User's Guide for *AeroLS* Aerodynamic Lifting Surface Analysis Program

Version 1.0.0

by

Joseph A. Huwaldt

December 21, 2002

License

For more information on AeroLS or to obtain the latest version (software, documentation and source code) visit the following world wide web address:
http://homepage.mac.com/jhuwaldt/java/Applications/AeroLS/AeroLS.html

Table of Contents

1. Introduction

AeroLS is an aerodynamic lifting surface analysis program. This means that it computes the effects of air flowing around thin (planform and camber only, no thickness) wings which are operating at a small angle of attack and sideslip in flow that is subsonic, steady, inviscid, and irrotational. AeroLS can be used to analyze any number of different lifting surfaces at a time, but can not analyze the effects of bodies such as fuselages in combination with wings. There are no limits (other than your computer's memory and solution run time) on the number of panels used to define your problem's geometry.

AeroLS is part of a large family of analysis codes that are referred to as "linear aero codes" because it assumes that the effects of planform, camber, thickness and friction are linearly separable. This means that these effects can be analyzed separately and then added together to give the total forces acting on the vehicle. AeroLS, at this time, only computes the effects of planform and camber. In reality, all these effects are not linearly separable, but for some cases (subsonic, high Reynolds number flow, on well designed wings at low lift coefficients), this assumption is not such a bad one and can give good 1st order values for educational or quick evaluation purposes.

AeroLS, at this time, makes no attempt to model or make corrections for the effects of leading edge suction, tip vortex roll up, wake distortion, or viscous fluid properties.

2. User Interface

AeroLS has a graphical user interface that allows the user to view the geometry to be analyzed, enter reference quantities, conduct the analysis, and view the results of the analysis. There is also a feature where you can create new trapezoidal, uncambered, paneled wing segments interactively. Unfortunately, there is no geometry editing capability in AeroLS.

When the program first starts up, the user is presented with a blank application window much like that shown in Figure 2.1 (but without any geometry showing). Appearance will vary depending on your operating system and graphical environment. In addition, there are three menus – File, Edit, and Analysis (that in MacOS X appear at the top of the screen and in other OS's appear just below the window's title bar).

2.1 View Regions

The application window has four regions: geometry view, reference quantities, components & segments, and tool bar. Each region which will be discussed below.

Geometry Views:

First, there are three areas where the geometry is shown in top view, side view, and front view respectively. The number showing in each view is the view scale (1/100th scale in Figure 2.1). The example geometry shown in Figure 2.1 is that used to discuss input file formats in Section 3. The user can zoom in and out with the +Zoom and -Zoom buttons in the button bar across the top of the window, but otherwise the geometry views can not be modified (sorry). This is a user interface element that seriously needs work. I hate it, but just haven't had time to do anything about it.

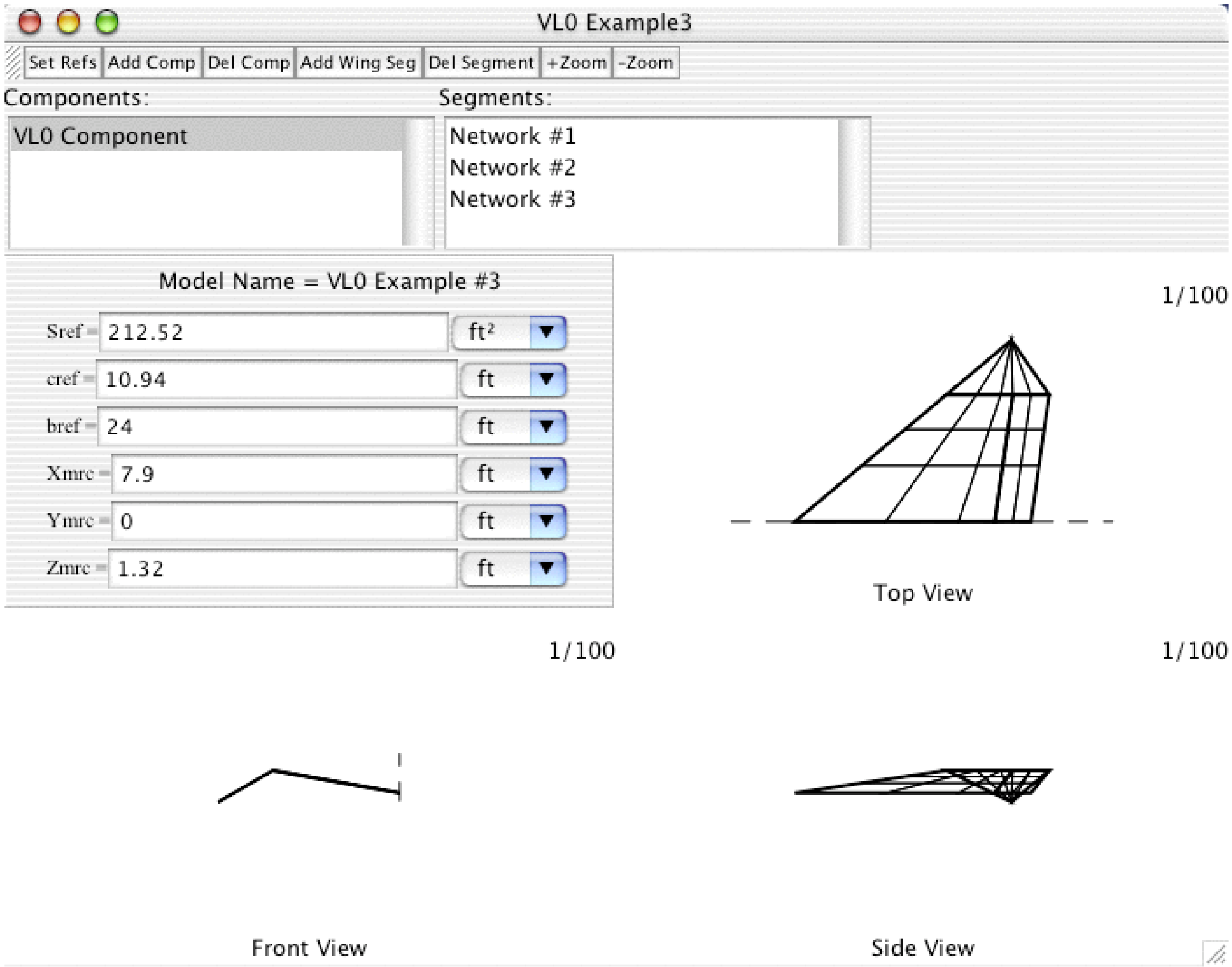

Figure 2.1: Main Document Window in AeroLS (MacOS X)

Reference Quantities:
The reference quantities region shows the name of the currently loaded model and the values of the currently stored reference quantities used to non-dimensionalize the analysis results. The reference quantities can not be changed in this region, but the units used to display them can. The reference quantities and model name are all entered or changed by clicking on the "Set Refs" button in the button bar (just below the title bar in Figure 2.1). This dialog will allow the user to enter a model name and reference quantities in a variety of units.

Components and Segments:
The components and segments region is used to show how the segments (networks or arrays of panels) are organized in the model. In the example shown in Figure 2.1, there is one component that contains three segments. This one component is referring to a single wing shown in the geometry view that is made up of three trapezoidal regions which are named "Network #1" through "Network #3".

A model must contain at least one component and may contain any number of different components. Components may be used to logically group together different parts of the model. For instance, you might have a wing component, a horizontal tail component, a vertical tail component and a canard component all in the same model. Each of these components can be made up of any number of segments (networks). Or, you can choose to put all the segments that make up your model into a single component. In the future, AeroLS may allow you to extract results on a component by component basis (like "How much lift is generated by the horizontal tail component?"), but at this time, there is no advantage to separating segments out into different components. AeroLS just lumps them all together to run the analysis.

A model must contain at least one segment and may contain any number of different segments. A segment is a network or array of quadrilateral panels that are to be analyzed. Segments may be organized and arranged in basically any way that is convenient to you. The example in Figure 2.1 is made up of three segments. The boundaries between the segments are identified by the slightly thicker lines used to draw them in the View Regions. See the Geometry section of this document for more information on segments and how they are arranged.

There are buttons in the tool bar that allow you to add and delete components and segments. Clicking on the "Add Component" button causes the program to ask you for a name for that component. It will then show up in the component list. Clicking on the "Add Segment" button will bring up a dialog that will allow you to construct simple trapezoidal wing segments. If a component does not already exist, clicking on "Add Segment" will first ask you to create a component in which to place the segment you are creating. If more than one component exists, select the component you want to add the segment to before selecting the "Add Segment" button.

2.2 Menus

The AeroLS program has three application menus: File, Edit and Analysis. Each will be discussed below.

The File menu, shown in Figure 2.2 contains the following items: Open, Close, Save, Save As..., and possibly Quit depending on your operating system.

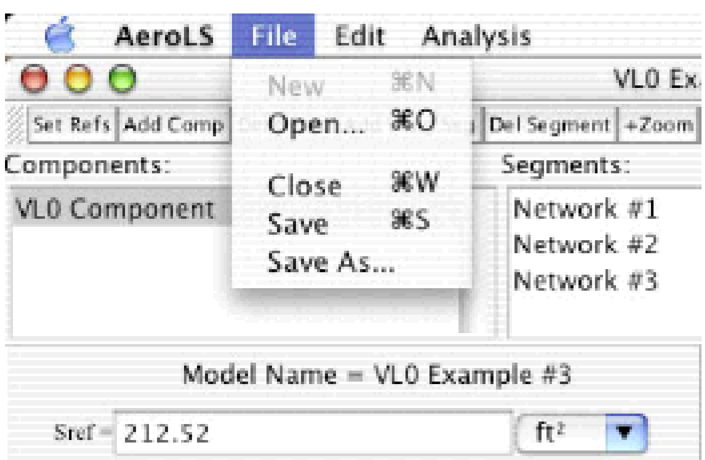

Figure 2.2: File Menu (MacOS X)

Open and Save/Save As... are used to read in and write out geometry and input parameters to one of the supported file formats (see the section of this document on Geometry and Input File Formats for more information).

The Close item will close this application window (and effectively exit the program since AeroLS only allows one document window to be open at a time).

There is not currently any printing capability in AeroLS. Once I've decided on a better way to view and edit geometry, then a printing mechanism will be added.

The Edit menu has no enabled items under it and so will not be discussed.

The Analysis menu, shown in Figure 2.3, has two items: Test Conditions, and Analyze.

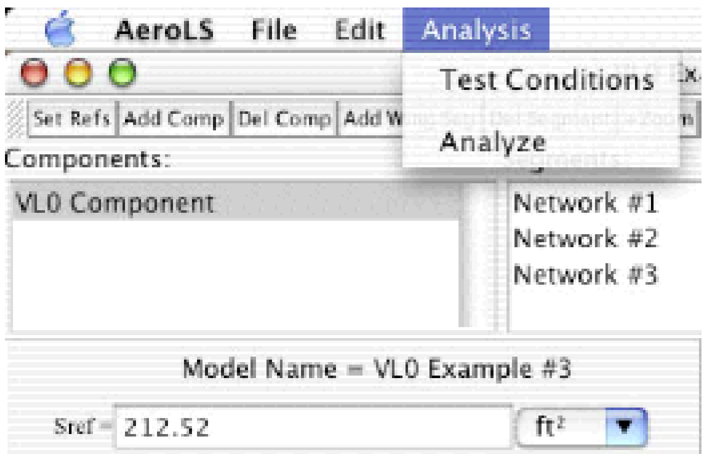

Figure 2.3: Analysis Menu (MacOS X)

2.3 Test Conditions Dialog

Choosing the item "Test Conditions" brings up a dialog, shown in Figure 2.4, that allows you to enter the conditions at which the analysis will be done. Angle of attack (AOA) sweeps may be entered for a range of Mach numbers at a specific sideslip angle (beta). No other sweep type is currently allowed. Future versions of AeroLS may add other sweep types and a different way of entering them.

Values are entered in the AOA and Mach columns by double-clicking on a cell and entering a value. You may insert additional cells (which will contain a value of 0.0 until you change it) by selecting a cell and clicking on the "Insert" button and you may select a cell and delete it by clicking on the "Delete" button. Angle of Attack (AOA) is entered in degrees. Angle of sideslip (sideslip or beta) may be entered in either degrees or radians. Mach number is non-dimensional. You may enter any value for AOA or sideslip, but remember that AeroLS is only valid in the "linear lift range". Mach numbers may have any value from 0.0 to, but not including, 1.0. Remember that AeroLS can not predict transonic effects. Any number of AOA and Mach breakpoints may be entered.

A Mach number of 0.0 implies an incompressible solution. Technically, all AeroLS solutions are incompressible, but a Prandtl-Glauert transformation is used to partially account for compressibility effects in the subsonic range.

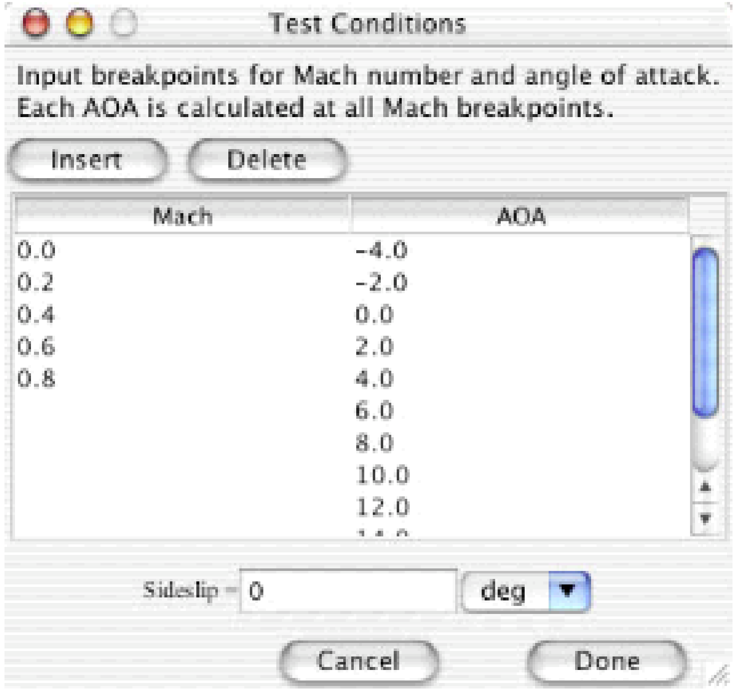

Figure 2.4: Test Conditions Dialog (MacOS X)

Finally, if you choose "Analyze" from the "Analysis" menu, a solution will be run at all the Mach and AOA breakpoints specified in the Test Conditions dialog. The run time for the solution will depend on the number of panels in the geometry and the number of Mach breakpoints (and to a much lesser extent on the number of AOA breakpoints). For typical problems the run time is measured in minutes or even seconds, not generally in hours. A progress bar will be shown for analysis jobs that take more than a few seconds to run.

2.4 Viewing Analysis Results

When an analysis run has completed, a results window will be displayed that looks something like that shown in Figure 2.5. This window simply displays a list of all the parameters that have been calculated by the program. Definitions of the data output in this list can be found in the section on Analysis Data and Output File Formats.

You may select one or more of the tables in the list shown in the results window and then click on one of the four buttons on the right hand side of the window labeled: "View", "Lookup", "Plot" or "Remove".

The "View" button allows you to view the table in tabular (spreadsheet like) form. When viewed in this way, the numbers in the tables can be copied to the clipboard and then pasted directly into a spreadsheet program such as Microsoft Excel or OpenOffice.

The "Lookup" button brings up a dialog that allows you to do interactive table lookups into the table of results. This allows you to specify an angle of attack and Mach number (for instance) and find the value that corresponds to the input values with linear interpolation if necessary.

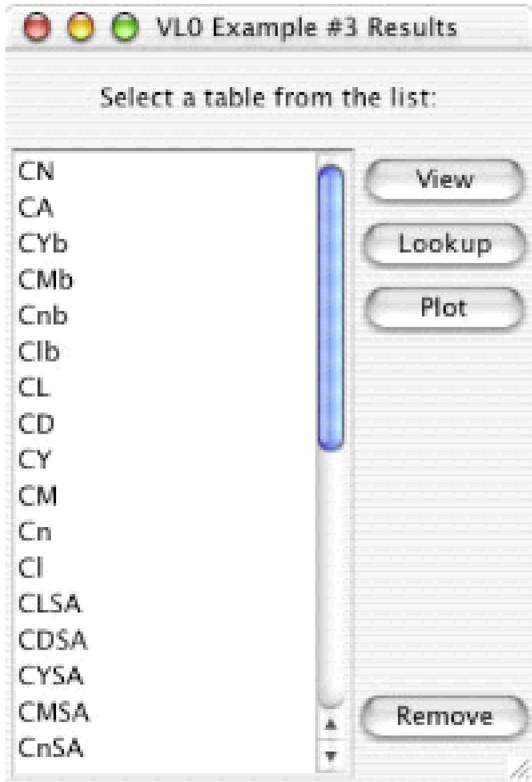

Figure 2.4: Results Window (MacOS X)

The "Plot" button allows you to create a simple line-graph type plot of the selected tables. Unfortunately, printing of these plots has not yet been implemented. An example plot can be found in Figure 2.5.

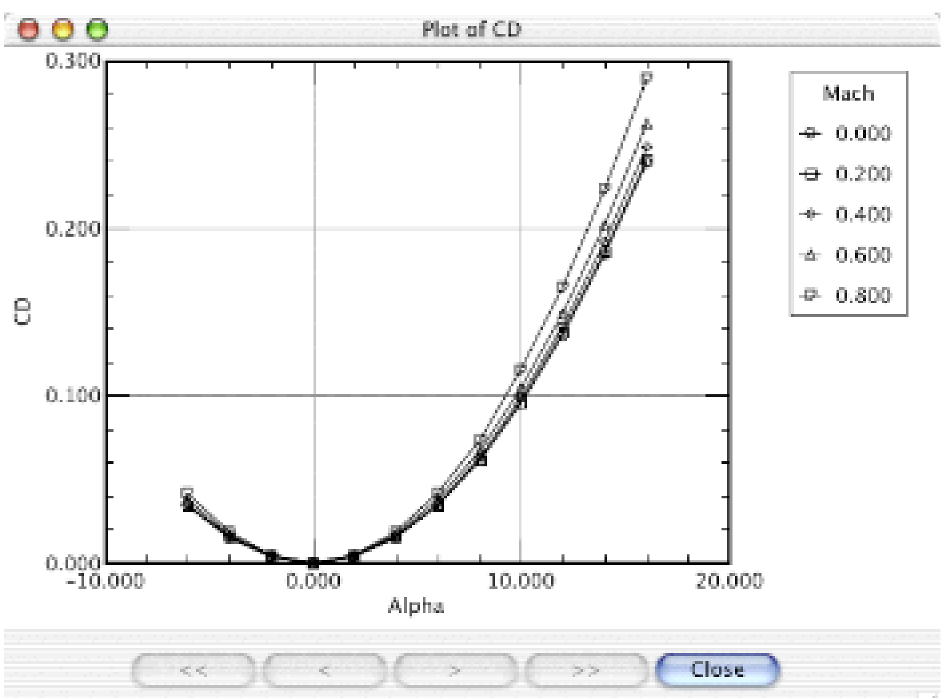

Figure 2.5: Example Plot Output from AeroLS (MacOS X)

The “Remove” button will delete the selected tables from the list of tables presented.

The results window also has a “File” menu associated with it (not shown in Figure 2.5) that allows you to save the results out to one of the supported output file formats (see the section on Analysis Data and Output File Formats for more information).

3. Geometry

The coordinate system used by AeroLS is shown in Figure 3.1. This is a body fixed coordinate system where the origin is typically located forward of the configuration being analyzed. The X-coordinate axis runs along the longitudinal length of the vehicle and is positive aft. The Z-coordinate axis runs vertically and is positive up. The Y-coordinate is then found by the right-hand rule and is positive starboard or to the right of the vehicle (as viewed from behind looking forward).

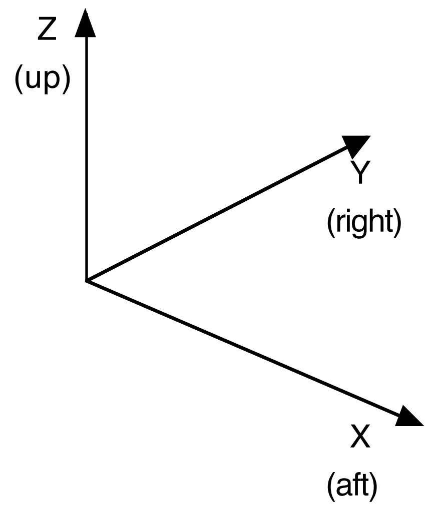

Figure 3.1: Coordinate System used by AeroLS

Geometry in AeroLS is organized into a series of components each of which contains a series of networks or segments that define the wings to be analyzed. Components can be used to group related segments. For instance, you might have a main wing component that contains several segments or networks that define the wing. Figure 3.2 shows a wing component that contains two segments (networks). You might then also have a horizontal tail component that contains the segments that make up the horizontal tail and so on. Or, you might put all your segments in a single component. For the current version of AeroLS there is no real advantage to separating your geometry into different components. However, future versions may make use of this feature, for example, to provide information on how much a particular component is contributing to the total forces and moments.

The geometry analyzed by AeroLS is defined by a series of quadrilateral panels (panels on the wing tip can have the outboard edge collapsed and so appear to be triangles). The corner points for the panels are defined by a series of networks, or arrays, of points (see Figure 3.2). The networks can be in any order and can be organized in any way that is convenient for defining the wing planform. However, slightly improved accuracy is gained if a single network stretches from the leading edge to the trailing edge, especially for highly cambered surfaces such as those with flaps deflected (see the AeroLS Theory Manual for more information). Each network is made up of a series of columns or strings of points. Each column is made up of a series of 3D points. The points in a single column are ordered from leading edge to trailing edge. The columns are ordered from inboard to outboard.

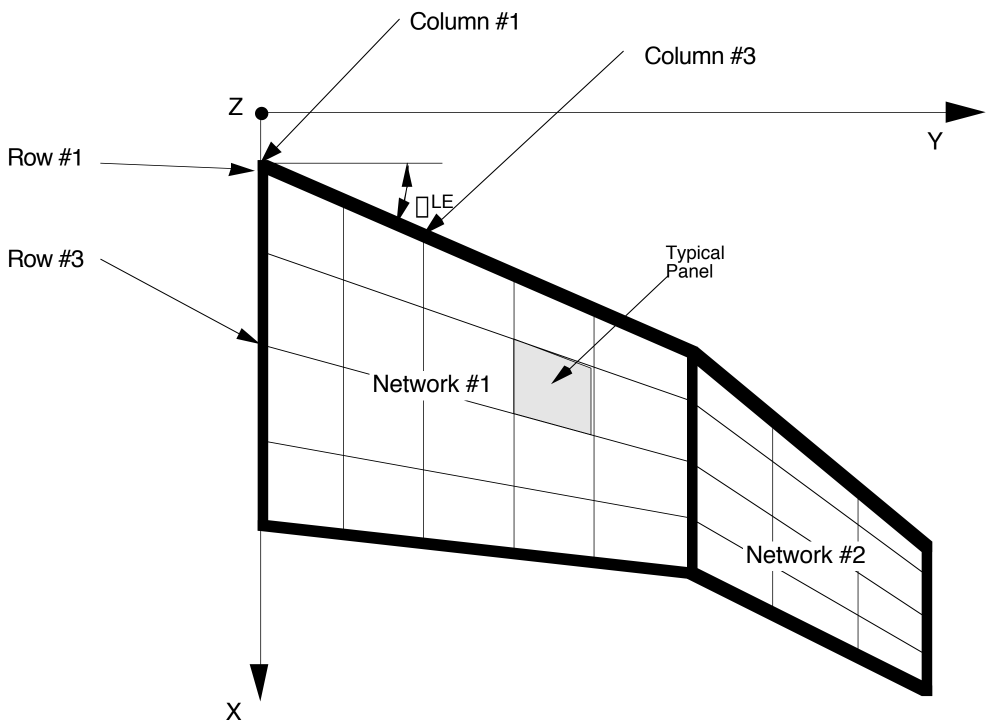

Figure 3.2: Definition of Point Ordering in AeroLS

Using this ordering system, the 1st point, in the 1st column of the 1st network is the apex of the wing as shown in Figure 3.2. The next point is aft of the apex along the root chord line. The next point is aft of that, etc. until the trailing edge of the 1st network is reached. Then the next point is the 1st point of the 2nd column of the 1st network and so on. The last point in the 1st network is the most trailing edge point of the most outboard column of the network.

It is not necessary for the inboard and outboard edges of the panels to be strictly streamwise, as is shown in Figure 3.2. For best accuracy, the inboard and outboard panel edges should be oriented to follow the local flow direction.

The main limitation on geometry is that the control points of one segment must never coincide with the vortex lines trailing off of another segment. This most often happens when a horizontal tail follows directly behind a wing. The best way to avoid this problem is to mount the aft wing a small distance above or below the trailing edge of the forward wing. If two wing surfaces must be in the same XY plane, then care must be taken to make sure that the panels of each surface have the same spanwise spacing.

Another limitation in this version of AeroLS is that will only work with geometry that is symmetric about the XZ plane and that is provided as a half-model. This is due to a limitation in the current user interface and not in the solver which is general.

AeroLS provides a very minimal ability to create paneled wing segments (networks) internally using user supplied information about the planform. A main drawback to this approach is that you can only create twisted but uncambered wings. To create a cambered wing, you will need to create the geometry some other way and import it through one of the supported file formats discussed below (probably with the GGP format since the VL0 format doesn't support camber and the XML format is complex).

Creating a wing segment is done by clicking on the "Add Wing Seg" button in the application window. This will ask you to name a component if you did not already have one defined and then bring up a dialog where you can enter various combinations of parameters that are commonly used to define wing geometry. This dialog works by allowing you to enter any combination of parameters that you want. Once enough information has been entered to calculate some other parameter it will be filled in automatically. Once all the parameters have been filled in either manually or automatically, the wing segment can be created.

The parameters in this dialog are defined as follows:

Xrle, Yrle, Zrle	X,Y,Z-coordinates of the root chord leading edge
Xtle, Yrle, Zrle	X,Y,Z-coordinates of the tip chord leading edge
span	True (not projected) span or distance from root to tip
AR	Aspect ratio defined as span*span/area
area	The trapezoidal area of the segment
Cr	The root chord length
Ct	The tip chord length
taper	The taper ratio (tip chord length / root chord length)
SweepLE	Leading edge sweep angle as defined in Figure 2.2.
Sweep 1/4c	The sweep angle of the 1/4 chord line
SweepTE	The trailing edge sweep angle
MAC	The mean aerodynamic (geometric) chord length
dihedral	The angle the segment makes with the Y axis (0 = horiz, 90° = vert)
twist	The difference in incidence angles between the tip and root chords
ir	The incidence of the root chord line
it	The incidence of the tip chord line

This dialog also allows you to specify the number of panels in the chordwise and spanwise directions and how those panels are distributed. Chordwise distribution options are shown in Figure 3.3. These same distributions can be specified in the spanwise direction as well.

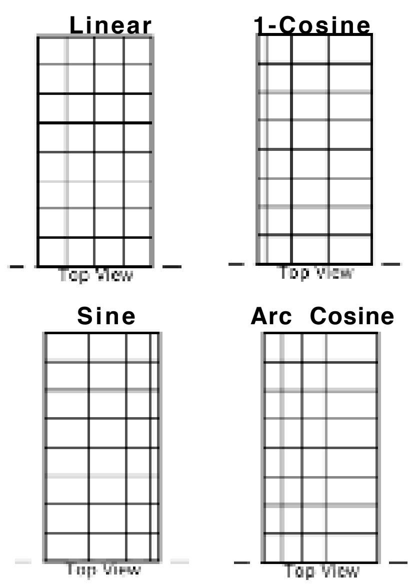

Figure 3.3: Chordwise Panel Distribution Options

4. Input File Formats

At this time, AeroLS can read or input three different file formats: XML, VL0 Geometry, and GGP. AeroLS saves geometry and other input data into an extendible XML format file. This can be thought of as the native format for AeroLS. A VL0 file is a simple ASCII format that allows you to easily specify uncambered trapezoid geometry, and GGP is a more general ASCII format for storing paneled geometry.

4.1 XML Format

AeroLS saves out, and can read back in, geometry and some input parameters to/from an XML formatted file. This is the most complex, but most flexible, format that AeroLS supports. XML files are text files that contain data stored with tags that are organized in a hierarchal manner. If a particular tag is missing from the XML file or if a particular tag is incorrectly formatted, then AeroLS will simply ignore that element and use a default value. XML files are human readable and are usually pretty easy to understand. It can be more difficult to get one properly formatted if you are creating it from scratch – I strongly recommend using an existing file as a template for creating a new one or use an editor that will take care of the syntax for you.

AeroLS's XML format for geometry requires you to specify the location of the corner points of each panel arranged into sets of networks. This allows the user to specify camber and effects like continuous dihedral change.

An example XML file readable by AeroLS can be found in Table 4.1. Tags are identified by "<tag>...</tag>" pairs and the <document>...</document> tag pair is required. All other tags are optional. Each tag recognized by AeroLS will now be described.

<modelName>
This tag allows you to set the name of the configuration/vehicle/model being analyzed.

<parameters>
This tag indicates that a set of parameter data is to follow. Parameters are single value variables used primarily to set reference quantities.

Table 4.1: Example AeroLS XML Input File


```
<?xml version="1.0" encoding="UTF-8"?>
<document>
   <modelName>VL0 Example #3</modelName>
   <parameters>
      <param name="cref">
         <value>10.9393</value>
         <unit>Feet</unit>
      </param>
        <param name="Sref"><value>212.52</value><unit>SqFeet</unit></param>
        <param name="bref"><value>24.0</value><unit>Feet</unit></param>
   </parameters>
   <geometry>
      <vehicle name="VL0 Example #3">
         <component name="VL0 Component">
            <network name="Network #1" strings="4" points="4">
             <array name="X">
0.0,1.828798,3.2918406,4.02336,1.3439113,2.6350713,3.6680021,4.184466,2.215637
,3.158059,3.9119985,4.288967,3.051043,3.659257,4.145829,4.389115,</array>
             <array name="Y">
0.0,0.0,0.0,0.0,1.1277595,1.1277595,1.1277595,1.1277595,1.8592788,1.8592788,1.
8592788,1.8592788,2.56032,2.56032,2.56032,2.56032,</array>
             <array name="Z">
0.0,0.0,0.0,0.0,0.19885442,0.19885442,0.19885442,0.19885442,0.327841,0.327841,
0.327841,0.327841,0.45145348,0.45145348,0.45145348,0.45145348,</array>
          </network>
            <network name="Network #2" strings="4" points="3">
             <array name="X">
4.02336,4.3891196,4.75488,4.1844683,4.550228,4.9159884,4.288971,4.654731,5.020
491,4.3891196,4.7548795,5.12064,</array>
             <array name="Y">
0.0,0.0,0.0,1.1277595,1.1277595,1.1277595,1.8592788,1.8592788,1.8592788,2.5603
2,2.56032,2.56032,</array>
             <array name="Z">
0.0,0.0,0.0,0.19885442,0.19885442,0.19885442,0.327841,0.327841,0.327841,0.4514
5348,0.45145348,0.45145348,</array>
          </network>
            <network name="Network #3" strings="2" points="6">
             <array name="X">
3.051048,3.6592617,4.1458354,4.38912,4.754881,5.12064,4.358638,4.358638,4.3586
38,4.358638,4.358638,4.358638,</array>
             <array name="Y">
2.56032,2.56032,2.56032,2.56032,2.56032,2.56032,3.6576,3.6576,3.6576,3.6576,3.
6576,3.6576,</array>
             <array name="Z">
0.45143926,0.45143926,0.45143926,0.45143926,0.45143926,0.45143926,-
0.18207568,-0.18207568,-0.18207568,-0.18207568,-0.18207568,-
0.18207568,</array>
          </network>
        </component>
      </vehicle>
   </geometry>
   <testConditions>
      <table name="Mach" length="5">0.0,0.2,0.4,0.6,0.8,</table>
     <table name="AOA" length="11">
-4.0,-2.0,0.0,2.0,4.0,6.0,8.0,10.0,12.0,14.0,16.0,</table>
   </testConditions>
</document>
```

<param name="pname">
Identifies a parameter with a specified name. AeroLS currently recognizes the following parameter names. Any additional parameter names are ignored.

cref	-	Reference chord length, default = 1.0
bref	-	Reference span length, default = 1.0
Sref	-	Reference area, default = 1.0
Xmrc	-	X-coordinate of moment reference location, default = 0.0
Ymrc	-	Y-coordinate of moment reference location, default = 0.0
Zmrc	-	Z-coordinate of moment reference location, default = 0.0
C.P.	-	Panel control point location, default = 0.75

Parameters have the following optional tags:

<value>
The single precision numerical value assigned to the parameter.

<unit>
The Java class name (Java package jahuwaldt.tools.units) of the unit assigned to the parameter. The following units are currently recognized by AeroLS:

Feet	-	U.S. customary feet (ft)
Inches	-	U.S. customary inches (in)
Meters	-	S.I. meters (m)
Centimeters	-	S.I. centimeters (cm)
SqFeet	-	Square feet (ft^2)
SqInches	-	Square inches (in^2)
SqMeters	-	Square meters (m^2)
SqCentimeters	-	Square centimeters (cm^2)
Degrees	-	Angular unit of degrees ($1/360^{th}$ of a circle)
Radians	-	Angular unit of radians (2π radians per circle)

<testConditions>
Identifies test conditions at which analysis is to be done. Test conditions are specified with named tables of comma separated values. The following names are currently recognized by AeroLS:

Mach	-	Mach number
AOA	-	Angle of attack, each AOA is computed at each Mach & Beta
Beta	-	Angle of sideslip, positive wind from the right

The only limitation in these breakpoint tables is that Beta may currently only have a single value. If more than one value is supplied, only the 1^{st} one is used.

Test conditions have the following sub-tag:

<table name="tname" length="#">
The name of the table and the number of breakpoints go in the tag. The breakpoint values are separated by commas.

<geometry>
Contains a list of all the vehicles to be analyzed. Currently, only a single vehicle is supported by AeroLS. If multiple vehicles are present, only the 1st one will be used.

<vehicle name=”vname”>
Identifies the name of the vehicle in the tag and contains a list of components contained in that vehicle. Currently, only a single vehicle is supported and usually has the name of the model (<modelName>).

<component name=”cname”>
Identifies the name of each component in the tag and contains a list of networks contained in this component. These correspond to the components in the AeroLS user interface.

<network name=”nname”>
Identifies the name of the network in the tag and contains three named arrays: “X”, “Y”, and “Z”. Networks correspond to the segments in the AeroLS user interface.

<array name=”aname”>
“aname” must be “X”, “Y”, or “Z” and all three must be present. Any additional names are ignored. Each array contains a comma separated list of appropriate coordinate values. Coordinate values must always be specified in units of meters! Each array must have an equal number of points.

4.2 VL0 Format

AeroLS can read in “VL0 GMETRY.DAT” formatted files. These are referred to from now on as VL0 format files and must have an extension of “.vl0” to be recognized as such by AeroLS. VL0 is a set of vortex lattice codes written by engineers at The Boeing Company in the early 1990’s and is rarely (ever?) used anymore. This geometry input format has the advantage of being very simple and compact. It has the drawback of it not being possible to specify camber.

The VL0 format allows you to specify a series of trapezoids that define the planform geometry. An example of the format is given in Table 4.2 and a key to all the parameters is given in Table 4.3. The geometry specified in Table 4.2 is identical to that shown in the XML file demonstrated in Table 4.1. The relative ease of use of the VL0 format is obvious.

AeroLS prompts the user for the length units when a VL0 file is read in and the units are limited to the following unit types: inches, feet, centimeters, or meters. Area units must be consistent with the length units (e.g.: if inches are used for length units, then square inches must be used for area units).

Table 4.2: Example of the VL0 GMETRY.DAT Geometry Format

```
212.52 10.9393 5.1642 24. .75 3
8.4 13.2 4.39 49.9979 10. 0. 0. 0.
3 3
0. 45.4545 81.8182 100.
0. 44.0476 72.6190 100.
8.4 2.4 2.4 8.1301 10. 13.2 0. 0.
2 3
0. 50. 100.
0. 44.0476 72.6190 100.
3.6 6.79 0. 49.9979 -30. 10.01 8.4 1.4811
5 1
0. 29.3881 52.8987 64.6539 82.3270 100.
0. 100.
```

Table 4.3: Key to VL0 GMETRY.DAT Geometry Format

S_{ref}	c_{ref}	X_{cref}	b_{ref}	CP	n		
B_T	C_R	C_T	L_{LE}	G	x_o	y_o	z_o
n_c	n_b						
x_1	x_2	x_3	...	x_{nc}	x_{nc+1}		
y_1	y_2	y_3	...	y_{nb}	y_{nb+1}		

The last 4 lines are repeated for each trapezoid.

Where:

- S_{ref} = Reference area
- c_{ref} = Reference chord
- X_{cref} = Location of leading edge of cref along x axis.
- b_{ref} = Reference span
- CP = Location of control point on the aerodynamic panels as fraction of mean geometric chord of that panel. Typical value is 0.75.
- n = Number of trapezoids used to define the wing (integer)
- B_T = Trapezoid span
- C_R, C_T = Trapezoid root & tip chord lengths
- L_{LE} = Trapezoid leading edge sweep angle (deg)
- G = Trapezoid dihedral (deg) $|G| < 90$ deg
- x_o, y_o, z_o = Coordinates of the leading edge of the trapezoid root chord
- n_c = Number of panels in chordwise direction on the trapezoid (integer)
- n_b = Number of panels in the spanwise direction (number of chordwise strips) on the trapezoid (integer)
- xi = Percent trapezoid chord of the i^{th} line separating aerodynamic panels in the streamwise direction, starting at the leading edge.
- yi = Percent trapezoid span of the i^{th} line separating aerodynamic panels in the spanwise direction, starting at the root chord.

Note: x_1 and y1 will always be 0, and x_{nc+1} and y_{nb+1} will always be 100.

4.3 GGP Format

The last file format that AeroLS can read in is the GGP format. GGP is an ASCII format that, like VL0, was developed at The Boeing Company. It is a simple format that allows you to specify strings of points and group them together into arrays or networks. This format, like the XML format, requires the user to specify the location of each and every corner point of each and every panel. This allows the user to specify camber and effects like continuous dihedral change. The only drawback to the GGP format, relative to the XML format, is that it only allows you to specify geometry and not any other input parameters. However, this also makes the GGP format much easier to create by hand.

An example of the GGP format can be found in Table 4.4. This is the same geometry that has been presented in Table 4.1 and Table 4.2 for the XML and VL0 formats.

The 1st line of the GGP format has a Fortran style format statement. This tells the reader what format the numbers are in. In the example "(3G15.7)" means that there are three fixed width columns of general (floating point) format numbers that are 15 characters in width with 7 characters after (including) the decimal point. All that AeroLS cares about is the number of columns (3 in this case) and the number of characters in each column (15). Other than that, AeroLS can parse almost any number format that you place into those fixed width columns.

I'm not certain what the "*COM" line means and AeroLS ignores it. The "*DUP" and "*DUPT" lines seem to indicate that the format statement on the 1st line is repeated for each set of data in the file. AeroLS assumes that the format is duplicated no matter if these lines are present or not, so these lines are also ignored. Any line that starts with an asterix (except "*EOF" and "*EOD") will be considered a comment line and ignored.

The data for the remainder of the file is presented as a series of "runs". Each run corresponds to a string of points that make up a column in a network in AeroLS. A run in a GGP file starts with the run name, which for geometry files is a combination of the network and string name. In the example, the 1st run name is "A1x1". This translates to "network #1", "string #1". AeroLS determines where one network ends and another begins by watching for a change in the "network name" part of the run name (the string name part of the run name is ignored). For AeroLS, array or network names can be anything you like as long as they end in a number. When the network name or number changes, AeroLS will start a new network.

The end of a run is denoted by the presence of a "*EOD" or "*EOF" line.

The example in Table 3.4 contains 3 networks named "A1", "A2", and "A3". The first two networks contain 4 strings each. The third network contains only two strings.

Finally, the parameters contained in each run are given labels in the line immediately following the 1st run name. In this case "X Y Z". AeroLS looks for those particular parameter names and will ignore any other parameters that a GGP file may have. Those three "X", "Y", and "Z" must be present to read the file.

Table 4.4: Example of the GGP Geometry Format

```
(3G15.7)
*COM
*DUP
*DUPT
A1x1
 X  Y  Z
      0.             0.             0.
      5.999994       0.             0.
     10.8000024      0.             0.
     13.2            0.             0.
*EOD
A1x2
      4.4091648      3.6999984       .6523890
      8.6452544      3.6999984       .6523890
     12.0341354      3.6999984       .6523890
     13.7285712      3.6999984       .6523890
*EOD
A1x3
      7.2691619      6.099996       1.0755600
     10.3610979      6.099996       1.0755600
     12.8346536      6.099996       1.0755600
     14.071428       6.099996       1.0755600
*EOD
A1x4
     10.01           8.4000000      1.4811
     12.0054526      8.4000000      1.4811
     13.601819       8.4000000      1.4811
     14.4            8.4000000      1.4811
*EOD
A2x1
     13.2            0.             0.
     14.4            0.             0.
     15.6            0.             0.
*EOD
A2x2
     13.7285712      3.6999984       .6523890
     14.9285712      3.6999984       .6523890
     16.1285712      3.6999984       .6523890
*EOD
A2x3
     14.071428       6.099996       1.0755600
     15.271428       6.099996       1.0755600
     16.471428       6.099996       1.0755600
*EOD
A2x4
     14.4            8.4000000      1.4811
     15.6            8.4000000      1.4811
     16.8            8.4000000      1.4811
*EOD
A3x1
     10.01           8.4000000      1.4811
     12.005452       8.4000000      1.4811
     13.6018217      8.4000000      1.4811
     14.3999998      8.4000000      1.4811
     15.6000033      8.4000000      1.4811
     16.8            8.4000000      1.4811
*EOD
```

Table 4.4 Continued

```
A3x2
     14.3              12.                 -.5973
     14.3              12.                 -.5973
     14.3              12.                 -.5973
     14.3              12.                 -.5973
     14.3              12.                 -.5973
     14.3              12.                 -.5973
*EOD
```

5. Analysis Output and File Formats

AeroLS calculates a number of aerodynamic properties of the specified geometry and provides a list of these parameters in the user interface. The user may then choose to view the results, interpolate (lookup) values in the results, plot the results, or save them to one of the supported output file formats (from the File menu). At this time, AeroLS calculates the aerodynamic properties listed in Table 5.1.

Table 5.1: Aerodynamic Properties Calculated by AeroLS

CN	=	Body axis normal force coefficient	+ Up	(dimensionless)
CA	=	Body axis axial force coefficient	+ Aft	(dimensionless)
CYb	=	Body axis side force coefficient	+ To Starboard	(dimensionless)
CMb	=	Body axis pitching moment coefficient	+ Nose Up	(dimensionless)
Cnb	=	Body axis yawing moment coefficient	+ Nose Right	(dimensionless)
Clb	=	Body axis rolling moment coefficient	+ RT Wing Down	(dimensionless)
CL	=	Wind axis lift force coefficient	+ Up	(dimensionless)
CD	=	Wind axis drag force coefficient	+ Aft	(dimensionless)
CY	=	Wind axis side force coefficient	+ To Starboard	(dimensionless)
CM	=	Wind axis pitching moment coefficient	+ Nose Up	(dimensionless)
Cn	=	Wind axis yawing moment coefficient	+ Nose Right	(dimensionless)
Cl	=	Wind axis rolling moment coefficient	+ RT Wing Down	(dimensionless)
CLSA	=	Stability axis lift force coefficient	+ Up	(dimensionless)
CDSA	=	Stability axis drag force coefficient	+ Aft	(dimensionless)
CYSA	=	Stability axis side force coefficient	+ To Starboard	(dimensionless)
CMSA	=	Stability axis pitching moment coef.	+ Nose Up	(dimensionless)
CnSA	=	Stability axis yawing moment coef.	+ Nose Right	(dimensionless)
ClSA	=	Stability axis rolling moment coef.	+ RT Wing Down	(dimensionless)
CLalpha	=	Lift curve slope: $C_{L_\alpha} = \frac{\partial C_L}{\partial \alpha}$	alpha=0, beta=0	(1/radians)
CMalpha	=	Pitching moment slope: $C_{M_\alpha} = \frac{\partial C_M}{\partial \alpha}$	alpha=0, beta=0	(1/radians)
Xac	=	X coord of aero center location	alpha=0, beta=0	(meters)
CYbeta	=	Side force curve slope: $C_{Y_\beta} = \frac{\partial C_Y}{\partial \beta}$	alpha=0, beta=0	(1/radians)
Cnbeta	=	Yawing moment slope: $C_{n_\beta} = \frac{\partial Cn}{\partial \beta}$	alpha=0, beta=0	(1/radians)
Clbeta	=	Rolling moment slope: $C_{l_\beta} = \frac{\partial C_l}{\partial \beta}$	alpha=0, beta=0	(1/radians)
CLq	=	Lift due to pitch rate: $C_{L_{\hat{q}}} = \frac{\partial C_L}{\partial \hat{q}}$	alpha=0, beta=0	(1/radians)
CMq	=	Pitch damping: $C_{M_{\hat{q}}} = \frac{\partial C_M}{\partial \hat{q}}$	alpha=0, beta=0	(1/radians)
CYr	=	Side force due to yaw rate: $C_{Y_{\hat{r}}} = \frac{\partial C_Y}{\partial \hat{r}}$	alpha=0, beta=0	(1/radians)
Cnr	=	Yaw Mom due to yaw rate: $C_{n_{\hat{r}}} = \frac{\partial C_n}{\partial \hat{r}}$	alpha=0, beta=0	(1/radians)
Clr	=	Roll Mom due to yaw rate: $C_{l_{\hat{r}}} = \frac{\partial C_l}{\partial \hat{r}}$	alpha=0, beta=0	(1/radians)
CYp	=	Side force due to roll rate: $C_{Y_{\hat{p}}} = \frac{\partial C_Y}{\partial \hat{p}}$	alpha=0, beta=0	(1/radians)
Cnp	=	Yaw Mom due to roll rate: $C_{n_{\hat{p}}} = \frac{\partial C_n}{\partial \hat{p}}$	alpha=0, beta=0	(1/radians)
Clp	=	Roll Mom due to roll rate: $C_{l_{\hat{p}}} = \frac{\partial C_l}{\partial \hat{p}}$	alpha=0, beta=0	(1/radians)

AeroLS can write out analysis results to one of two ASCII file formats. One is a specially organized comma separated value (CSV) table format that can be easily read into a spreadsheet program such as Microsoft Excel or OpenOffice. The other is a more general, but more complex table format known as Functional Data Handling System ASCII (FDA or FDHS). Each will be briefly described below, but will not be completely defined here. It's best if you use examples output from AeroLS to work out the details.

5.1 CSV Format

The specially organized CSV format used by AeroLS is convenient for moving data to and from spreadsheet programs and it is very easy to read and create by hand. You can see an example of it in Table 5.2.

The CSV format used by AeroLS has one major limitation: all the tables of data stored in any one file must have exactly the same set of breakpoints. This limitation makes it a little troublesome for outputting results from AeroLS because AeroLS creates force & moment tables that are functions of Mach number and angle of attack, and it creates stability derivative tables that are only a function of Mach number. This means that these two different types of tables can not be saved to the same CSV file, but must be saved to separate CSV files. If you attempt to save a stability derivative and a force or moment table to the same CSV file, AeroLS will stop you and remind you that all the tables must have the same set of breakpoints.

In a CSV table file (see Table 5.2), the 1st four lines of the file are used for comments and can contain anything you want. AeroLS writes in the reference quantities used to non-dimensionalize the data into the first line.

The fifth line contains the number of breakpoints in each independent variable. In this case, there are 3 Mach breakpoints and 9 angle of attack breakpoints. Independent variables must be in the left most columns and technically they can occur in any order and have any number of independent variables although AeroLS, at this time, only writes out a maximum of two (Mach and Alpha; in that order).

The sixth line contains, first, the names of all the independent variables ("Mach", "Alpha") and then the names of all the dependent data tables contained in the file. In the example case, the dependent tables are CN, CA, CMb, CL, CD, and CM. There can be any number of dependent data tables in one file, but they all must share the same set of breakpoints.

The actual data begins on the seventh line. The independent data values are in the leftmost columns and the breakpoints must be ordered from the most negative value to the most positive value. When there is more than one independent variable, all the combinations of each independent variable must occur in order (i.e.: cycle through all the Alphas, then go to the next Mach number and cycle through all the Alphas again, etc.). The dependent values that correspond to each combination of the independent variables is then placed in the appropriate column.

Table 5.2: Example of a CSV Formatted Table Output File

References: Sref = 212.52002 ft?, cref = 10.9393 ft, bref = 24.0 ft, Xmrc = 7.899 ft, Ymrc = 0.0 ft, Zmrc = 1.32 ft

3	9						
Mach	Alpha	CN	CA	CMb	CL	CD	CM
0	-2	-0.10962	0	0.00848	-0.10955	0.00383	0.00848
0	0	0	0	0	0	0	0
0	2	0.10962	0	-0.00848	0.10955	0.00383	-0.00848
0	4	0.2191	0	-0.01694	0.21857	0.01528	-0.01694
0	6	0.32832	0	-0.02539	0.32653	0.03432	-0.02539
0	10	0.54543	0	-0.04218	0.53714	0.09471	-0.04218
0	12	0.65305	0	-0.0505	0.63878	0.13578	-0.0505
0	14	0.75988	0	-0.05876	0.7373	0.18383	-0.05876
0	16	0.86578	0	-0.06695	0.83224	0.23864	-0.06695
0.6	-2	-0.12022	0	0.01066	-0.12015	0.0042	0.01066
0.6	0	0	0	0	0	0	0
0.6	2	0.12022	0	-0.01066	0.12015	0.0042	-0.01066
0.6	4	0.24029	0	-0.02131	0.23971	0.01676	-0.02131
0.6	6	0.36007	0	-0.03193	0.3581	0.03764	-0.03193
0.6	10	0.59817	0	-0.05304	0.58909	0.10387	-0.05304
0.6	12	0.7162	0	-0.0635	0.70055	0.14891	-0.0635
0.6	14	0.83336	0	-0.07389	0.8086	0.20161	-0.07389
0.6	16	0.9495	0	-0.08419	0.91272	0.26172	-0.08419
0.8	-2	-0.13284	0	0.01384	-0.13276	0.00464	0.01384
0.8	0	0	0	0	0	0	0
0.8	2	0.13284	0	-0.01384	0.13276	0.00464	-0.01384
0.8	4	0.26552	0	-0.02767	0.26487	0.01852	-0.02767
0.8	6	0.39787	0	-0.04146	0.39569	0.04159	-0.04146
0.8	10	0.66097	0	-0.06888	0.65092	0.11478	-0.06888
0.8	12	0.79138	0	-0.08247	0.77409	0.16454	-0.08247
0.8	14	0.92084	0	-0.09596	0.89349	0.22277	-0.09596
0.8	16	1.04917	0	-0.10934	1.00853	0.28919	-0.10934

5.2 FDA Format

The Functional Data Handling System ASCII (FDA or FDHS) format was developed and is used by The Boeing Company. It is the most complete table format that any of my programs support. However, as a result, it is a little more difficult to create by hand than the CSV format. An example of the FDA format can be found in Table 5.3. This is the same data set found in the CSV file in Table 5.2.

FDA format files can hold any number of tables, each with different independent variables, each with a different number of breakpoints. As a result, a single FDA formatted file can hold all the tables output from AeroLS at the same time. This is the main advantage this format has over the CSV format. Technically, FDA files are limited to tables with 6 independent variables or less, however, AeroLS will write out a maximum of two independent variables (Mach and Alpha).

The first line of the file should be "*FDHS" to identify this as an FDA formatted file. Any line that starts with a semicolon, ";", is considered a comment line and is ignored. AeroLS writes in a few comments at the start of the file to list out the reference quantities used for non-dimensionalization.

Each table is listed out into the file, one after the other. Each table entry starts with the name of the table variable ("CN" for the 1st entry in Table 5.2). The table name is followed by the name of each independent variable, followed by the number of breakpoints in each independent variable.

The next line in each table entry should always read "all   all". These entries have meaning only in a very specific context and should otherwise read "all".

Next, the breakpoints for the last independent variable (given on the table name line) should be listed out in any floating point, space delimited or fixed column (with spaces between columns) format with a maximum of 6 columns of numbers per line. This should be followed by the breakpoints of the next independent variable and so on until the breakpoints for the 1st independent variable in the name line are entered.

Finally, the table values are written out, cycling through all the independent variable breakpoints from the 1st independent variable on the name line to the last. The numbers are again entered in any floating point, space delimited or fixed width column format with a maximum of 6 columns per line.

Table 5.3: Example of a FDA Formatted Table Output File

```
*FDHS
; References:   Sref = 212.52002 ft?, cref = 10.9393 ft, bref = 24.0 ft
; Xmrc = 7.899 ft, Ymrc = 0.0 ft, Zmrc = 1.32 ft
;  CN(Mach,Alpha)
CN      Alpha      Mach      9      3
all        all
         0.0000         0.6000         0.8000
        -2.0000         0.0000         2.0000         4.0000         6.0000
10.0000
        12.0000        14.0000        16.0000
        -0.1096         0.0000         0.1096         0.2191         0.3283
0.5454
         0.6531         0.7599         0.8658
        -0.1202         0.0000         0.1202         0.2403         0.3601
0.5982
         0.7162         0.8334         0.9495
        -0.1328         0.0000         0.1328         0.2655         0.3979
0.6610
         0.7914         0.9208         1.0492
;  CA(Mach,Alpha)
CA      Alpha      Mach      9      3
all        all
         0.0000         0.6000         0.8000
        -2.0000         0.0000         2.0000         4.0000         6.0000
10.0000
        12.0000        14.0000        16.0000
         0.0000         0.0000         0.0000         0.0000         0.0000
0.0000
         0.0000         0.0000         0.0000
         0.0000         0.0000         0.0000         0.0000         0.0000
0.0000
         0.0000         0.0000         0.0000
         0.0000         0.0000         0.0000         0.0000         0.0000
0.0000
         0.0000         0.0000         0.0000
;  CMb(Mach,Alpha)
CMb      Alpha      Mach      9      3
all        all
         0.0000         0.6000         0.8000
        -2.0000         0.0000         2.0000         4.0000         6.0000
10.0000
        12.0000        14.0000        16.0000
         0.0085         0.0000        -0.0085        -0.0169        -0.0254        -
0.0422
        -0.0505        -0.0588        -0.0670
         0.0107         0.0000        -0.0107        -0.0213        -0.0319        -
0.0530
        -0.0635        -0.0739        -0.0842
         0.0138         0.0000        -0.0138        -0.0277        -0.0415        -
0.0689
        -0.0825        -0.0960        -0.1093
```

Table 5.3 Continued

```
;  CL(Mach,Alpha)
CL      Alpha      Mach      9      3
all         all
         0.0000         0.6000         0.8000
        -2.0000         0.0000         2.0000         4.0000         6.0000
10.0000
        12.0000        14.0000        16.0000
        -0.1096         0.0000         0.1096         0.2186         0.3265
0.5371
         0.6388         0.7373         0.8322
        -0.1201         0.0000         0.1201         0.2397         0.3581
0.5891
         0.7006         0.8086         0.9127
        -0.1328         0.0000         0.1328         0.2649         0.3957
0.6509
         0.7741         0.8935         1.0085
;  CD(Mach,Alpha)
CD      Alpha      Mach      9      3
all         all
         0.0000         0.6000         0.8000
        -2.0000         0.0000         2.0000         4.0000         6.0000
10.0000
        12.0000        14.0000        16.0000
         0.0038         0.0000         0.0038         0.0153         0.0343
0.0947
         0.1358         0.1838         0.2386
         0.0042         0.0000         0.0042         0.0168         0.0376
0.1039
         0.1489         0.2016         0.2617
         0.0046         0.0000         0.0046         0.0185         0.0416
0.1148
         0.1645         0.2228         0.2892
;  CM(Mach,Alpha)
CM      Alpha      Mach      9      3
all         all
         0.0000         0.6000         0.8000
        -2.0000         0.0000         2.0000         4.0000         6.0000
10.0000
        12.0000        14.0000        16.0000
         0.0085         0.0000        -0.0085        -0.0169        -0.0254         -
0.0422
        -0.0505        -0.0588        -0.0670
         0.0107         0.0000        -0.0107        -0.0213        -0.0319         -
0.0530
        -0.0635        -0.0739        -0.0842
         0.0138         0.0000        -0.0138        -0.0277        -0.0415         -
0.0689
        -0.0825        -0.0960        -0.1093
```