are assuming here that the airfoils can sustain 50% laminar flow on the upper and lower surfaces. This case is often checked by aircraft designers and is especially prepared here for quick entry. You can simply press the buttons labeled [0%], [25%], and [50%] to set up these special cases. This assumes a constant transition (i.e. independent of AOA) throughout the operational range, which is not necessarily true, but handy for quick-studies.

STEP 8: Press the [50%] button to fill in the textboxes in the form. Select the option 'Smooth molded composite' for surface type. Press the [OK] button to accept the editing.

SURFACES is equipped with a handy tool to help you visualize your work. Let's turn it on.

STEP 9: Open the VLM Console. Select the 'Panel Results' tab find and check the option 'Upper surface laminar region' (see Figure 9.10-8). View the results in Figure 9.10-9).

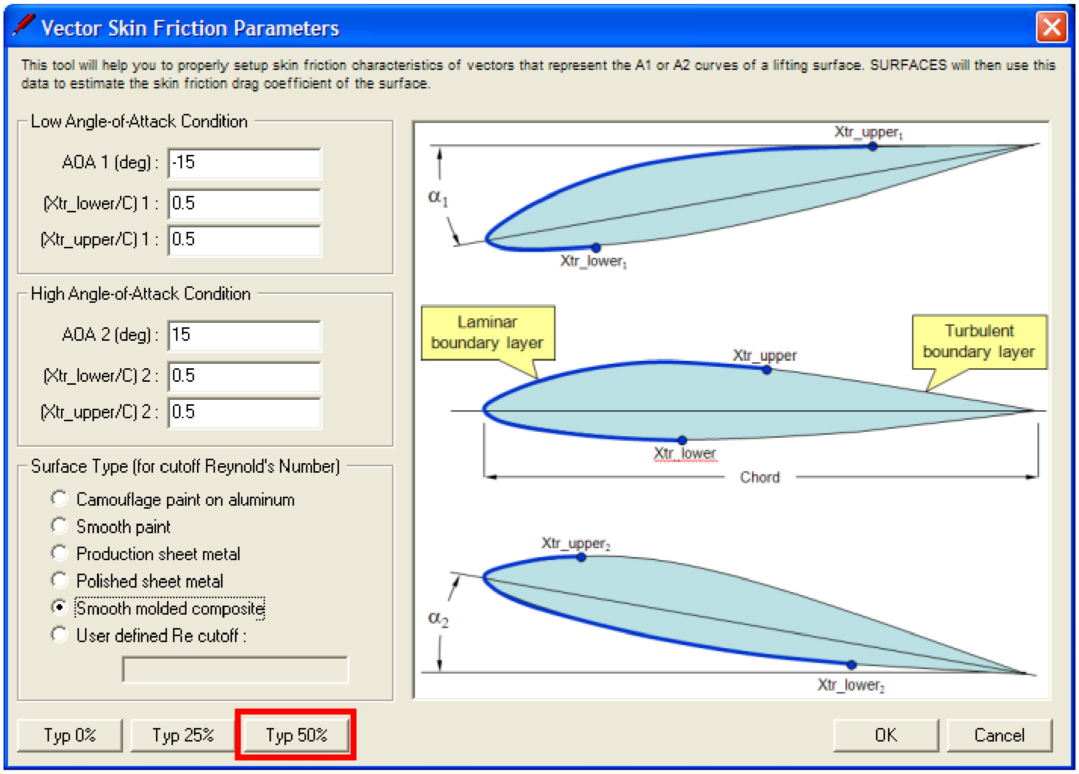


Figure 9.10-7: Entering laminar-to-turbulent transition information for the selected vectors.

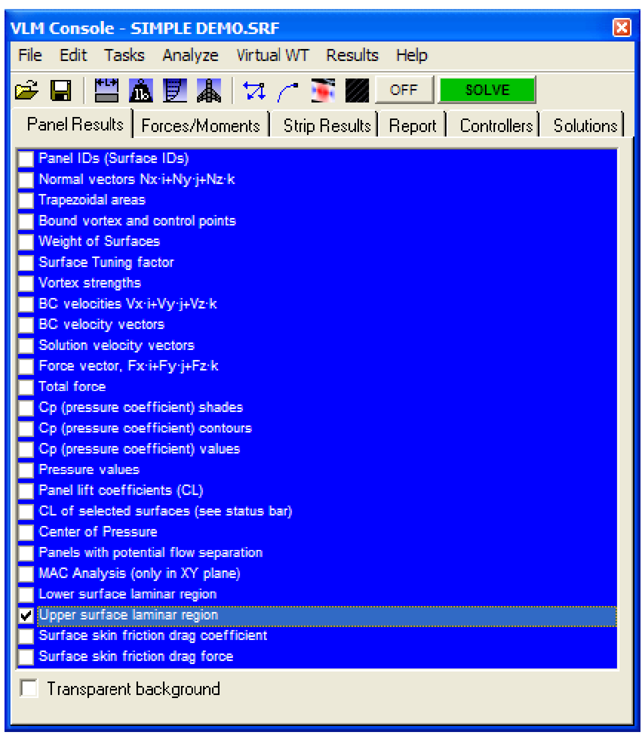


Figure 9.10-8: Display laminar-turbulent regions.

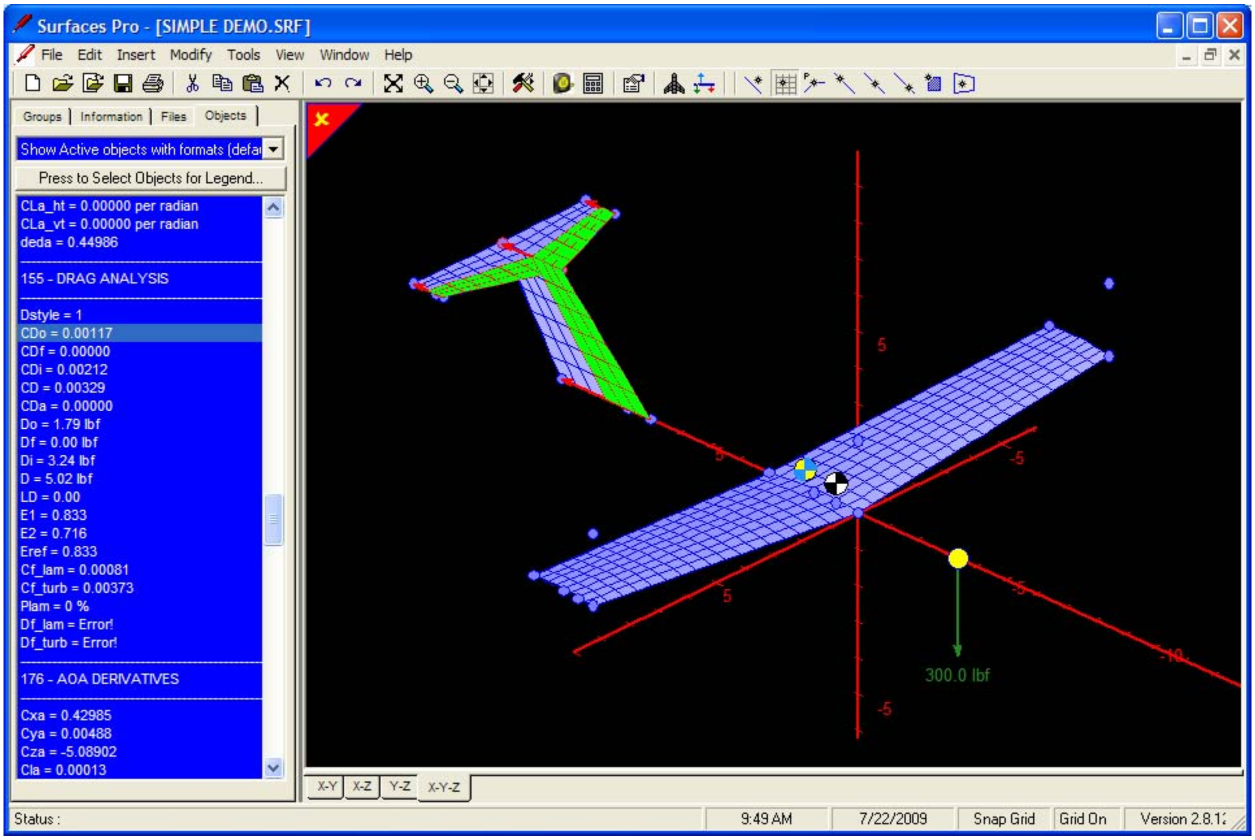


Figure 9.10-9: Image shows the laminar flow region (green) on the HT and VT. Note that when you select to enter the skin friction coefficient directly (see Cf_i in Figure 9.10-5), rather than using the A1/A2 curves, SURFACES won't know the extent of laminar flow and, thus, will not plot the green areas as shown here.

Note that at computation time, **SURFACES** will compare the actual AOA to the ones filled in Figure 9.10-7 and estimate the transitions at that angle-of-attack. If the AOA is less than the value AOA1, it will use the transition values entered for the low angle-of-attack condition. If the AOA is larger than AOA2 then it will use the values entered for the high angle-of-attack condition. Now let's set up the mixed boundary-layer conditions on the wing.

STEP 10: Select the wing tip vectors as shown in Figure 9.10-10.

STEP 11: Select **Tools->Distribute Laminar Transition for Selected Vectors…** from the worksheet. Enter the information shown in Figure 9.10-11.

STEP 12: Select the wing root vectors as shown in Figure 9.10-12. Again, make sure you use the rubberband when selecting the centerline vectors as there are two vectors there. The following assumes you did this correctly.

STEP 13: Select **Tools->Distribute Laminar Transition for Selected Vectors…** from the worksheet. Enter the information shown in Figure 9.10-13.

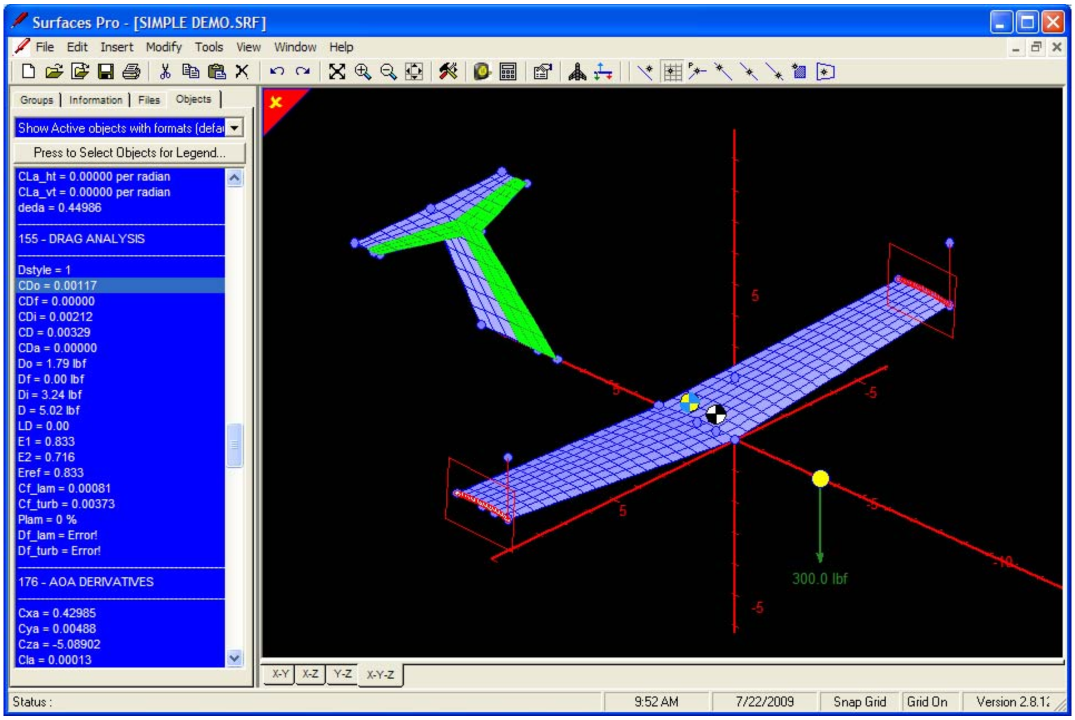


Figure 9.10-10: Selecting the wing tip vector in Step 10.

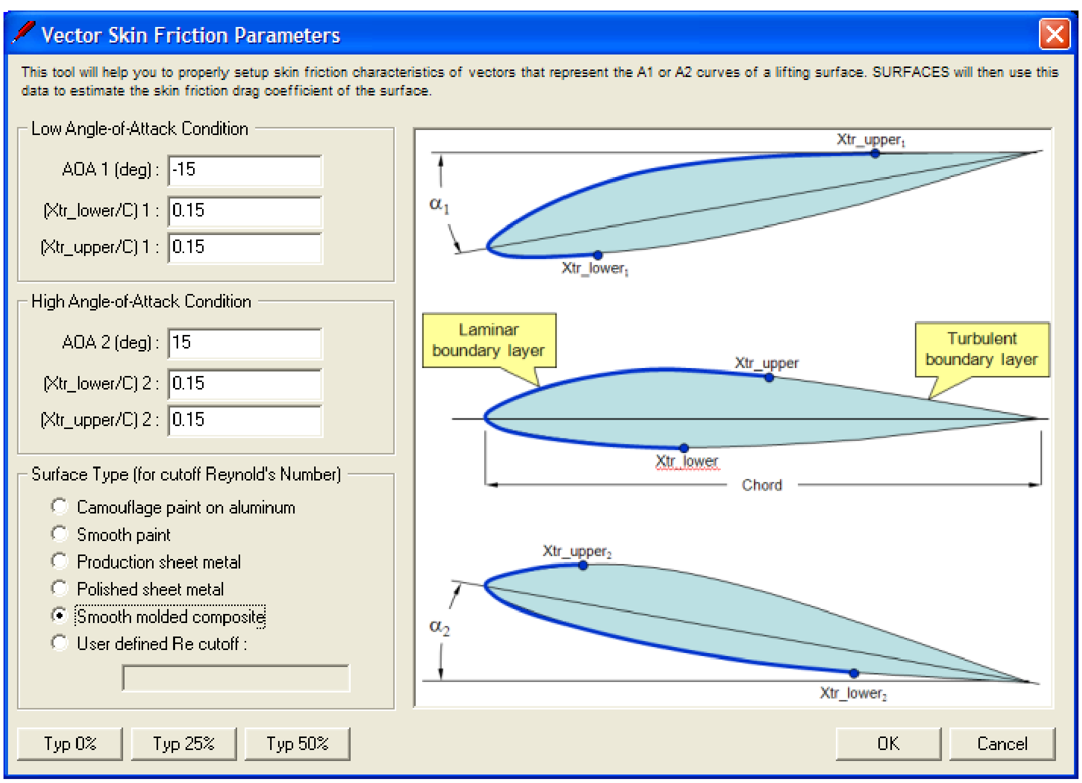


Figure 9.10-11: Entering transition information for the wing tip in Step 11.

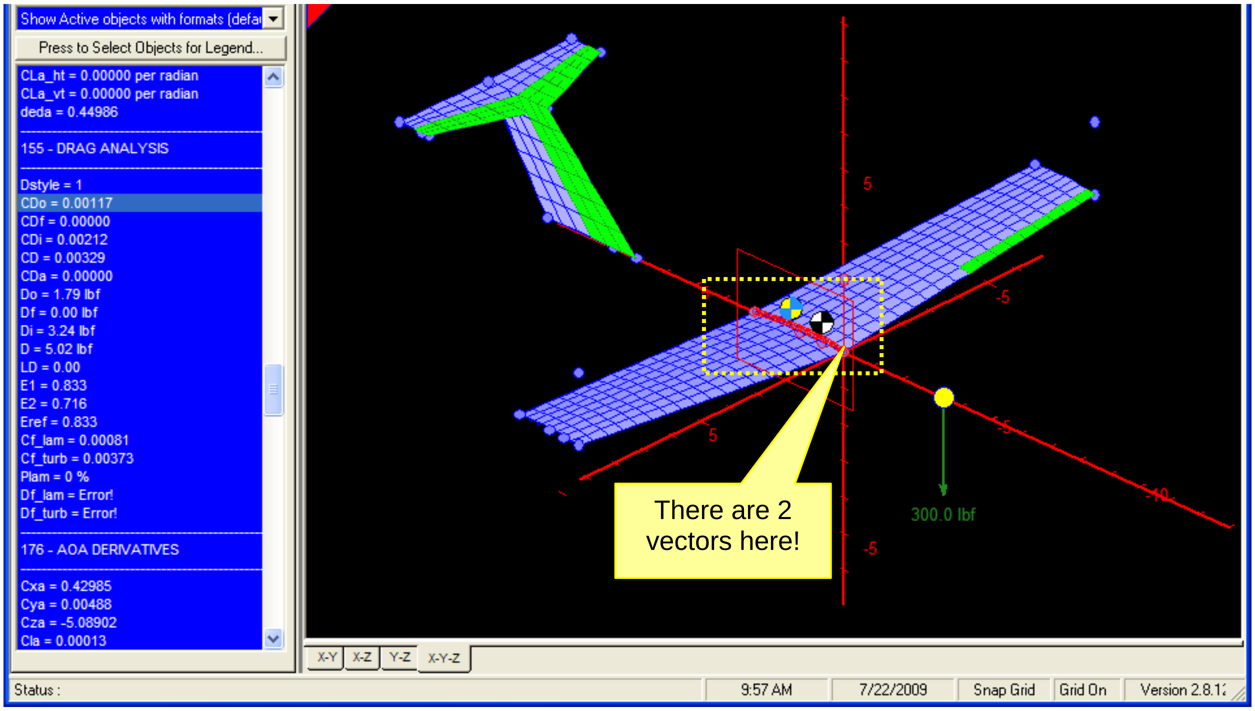


Figure 9.10-12: Selecting the wing root vector in Step 12.

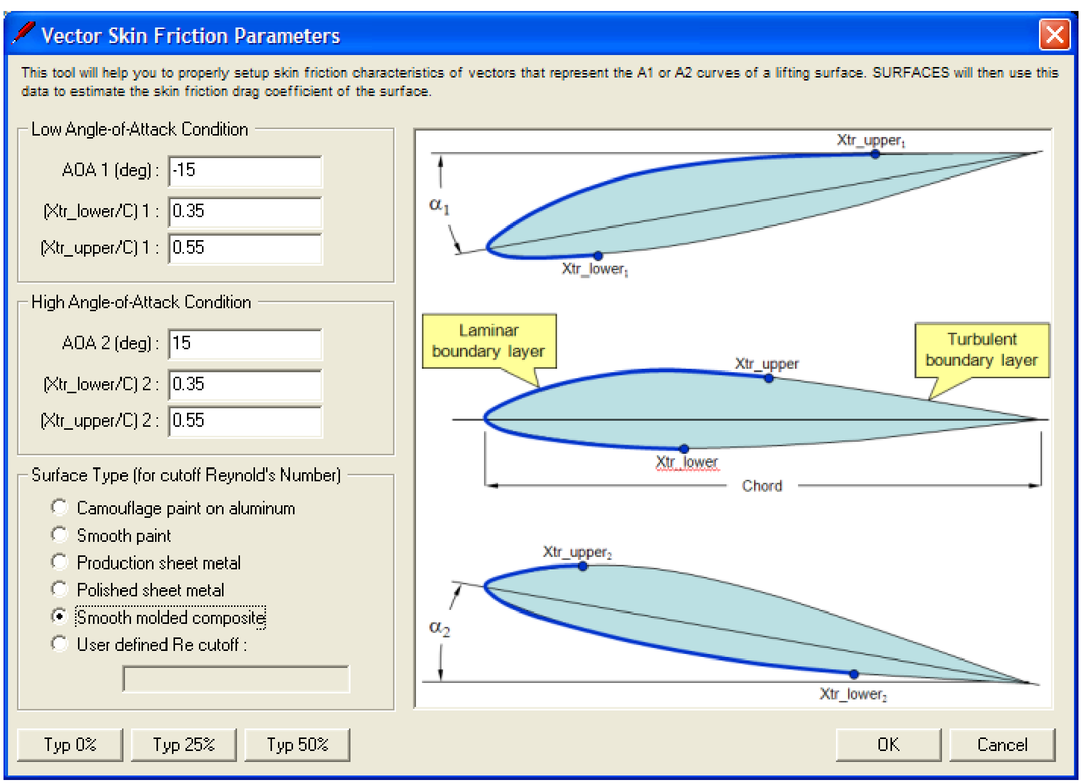


Figure 9.10-13: Entering transition information for the wing root in Step 13.

Now, only one thing remains. The wetted area for all the surfaces involved must be accounted for, or **SURFACES** won't be able to compute the skin friction drag coefficient. Let's do this.

STEP 14: In the math objects list under the Objects tab tab on the pane in left hand side of the worksheet, find the variable Swet. It should be in a block of variables under the title "REFERENCE PARAMETERS". Double-click on it to open the variable editor (see Figure 9.10-1) and enter the function [Swet(1,2,3,4,5)] (the order of the arguments doesn't matter here). This will calculate the wetted area of the selected surfaces. Press [OK] when done. .

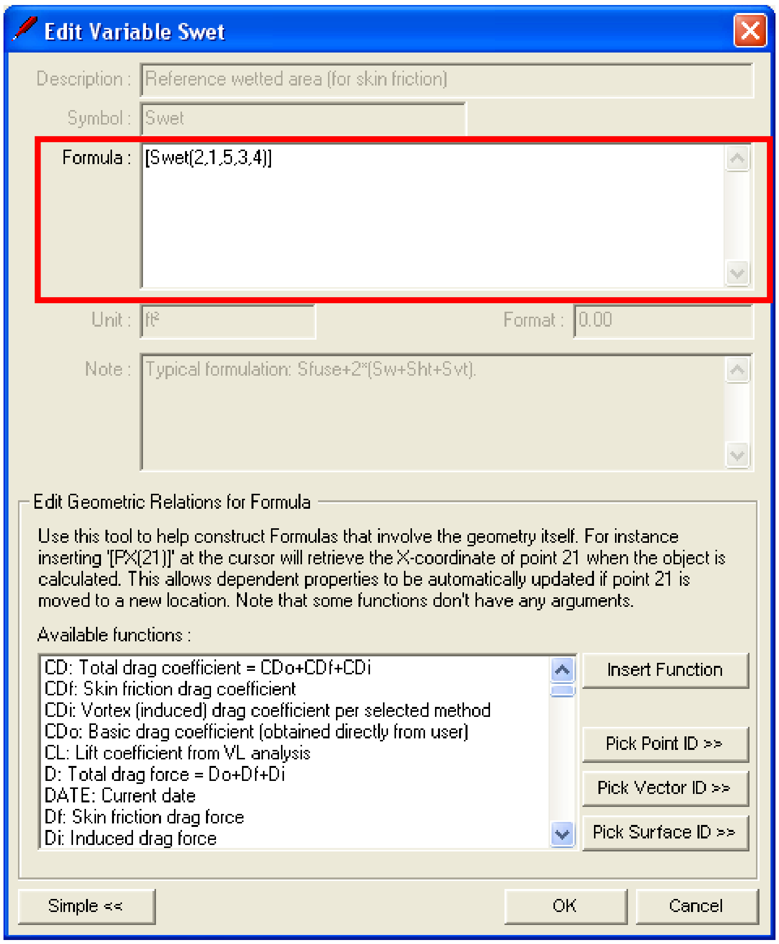


Figure 9.10-14: Editing variable Swet.

That's it. The model is ready to be used for drag estimation. The model with the entered laminar flow regions is shown in Figure 9.10-15. The reported skin friction drag coefficient for the entire aircraft is 0.00907, but this yields a skin friction drag of a 38 lbf. But there is more. **SURFACES** allows us to take a closer look at some other details about the skin friction drag. From the VLM Console's Panel Results tab you can select to have the program display the resulting skin friction drag coefficients or forces on each surface. For instance, Figure 9.10-16 shows that each half of the HT is generating 2.6 lbf of skin friction drag, while the VT produces some 3.9 lbf (remember that the airplane modeled is small, perhaps UAV sized). Additionally, it is of interest in noting that by setting the transition of all airfoils to 0% (turbulent airfoils) C_{Df} jumps to 0.01179 and skin friction drag to 49.4 lbf; i.e. by almost 30%!

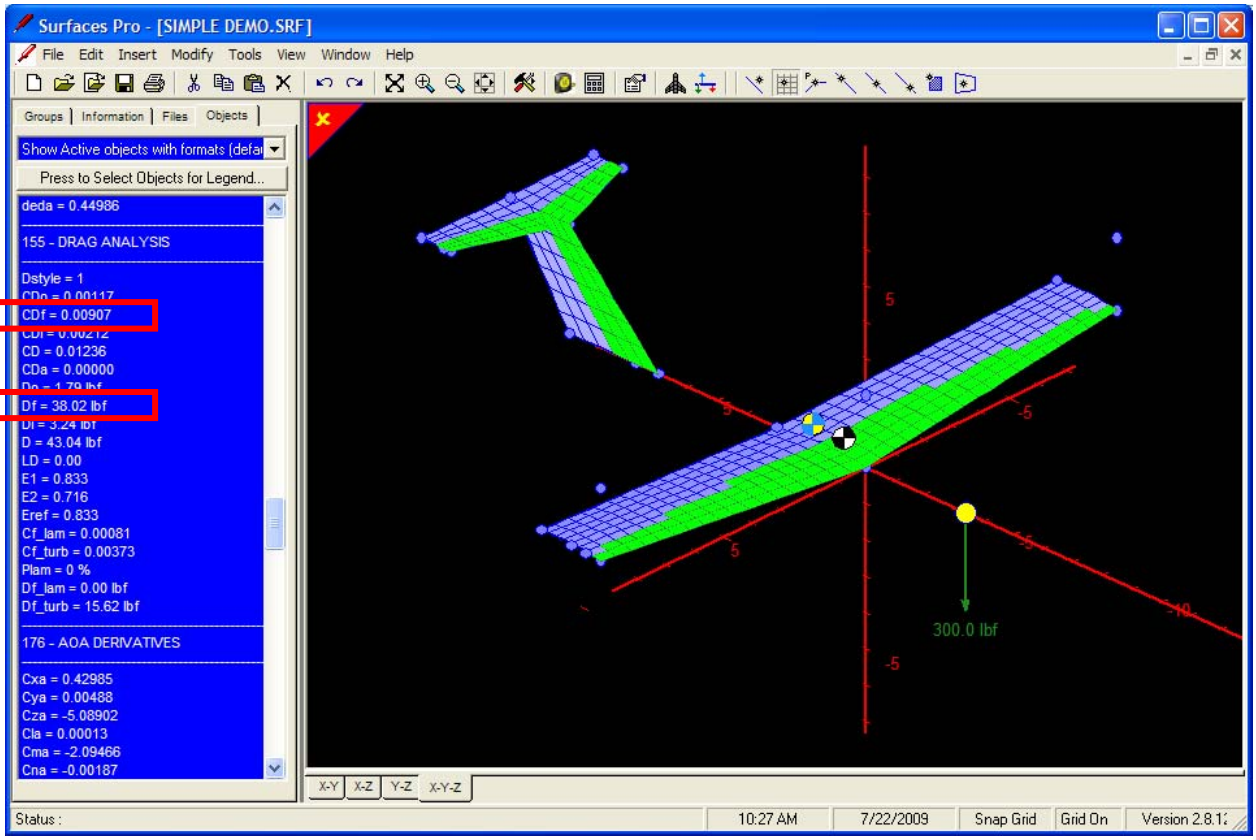


Figure 9.10-15: The model displaying the extent of laminar flow regions after Step 15 has been completed (green panels). Note the magnitude of the C_{Df} for the entire aircraft is 0.00907. This generates a skin friction drag of 38.02 lbf.

Furthermore, now that we have defined the drag for the airplane, we can learn a number of performance related things about it. This is done by creating the drag polar for the full airplane, but this is shown in Figure 9.10-17. It was obtained by running the Virtual Wind Tunnel (note that elevator deflection was set to 0°). Another interesting performance parameter obtained from the same VWT run is the L/D curve in Figure 9.10-18. From it we learn that the expected maximum L/D is 16.4 at an AOA of 6°.

We have just taken the first steps into a world of information about our design.

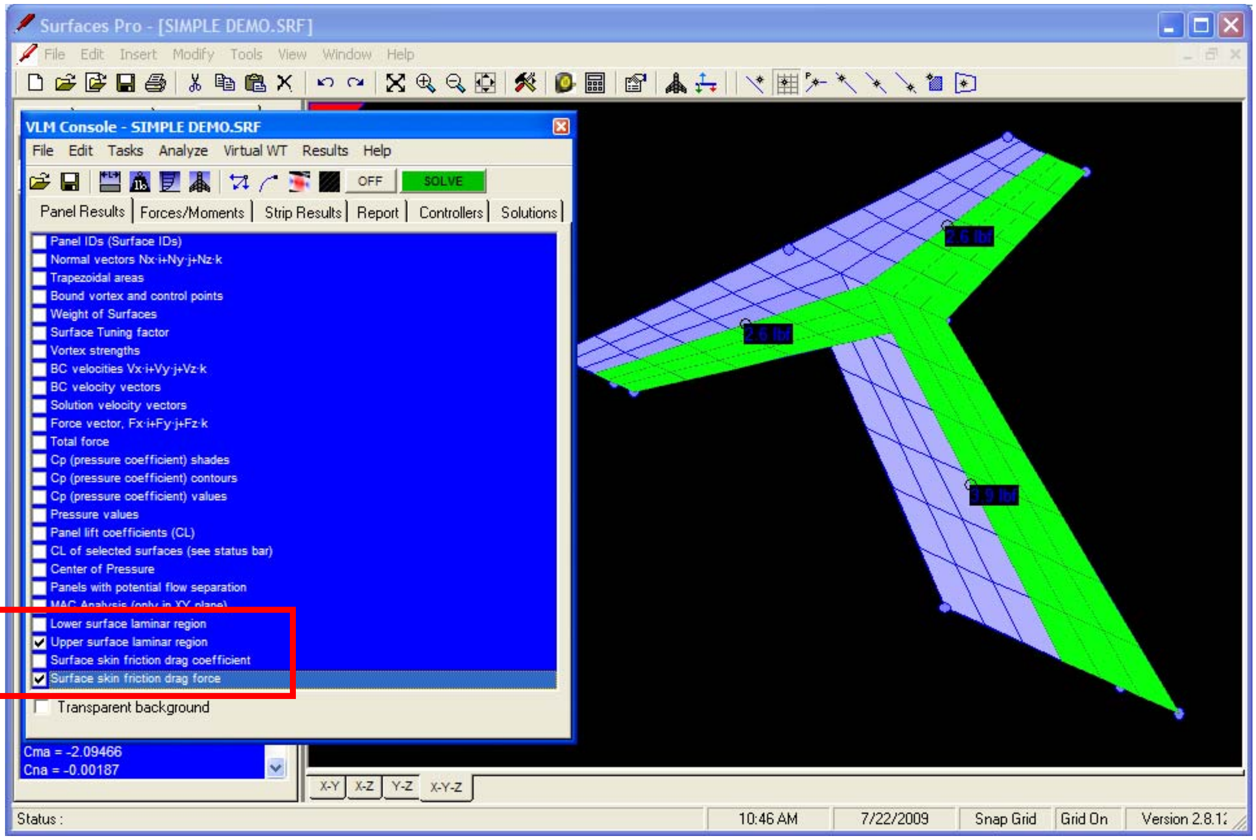


Figure 9.10-16: Displaying the skin friction drag on component basis.

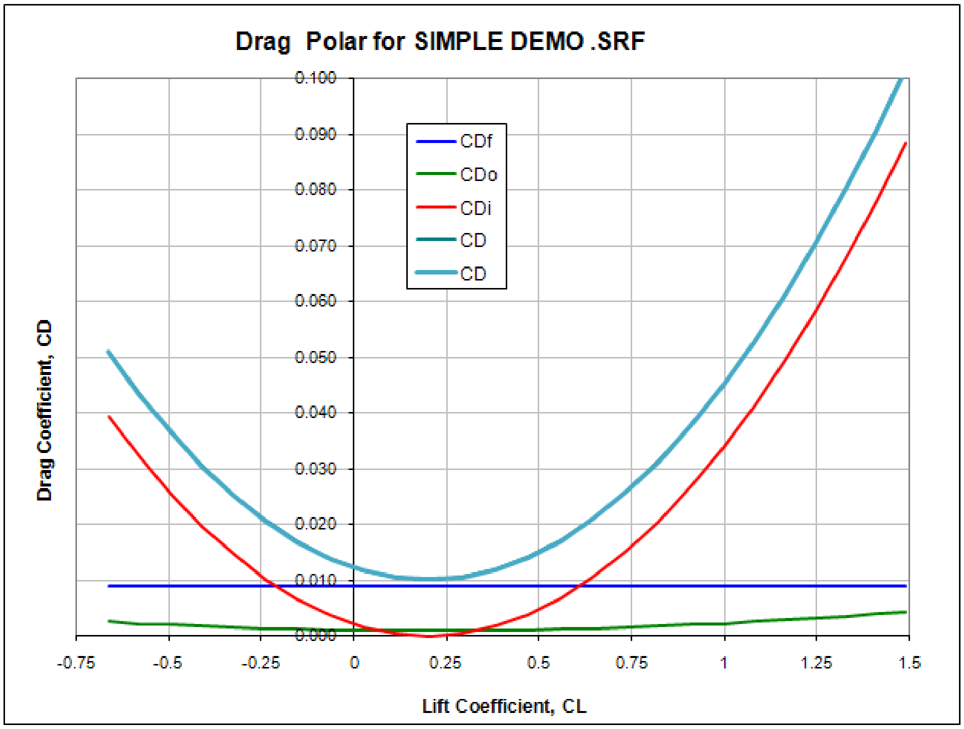


Figure 9.10-17: Drag polar generated by the Virtual Wind Tunnel for the example aircraft.

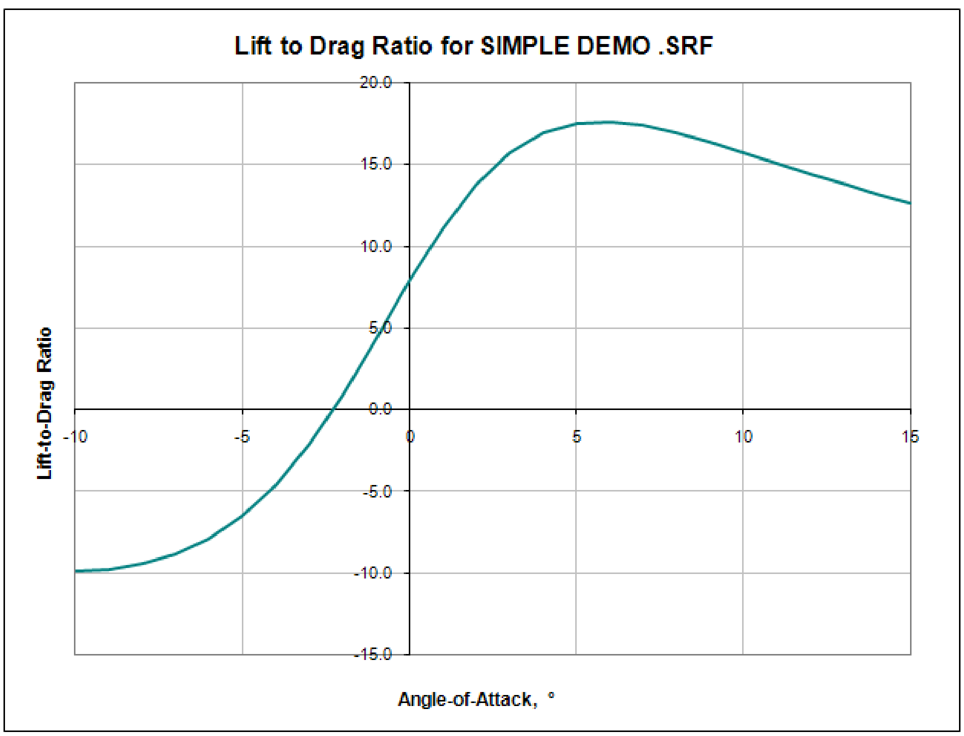


Figure 9.10-18: Variation of L/D with AOA, as generated by the Virtual Wind Tunnel.

9.11 Summary of SURFACES Drag Analysis Methodology

	Basic drag coefficient	Skin friction drag coefficient	Induced drag coefficient
Textbook representation	C_{Do}	C_{Df}	C_{Di}
SURFACES representation	`CDo`	`CDf`	`CDi`
Formulation allowed	User entry only	User entry or internal formulation	User entry or internal formulation
Internal formulation	No	Yes	Yes
Internal function name	-	[CDf]	[CDi]
How does it work?	Depends on user entry.	[CDf] returns the skin friction coefficient by summing up skin friction coefficients assigned to selected surfaces. The function calculates the area of the surface and multiplies with the user entered skin friction coefficient.	[CDi] returns the induced drag using one of three modeling techniques; surface integration, $k \cdot CL^2$ method, or Trefftz plane integration.
Affected by compressibility	Yes	Yes	Yes
Built-in compressibility	Yes	Yes	Yes
Can use Frankl-Voishel	No	Yes	No
Can use Prandtl-Glauert	Yes	No	Yes
Can use Karman-Tsien	No	No	Yes
Can use Laitone	No	No	Yes
Can use User Defined	No	No	Yes

References:

1. *Aircraft Performance and Design*. Anderson, John D., McGraw-Hill, 1999.
2. *Convair Performance Methods*.
3. *USAF DATCOM*. Hoak, D. E. et al, Flight Control Division, Air Force Flight Dynamics Laboratory, 1970.
4. *Aircraft Design: A Conceptual Approach*. Raymer, Daniel P., AIAA Education Series1989.
5. *Aerodynamics, Aeronautics, and Flight Mechanics*. McCormick, Barnes W., John Wiley & Sons, 1979.
6. *Modern Compressible Flow*. Anderson, John D., McGraw-Hill, XXXX.
7. *Airplane Aerodynamics and Performance*. Roskam, Jan, DARcorporation, 1997.
8. *Boundary Layers*. Young, A. D., AIAA Education Series, 1989.

10. Validation Samples

Validation 1: 2-D Flat Plate Airfoil

V1.1 Model

A high aspect ratio (AR=20) wing model was constructed to obtain 2-D pressure coefficients for comparison to theoretical data. The model has a wing span of 20 units and a chord of 1 unit. The Angle-of-Attack is 10° at an airspeed of 10 unit/sec and density of 1 mass unit/length^3. The Cp at the center of the model was obtained for 2, 5, 10, and 15 chord wise panel density. Each of the two surfaces has 34 span wise panels.

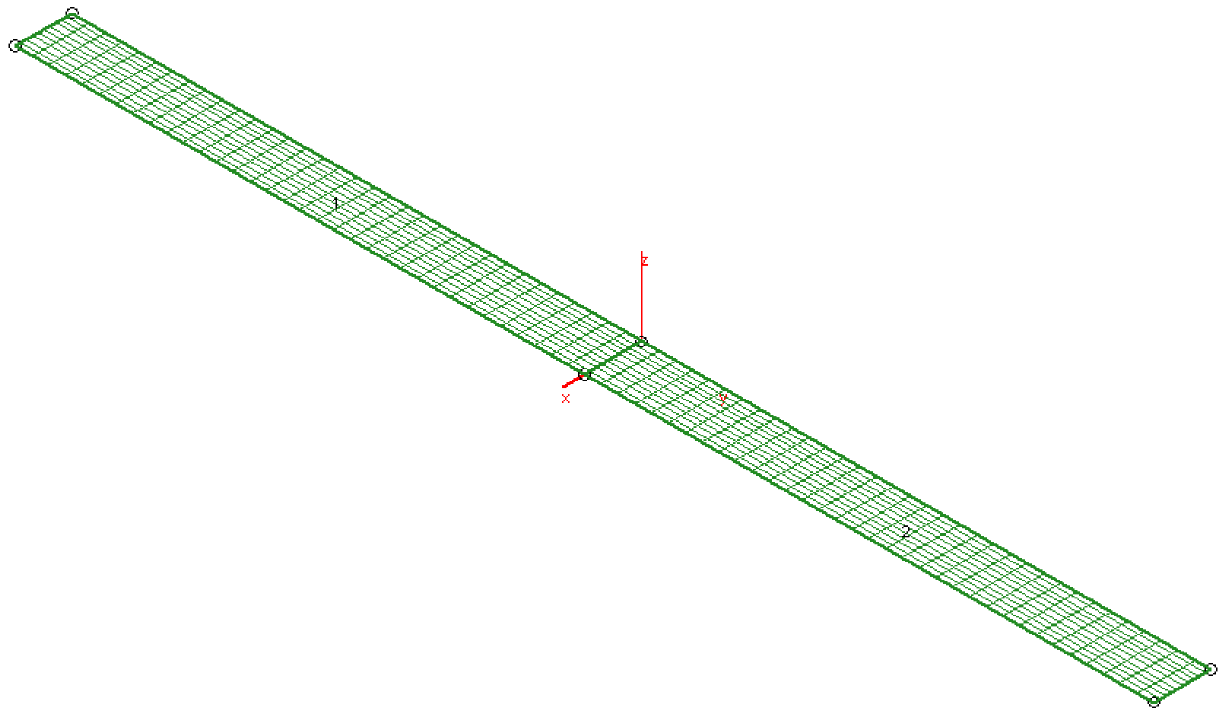


Figure 1-1: High aspect ratio wing used to evaluate the 2-D Cp.

V1.2 Expected Result

Is obtained from the book *Aerodynamics, Aeronautics, and Flight Mechanics*, by Barnes W. McCormick. The data is obtained from Figure 3.17 on page 87.

V1.3 Results from SURFACES

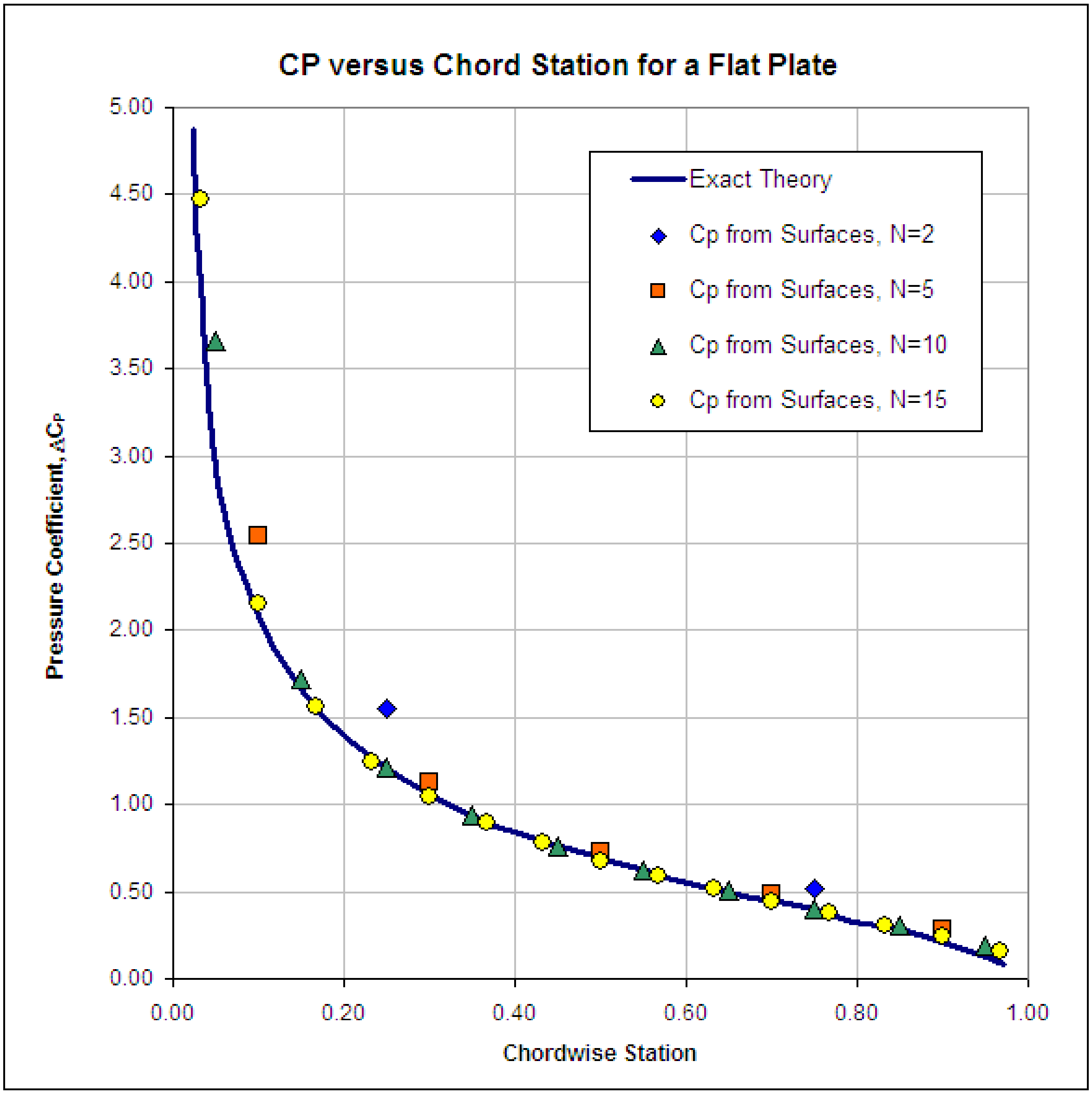


Figure 1-2: 2-D Cp for various panel densities from SURFACES compared to exact theory.

X	CP
0.02	4.88
0.03	4.47
0.03	4.22
0.03	4.01
0.04	3.45
0.04	3.23
0.05	2.93
0.07	2.52
0.10	2.06
0.13	1.77
0.17	1.54
0.21	1.35
0.26	1.16
0.34	0.96
0.39	0.85
0.46	0.75
0.54	0.64
0.60	0.55
0.68	0.47
0.74	0.41
0.79	0.33
0.86	0.27
0.92	0.18
0.97	0.08

Data from Figure 3.17

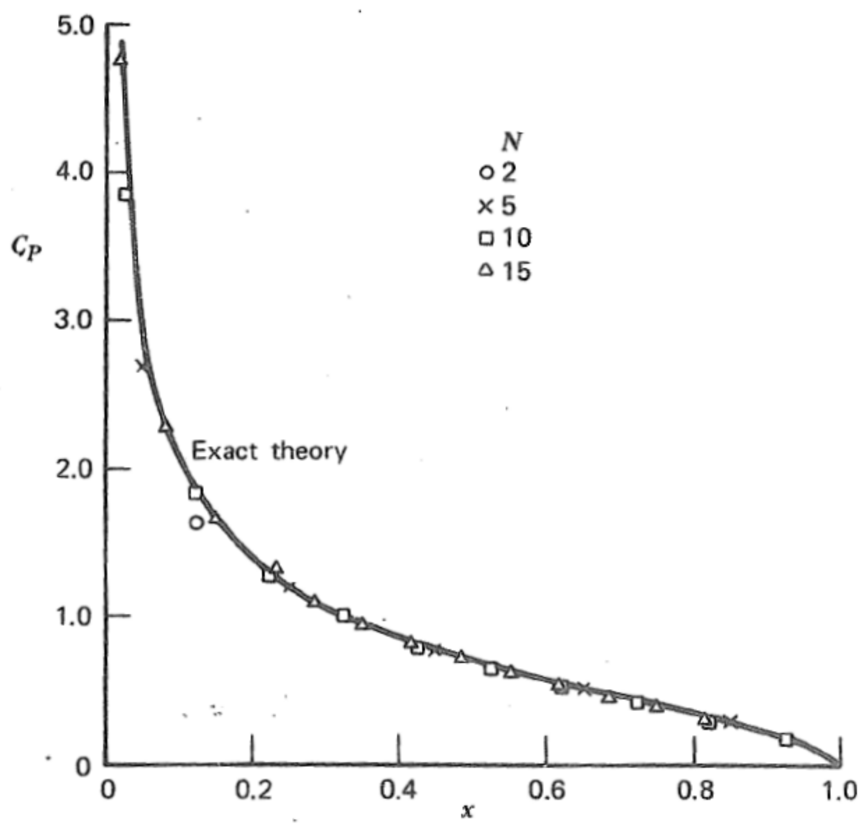


Figure 1-2: 2-D Cp from Figure 3-17 of reference document.

Validation 2: 3-D Properties of Two Wings

V2.1 Models

Two moderately high aspect ratio wing models were constructed to compare results from the VLM to a standard 3-D aerodynamic analysis. The models have a wing span of 10 ft and a chord of 1 ft. One model has a 0° leading edge sweep and the other 35°. The angle of attack is 10° at an airspeed of 100 KCAS (168.8 ft/s) and density of 0.002378 slugs/ft^3. Each of the two surfaces has 32 spanwise and 8 chord-wise panels.

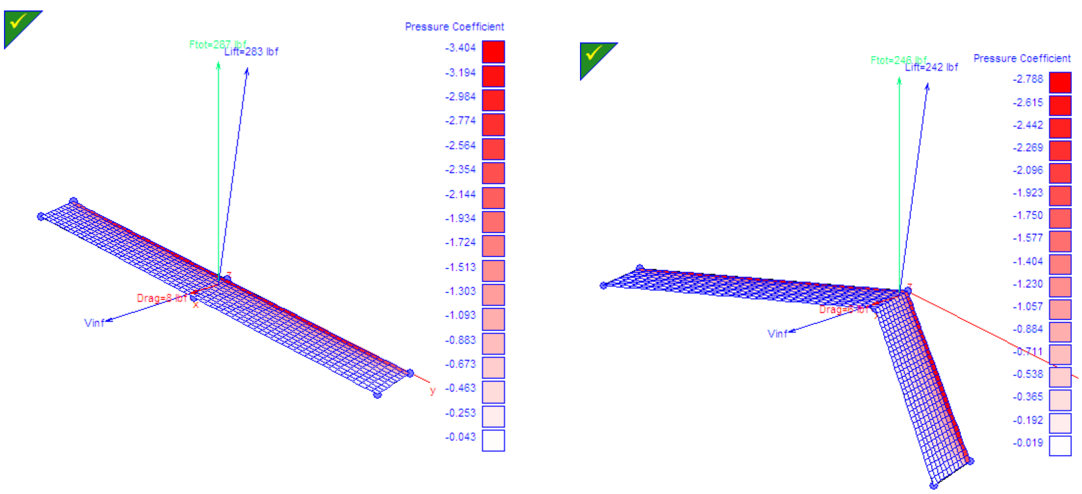


Figure 2-1: The two 3-D wing models.

V2.2 Expected Result

The following parameters are given:

Airspeed	V = 168.8 ft/s (M = 168.8/1116 = 0.151)
Wing area	S = 10 x 1 = 10 ft^2
Aspect Ratio	AR = b^2 / S = 10^2/10 = 10

Assume a 2-D lift curve slope of $C_{L_{\alpha 2D}} = 0.1063 \text{ per deg}$ (for NACA0009, from *Theory of Wing Sections*, by Abbott and Doenhoff). Start by computing a 3-D lift curve slope from Method 1 of USAF DATCOM Section 1, page 1-7.

$$C_{L_\alpha} = \frac{2\pi \cdot AR}{2 + \sqrt{\frac{AR^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2 \Lambda_{C/2}}{\beta^2}\right) + 4}}$$

Where; AR = Wing Aspect Ratio = 10

β = Mach number parameter (Prandtl-Glauert) = $(1-M^2)^{0.5}$ = 0.989

κ = Ratio of 2D lift curve slope to 2π = 0.1063 x (180/ π)/(2π) = 0.96934

$\Lambda_{c/2}$ = Sweepback of mid-chord = 0° and 35°

$\Lambda_{c/2} = 0°$	$\Lambda_{c/2} = 35°$
$C_{L_\alpha} = \dfrac{2\pi \cdot AR}{2+\sqrt{\dfrac{AR^2\beta^2}{\kappa^2}\left(1+\dfrac{\tan^2\Lambda_{C/2}}{\beta^2}\right)+4}} = \dfrac{2\pi\cdot 10}{2+\sqrt{\dfrac{100\times 0.989^2}{0.96934^2}\left(1+\dfrac{0}{0.989^2}\right)+4}}$ $= 5.068 \text{ per rad} = 0.08846 \text{ per deg}$	$C_{L_\alpha} = \dfrac{2\pi \cdot AR}{2+\sqrt{\dfrac{AR^2\beta^2}{\kappa^2}\left(1+\dfrac{\tan^2\Lambda_{C/2}}{\beta^2}\right)+4}} = \dfrac{2\pi\cdot 10}{2+\sqrt{\dfrac{100\times 0.989^2}{0.96934^2}\left(1+\dfrac{\tan^2(35°)}{0.989^2}\right)+4}}$ $= 4.286 \text{ per rad} = 0.07480 \text{ per deg}$
The lift coefficient at 10° is thus: $C_L = 10° \times C_{L_\alpha} = 0.8846$	The lift coefficient at 10° is thus: $C_L = 10° \times C_{L_\alpha} = 0.7480$
The total lift of the wing is $L = \tfrac{1}{2}\rho V^2 S C_L = \tfrac{1}{2}(0.002378)(168.8)^2(10)(0.8846) = 299.7\ lb_f$	The total lift of the wing is $L = \tfrac{1}{2}\rho V^2 S C_L = \tfrac{1}{2}(0.002378)(168.8)^2(10)(0.7480) = 253.4\ lb_f$
Induced drag is found from the standard relation $C_{Di} = \dfrac{C_L^2}{\pi\cdot AR} = \dfrac{(0.8846)^2}{\pi\cdot(10)} = 0.02491$ $D_i = \tfrac{1}{2}\rho V^2 S C_{Di} = \tfrac{1}{2}(0.002378)(168.8)^2(10)(0.02491) = 8.4\ lb_f$	Induced drag is found from the standard relation $C_{Di} = \dfrac{C_L^2}{\pi\cdot AR} = \dfrac{(0.7480)^2}{\pi\cdot(10)} = 0.01781$ $D_i = \tfrac{1}{2}\rho V^2 S C_{Di} = \tfrac{1}{2}(0.002378)(168.8)^2(10)(0.01781) = 6.0\ lb_f$
Lift to drag ratio: $\dfrac{L}{D} = \dfrac{0.8846}{0.02491} = 35.5$	Lift to drag ratio: $\dfrac{L}{D} = \dfrac{0.7480}{0.01781} = 42.0$

V2.3 Results from SURFACES

Summary for wing with 0° leading edge sweep*:

Parameter	Symbol	Classic Method	*SURFACES*
Lift curve slope	$C_{L\alpha}$	0.0885	0.0860
Lift coefficient	C_L	0.885	0.845
Induced drag coefficient	C_{Di}	0.0249	0.0227
Lift force	L	300 lb_f	286 lb_f
Induced drag force	D_i	8.4 lb_f	7.7 lb_f
Lift-to-drag ratio	L/D_i	35.5	37.2

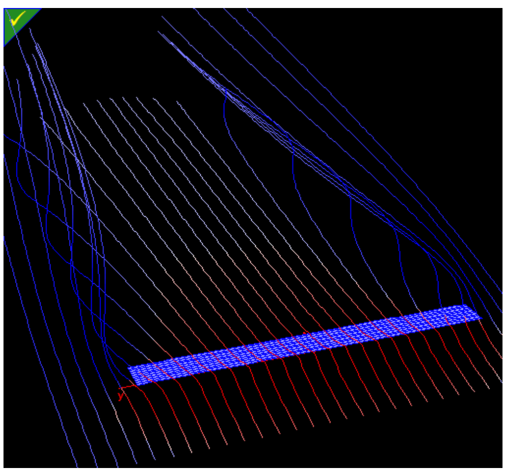
*Span efficiency for both cases is unrealistically assumed to be 1.

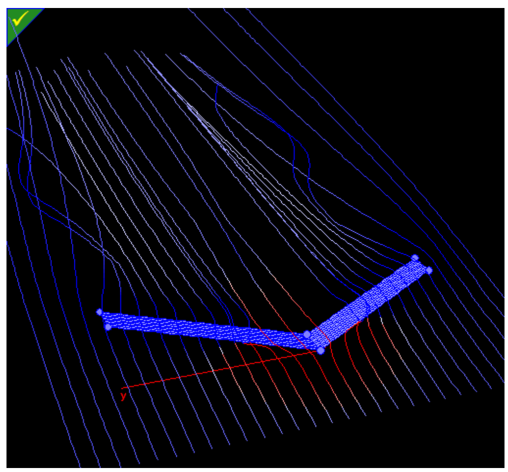
Summary for wing with 35° leading edge sweep*:

Parameter	Symbol	Classic Method	*SURFACES*
Lift curve slope	$C_{L\alpha}$	0.0748	0.07365
Lift coefficient	C_L	0.748	0.723
Induced drag coefficient	C_{Di}	0.0178	0.0166
Lift force	L	253 lb_f	245 lb_f
Induced drag force	D_i	6.0 lb_f	5.6 lb_f
Lift-to-drag ratio	L/D_i	42.0	43.5

*Span efficiency for both cases is unrealistically assumed to be 1.

Printout from **SURFACES**:





Great OWL Publishing
Engineering Software

Validation 3: Warren 12 Wing

V3.1 Model

The Warren-12 wing is a standard Vortex-Lattice model used to check the accuracy of vortex lattice codes. It provides a ready check case for the evaluation of any new or modified code, as well as a check on the panel scheme layout. This wing is known as the Warren 12 planform, and is defined, together with the “official” characteristics from previous calculations, in Fig. 3-1 below.

For the results cited, the reference chord used in the moment calculation is the average chord (slightly nonstandard, normally the reference chord used is the mean aerodynamic chord) and the moment reference point is located at the wing apex (which is also nonstandard).

“Published” Data:

$AR = 2.83$
$\Lambda_{LE} = 53.54°$
$C_{ref} = 1.00$
$X_{CG} = 0.00$
$S_{wing} = 2.83$
$C_{L\alpha} = 2.743$ / rad
$C_{M\alpha} = -3.10$ / rad

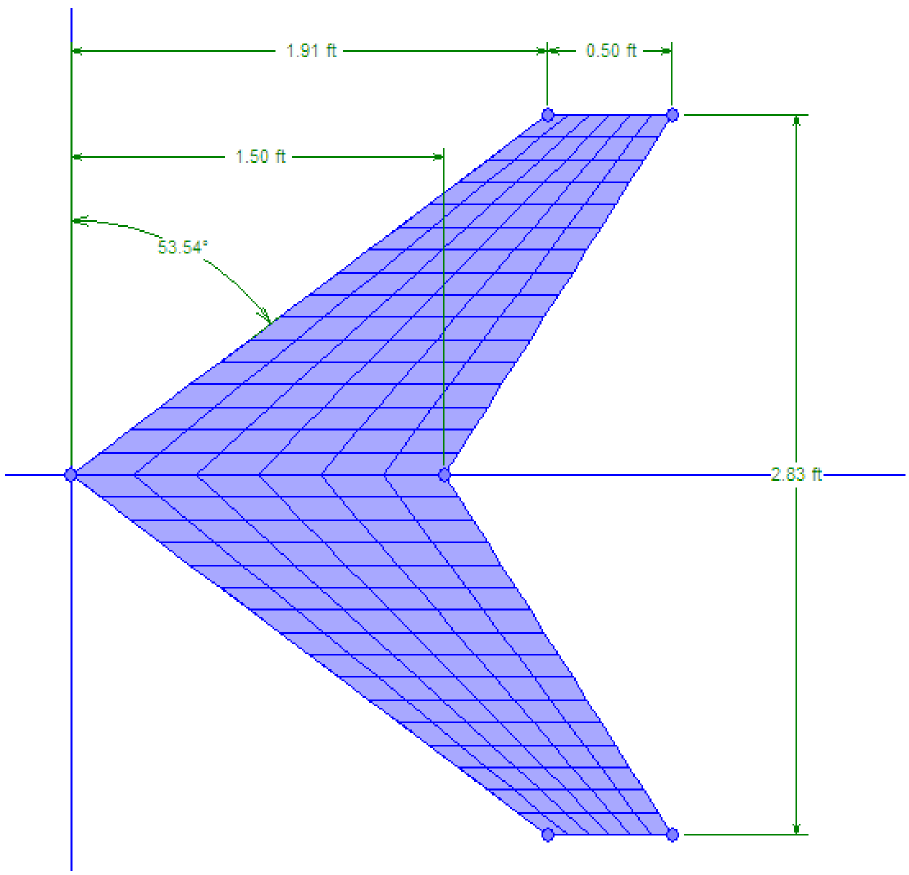


Figure 3-1: Warren-12 planform

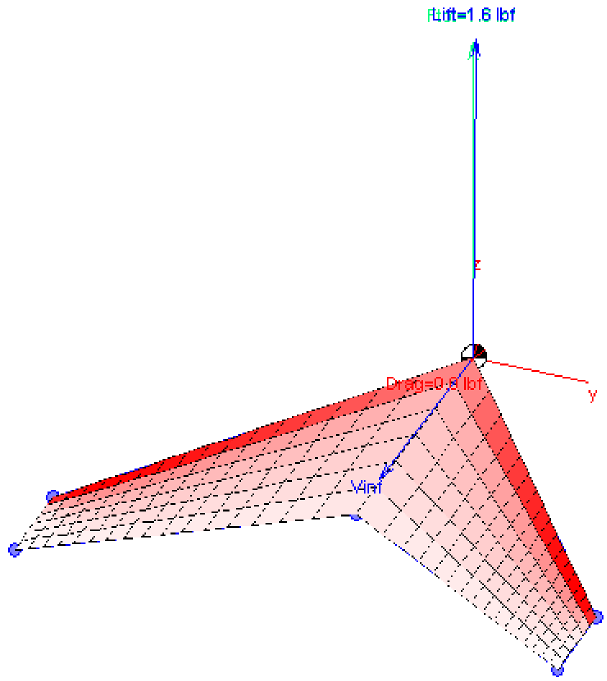


Figure 3-2: Warren-12 planform VL results

V3.2 Expected Result

The following results are expected:

$C_{L\alpha}$ = 2.743 / rad
$C_{M\alpha}$ = -3.10 / rad

*V3.3 Results from **SURFACES***

The following results where obtained from **SURFACES** for 6 chordwise by 16 spanwise panels on each wing (total of 192 panels):

$C_{L\alpha}$ = 2.790 / rad
$C_{M\alpha}$ = -3.174 / rad

The following results where obtained from **SURFACES** for 8 chordwise by 24 spanwise panels on each wing (total of 384 panels):

$C_{L\alpha}$ = 2.776 / rad
$C_{M\alpha}$ = -3.152 / rad

The following results where obtained from **SURFACES** for 16 chordwise by 36 spanwise panels on each wing (total of 1296 panels):

$C_{L\alpha}$ = 2.767 / rad
$C_{M\alpha}$ = -3.139 / rad

Validation 4 : Bertin-Smith 2-D Wing

V4.1 Model

Calculations for a highly swept back, high aspect ratio wing is provided in the text *Aerodynamics for Engineers* by Bertin and Smith. This wing has detailed calculations shown in Example 6-2 (page 198) in the text. The model in the text was recreated using **SURFACES**. Additionally, a comparison to another VLM code (*Tornado*) is made.

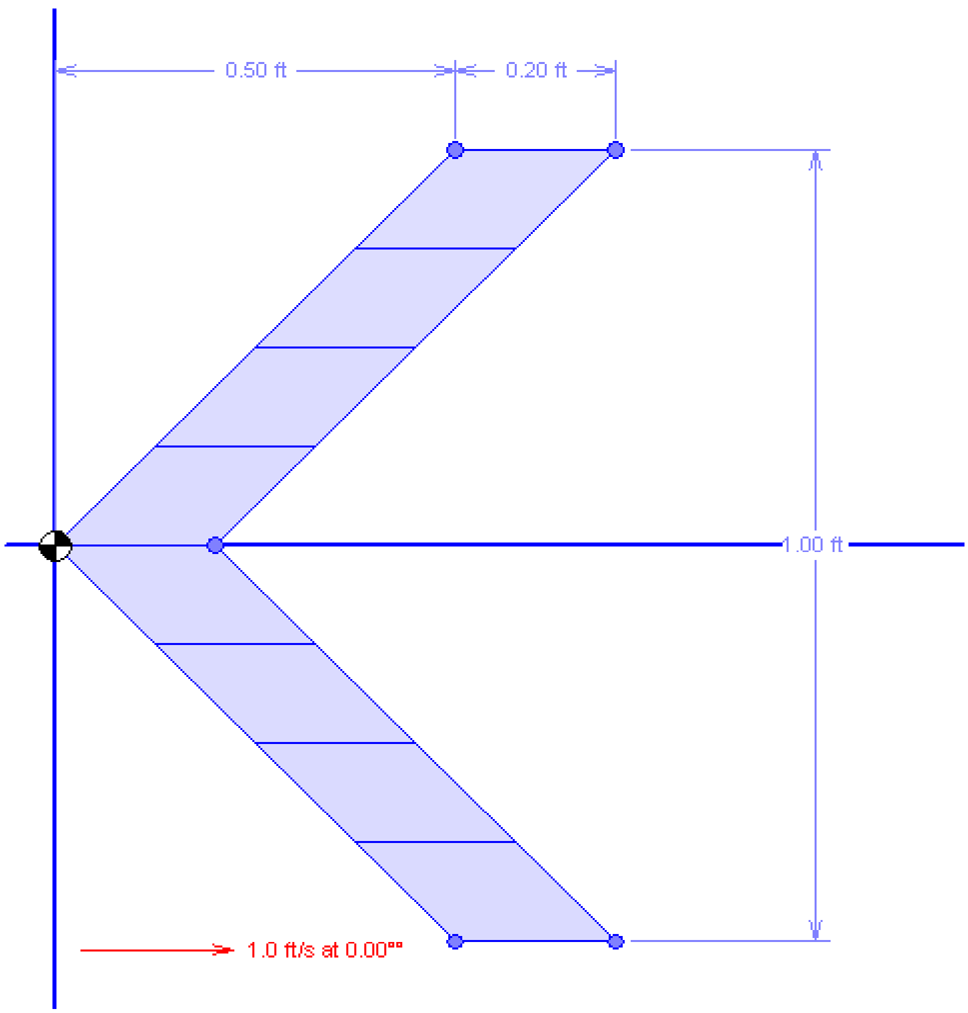


Figure 4-1: The Bertin-Smith swept back wing.

V4.2 Expected Result

Is obtained from the book *Aerodynamics for Engineers*, by Bertin and Smith. The data is obtained from the calculations on page 202, but the resulting lift curve slope is:

$$C_{L\alpha} = 0.05992 / ° = 3.433 / \text{rad}$$

V4.3 Results from SURFACES

The following results where obtained from **SURFACES** for 1 chordwise by 4 spanwise panels on each wing (total of 8 panels):

$$C_{L\alpha} = 0.06011\ /\ ° = 3.442\ /\ rad$$

SURFACES yields a difference of 0.26%. Another VLM code, called *Tornado*, considers the same problem. In his Master Thesis, "*A Vortex Lattice MATLAB Implementation for Linear Aerodynamic Wing Applications*" the author of *Tornado*, Mr. Tomas Melin, reports a lift curve slope of 3.450 /rad using *Tornado*. The difference using that code is 0.5%.

It can be seen that both codes are very close to the theoretical calculations in the source, but **SURFACES** yields less difference than *Tornado*. It should also be noted that the calculations in the source only carries 4 significant digits through the calculations – **SURFACES** uses a double floating point accuracy.

Summary:

Parameter	Symbol	Bertin-Smith	TORNADO	*SURFACES*
Lift curve slope	$C_{L\alpha}$	3.433	3.450 (0.50%)	3.442 (0.26%)

Validation 5: Cessna 172

Comparison of Several Codes

V5.1 Model

A model of the Cessna 172 was constructed to compare stability derivatives from **SURFACES** to other VLM codes (AVL, VIRGIT, TORNADO) and the panel code CMARC, as well as published Cessna data. The model has the camber line of the NACA 2412 airfoil of the Cessna 172. Additionally, it has a 1°30' angle-of-incidence at the root of the wing and –1°30' at the tip, and a 1°44' dihedral like the original airplane. A sweep of parameters was performed at an airspeed of 178.9 ft/s, at an altitude of 4921 ft (ρ = 0.002054 slugs/ft^3), and at a weight of 2207 lbs.

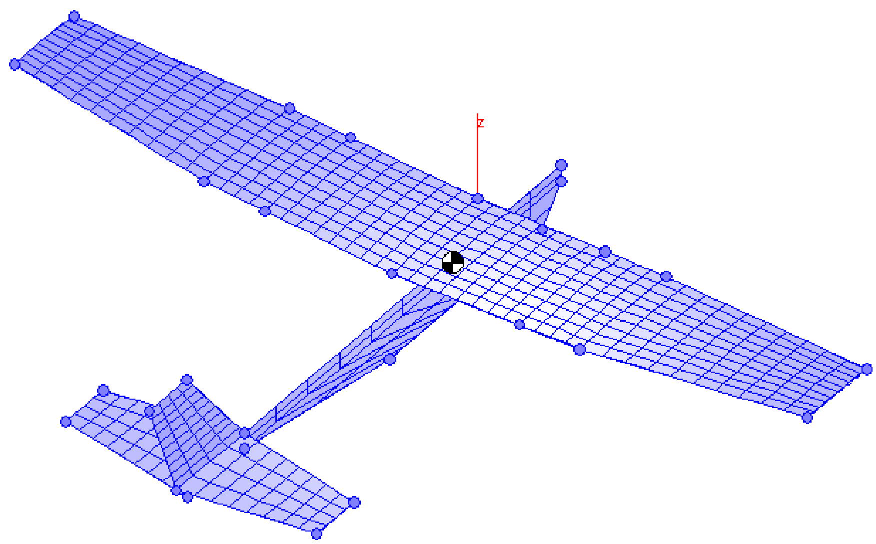


Figure 5-1: A Model of the C-172

V5.2 Expected Result

Range for $C_{L\alpha}$:
The following parameters are given:

Wing area S = 174 ft²
Aspect Ratio AR = b² / S = 36.08²/174 = 7.48

Assume a 2-D lift curve slope of $C_{L_{\alpha 2D}} = 0.107 \ \text{per deg}$ (for NACA 2412, from *Theory of Wing Sections*, by Abbott and Doenhoff, page 478).

Compute a 3-D lift curve slope from Method 1 of USAF DATCOM Section 1, page 1-7.

$$C_{L_\alpha} = \frac{2\pi \cdot AR}{2 + \sqrt{\frac{AR^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2 \Lambda_{C/2}}{\beta^2}\right) + 4}}$$

Where; AR = Wing Aspect Ratio = 7.48
β = Mach number parameter (Prandtl-Glauert) = $(1-M^2)^{0.5} \approx 1$
κ = Ratio of 2-D lift curve slope to 2π = 0.107 x $(180/\pi)/(2\pi)$ = 0.97572
$\Lambda_{c/2}$ = Sweepback of mid-chord = 0°

$$C_{L_\alpha} = \frac{2\pi \cdot AR}{2 + \sqrt{\frac{AR^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2 \Lambda_{C/2}}{\beta^2}\right) + 4}}$$

$$= \frac{2\pi \cdot 7.48}{2 + \sqrt{\frac{7.48^2}{0.97572^2}(1) + 4}} = 4.74 \text{ per rad} = 0.08267 \text{ per deg}$$

Range for $C_{n\beta}$:
Consider the following check for $C_{n\beta}$. The height, root, and tip chord of the fin is 5.50 ft, 4.25 ft, and 2.30 ft, respectively. The leading edge sweep is 40°. The airfoil is a NACA 0009 airfoil, whose properties are discussed in Validation Sample 2. Using this data we compute the following lift curve slope for the fin:

Fin area	$S_{fin} = ½ \cdot (4.25 + 2.30) \cdot 5.50 = 18.01$ ft²
Aspect Ratio	$AR = b_{fin}^2 / S_{fin} = 5.50^2/18.01 = 1.679$

Assume a 2-D lift curve slope of $C_{L_{\alpha 2D}} = 0.1063$ per deg (for NACA0009, from *Theory of Wing Sections*, by Abbott and Doenhoff).

Compute a 3-D lift curve slope from Method 1 of USAF DATCOM Section 1, page 1-7.

$$C_{L_\alpha} = \frac{2\pi \cdot AR}{2 + \sqrt{\frac{AR^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2 \Lambda_{C/2}}{\beta^2}\right) + 4}}$$

Where; β = Mach number parameter (Prandtl-Glauert) = $(1-M^2)^{0.5} \approx 1$
κ = Ratio of 2D lift curve slope to 2π = 0.1063 x $(180/\pi)/(2\pi)$ = 0.96934
$\Lambda_{c/2}$ = Sweepback of mid-chord ≈ 28°

$$C_{L_\alpha} = \frac{2\pi \cdot AR}{2 + \sqrt{\frac{AR^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2 \Lambda_{C/2}}{\beta^2}\right) + 4}}$$

$$= \frac{2\pi \cdot 1.679}{2 + \sqrt{\frac{1.679^2}{0.96934^2}\left(1 + \frac{\tan^2(28°)}{1^2}\right) + 4}} =$$

$$= \frac{10.54947}{2 + \sqrt{3.00019 \times 1.28271 + 4}} = 2.197 \text{ per rad} = 0.03835 \text{ per deg}$$

If one considers the fin at a β = 1°, the fin lift coefficient is given by $C_L = 1° \times C_{L_\alpha} = 0.03835$. The total lift of the fin at V = 178.9 ft/s and ρ = 0.002054 slugs/ft^3 is found to be

$$L_{fin} = \tfrac{1}{2}\rho V^2 S C_L = \tfrac{1}{2}(0.002054)(178.9)^2(18.01)(0.03835) = 22.7\ lb_f$$

Assuming a tail arm from reference point of 16.0 ft, the total moment is found to be 363.2 ft·lb$_f$, which yields a C_n of:

$$C_n = \frac{N}{\tfrac{1}{2}\rho V^2 S b} = \frac{363.2}{\tfrac{1}{2} \times 0.002054 \times 178.9^2 \times 174 \times 36.17} = 0.00176$$

Since N equals 0 ft·lb$_f$ at β = 0°, $C_{n\beta}$ can be found to be:

$$C_{n\beta} = \frac{\partial C_n}{\partial \beta} \approx \frac{0.00176}{1°} = 0.00176 \text{ per } ° = 0.1006 \text{ per rad}$$

From this, a reasonable $C_{n\beta}$ for this plane should be of the order of 0.03-0.17, depending on the contribution of other components of the airplane.

V5.3 Results from SURFACES

The following results where obtained from **SURFACES** and compared to that of other VLM codes. The data is obtained from the *Tornado* manual, pages 34-38. All the stability derivatives presented below are evaluated at α = 0.

TABLE 5-1: Stability Derivatives at α = 0:

	TEST[15]	AVL	VIRGIT	CMARC	TORNADO	SURFACES	NOTE
$C_{L\alpha}$	4.6	4.98	5.25	5.214	5.2763	5.128/5.1803	1
$C_{D\alpha}$	0.13	0	-0.005	0.086	-0.022	0.051/0.146	2
$C_{Y\alpha}$	-	0	0	0	0	0	-

[15] Comparison data is obtained from *Airplane Flight Dynamics and Automatic Flight Controls*, by Jan Roskam. Appendix C, page 592.

$C_{l\alpha}$	-	0	0	0	0	0	-
$C_{m\alpha}$	-0.89	-0.33	-0.85	-1.432	-1.498	-1.148/	3
$C_{n\alpha}$	-	0	0	0	0	0	-
$C_{L\beta}$	-	0	0	0	0	0	-
$C_{D\beta}$	-	0	0	0	0	0	-
$C_{Y\beta}$	-0.31	-0.26	-0.24	-0.104	-0.3	-0.370/-0.341	4
$C_{l\beta}$	-0.089	0.33	0.007	0.063	0.025	-0.0479/-0.045	5
$C_{m\beta}$	-	0	0	0	0	0	-
$C_{n\beta}$	0.065	0.092	0.1	0.042	0.12	0.117/0.0911	6
C_{Lp}	-	0	0	0	0	0	-
C_{Dp}	-	0	0	0	0	0	-
C_{Yp}	-	-0.066	-0.1	-0.015	-0.039	-0.110	7
C_{lp}	-0.47	-0.325	-0.52	-0.995	-0.526	-0.510/-0.508	7
C_{mp}	-	0	0	0	0	0	-
C_{np}	-0.03	-0.007	-0.01	-0.133	-0.006	-0.0056/0.018	7
C_{Lq}	-	9.41	9.3	9.003	10.18	7.894/9.111	8, 9
C_{Dq}	-	0	0	0	0.128	0.432/0.256	8, 10
C_{Yq}	-	0	0	0	0	0	-
C_{lq}	-	0	0	0	0	0	-
C_{mq}	-12.4	-14.43	-15	-17.155	-14.96	-12.156/14.3	8, 9
C_{nq}	-	0	0	0	0	0	-
C_{Lr}	-	0	0	0	0	0/0.0029	-
C_{Dr}	-	0	0	0	0	0	-
C_{Yr}	0.21	0.209	0.23	0.45	0.271	0.296/0.306	7, 9
C_{lr}	0.096	0.021	0.008	0.195	0.009	0.101/0.0926	7, 9
C_{mr}	-	0	0	0	0	0	-
C_{nr}	-0.099	-0.075	-0.095	-0.212	-0.11	-0.115/-0.119	

All derivatives are per radian.

NOTES:

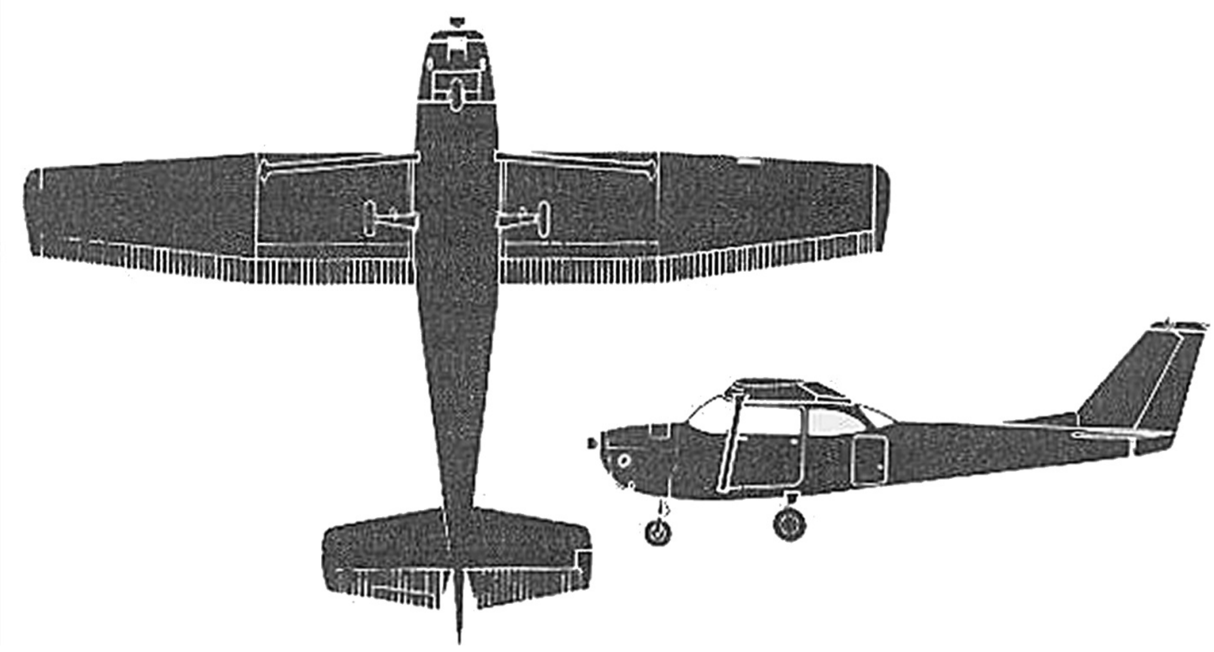
(1) There is a known difference in input geometry, which will likely cause numerical discrepancies. It is not known if the other VLM codes included washout, dihedral, and wing camber like the **SURFACES** model.

(2) A value of zero is expected at $C_L = 0$ only if the airfoil of the wing is symmetrical (flat plate).

(3) The different values are primarily due to the different reference locations, but also due to possible power effects. For instance, **SURFACES** and *VIRGIT* use 29.5% of MAC, *Tornado* uses 31.9% MAC. *AVL* and *CMARC* reference points are unknown. **SURFACES** has the reference point located 2 ft below the wing plane and does not account for power effects – it is unknown where the other codes place the vertical location of the reference point, or if propeller normal force is accounted for.

(4) Note that for **SURFACES** the standard coordinate system is used with the Angle-of-Yaw (positive beta) coming from the left, rather than the right. Consequently, a sign change is added to compare to

the other codes.
(5) **SURFACES** evaluated a restoring dihedral effect for the C-172 – the only one of the above codes.
(6) [Deleted]
(7) The rate of roll and yaw derivatives are obtained with respect to P·Bref/(2·Vinf). For that reason, derivatives with respect to P or R are multiplied by the factor 2·Vinf/Bref.
(8) The rate of pitch derivative is obtained with respect to P·Cref/(2·Vinf). For that reason, derivatives with respect to Q are multiplied by the factor 2·Vinf/Cref.
(9) Differences are most likely due to modeling differences and differences in location of reference point.
(10) A change in lift should be associated with a change in drag. It is not known why *Tornado* and **SURFACES** are the only codes to display a value here.



V5.4 Comparison of Codes

Table 5-1 prompts some interesting questions – for instance, how do the codes compare? Table 5-2 displays one such comparison. Here, a grade from 1 (worst) to 5 (best) is assigned to those stability parameters that can be compared to the source. The parameters are compared by computing difference using:

$$\Delta difference = \left| \frac{P_{CODE} - P_{SOURCE}}{P_{SOURCE}} \right|$$

Then, the code with the largest difference scores 1 and the code with the smallest one 5. A total of 30 derivatives are considered in Table 5-1, of which 12 have a value from the source document (*Airplane Flight Dynamics and Automatic Flight Controls*, by Jan Roskam). The highest total score a code can receive is 5 x 12 = 60. The lowest total score is 12. The scores for the 5 codes are compared in Table 5-2:

Table 5-2: Comparison of Several VLM Codes and the Panel Code CMARC.

	AVL	VIRGIT	CMARC	TORNADO	SURFACES
$C_{L\alpha}$	5	2	3	1	4
$C_{D\alpha}$	3	2	5	1	4
$C_{m\alpha}$ (CMy_{α})	2	5	3	1	4
$C_{Y\beta}$	4	2	1	5	3
$C_{l\beta}$ (CMx_{β})	1	4	2	3	5
$C_{n\beta}$ (CMz_{β})	4	3	5	1	2
C_l, P (CMx, P)	2	4	1	3	5
C_n, P (CMz, P)	4	5	1	3	2
C_m, Q (CMy, Q)	4	2	1	3	5
C_Y, R	5	4	1	3	2
C_l, R (CMx, R)	4	2	1	3	5
C_n, R (CMz, R)	2	5	1	4	3
TOTAL SCORE	40	40	25	31	44
Number of 1s	1	0	7	4	0
Number of 2s	3	5	1	0	2
Number of 3s	1	1	1	6	2
Number of 4s	5	3	0	1	3
Number of 5s	2	3	2	1	4

Table 5-2 shows that **SURFACES** scores highest (44 points). CMARC scored worst (25 points). Two codes never scored worse than 2, VIRGIT and **SURFACES**. On the other hand, AVL, CMARC, and TORNADO all have at least one worst score. The most frequent low score (1) was received by CMARC, 7 times. The most frequent high score (5) was received by **SURFACES**, 4 times. In fact, **SURFACES** was the only code to correctly compute a restoring dihedral effect for the Cessna 172.

Validation 6: 2-D C_L, C_D, C_M for NACA 23012

V6.1 Model

A high aspect ratio (AR=20) wing model was constructed to perform a 3-D similarity evaluation to a standard 3-D aerodynamic analysis. The model has a wing span of 20 ft and a chord of 1 ft. An angle sweep of attack from –8° through 8° at an airspeed of 100 ft/sec and density of 0.002378 slugs/ft^3 was performed. The model has 16 chordwise and 60 spanwise panels. The panels form the camber line of the NACA 23012 airfoil. The purpose of this validation is to demonstrate how **SURFACES** simulates airfoil properties.

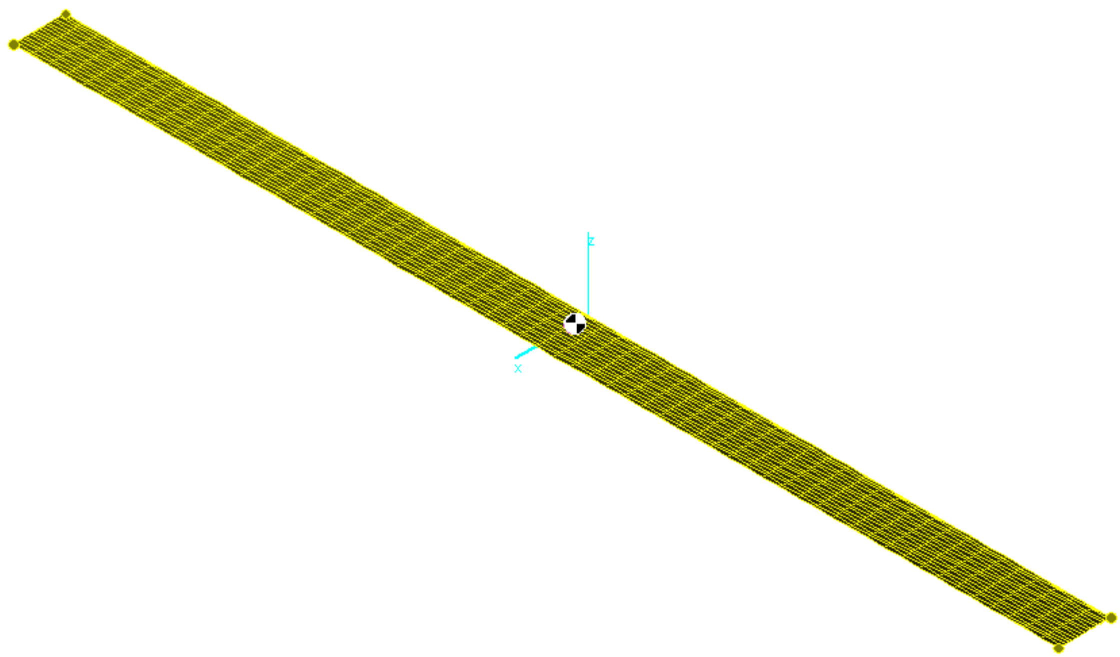


Figure 6-1: 3-D wing model with a 23012 airfoil

V6.2 Expected Result

The following parameters are given:

Airspeed	V = 100 ft/s
Wing area	S = 20 x 1 = 20 ft²
Aspect Ratio	AR = b² / S = 20²/10 = 20

The 2-D lift curve slope of $C_{l_\alpha} = 0.1051 \text{ per deg}, C_{l_0} = 0.1233$, and $C_{m_\alpha} = 0.00020\alpha - 0.01198$ is obtained from interpolation (for NACA 23012, from *Theory of Wing Sections*, by Abbott and Doenhoff).

Compute a 3-D lift curve slope from Method 1 of USAF DATCOM Section 1, page 1-7.

$$C_{L_\alpha} = \frac{AR \cdot C_{l_\alpha}}{2 + \sqrt{\frac{AR^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2 \Lambda_{C/2}}{\beta^2}\right) + 4}}$$

Where; AR = Wing Aspect Ratio = 20
β = Mach number parameter (Prandtl-Glauert) = $(1-M^2)^{0.5} \approx 1$
κ = Ratio of 2D lift curve slope to $2\pi = 0.1051 \times (180/\pi)/(2\pi) = 0.95840$
$\Lambda_{c/2}$ = Sweepback of mid-chord = 0°

$$C_{L_\alpha} = \frac{2\pi \cdot AR}{2 + \sqrt{\frac{AR^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2 \Lambda_{C/2}}{\beta^2}\right) + 4}}$$

$$= \frac{2\pi \cdot 20}{2 + \sqrt{\frac{400}{0.95840^2}\left(1 + \frac{0}{1^2}\right) + 4}} = 5.472 \text{ per rad} = 0.09551 \text{ per deg}$$

Compute zero lift angle for the 2D airfoil using: $C_{l_0} = -\alpha_0 \cdot C_{l_\alpha} \quad \Leftrightarrow \quad \alpha_0 = -\frac{C_{l_0}}{C_{l_\alpha}} = -\frac{0.1233}{0.1051} = \text{-}1.173$

Compute lift at zero angle for the 3D wing using: $C_{L_0} = -\alpha_0 \cdot C_{L_\alpha} = -(-1.173) \cdot 0.09551 = 0.1121$

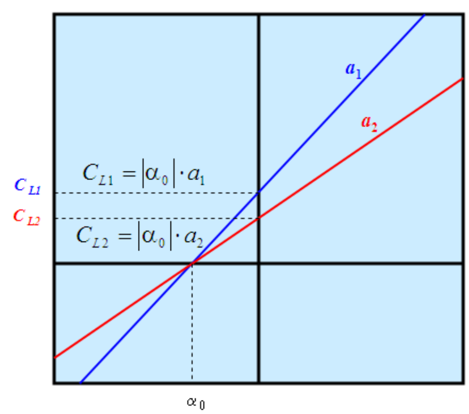
Compute pitching moment for 3D wing:

$$C_{m_\alpha} = C_{l_\alpha} \cdot \Delta x \quad \Leftrightarrow \quad \Delta x = \frac{C_{m_\alpha}}{C_{l_\alpha}}$$

$$\left(C_{m_\alpha}\right)_{3D} = C_{L_\alpha} \cdot \Delta x = C_{L_\alpha} \cdot \left(\frac{C_{m_\alpha}}{C_{l_\alpha}}\right)$$

$$= (0.09551) \cdot \left(\frac{-0.01198}{0.1051}\right)$$

$$= -0.01089$$



V6.3 Results from SURFACES

Summary:

Parameter	Symbol	Experiment[16]	Classical Method	*Surfaces*
Lift curve slope	$C_{L\alpha}$	0.1051 (2-D value)	0.0955	0.0943
Lift coefficient intercept	C_{L0}	0.1233	0.1121	0.1194
Moment coefficient	$C_{M\alpha}$	$0.00020 \cdot \alpha$ - 0.01198		$0.00016 \cdot \alpha$ - 0.01888

[16] *Theory of Wing Sections*, by Abbott & Doenhoff, graph on page 498.

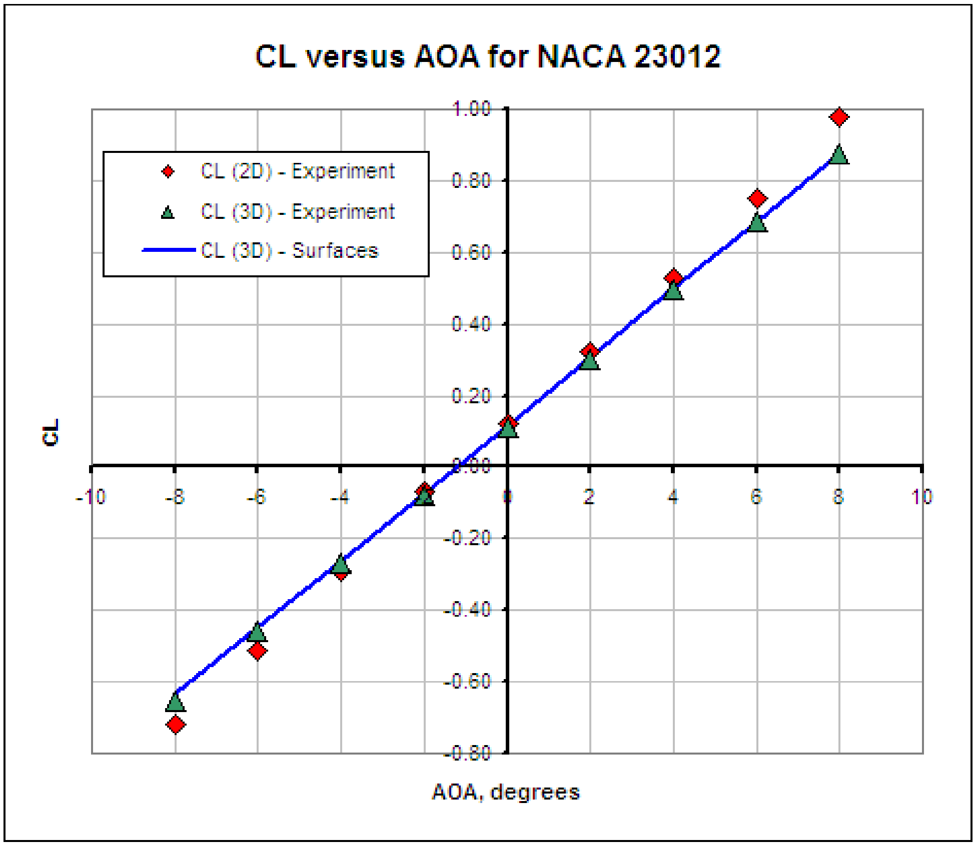


Figure 6-2: 3-D wing model with a 23012 airfoil camber line.

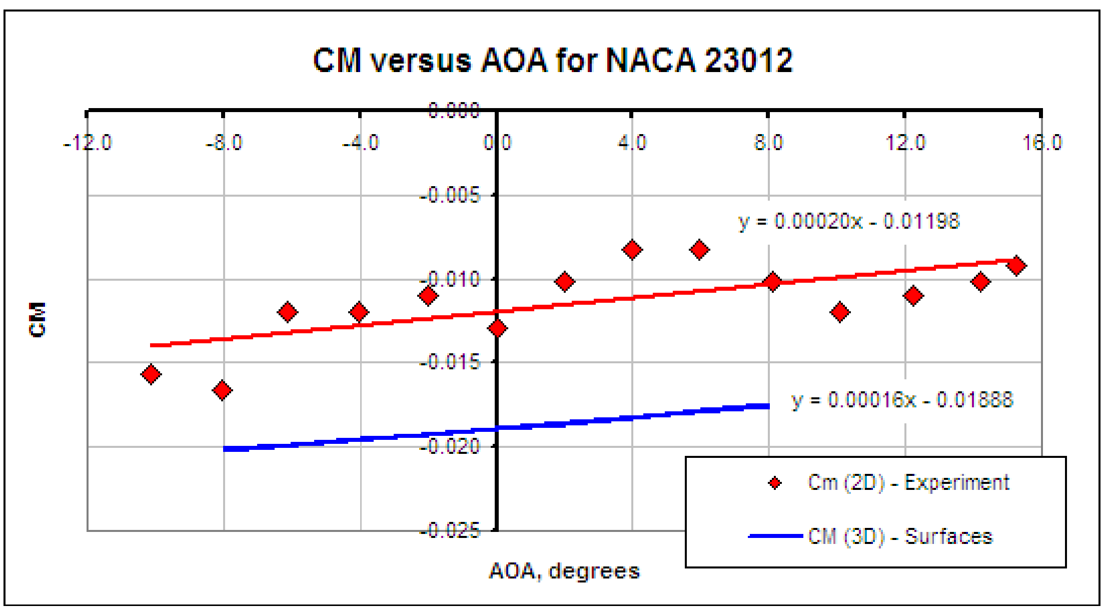


Figure 6-3: 3-D wing model with a 23012 airfoil camber line.

Validation 7: F-104 Starfighter

V7.1 Model

A model of the Lockheed F-104 Starfighter was constructed to compare selected stability derivatives from **SURFACES** to that presented in the text *Flight Stability and Automatic Control*, by Robert C. Nelson. The data can be found in Appendix B of the text, on page 253.

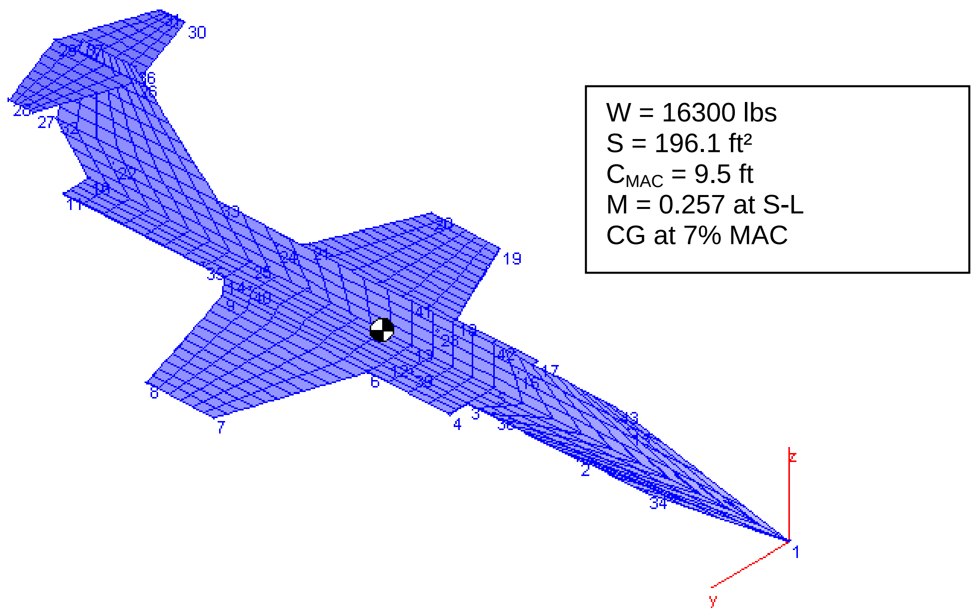


Figure 7-1: 3-D Vortex-Lattice model of the F-104 Starfighter

V7.2 Results from SURFACES

Summary:

Parameter	Symbol	Source[17]	SURFACES	%Difference
Lift coefficient	C_L	0.735	0.717[18]	2.4%
Drag coefficient	C_D	0.263	0.175[19]	33.5%
Lift curve slope	$C_{L\alpha}$	3.44	3.36	2.4%
Drag curve slope	$C_{D\alpha}$	0.45	0.66[20]	45.8%
Moment slope (CMyα)	$C_{M\alpha}$	-0.64	-0.756	18.1%
Side force slope (CFyβ)	$C_{Y\beta}$	-1.17	-1.12	4.3%
Dihedral effect (CMxβ)	$C_{l\beta}$	-0.175	-0.156	10.9%
Weathercock stability (CMzβ)	$C_{n\beta}$	0.50	0.491	1.8%

All derivatives are per radian. At M=0.257

[17] *Flight Stability and Automatic Control*, by Robert C. Nelson.
[18] Note that V = 0.257 x 1116 ft/s = 286.8 ft/s. Therefore, Lift is ½ ·0.002378·286.8² ·196.1·0.735 = 14097 lbf. This is the same lift SURFACES generated to get the given lift coefficient.
[19] Using the *surface integration* method
[20] This is highly dependent on drag model. Here, $C_{D\alpha} = (0.0009474\cdot\alpha - 0.0004737)*180/\pi$, which at $\alpha = 12.6$ becomes 0.656.

Figure 7-2: A Starfighter in flight

Image from http://www.starfighters.net/gallery/1999gallery/1999gallery.html

Validation 8: Ryan Navion

V8.1 Model

A model of the Ryan Navion was constructed compare to the analysis of Example Problem 2.1 found in Robert C. Nelsons "*Flight Stability and Automatic Control*", on pages 53-58. The VL model was based on the three-view in Figure 8-1.

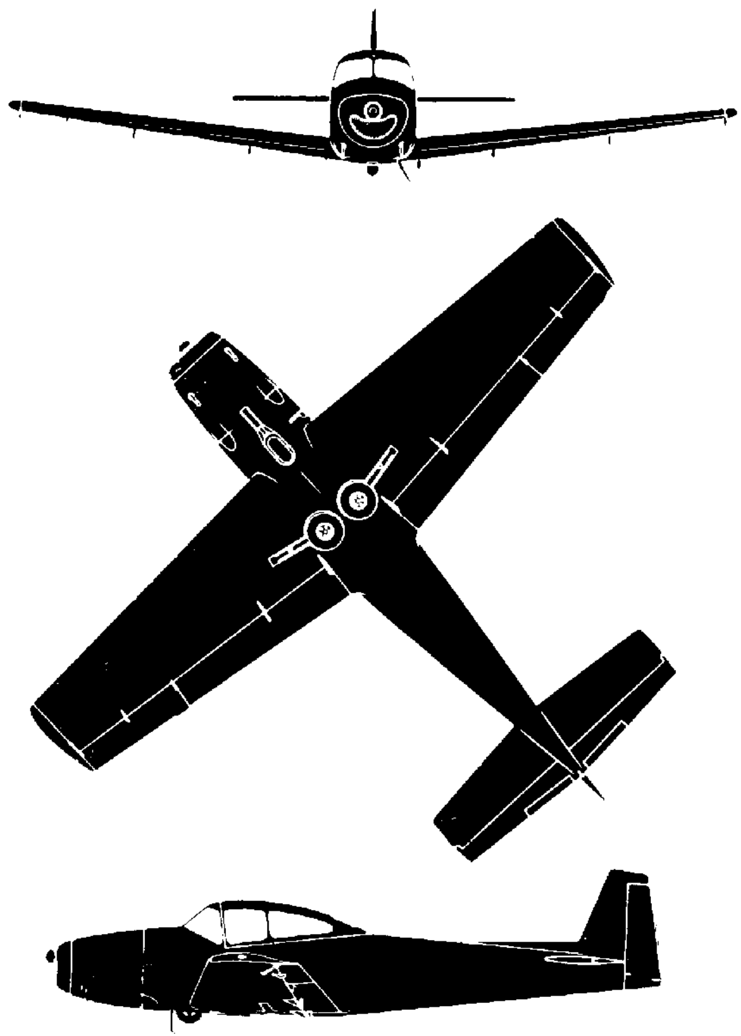


Figure 8-1. A three-view drawing of the Ryan Navion.

The reference document determines several parameters for the Navion in Problem 2.1. The calculation of selected parameters is repeated in Section V8.1 for convenience.

Figure 8-2. A Ryan Navion in flight – Photographer unknown.

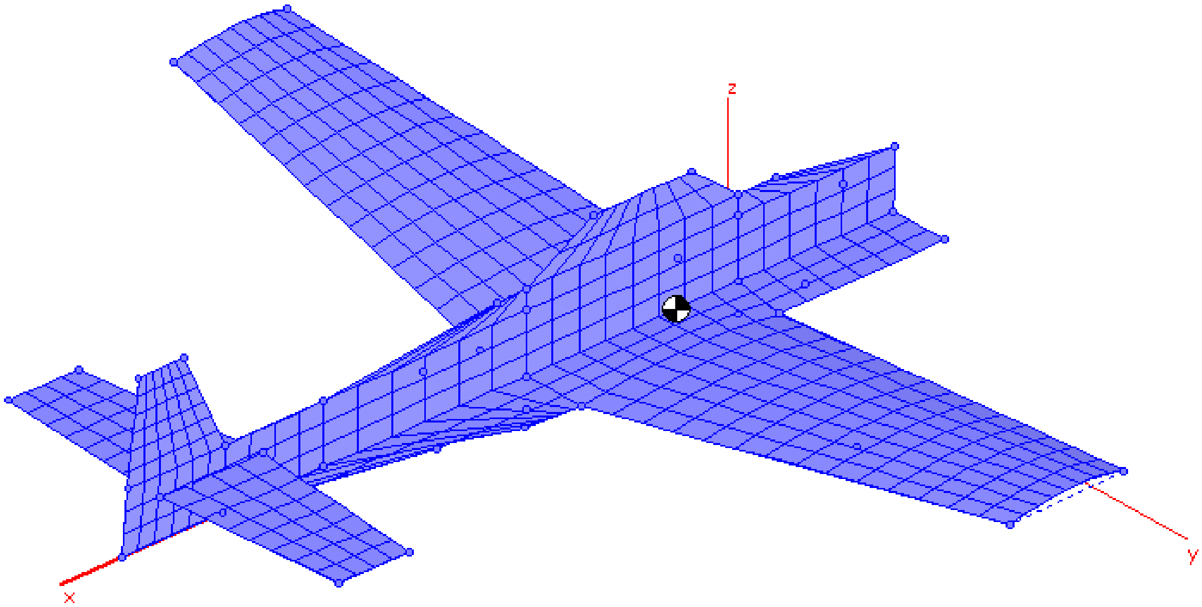


Figure 8-3. The SURFACES Vortex-Lattice model of the Ryan Navion.

V8.2 Expected Result

The stick-fixed neutral point is estimated from Equation (2.37) in the reference document, here written using variables more consistent with this document:

$$\frac{X_{NEU}}{C_{REF}} = \frac{X_{AC}}{C_{REF}} - \frac{C_{M_{\alpha f}}}{C_{L_{\alpha w}}} + \eta V_{HT} \frac{C_{L_{\alpha t}}}{C_{L_{\alpha w}}} \left(1 - \frac{d\varepsilon}{d\alpha}\right)$$

Where (note that numerical values are obtained from the document);

> C_{REF} = Reference wing chord = 5.7 ft
> $C_{L\alpha w}$ = Slope of wing lift coefficient = 4.3 per rad
> $C_{L\alpha t}$ = Slope of HT lift coefficient = 3.91 per rad
> $C_{M\alpha f}$ = Slope of fuselage moment coefficient = 0.12 per rad
> $d\varepsilon/d\alpha$ = Variation of downwash with angle-of-attack = 0.45
> V_{HT} = Horizontal tail volume = 0.66
> X_{AC} = Aerodynamic center of wing-body combination = 1.425 ft
> η = Tail efficiency = 1

$$\frac{X_{NEU}}{C_{REF}} = \frac{1.425}{5.7} - \frac{0.12}{4.3} + (1)(0.66)\frac{3.91}{4.3}(1 - 0.45) = 0.552$$

Note that the reference document (which is a First Edition) states the Xneu is at 0.37, but in conversation with the author (R. C. Nelson) it was confirmed this was an error that had been corrected for later editions of the book.

Note that the planform properties of the VL model were determined using **SURFACES**' built in tool, which printed out the following analysis report:

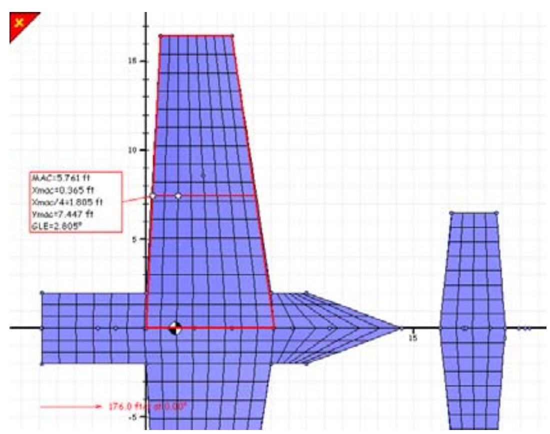
```
MEAN AERODYNAMIC CHORD ANALYSIS

Surface chord, root ..............    Cr = 7.200 ft
Surface LE, root .................    Xr = 0.000 ft
                 .................    Yr = 0.000 ft

Surface chord, tip ...............    Ct = 4.022 ft
Surface LE, tip ..................    Xr = 0.806 ft
                 .................    Yr = 16.446 ft

Surface half span ................ Bhalf = 16.446 ft
Surface span .....................     B = 32.893 ft
Surface half area ................ Shalf = 92.28 ft²
Surface  total  area  ...............   Stot  =  184.56
ft²

Surface LE sweep angle ...........   GLE = 2.805°
Surface aspect ratio .............    AR = 5.8621
Surface taper ratio ..............    TR = 0.5586
Surface Mean Aerodynamic Chord ...  Cmac = 5.761 ft
Surface MAC location .............  Xmac = 0.365 ft
                     .............  Ymac = 7.447 ft
```



This information can be used when calculating the CG and neutral point locations as percentages of the Mean Aerodynamic Chord (MAC). For instance, the CG located at Xcg = 2.0465 ft becomes 100·(2.0465-0.365)/5.7 = 29.5% MAC.

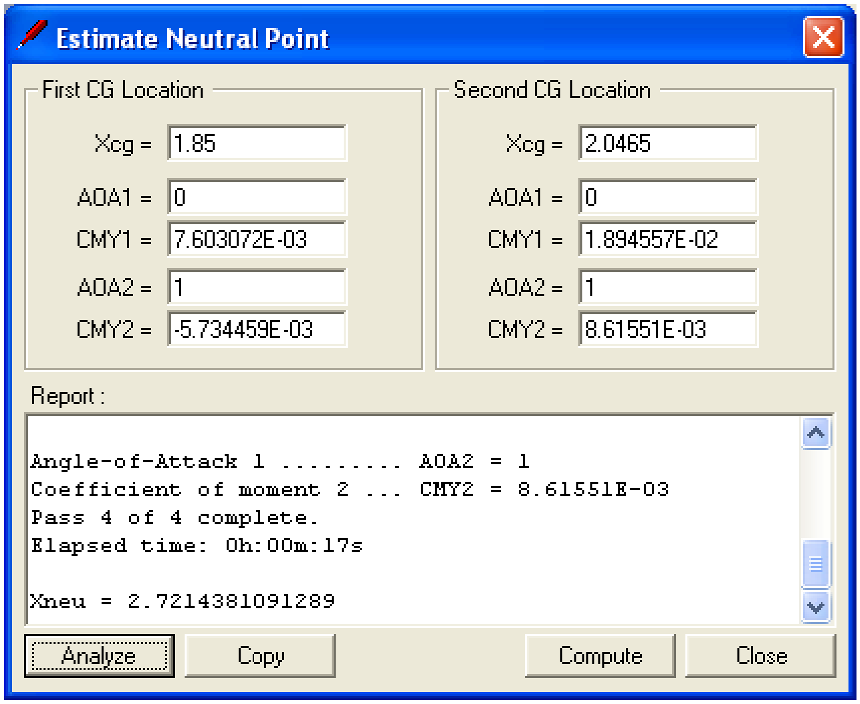


Figure 8-4. Computing the neutral point.

The neutral point was estimated by computing the slope of the CMY curve for two different values of Xcg; 1.85 ft and 2.0465 ft. The corresponding values of CMY for two angles-of-attack (AOA1 and AOA2) was evaluated (**SURFACES** provides a tool to make this simple, shown in Figure 8-4). The resulting Xneu is 2.721 ft. This corresponds to:

$$\frac{X_{NEU}}{C_{REF}} = 100\,\frac{(2.721 - 0.365)}{5.7} = 41.3\%MAC$$

V8.3 Results from SURFACES

Summary (note that values from Nelson and Schmidt appear to be from the same source):

	Ryan Navion				
	Source/ Symbol	*Flight Stability and Automatic Control*, R. C. Nelson	VLM using SURFACES	Panel Method using CMARC[21] (DWT)	*Introduction to Aircraft Flight Dynamics*, Louis V. Schmidt[22]
Air density	ρ	0.002378 slugs/ft³			
Outside Air Temperature	OAT	518.69 °R			
Speed of sound	ao	1116 ft/s			
Altitude	H	0 ft			
Far field speed	Vinf	178 ft/s			
Mach Number	M	0.159			
Baseline AOA	AOA	-	0.88°	0.76°	0.6°
Reference span	Bref	33.40 ft			

[21] Source: http://www.aerologic.com/stab/corr.html.
[22] Document is cited in footnote 1.

Reference wing chord	Cref	5.70 ft			
Reference wing area	Sref	184.0 ft²			
Reference aspec ratio	ARref	6.06			
Reference weight	W	2750 lbs			
Center of gravity (along X-axis)	Xcg	0.295 Cref	0.295 Cref (2.0465 ft)	0.25 Cref	-
Neutral point (along X-axis)	Xneu	0.552 Cref	0.413 Cref (2.721 ft)	0.38 Cref	-
Moment of inertia about X-axis	Ixx	1048 slugs·ft²			
Moment of inertia about Y-axis	Iyy	3000 slugs·ft²			
Moment of inertia about Z-axis	Izz	3530 slugs·ft²			
Product of inertia about X-axis	Ixy				
Product of inertia about Y-axis	Ixz				
Product of inertia about Z-axis	Iyz				
Lift coefficient for α = 0°	CLo	0.375[23]	0.329	-	-
Slope of lift coefficient	CLA	4.44	4.722	5.15	4.44
Lift coefficient	CL	0.41	0.406	0.415	0.415
Parasitic drag coefficient	CDo	0.0390	0.039	-	-
Total drag coefficient	CD	0.05	0.04952 (from quadratic drag polar)	-	0.051
Slope of drag coefficient at α = 0°	CDA	0.33	0.258 (quantic fit at α = 0.88°)	-	0.330
Span efficiency (Oswald's)	e	0.85	0.85	-	-
AOA DERIVATIVES	CXA	0.330	0.268	-	-
	CYA	-	0	-	-
	CZA	-4.850	-4.935	-	-
	CMXA	-	0	-	-
	CMA	-0.683	-0.584	-0.8721	-0.683
	CMZA	-	0	-	-
AOY DERIVATIVES	CXB	-	0	-	-
	CYB	-0.564	-0.5065	-	-
	CZB	-	0	-	-
	CLB	-0.0740	-0.07723	-	-
	CMB	-	0	-	-
	CNB	0.0710	0.07639	-	-
d(AOA)/dt DERIVATIVES	CXTA	-	Not predicted	-	-
	CYTA	-	Not predicted	-	-
	CZTA	1.7000	Not predicted	-	-
	CLTA	-	Not predicted	-	-
	CMTA	-4.36	Not predicted	-	-
	CNTA	-	Not predicted	-	-
U-DERIVATIVES	CXU	-0.1000 (?)	Not predicted	-	-
	CYU	-	Not predicted	-	-
	CZU	-	Not predicted	-	-
	CLU	-	Not predicted	-	-
	CMU	-	Not predicted	-	-
	CNU	-	Not predicted	-	-
P-DERIVATIVES	CXP	-	0	-	-
	CYP	-	0	-	-
	CZP	-	0	-	-
	CLP	-0.41	-0.44	-	-
	CMP	-	0	-	-
	CNP	-0.0575	-0.0652	-	-
Q-DERIVATIVES	CXQ	-	0	-	-
	CYQ	-	0	-	-
	CZQ	-	-8.99	-	-
	CLQ	-	0	-	-
	CMQ	-9.96	-12.98	-6.87	-9.96

[23] From analysis on page 54 of Reference document.

	CLQ	-	0	-	-
R-DERIVATIVES	CXR	-	0	-	-
	CYR	-	0.4059	-	-
	CZR	-	0	-	-
	CLR	0.1070	0.1374	-	-
	CMR	-	0	-	-
	CNR	-0.1250	-0.1557	-	-

Additional comparison based on a table from the source http://www.aerologic.com/stab/corr.html.

	CMARC (DWT)	Perkins & Hage	Etkin	Seckel[24]	Datcom	SURFACES	Wind Tunnel	Flight Test
Angle of attack, α	-0.732	-	-	-	-	-0.732°	-	-
Elevator deflection, de	8.28	-	-	-	-	-	-	-
C_L	0.271	-	-	-	-	0.269	-	-
C_D	-	-	-	-	-	-	-	-
$C_{L\alpha}$	5.23	4.36	4.25	4.54	5.5	4.722	4.52	6.04
$C_{D\alpha}$	-	-	-	-	-	-	-	-
$C_{M\alpha}$	-0.91	-.83	-.715	-0.545	-1.24	-0.584	-0.95	-
C_{Mq}	-6.99	-9.6	-9.75	-9.5	-12.98	-12.98	-	-
$C_{M\alpha de}$	-2.99	-3.0	-4.91	-4.91	-6.58	-	-	-
$d\varepsilon/d\alpha$	0.428	-	-	-	-	-	-	-
C_{Lde}	0.66	-	-	-	-	-	-	-
C_{Mde}	-1.68	-	-	-	-	-	-1.42	-1.42

24 Seckel E. “Stability and Control of Airplane and Helicopters”, Academic Press, 1964.

Validation 9: Comparison to NACA R-1208

V9.1 Introduction

This validation compares **SURFACES** analysis to the swept back wing featured in the NACA report R-1208. In the report a highly swept back, high aspect ratio wing compares three numerical methods to wind tunnel test results. In this validation sample, a similar approach will be taken and the section lift coefficients from **SURFACES** will be compared to the wind tunnel test results. The wing planform is shown in Figure 9-1.

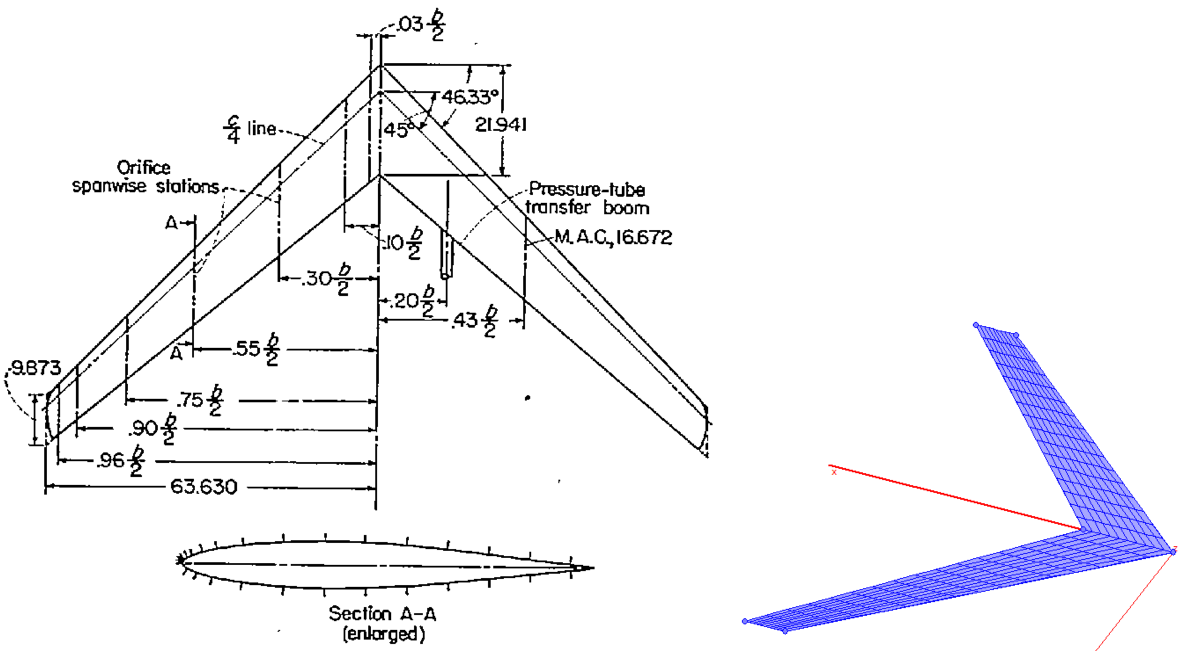


Figure 9-1: The swept-back wing wind tunnel tested per NACA R-1208. Inserted image shows the SURFACES VL model.

Three VL models were generated; one has 16 spanwise panel per wing side, the second one has 32 spanwise panels, and the third has 64 spanwise panels per side. The comparison takes place at 4.7° angle of attack, per the NACA report..

V9.2 Expected Result

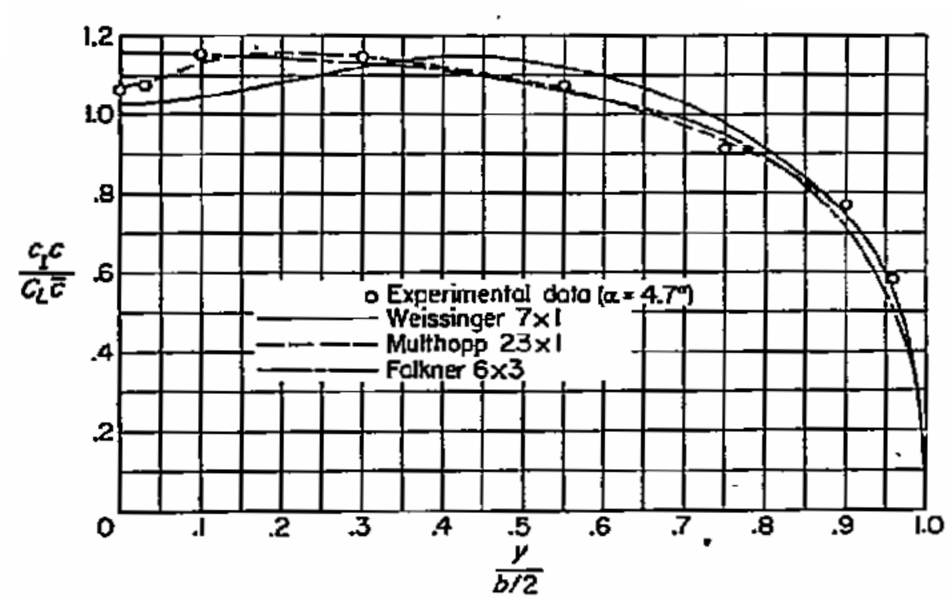


Figure 9-2: Original graph of spanwise loading from NACA R-1208.

V9.3 Results from SURFACES

The comparison of the numerical to the experimental data shows a close agreement, but also that the accuracy improves with number of panels.

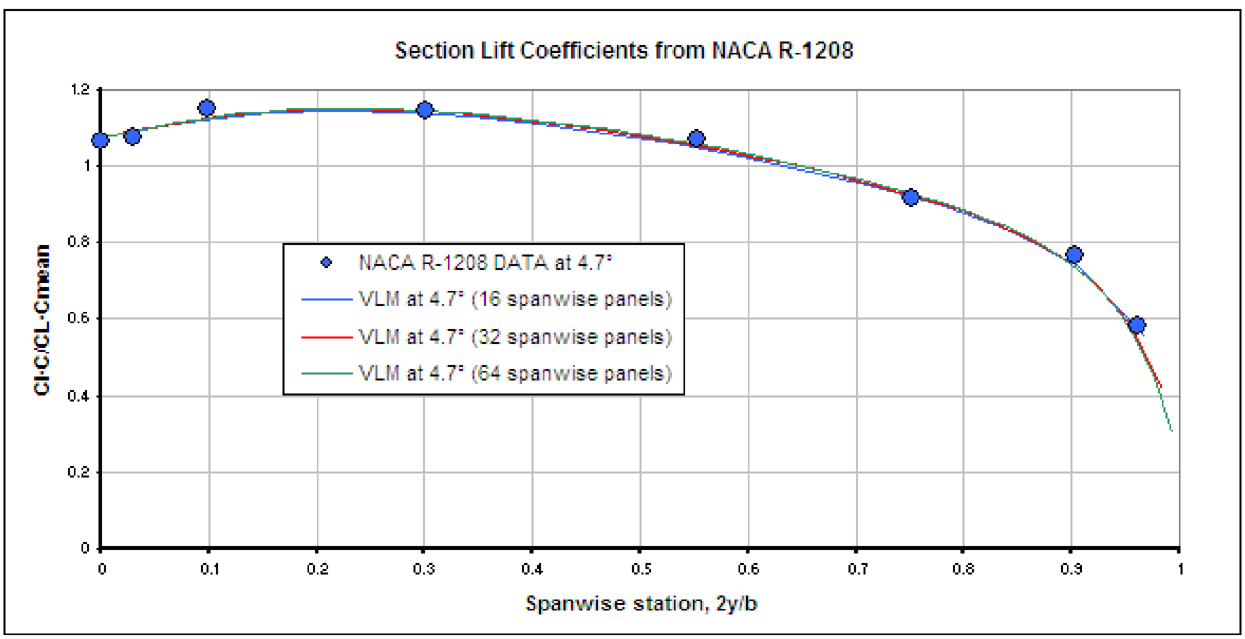


Figure 9-3: Comparing spanwise loading from SURFACES to experimental data from NACA R-1208.

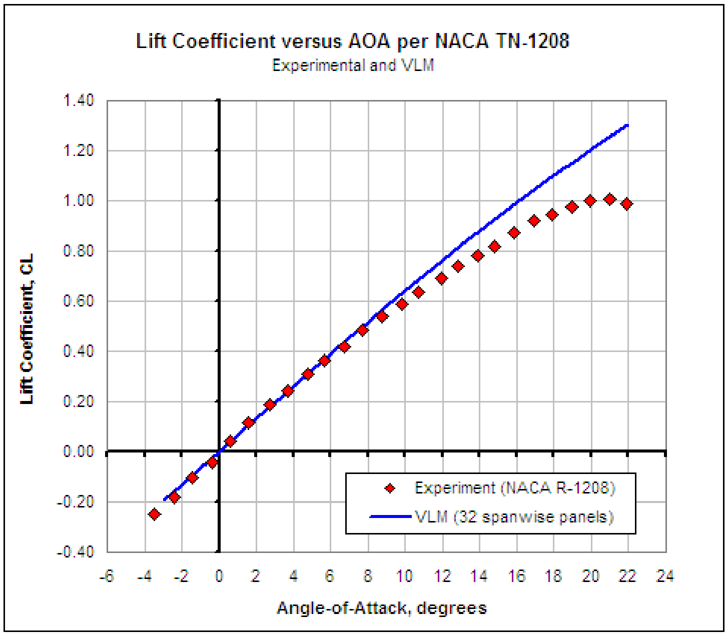


Figure 9-4: Comparing lift curve from SURFACES to experimental data from NACA R-1208.

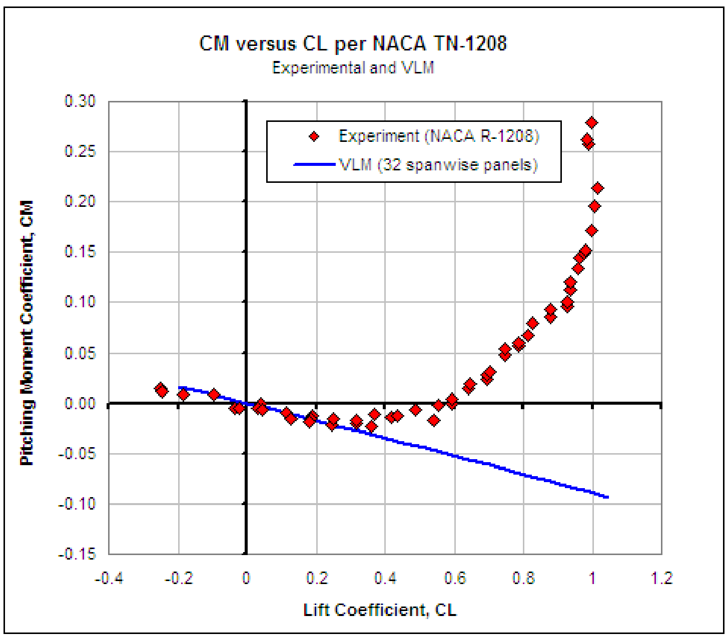


Figure 9-5: Comparing moment curve from SURFACES to experimental data from NACA R-1208. The experimental data shows the well known early tip stall phenomena of swept back wings, caused by spanwise flow near the tips. This is reproduced here to remind the user that all inviscid codes (vortex-lattice, doublet-lattice, panel-codes, etc) do not model this viscous phenomena accurately because the mathematical solution forces the flow to stay attached.

Validation 10: Comparison to NACA TN-1422

V10.1 Introduction

This validation compares **SURFACES** analysis to two of the three tapered and twisted wings featured in the NACA report TN-1422. This report compares several aerodynamic properties of three wings obtained in wind tunnel tests. In this validation sample the section lift coefficients, lift curves, and moment curves for two of these wings (from hereon referred to as WING 2 and WING 3) from **SURFACES** will be compared to the wind tunnel test results. The general planform shape is shown in Figure 10-1, and is reproduced from the original document.

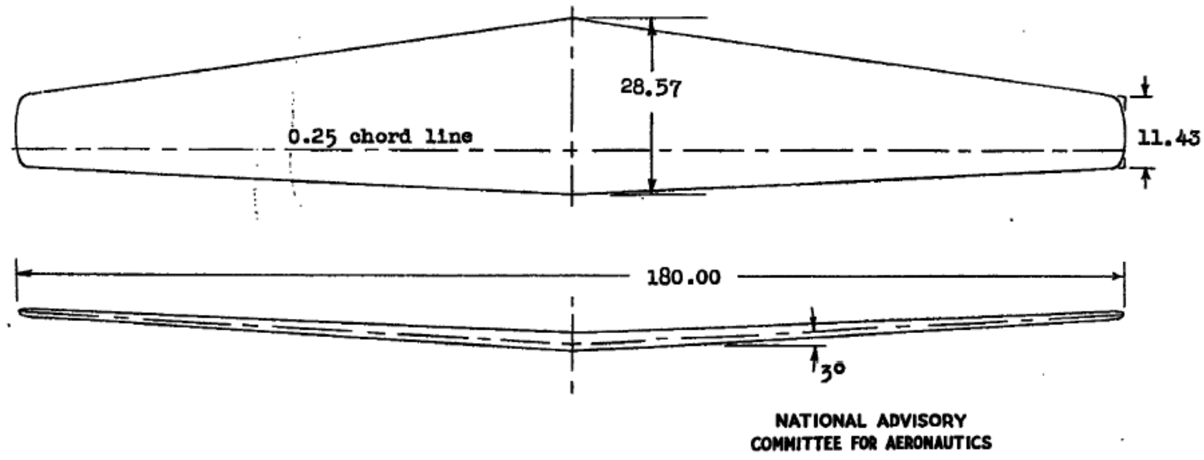


Figure 1.- General dimensions of 10-percent-thick wings tested in Langley 19-foot pressure tunnel. Aspect ratio, 9; ratio of root chord to tip chord, 2.5. (All dimensions are in inches.)

Figure 10-1: The general shape of the wind tunnel model tested per NACA TN-1422.

V10.2 Results from SURFACES

The comparison of the numerical to the experimental data shows a close agreement.

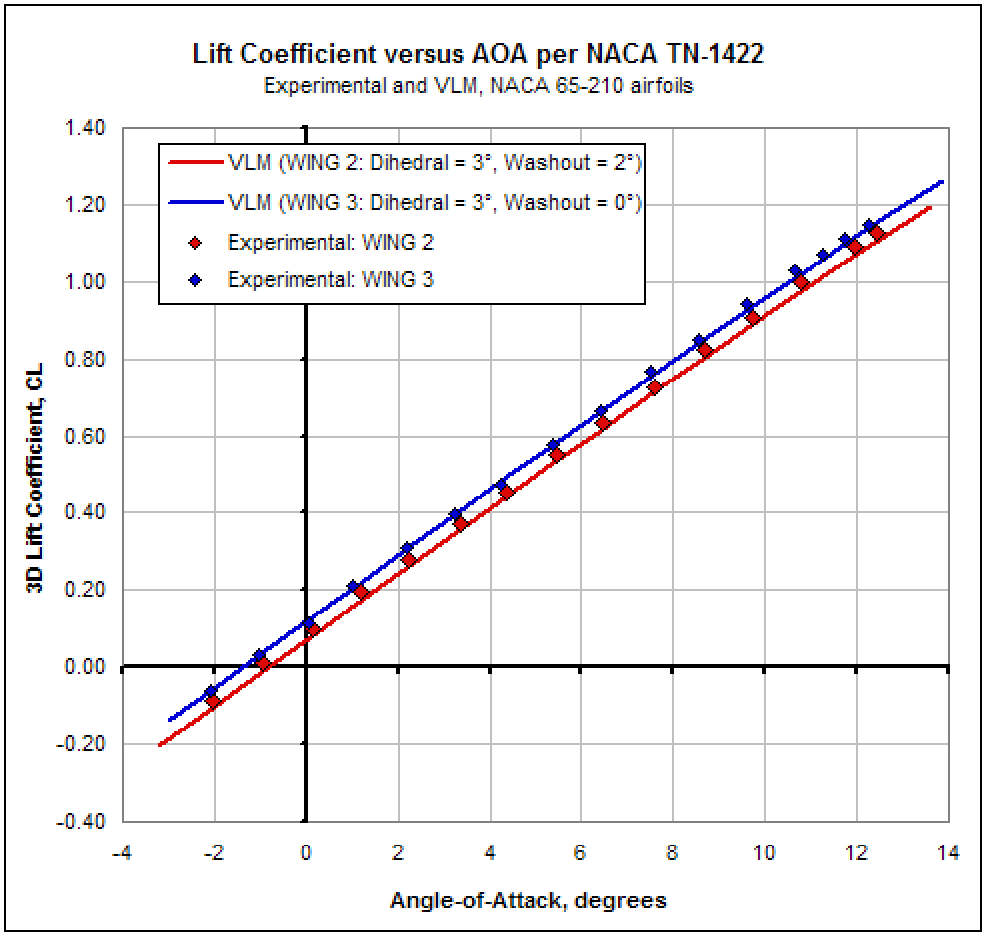


Figure 10-2: Match for the lift curve for the twisted and untwisted wings.

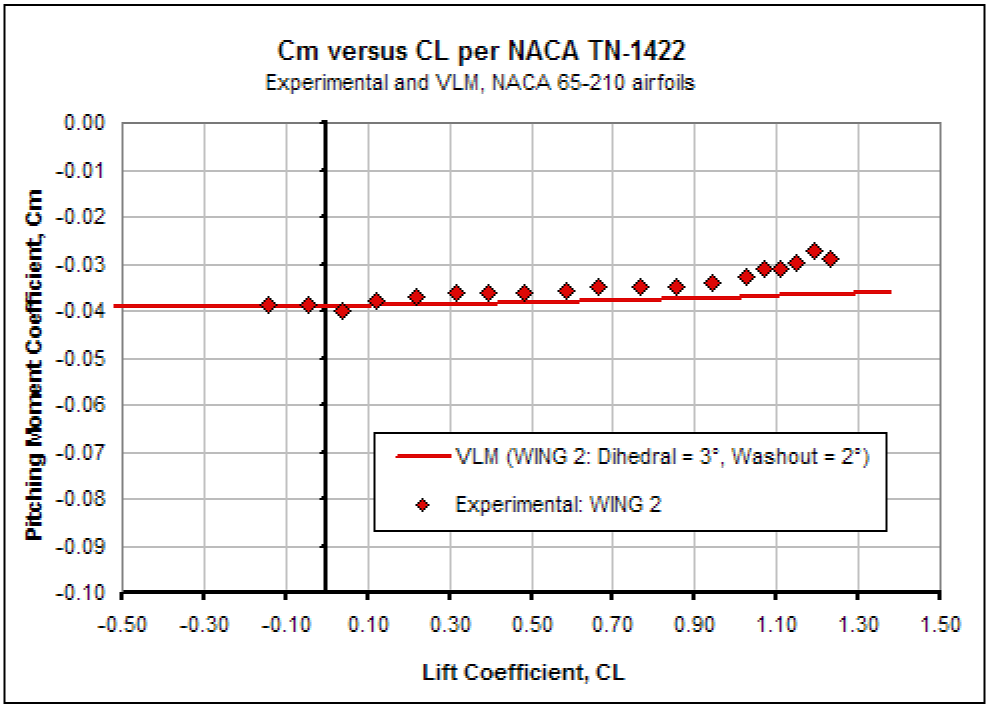


Figure 10-3: Match for the pitching moment for the twisted wing. Note the deviation at higher values of the lift coefficient, which is caused by viscous effects.

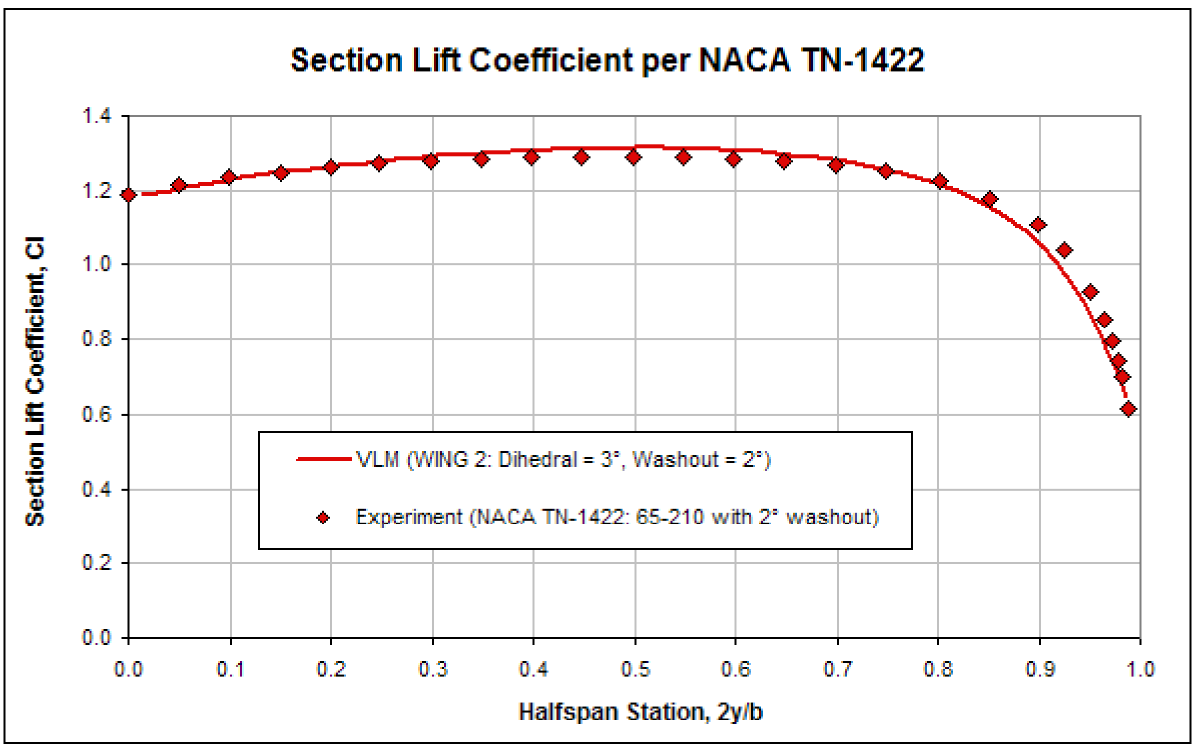


Figure 10-4: Lift distribution at stall for the twisted wing.

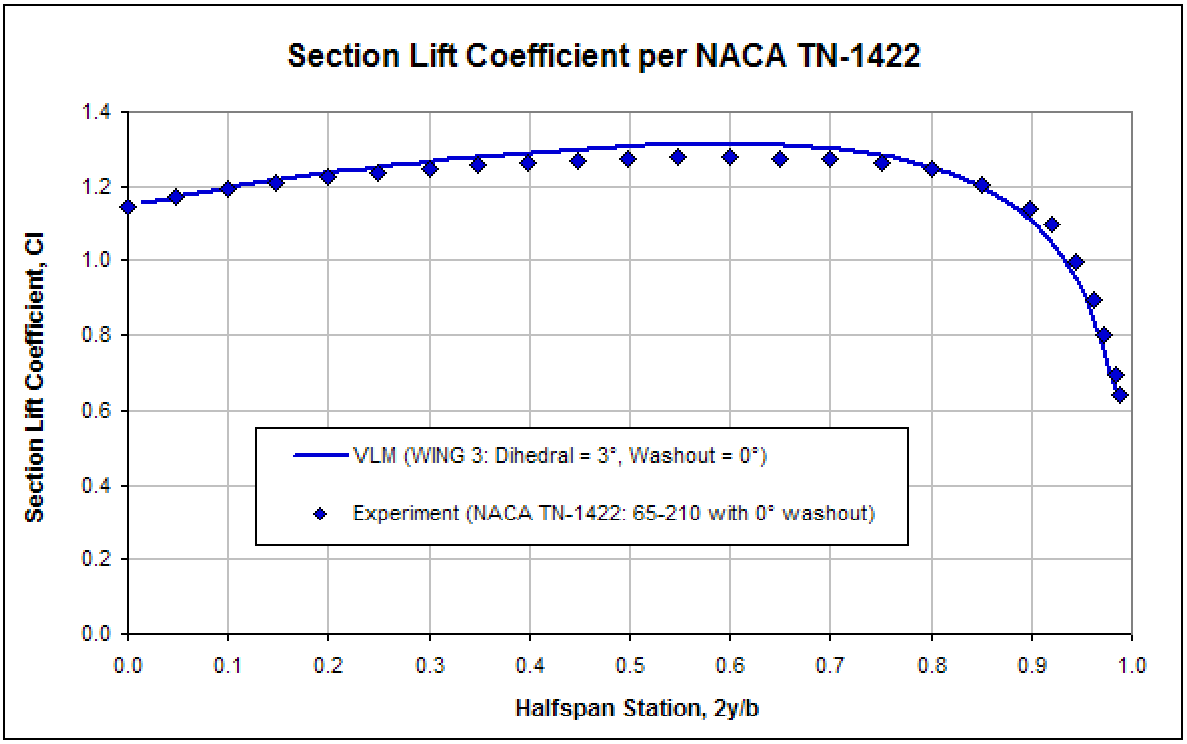


Figure 10-5: Lift distribution at stall for the untwisted wing.