A CAD INTERFACE FOR THE FAST DESIGN OF GLIDER GEOMETRIES

By P.W. Jordaan and A.S. Jonker

Presented at the XXVI OSTIV Congress, Bayreauth Germany

ABSTRACT

This paper deals with the development of a dedicated software code running in CAD software to create sailplane geometries using the basic dimensions and airfoil data. The ease of changing input data makes it an ideal development tool. To assist Flow Analysts to obtain the correct information of the intended design in a format suitable for their analysis programs, data files describing the entire sailplane or sections thereof can be produced. These analysis programs have very powerful computational capabilities to solve distribution fields relating to flow, pressure, stress and temperature but lacks the ability to quickly and accurately enter the data of complex surfaces and profiles, typically used in sailplanes. Output data can be generated to reduce the grid generation time for analysis programs. The generated 3D full scale model can also be used to create detail manufacturing and layout drawings. Computer aided manufacturing data can be extracted to create moulds required for manufacturing of complex sailplane surfaces.

INTRODUCTION

There are many ways to solve a problem but solving the problem of optimizing an aerodynamic design of a modern sailplane, is by no means a simple task. Many methods have been developed in the past, each with its own powerful attributes and limitations. Developments in mathematical procedures, computer hardware and analysis software have aided many powerful longstanding problems. Many three dimensional (3D) flow problems however, still need investigation but requires a lot of effort notwithstanding current modern design facilities.

The most commonly used computational technique in aerodynamic design is the Panel method. This method gives good results where viscous and compressibility effects can be ignored. With more powerful computers, it is possible to solve the averaged Navier Stokes equations which gives better results where viscous and compressibility effect are present. They require, however, much more computational effort to solve and the computational grid must be developed for the complete three dimensional flow field. This is normally very time consuming especially if a complex shape like a sailplane is to be modeled.

Sailplane geometry is a collection of curved and warped surfaces generated by complex curves. To build a structured computational 3D grid of node points (vertices) and splines to fully describe the object and its environmnt, is a tedious task to be performed accurately. The set of data generated holds true for only one geometric layout and has to be repeated for every combination in the resarch analysis.

The purpose of this paper is to present a CAD interface that can calculate sailplane geometries and produce the required grid points to be used in computational flow techniques and for manufacturing.

THE CAD PLATFORM

The Computer Aided Drafting software of the 70s has developed into powerful Computer Aided Design (CAD) tools in recent years. The original motivation was to use the repeatability and accuracy of the computer to perform some of the tedious tasks of the draftsman. The ease of modifying and copying line drawings was some of the biggest benefits of using CAD. As more complex mathematical matrix operations were introduced into software, more complex data manipulations became possible. To make it possible to understand the computers mind, powerful display routines were developed to visualize 3D virtual reality objcts. The development of more powerful hardware stimulated the CAD software designers to always go one step further. Faster processors, more RAM (memory) faster high resolution big screens and laser and color printers, set the scene for current day CAD software to become the center of the design process. The designers, from the idea stage to final approval, along with the flow and stress analysts, tooling designers, production planners and parts manufacturing, financial controllers and marketing, all now connect to the main CAD database. Each department enters its data into the database and draws the necessary information from it. The evolution of CAD made concurrent engineering a reality.

A good quality CAD system also makes it possible for end users to access its powerful routines. Dedicated software can thus be developed which uses the CAD software as a platform to produce designs. The main output of a CAD system is the display on the screen and a hard copy printout. The data generated internally by the CAD software however, is also accessible to the user and can be used to perform manipulations and to create printout or data files. These data files can be accessed by the analysts to be used in flow or stress software.

In creating this CAD interface, Cadkey is used as the CAD platform. Cadkey has a powerful programming language called CADL (Cadkey Advanced Design Language) within the structure of the Cadkey program. The flexibility of this language allows a user to design and store entities (geometric items) within Cadkey, as well as access the Cadkey database. It contains typical computer language elements like register variables, data types, variable declarations, format specifiers, mathematical functions and program control statements. It has a compiler to make the program a Cadkey executable file which can be called from Cadkey.

THE INTERFACE PROGRAM

The purpose of the interface is to allow a designer to generate a complete sailplane geometry and to make changes to it very easily. The geometry model can then be used as the basis from which the data rquired for the generation of a computational flow grid can be developed. Ilso be used to generate the detail sailplane design from which manufacturing drawings can be generated and eventually, the toolpaths for the manufacturing of the molds can be created from this data.

The interface program performs the following tasks:

1. Read the sailplane's basic dimensions and profile specifiers from input files.
2. Validate the data.
3. Generate the sailplane geometry.
4. Generate the required output data.

THE INPUT DATA

The input data is read from input files. There are three types of input files. These are:

1. The master input file.
2. Airfoil data files.
3. Fuselage data file.

The master input file contains all the geometry data for the sailplane like wing data, fuselage data, tail data and winglet data. The data is typically organized in rows where a single row will contain all the data of a single entity like a wing subsection. The data contained in a row is the airfoil file names for the planform ends, and the dimensions like the chords, span, dihedral and twist. A wing can be constructed using many wing subsection (planforms). Each planform can have its own starting and ending profiles, twist, sweep and dihedral. It is thus possible to model any shape such as a polyhedral wing as used on the Discus 2.

If two sucessive planforms have different dihedrals, a wing fillet is constructed from the tip of the first planform to the root of the next. To position the second and subsequent planforms, an axis has to be specified around which to rotate the next planform. This axis is specified as cord units above or below the leading and trailing edge of the root and tip. A reference point has to be specified on the axis, that will align the two planforms. A smooth transition fillet is constructed between the two planforms. The transition between the wing tip and a winglet with a smaller cord length and a very large dihedral can be constructed this way, as shown in Figure 1.

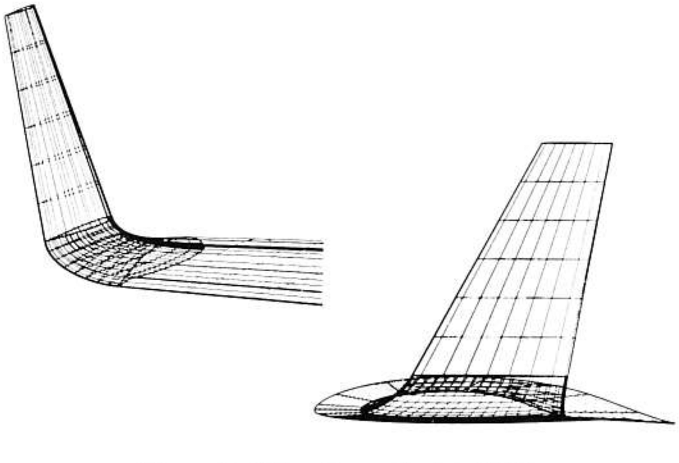

Figure 1. Wingtip - winglet fillet.

The airfoil data files contains the airfoil co-ordinate data. The same format as produced by XFOIL is used which is a x and y co-ordinate with the co-ordinates running from the trailing edge of the airfoil over the top surface, round the leading edge and back to the trailing edge.

The fuselage data file contains the co-ordinate data for the fuselage. Different standards exist for presenting fuselage in co-ordinate format. The elliptical fuselage profile data is entered as a data set x, h, v, y and t starting from x = 0.0 at the nose to the maximum x at the tail end. The horizontal h-data describes the major and vertical v-data the minor axes of the elliptical profile. The center of the profile can be offset by the y-data above or below the x-axis. The t-data ındicates the angle of the profile in the x-y plane.

The wing-fuselage fillet profile follows a simlar procedure as for the planforms. The root profile of the fillet will be projected from the wing root of the first planform to the fuselage. The fuselage fillet tip profile is a x-z profile to create a curve on the planform to indicate the tangency line of the fuselage fillet on the wing. If this data is not supplied, the fillet stops at the root of the planform. Fillet calculations will create smooth transitions with these two profiles as the lines of tangency with the fuselage and wing respectively as shown in Figure 2.

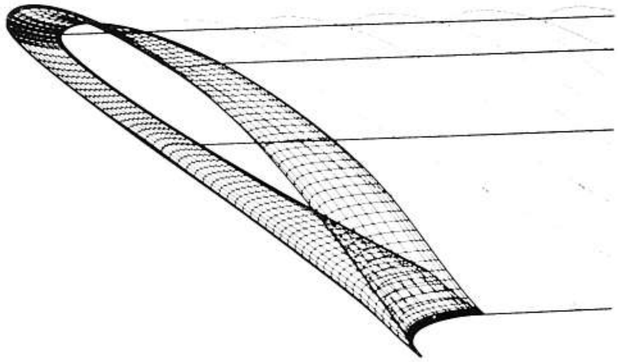

Figure 2. Fuselage - Wing fillet

DATA VALIDATION

The data have to be validated in order to perform a successful calculation and generate a sailplane geometry. This section makes sure the orientation of the profile is to the user's specification. If the orientation is incorrect, successive mirror transformations are performed until the correct orientation is observed and accepted. The location data of wing and fillet profiles have to make generation of the body possible before further calculations are performed. If the size or location of a profile cannot be projected to the fuselage, calculations will not commence.

GEOMETRY GENERATION

The profile data is converted to a standard spline with a set number of 41 node points. Node numbering starts at 0 at the trailing edge and continues around the top to the leading edge at node 70. Numbering continues from the leading edge along the bottom to the trailing edge. This procedure is required to ensure proper alignment of the different profiles on the fuselage stations and those of the different planforms (wing panels). The node points are arranged more closely spaced on the profile to ensure good definition at the top and bottom of the fuselage an in the leading edge nose area and the trailing edge tip of the airfoils.

The fuselage is calculated and the station splines created. The station splines are then connected with longeron splines. This procedure creates a mesh of 3D splines which fully describes the fuselage.

The wing root profile is created using the converted standard spline. This spline is scaled to the root cord length. The tip profile is scaled to the tip cord length and is positioned on the root profile with the refrence points of each profile aligning. The tip will then be rotated through twist, repositioned in the z-direction to the span distance and moved to the new sweep position. The planform (root and tip profiles) can now be rotated through the dihedral around the x-axis of the root profile. Finally, the first planform is repositioned to a position refrenced to the fuselage. Subsequent planforms will be repositioned to a position referenced to the previous planform and aligning the specified two axes and reference points.

The fuselage fillet is generated by following the same steps as for the wing sections and can be of different airfoil data. The fillet profile will be positioned referenced to the root profile of the first planform. The 2D profile is then projected to the fuselage to create the 3D fuselage tangency profile. The tip of the fuselage fillet is created by using the root profile of the first planform. If a fuselage fillet tip profile is provided, the profile is projected onto the wing section to form the 3D wing tangency profile. 3D spanwise splines are generated tangent at each node point of the fuselage fillet root on the fuselage and the fillet tip on the wing. Each of these created spanwise splines are used to create cord-wise splines to fully describe the transition surface.

At this stage the wing fillets are generated if the dihedral of the planforms differ. 3D splines are generated tangent to each node point of the tip and subsequent wing's root. Each of these created spanwise splines are used to create cordwise splines. Although the spanwise splines are complex 3D spins, the cordwise splines are 2D profiles.

Similar procedures as for creating the wing geometry are followed for creating the tail geometries.

The complete calculation is performed in less than 20 seconds leaving a full scale 3D model of the entire sailplane in the Cadkey database. Any further development can be performed from this stage. Changes to any of the geometry of the sailplane, like a different airfoil profile or planform dihedral, requires only a few seconds. Figure 3 shows a complete sailplane geometry generated by the interface program.

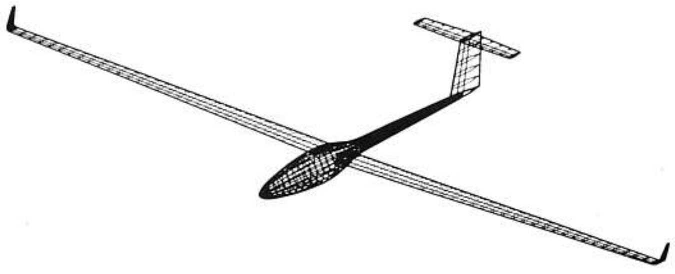

Figure 3. Generated sailplane geometry.

FLOW FIELD GRID GENERATION

The flow analysis programs lack the powerful geometric generating capabilities of a good CAD system. After the sailplane's geometry has been generated, or parts thereof as specified, the block data to be used for flow analysis can be created.

To develop an interface one has to understand the two fields to be interfaced. This paper will not deal with all the intricacies of grid generation, as the requirement for grid points will be determined by the analyst. The grid points make up the cells within a block. These cells are used to perform the flow calculations. The spacing of cells can be linear, exponential, etc. to increase the number of cells in the region where the geometry might dictate more accurate calculations.

Two approaches were however considered for the interface i.e. a structured block and a body fitted block configuration.

For the structured block approach the flow field is divided into blocks that mesh to each other. The number of node points making up the cells in a block of connecting blocks have to be the same. Boundary cells of blocks adjacent to an object with complex contours, can however, be formed that describe the flow around the contour of the body although the cells does not describe the contour of the body. See block marked x in figure 4. This could lead to inaccuracies and unwanted flow behaviour.

For the body fitted approach, blocks are so formed (or deformed) that the boundary cells of the blocks follow the contour of the body exactly. One face of each boundary cell makes up the face of the body and avoids the problem of the structured blocks in this area. See Figure 5.

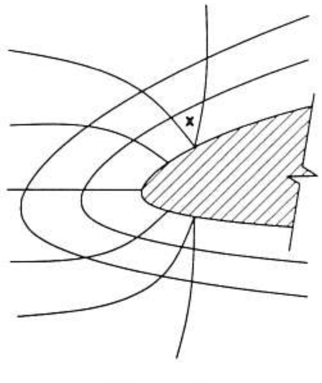

Figure 4.
Structured block

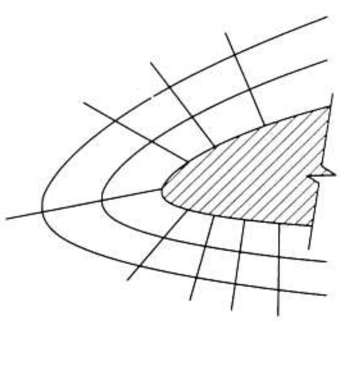

Figure 5.
Body fitted block

The data describing the body can also be imported into most flow analysis software using IGES. The analysts then have to construct the blocks around the body and generate cells within each block to satisfy the flow computational requirements. The building process, once performed, can be programmed and stored and used repeatedly on different runs, provided the data is compatible with the original building procedure.

However, it is a much faster procedure if the required data can be imported directly as data files with all the required block data already created. A minimum of intervention is required to start the following flow calculation using different sailplane geometries. Many more flow analysis runs can be performed in a specific time this way.

The grid data comprises of a vertex list and a spline list. The spline list contains the spline number, the number of vertices making up the spline and a list of the vertex numbers. The vertex list contains the vertex number and the x,y and z co-cordinates of the vertex. The data describing the body can thus be imported as text files and the body contours recreated in the flow program. A spline is a line passing through a number of points. In cubic paramtric splines, two points will normally describe a straight line.

Additional data like patch data and block data can also be used to further describe the blocks that are required to describe the flow field. Patch data comprises a patch number and the four splines making up the edges. Block data contains a block number and the eight node points (vertics) making up the block. The order of the vertex numbers for a block will describe the block. These block vertices have to be added to the vertex list. Some flow programs will then recreate the splines between the block vertices to create the block.

To be able to create block data it is required to know the rationale in creating the vertex points of a block. If a block has to follow the contour of the body, at what distance must the boundary be from the object? The relationship between these points have to be established in order to create new block vertices, should the contour of the body change, e.g. a different wing profile. The outside boundary of blocks could be fixed points at a set distance from the body which will not be influenced by changes in body geometry. These relationships have to be entered into the interface program. An example of a structured block generated by the interface is shown in Figure 6.

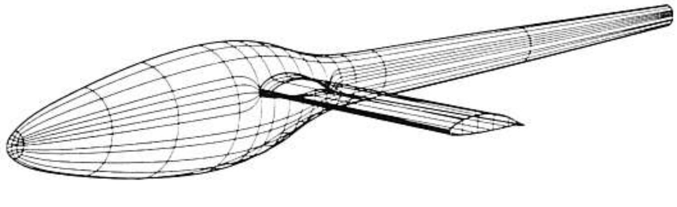

Figure 6. Structured block generated by the interface.

In the interface program the type of block data to be generated has to be specified. The file output type has to be specified as StarCD or Flow++. The output be created to the format as required by these analysis programs. Block and panel data only applies to Flow++ data files.

MANUFACTURING

The manufacturing process starts wth a set of good quality drawings (data pack). with the 3D model fully created and in full scale, drawings of any of the parts or section can be created using the powerful drawing features of Cadkey. Creating sections at any fuselage or wing station is now possible leaving no room for mistakes or inaccuracy and saving a lot of design and drawing time. Full scale templates can now be printed out to assist in the manufacturing process. Further detail design of the cockpit layout and control devices can be performed in 3D.

To create the molds for manufacturing of the complex surfaces of sailplanes Computer Aided Manufacturing (CAM) procedures can be implemented. The surface of the molds can be machined on a multi-axis milling machine. The tool paths for these surfaces can be created by any of the many available CAM software by directly importing from the Cadkey data base.

CONCLUDING REMARKS

A CAD interface program code was designed to create accurate sailplane geometry from basic data in a few seconds. Airfoil and fuselage co-ordinate data is used to construct a full scale 3D model on a Cadkey 3D design software data base. Fillets routines are used to create the complex fillets between fuselage and wings and betwen wings and winglets.

In order to perform flow analysis using 3D computational techniques, the required vertex list and spline list data files area quickly extracted from the CAD data base.

Manufacturing drawings can easily be compiled of any part or section using the full scale 3D model of the entire sailplane. Toolpath data can be created to manufacture the molds to create the complex surfaces.