The Role of Soccer in Defining Identity in World Literature

Drew Woolley
Will It Fly? A Computer-Based Aircraft Design Tool

By
Eric G. Woodfin

UNDERGRADUATE
SENIOR HONORS PROJECT

MAE 496-30
DECEMBER 15, 1994
Abstract

The purpose of this project was to produce a user-friendly, yet reasonably accurate computer application for aircraft design and performance prediction. The result of this endeavor is a beta version of "Will It Fly," a Visual Basic-based application that integrates wing design, engine selection, and cabin design to meet specific aircraft mission requirements. The program is suitable for sophomore college-level, and provides the technical framework needed to successfully teach elementary aerodynamics and aircraft design. This paper describes the program's architecture and the philosophy behind its layout. Although written for the college level, potential modifications to the program are documented which would allow an educational version suitable for middle through high school level students. Development of this secondary-level version would be beneficial to the future of science education by exposing these students to the design process.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INTRODUCTION</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>BACKGROUND</strong></td>
<td>1</td>
</tr>
<tr>
<td>The Nature of Design</td>
<td>1</td>
</tr>
<tr>
<td>Educational Considerations</td>
<td>2</td>
</tr>
<tr>
<td><strong>A COMPUTER-BASED DESIGN TOOL</strong></td>
<td>3</td>
</tr>
<tr>
<td>Aircraft Design Program Philosophy/Goals</td>
<td>4</td>
</tr>
<tr>
<td>Will It Fly? for Windows</td>
<td>5</td>
</tr>
<tr>
<td>Will It Fly? Program Structure</td>
<td>5</td>
</tr>
<tr>
<td>Will It Fly? Analysis Techniques</td>
<td>6</td>
</tr>
<tr>
<td>Secondary School Version Modifications</td>
<td>7</td>
</tr>
<tr>
<td><strong>CONCLUSIONS</strong></td>
<td>8</td>
</tr>
<tr>
<td><strong>APPENDIX A: Will It Fly? for Windows 1.0 User’s Manual</strong></td>
<td>A-1</td>
</tr>
<tr>
<td><strong>APPENDIX B: Vortex-Lattice Code Module</strong></td>
<td>B-1</td>
</tr>
</tbody>
</table>
INTRODUCTION

*Will It Fly?* is a computer-based educational tool intended for teaching the design process to sophomore engineering students. The overall goal of the program is to create a design interface that successfully performs this function, but is user-friendly. The interface should also be visually appealing in order to motivate students in learning about the design process.

To develop a software package such as *Will It Fly?*, careful consideration of the user’s educational needs and program requirements to meet these needs is required. For this reason, background information about this project will be discussed first, followed by the program philosophy and layout.

BACKGROUND

The Nature of Design

When creating design-education tools, the nature of the design process must be considered. There are several characteristics of design that should be conveyed by such tools. The following are a few:

- Design is a synergy of various disciplines.
- Design is iterative.
- Design includes a creative element.

First of all, design is a synergy of various disciplines. The design of a product never involves only one subject area, but rather unites several. For example, in engineering design, application of mathematics and physics are both required. The concepts of algebra and geometry are used to calculate a part’s dimensions. Physics must be used to analyze its performance or its manufacturing process. Physical analysis also requires the exercise of mathematical skills. Therefore, various disciplines are interrelated in multiple ways.

Next, design is an iterative process. Rarely does the first conception of a design survive to become the final design. Many times, a preliminary analysis shows that a design will not perform as intended or will simply fail to perform at all. After this initial analysis, iteration on the design is performed to obtain the desired performance. This iterative process usually involves the following steps:

1. Isolate a specific design parameter.
2. Make a small change to the parameter.
3. Reanalyze the design.
4. Note the overall effect of the parameter change.
5. Repeat as necessary to optimize the design.

While successful application of the laws of mathematics and physics is crucial to the correct analysis of a design, there is another element of the design process which is significant. This element is simply creativity on the part of the designer. It should be pointed out that a creative person is not necessarily a good designer. But, through design experience, this creativity can be transformed into an intuitive ability to devise creative solutions to design problems. The impact of this kind of creative problem-solving can be very significant in terms of a product’s overall appearance, cost, and performance.

**Educational Considerations**

When creating design-education tools, the educational needs of the student must be considered. These needs vary with the target age group, and will be discussed on that basis.

On the college level, engineering design courses are typically taught as “senior design,” meaning that a student must be in his or her fourth year before experiencing project-length design courses. At the University of Alabama in Huntsville, for example, the Department of Mechanical and Aerospace Engineering teaches design in two courses: Introduction to Engineering Design (MAE 493), and Mechanical Engineering Senior Design (MAE 465). Each of these courses are taught at the senior level, which means a student takes them no sooner than his or her fourth year. Such late exposure to the design process deprives the student of valuable experience in independent design. Therefore, the need exists for sophomore to junior-level design coursework in the college curriculum.

Computer-based tools are appropriate for both the college and secondary school level. On the college-level, course-specific design software can be used as a tool to demonstrate the use of advanced techniques of analysis. Such software can be incorporated into semester-long design projects that demonstrate the material presented during the term. Assignments of this type illustrate the relationships between design parameters and provide valuable design experience early in a student’s college career.

The need for training in the design process is not limited to the college level. This training is also lacking in our nation’s secondary schools. At the present time, there is a declining interest in science, engineering, and mathematics among high school students. In fact, there is a general avoidance of these subject areas (1:196). One reason for this decline is that the typical high school student is not exposed to material that demonstrates the usefulness of advanced subjects such as high school geometry, algebra, precalculus, and physics. Weakened teaching staffs and relatively low learning expectations also seem to be a major contributor to this alarming situation (1:196). Another cause is the lack of new and interesting ways to bring mathematical and scientific concepts to life. One way to accomplish this is to use computer-based tools in the classroom.
The need for computer-based design tools in secondary schools can be established by simply looking to science educators. Most science education reform proposals incorporate increased use of computers in the classroom. Although computers are found in most junior high and high schools today, students are generally given little exposure to them. Over 85% of middle-school students never use computers (2:77). For students who do use computers, they typically use them less than fifteen minutes per week. Most students do not know how to turn a computer on, much less interact and learn with it. The advanced concepts of computer-aided design and modeling are certainly never covered.

This deficiency should be addressed, according to science educators. Computers should be an integral part of the secondary school student’s science education. This requires the development of suitable educational software. Some of the characteristics required of this type of software include the following:

- Integration of science and mathematics (3:6)
- Systems modeling (2:78)
- Simulations (2:78)
- Tutorials/On-line help (2:78)

These needs can also be satisfied through computer-based design tools. The approach for this age group differs from college-level, though. At the secondary level, software of this type shouldn’t emphasize advanced techniques, but rather should serve to demonstrate the relationships that exist between the various components of a design. This will allow the student to establish cause and effect relationships between design parameters by letting them see how changes favorably or adversely affect overall performance of systems.

It is evident that the development of computer-based design tools is needed, for both the college and secondary school levels. The goal of this project is to produce such a design tool for the college sophomore level. Appropriate modifications can then be made to use the tool at the secondary level.

A COMPUTER-BASED DESIGN TOOL

Aircraft design was chosen as the basis for this design tool. There are several reasons aircraft design is suitable for this endeavor. First of all, aircraft design is synergistic, in that it requires the integration of mathematics and physics, which as previously mentioned is part of the nature of the design process. In fact, a synergy between the different components of an aircraft can be realized through design work. Second, aircraft design requires iteration of parameters. To reach an optimum combination of components requires iteration through a variety of design variables. Also, creativity is necessary in the design of aircraft configurations when nontraditional
solutions are considered. Aircraft design is also suitable in that virtually everyone is familiar with airplanes. This universality of the concept of flight makes the software accessible to everyone. Thus, aircraft design has the required characteristics of the design process and is suitable as a basis for a design-education tool.

Educational aircraft design tools of this type are not without precedent. A similar effort by Charles Eastlake of Embry-Riddle Aeronautical University has proven to be very effective in stimulating interest in science and engineering. Eastlake's design program is DOS-based, with no graphical interface. Also, his program lacks advanced techniques of analysis. Despite these weaknesses, he reports great success in motivating students with the program (4:4). Eastlake's success supports the use of aircraft design as the basis of design-education tools.

Now that the avenue through which to teach design is chosen, consideration of the software package itself is required. The first question that comes to mind is "What programming language will be used?" This takes serious thought, since there is a plethora of programming languages available. The major demand placed on the programming language is that it have the ability to produce an aesthetic user interface, while providing the programmer with extensive capabilities. In other words, the resulting interface should be powerful, yet "user-friendly." For these reasons, Microsoft Visual Basic 3.0 was chosen as the programming environment. It's ability to produce Windows-based applications gives it the capacity to produce a visually-appealing, easy-to-use interface.

Aircraft Design Program Philosophy/Goals

With subject and programming tool in hand, goals for the program were established. These are based on the requirements of educators and students as mentioned previously. They include:

- User-friendly interface
- Visual presentation of design information
- Accurate design analysis
- Ability to iterate and optimize designs
- Potential for secondary school version

First of all, the program's interface, or front end, must be user-friendly. Since Visual Basic was chosen, the resulting program is Windows-based. Therefore, the program can have many of the convenient features usually found in Windows applications. These include pull-down menus, button toolbars, drop-down edit boxes, list boxes, and generic diagrams to aid data entry. An interface that includes these features is very aesthetic and should keep the user interested.

Next, the program should present design information visually. The graphical nature of the Windows environment affords many avenues through which to do this.
Data graphs are one way to display data sets and provide insight into the nature of the data quickly. Spinners allow the user to scroll through design results, displaying only one piece of data at a time. Magnitude gauges, much like graphs, allow the user to quickly see comparisons between values. Color changes can also be used to convey the nature of data to the user. For example, when a "bad" value is encountered, an onscreen object may turn red. When a "good" value is encountered, the object may turn green. These and other methods of visually-appealing display should be used throughout the aircraft design program to stimulate and retain the interest of the user.

While providing a fun, interesting way to do design, the program must also provide accurate results, in order to reinforce theory learned in the classroom. This may be done through application of textbook theory to computer-based numerical methods. Two areas where this is useful in aircraft design is in the analysis of the wing aerodynamic parameters, and in the overall performance of the aircraft.

While providing an accurate, user-friendly interface, the program interface must also be "designer-friendly." The front end should be freeform enough to allow the user to switch from screen to screen quickly. This would allow the user to input and analyze design changes quickly. This freeform front end supports the iterative nature of the design process, providing the opportunity to improve the current design based on previous outcomes. (5:5)

As mentioned earlier, the focus of this project is the college sophomore level. At the same time, though, the potential for a secondary school version exists. Through simple modifications and additions, the resulting program should be adaptable to this level.

**Will It Fly? for Windows**

The program developed is called "Will It Fly? for Windows." It is a beta version, which simply means that it is a first cut. The program has minimal hardware requirements (4MB RAM, 2MB hard drive space, VGA graphics) and can run on any machine that has Microsoft Windows 3.x installed. For this reason, the program is not only "user-friendly," but also "budget-friendly." For detailed information about Will It Fly?, please consult Appendix A, which contains the program’s user manual.

**Will It Fly? Program Structure**

*Will It Fly?* contains a series of screens which allow the user to enter design data related to various components of an aircraft. These areas are:

- Itinerary - leaving and destination cities (based on latitude and longitude)
- Wing Design - wing shape description/analysis
- Engine Selection - engine selection and number of engines needed
Will It Fly? A Computer-Based Aircraft Design Tool

- Cabin Options - passenger data

The Will It Fly? interface features a screen or dialog box for each of these design areas. Each of these screens has default design data entered upon startup. A design checklist is included to let the user see the areas in which he or she has changed default values.

Two full-screen summaries, the wing summary and overall design summary, are also included to give the user in-depth information. A wing design summary appears after analysis of the wing-shape. This summary displays data that can be used by the college user to gain a deeper knowledge of aerodynamics. This data comes in the form of calculated geometric and aerodynamic parameters that appear in separate windows on the screen. Also, graphs of aerodynamic coefficients are displayed for continuity.

The overall design summary is the pinnacle of the Will It Fly? program. It is here that the overall performance of the designed configuration is calculated and displayed. The maximum range of the aircraft is displayed here, and is compared with the range required as determined by the itinerary. At this point, the user may return to other design screens and change parameters. The overall summary can be recalculated at any time, allowing the user to see the overall impact of his or her changes.

Will It Fly? Analysis Techniques

The design data on the summary screens is calculated through the use of numerical methods. The wing design, for example, uses the vortex-lattice method, which is the single largest procedure in Will It Fly? The vortex-lattice code module (6) is included in Appendix B.

The vortex-lattice technique (7:261-282) specifies a grid on a proposed wing planform. A horseshoe-shaped vortex filament is placed on each panel, with the assumption that the air flow must be tangential to the wing’s surface (in other words, air cannot pass through the wing) at specific control points. Filament geometry is used to produce a set of simultaneous equations to solve for the “strength” of each filament. These strengths are then used to calculate lift and induced drag as well as other aerodynamic properties of the wing.

On the overall design summary screen, the fundamental aircraft performance equation (8:378-384) is used to calculate the range of the aircraft:

\[
R = \frac{V_C \left( \frac{C_L}{C_D} \right)}{TSFC} \ln \left( \frac{1}{1 - FF} \right)
\]
As shown by the equation, the range is a function of the cruise velocity \( V_c \), the thrust specific fuel consumption (TSFC), the fuel fraction (FF), and most importantly, the lift to drag ratio, \( C_L/C_D \), which is a function of wing shape. The fuel fraction is determined based on the desired range as shown in the table below:

<table>
<thead>
<tr>
<th>Range Condition (nautical miles)</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>R&lt;=750</td>
<td>0.333</td>
</tr>
<tr>
<td>750&lt;R&lt;=1500</td>
<td>0.375</td>
</tr>
<tr>
<td>R&gt;1500</td>
<td>0.400</td>
</tr>
</tbody>
</table>

In the current version of the program, a cruise velocity of 460 knots and cruise altitude of 30,000 feet is assumed and held constant. TSFC is determined by the engine selected. Therefore, the range of the aircraft design is really a function of lift to drag ratio only, as all other independent variables in the equation are held constant. Since the goal of the design is to obtain the maximum range for a configuration, the lift to drag ratio must be maximized in order to maximize the equation for \( R \). This is a focal point for design iteration in *Will It Fly?*, the goal being to design a wing with a maximum \( C_L/C_D \).

Various other equations are used throughout the program to aid the user in making design choices. One issue the user faces in engine selection is takeoff thrust. The calculated takeoff thrust is displayed on the wing summary and the engine selection screen. This aids the user in selecting a specific engine and number of engines to provide the needed thrust to takeoff.

Takeoff thrust, \( T_{TO} \), is determined by the wing shape and is calculated by using the following equation:

\[
T_{TO} = \frac{1}{2} \rho_{SL} V_{TO}^2 S_W C_{D_{TO}}
\]

The density of air at sea-level, \( \rho_{SL} \), is a constant. The takeoff velocity, \( V_{TO} \), is held constant at 145 knots. \( S_W \) is the area of the proposed wing shape. \( C_{D_{TO}} \) is assumed to be \( C_D(\alpha=5^\circ) \) as calculated by the vortex-lattice method.

**Secondary School Version Modifications**

One goal mentioned previously is to retain the potential to produce a secondary school version of *Will It Fly*? The ease with which programming changes can be made through Visual Basic provides the means for doing this. Some future modifications include:

- Hide advanced technical information
- Incorporate more visual aids
- Produce preliminary design worksheets
The current version of *Will It Fly?* includes advanced technical data and techniques. In a secondary level version, not only would this surplus of data be useless to the student user, it would only serve to confuse him or her. Therefore, one necessary modification to the program is to hide much of the data output. This includes removing the option to create a vortex-lattice data file, which is currently a program option. This alteration would make the program more focused for the younger user, eliminating the potentially scary or frustrating excess output. Removal of this information would allow the young user to concentrate on observing the impact of wing geometry on the lift to drag ratio, $C_L/C_D$.

Although the program now exhibits an attractive interface, more visual information is needed to support the needs of the younger user. One modification would be to include an on-line help system specially tailored to the secondary school level. Another idea is to create a set of aerodynamic “flash cards.” These would serve to reinforce the concepts demonstrated by the program by providing the option to view the cards from anywhere in the program. Also, a separate menu option could be included to allow students to view the cards without necessarily working through a design.

Another modification that is key to emphasizing the synergy of mathematics and physics and their relation to the real world is the inclusion of preliminary design worksheets. These worksheets would show students how to apply geometry to calculating design parameters such as wing area, and algebra to determine aircraft weight and takeoff thrust. Also, the worksheets could let students use geographical data to determine the range needed by their design. These worksheets could be used as a precursor to using the computer program. Students could see how mathematics and physics fit into the scheme of aircraft design, and then confirm their results with the program. After confirming their own calculations, students could use the program to further refine their design by changing settings and seeing how the overall design is affected. This lets the students establish cause and effect relationships between design variables, a point stressed by science educators.

**CONCLUSIONS**

*Will It Fly?* for Windows is potentially a valuable design-education tool for early college design classes. Its visually-appealing interface serves to motivate and interest users to pursue more proficient design configurations. Through use of the program, users see the relationships between the various components of the aircraft. Most importantly, the program allows the user to see the iterative nature of design by allowing him or her to easily change design values and then reanalyze the design.

Even though the current college version has immense educational value, a real potential to create a secondary school package exists. The use of such programs can have an impact on the future of science education, through the early introduction of design and
its demonstration of cause and effect relationships in the real world. Although further development of the college version is warranted, the development of the secondary level version is foreseeable in the near future and will be an exciting continuance of this project.
REFERENCES


4. Eastlake, C. An aircraft design project for the high school level. Proceedings of the 32nd AIAA Sciences Meeting and Exhibit. 1994 January 10-13; Reno, NV. AIAA 94-0859


BIBLIOGRAPHY


Bybee, R.W.  [and others].  Science and technology for the middle years: Frameworks for curriculum and instruction.  The National Center for Improving Science Education: 1990.


Eastlake, C.  An aircraft design project for the high school level.  Proceedings of the 32nd AIAA Sciences Meeting and Exhibit.  1994 January 10-13; Reno, NV. AIAA 94-0859


APPENDIX A

Will It Fly for Windows 1.0 User’s Manual
INTRODUCTION

Will It Fly? is an aircraft design program developed to allow lower level engineering students to produce preliminary airplane designs. Although the program requires little knowledge of aerodynamics, it provides accurate predictions of flight performance through the use of numerical methods. This allows students to accurately analyze their proposed designs while learning aerodynamic principles along the way. The program is flexible enough to allow "what if" analysis, in which changes to the design can be made on the fly. These changes can immediately be analyzed, allowing students to see what effect these changes have on the overall performance of the vehicle.

GETTING STARTED

Minimum System Requirements:
- Any IBM-compatible machine with an 80386 processor or higher (80486 recommended).
- Microsoft Windows version 3.0 or later
- VGA graphics or better
- 4 megabytes of memory
- 1 megabyte of hard drive space
- A mouse or other pointing device

To Install Will It Fly? for Windows:
1. Start Microsoft Windows.
2. Insert the disk in drive A.
3. From the file menu in Program Manager, choose Run.
4. Type a:setup
5. Press ENTER.
6. Follow the instructions on the setup screen.

PROGRAM SCREENS

Will It Fly? is completely menu-driven, with pull-down menus located on the main screen.

Figure 1. Will It Fly? Pull-Down Menu Structure
Will It Fly? is structured in such a way that the user can change a set of options and see how it affects the overall design. Each set of options may be set in any order, and the user has the option to use default settings for some option groups. The four main sets of options are:

- Wing Design
- Engine/Propulsion Options
- Cabin/Fuselage Options
- Itinerary

Because of the free-form way Will It Fly? allows you to create a design, it seems the best way to explain how to use the program is to describe each screen, one by one. A detailed description of each screen follows, with an explanation of each of its options. An example design is also included.

Will It Fly Version 1.0 Screens

Main Screen
Wing Design Screen
Engine/Propulsion Screen
Cabin/Fuselage Screen
Itinerary Screen
Wing Design Summary
Overall Design Summary

A summary of each screen follows.
Main Screen

HOW TO ACCESS THIS SCREEN: This screen is available upon program startup. All other program screens return here when they are closed.

PURPOSE: This screen contains all menus used by Will It Fly? It also has a large, convenient button bar at the bottom of the screen which gives quick access to all major program functions.

Figure 2. Will It Fly? Main Screen
Wing Design Screen

HOW TO ACCESS THIS SCREEN:  Menu: Specifications + Aero + Give Corner Points
   Button Bar: Aero + Give Corner Points

X-Y Coordinates

Enter the x, y coordinates for points A, B, C, D, E, and F, following the diagram on the screen. These points will be used to draw and analyze the wing.

No. of Spanwise Panels

Enter the number of panels to divide the wing into along its span. The vortex-lattice method uses these panels to analyze the wing.

No. of Chordwise Panels

Enter the number of panels to divide the wing into along its chord. The vortex-lattice method uses these panels to analyze the wing.

Maximum Camber

Enter the maximum Z/C of the wing.

Write Output to File

Check this box if you want the vortex-lattice analysis to be written to a file when the wing is analyzed.

Calculate

Starts the vortex-lattice analysis of the wing and shows the wing design summary. (The run time for this option varies according to your machine's processor speed. A 486DX33 or faster is recommended.)

Latest Wing Design

Shows the wing summary screen.

View This Wing

Shows the shape of the current points on the screen. This option does NOT perform the complete wing analysis. Use this feature when you want to "play" with the wing shape before analyzing it.
Propulsion Screen

HOW TO ACCESS THIS SCREEN:  Menu: Specifications + Engine/Propulsion
Button Bar:  Propulsion

Engines

A list box displays the available engines. Select an engine by clicking on the one you want. When you do this, the specifications for your choice will appear on the right side of the screen. The default engine is the first engine in the alphabetical list.

Specifications

Displays specifications for the selected engine. These specifications include:

- Specific Fuel Consumption
- Efficiency
- Thrust Rating
- Engine Weight
- Engine Type
Cabin Screen

HOW TO ACCESS THIS SCREEN: Menu: Specifications + Cabin/Fuselage
Button Bar: Cabin

No. of Passengers

Enter the number of passengers to be transported. The default value is 200.

Lbs. Luggage per Passenger

Enter the weight of the luggage carried by each passenger. The default value is 20 lbs.

Average Passenger Weight

Enter the average passenger weight here. The default value is 170 lbs.
Itinerary Screen

The purpose of the Itinerary Screen is to pick the leaving and destination cities and calculate the distance to be traveled by the aircraft.

HOW TO ACCESS THIS SCREEN: Menu: Specifications + Itinerary

Button Bar: Itinerary

To create an itinerary:

Select Leaving Coordinates
1. Click on the "Leaving" option button on the left side of the screen.
2. The yellow text box on the left side of the screen now says “Choose Continent.” Select a continent to leave by clicking one of the continent buttons on the world map.
3. The list box on the left side of the screen now says “Choose City.” Pull the list down and select a city. The latitude and longitude coordinates for that city are entered below the city name.

Select Destination Coordinates
1. Click on the "Destination" option button on the right side of the screen.
2. The yellow text box on the right side of the screen now says “Choose Continent.” Select a continent to go to by clicking one of the continent buttons on the world map.
3. The list box on the right side of the screen now says “Choose City.” Pull the list down and select a city. The latitude and longitude coordinates for that city are entered below the city name.

Calculate Distance (optional)
You may now calculate the distance between the cities you have selected. Do so by clicking the “Calculate Distance” button in the lower center of the screen. (The default distance is 500 miles. No cities are entered as defaults.)

**Please note that the leaving and destination points may be changed at any time while at this screen. Simply follow the instructions above.

Confirm Itinerary
Click this button to save (in memory) the itinerary settings currently on the screen.

Cancel
Click this button to cancel any itinerary changes. The itinerary reverts to the itinerary set when the screen was first called up.
Design Checklist Screen

This dialog box shows the status of the current design project. It indicates which options have been customized by the user and which are still set at defaults.

HOW TO ACCESS THIS SCREEN: Menu: Design + Checklist
Button Bar: Design Button + Checklist

- Wing Design Check Box
- Engine Selection Check Box
- Cabin Options Check Box
- Itinerary Check Box

Unchecked boxes indicate default settings. Checked boxes indicate the option has been customized.

Overall Design Summary Button
Calculates and displays the Overall Design Summary Screen.

Close Button
Closes the design checklist.
Wing Design Summary Screen

This screen contains a summary of the wing analysis results. It appears after the wing design is calculated, or is available at anytime through the pull-down menus (Design + Latest Wing Design).

HOW TO ACCESS THIS SCREEN: Menu: Design + Latest Wing Design

Button Bar: Design Button + Latest Wing Design

Print Graphs Button

Prints two pages of graphs:
- $C_L$, $C_D$, and $(C_L/C_D)/6$ vs. $\alpha$ on the first page
- $C_L/C_D$ vs. $\alpha$ on the second page

$C_L$, $C_D$ Button

This button is located on the graph. It toggles between the $C_L$ & $C_D$ vs. $\alpha$ graph and the $C_L/C_D$ vs. $\alpha$ graph.

$\alpha$ = (Spinner Button)

Changes the value of $\alpha$ in steps of one. ($1 \leq \alpha \leq 30$)

Overall Design Summary Screen

This screen contains a summary of the overall design. The design summary is recalculated each time this screen is called up, using the current parameters from all screens.

HOW TO ACCESS THIS SCREEN: Menu: Design + Overall Summary

Button Bar: Design Button + Overall Summary
AN EXAMPLE DESIGN...

Will It Fly? provides the user with four groups of aircraft options, some or all of which may be modified from preset default values. These areas include:

- Itinerary
- Wing Design
- Engine Selection
- Cabin Options

To demonstrate each of these areas, the following example design modifies each option group.

Start Will It Fly?

Start the program by double clicking the Will It Fly? icon on the program manager desktop.

Will It Fly?

Set Itinerary

Next, choose a departure and destination point for the mission. The final design must be able to travel the distance between these two points.

- From the main screen, click the “Itinerary” button. The “Itinerary” screen will appear.

Select Leaving Coordinates

1. Click on the “Leaving” option button on the left side of the screen.
2. The yellow text box on the left side of the screen now says “Choose Continent.” Select North America by clicking the “North America” button on the world map.
3. The list box on the left side of the screen now says “Choose City.” Pull the list down and select “Los Angeles.” The latitude and longitude coordinates for Los Angeles are entered below the city name.

Select Destination Coordinates

1. Click on the “Destination” option button on the right side of the screen.
2. The yellow text box on the right side of the screen now says “Choose Continent.” Select Europe by clicking the “Europe” button on the world map.
3. The list box on the right side of the screen now says "Choose City." Pull the list down and select "Paris." The latitude and longitude coordinates for Paris, France are entered below the city name.

- **Calculate Distance**
  You may now calculate the distance between Los Angeles and Paris. Do so by clicking the "Calculate Distance" button in the lower center of the screen. The distance should read 4,916 nautical miles, and 5,653 statute miles.

After these steps, the Itinerary screen should look like Figure 3.

![Figure 3](image)

- Now, press the Confirm Itinerary button and press OK to save the information.
Design the Wing

- From the main screen, press the "Aero" button at the bottom of the screen. Two options will appear. Select "Give Wing Corners." The screen in Figure 4 should appear.

- Before changing anything, view the default wing by clicking the "View This Wing Shape" button. The screen in Figure 5 will appear. Press "Close" to close the screen.

- Now, in the box labeled "Wing Corner Points," enter the following:

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th></th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>40</td>
<td></td>
<td>-80</td>
</tr>
<tr>
<td>C</td>
<td>55</td>
<td></td>
<td>-80</td>
</tr>
<tr>
<td>D</td>
<td>25</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>55</td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>F</td>
<td>40</td>
<td></td>
<td>80</td>
</tr>
</tbody>
</table>

- In the box labeled "No. Spanwise Panels," type "12."

- In the box labeled "Angle of Attack," type "5."
Now, to analyze the wing, click the "Calculate" button. Depending on your machine speed, this will tie your machine up from about 10 seconds to one minute. When the analysis is complete, the Wing Design Summary screen appears (Figure 6).

- Click the "Close" button to close the Wing Design Summary screen.

Select Engine

- From the main screen, click the "Propulsion" button. The "Select Engine" dialog box will appear.
- In the box labeled "Engines," click on "JT4A-3" and specify 2 engines in the "No. of engines" field. Specifications for this engine will appear in the "Specifications" box. The screen should look like Figure 7:
- Click "OK" to close the dialog box.
Set Cabin Options

- From the main screen, click the “Cabin” button. The “Cabin” dialog box will appear.

- Make the following changes:
  ⇒ Number of Passengers = 250
  ⇒ Lbs. Luggage/Passenger = 24
  ⇒ Avg. Passenger Weight (lbs.) = 160

The dialog box should look like Figure 8:
The preliminary design process is now complete.

Figure 8.
Review the Design

- From the main screen, click the "Design" button. Select the "Checklist" option.
- A dialog box will appear, like that in Figure 9. Note that all four major options are checked. This shows that some or all of the defaults values have been changed for these areas.

Figure 9.
Run the Design Summary

From the Design Checklist, run the design summary by pressing the Overall Design Summary button. The screen in Figure 10 will appear.

The green gauges in the lower right-hand corner of the screen indicate that the maximum possible range of the aircraft is greater than the required range. Therefore, the design can complete the mission.
APPENDIX B

Vortex-Lattice Code Module
Sub VORTEX1()

On Error GoTo ErrHandler
If vlm1!Check1.Value = True Then
    vlm1!CMDialog1.Filter = "All Files (*.*)|Text Files (*.txt)"
    vlm1!CMDialog1.FilterIndex = 2
    vlm1!CMDialog1.Action = 2
    Open vlm1!CMDialog1.Filename For Output As #1
End If
vlm1.MousePointer = 11

PI = 4 * Atn(1)
RTD = 180 / PI

'Take wing coordinates, no. panels, and camber from screen
Xa = Val(vlm1!Xa.Text)
Ya = Val(vlm1!Ya.Text)
Xb = Val(vlm1!Xb.Text)
Yb = Val(vlm1!Yb.Text)
Xc = Val(vlm1!Xc.Text)
Yc = Val(vlm1!Yc.Text)
Xd = Val(vlm1!Xd.Text)
Yd = Val(vlm1!Yd.Text)
Xe = Val(vlm1!Xe.Text)
Ye = Val(vlm1!Ye.Text)
Xf = Val(vlm1!Xf.Text)
Yf = Val(vlm1!Yf.Text)

WingSummary!Xa.Text = Xa
WingSummary!Ya.Text = Ya
WingSummary!Xb.Text = Xb
WingSummary!Yb.Text = Yb
WingSummary!Xc.Text = Xc
WingSummary!Yc.Text = Yc
WingSummary!Xd.Text = Xd
WingSummary!Yd.Text = Yd
WingSummary!Xe.Text = Xe
WingSummary!Ye.Text = Ye
WingSummary!Xf.Text = Xf
WingSummary!Yf.Text = Yf

NS = Val(vlm1!Spanwise.Text)
NC = Val(vlm1!Chordwise.Text)
ZMC = Val(vlm1!Camber.Text)
\[ N = \text{NS} \times \text{NC} \]
\[ \text{DY} = (\text{Yf} - \text{Yb}) / \text{NS} \]
\[ \text{DO} = (\text{Xd} - \text{Xa}) / \text{NC} \]
\[ \text{DT} = (\text{Xc} - \text{Xb}) / \text{NC} \]
\[ I = 0 \]
\[ \text{N1} = \text{NS} / 2 \]

'Left wing \( X(I), Y(I) \)

For \( K = 1 \) To \( \text{N1} \)

For \( J = 1 \) To \( \text{NC} \)

\[ I = I + 1 \]

\[ Y(I) = \text{Yb} + (K - .5) \times \text{DY} \]

\[ X(I) = (\text{Xb} + (J - .25) \times \text{DT} - \text{Xa} - (J - .25) \times \text{DO}) / \text{Yb} \times Y(I) + \text{Xa} + (J - .25) \times \text{DO} \]

\[ \text{XLE} = (\text{Xb} - \text{Xa}) / \text{Yb} \times Y(I) + \text{Xa} \]

\[ \text{XTE} = (\text{Xc} - \text{Xd}) / \text{Yb} \times Y(I) + \text{Xd} \]

\[ \text{XXC} = (X(I) - \text{XLE}) / (\text{XTE} - \text{XLE}) \]

\[ \text{dZdX(I)} = 4 \times \text{ZMC} \times (1 - 2 \times \text{XXC}) \]

Next \( J \)

Next \( K \)

'Right wing \( X(I), Y(I) \)

\[ N2 = \text{N1} + 1 \]

For \( K = \text{N2} \) To \( \text{NS} \)

For \( J = 1 \) To \( \text{NC} \)

\[ I = I + 1 \]

\[ Y(I) = (K - \text{N1} - .5) \times \text{DY} \]

\[ X(I) = (\text{Xf} + (J - .25) \times \text{DT} - \text{Xa} - (J - .25) \times \text{DO}) / \text{Yf} \times Y(I) + \text{Xa} + (J - .25) \times \text{DO} \]

\[ \text{XLE} = (\text{Xf} - \text{Xa}) / \text{Yf} \times Y(I) + \text{Xa} \]

\[ \text{XTE} = (\text{Xe} - \text{Xd}) / \text{Yf} \times Y(I) + \text{Xd} \]

\[ \text{XXC} = (X(I) - \text{XLE}) / (\text{XTE} - \text{XLE}) \]

\[ \text{dZdX(I)} = 4 \times \text{ZMC} \times (1 - 2 \times \text{XXC}) \]

Next \( J \)

Next \( K \)

'Left wing vortex filament loc. \( X1(I), Y1(I); X2(I), Y2(I) \)

\[ I = 0 \]

For \( K = 1 \) To \( \text{N1} \)

For \( J = 1 \) To \( \text{NC} \)

\[ I = I + 1 \]

\[ Y1(I) = \text{Yb} + (K - 1) \times \text{DY} \]

\[ Y2(I) = \text{Yb} + K \times \text{DY} \]

\[ \text{XT1} = (\text{Xb} + (J - .75) \times \text{DT} - \text{Xa} - (J - .75) \times \text{DO}) / \text{Yb} \]

\[ \text{XT2} = \text{Xa} + (J - .75) \times \text{DO} \]

\[ \text{X1(I)} = \text{XT1} \times Y1(I) + \text{XT2} \]

\[ \text{X2(I)} = \text{XT1} \times Y2(I) + \text{XT2} \]

\[ \text{XM(I)} = \text{XT1} \times Y(I) + \text{XT2} \]

Next \( J \)

Next \( K \)

'Right wing vortex filament loc. \( X1(I), Y1(I); X2(I), Y2(I) \)

For \( K = \text{N2} \) To \( \text{NS} \)

For \( J = 1 \) To \( \text{NC} \)

\[ I = I + 1 \]

\[ Y1(I) = (K - \text{N1} - 1) \times \text{DY} \]

\[ Y2(I) = (K - \text{N1}) \times \text{DY} \]

\[ \text{XT1} = (\text{Xf} + (J - .75) \times \text{DT} - \text{Xa} - (J - .75) \times \text{DO}) / \text{Yf} \]

\[ \text{XT2} = \text{Xa} + (J - .75) \times \text{DO} \]

\[ \text{X1(I)} = \text{XT1} \times Y1(I) + \text{XT2} \]

B-3
\[ X2(I) = XT1 \times Y2(I) + XT2 \]
\[ XM(I) = XT1 \times Y(I) + XT2 \]
Next J
Next K
\[ S = ((Xd - Xa) + (Xe - Xf)) \times Yf \]
\[ B = Yf - Yb \]
\[ AR = B \times 2 / S \]
\[ CBAR = S / B \]
\[ CROOT = Xd - Xa \]
\[ XCL4 = Atn(((.25 \times Xe + .75 \times Xf) - .25 \times CROOT) / Yf) \times RTD \]
\[ CT = Xe - Xf \]
\[ TR = CT / CROOT \]
\[ XLAMB = Atn((Xf - Xa) / Yf) \times RTD \]

*Write to output file*

If vlm1ICheck1.Value = True Then

Print #1, , , "VORTEX LATTICE METHOD"
Print #1, , , " CAMBERED WING"
Print #1, "No. of spanwise panels = "; NS
Print #1, "No. of chordwise panels = "; NC
Print #1, "Total no. of panels = "; NS * NC
Print #1,
Print #1, "WING CORNER POINTS:"
Print #1, "XA = " & Format(Xa, "0.0000")", "YA = " & Format(Ya, "0.0000")
Print #1, "XB = " & Format(Xb, "0.0000")", "YB = " & Format(Yb, "0.0000")
Print #1, "XC = " & Format(Xc, "0.0000")", "YC = " & Format(Yc, "0.0000")
Print #1, "XD = " & Format(Xd, "0.0000")", "YD = " & Format(Yd, "0.0000")
Print #1, "XE = " & Format(Xe, "0.0000")", "YE = " & Format(Ye, "0.0000")
Print #1, "XF = " & Format(Xf, "0.0000")", "YF = " & Format(Yf, "0.0000")
Print #1,
Print #1, "VORTEX FILAMENT, CONTROL POINT LOCATIONS, AND CAMBER LINE SLOPES"
Print #1, "Panel No.", "X1N(I)", "Y1N(I)", "X2N(I)", "Y2N(I)", "X(I)", "Y(I)", "dZdX(I)"
For I = 1 To N
  Print #1, Format(I, "0"), Format(X1(I), "###.000"), Format(Y1(I), "###.000"), Format(X2(I), "###.000"), Format(Y2(I), "###.000"), Format(X(I), "###.000"), Format(Y(I), "###.000"), Format(dZdX(I), "###.000")
Next I

End If

*Compute W M,N Matrix*

For K = 1 To N
  For J = 1 To N
    \[ XN1 = X(K) - X1(J) \]
    \[ YN1 = Y(K) - Y1(J) \]
    \[ XN2 = X(K) - X2(J) \]
    \[ YN2 = Y(K) - Y2(J) \]
    \[ S1 = Sqr(XN1^2 + YN1^2) \]
    \[ S2 = Sqr(XN2^2 + YN2^2) \]
    \[ T1 = 1# / (XN1 * YN2 - XN2 * YN1) \]
    \[ T2 = ((X2(J) - X1(J)) * XN1 + (Y2(J) - Y1(J)) * YN1) / S1 \]
    \[ T3 = ((X2(J) - X1(J)) * XN2 + (Y2(J) - Y1(J)) * YN2) / S2 \]
    \[ T4 = (1# + XN1 / S1) / YN1 \]
$T_5 = (1\# + XN_2 / S_2) / YN_2$

$A(K, J) = T_1 * (T_2 - T_3) - T_4 + T_5$

Next J

Next K

'Invert matrix and solve

$M = N + 2$

$KP_2 = M - 1$

$L = N - 1$

$\alpha = 0\#$

For $KA = KP_2$ To $M$

For $K = 1$ To $N$

$A(K, KA) = -(\alpha - dZdX(K))$

Next $K$

$\alpha = \alpha + 1\#$

Next $KA$

For $K = 1$ To $L$

$JJ = K$

$BIG = \text{Abs}(A(K, K))$

$KP_1 = K + 1$

'Search for largest pivot element

For $I = KP_1$ To $N$

$AB = \text{Abs}(A(I, K))$

If $(BIG - AB) < 0$ Then

$BIG = AB$

$JJ = I$

End If

Next $I$

'Decide necessity of row interchange

If $(JJ - K) <> 0$ Then

'Row interchange

For $J = K$ To $M$

$TEMP = A(JJ, J)$

$A(JJ, J) = A(K, J)$

$A(K, J) = TEMP$

Next $J$

End If

'Calculate new matrix elements

For $I = KP_1$ To $N$

$QUOT = A(I, K) / A(K, K)$

For $J = KP_1$ To $M$

$A(I, J) = A(I, J) - QUOT * A(K, J)$

Next $J$

Next $I$

For $I = KP_1$ To $N$

$A(I, K) = 0\#$

Next $I$

Next $K$

'Solve for gammas

For $KA = 1$ To $2$

$M = N + KA$

$G(N, KA) = A(N, M) / A(N, N)$

For $N_2 = 1$ To $L$

$SUM = 0\#$

$I = N - N_2$

Next $N_2$

Next $KA$
IP1 = I + 1
For J = IP1 To N
    SUM = SUM + A(I, J) * G(J, KA)
Next J
G(I, KA) = (A(I, M) - SUM) / A(I, I)
Next N2
Next KA

'Write filament strengths to file
If vlm1Check1.Value = True Then
    Print #1, "FILAMENT STRENGTHS : GAMMA = GAMMA/4*PI*UINF"
    Print #1, "GAMMA1 @ ALPHA=0.0 ; GAMMA2 @ ALPHA=1RAD"
    Print #1, "PANEL No.", "GAMMA1", "GAMMA2"
For I = 1 To N
    Print #1, I, Format(G(I, 1), "0.00000"), Format(G(I, 2), "0.00000")
Next I
End If

'Calculate span load dist.
For I = 1 To NS
    CC(I, 1) = 0#
    CL(I, 2) = 0#
    K = (I - 1) * NC + 1
    KU = I * NC
    For J = KT o KU
        CC(I, 1) = CC(I, 1) + G(J, 1)
        CL(I, 1) = CC(I, 1)
        CL(I, 2) = CL(I, 2) + G(J, 2)
    Next J
    CC(I, 1) = 8# * PL * CC(I, 1)
Next I

'Calculate CLALPHA
    CL1 = 0#
    CL2 = 0#
    For I = 1 To NS
        CL1 = CL1 + CL(I, 1)
        CL2 = CL2 + CL(I, 2)
    Next I
    CL1 = 8# * PI * B / NS * CL1 / S
    CL2 = 8# * PI * B / NS * CL2 / S
    CLA = CL2 - CL1
    ALO = -CL1 / CLA
    ALOD = ALO * RTD

'Write span load distribution at alpha = 0.0
If vlm1Check1.Value = True Then
    Print #1, "SPAN LOAD DISTRIBUTION AT ALPHA=0.0"
    Print #1, "STATION", "CC(I, 1)"'
    For I = 1 To NS
        M = (I - 1) * NC + 1
        Print #1, I, Format(CC(I, 1), "0.00000"), Format(Y(M), "0.00000"))
    Next I
'Calculate CDI
CDI = 0#
For I = 1 To NS
M = (I - 1) * NC + 1
CDJ(I) = 0#
For J = 1 To NS
K = (J - 1) * NC + 1
CDJ(I) = CDJ(I) + CC(J, 1) * (1# / (Y1(K) - Y(M)) - 1# / (Y2(K) - Y(M))
Next J
CDI = CDI + CC(I, 1) * CDJ(I)
Next I
CDI = -(CDI * B / (NS * 8# * PI * S))
CDICL = CDI * PI * AR / CL1 ^ 2
'Moment Coeff. and Center of Pressure
SL = 0#
CM1 = 0#
CM2 = 0#
For I = 1 To N
SL = SL + G(I, 1)
CM1 = CM1 + G(I, 1) * XM(I)
CM2 = CM2 + G(I, 2) * XM(I)
Next I
CP = CM1 / SL
CMA = -(CM2 - CM1) * 8# * PI * B / (NS * S * CBAR)
CM1 = -8# * PI * B / (NS * S * CROOT) * CM1
CMR = CMA * CBAR / CROOT

'Write wing parameters to file
If vlm1Check1.Value = True Then
Print #1, "Wing Area "; "; Format(S, "0.00000")
Print #1, "Wing Span "; "; Format(B, "0.00000")
Print #1, "Aspect Ratio "; "; Format(AR, "0.00000")
Print #1, "Taper Ratio "; "; Format(TR, "0.00000")
Print #1, "Cl4 Sweep "; "; Format(XCL4, "0.00000")
Print #1, "Leading Edge Sweep (degrees) "; "; Format(XLAMB, "0.00000")
Print #1, "Wing Root Chord "; "; Format(CROOT, "0.00000")
Print #1, "Mean Aerodynamic Chord "; "; Format(CBAR, "0.00000")
Print #1, "CLAlpha (per radian) "; "; Format(CLA, "0.00000")
Print #1, "CL (at Alpha = 0) "; "; Format(CL1, "0.00000")
Print #1, "Alpha L=0 (Degrees) "; "; Format(ALOD, "0.00000")
Print #1, "CDI at (Alpha=0) "; "; Format(CDI, "0.00000")
Print #1, "Normalized (CDI*pi*AR/CL^2) "; "; Format(CDICL, "0.00000")
QCR = CP / CROOT
Print #1, "CM (Alpha=0.0) "; "; Format(CM1, "0.00000")
Print #1, "CMA (per rad) "; "; Format(CMA, "0.00000")
Print #1, "XCP/ROOT "; "; Format(QCR, "0.00000")
Print #1, "Center of Pressure "; "; Format(CP, "0.00000")
Print #1, "(CMAAlpha is referenced to XA,YA)"
Print #1,
Print #1, "VORTEX LIFT EFFECTS"
End If
'Display wing geometric parameters on wing summary form
WingSummary!WingArea.Text = Format(S, "0.00000")
WingSummary!WingSpan.Text = Format(B, "0.00000")
WingSummary!AspectRatio.Text = Format(AR, "0.00000")
WingSummary!TaperRatio.Text = Format(TR, "0.00000")
WingSummary!Sweep.Text = Format(XLAMB, "0.00000")
WingSummary!RootChord.Text = Format(CROOT, "0.00000")
WingSummary!MeanAeroChord = Format(CBAR, "0.00000")
WingSummary!text12 = Format(XCL4, "0.00000")

'Display wing aerodynamic parameters on wing summary form
WingSummary!Text1.Text = Format(CLA, "0.00000")
WingSummary!Text2.Text = Format(CL1, "0.00000")
WingSummary!Text3.Text = Format(ALOD, "0.00000")
WingSummary!Text4.Text = Format(CDI, "0.00000")
WingSummary!Text5.Text = Format(CDICL, "0.00000")
WingSummary!Text6.Text = Format(CM1, "0.00000")
WingSummary!Text7.Text = Format(QCR, "0.00000")
WingSummary!CenterOfPressure.Text = Format(CP, "0.00000")
WingSummary!MomentCoeff.Text = Format(CMA, "0.00000")

'Vortex lift effects
XKP = CLA
XKVLE = (XKP - XKP ^ 2 * (CDICL / (AR * PI))) / Cos(XLAMB / RTD)

If vlm1!Check1.Value = True Then
    Print #1, "KP="; Format(XKP, "0.00000"), "KVLE = "; Format(XKVLE, "0.00000")
    Print #1, " ANGLE"
    Print #1, "Degrees Radians", "CLPOT", "CLV", "CDIPOT", "CDIV", "CL", "CDvortex", "CD", ", CL/CD"
    Print #1,
End If

alphamax = 0
ldmax = 0
For I = 1 To 30
    AD(I) = I / RTD
    D(I) = AD(I) - ALO
    CLPOT = XKP * Sin(D(I)) * Cos(D(I)) ^ 2
    CLV = XKVLE * Cos(D(I)) * Sin(D(I)) ^ 2
    CDIP = XKP * (Sin(D(I))) ^ 2 * Cos(D(I))
    CDIV = XKVLE * (Sin(D(I))) ^ 2 * Cos(D(I))
    CLT = CLPOT + CLV
    CLL(I) = CLPOT + CLV
    CDT = CDIP + CDIV
    CD(I) = CDIP + CDIV
    CD0 = .022 * (.725 + .275 * (200 / S))
    CDD(I) = CDIP + CDIV + CD0
    LD = CLT / CDT
    LoD(I) = CLT / CDD(I)
'Find alpha for CL/CD max
If LoD(I) > lodmax Then
    lodmax = LoD(I)
    alphmax = I
End If

If vlm1ICheck1.Value = True Then
    Print #1, I, Format(AD(I), "0.0000"), Format(CL POT, "0.0000"), Format(CL V, "0.0000"),
    Format(CDIP, "0.0000"), Format(CDIV, "0.0000"), Format(CL T, "0.0000"), Format(CDT, "0.0000"),
    Format(CDD(I), "0.0000"), Format(LoD(I), "0.0000")
End If

Next I

WingSummary!AlphaMax.Text = alphmax

'Write CLT and CLD to the WingSummary screen
I = Val(vlm1IAttackAngle.Text)
AD(I) = I / RTD
D(I) = AD(I) - ALO
CLPOT = XK P * Sin(D(I)) * Cos(D(I)) ^ 2
CLV = XKVLE * Cos(D(I)) * Sin(D(I)) ^ 2
CDIP = XK P * (Sin(D(I)) ^ 2 + Cos(D(I))
CDIV = XKVLE * (Sin(D(I))) ^ 3
CLT = CLPOT + CLV
CD0 = .022 * (.725 + .275 * (200 / S))
CDT = CDIP + CDIV + CD0
LD = CLT / CDT
WingSummary!AttackAngle.Text = I
WingSummary!CL.Text = Format(CL T, "0.000")
WingSummary!CD.Text = Format(CDT, "0.000")
WingSummary!LoverD.Text = Format(LoD(I), "0.000")

'Calculate required takeoff thrust
RhoSL = .0023769 'slugs/ft^3
VelocityTO = 245 'ft/sec
ThrustTO = .5 * RhoSL * VelocityTO ^ 2 * S * CDD(5) 'Takeoff at alpha = 5°
WingSummary!TakeoffThrusl.Text = Format(ThrustTO, "0,000.##")
WingSummary!CDo.Text = Format(CDO, "0.000")
SelectEngine!TakeoffThrust.Text = Format(ThrustTO, "0000.##")

'Write CL, CD, and CL/CD to the graph
WingSummary!CoeffGraph.Autolnc = False
WingSummary!Graph1_Autolnc = False
For I = 1 To 3
    WingSummary!CoeffGraph.ThisPoint = I
    WingSummary!CoeffGraph.ThisSet = I
    For J = 1 To 30
        WingSummary!CoeffGraph.ThisPoint = J
        WingSummary!CoeffGraph.XPosData = J
    Next J
Select Case I
    Case 1
        WingSummary!CoeffGraph.GraphData = CD(J)
    Case 2
WingSummary1CoeffGraph.GraphData = CLL(J)

Case 3
   WingSummary1CoeffGraph.GraphData = LoD(J) / 6
End Select

Next J
Next I

WingSummary1Graph1.ThisSet = 1
For I = 1 To 30
   WingSummary1Graph1.ThisPoint = I
   WingSummary1Graph1.XPosData = I
   WingSummary1Graph1.GraphData = LoD(I)
Next I

WingSummary1CoeffGraph.DrawMode = 2
WingSummary1Graph1.DrawMode = 2

' WRITE SOME PARAMETERS TO AN ONSCREEN GRID TO TEST THE VLM CODE
For I = 0 To 2
   vlm1IGrid1.ColWidth(I) = vlm1IGrid1.Width / 3
Next I

vlm1IGrid1.Row = 0
vlm1IGrid1.Col = 0
vlm1IGrid1.Text = "Degrees"
vlm1IGrid1.Col = 1
vlm1IGrid1.Text = "CL"
vlm1IGrid1.Col = 2
vlm1IGrid1.Text = "CDvortex"
vlm1IGrid1.Col = 3
vlm1IGrid1.Text = "CL/CD"

For I = 1 To 30
   vlm1IGrid1.Row = I
   AD(I) = I / RTD
   D(I) = AD(I) - ALO
   CLPOT = XKP * Sin(D(I)) * Cos(D(I)) ^ 2
   CLV = XKVLE * Cos(D(I)) * Sin(D(I)) ^ 2
   CDIP = XKP * (Sin(D(I))) ^ 2 * Cos(D(I))
   CDIV = XKVLE * (Sin(D(I))) ^ 3
   CLT = CLPOT + CLV
   CDT = CDIP + CDIV
   vlm1IGrid1.Col = 0
   vlm1IGrid1.Text = I
   vlm1IGrid1.Col = 1
   vlm1IGrid1.Text = Format(CLT, "0.000")
   vlm1IGrid1.Col = 2
   vlm1IGrid1.Text = Format(CDT, "0.000")
   vlm1IGrid1.Col = 3
   vlm1IGrid1.Text = Format(LoD(I), "0.000")
Next I

If vlm1!Check1.Value = True Then
    Close #1
End If

WingSummary.Show 1

ErrHandler:
vlm1!Label13.Caption = "An error has occurred in the Vortex-Lattice Code Module."
Exit Sub

End Sub
December 15, 1994

Dr. Ann Boucher  
Director - Honors Program  
University of Alabama in Huntsville  
Morton Hall  
Huntsville, AL 35899

Dear Dr. Boucher:

This is to inform you that U.A.H. will become the licensed user of Visual Basic 3.0 upon delivery of the original software package to the university. The Visual Basic package accompanies this letter. Upon receipt, the license is officially transferred, providing the university accepts the terms of the license. This acceptance of the license terms is assumed upon your acceptance of the package.

Sincerely,

Eric G. Woodfin

Eric G. Woodfin