13th Australasian Fluid Mechanics Conference
Monash University, Melbourne, Australia
13-18 December 1998

ICARUS - A RAPID SOLUTION FOR AERODYNAMIC ANALYSIS

Richard POOLE and John MILTHORPE

School of Aerospace and Mechanical Engineering
Australian Defence Force Academy, Canberra, Australian Capital Territory, AUSTRALIA

ABSTRACT

Icarus is an aerodynamic analysis program aimed at providing rapid solution to problems involving three-dimensional lifting surfaces. A rapid solution is obtained through the use of a simple analysis method, as well as an easy to use graphical interface. Icarus is particularly intended to help teach undergraduate students of aerodynamics, who need access to an inexpensive and simple analysis tool. Undergraduate students possess a wide range of computing equipment of no particular standard, and so easy portability across platforms is essential.

The program uses the Java programming language to enable cross platform portability and make use of the Internet orientated features of the language. A lifting surface method with discrete vortex rings placed along the wing camber line is implemented. This allows for analysis of wings with low aspect ratio, sweep and camber, as well as systems of multiply wings. This provides a large range of applicability, without sacrificing speed or simplicity.

The user interface makes use of data tables, dialog boxes and wireframe views to allow for expeditious design of the wing system to be analysed. It also makes use of graphs and tabulated data to readily analyse the data obtained in the solution of the system.

Icarus was validated against a range of analytic, numerical and experimental results and shown to produce accurate results for planar wings of various sweep and aspect ratio. It also produces accurate results for non-planar wings and tandem wing systems.

INTRODUCTION

The rapid increase in the performance of computers over the last decade or two has seen a corresponding increase in the complexity and accuracy of modern CFD codes (Wendt, 1996). However, whilst computers are increasing in power and performance, the average time to generate a solution to CFD codes has remained reasonably constant (Kroo, 1992).

Current software generally uses complex, highly accurate methods that are often unnecessarily complex, accurate and specific for many uses. Such software is also expensive to purchase and requires a high level of knowledge and skill to operate effectively. There is a lack of programs suitable for use in solving three-dimensional aerodynamic problems quickly and easily (Poole, 1997).

Software that provides a rapid solution at design time allows for more extensive optimisation to be carried out, resulting in a better initial design. This is desirable as life cycle costs are often locked in early in the design stage (Stephen and Schirmer, 1990). Also, during initial design, much of the detail of the design is unknown and more comprehensive analysis would be futile (Kroo, 1992). Easy-to-use aerodynamic analysis software also provides considerable opportunity for people without experience with aerodynamic design to investigate the impact of wing geometry on aerodynamic performance.

OVERVIEW OF ICARUS

Icarus was developed to provide a cheap, easy to use alternative to current software for the modelling of three-dimensional wings and lifting surfaces. It provides a simple graphical user interface and rapid solution of problems and is ideal for initial design and educational uses.

Icarus uses a vortex lattice method that models a wing by placing discrete vortex ring elements along the camberline of the wing. The use of vortex ring elements allows definition of quadrilateral panels and hence the effects of camber and twist can be more easily modelled than with rectilinear panels (Katz and Plotkin, 1991).

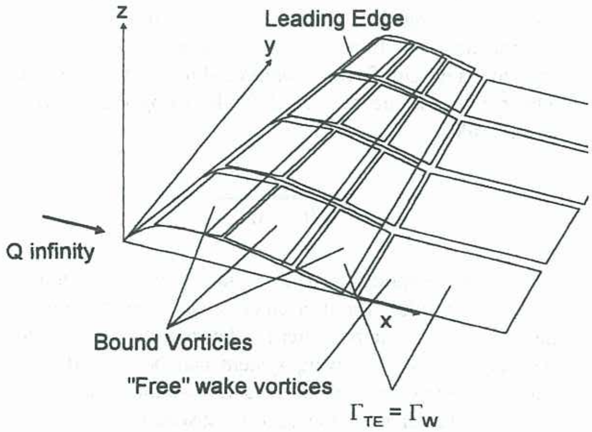

Figure 1: Vortex Ring model of a thin wing

The vortex rings are placed on the quarter chord line of each panel and continue through to the quarter chord of the following panel (Figure 1). The collocation point is defined at the center of the vortex ring, as well as the panel normal vector. Hough (1973) has shown that placing the vortex at the quarter chord produces the most accurate results and also satisfies the two-dimensional Kutta condition. The three-dimensional trailing edge condition is satisfied by setting the strength of the wake panels to be equal to the strength of the shedding panel at the trailing edge.

The wake is modelled as a linear wake with a single vortex ring extending downstream of the wing. The wake leaves the trailing edge at the angle of local camber. According to Bertin and Smith (1979), modelling the wake as leaving the trailing edge parallel to the free stream, or parallel to the body at the trailing edge will result in similar accuracy, within the assumptions of linearised theory. Having the wake leaving the trailing edge parallel to the body was chosen as it reduces computation time when considering a range of angles of attack.

The influence of all of the vortex rings on each collocation point is used to create a set of simultaneous equations that are solved by satisfying the Neumann boundary condition in which there is zero normal flow across the solid surface.

$$\nabla(\Phi + \Phi_\infty) \cdot \mathbf{n} = 0$$

The resultant circulation at each panel is used to calculate the local velocity at each panel, which is then used to calculate the force on each panel, employing the Kutta-Joukowski theorem of lift:

$$\mathbf{F} = \rho \mathbf{q} \times \Gamma$$

Once the force at each panel has been calculated, it can be summed over the whole wing to produce the lift, drag and side-force for the wing. The moment produced by each panel can also be summed to give an overall moment for the wing.

Icarus has the ability to model wings with low aspect ratio, sweep, twist, camber and systems of multiple wings. It can also be used to calculate side forces and moments on asymmetric wing systems or wings in sideslip. The model is applicable for thin wings (less than 15% thick) in subsonic, inviscid flow. Compressibility effects in the subsonic regime are accounted for by using the Prandtl-Glauert rule:

$$C_p = \frac{C_{p,0}}{\sqrt{1 - M_\infty^2}}$$

Icarus uses a graphical interface to facilitate user input and analysis of results. The user enters a new wing through a dialog box or by simply entering the required values into the data display. The wing system can be viewed in a standard 3-view diagram or as a three-dimensional wire diagram. This enables the user to ensure that the wing geometry has been entered correctly. Once the geometry has been adequately defined, a solution is generated for the system.

Once a solution is generated, the user can access data about the wing or wing system such as the spanwise lift distribution, coefficient of lift, coefficient of drag, sideforce and moments about the x, y and z axes. The data can be analysed graphically or converted to text format for transfer to other programs such as Microsoft Excel or MATLAB.

JAVA

Icarus is written using the Java programming language developed by Sun. Java is an object-orientated programming language that is portable, secure, robust and platform independent (Van der Linden, 1997). This makes Java ideal for use across the Internet, with its many different platform types and security problems.

The use of Java allows Icarus to be deployed across the Internet. This solves a lot of the problems associated with distributing software. It is of particular benefit to students who have ready access to the Internet but might not be able to afford to pay for a copy of the software.

Cross platform portability allows everyone to run the same code, no matter what system they use. This saves on development time as the software only has to be written once and it will run on any platform type. Platform independence is achieved by compiling the source code into byte code. This byte code is then converted to machine code at execution time. This requires a run-time compiler that is platform dependent, however, once the run-time compiler is installed, any Java program will be able to be executed.

Java has many other features that assist in development and execution of programs. It provides automatic garbage collection in which memory is allocated and reclaimed dynamically, preventing troublesome memory leaks. Java also provides run-time array bounds checking, to ensure that array sizes are not violated, resulting in lost and corrupt data.

All these features of Java come at a slight cost. The added overhead in garbage collection, run-time compilation and array bounds checking result in Java running slightly slower than a comparable language without these features (such as C++). Current run-time compilers have Java running 20-40 percent slower than equivalent C++ code, but the next generation of compilers have the potential to run faster than equivalent C++ code (van Hoff, 1998).

The security features implemented in Java to prevent rogue programs from disrupting systems have resulted in some restrictions whilst operating over the Internet. Programs executed via the Internet cannot access local hardware such as drives and printers and therefore the program cannot read, write or access files on the local system and the cannot print information on a local printer. These problems can be avoided by executing the program as a stand-alone application, isolated from the Internet.

VALIDATION OF ICARUS

The role of validation in the development of analysis software is to ensure that the software will accurately represent real world problems. This often requires comparisons between the results generated by the software and results obtained from real world observation. Sometimes this comparison is not possible, as the data is not available or is inaccurate, due to the limitations of the environment. In these cases comparisons must be made to theoretical calculations or previously validated programs (Marvin, 1995). Validation against experimental results will ensure that the right method has been implemented for solving the particular problem, whilst validation against other programs and theoretical results will ensure that the method is being solved correctly.

Several series of tests were conducted in the validation of Icarus. The tests were designed to investigate the performance of Icarus starting with simple, single wing configurations and working towards more complicated multiple wing systems.

The first series of test carried out compared the results for spanwise lift distribution generated by Icarus, to those calculated by Falkner (1948). This series examined the effects of changes to the wing geometry, such as wing sweep and aspect ratio. Falkner uses a discrete vortex lattice method to calculate a series of polynomials to describe the spanwise distribution. The method used by Falkner is similar to that developed by Multhopp (Jones and Cohen, 1957) and is often used for comparisons of spanwise lift distributions (Katz, 1985).

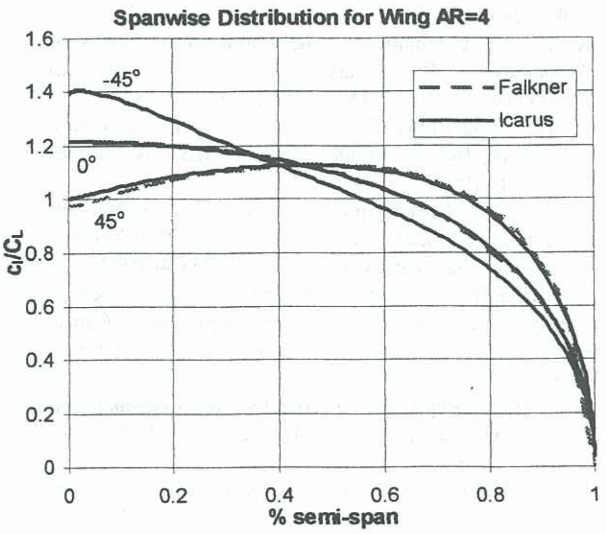

Figure 2: Comparison of spanwise lift distribution between Icarus and Falkner

Figure 2 shows one of the test cases comparing the spanwise lift distribution of wings with an aspect ratio of four. The abscissa is the percent semi-span of the wing and the ordinate is the ratio of the local lift coefficient to the total lift coefficient of the wing. Icarus compares very well with the results obtained by Falkner, the largest error being less than 3 percent. Icarus used a 5 by 100 panel grid while Falkner used 126 vortices distributed evenly over the semi-span for his solution.

The second series of tests were designed to evaluate the performance of Icarus with non-planar configurations. One of the test cases involved calculating the lift curve slope for wings with various dihedral. The results were then compared to the results obtained by Kalman, Rodden and Giesing (1971) using a doublet-lattice method. Figure 3 shows that Icarus achieved excellent correlation with Kalman, Rodden and Giesing (1971).

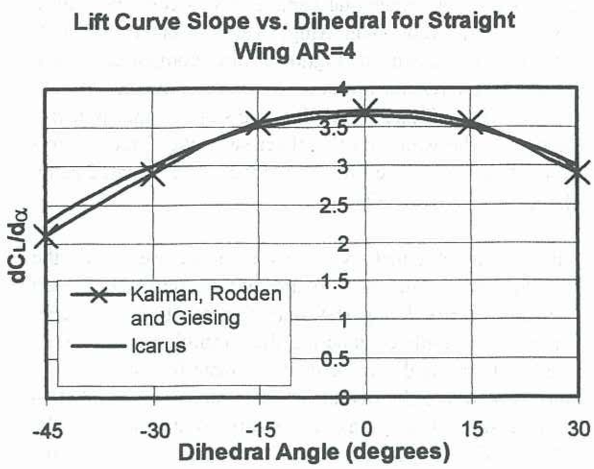

Figure 3: Comparison of the effect of dihedral on the lift curve slope of a straight wing

The third series of tests evaluated Icarus in the use of multiple wing systems. Figure 4 shows the results obtained with a tandem wing system. The ordinate is the lift curve slope of the wing and the abscissa is the ratio of the separation distance to the span of the wing. Both wings in the tandem system have the same span, aspect ratio and incidence with Wing 2 being the rear wing. The comparison is made with the experimental results presented by Hoerner (1965).

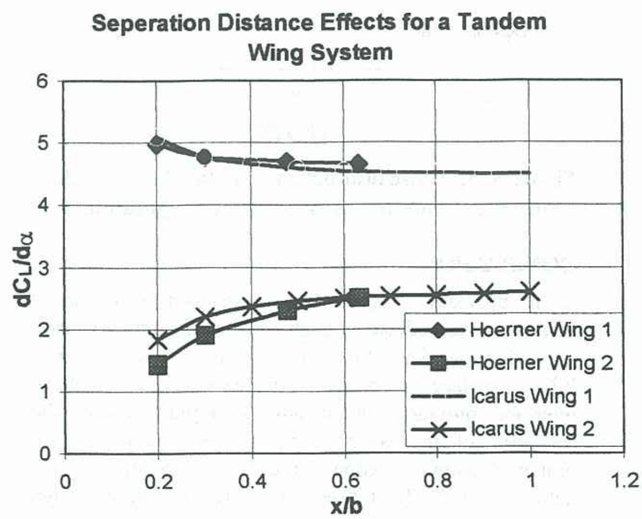

Figure 4: Separation effects in a tandem wing system

Wing 1 shows excellent correlation to the results of Hoerner (1965), however, Wing 2 has an error of 20 percent at x/b = 0.2 which decreases with increasing x/b. This error is due to the effects of the wake shed from the trailing edge of Wing 1. Icarus uses a simple, linear wake model to reduce computation time but in doing so, reduces the accuracy of wake effects.

A further series of tests were conducted to confirm the ability of Icarus to model multiply wing systems. This series of tests analysed a canard-wing system. The system comprised of a rectangular canard, AR=4 coplanar with a 25° forward-swept main wing, AR=5.7, taper ratio 0.4. The results shown in Figure 5 are compared to the experimental results obtained by Lombardi and Morelli (1994). The ordinate is the lift curve slope of the spanwise section of the wing and the abscissa is the percent semi-span of the wing. The lift curve slope was calculated using the results for $\alpha=0$ and $\alpha=4$.

The results obtained by Icarus compare well with the experimental results of Lombardi and Morelli (1994) with the largest error being less than 5 percent. A 5 percent error is acceptable considering the simplified wake model used by Icarus and is more than adequate for initial design purposes, where the overall trend is more important than exact results. Icarus has the ability to demonstrate the effects of the interference of a canard on the lift distribution of the main wing.

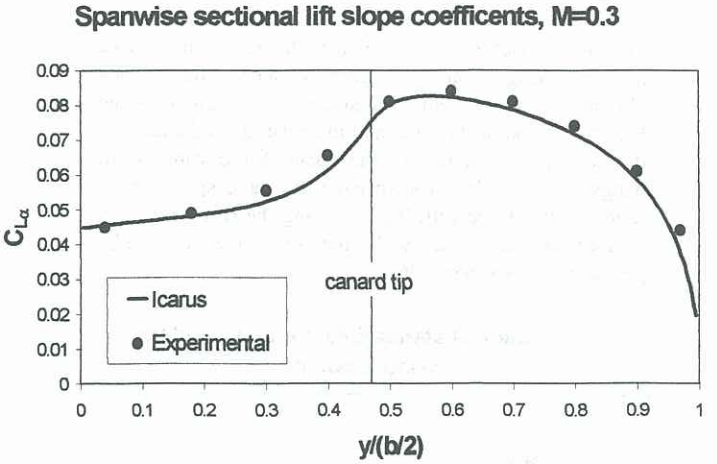

Figure 5: Spanwise distribution of section lift curve slope on the main wing in a canard-wing configuration

CONCLUSION

Icarus provides a rapid, easy to use alternative current software for aerodynamic analysis. It uses a lifting surface method to achieve fast solution to three-dimensional lifting surface problems. An easy-to-use graphical interface provides the means to rapidly create the geometry of the wing to be analysed and provides graphical output to allow for easy interpretation of the data generated. This makes Icarus ideal for initial design analysis and educational use.

The use of Java means that Icarus can be executed on any platform anywhere, via the World Wide Web. This solves a lot of distribution and compatibility problems and negates the need to re-write code and set up expensive distribution networks. It also provides university departments and students access to aerodynamic analysis software that they might not otherwise be able to afford.

Validation was carried out to ensure that the results being produced by Icarus are accurate. This involved comparisons with results obtained from a range of analytic, numerical and experimental results. The validation showed that Icarus is capable of modelling planar and non-planar wing configurations as well as systems of multiple wings.

REFERENCES

BERTIN, J and SMITH, M: *Aerodynamics for Engineers*, Prentice-Hall, New Jersey, 1979.

FALKNER, V. M.: "Calculated Loadings due to Incidence of a Number of Straight and Swept-back Wings", *Aeronautical Research Council Reports and Memoranda*, No. 2596, June 1948.

HOERNER, S. F.: *Fluid-Dynamic Drag*, Hoerner Fluid Dynamics, New Jersey, 1965.

HOUGH, G.R.: "Remarks on Vortex Lattice Methods", *Journal of Aircraft*, Vol. 10, May 1973.

JONES, R and COHEN, D: "Aerodynamics of Wings at High Speeds", *High Speed Aerodynamics and Jet Propulsion*, Vol. 7, Oxford University Press, London, 1957.

KALMAN, T.P.; RODDEN W.P. and GIESING J.P.: "Application of the Doublet-Lattice Method to Nonplanar Configurations in Subsonic Flow", *Journal of Aircraft*, Vol. 8, June 1971.

KATZ, J.: "Calculation of the Aerodynamic Forces on Automotive Lifting Surfaces", *Journal of Fluids Engineering*, Vol. 107, December 1985.

KATZ, J and PLOTKIN, A: *Low-Speed Aerodynamics: From Wing Theory to Panel Methods*, McGraw-Hill Inc., New York, 1991.

KROO, I: "Aerodynamic Analyses for Design and Education", *AIAA 10th Applied Aerodynamics Conference*, Palo Alto, California, June 22-24, 1992.

LOMBARDI, G and MORELLI, M: "Pressure Measurements on a Forward-Swept Wing-Canard Configuration", Journal of Aircraft, Vol. 31, February 1994.

MARVIN, Joseph G.: "Perspective on Computational Fluid Dynamics Validation", *AIAA Journal*, Volume 33, No. 10, October 1995.

POOLE, R.A.: *Development of Aerodynamic Analysis Software*, Thesis, UNSW, 1997.

STEPHEN, A and SCHIRMER, M: "Logistics: What Does It Have To Do With Maintenance", *1990 International Maintenance Management Conference*, Melbourne, Australia, August 20-22, 1990.

VAN DER LINDEN, P: *just Java 2nd ed.*, Prentice Hall, New Jersey, 1997.

VAN HOFF, A: "Java: Getting Down to Business", *Dr Dobbs Journal*, Vol. 23, Issue 1, January 1998.

WENDT (Ed.): *Computational Fluid Dynamics 2nd ed.*, Springer-Verlag, Berlin, 1996.