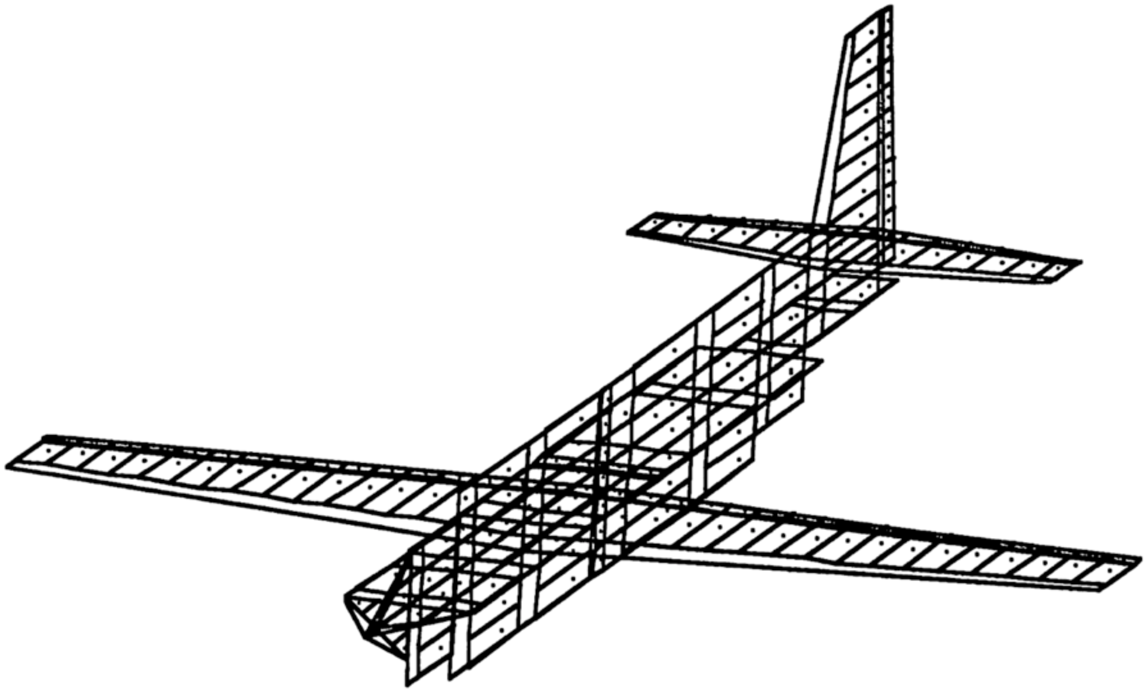


LinAir 4

A Nonplanar, Multiple Lifting Surface Aerodynamics Program

Revision Date: February 8, 2011

by Ilan Kroo

Desktop Aeronautics, Inc.

1900 Embarcadero Road, Ste. 101
Palo Alto, CA 94303

Table of Contents

Introduction

Overview

LinAir is a program for computing the aerodynamic characteristics of multi-element, nonplanar lifting surfaces. It can be used to determine the appropriate wing twist for a new design, the expected performance of a given wing geometry, the proper angles of incidence for tail or canard surfaces, or the stability characteristics of a new configuration.

LinAir was developed first in 1982 and was modified over the following years for various applications. It has been run on computers ranging from small laptops to VAX's and Cray's. LinAir is now used in courses at several universities, by companies such as Boeing, AeroVironment, Northrop, and Lockheed, and by researchers at NASA to obtain a quick look at new design concepts.

LinAir is clearly not the last word in aircraft aerodynamic analysis methods. Commercial packages are available from companies such as Boeing Computer Services and Analytical Methods, Inc. that provide more accurate results and permit modeling of more arbitrary bodies. These programs are, however, much larger, more difficult to use, and orders of magnitude more costly. LinAir is intended to bridge the gap between these big codes and the simple, approximate methods often used in advanced design.

This manual describes the basic use of the program, the theory on which the calculations are based, and several examples illustrating the accuracy of the method in various applications. The program is very easy to use, but the calculations are complex and there are many subtleties. Please read this manual before running the program (the theory section is optional).

About the New Version

For version 4, LinAir has been re-written as a Java 2 application. Aside from the obvious benefit of cross-platform compatibility, this allows the code to be modular in nature and hence easily extensible, to the point that a user with some background in object-oriented programming and the LinAir SDK (release date still TBD) would be able to write custom plug-ins.

The changes and/or additions relative to LinAir 3 can be summarized as follows:

- The user interface has received a complete overhaul.
- The number of panels allowed is now 60.
- An additional section property to define lift at $\alpha = 0$ has been added.
- Section properties are now defined at both the panel root and tip.

- Panel twist is now defined by providing panel root and tip incidences (as in LinAir 1.4).
- Input file format is now XML.
- A file converter has been incorporated to translate from older style LinAir input files to the newer XML format.

The base distribution includes the following components:

- **Geometry:** Geometry provides facilities for generating and editing LinAir geometries and defining flow properties. An interactive 3-D view of the geometry is provided.

- **Lift Distribution:** Computes the lift and C_l distribution of each element. Provides both plots and a table of relevant results.

- **Element Forces:** Computes the forces and moments on each element and provides a table of relevant results.

- **Alpha Sweep:** Plots an alpha sweep of the geometry at a particular Mach number and over a given alpha range. The user has the option to plot any combination of the overall forces and moments.

- **View XML Data:** Analogous to the *Input File* component from previous versions of LinAir. Displays the input file in raw form.

- **Convert Input Files:** Imports older format (i.e. pre LinAir 4) input files and generates equivalent XML files.

The Pro version of LinAir 4 also includes a tool for rapid parameter trade studies, integration with a non-linear constrained optimizer, stability derivative computations and a linear dynamics package.

Update Policy and Technical Support

Registered users of LinAir may receive updates to the program including bug fixes or compatibility improvements if they have an active support and maintenance agreement in place. Substantial revisions of the program may be offered at a reasonable upgrade price.

Registered users may also call to request technical support. This includes questions about the program and its use, but does not include aerodynamic consulting services – you are on your own as far as figuring out what the answers mean or how best to panel a particular configuration. If you suspect that you have found a bug (these things happen although we have tested this program for some time) please call and let us know so that we can suggest a workaround, let others know about it, and correct it in the next upgrade. One easy way to have problems solved is to FAX or e-mail a copy of your input file and the puzzling results to support@desktopaero.com. We'll try to run the case here and get back to you soon with a diagnosis.

We would appreciate letters with comments about the program or interesting applications you have found. Thanks for your interest.

References

1. Pearson, H., Anderson, R., "Calculation of the Aerodynamic Characteristics of Tapered Wings with Partial-Span Flaps," NACA Rpt. 665, 1939.

2. Schlichting, H., Truckenbrodt, E., Aerodynamics of the Airplane, McGraw Hill, 1979.

3. McCormick, B., Aerodynamics, Aeronautics, and Flight Mechanics, Wiley, 1979.

4. Abbott, I. H., Von Doenhoff, A. E., Theory of Wing Sections, McGraw Hill, 1949, Dover Edition, 1959.

5. Ashley, H., Landahl, M., Aerodynamics of Wings and Bodies, 1965, Dover Edition, 1985.

Getting Started

System Requirements

Aside from the basic minimums required to run any modern operating system, LinAir 4.3 requires the following:

- A Java Runtime Environment (JRE), version 1.6 or higher (also referred to as Java 6 or JRE6). This includes most versions of Windows, Solaris, Linux, and Mac OS. The JRE is freely available for download at http://www.java.com.
- A color monitor with a resolution of at least 1024x768.

Installing and Running LinAir 4.3

Windows
1. Copy the LINAIR/XP32 or LINAIR/XP64 directory to your local harddisk, depending on if you will run LinAir with a 32 or 64 bit JVM. If you are not sure which to use, try the XP32 first.

2. Place the license file license.lic in the LINAIR/XP32/lib or LINAIR/XP64/lib directory.

3. Launch LinAir.exe to begin

Mac OSX
1. Copy the LINAIR/OSX64 directory to your local harddisk.

2. Place the license file license.lic in the LINAIR/OSX64 directory.

3. Launch LinAir.app to begin

> Note: If LinAir does not start due to the Java version being out of date, make sure you download the lastest version using Apple’s Software Update mechanism. The version may also need to be selected as the default using the Java Preferences.app located in /Applications/Utilities.

Linux
1. LinAir is a Java program with a graphical user interface and must be used in a desktop environment. Make sure you have Java 1.6 installed and included in your PATH.

2. Copy the LINAIR/LINUX64 directory to your local harddisk.
Place the license file license.lic in the LINAIR/LINUX64/lib directory.

3. Launch the LinAirLauncher class by either running the provided LinAir.sh script, or entering the following command in a terminal window, from the LINAIR/LINUX64 directory.

```
java -cp lib LinAirLauncher
```

Older Versions of LinAir 4.

- Older releases were installed as a Java jar file. Those can be run from the command line as follows:

  From within the install directory, issue the following command:

  ```
  java –jar LinAir.jar
  ```

- Presuming install defaults were kept, LinAir will be accessible from the "Start" menu for Windows users. For Mac OS X users, LinAir will be in the Applications folder.

- Known Issue on Macs: The original release of LinAir 4 under Mac OS X, required an older Java 1.3.1 VM. Should your version appear to have this issue, contact support@desktopaero.com for an updated copy
  .

Using LinAir

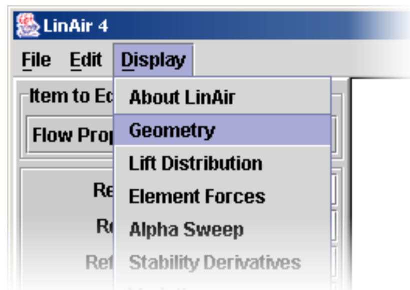
Quick Start

This section is merely intended as a quick survey of the capabilities offered by LinAir. For an in-depth look at the use of each of LinAir's bundled applets, please refer to the section entitled User's Guide.

The LinAir interface follows the general GUI conventions, providing "File" and "Edit" menus containing most, if not all, of the expected functions. The third menu, "Display," allows the user to switch between the available applets.

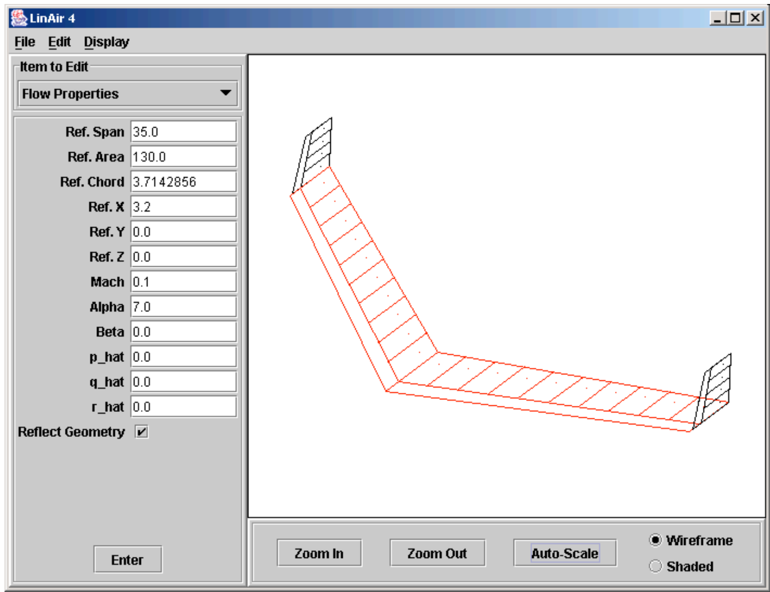
- About LinAir:

  On startup, LinAir displays an "About LinAir" screen with a short program description along with the build version and date. As this particular applet does not provide much in the way of interactivity, the user can immediately employ the handy Display menu to switch to the "Geometry" applet.

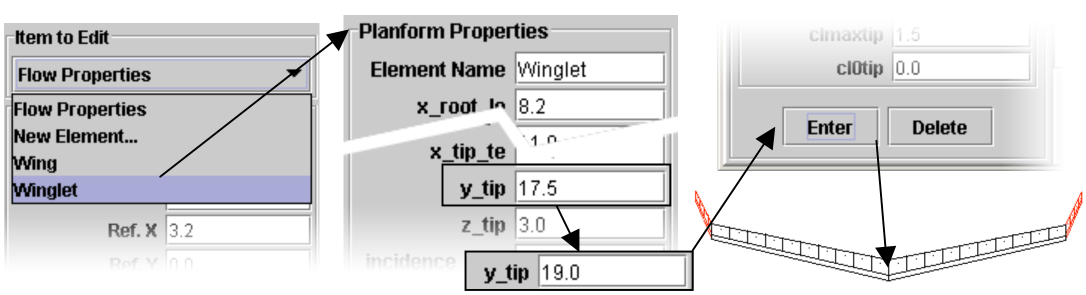


- Geometry:

  Unless the user has loaded an input file (this guide assumes you have not), LinAir will default to a simple wing with winglets geometry. As with most other applets in LinAir, fields and selections for user input are provided on the left, and interactive results are displayed on the right. In this case, the interactive result happens to be a 3D rendition of the current geometry.



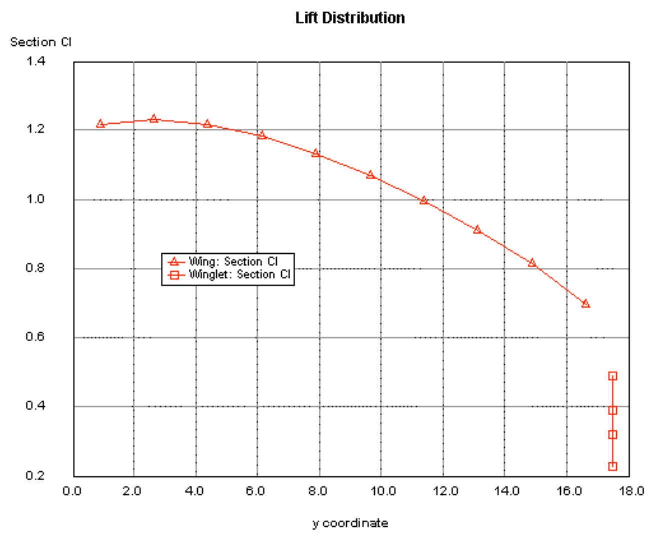
Individual geometry elements can be selected and edited using the pull-down in the upper left corner identified as “Item to Edit.” The adventurous user may opt to modify the geometry slightly. By selecting “Winglet” from the list of items in the “Item to Edit” pull-down, a list of the geometric and aerodynamic properties of the winglet is displayed. Having decided that the winglets should be canted out slightly, the user may increase the y_tip value from 17.5 to 19.0. The change is committed by clicking the “Enter” button. A quick look at the geometry should reveal that this has had the expected effect.



If satisfied with the current geometry, the Display menu may be used to switch to the Lift Distribution applet.

- Lift Distribution:

  This applet generates a plot of the lift distribution for the current geometry. As stated previously for a typical applet, user interaction is provided for on the left and computed results displayed on the right. To compute the lift distribution using the default conditions, click the “Compute” button. Results should appear almost instantaneously, and in addition to the total forces and moments listed in the Results section (presuming no major changes from the default geometry), should look something like the following:

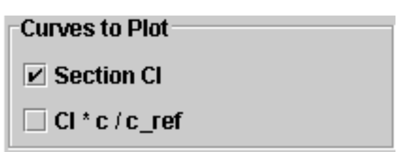


Data in tabular form can be accessed using the tabs:

Plot | Table | Table (Tab-delimited)

Element	Panel	y coordinate	Section Cl	Cl * c / Cavg
Wing	1	0.875	1.21654070410...	1.27683654286...
Wing	2	2.625	1.23324471225...	1.26684214518...
Wing	3	4.375	1.21823412364...	1.22323489312...
Wing	4	6.125	1.18218091754...	1.15856630639...
Wing	5	7.875	1.13122938109...	1.08085590995...
Wing	6	9.625	1.06873337610...	0.99466674878...
Wing	7	11.375	0.99604607299...	0.90245574065...
Wing	8	13.125	0.91250503991...	0.80504631667...
Wing	9	14.875	0.81453940076...	0.70199357067...
Wing	10	16.625	0.69772733622...	0.60006289554...
Winglet	1	17.5	0.48892136217...	0.39104642697...
Winglet	2	17.5	0.39121256922...	0.28304695165...
Winglet	3	17.5	0.31862219530...	0.20841803067...
Winglet	4	17.5	0.22691475461...	0.13420825980...

In addition to section C_l, $C_l * c / c_{ref}$ can also be plotted using the checkboxes:



- Element Forces:

  This applet provides tabulated results of the forces and moments for each element. Clicking “Compute” will initiate the analysis; results should be produced very quickly.

- Alpha Sweep:

Very similar to the Lift Distribution applet in terms of interface, this applet produces an alpha sweep of the geometry within the given bounds. Click compute for a look at the plot. Feel free to clear up or embellish the plot by selectively suppressing or enabling curves (by un-checking or checking the appropriate checkboxes).

- View XML Data:

  This applet displays the contents of the XML definition file for this geometry. For the moment, suffice to say that the details of this file are substantially beyond the scope of this quick guide.

- Convert Input Files:

  Finally, this applet is used to convert older (i.e. pre LinAir 4) format input files to the LinAir 4 XML format.

This concludes the quick start tour!

User's Guide

The purpose of this section is to discuss in depth the usage of each of the bundled LinAir 4 applets. This includes definitions of each of the input parameters and result fields, applet-specific options and where applicable, how to interact with the results. It does not include any discussion of the theory involved. For that, please refer to the Theory section.

Generally Relevant Items:

- File Menu:

Open...	By selecting this item you will be asked for the name of an input file via the standard file dialog box. Select the name of a pre-prepared input file and then click on the Open button. If you change your mind, click on the Cancel button.
Save	This causes the current working file to be saved to disk. If no file is open, a new file is created and saved under the name "LinAir_Geometry.xml."
Save As…	Brings up a standard file dialog that allows the user to save the current working file to disk using a specified name.
Page Setup...	Allows you to set the printer settings before printing. It is usually not necessary to change these settings but if you want to print a reduced size image or print the image rotated on the screen, select the appropriate settings.
Print...	Sends the current screen image to the default printer. Before printing you will be presented with a dialog box from which to select the printing quality, number of pages, etc. It is usually not necessary to change these defaults and printing begins after you click on the OK button.
Quit	Stops LinAir and closes all related files. Note that this will not save working files automatically. Please save your file before quitting!

- Edit Menu:

Cut	Cuts the currently highlighted text and places it in the system clipboard.
Copy	Copies the currently highlighted text to the system clipboard.

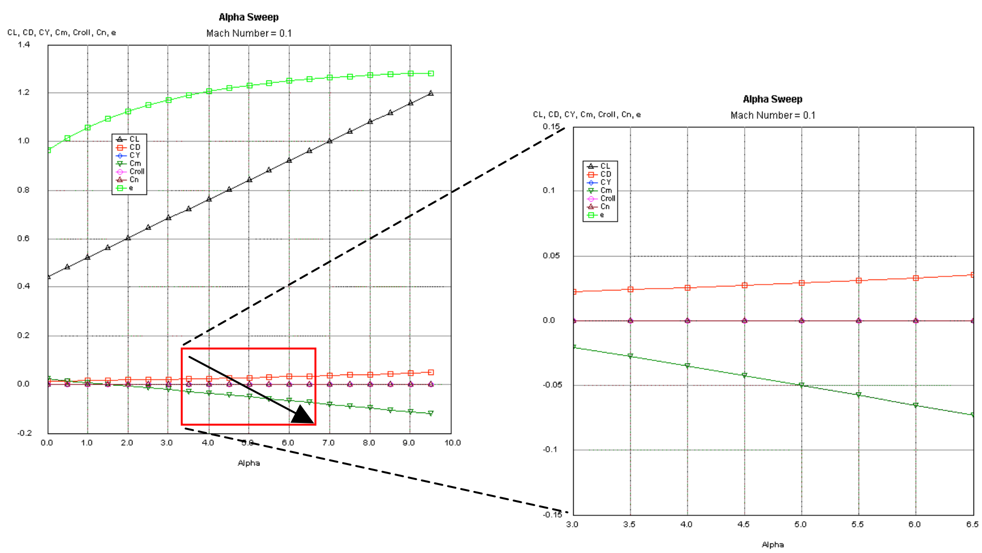
Paste	Pastes the current contents of the clipboard into the current field.
Clear	Clears the contents of the current field.
Select All	Highlights all text within the scope of the current field.

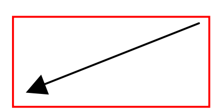
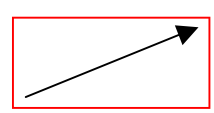
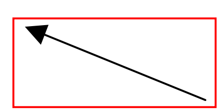
Note: these commands are only relevant when applied to text.

- Plots:

  Plots in LinAir are reasonably interactive, allowing for zooming, selective or collective reversal of axes, re-scaling and arbitrary placement of the legend box. The available actions are:

  - Zooming: Zooming is accomplished by clicking and dragging on the plot. A bounding box appears, which defines the region that will be expanded to fill the plot.



  - Selective/Collective Reversal of Axes: Similar to zooming, this is accomplished by clicking and dragging to define a bounding box. However, rather than dragging in the conventional south-east direction, the bounding box is defined as follows:
    - Dragging in a south-westerly direction will flip the x-axis. 
    - Dragging in a north-easterly direction will flip the y-axis. 
    - Finally, dragging in a north-westerly direction will collectively flip both the x- and y-axes: 
  - Re-scaling plots is accomplished by Shift-Clicking anywhere on the plot.

- Legends can be dragged around freely anywhere within the plot.

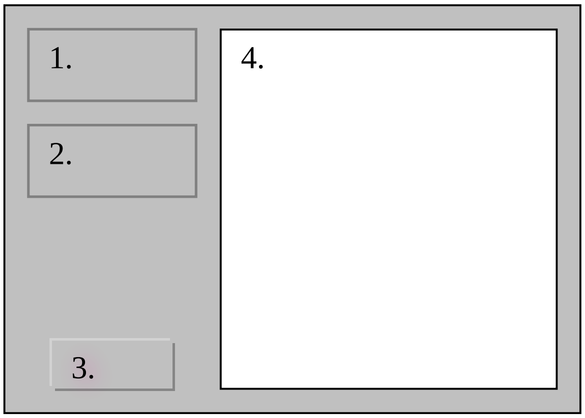
- Tables:

  Tables in LinAir are superficially similar to spreadsheets, in that cell ranges can be arbitrarily selected and copied into a spreadsheet program. Note that spreadsheet functionality is not provided – the likeness is merely visual!

  Some versions of the Mac OS X Java VM do not allow for copying and pasting from tables. As a workaround for this, all applets with tables also have a “Table (Tab-delimited)” tab that outputs the table in tab-delimited plain text format, which can be safely copied and pasted.

- Applets:

  Where it is reasonable to do so, LinAir uses a standardized base user interface that follows the following convention:



1. An “Inputs” section, where a number of user-editable inputs relevant to the calculation being made are listed.
2. A “Results” section, where a select group of results is listed.
3. A “Compute” button, which commits any changes made by the user in section 1.), then computes (or re-computes) the results.
4. An output region, which serves as the main area to present results such as plots and tables.

In some cases, the “Inputs” and/or “Results” sections may be redundant or unnecessary and will be omitted. On occasion, a third section may be added below “Inputs” and “Outputs”. An example of this is the “Lift Distribution” applet, which adds a “Curves to Plot” section that allows the user to interactively enable and disable the different types of curves.

About LinAir

The program splash page; refer here for version and build information when contacting Desktop Aeronautics, Inc. with support queries.

Geometry

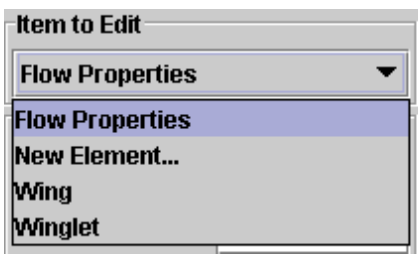
This applet is used to build and edit LinAir geometries, and also provides an interactive 3D visualization of the current state of the geometry.

Variable values are changed by typing the desired values in the labeled text fields. Values can be changed singly or as a group, but always remember that following an edit, the “Enter” button must be pressed to commit these changes.

As they are not discussed here, please refer to the section “About the Variables” for a list of variable definitions.

<u>Editing the Flow Properties:</u>

Flow properties can be accessed by selecting the “Flow Properties” item in the “Item to Edit” pull-down menu.



In addition to the list of flow properties, a checkbox labeled “Reflect Geometry” is also present. When selected, this checkbox indicates that the geometry should be reflected about the x-z plane, producing a symmetric geometry. Leave this unchecked for asymmetric geometries.

Note that the applet will default to the Flow Properties page when accessed through the Display menu.

<u>Editing an Individual Element:</u>

The geometric properties of an individual element may be accessed by selecting an element name from the “Item to Edit” pull-down menu. The properties of an element are separated into two sub-groups: “Planform Properties” and “Section Properties.”

Pressing the “Delete” button will delete an existing element.

Adding an Element to the Geometry:

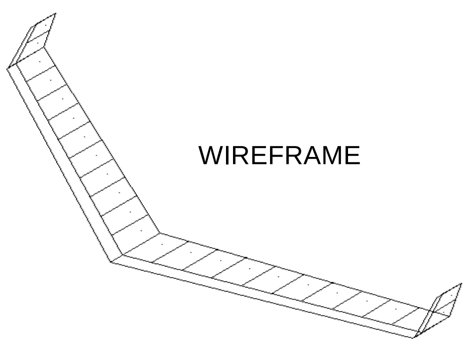
Elements can be added to the geometry by selecting the "New Element…" item from the "Item to Edit" pull-down menu. Definition of planform and section properties is exactly as for the editing of pre-existing elements, excepting of course that undefined elements cannot be deleted.

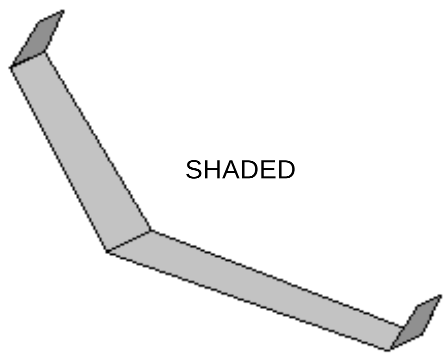
Interacting with the 3D View:

The 3D rendition of the geometry can be directly rotated about its geometric center by clicking and dragging anywhere in the view window. Some additional behaviors can be accessed through the series of action buttons below the view:

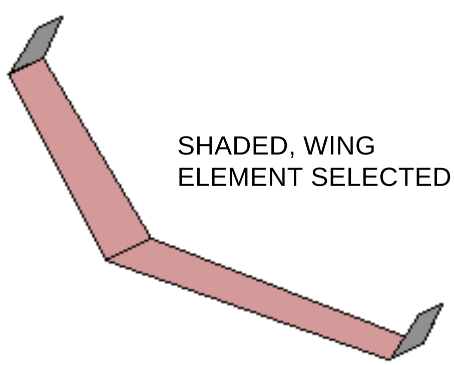
Zoom In	Zooms into the geometry, enlarging it in the view window.
Zoom Out	Zooms out from the geometry, shrinking it in the view window.
Auto-Scale	Resets the geometry to its default size.

The "Wireframe" and "Shaded" radio buttons switch from a wireframe view to a flat-shaded view.





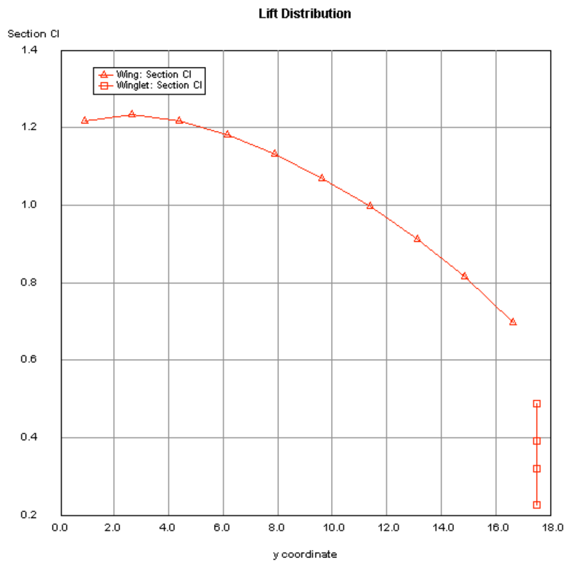
Finally, when editing individual elements, the selected element is highlighted red.

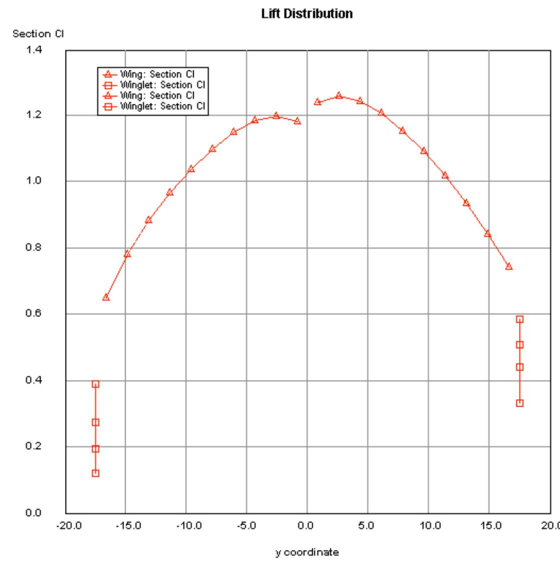


Lift Distribution

The Lift Distribution applet provides a plot of the computed lift and C_l distributions, along with an accompanying table of the lift and C_l data from which the plot is generated. Note that the local lift is nondimensionalized by the average chord and freestream dynamic pressure.

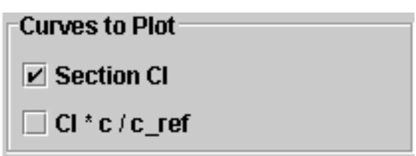
For conditions where both the flow and geometry are symmetric, only the half-span data is generated and plotted. Full-span data and plot is generated in all other cases.





Selecting Which Curves to Plot:

This applet provides the section – "Curves to Plot" – (in addition to the standard "Inputs" and "Results" sections) that allows the user to select which curves should be plotted.



To show a particular curve, select the associated checkbox; to suppress, deselect the checkbox.

Displaying Tabulated Lift Distribution Data:

To display tabulated data, select the "Table" tab.

Plot | Table | Table (Tab-delimited)

Element	Panel	y coordinate	Section Cl	Cl * c / Cavg
Wing	1	0.875	1.216540704109...	1.276836542866...
Wing	2	2.625	1.233244712258...	1.266842145187...
Wing	3	4.375	1.218234123648...	1.223234893123...
Wing	4	6.125	1.182180917541...	1.158566306399...
Wing	5	7.875	1.131229381090...	1.080855909958...
Wing	6	9.625	1.068733376109...	0.994666748782...
Wing	7	11.375	[illegible]	[illegible]

Note that the y-coordinate listed is the location of the panel vortex, and not an edge of the panel.

Element Forces

This applet provides a table of the force and moment contributions of each element making up the total geometry. Note that the reference lengths used to nondimensionalize the element forces and moments are the configuration reference values, not the element dimensions. So, for example, small surfaces will have small CL contributions even though the CL based on the element surface area may be large.

Table | Table (Tab-delimited)

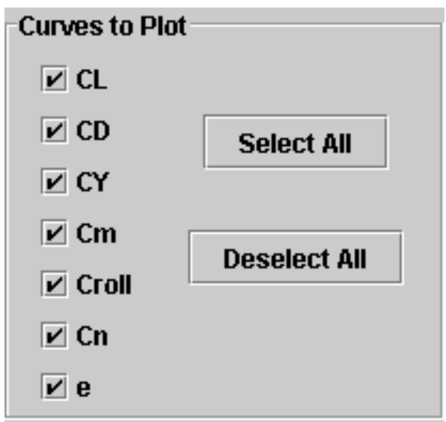
Elemen...	x_root LE	y_root LE	z_root LE	CL	CD	CY	Croll	Cm	Cn
Wing	0.0	0.0	0.0	0.5124...	0.0216...	0.0202...	-0.111...	-0.0405...	-0.011...
Winglet	8.2	17.5	0.0	0.0033...	-0.0023...	-0.0280...	-0.002...	-0.0055...	0.0025...
Wing	0.0	-0.0	0.0	0.4829...	0.0202...	-0.0206...	0.1040...	-0.0332...	0.0118...
Winglet	8.2	-17.5	0.0	0.0011...	-0.0018...	0.0149...	9.6639...	-0.0016...	-0.001...

As for the Lift Distribution applet, cases of symmetric flow and geometry will produce results for the half-span only.

Alpha Sweep

This applet provides a quick way to obtain forces and moments over a range of angles of attack. Provided in the "Inputs" section with a freestream Mach number, and low and high limits for the angle of attack (alpha), the applet will loop from the minimum to the maximum angle, and generate a plot from the results. The "Inputs" section also allows for the specification of a particular alpha value; this angle of attack is used to generate the data for the "Results" section.

<u>Selecting Which Curves to Plot:</u>
As for the "Lift Distribution" applet, a "Curves to Plot" section is provided, with the added option of being able to select and deselect all curves in one step using the "Select All" and "Deselect All" buttons.



As before, to show a particular curve, select the associated checkbox; to suppress, deselect the checkbox.

View XML Data

This applet renders a dynamic representation of the XML definition file that reflects the current state of the geometry, and may therefore not necessarily represent the loaded file (if one is loaded).

<u>Editing the File</u>
The contents of the file can be directly manipulated and the changes propagated into memory by clicking the "Apply Changes" button. Note that this requires some understanding of what the various terms in the XML file mean.

<u>Creating New Input Files</u>
Note also that an XML file will be displayed even if no file has been loaded, which makes this applet very useful for generating new input files. It is of course also

entirely possible to create input files by hand, either from scratch or by using another input file as a template.

Convert Input Files

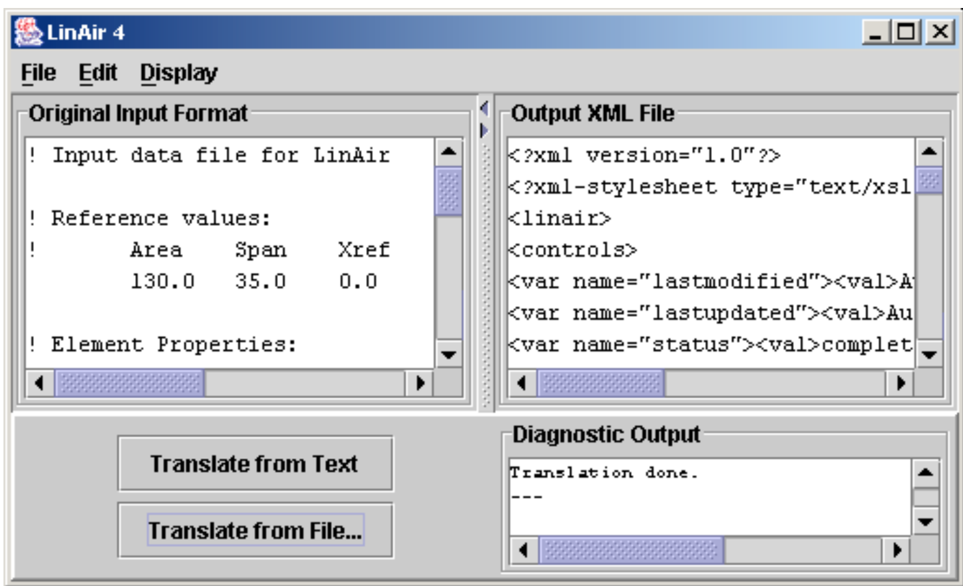
This applet translates older format LinAir input files into the current (XML) format. Note that the translated file is loaded into memory immediately, replacing whatever geometry is currently loaded.

<u>The "Translate from Text" Button</u>

The first method for translating an input file is by cutting and pasting the text of the older file into the left text pane then clicking "Translate from Text". The translated input file will be displayed in the right text pane, and provided the translation was successful, the geometry defined by the input file will be loaded.

<u>The "Translate from File…" Button</u>

The second method for translating an input file is to click the "Translate from File…" button and select the older format input file in the file dialog that is presented. As for the first method, the translated input file appears in the right pane and the geometry is loaded into memory. The original input file is also displayed in the left pane as reference.



<u>Diagnostics</u>

Despite efforts to make the translating routines as robust as possible, there may be occasions when the translation fails. In such cases, the "Diagnostic Output" window may provide some insight into why the routine was unhappy with the input file.

About the Variables

The variables available in the LinAir program are listed below. Variables are listed as follows: as described in the LinAir program, as an XML variable name used in the input file and as a short description of the variable. Where relevant, this description will be the variable name as typically depicted in aeronautical texts (please see the Theory section for definitions). Note that the reference values must be in consistent units. If s_{ref} is specified in square feet, b_{ref} should not be in inches, but any consistent set of units will do.

Global Properties

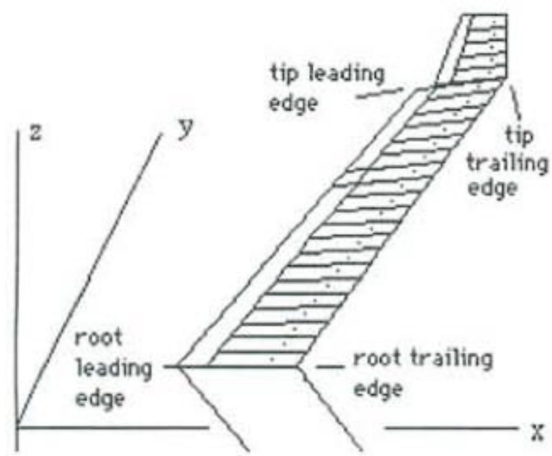
GUI	**XML**	**Standard Textbook Representation**
Ref. Span	bref	b_{ref}
Ref. Area	sref	S_{ref}
Ref. Chord	cref	c_{ref}
Ref. X	xref	X_{ref}
Ref. Y	yref	Y_{ref}
Ref. Z	zref	Z_{ref}
Mach	mach	M
Alpha	alpha	α
Beta	beta	β
p_hat	phat	p_{hat}
q_hat	qhat	q_{hat}
r_hat	rhat	r_{hat}
Reflect Geometry *	reflectgeometry (selected = 1 deselected = 0)	A checkbox which, when selected, indicates that the geometry should be reflected about the *x-z* plane, producing a symmetric geometry. Leave this unchecked for asymmetric geometries.

Planform Properties

Element Name	ElementName	Descriptive name for this element.
x_root_le	xrootle	*x*-coordinate of the element root leading edge.

x_root_te	xrootte	*x*-coordinate of the element root trailing edge.
y_root	yrootle	*y*-coordinate of the element root leading edge.
z_root	zrootle	*z*-coordinate of the element root leading edge.
x_tip_le	xtiple	*x*-coordinate of the element tip leading edge.
x_tip_te	xtipte	*x*-coordinate of the element tip trailing edge.
y_tip	ytiple	*y*-coordinate of the element tip leading edge.
z_tip	ztiple	*z*-coordinate of the element tip leading edge.
incidence_root	rooti	Angle of incidence of the element root in degrees.
incidence_tip	tipi	Angle of incidence of the element tip in degrees.
# Panels	npan	Number of panels per element used for the analysis. LinAir divides each element into evenly spaced spanwise panels with a size determined from this quantity.

In LinAir, the geometry is defined by specifying the element corner points. For each linearly tapered element, the following parameters must be specified: the *x*-coordinates of root leading edge, root trailing edge, tip leading edge, and tip trailing edge and the *y* and *z* coordinates of the root and tip. If you select the Reflect Geometry option (in the Flow Properties tab), you need only input the geometry of the positive y side of the airplane.



Note: it is sometimes difficult to decide which end of the element constitutes the root. For simple wings this is straightforward, but for vertical surfaces it may not be clear. LinAir distinguishes between the element root and tip only to determine the upper surface of the section so that the local section C_l will be assigned the correct sign.
The convention used here is that the upper surface is in the direction of the vector formed by taking the cross product of a vector in the x direction with a vector pointing from root to tip. Thus, the right side of a wing with the root at the centerline will have the upper surface on top; an upward pointing winglet with the root attached to the wing tip will have the "upper" surface facing inward; and a downward pointing winglet will have its upper surface toward the outside. This only affects the sign of the local C_l; it is therefore

important in only two situations: 1. when examining the C_l distribution and 2. when selecting the parabolic profile drag fit (see next page).

Section Properties

GUI	XML	Standard Textbook Representation
cdp0root	cdp0root	Element root C_{D0}
cdp1root	cdp1root	Element root C_{D1}
cdp2root	cdp2root	Element root C_{D2}
cm0root	cm0root	Element root C_{m_0}
clmaxroot	clmaxroot	Element root $C_{l_{\max}}$
cl0root	cl0root	Element root C_{l_0}
cdp0tip	cdp0tip	Element tip C_{D0}
cdp1tip	cdp1tip	Element tip C_{D1}
cdp2tip	cdp2tip	Element tip C_{D2}
cm0tip	cm0tip	Element tip C_{m_0}
clmaxtip	clmaxtip	Element tip $C_{l_{\max}}$
cl0tip	cl0tip	Element tip C_{l_0}

These section properties are used to make up, in part, for the assumption of inviscid flow. Although LinAir cannot compute the viscous drag of an airfoil section, data of this sort is often available to the user, and LinAir can incorporate such information into the calculation of forces and moments. The two-dimensional profile drag of the sections comprising each element may be represented with a parabolic fit: $C_d = C_{D0} + C_{D1} \bullet C_l + C_{D2} \bullet C_{l2}$. C_{m_0} is used to improve accuracy of the results using few elements but with highly cambered surfaces. The next parameter is the section maximum lift coefficient, Clmax. LinAir computes the local lift coefficient of the section and reduces the circulation if necessary so that the section C_l never exceeds this value. This is a very crude method of modeling the effects of separated flow but it sometimes illustrates a qualitative phenomena which would not be apparent without this treatment. If you do not wish to limit the section circulation, set $C_{l_{\max}}$ to a large, positive number. The final parameter is C_{l_0}, which can be used when modeling airfoils that produce lift at zero angle of attack.

Other Properties

The following variables are present only in the input file, and can be edited using a text editor or the "View XML Data" window in LinAir. Don't forget to click the "Apply Changes" button at the bottom of the screen after editing.

configurationname	Descriptive name for the configuration
wakelocation	Specifies the location of the wake leaving the trailing edge of each element (see discussion below).

Some Notes on the Wake Location:

A value of 0.0 indicates that the wake is to leave the trailing edge in the freestream direction. A value of 1.0 indicates that the wake is to remain along the x-axis, independent of angle of attack. Values in-between are also permitted. If the configuration is at an angle of sideslip the wake would actually be skewed to one side. This often causes numerical problems with elements downstream so LinAir assumes that the wake remains parallel to the *x*-*z* plane. It is sometimes very desirable to let the wake become asymmetric, however, and although one must check for possible numerical problems, the wake can be placed in the true freestream direction by setting wake position to a negative value.

In summary:

Wake Position Value	**Result**
1.0	Wake along *x*-axis direction. (safest)
0.0	Wake displaced from *x*-*y* plane but still parallel to *x*-*z* plane.
-1.0	Wake displaced from *x*-*z* plane but parallel to *x*-*y* plane.
-0.001	Wake lines essentially parallel to the freestream.

Sample Input File


```xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="linair.xsl" ?>
<linair>
<controls>
<var name="lastmodified"><val>Sep 29, 2003 2:07:55 PM</val></var>
<var name="lastupdated"><val>Sep 29, 2003 2:07:55 PM</val></var>
<var name="status"><val>complete</val></var>
<var name="comment"><val>LinAir Input File</val></var>
<class name="mainClass"><val>la4sub</val></class>
</controls>
<inputs>
<var name="configurationname"><val>LinAir Geometry</val></var>
<var name="sectionClass"><val>lasection</val></var>
<var name="sref"><val>130.0</val></var>
<var name="bref"><val>35.0</val></var>
<var name="cref"><val>3.7142857142857144</val></var>
<var name="xref"><val>3.2</val></var>
<var name="yref"><val>0.0</val></var>
<var name="zref"><val>0.0</val></var>
<var name="nelem"><val>2.0001</val></var>
<var name="alpha"><val>7.0</val></var>
<var name="beta"><val>0.0</val></var>
<var name="phat"><val>0.0</val></var>
<var name="qhat"><val>0.0</val></var>
<var name="rhat"><val>0.0</val></var>
<var name="mach"><val>0.1</val></var>
<var name="wakelocation"><val>1.0</val></var>
<var name="reflectgeometry"><val>1.0</val></var>
<var name="ElementName"><val>Wing</val><val>Winglet</val></var>
<var name="xrootle"><val>0.0</val><val>8.2</val></var>
<var name="yrootle"><val>0.0</val><val>17.5</val></var>
<var name="zrootle"><val>0.0</val><val>0.0</val></var>
<var name="xrootte"><val>4.0</val><val>11.0</val></var>
<var name="xtiple"><val>8.0</val><val>10.0</val></var>
<var name="ytiple"><val>17.5</val><val>17.5</val></var>
<var name="ztiple"><val>0.0</val><val>3.0</val></var>
<var name="xtipte"><val>11.0</val><val>12.0</val></var>
<var name="rooti"><val>10.0</val><val>0.0</val></var>
<var name="tipi"><val>0.0</val><val>0.0</val></var>
<var name="npan"><val>10.0</val><val>4.0</val></var>
<var name="cdp0root"><val>0.0070</val><val>0.0070</val></var>
<var name="cdp1root"><val>0.0</val><val>0.0</val></var>
<var name="cdp2root"><val>0.0030</val><val>0.0030</val></var>
<var name="cl0root"><val>0.0</val><val>0.0</val></var>
<var name="cm0root"><val>0.0</val><val>0.0</val></var>
<var name="clmaxroot"><val>1.5</val><val>1.5</val></var>
<var name="cdp0tip"><val>0.0070</val><val>0.0070</val></var>
<var name="cdp1tip"><val>0.0</val><val>0.0</val></var>
<var name="cdp2tip"><val>0.0030</val><val>0.0030</val></var>
<var name="cl0tip"><val>0.0</val><val>0.0</val></var>
<var name="cm0tip"><val>0.0</val><val>0.0</val></var>
<var name="clmaxtip"><val>1.5</val><val>1.5</val></var>
</inputs>
<results>
<var name="CL"><val>0.0</val></var>
<var name="CD"><val>0.0</val></var>
<var name="Cy"><val>0.0</val></var>
<var name="Cr"><val>0.0</val></var>
<var name="Cm"><val>0.0</val></var>
<var name="Cn"><val>0.0</val></var>
```

```
<var name="Cx"><val>0.0</val></var>
<var name="Cz"><val>0.0</val></var>
<var name="e"><val>0.0</val></var>
<var name="CL_element"><val>0.0</val><val>0.0</val></var>
<var name="CD_element"><val>0.0</val><val>0.0</val></var>
<var name="CY_element"><val>0.0</val><val>0.0</val></var>
<var name="Cr_element"><val>0.0</val><val>0.0</val></var>
<var name="Cm_element"><val>0.0</val><val>0.0</val></var>
<var name="Cn_element"><val>0.0</val><val>0.0</val></var>
</results>
</linair>
```

Using the Results - Some Suggestions

Lift Distributions

It is usually a good idea to examine the computed lift distributions by selecting the Lift Distribution option from the Display menu. Unusual discontinuities or negative lift values are indications of a possible error in the input geometry or a poor paneling layout. The lift distribution can also indicate a problem with the design: if the winglet local C_l is 3.0 it is a good bet that the winglet incidence is too high or the chord too small; separation would be hard to avoid.

Stability and Trim

It is often necessary to determine the stability of a lifting system or the position of the aerodynamic center. This can be done as follows: Compute the lift and moment coefficients at two angles of attack and approximate the derivative, dC_m/dC_L.

The negative of this derivative is the static margin, a measure of aircraft stability (negative values of dC_m/dC_L correspond to stable designs). Note, however, that the static margin is given in units of the reference chord, $c_{ref} = S_{ref} / b_{ref}$, not the mean aerodynamic chord which is often used as a reference dimension for static margin.

The aerodynamic center or neutral point is located a distance $c_{ref} \cdot (dC_m/dC_L)$ in front of the current reference center.

If the moment reference center (often coincident with the center of gravity) is not located in the plane of the wing, a nonlinear variation of C_m with C_L will appear. This means that the center of gravity position for neutral stability changes with α. It may be useful to plot C_m vs. C_L over the appropriate range of α in this case.

Using Approximate Linearity

The approximate linearity of lift and moment with angle of attack may be used to advantage in the design process. Rather than trying many angles of attack to achieve a desired lift coefficient, compute the lift curve slope and the lift at zero α and solve for the desired angle of attack. This technique is especially useful when you want to find the minimum drag at a fixed C_L while trimming the aircraft. A straightforward trial and error approach would require a great deal of time and patience, but using the fact that the change in lift and moment is approximately linear in both angle of attack and incidence angles of the root and tip sections, the problem is considerably simplified. Further simplification is possible by noticing that for a given shape of the twist distribution, the overall drag is given by the quadratic relation between angle of attack, twist amplitude, and C_D. This can be seen in the case of a single wing element in the simple relation:

$$C_D = A\alpha^2 + B\alpha\theta + C\theta^2$$

where:

A, B, and C are constants (a function of planform only), α is the angle of attack at the root, and θ is the wing twist angle.

In the more general case with multiple nonplanar surfaces a similar expression can be derived and used to simplify the design process (look for an optimizing, self-trimming version of LinAir in the near future).

Applicability

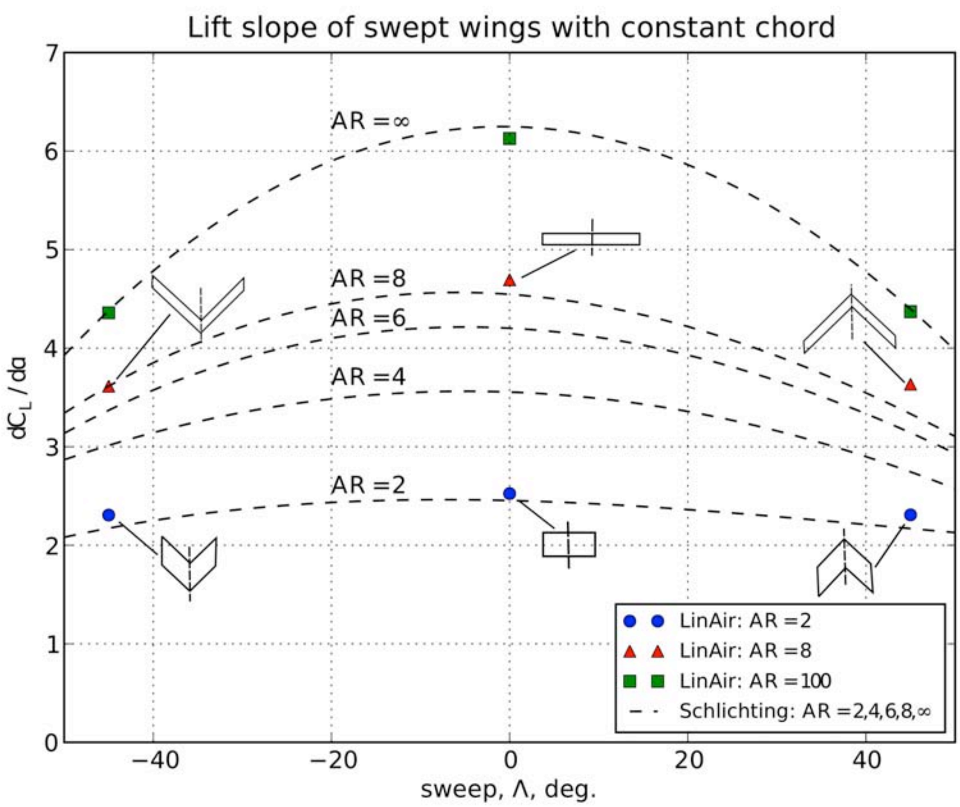
Remember that LinAir is solving the equation for irrotational, inviscid, linearized flow. It is quite willing to compute answers in situations where these assumptions are not likely to hold. High angles of attack, transonic or supersonic Mach numbers, or inappropriate panel layouts can and will give results which bear little resemblance to reality. Be sure to check the local C_l values to be sure that they are small enough to justify the assumption of attached flow (typically < 1.5 to 2.0) and do not vary wildly (indicative of poor paneling with interference between upstream vortices and control points).

Examples

Comparisons with Experiment and Theory

Lift curve slope

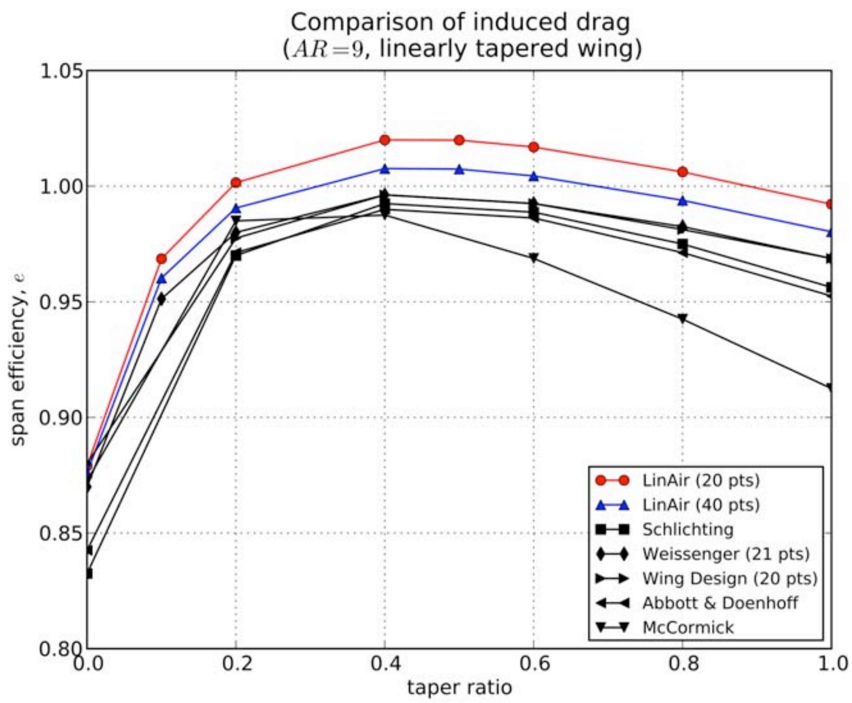
The figure below shows a comparison of LinAir 4 and another extended lifting line method (from reference 2) as a function of sweep and aspect ratio. Agreement is generally quite good; but insufficient detail is given to determine which of the methods is more accurate.



Lift curve slope (per radian) of swept-back wings of constant chord vs. sweepback angle and aspect ratio, *AR*. (Solid line - reference 2, x - LinAir)

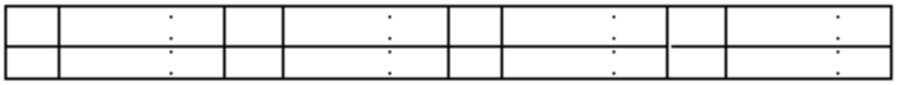
Induced drag

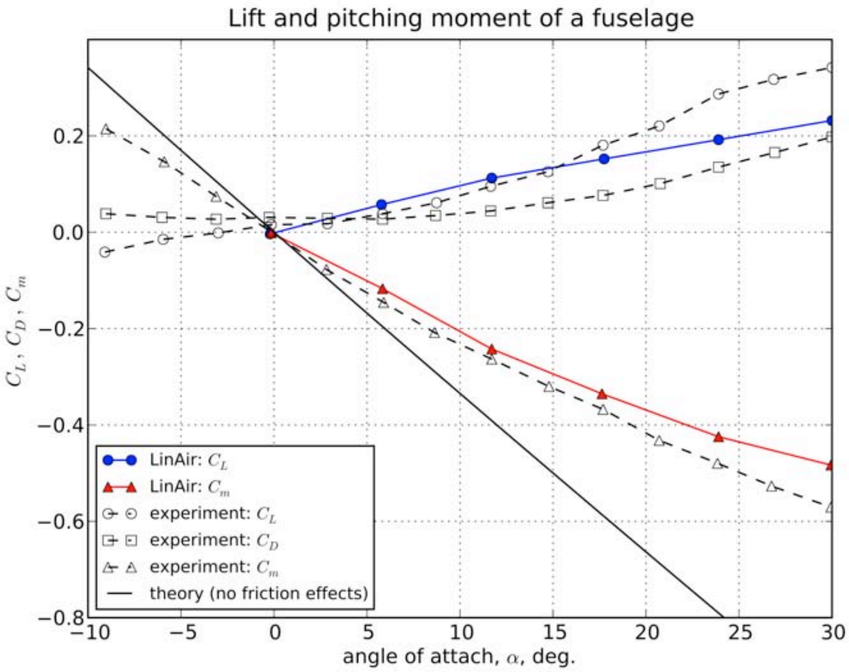
The span efficiency factor computed from LinAir provides a useful means for comparing the induced drag with other results. Note, however, that a 1% change in span efficiency corresponds to a very small change in the total aerodynamic force. (For an aspect ratio 9 wing at C_L = 1.0, 1% of the induced drag is a force of only 3/10,000 of the lift.) Nevertheless, it is usually possible to estimate the drag with nearly this accuracy. The figure below shows the effect of taper ratio on span efficiency for rectangular planform wings from several texts and analysis methods. Notice that there is substantial disagreement between the methods, but LinAir with a sufficient number of panels agrees well with most other results. It should be noted, however, that the method used by LinAir almost always slightly underpredicts induced drag. With 20 panels it is 2% to 3% optimistic while with 40 panels the result is about 1% low. A correction for this effect in many vortex lattice methods is obtained by indenting the last horseshoe vortex by 1/4 of the panel width. A variant of this idea works well in LinAir: reduce the span of the element by 1/4 of the panel width (or multiply by the factor 1 - 1/(4n)). Alternatively one may simply apply the correction factor, 1 - 1/(2n) to the span efficiency.

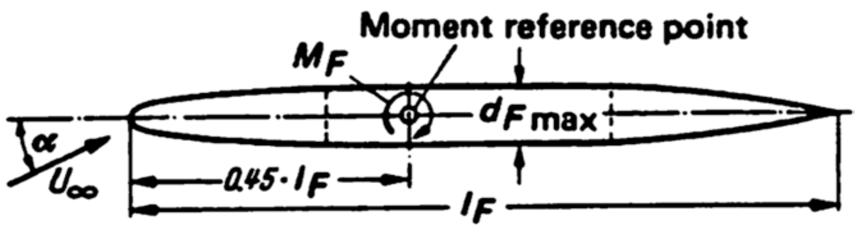


Fuselage Modeling

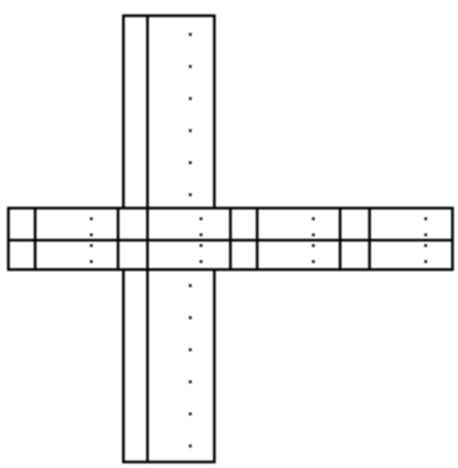
Although LinAir is intended primarily as a tool for modeling lifting surfaces, it may be used to approximate the effect of components such as fuselages but less accurately. The lift and pitching moment of a fuselage as predicted by LinAir is compared with experiment and a simple analytical method in the following figures. Despite the great simplicity of the LinAir model, the agreement is excellent. Note that lift, drag, and moment coefficients in this figure were made dimensionless with reference lengths that are functions fuselage volume and length, not simply the plan area.

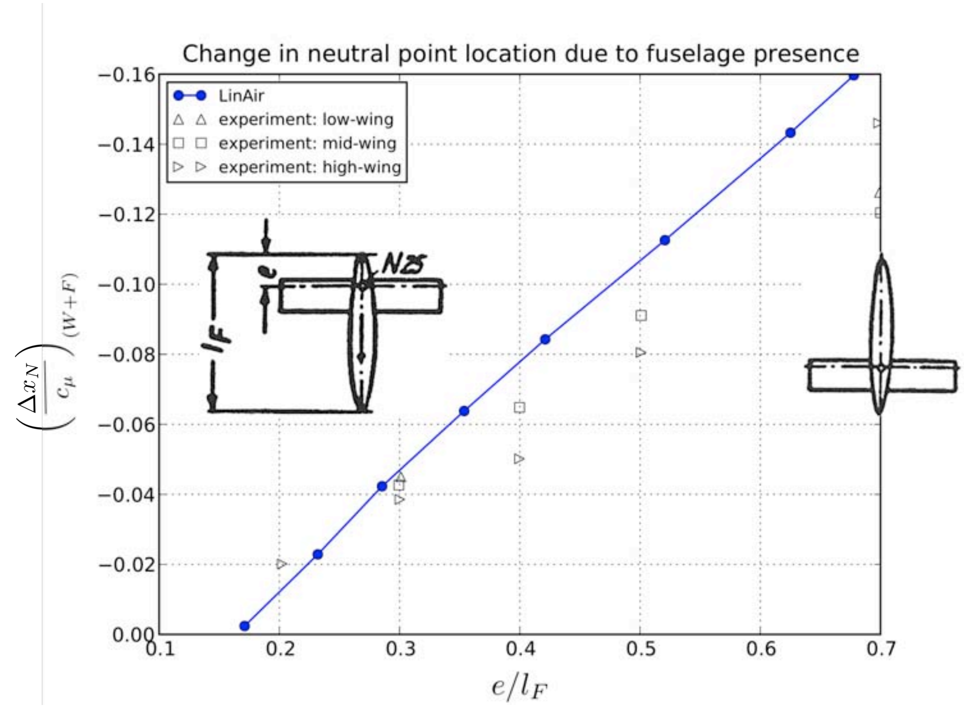




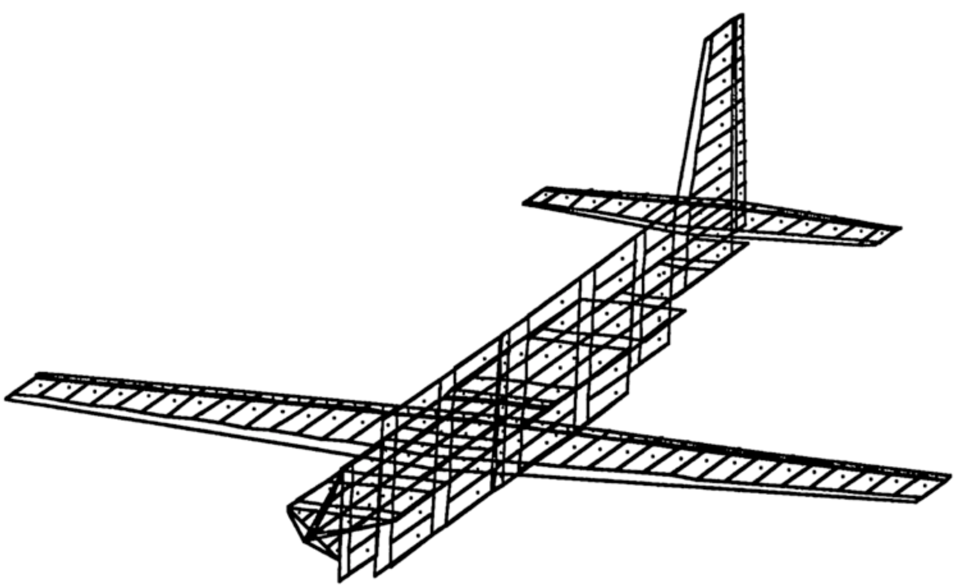


The second plot illustrates the change in neutral point position due to the presence of a fuselage for various wing locations. Again, the fuselage model is extremely simple, but the results, especially in the range of usual wing positions is quite good.



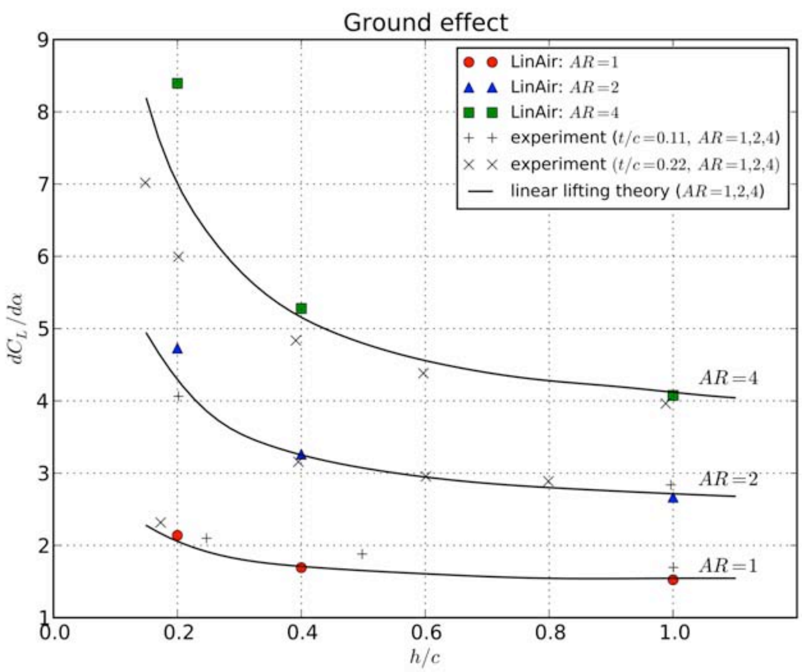


When sideslip is to be modeled, it is possible to represent the fuselage as a cruciform shape as shown in the model below; note the control surfaces on the wing and tails and the fuselage modeling.

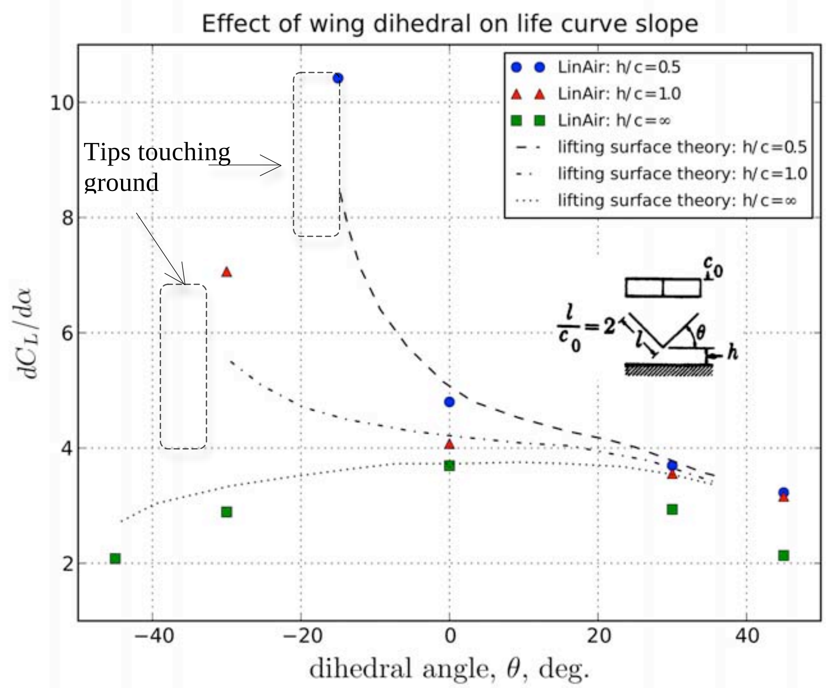


Ground Effect and Nonplanar Wings

The change in lift curve slope with height above the ground (in units of the reference chord) is shown in the figure below. LinAir results for the aspect ratio 4 wing agree quite well with experiments and with another lifting surface method (Ref. 5). Some of the discrepancy very near the ground may be associated with viscous effects. These results were obtained using a 20-panel LinAir model with an image wing (with the opposite incidence) "flying underground". Here the nonlinear effect associated with force computations based on near-field velocities (see theory section) is evident. The lift curve slope at higher angles of attack is significantly smaller than at low angles of attack. This is caused by the velocity induced by the bound vorticity of the image system in a direction opposite to the freestream that reduces the wing's lift.

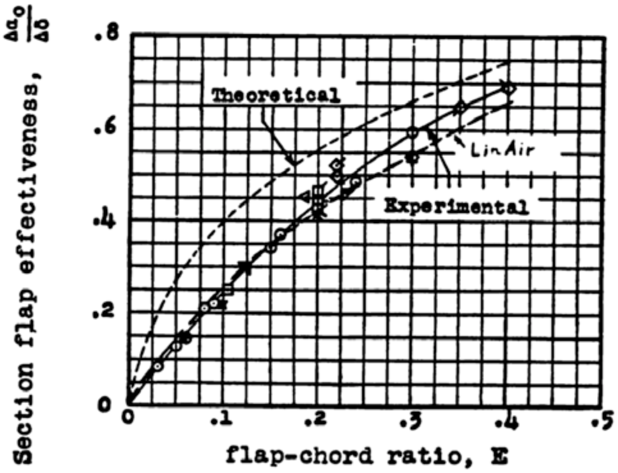


The second plot illustrates the effect of wing dihedral on lift curve slope. This version of LinAir predicts a larger effect than the lifting surface theory cited in reference 5 but it is not clear which is most accurate for this application.

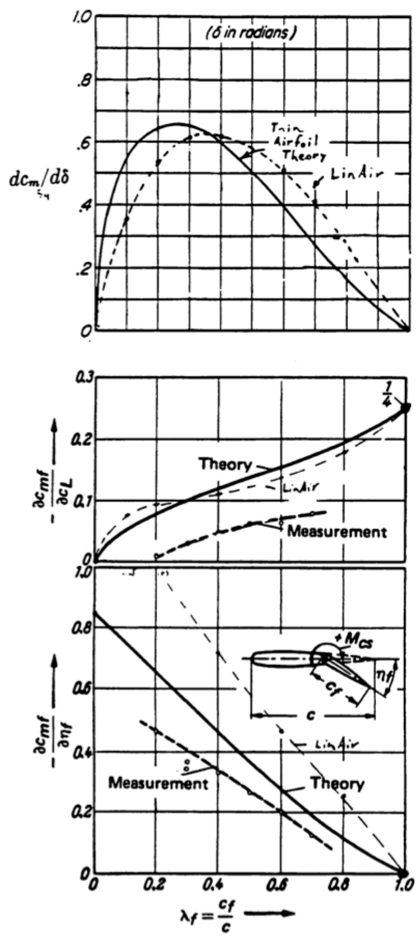


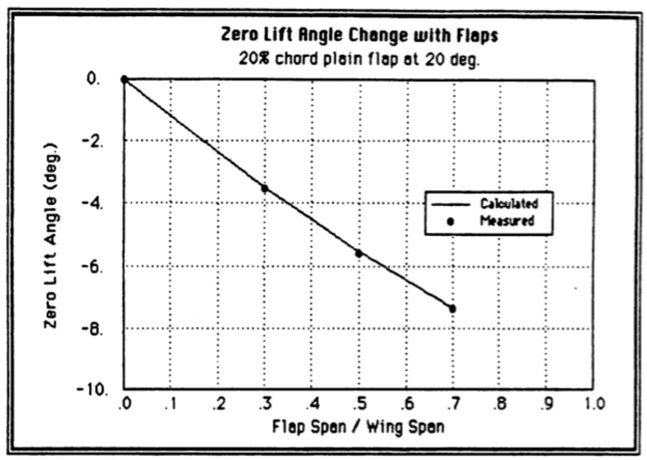
Modeling Control Surfaces

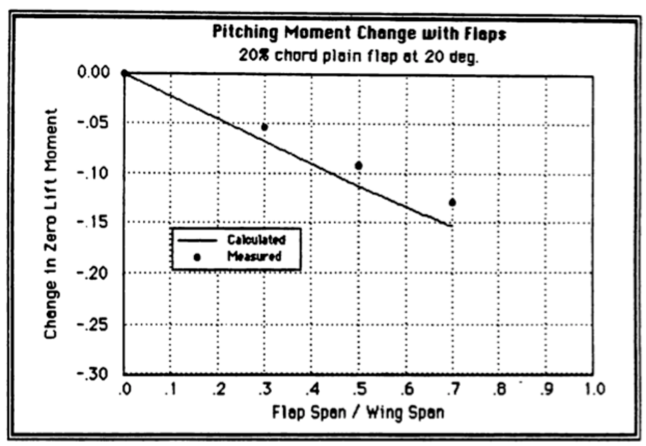
A wing with flaps or control surfaces may be modeled in two pieces: an element representing the fixed part of the wing, and a smaller element representing the control surface with its leading edge at the trailing edge of the first element. This seems like a rather crude representation of the true geometry, but again, it does surprisingly well in many cases. The figure below shows the computed and measured values of flap effectiveness for sections with various flap chord ratios. The theoretical curve shown is based on thin airfoil theory while the LinAir results were obtained on a wing with an aspect ratio of 100. The next figure (on the following page) shows how flaps with different flap chord ratios affect section pitching moment and moment about the flap leading edge. We would not expect LinAir to predict the latter quantity with any accuracy because of the simplicity of the model; yet the results for hinge moment due to angle of attack are not bad. The effect of flap deflection on hinge moment, however, are not as good.



The next figure shows the change in zero lift angle and moment for a wing with partial span flaps. The results agree very well with the experimental data, illustrating the utility of this technique when overall forces and moments must be computed.



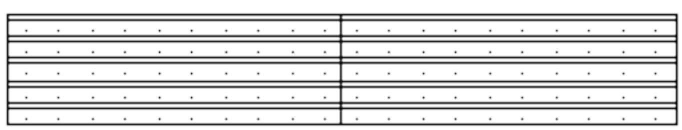




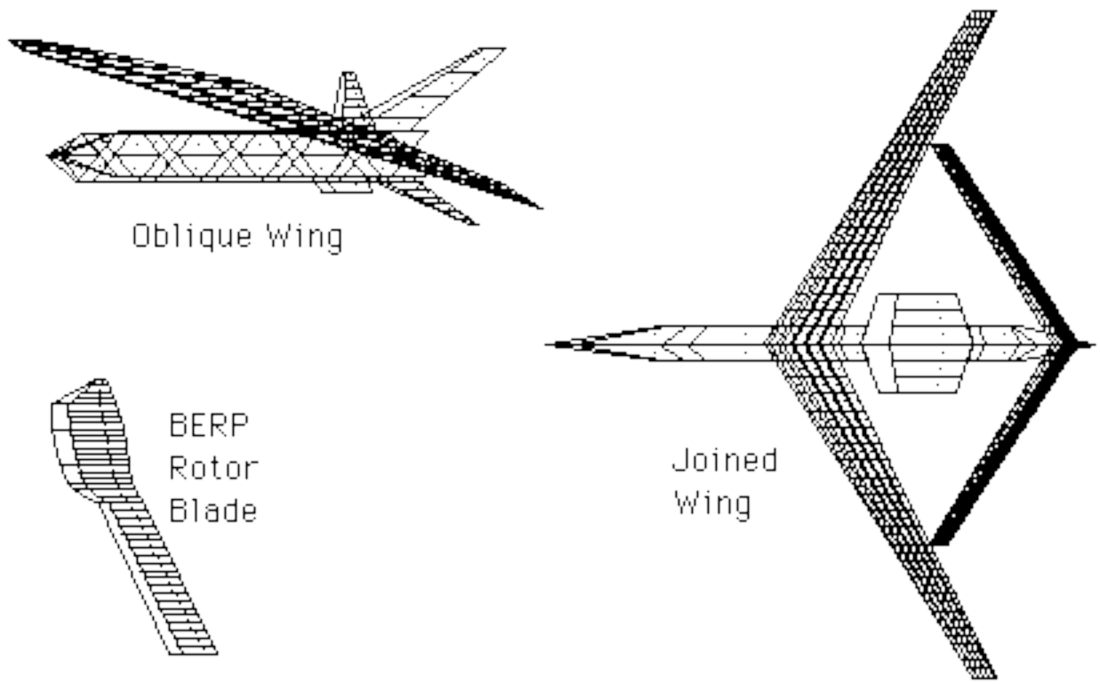
Modeling Airfoils with Camber

Usually a wing surface is represented with a single bound vortex on the 1/4 chord line. This means that the lift acts at the 1/4 chord and the pitching moment at zero lift is 0. The airfoil section's zero lift pitching moment must, therefore, be added (to the results by hand, using the appropriate nondimensionalization, or to the inputs using the cm0root and cm0tip variables). Similarly, the angle of attack and incidence angles may be taken as the airfoil's angle from zero lift rather than the geometric incidence (unless the input variables cl0root and cl0tip are used). So, for example, if the wing uses a 2412 airfoil, one would specify the incidence of the root and tip of the element as the actual incidence plus 2 degrees since the zero lift angle of the 2412 section is about -2 degrees. One would then subtract about 0.04 from the computed C_m to account for section moment about the quarter chord. In addition, one would include the actual section drag characteristics through the C_{D0}, C_{D1}, and C_{D2} terms.

Despite the simplicity of this approach, it is often all one needs to estimate lift, drag, and moment quite well. It is also possible to include camber effects explicitly by modeling the wing as two or more elements stacked leading edge to trailing edge. In fact, control surfaces may be modeled quite well using this method as shown in the previous section. In this case it is very important to align the panels and to avoid spanwise discontinuities in the bound vortex positions. The model below shows how 5 elements may be aligned to simulate a single cambered wing. Each element is set at an angle of incidence corresponding to the local airfoil slope. This modeling approach provides reasonable lift and moment results directly, but slows down the computation, and makes interpretation of the results more difficult since the spanwise lift distribution must be computed by summing the lift on each chordwise panel at a given spanwise location.



Analysis of Complete Configurations



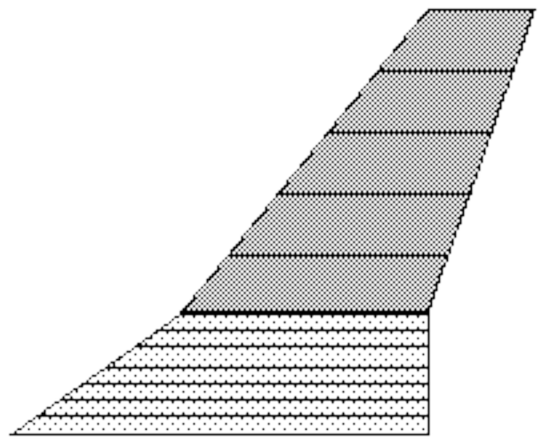
LinAir has been applied to the analysis of many aircraft designs from model aircraft to manned research aircraft to VSTOL fighters. The figures above, from reference 6, illustrate the application of LinAir to an unconventional aircraft design which was later tested in a low speed wind tunnel at NASA's Ames Research Center.

Hints and Limitations

Paneling Style and Convergence

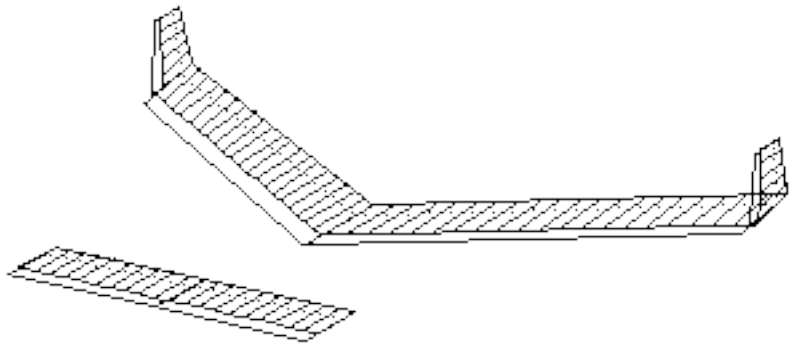
Proper modeling of the geometry is of critical importance to obtaining accurate answers from LinAir, especially when the configuration is complex. Here are a few of the do's and don'ts of paneling.

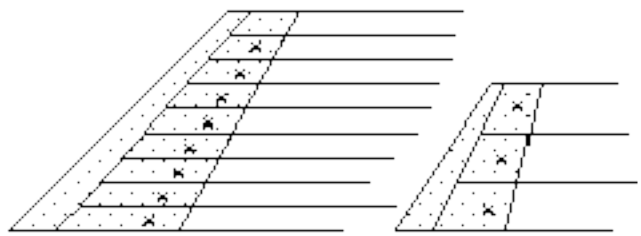
Try to distribute the panels as uniformly as possible. When one element (a wing, say) has a semi-span of 10 feet with 10 panels and a second element (e.g. an attached winglet) has a semi-span of 3 feet, it is best not to put 10 or 20 panels on the second element since the panel density changes dramatically. The answers will usually be quite close even in this case but it is a good idea to check results by using a different number of panels.



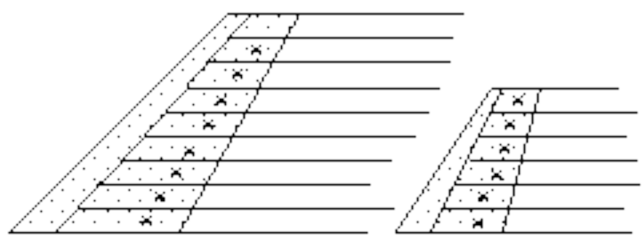
Uneven paneling is to be avoided.

Panel spacing is critical when the wake of one element passes very close to another, downstream, element. It is a good idea to line up panels in this case. For example, if a wing with semi-span 10 lies downstream of a canard of semi-span 6, it would be best to place 12 panels on the canard and 20 on the wing. Similarly, the method works best on surfaces without sharp discontinuities in planform, although here again the method often works better than it probably should.



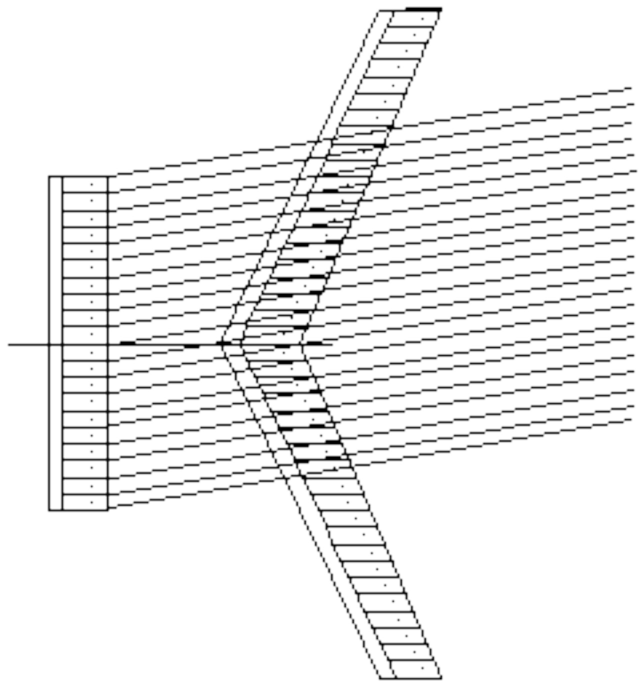


Wrong. Upstream wake crosses control point.



Right. Upstream wake aligned with downstream panels.

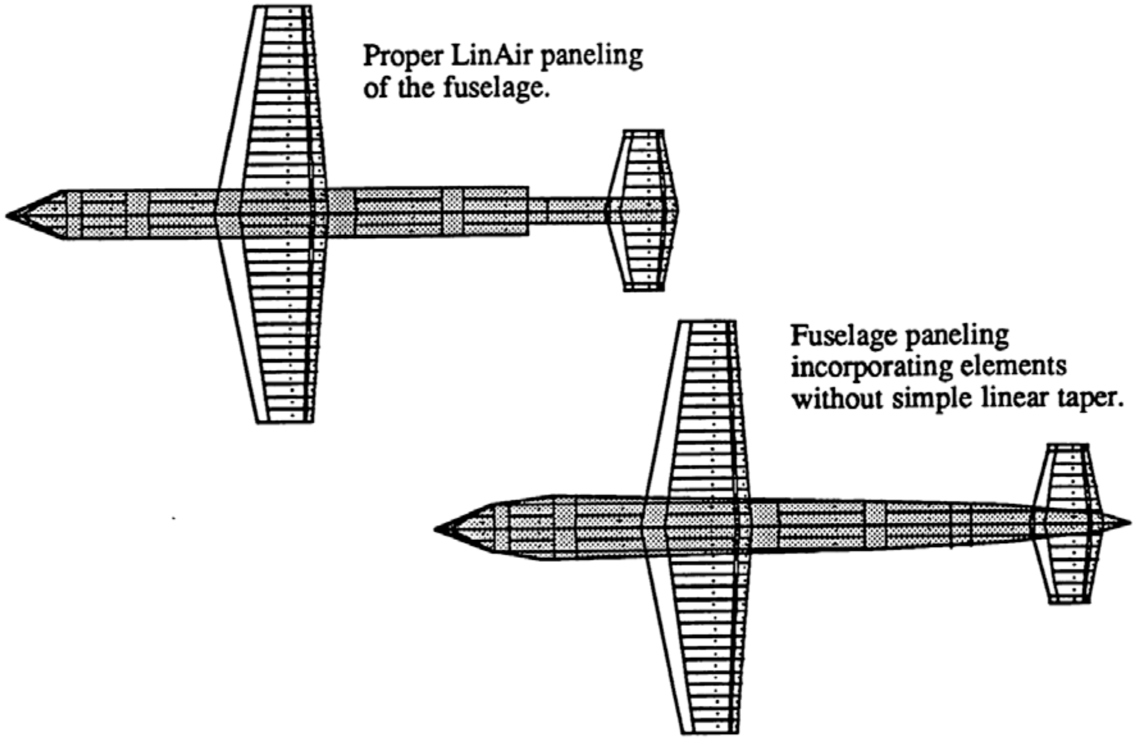
Interference between trailing vortices and downstream control points is very likely when several elements are placed on the chord and some sideslip is specified and the wake is allowed to move (wake position 1).



Canard wake lines may cross control points on the wing when the configuration is at an angle of sideslip and the wake is allowed to follow the free-stream. Solution: set wake position parameter to 1.0. This keeps the wake parallel to the *x* axis. (If there is sufficient vertical gap between the wing and canard this may not be a problem - always examine the lift distribution.)

Try changing the number of panels to be sure that you have represented the surfaces with sufficient resolution. If you are looking for accurate drag results you may have to include quite a few panels; the drag is often under-predicted when the panel density is too small. See the examples in the start of this section for details.

It is important to remember that lifting elements in LinAir are linearly tapered. This means that the chord varies linearly from the root to the tip of the element. LinAir 4 no longer allows arbitrary element corner points, but forces the leading and trailing edge *y*- and *z*-coordinate to be the same value. This may seem limiting, but it is more consistent with the aerodynamic model. For instance, the fuselage model shown on the right hand side of the next figure may appear more realistic, but the version with streamwise edges shown on the left is a better LinAir representation.



Other Versions of LinAir

LinAir Pro, which includes a built-in, user-configurable optimizer, computation of stability derivatives, a dynamic stability package and facilities for quickly and easily performing parametric variations is also available.

Custom versions of LinAir have also been developed with features such as unsteady aerodynamics, wake roll-up calculations, aeroelastic effects, and thickness effects. We would be happy to discuss your requirements and develop a program that is tailored for your application.

Batch Mode

You may run LinAir from the Terminal to call the batch mode. This is done by running LinAir, then navigating to your saved XML input file and entering

```
java CaffeApp inputfilename.xml
```

(For this to work, you must enter the variables you would like to see in the results section of the xml input file.)

This will then produce in the Terminal the following debug script:

```
Reading control data...
Reading inputs...
Computing...
linkedvar not found by getSize.
Updating the file: inputfilename.xml
Writing to scratch file.
```

Disregard this and open the input file. (In OSX Terminal "cat inputfilename.xml") The results will have overwritten the prior values for any variables that were entered into the "results" section of the input file – the portion that appears between the <results> and </results> tags. Note: If you do not name any variables in the results section of the input file, no results will be published.

Theory

Basic Definitions

Terminology

The following nomenclature is used throughout this manual and in the LinAir program:

C_L	Lift coefficient = Lift $/ (q\ S_{ref})$
C_D	Drag coefficient = Drag $/ (q\ S_{ref})$
C_X	Force coefficient in direction of x axis
C_Y	Sideforce coefficient = Sideforce (in y direction) / ($q\ S_{ref}$)
C_Z	Force coefficient in direction of x axis
C_m	Moment coefficient = Moment about reference center / ($q\ S_{ref}\ c_{ref}$)
C_r	Rolling moment coefficient = Rolling moment / ($q\ S_{ref}\ b_{ref}$)
C_n	Yawing moment coefficient = Yawing moment / ($q\ S_{ref}\ b_{ref}$)
$C_{l_{\max}}$	Section maximum local lift coefficient. LinAir computes the local lift coefficient of the section and reduces the circulation if necessary so that the section C_l never exceeds this value. This is a very crude method of modeling the effects of separated flow but it sometimes illustrates a qualitative phenomena which would not be apparent without this treatment. If you do not wish to limit the section circulation, set $C_{l_{\max}}$ to a large, positive number.
q	Dynamic pressure = $0.5 * \rho * U^2$
ρ	Air density
U	Freestream velocity
S_{ref}	Reference area specified by the user. Usually taken to be the wing planform area, S_{ref} is used to compute the dimensionless coefficients C_L, C_D, and C_m.
b_{ref}	Reference span specified by the user. Often taken to be the wing projected span, bref is used in several parts of the calculation. These include: normalization of roll and yaw moment coefficients, calculation of span efficiency factor, and (combined with S_{ref}) the reference chord. LinAir 4 allows the user to specify C_{ref}. LinAir 3 fixed $C_{ref} = S_{ref} / b_{ref}$. Users now have the flexibility to specify whatever value they feel is appropriate. It is common practice to use the mean aerodynamic chord of the wing for this purpose and if this is desired, the results must be appropriately scaled. There is usually no need to express results in terms of MAC, however.

c_{ref} — Reference chord often defined as S_{ref} / b_{ref} . * LinAir 4 allows the user to specify C_{ref}. LinAir 3 fixed C_{ref} = S_{ref} / b_{ref}. Users now have the flexibility to specify whatever value they feel is appropriate.

X_{ref}, Y_{ref}, Z_{ref} — *x*-, *y*-, *z*-coordinates of the moment reference center. LinAir uses a body axis system with the *X* axis pointing downstream when the configuration is at zero angle of attack, the *Y* axis pointing to the right when facing upstream, and the *Z* axis pointing up. (See figure in section 2.) The moment reference is often taken to be the center of gravity position although this need not always be the case.

C_{D0} ** — Profile drag fit constant term

C_{D1} ** — Profile drag fit linear term

C_{D2} ** — Profile drag fit quadratic term

C_{m_0} — Section pitching moment at zero lift. This is to improve the accuracy of results using few elements but with highly cambered surfaces.

C_{l_0} — Section lift at zero angle of attack (α)

e — $(C_L^2 / \pi AR) / C_D$ When viscous drag is 0, *e* is the span efficiency.

C_l — Section lift coefficient (nondimensionalized by local chord and *q*)

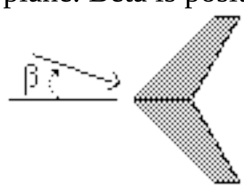
C_d — Section drag coefficient (nondimensionalized by local chord and *q*)

M — Freestream Mach number = *U* / sound speed (Must be < 1.0 in LinAir.)

U, V, W — Velocity components in the *x*-, *y*-, and *z*-directions

α — Angle of attack in degrees measured between the freestream and the *x*-*y* plane. This is the angle that the flow makes with the *x*-*y* plane.

β — Angle of sideslip in degrees measured between the freestream and the *x*-*z* plane. Beta is positive when the flow is coming from the right.



The entire configuration may be rotated in order to compute damping terms. Usually the rotation rates will be set to zero but they are useful for computing aircraft stability derivatives. Specifying rotations changes the boundary conditions but does not change the wake shape. Thus a rapidly rolling wing will indeed show an induced thrust, but because the wake remains flat, not helical, the slipstream associated with this "propeller" will not be predicted. For the purposes of computing stability derivatives, the dimensionless rates should be set to small values (~ .01).

p_{hat} — Dimensionless roll rate (about x-axis) = roll rate · $b / (2 U)$.
Positive for a roll to the right, in radians per second, U is the freestream velocity in ft/sec, and b_{ref} is the reference span in feet. Note that other consistent units (radians/hr, km/hr, and km for example) will give the same value for phat. The configuration rolls about the *x*-axis.

q_{hat}	Dimensionless pitch rate (about *y*-axis) = pitch rate $\cdot\ c_{ref} / (2\ U)$. Positive pitch rate is nose up.
r_{hat}	Dimensionless yaw rate (about *z*-axis) = yaw rate $\cdot\ b / (2\ U)$. Positive yaw rate is nose right.
U_x	Partial derivative of *x* component of local flow velocity with x
V_y	Partial derivative of *y* component of velocity with respect to *y*
W_z	Partial derivative of *z* component of velocity with respect to *z*
Γ	The vector circulation or vortex strength at a section on the wing
U_n	Component of freestream + rotational velocities normal to panel
F	Total resultant force vector
C_{n_p}	Yawing moment per unit dimensionless roll rate

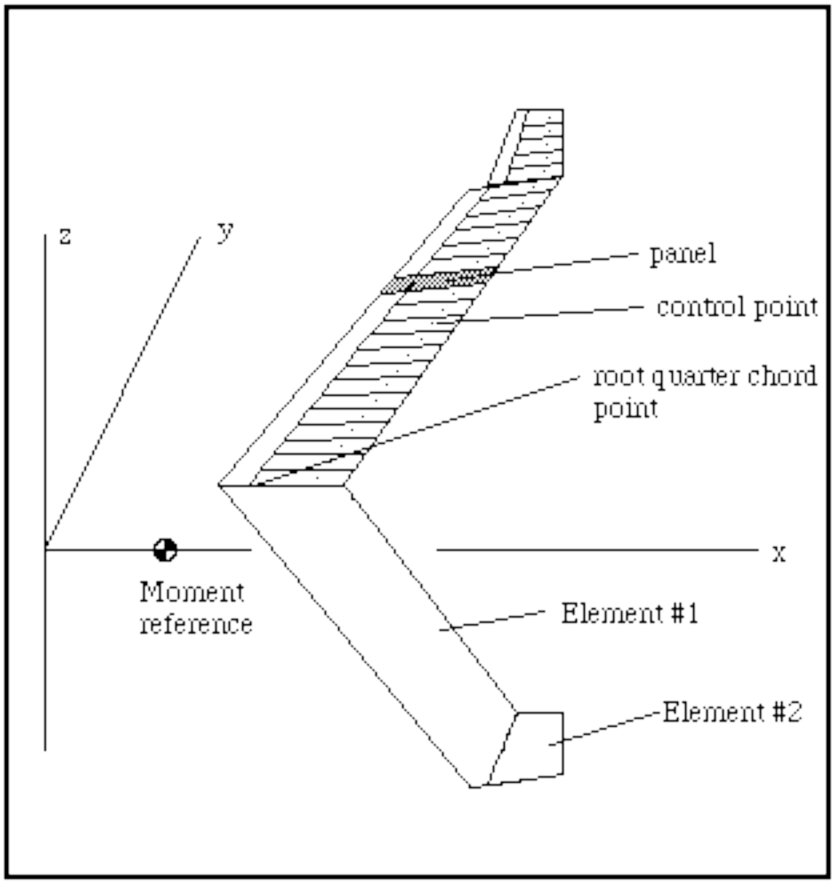
* Note that the reference chord used here is the mean geometric chord rather than the more common mean aerodynamic chord of the wing. In this program, the "wing" may consist of several elements or may not exist at all. For this reason, the reference chord is taken to be the mean geometric chord of the reference planform. This is a perfectly acceptable definition but must be remembered when computing static margin and other quantities often nondimensionalized by MAC.

** Although LinAir cannot compute the viscous drag of an airfoil section, data of this sort is often available to the user, and LinAir can incorporate such information into the calculation of forces and moments. The two dimensional profile drag of the sections comprising each element may be represented with a parabolic fit:

$C_d = C_{D0} + C_{D1} \cdot C_l + C_{D2} \cdot C_{l2}$.

Units and Coordinates

The coordinate system is defined as follows: At zero angle of attack the *x*-axis points downstream, the *z*-axis points up (vertically), and the *y*-axis points to the right when facing upstream. The coordinate system is a "body axis" system. The coordinates of the elements thus do not change with angle of attack. The rotation rates and moments are measured in the following senses. Pitch: positive for right-handed rotation about y axis (nose up). Roll: positive for left-handed rotation about *x*-axis (right wing down). Yaw: positive for left-handed rotation about *z*-axis (nose right). These are the usual conventions for studies of aircraft dynamics. The directions of the *x*- and *z*-axes are conventional wind tunnel nomenclature.



Areas, lengths, and coordinates may be specified in any consistent set of units. If you prefer area in square meters, also enter the span in meters and other coordinates in meters. Feet and square feet, inches and square inches, or fathoms and square fathoms are likewise all acceptable. Angular quantities are not dimensionless, however. All angles (i.e. angle of attack and sideslip) must be specified in degrees. The dimensionless rates are formed by taking the rotation rates in radians per unit time and multiplying by the appropriate factor involving speed and reference length.

Theory Summary

LinAir solves the Prandtl–Glauert equation, a linear partial differential equation describing inviscid, irrotational, subsonic flow:

(1) $(1 - M^2)\, U_x + V_y + W_z = 0$

with U, V, and W the three components of the flow velocity in the x, y, and z directions respectively and M, the freestream Mach number.

This linear equation is solved by superposition of known solutions, discrete line vortices. LinAir represents the wing surfaces with discrete vortex lines forming skewed horseshoe vortices. The vortex strengths are adjusted so that the flow is tangent to the surfaces at a series of control points. These points are located on the 3/4 chord line of each element, at the (lateral) center of each panel. The “bound vortex” is located at the 1/4 chord line. The program computes the solution to a linear system of equations:

(2) $[AIC]\{\Gamma_I\} = \{U_{n_i}\}$

where:

$[AIC]$ is a matrix of aerodynamic influence coefficients – the effect of panel i on panel j,

$\{\Gamma_i\}$ is an array of circulation strengths,

$\{U_{n_i}\}$ is an array containing the component of freestream velocity normal to the panel.

Once this system of equations is solved (using the method of LU decomposition and Gaussian elimination) the force and moment contribution of each panel is computed from the Kutta–Joukowski relation:

(3) $F = \rho V \times \Gamma$.

The method is known as “extended lifting line theory” or a discrete vortex Weissenger method (reference 2, 5). Although this represents the basic theory behind the computations, many subtleties appear in its implementation and some extensions to the basic method have been added. The following paragraphs describe some of the details of the method used in LinAir 4.

The geometry actually consists not of simple skewed horseshoe vortices, but of vortex lines extending from root to tip and trailing vortex lines that follow the surface from the bound vortex to the trailing edge. After leaving the trailing edge these vortices extend downstream in a direction specified by a parameter in the input file.

The *AIC*'s represent the component of velocity in the direction of the normal vector to the i^{th} panel produced by unit vortex strength on the j^{th} panel. These velocities are computed using the Biot–Savart law, modified to account for the $(1 - M^2)$ term in the differential equation (1). That is, the Prandtl–Glauert similarity law is applied directly during the computation of all induced velocities.

$\{U_{n_i}\}$ in equation (2) represents both the component of the freestream normal to the panel at the control point and the component associated with the rotations *p*, *q*, and *r*.

To compute the forces and moments acting on the configuration, the Kutta–Joukowski relation is applied, but the velocities needed in that expression are computed at the center of the bound vortex, not at the control point. This yields the correct value of the induced drag, but leads to several additional complications. Use of the local velocities permits modeling of several characteristics not obtainable from simpler codes. These include: the correct sign for derivatives such as C_{n_p}, the increased lift on the upper wing of an unstaggered biplane, and more accurate representations of the distribution of induced drag.

Care has been taken to avoid the numerical difficulties associated with the use of near-field force calculations while retaining important effects. Few vortex lattice codes, for example, can predict the sideforce generated by a yawed wing with the accuracy of LinAir.

Viscous drag is added to the forces and moments by integration over each of the elements. While lift-dependent profile drag is beyond the scope of linear theory, inclusion of these effects, through the constants C_{D1} and C_{D2}, produces a more realistic prediction of the total vehicle characteristics.

LinAir Pro 4 – Manual Supplement

Features of the Pro Version

In addition to those items previously discussed, these additional components are provided in the Pro version:

• Stability Derivatives: Computes stability derivatives and returns results in tabular form.

• Variations: Given two parameters and ranges for each, performs a parametric analysis of one parameter against the other. Results are provided in tabular and plot form (this supersedes LinAir 3's batch mode).

• Optimizer: A user-configurable optimizer that minimizes an objective function given a series of design variables and constraints.

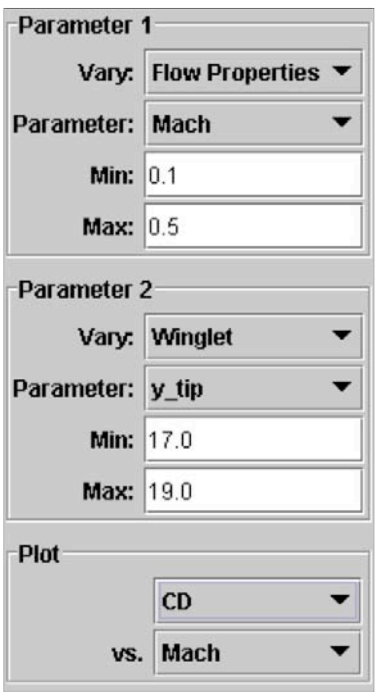
• Dynamics: A linear dynamics package that computes frequencies and damping ratios of the short period, phugoid, dutch roll, spiral and roll modes.

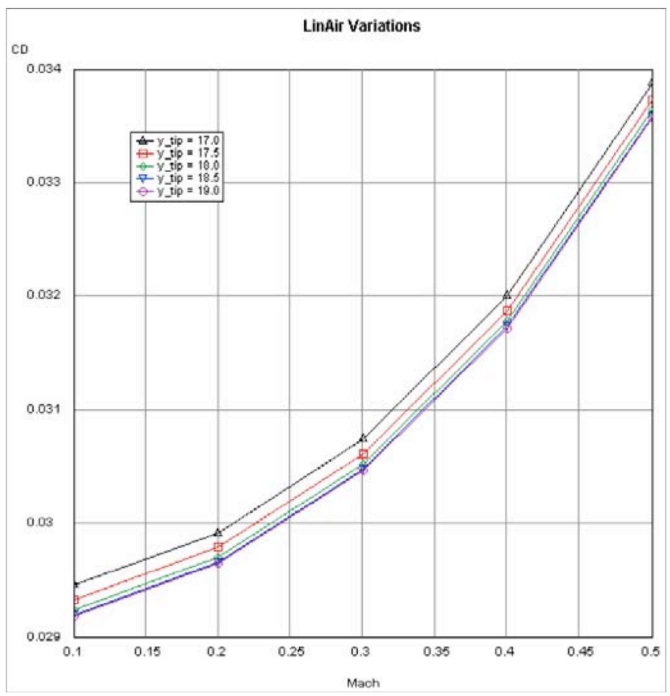
Quick Start

This section serves to quickly introduce the applets specific to the Pro Version of LinAir. Please refer to the User's Guide section for more detailed coverage of these components. The Pro specific components are accessed through the Display menu, just like the base LinAir components:

• Stability Derivatives:
As visually and interactively austere as the Element Forces applet, but serving its purpose nonetheless, clicking "Compute" here will produce a table of the stability derivatives for the current geometry.

• Variations:
This applet allows the user to perform parametric variations of essentially any one parameter versus any other parameter. If, for instance, the user desired a study of the effect of winglet cant angle on total drag (C_D) over a range of Mach numbers, the applet would be configured as follows:





Then clicking "Compute" and selecting the Plot tab would reveal (something like) the plot shown above.

• Optimizer: This applet provides an interface to the LinAir optimizer. As the setup and operation of this applet is somewhat more involved, it will not be covered here. Please see the User's Guide for details.

• Dynamics: This applet provides an interface to the linear dynamics package. The mass properties of the vehicle need to be input. Clicking the “Compute” button will result in a plot of the short period, phugoid, dutch roll, spiral divergence and roll modes of the vehicle. Tabular data is also available.

User’s Guide

Usage of those applets specific to the Pro version is covered here.

Stability Derivatives

This applet provides a table of the stability derivatives for the geometry at the conditions specified in the “Inputs” section.

Table | Table (Tab-delimited)

Parameters	CL	CD	Cx	Cz	Cy	Cm	Cn	Croll
Alpha	0.082609...	0.004474...	-0.02361...	0.080979...	9.566926...	-0.02094...	-2.52068...	-5.80516...
Mach	0.076218...	0.004179...	-0.00515...	0.076157...	-4.50191...	-0.01544...	-8.51454...	-8.02493...
Beta	-0.00157...	9.065177...	-1.19922...	-0.00160...	-0.00418...	-2.92094...	3.357823...	-0.00407...
p_hat	0.059301...	-0.01334...	-0.02256...	0.056976...	-0.03971...	-0.05121...	-0.18255...	-0.71167...
q_hat	6.181357...	-0.01300...	-0.76511...	6.133834...	0.021771...	-4.38919...	-0.03806...	-0.06141...
r_hat	-0.01963...	-0.00455...	0.005937...	-0.01905...	0.155183...	0.026500...	-0.01254...	0.243505...

Stability Derivatives Properties & Variables

The following new variables are available to the Pro version. Variables are listed as follows: as described in the LinAir program, as an XML variable name used in the input file and as a short description of the variable.

LinAir GUI	XML	Standard Textbook Representation
CL Alpha	CL_alpha	$dC_L/d\alpha$
CD Alpha	CD_alpha	$dC_D/d\alpha$
Cx Alpha	Cx_alpha	$dC_x/d\alpha$
Cz Alpha	Cz_alpha	$dC_z/d\alpha$
Cy Alpha	Cy_alpha	$dC_y/d\alpha$
Cm Alpha	Cm_alpha	$dC_m/d\alpha$
Cn Alpha	Cn_alpha	$dC_n/d\alpha$
Croll Alpha	Cr_alpha	$dC_r/d\alpha$
CL Mach	CL_mach	dC_L/dM
CD Mach	CD_mach	dC_D/dM
Cx Mach	Cx_mach	dC_x/dM
Cz Mach	Cz_mach	dC_z/dM

Cy Mach	Cy_mach	dC_y/dM
Cm Mach	Cm_mach	dC_m/dM
Cn Mach	Cn_mach	dC_n/dM
Croll Mach	Cr_mach	dC_r/dM
CL Beta	CL_beta	$dC_L/d\beta$
CD Beta	CD_beta	$dC_D/d\beta$
Cx Beta	Cx_beta	$dC_x/d\beta$
Cz Beta	Cz_beta	$dC_z/d\beta$
Cy Beta	Cy_beta	$dC_y/d\beta$
Cm Beta	Cm_beta	$dC_m/d\beta$
Cn Beta	Cn_beta	$dC_n/d\beta$
Croll Beta	Cr_beta	$dC_r/d\beta$
CL p_hat	CL_phat	$dC_L/d\hat{p}$
CD p_hat	CD_phat	$dC_D/d\hat{p}$
Cx p_hat	Cx_phat	$dC_x/d\hat{p}$
Cz p_hat	Cz_phat	$dC_z/d\hat{p}$
Cy p_hat	Cy_phat	$dC_y/d\hat{p}$
Cm p_hat	Cm_phat	$dC_m/d\hat{p}$
Cn p_hat	Cn_phat	$dC_n/d\hat{p}$
Croll p_hat	Cr_phat	$dC_r/d\hat{p}$
CL q_hat	CL_qhat	$dC_L/d\hat{q}$
CD q_hat	CD_qhat	$dC_D/d\hat{q}$
Cx q_hat	Cx_qhat	$dC_x/d\hat{q}$
Cz q_hat	Cz_qhat	$dC_z/d\hat{q}$
Cy q_hat	Cy_qhat	$dC_y/d\hat{q}$
Cm q_hat	Cm_qhat	$dC_m/d\hat{q}$
Cn q_hat	Cn_qhat	$dC_n/d\hat{q}$
Croll q_hat	Cr_qhat	$dC_r/d\hat{q}$
CL r_hat	CL_rhat	$dC_L/d\hat{r}$

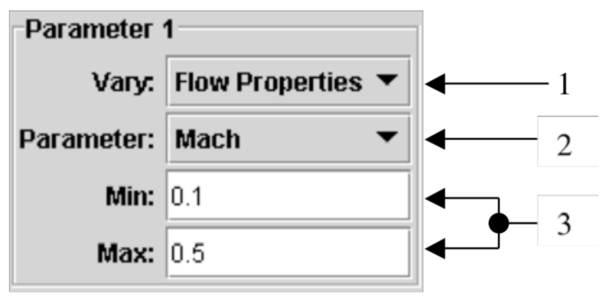
CD r_hat	CD_rhat	$dC_D/d\hat{r}$
Cx r_hat	Cx_rhat	$dC_x/d\hat{r}$
Cz r_hat	Cz_rhat	$dC_z/d\hat{r}$
Cy r_hat	Cy_rhat	$dC_y/d\hat{r}$
Cm r_hat	Cm_rhat	$dC_m/d\hat{r}$
Cn r_hat	Cn_rhat	$dC_n/d\hat{r}$
Croll r_hat	Cr_rhat	$dC_r/d\hat{r}$

Variations

The Variations applet allows the user to perform parametric variations using virtually any two LinAir input parameters. Computed data can be presented in either tabular or plot format.

Specifying a Parameter

1.) The parameters are grouped into a "Flow Properties" group, along with groups of planform and section properties for each element making up the geometry. These latter groups are identified by their respective element names. These groups are selected through the "Vary" pull-down menu.



2.) Once a group has been selected, individual parameters can be chosen from the "Parameter" pull-down menu.

3.) Finally, having located and specified the desired parameter, upper and lower bounds should be provided for the parametric analysis. These should be entered into the "Min." and "Max." fields.

For example, to specify the wing reference area (s_{ref}, a flow property) as a parameter, the user would select the "Flow Properties" group from the "Vary" pull-down menu, and then select the flow property "s_{ref}" from the "Parameter" pull-down menu.

Minimum and maximum values would be specified in the "Min." and "Max." fields as necessary.

The second parameter is specified in exactly the same fashion.

Plotting

Plots are configured through the “Plot” box. The y-axis parameter (ordinate) is the first item, whilst the x-axis parameter (abscissa) is the second.

Each curve is of constant parameter 2, and varying parameter 1; i.e. if parameter 1 was alpha and parameter 2 Mach number, each curve would have a constant Mach number and varying alpha.

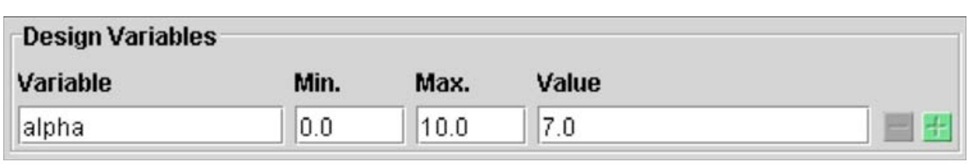
Optimization

This applet provides an interface to the LinAir optimizer. This system attempts to minimize a given objective by varying the given design variables over the specified range, optionally subject to a number of constraints. The problem space is currently searched using a simplex search. This is a non-gradient based scheme that is fairly robust and reasonably efficient, but starts to have problems within search spaces of greater than 10-14 dimensions. Please limit problem dimensionality accordingly.

Notes on the Interface
Some new interface elements are introduced in the Optimizer applet to facilitate the addition and removal of constraints and variables. The “plus” () button adds a new variable slot to the end of the current list, whilst the “minus” () button removes the variable at the current slot from the analysis. If the “minus” button is grayed out () this indicates that the action is disabled for this case.

Running the Optimizer
At a minimum, the optimizer should be provided with one design variable, an objective, and the number of design iterations that the optimizer should perform. Optionally, constraints may also be imposed to ensure that the optimal point does not violate certain results. To prepare for a run:

1. Specify the design variables. Since the interface will not allow any less than one variable, at a minimum the following default entry is always provided (unless the defaults are overridden by settings loaded from an input file):



The variables named in this list (in the “Variable” field) provide the optimizer with the design variables it is allowed to vary during the optimization, along with the allowable range for the given design variable (through the “Min.” and “Max.” fields). A starting value can also be specified in the “Value” field.

<u>A Note on Specifying Variable Names:</u>
It is clear how to specify a flow property as a design variable, since these values affect the geometry as a whole. To thus use the angle of attack, for instance, one would simply enter “alpha” in the “Variable” field. Note that the variable names appearing here are consistent with those in the XML input file, and not necessarily with the names appearing in the “Geometry” pane of the user interface. Use the “View XML Data” pane to view the XML file.

What may not be as obvious, at least in a multi-element geometry, is specifying a parameter associated with a single element, such as the x-coordinate of the element root leading edge, “xrootle”. In such cases, the variable name should be suffixed with a period followed by the element name. For example, if the element in question were named “Winglet”, the variable name, as supplied here would be “xrootle.Winglet”.

Very Important: Note that for the optimizer to work element names should not have any spaces in them; i.e. “Inboard_Wing” is OK, while “Inboard Wing” is not.

That having been said, essentially any input parameter can be specified as a design variable; please refer to the section “About the Variables” for a list of variable definitions.

2. Specify the constraints.

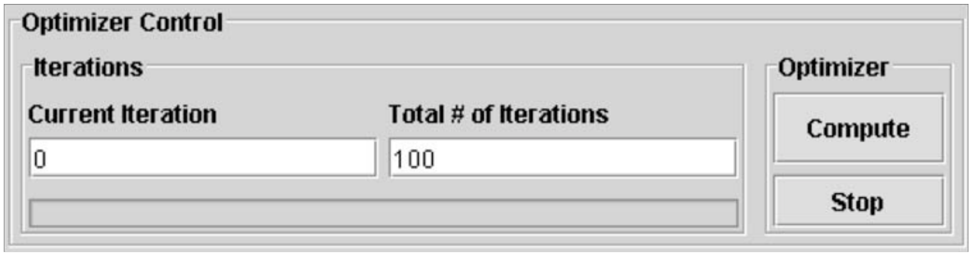
This section differs slightly in that it allows the deletion of all fields so that the optimizer can be run without constraints, if so desired. Otherwise, constraints are specified similarly to the design variables, with the result to be constrained in the “Variable” field, and the upper and lower bounds in the “Max.” and “Min.” fields, respectively.

3. Specify the objective.

Only one entry is allowed in this section, indicating the result that the optimizer should attempt to minimize.

4. Specify the number of iterations.

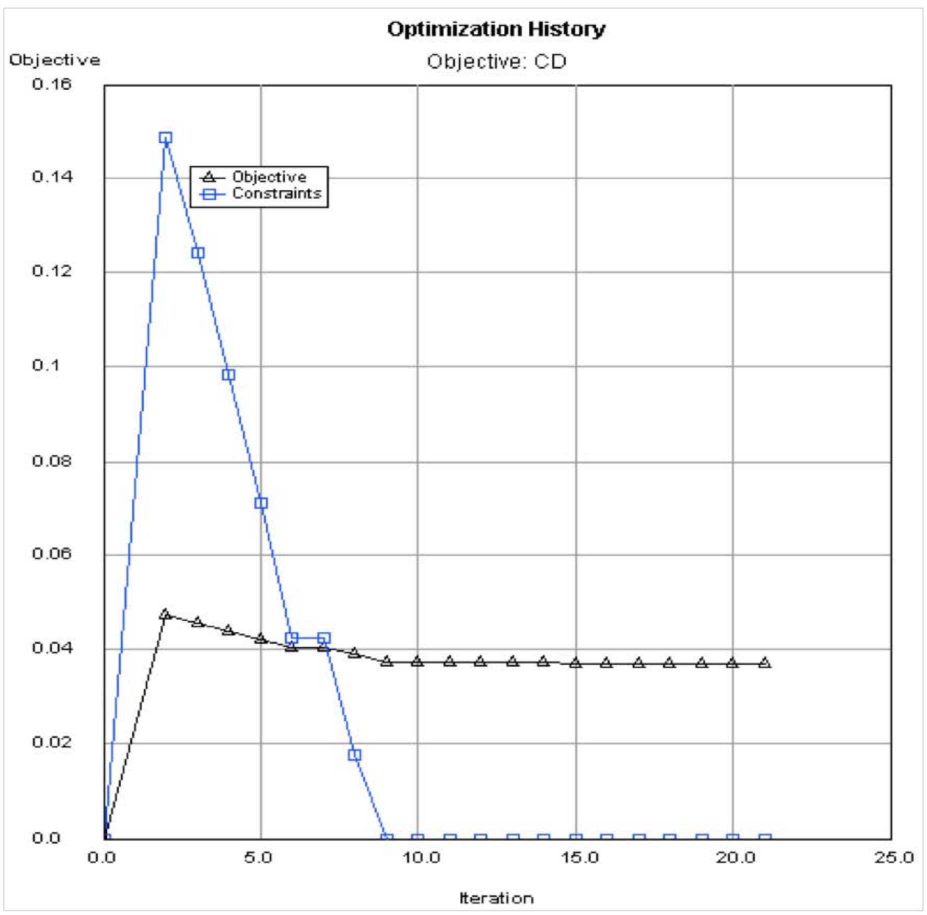
The number of iterations for the optimizer to perform should be specified in the “Total # of Iterations” field of the “Optimizer Control” section.



5. Start the optimizer.

Once satisfied with the settings, pressing the “Compute” button starts the optimizer. To abort an optimization run once started, press the “Stop” button. Note that the applet may still take a moment to respond, since the optimization runs in a separate thread. The optimizer provides visual feedback as to its progress in three ways:

1. A report of the current iteration number, through the “Current Iteration” field.
2. A progress bar.
3. A plot of the “Optimization History”, which shows the current objective value, along with a composite, scaled value of the constraint state. If no constraints are being violated, this value should be zero.



Optimizer Properties

Design Variables: Variable	designVars	Design variables available to the optimizer
Design Variables: Min	designVarMin	Lower bound for corresponding design variable
Design Variables: Max	designVarMax	Upper bound for corresponding design variable
Contraints: Variable	contraintVars	Variables that the optimizer should treat as constraints on the optimization
Constraints: Min	constraintVarMin	Lower bound for corresponding constraint variable
Constraints: Max	constraintVarMax	Upper bound for corresponding constraint variable
Objective: Variable	objectiveName	Objective variable that the optimizer should attempt to minimize
Objective: Min	objectiveMin	Lower bound for the objective value
Objective: Max	objectiveMax	Upper bound for the objective value
Total # of Iterations	maxIterations	Number of iterations in the optimization

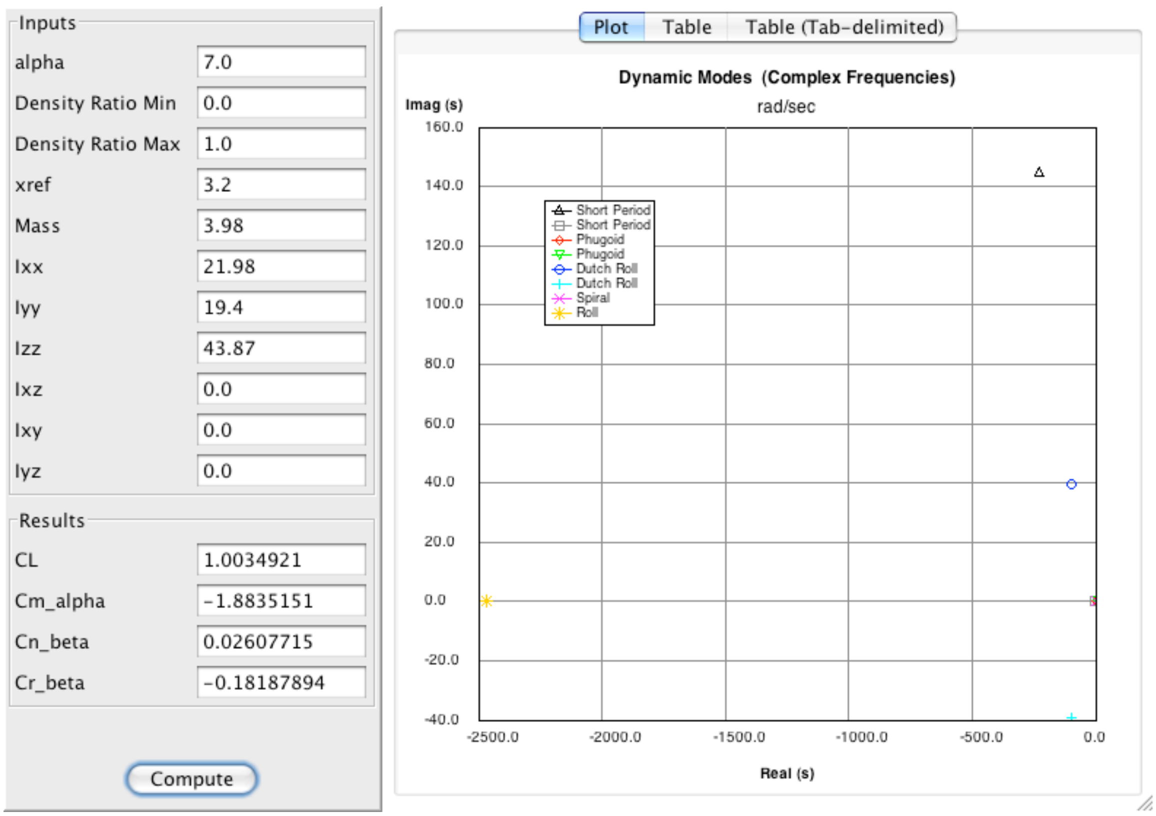
Dynamics

Vehicle inertias can be input in the “Dynamics” pane of the user interface or specified in the XML input file. The US unit system is used. The XML syntax is:

```
<var name="weight"><val>615.7</val></var>
<var name="Ixx"><val>103.8</val></var>
<var name="Iyy"><val>712.8</val></var>
<var name="Izz"><val>719.3</val></var>
<var name="Ixz"><val>137.5</val></var>
<var name="Ixy"><val>0.0</val></var>
<var name="Iyz"><val>0.0</val></var>
```

The data is used to build the inertia matrix as follows:

Ixx	-Ixy	-Ixz
-Ixy	Iyy	-Iyz
-Ixz	-Iyz	Izz



Variables of the Dynamics section

LinAir GUI	XML
Mass	mass
Ixx	Ixx
Iyy	Iyy
Izz	Izz
Ixz	Ixz
Ixy	Ixy
Iyz	Iyz

Density Ratio Min	drMin
Density Ratio Max	drMax
CL_alpha-dot	CL_adot
CD_alpha-dot	CD_adot
Cy_alpha-dot	Cy_adot
Cm_alpha-dot	Cm_adot
Cn_alpha-dot	Cn_adot
Cr_alpha-dot	Cr_adot

To access the results displayed in the GUI in the XML output file, these variables have to be added to the results section of the XML file prior to running LinAir:

```
<var name="RealEig"><val>0.0</val></var>
<var name="ImagEig"><val>0.0</val></var>
<var name="VectorString"><val>"empty"</val></var>
```

After executing LinAir, the real and imaginary parts of the eigenvalues will appear in the XML file like this:

```
<var name="RealEig"><val>-232.1418471160726</val><val>-
0.14382069626344673</val><val>0.0</val><val>0.0</val><val>-
38.08701038659332</val><val>-
38.08701038659332</val><val>0.002512110229587355</val><val>-
2547.983188035416</val><val>-7.28583859910259E-17</val></var>
```

And the following matrix will appear that contains normalized eigenvector magnitudes:

Normalized, dimensionless eigenvectors.
(magnitude only)

Variable	Mode # 1	2	3	4	5	6	7	8	9
Longitudinal motions									
u	0.0351	0.0107	0.0351	0.0351	1.1777E-16	1.1777E-16	-6.7404E-17	4.5575E-20	5.9252E-19
w	0.7935	0.0017	0.7935	0.7935	6.6325E-16	6.6325E-16	1.0713E-17	5.9282E-21	9.5015E-20
q	0.5135	2.6736E-4	0.5135	0.5135	4.0962E-16	4.0962E-16	-1.7306E-18	4.3867E-20	1.4256E-20
tha	1.0	1.0	1.0	1.0	9.1828E-14	9.1828E-14	6.6231E-15	8.3629E-17	5.7913E-17
Lateral motions									
v	3.1826E-17	2.2308E-19	3.1826E-17	3.1826E-17	0.1848	0.1848	2.5672E-5	6.8838E-4	7.2095E-19
p	1.5189E-16	1.2171E-18	1.5189E-16	1.5189E-16	0.1871	0.1871	1.3180E-5	1.0	2.8878E-19
r	1.8407E-16	2.0389E-19	1.8407E-16	1.8407E-16	1.0	1.0	1.6345E-4	0.0068	3.8798E-18
phi	1.0241E-17	1.7888E-17	1.0241E-17	1.0241E-17	0.1899	0.1899	1.0	4.1310E-5	1.0
psi	8.4503E-18	9.6060E-17	8.4503E-18	8.4503E-18	0.0355	0.0355	0.0806	0.0060	2.1476E-15 </val></var>

'RealEig' is the real part of the vector of eigenvalues for the linear system. That tells you if the dynamic modes are stable or unstable.
'VectorString' is a list of the corresponding eigenvectors.

The modes are:
1,2: Short period
3,4: Phugoid
5,6: Dutch Roll
7: Spiral
8: Roll
9: Kinematic

Sample Input File


```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="linair.xsl" ?>
<linair>
<controls>
<var name="lastmodified"><val>Oct 6, 2003 2:54:33 PM</val></var>
<var name="lastupdated"><val>Oct 6, 2003 2:54:33 PM</val></var>
<var name="status"><val>complete</val></var>
<var name="comment"><val>LinAir Input File</val></var>
<class name="mainClass"><val>la4sub</val></class> <class
name="objectiveClass"><val>la4sub</val></class>
</controls>
<inputs>
<var name="configurationname"><val>LinAir Geometry</val></var>
<var name="sectionClass"><val>lasection</val></var>
<var name="sref"><val>130.0</val></var>
<var name="bref"><val>35.0</val></var>
<var name="cref"><val>3.7142857142857144</val></var>
<var name="xref"><val>3.2</val></var>
<var name="yref"><val>0.0</val></var>
<var name="zref"><val>0.0</val></var>
<var name="nelem"><val>2.0001</val></var>
<var name="alpha"><val>7.0</val></var>
<var name="beta"><val>0.0</val></var>
<var name="phat"><val>0.0</val></var>
<var name="qhat"><val>0.0</val></var>
<var name="rhat"><val>0.0</val></var>
<var name="mach"><val>0.1</val></var>
<var name="wakelocation"><val>1.0</val></var>
<var name="reflectgeometry"><val>1.0</val></var>
<var name="ElementName"><val>Wing</val><val>Winglet</val></var>
<var name="xrootle"><val>0.0</val><val>8.2</val></var>
<var name="yrootle"><val>0.0</val><val>17.5</val></var>
<var name="zrootle"><val>0.0</val><val>0.0</val></var>
<var name="xrootte"><val>4.0</val><val>11.0</val></var>
<var name="xtiple"><val>8.0</val><val>10.0</val></var>
<var name="ytiple"><val>17.5</val><val>17.5</val></var>
<var name="ztiple"><val>0.0</val><val>3.0</val></var>
<var name="xtipte"><val>11.0</val><val>12.0</val></var>
<var name="rooti"><val>10.0</val><val>0.0</val></var>
<var name="tipi"><val>0.0</val><val>0.0</val></var>
<var name="npan"><val>10.0</val><val>4.0</val></var>
<var name="cdp0root"><val>0.0070</val><val>0.0070</val></var>
<var name="cdp1root"><val>0.0</val><val>0.0</val></var>
<var name="cdp2root"><val>0.0030</val><val>0.0030</val></var>
<var name="cl0root"><val>0.0</val><val>0.0</val></var>
<var name="cm0root"><val>0.0</val><val>0.0</val></var>
<var name="clmaxroot"><val>1.5</val><val>1.5</val></var>
<var name="cdp0tip"><val>0.0070</val><val>0.0070</val></var>
<var name="cdp1tip"><val>0.0</val><val>0.0</val></var>
<var name="cdp2tip"><val>0.0030</val><val>0.0030</val></var>
<var name="cl0tip"><val>0.0</val><val>0.0</val></var>
<var name="cm0tip"><val>0.0</val><val>0.0</val></var>
<var name="clmaxtip"><val>1.5</val><val>1.5</val></var>
<var name="designVars"><val>alpha</val></var>
<var name="designVarMin"><val>0.0</val></var>
<var name="designVarMax"><val>10.0</val></var>
<var name="constraintVars"><val>CL</val></var>
<var name="constraintVarMin"><val>0.99</val></var>
<var name="constraintVarMax"><val>1.01</val></var>
<var name="objectiveName"><val>CD</val></var>
<var name="objectiveMin"><val>0.0</val></var>
<var name="objectiveMax"><val>0.1</val></var>
<var name="maxIterations"><val>100.0001</val></var>
```

```
</inputs>
<results>
<var name="CL"><val>0.0</val></var>
<var name="CD"><val>0.0</val></var>
<var name="Cy"><val>0.0</val></var>
<var name="Cr"><val>0.0</val></var>
<var name="Cm"><val>0.0</val></var>
<var name="Cn"><val>0.0</val></var>
<var name="Cx"><val>0.0</val></var>
<var name="Cz"><val>0.0</val></var>
<var name="e"><val>0.0</val></var>
<var name="CL_element"><val>0.0</val><val>0.0</val></var>
<var name="CD_element"><val>0.0</val><val>0.0</val></var>
<var name="CY_element"><val>0.0</val><val>0.0</val></var>
<var name="Cr_element"><val>0.0</val><val>0.0</val></var>
<var name="Cm_element"><val>0.0</val><val>0.0</val></var>
<var name="Cn_element"><val>0.0</val><val>0.0</val></var>
</results>
</linair>
```

If you want other results, be sure they appear in the results section, e.g.
For the lift curve slope (CL_alpha), the following would be added into the results section of the xml input file (the part that appears between <results></results>):
<var name=”CL_alpha”><val>0.0</val></var>
Note that the numeric value in the <val> element is not important – it gets overwritten by LinAir when it outputs results.