Advances in Engineering Software 40 (2009) 479–486

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Parametric design of aircraft geometry using partial differential equations

Michael Athanasopoulos *, Hassan Ugail, Gabriela González Castro

School of Informatics, University of Bradford, Richmond Road, Bradford, West Yorkshire BD7 1DP, UK

ARTICLE INFO

Article history:
Received 1 November 2007
Received in revised form 22 April 2008
Accepted 20 August 2008
Available online 20 November 2008

Keywords:
Aircraft design
Parametric design
PDE surfaces

ABSTRACT

This paper presents a surface generation tool designed for the construction of aircraft geometry. The software generates complex geometries which can be crafted or modified by the user in real time. The surface generation is based on Partial Differential Equations (PDEs). The PDE method can produce different configurations of aircraft shapes interactively. Each surface is generated by a number of curves representing the character lines of a given part of the aircraft shape that can be manipulated in real time. Different surfaces then blend to create the full shape of the airplane. An important function of the proposed tool is its ability to change the aircraft shape through the adjustments of parameters associated with the initial curves. The user can apply linear transformations to the curves generating the airplane through simple input from the computer keyboard and the mouse. The updated curves can then be used to generate the surface leading to different configurations of a given airplane shape. The work presents detailed descriptions on the PDE method, parametric design and manipulation of aircrafts along with graphical demonstrations of its abilities and a series of examples to illustrate the capacity of the methodology implemented.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

The growth of computer hardware and the progress of computational algorithms over the last few years have given us new alternatives for creating airplane geometry. Complex and time consuming computations are now affordable and they are easy to use through the present Computer-Aided Design (CAD) packages which provide various tools for parametric design of complex geometries and surfaces in general. Current CAD packages utilize parametric design by linking dimensions and variables to geometry in a way such that when the values change the shape of the object change as well. Thus, design and modification of an object or a group of objects can be performed very fast and in real time. It is also very common to represent surfaces in terms of polynomial functions of two parameters. The polynomial based methods used in CAD depend on the type of polynomial chosen.

Examples of such surfaces are: Bezier surfaces [1], B-Splines [2], Rational B-Splines [3] and Non-uniform rational B-Splines (NURBS) [4]. When using polynomial based methods the surface patch is usually generated using a set of control points, by manipulating these control points the surface shape changes. One problem that arises here is that the control points are usually too many, making it difficult to manipulate the underlying geometry. Thus, there is a need for developing techniques that allow the user to manipulate the surface effectively using minimum number of controls.

One of the major tasks in the design phase of a new aircraft is the definition of its configuration along with the main geometric characteristics. Present CAD packages provide several tools for the parametric description of complex geometries and surface definition is usually accomplished by utilizing such CAD systems. For an aerodynamic early phase conceptual design, a step before the application of CAD is needed, i.e. a toolbox that will produce generic and parameterized aerodynamic surfaces, which will take into account the special needs and constraints for the conceptual design of an aircraft. Besides the well known general CAD packages, very few are specialized in aircraft design. Below we highlight some of these.

Klein and Sobieczky [5,6] present examples of aerodynamic design of high speed airfoils and wings which is carried out by their Genetic Algorithm software. They use explicit functions, called basic functions to describe curves needed in the design of aircraft surfaces. Their goal is achieved by establishing a flexible input data generator for both direct and inverse design. The geometry and flow quality are modelled by a set of analytical functions with parameterized input. For communication purposes with CAD packages, the resulting surface is interpolated using NURBS surfaces.

A surface generation software named Generic Parameterized Aircraft Surface (Ge.P.A.S.) developed by Sarakinos and Valakos [7] is also designed for the construction of aircraft aerodynamic surfaces. The surface generation procedure is parameterized and different aircraft configurations can be produced in an interactive manner. The surface generation procedure is based on the use of NURBS curves and surfaces. Fuselage type surfaces are constructed

* Corresponding author.
E-mail address: m.athanasopoulos1@bradford.ac.uk (M. Athanasopoulos).

0965-9978/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2008.08.001

inside a scalable reference volume, where common basic curves in successive parallel planes are transformed to form the corresponding cross sections. Wing type surfaces are constructed in a more straightforward manner, using standard formulations for wing geometry definition and a database of wing sections to select from.

A number of numerical and experimental tools have been developed at the Department of Aeronautical Planning and are devoted to aircraft design for low subsonic flow [8]. These tools comprise numerical codes for analysis, design and simulation, experimental tests and are particularly oriented to light aircraft design. The user also has the choice of virtually flying the aircraft that is being designed using an interactive platform flight simulation.

The work entitled "Aircraft Design Support using Knowledge Engineering and Optimisation Techniques" [9] is an example of parametric airplane modelling showing that a proper combination of object oriented programming, rule based instantiation of objects and a geometry engine allows parametric modelling in the optimisation sense. The principle and implementation of High-Level Primitives (HLPs), i.e. functional building blocks, known as Multi-Model Generator (MMG) has proved to be a good approach to the problem of parametric modelling of complex products. It is also shown how these parametric models can be used and initialized in Design and Engineering Engines (DEEs). A DEE facilitates initiation of design parameters and variables, instantiates HLPs and creates Multiple Models to support the required multi-disciplinary perspective on the system as required in multi-disciplinary analysis and optimisation. DEE offers a framework for design decisions in the conceptual design phase.

Multi Model Generator (MMG) is another knowledge-based parametric tool for reproducing a conventional aircraft [10], whose purpose is to introduce the MMG into a dedicated Design and Engineering Engine (DEE) for performing load calculation in the preliminary design phase. For this purpose, MMG has to be capable of supplying different models of the same product, i.e. structure, mass and aerodynamic models in order to feed a set of analysis tools. The generated models are extracted from a Knowledge-Based Engineering (KBE) product tree, which is capable of holding the knowledge of the complete aircraft product. The definition of the aircraft is fully parametric so that consistent models for the different disciplines can be generated for a large variety of aircraft configurations by varying a single input file. In this case, the aircraft is modelled as an assembly of components, which in turn is built up as an assembly of high-level primitives.

RAGE [11] is another package that has been designed for the generation of aerodynamic models, where central parameterized geometry definition has been used. It is used at the preliminary stage of aircraft design for preliminary parametric studies and optimisation. Designers can develop aircraft geometries that vary from very simple to quite detailed ones for analysis with an assortment of computational aerodynamics tools in a very fast manner. Aircrafts are designed with the use of fuselage, wings and nacelle components which are available in the software. Each component is generated by a number of subcomponents that define the geometry. RAGE is able to output files that are compatible with several aerodynamic analysis codes and is compatible with most of the major CAD packages.

RDS-Professional [12,13] is an aircraft design and analysis program suitable for conceptual design and trade studies, technology evaluations, and preliminary performance predictions. It also supports 3D design and incorporates analysis modules for aerodynamics, weight, propulsion and cost. The software automates classical techniques for aircraft design used in industry as well as computational tools that perform trade studies and early stage optimisation. Aircraft geometry is produced using lofting techniques interactively, whilst aircraft components are designed using geometry parameters possessing some physical meanings.

The method that is being used in this paper for the construction of airplane surfaces is the so called PDE method [14,15]. Unlike spline techniques, the PDE method can produce complex surfaces in terms of a small set of design variables or parameters. The shape of the surface is defined through boundary curves and a small set of design parameters, taking a boundary value approach to the problem of surface design. This approach also allows the design system to be extended so that the functionality of the object can be taken into account at an early stage of the design process. Additionally, it has been shown that the use of the PDE method can significantly reduce the computational cost associated with the process of designing and optimising the performance of either a given airplane or specific components such as wings [16,17]. Here, the optimisation of the airplane geometry takes advantage of a small set of design parameters required by the PDE method to define complex geometry. Thus, using a small geometry parameter set it is possible to optimise the shape of an aircraft automatically in a reasonable time scale. For example it is possible to automatically compute a shape which minimises lift or drag.

The aim of this work is to discuss the advantages of using the PDE method in modelling airplanes in contrast with the use of other existing methods and to show how airplane surfaces with more general boundary conditions can be constructed and manipulated in real time. Additionally, the authors aim to demonstrate that this technique is capable of representing and manipulating an already existing geometric model of an airplane without any prior knowledge on the part of the designer regarding the mathematical details of the PDE method itself. For the purpose of explaining the use of the PDE method in airplane design, we will use examples illustrating the various designs, transformations and modifications of airplane models. All the examples presented throughout this work have been created interactively in real time.

The rest of this paper is organized as follows. Section 2 describes the mathematical details behind the PDE method and the solution in its standard formulation. Section 3 explains how the PDE method can be used to create a generic aircraft shape. Details are also discussed on the parametric representation of the airplanes and how to control aircraft shapes through adjustments of parameters associated with the curves. Section 4 contains examples and demonstration of various shapes of aircrafts using the proposed software and finally the conclusions of this work are presented in Section 5.

2. The PDE method

The boundary value approach based on the PDE method can be summarized mathematically in the following manner. Let $X(u,v)$ be a function defining a surface in 3D space in a domain Ω (with boundary $\partial\Omega$, on which some boundary data is specified).

Here u and v represent the parametric coordinates of a point in Ω, and $X(u,v)$ as a mapping from that point in (u,v) to a point in 3D space such that $R^2(\Omega) \to E^3$. We regard X as the solution to a partial differential equation of the form

$$\underline{X}(u,v) = (x(u,v), y(u,v), z(u,v)) > \qquad (1)$$

Thus, the full problem consists of finding the solution to a set of three PDEs, one for each Cartesian coordinates (x,y,z) where a is an intrinsic parameter.

$$\left(\frac{\partial^2}{\partial u^2} + a^2\frac{\partial^2}{\partial v^2}\right)^2 \underline{X}(u,v) = 0. \qquad (2)$$

A wide variety of methods exist for finding the solution of elliptic PDEs similar to that shown in Eq. (2). These include elementary separation of variables, Green's Functions, and sophisticated numerical techniques. The approach undertaken in this work is based on the approximate solution to Eq. (2). The solution is then given in terms of analytic functions.

Taking the region Ω of (u, v) parameter space to be the region corresponding to $(0 \leqslant u \leqslant 1, 0 \leqslant v \leqslant 2\pi)$, Eq. (2) is solved subject to the boundary conditions on the solution which relate how $\underline{X}(u, v)$ and its normal derivatives $\partial\underline{X}/\partial n$ vary along $\partial\Omega$. The boundary conditions imposed on the solution are of the form

$$\underline{X}(0, v) = \underline{P}_0(v), \tag{3}$$

$$\underline{X}(1, v) = \underline{P}_1(v), \tag{4}$$

$$\underline{X}_u(0, v) = \underline{d}_0(v), \tag{5}$$

$$\underline{X}_u(1, v) = \underline{d}_0(v), \tag{6}$$

The boundary conditions $\underline{P}_0(v)$ and $\underline{P}_1(v)$ determine the value of $\underline{X}(u, v)$ of the surface patch at $u = 1$ and $v = 1$, respectively. The derivative conditions $\underline{d}_0(v)$ and $\underline{d}_1(v)$ determines the values of the normal derivatives at the respective boundary of the surface. The derivative conditions play a very important role in determining the overall shape of the surface. Both the size and the direction of the derivative vector can be changed interactively to bring about changes in the shape of the surface. Since the derivative vector is defined using a space curve, the shape and position of this curve relative to the positional boundary curve determine both the direction and the magnitude of the derivative vector.

In order to generate a typical PDE surface, Eq. (2) is solved over a finite region Ω of the (u, v) parameter space subject to the boundary conditions with periodic boundary conditions, υ being the periodic parameter, and using the method of separation of variables. Thus, the analytic solution of Eq. (2) can be written as

$$\underline{X}(u, v) = \underline{A}_0(u) + \sum_{n=1}^{\infty}[\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)], \tag{7}$$

where

$$\underline{A}_0 = \underline{\alpha}_{00} + \underline{\alpha}_{01}u + \underline{\alpha}_{02}u^2 + \underline{\alpha}_{03}u^3, \tag{8}$$

$$\underline{A}_n = \underline{\alpha}_{n1}e^{anu} + \underline{\alpha}_{n2}ue^{anu} + \underline{\alpha}_{n3}e^{-anu} + \underline{\alpha}_{n4}ue^{-anu}, \tag{9}$$

$$\underline{B}_n = \underline{b}_{n1}e^{anu} + \underline{b}_{n2}ue^{anu} + \underline{b}_{n3}e^{-anu} + \underline{b}_{n4}ue^{-anu}, \tag{10}$$

and a_{n1}, a_{n2}, $a_{n3,}$ a_{n4}, b_{n1}, b_{n2}, a_{n3} and b_{n4}, are vector-valued constants, whose values are determined by the imposed boundary conditions at $u = 0$ and $u = 1$. For a given set of boundary conditions, it is necessary to Fourier analyze the boundary conditions and identify the various Fourier coefficients in order to find a one to one correspondence between the boundary Fourier coefficients of the boundary conditions and the constants a_i and b_i, respectively. For an approximate solution, the Fourier series can be truncated (typically $N = 6$) and a remainder term is included. Thus, Eq. (7) leads to

$$\underline{X}(u, v) = \underline{A}_0(u) + \sum_{n=1}^{N}[\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)] + \underline{R}(u, v), \tag{11}$$

where $R(u,v)$ is a remainder function defined as

$$\underline{R}(u, v) = \underline{r}_1(v)e^{wu} + \underline{r}_2(v)ue^{wu} + \underline{r}_3(v)e^{-wu} + \underline{r}_4(v)ue^{-wu} \tag{12}$$

where $w = N(a + 1)$ and r_1, r_2, r_3, r_4 are responsible for satisfying the original boundary conditions and are obtained by considering the difference between the original boundary conditions and the boundary conditions satisfied by the function

$$\underline{F}(u, v) = \underline{A}_0(u) + \sum_{n=1}^{N}[\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)]. \tag{13}$$

Fig. 1 shows a sequence of surfaces that illustrate the effect of derivative condition X_u on the shape of the surface. Note that all the surfaces shown in Fig. 1 have the same boundary conditions on the function X whereas the boundary condition on the function X_u at $u = 0$ has been varied.

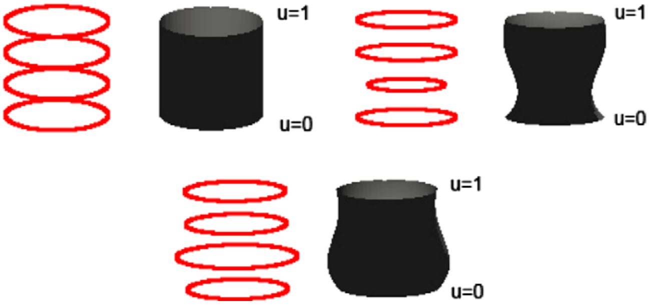


Fig. 1. The effect of derivative conditions on a PDE surface.

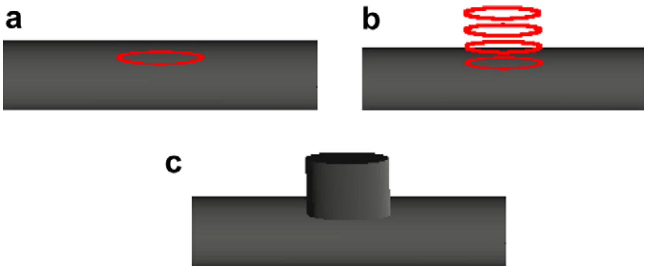


Fig. 2. Blending surfaces. (a) Curve on the surface resulting from the planar curves on the (u, v) space. (b) Curves corresponding to the blend surface. (c) Original surface with along with the blend surface.

The basic idea in creating any freeform surface patch using the PDE method is to define the boundary conditions appropriately and seek the solution of the PDE that defines the surface in question. Once a surface patch has been created, we can seek the PDE solution of the next surface patch by evaluating the boundary conditions using the next set of curves in order to create the adjacent surface patch. In order to create more complex shapes we need to be able to create objects that will contain more than one surface patch. For example, to create a two patch fuselage surface we need more curves to control the overall shape of the object, thus making it more detailed. In Fig. 4 one can see the generating curves for a two patch surface.

Another function that is very important for many practical designs is to blend one surface, or part of it, to another. For this reason a given portion of a surface patch may need to be chosen to blend with another surface. First we need to determine the exact position of the surface from where the portion needs to be trimmed, as illustrated in Fig. 2a. Any chosen point in the (u, v) parameter space will have an associated point on the surface. Thus, by creating a new curve on the parametric domain, points on the surface that map onto the points in the parameter space can be identified. Note that any plane curve drawn in the (u, v) parameter space will be guaranteed to lie on the surface. That curve can then be manipulated interactively by the user. Then we can insert more curves to design the second object we want to blend as shown in Fig. 2b and by using the PDE method we can generate the blend surface as shown in Fig. 2c.

3. The use of the PDE method for designing a generic airplane shape

In this section we discuss how to use the PDE method for designing a generic airplane shape. To this end, we have developed a computer program allowing us to design and manipulate curves and to use the PDE method to create the objects. The programming was designed and implemented using Visual C++ with MFC (Microsoft Foundation Class) along with OpenGL and GLUT for rendering and manipulating the 3D objects.

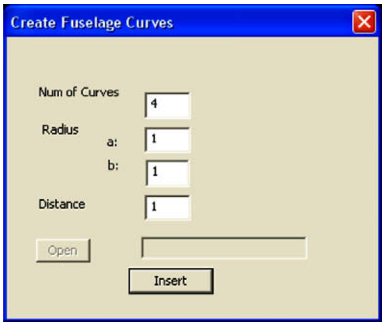
The basic idea in creating any surface patch or an airplane shape using the PDE method is to define the boundary conditions appropriately and seek the solution to its corresponding PDE for generating the associated surface. In order to use the PDE method to design airplanes, we first split the whole design procedure into several hierarchical steps. For example, we start with the main body or the fuselage of the airplane before we design its wings or the tail. Thus, we first create the fuselage, which in our case contains at least two surface patches in order to obtain a more realistic shape.

To create the fuselage object we insert the initial curves for the creation of the fuselage, as shown in Fig. 3. The user can choose the number of curves, their radii that will control the shape of the curves and the distance between each curve. These curves can then be manipulated in the plane via transformation of the control points to achieve the desired shape. Various geometric transformations such as translations, rotations and scaling can then be applied to this plane curve in order to create a curve in 3D space.

As mentioned before, for more realistic results of the actual fuselage shape, we need at least 2 surfaces patches. For creating 2 surface patches we need to insert 7 curves, since each surface patch requires 4 curves, and connect them using as initial curve of the second patch with the last curve of the first patch. Thus, using the PDE method we can generate a surface for these curves. As one can see in Fig. 4, the last curve of the first patch is the first curve for the second patch. Thus, we can create as many patches as required and they will all be blended to each other smoothly.

Having created the fuselage curves, as shown in Fig. 5a, we can then transform every curve until we reach the desired shape. To generate the surface between each curve, we have to use the PDE method and then display the output shape as shown in Fig. 5b. The algorithm is based on the following steps.

- Generate the curves and save them to an external file for later use.
- Decide on the number of surface patches required to generate the shape.
- Use derivatives to calculate the distance between each curves.
- Fourier analyze the boundary conditions and identify the different Fourier coefficients.
- Compute the analytic solution of the PDE for a given surface resolution.
- Export the generated surface as an obj file.


Create Fuselage Curves

Num of Curves: 4

Radius a: 1

b: 1

Distance: 1

Fig. 3. Software menu for insertion of fuselage curves.

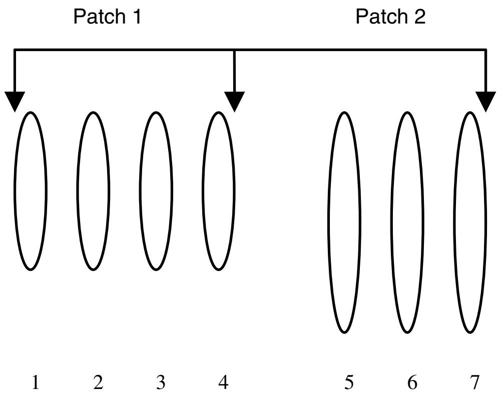


Fig. 4. The generating curves for a 2 patch surface.

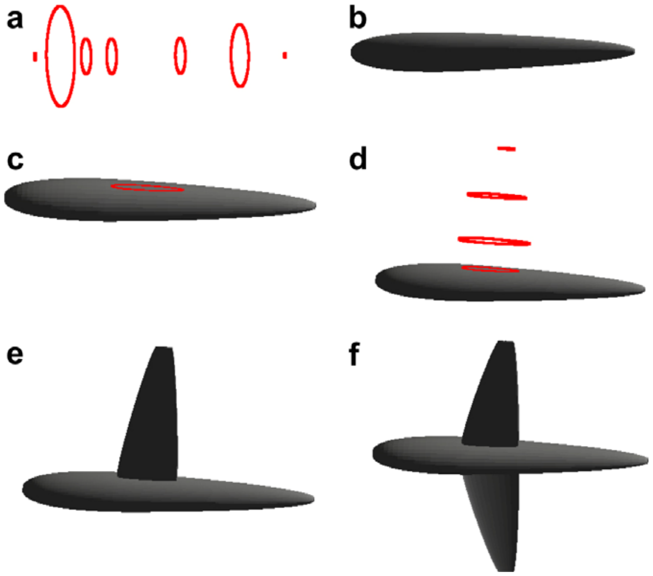


Fig. 5. Construction of a generic aircraft shape. (a) Fuselage curves. (b) Two patches fuselage object. (c) The initial curve for the wing. (d) The generating curves of the wing. (e) Wing and fuselage objects blended. (f) The final basic shape of the airplane.

After the fuselage surface patch is generated, the next step is to create the wing object. Again we follow the same procedure for generating a surface patch by solving the PDE. Before we create the surface patch for the wing, we need to use the blending function to create the first curve that will connect the fuselage with the wing. To this end, we need to use the fuselage boundary coefficients and create a curve in the (u, v) space that contains the required coefficients. This curve on the (u, v) space will be the first curve to create the wing object as shown in Fig. 5c. We can translate or scale the curve on the (u, v) space in order to obtain its desired shape on the fuselage. Then, we can insert more curves through a menu to design the wing as shown in Fig. 5d. Using the PDE method we generate the surface for the wing curves and display the created object as shown in Fig. 5e. To create the other side of the wing, a simple mirror symmetry is applied to the wing curves which then results in the basic shape of airplane as shown in Fig. 5f.

3.1. Parametric representation of an airplane

So far we have described how to construct generic airplane surfaces by constructing curves and using them as boundary conditions to generate the surface through the PDE method. The next step is to represent various shapes using parametric design. The method to construct the surface remains the same; however we need to be able to manipulate the shape of any curve to design various airplane shapes. Surface manipulation is carried out intuitively in an interactive environment by using a set of parameters which are associated with the boundary conditions. Each curve that is created within the application is stored in a file for later manipulation. Each transformation, such as translation, scaling or rotation, is based on parameters to change the original shape of the curve to a desired shape by the user. To that effect, we use the mouse coordinates and update the parameter according to the axis the user has selected. Fig. 6 shows the transformation axis that controls the selected curves. By selecting a transformation from the menu and a curve, an axis appears that controls the X, Y and Z coordinates of the curve in the three dimensional space.

Each transformation updates the curves differently, i.e. for scaling we need to multiply each axis by a given number

$$\begin{aligned} x' &= s_x * x, \\ Y' &= s_y * y, \\ Z' &= s_z * z, \end{aligned} \tag{14}$$

where s_x, s_y or s_z are the parametric values that update the axes by multiplying with the original x, y and z coordinates, respectively, and give us the new scaled curve. For rotating the curve around a specified axis, we need to select which axis we need to transform and apply the formula of that axis with the angle we need to rotate it. For example Eq. (15), states the new coordinates of any given point of the curve after rotating it around the x-axis by the angle e

$$\begin{aligned} X' &= x \\ y' &= \cos(e) * y - \sin(e) * z \\ z' &= \sin(e) * y + \cos(e) * z \end{aligned} \tag{15}$$

Finally, translation is another transformation we can apply to each curve and it is carried out by simply adding or subtracting a point to any given point. The new coordinates of any given point on the curve are now given by Eq. (16), where t_x, t_y and t_z represent the components of the translation point.

$$\begin{aligned} X' &= x + t_x, \\ y' &= y + t_y, \\ z' &= z + t_{z.} \end{aligned} \tag{16}$$

After completing all the required transformations, the updated curve is stored in a file containing all the curves to create the selected object. There are two types of external files that are used in the application we have created. They are:

- The CurveFiles containing only the curve coordinates for each object, such as, fuselageCurveFile, wingCurveFile, etc. These files are updated in real time with each transformation and are sent to the PDE method to construct the final object.
- The shapeFiles are produced by the algorithm and contain the generated object, such as, fuselageShape, wingShape, etc.

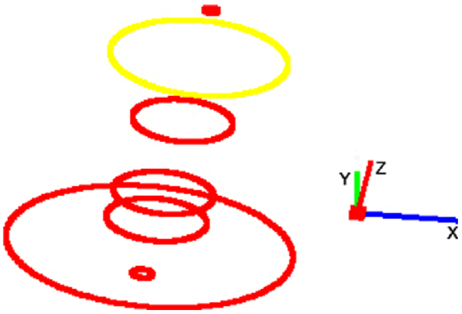


Fig. 6. The transformation axis.

3.2. Controlling aircraft shapes

By transforming each curve we can change the way the curve is represented in our system and eventually the overall shape it gives rise to. The same procedure as before is applied here, i.e. we use parameters to transform curves, but the only difference is that we transform more than one curve to change the entire shape interactively. For example, to manipulate the fuselage object we have two options; we scale each of its patches separately or we translate it to increase or decrease the length.

Fig. 7 shows the fuselage of the airplane, in the scaling mode. The user can select each of these vectors, depending on the number of patches the object has, and scale the fuselage in the x, y or z axis. The curves can also be translated separately in this mode for better design. Again using parametric design we can achieve transformations by dragging the mouse along the desired direction. The fuselage object can be manipulated according to the number of surface patches constructed. The wing, tail or side tail objects need to be transformed in a different way because they are connected with the fuselage object. As mentioned before, in order to create any object that is attached to fuselage, we need to create a curve in the fuselage's parametric domain. Once this curve has been created on the fuselage, its shape can be changed and it can be placed

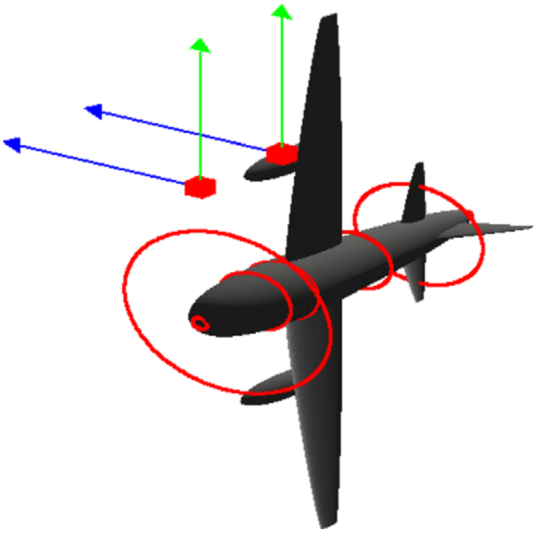


Fig. 7. Two different vectors for changing the fuselage shape, each vector transforms the selected surface patch.

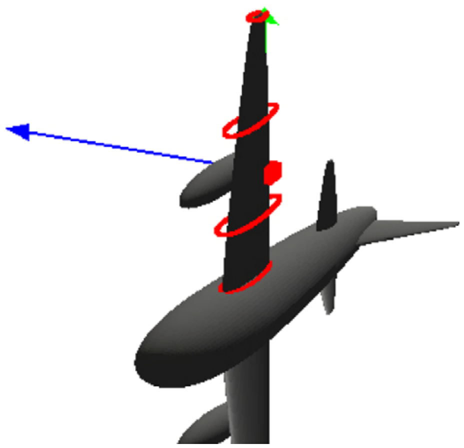


Fig. 8. Scaling and translating the wing object using transformation parameter.

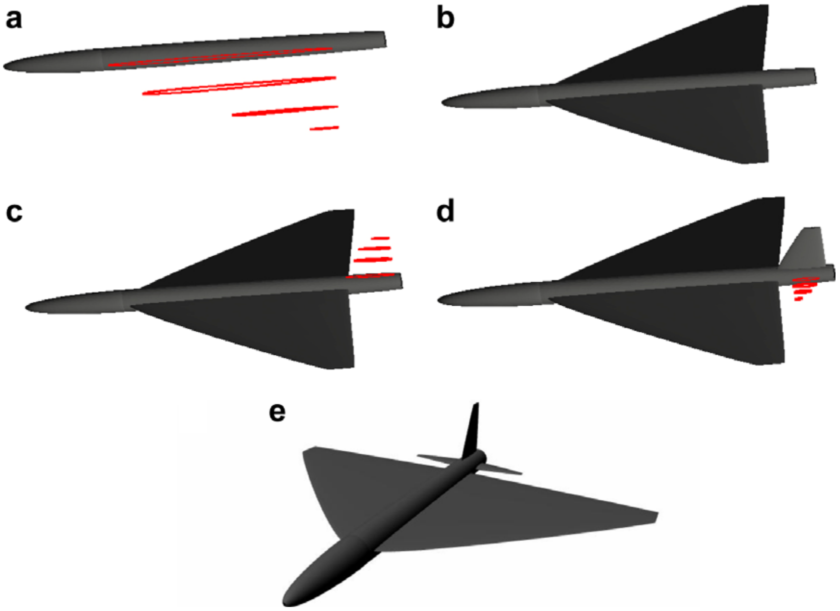


Fig. 9. The steps to construct a delta wing configuration.

anywhere on the fuselage by means of applying the transformation techniques.

The wing object has one vector for transformations; that is, it can be translated or scaled in any direction. Fig. 8 shows the wing section of the airplane, being scaled down and translated. This way, we can change the overall shape of the airplane by simply scaling or translating any piece comprising the geometric model of the airplane as required by the user.

4. Examples

This section contains several airplane configuration examples in order to demonstrate the range of applications using the methodology presented here.

Fig. 9 shows the first example, which consists of a delta wing configuration airplane. To construct such a configuration the user first has to design the fuselage, which is the base for the wing, tail and side tails. From the insert menu of the application we insert 7 curves to construct a two patch object. Next we transform each curve separately until the final shape looks like an aircraft fuselage. The second step is to create the wing object and blend it with the fuselage. For that we need to insert an initial curve from the wing menu that will control the position of the wing generating curves. The generating curves for the wing are shown in Fig. 9a and they are controlled by the initial curve, which is then created on the fuselage surface. The number of wing curves depends on the number of surface patches defining the particular shape of the wing. In this example we create a single surface patch, which contains 4 curves, for a two surface patch we will need to insert 7 curves. The next step is to transform each curve separately until we achieve the desired shape and use the PDE method to create the wing surface. When the wing configuration is ready, we clone the wing object to create the second wing as shown Fig. 9b. The next part as shown in Fig. 9c and d is to create the side tail and tail object where the process is exactly the same as that used in

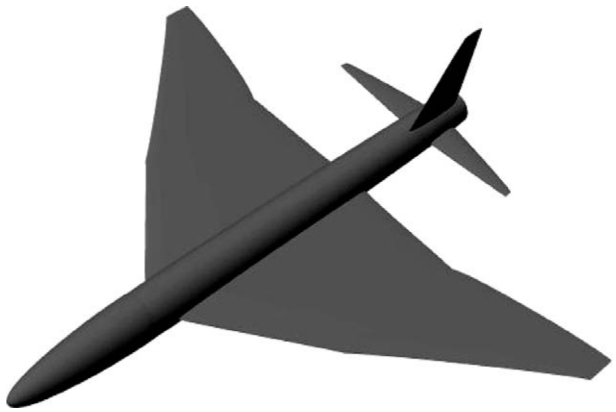


Fig. 10. Double delta wing configuration.

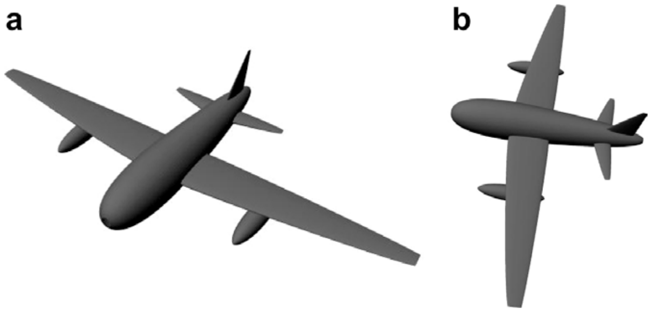


Fig. 11. Airplane model B-17, generic wing, tail and side wings configuration.

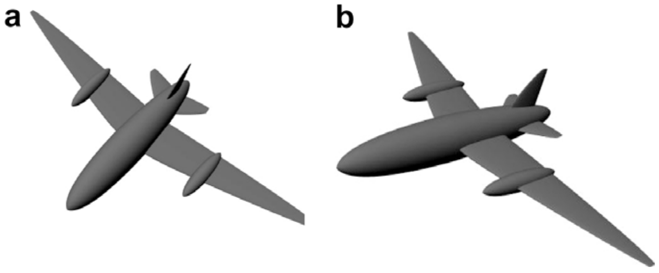


Fig. 12. Generic configurations, with engines blended with wings.

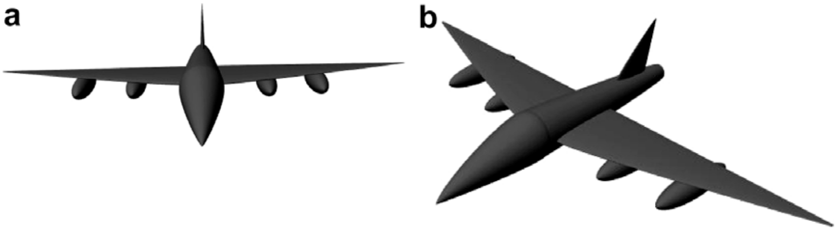


Fig. 13. Fighter configurations with delta wing shape and four engines.

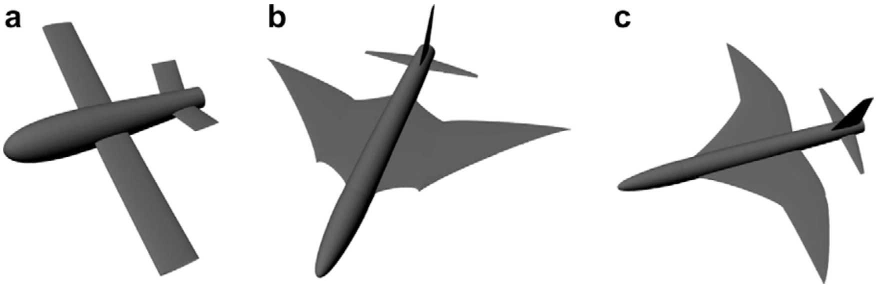


Fig. 14. Various other configurations.

constructing the wing object. We select from the tail menu the initial curve, we insert the number of generating curves we want to design the object, generate the surface and copy it on the other side. Fig. 9e shows the complete Delta wing configuration.

The second airplane example, shown in Fig. 10, has a double delta wing configuration. There are two alternatives for constructing it. The first one consists of following the same procedure as before, constructing the plane from scratch and transforming the wing object until we achieve the double delta wing configuration.

The second alternative is much easier and faster. From the Planes menu we select delta wing and the delta wing plane along with it control curves are rendered on the computer screen. From there, we can easily transform any part of the aircraft and change its configuration. To change the delta wing into a double delta wing, we can simply select one of the 3 transformation buttons. Thus, by selecting translate or scale and clicking the wing object, the curves that construct the object appear.

There are two ways to manipulate the object at this stage and they both depend on the transformation selection we have undertaken. If we have selected to translate the object then by pressing the left or right button on the keyboard the wing translates its position along the fuselage object. If we want to translate separately each curve then click on the select individual curve check button to enable that function. If we want to scale the object by using the up, down, left, right buttons we scale it, or by selecting the individual curve we can select and scale each curve separately. In conclusion, by transforming the curves we construct a double delta configuration starting from a delta configuration.

Figs. 11–14 show additional examples that demonstrate various configurations of airplanes. In these examples the planes contain engines that are incorporated into the wing section in the same manner as the wings are blended to the fuselage.

5. Conclusion

The surface generation software presented in this work has been designed for the construction of generic aircraft shapes. The surface generation is based on the use of PDE method and is fully parameterized so that completely different aircraft configurations may be generated. Unlike other techniques, the PDE method can parameterize complex surfaces in terms of a small set of design variables, instead of hundreds of control points. The shape of the surface is then defined through boundary conditions and a small set of design parameters. Taking a boundary value approach to the problem of surface design has the advantage that most of the information required to define a surface is contained at the boundary or character lines representing that object. The representation and the control of aircraft shapes is undertaken through the adjustments of parameters associated with the curves. Thus, the user can interactively transform the parameters and change the shape of the aircraft without having any knowledge of the mathematical theory behind the PDE method.

The software is designed in a user friendly interactive manner in order to allow the user to design and manipulate any aircraft configuration in real time. Improvements to be included in the software are: more airplane parts, more tools for designing accurate airplanes, add materials, colours and textures to the airplane to make it look more realistic and export 3D objects in various formats so they can be used with commercial CAD packages. Future work could also be undertaken in either simulation by using a dynamics engine or stress/strain analysis; for instance, simulating an aircraft under several weather conditions or applying stress analysis to determine the stress in materials and structures subjected to static or dynamic forces.

References

[1] Bezier P. The mathematical basis of UNISURF CAD system. London: Butterworths; 1986.
[2] Woodward CD. Skinning techniques for interactive B-spline surface interpolation. Comput Aided Des 1988;20:441–51.
[3] Tiller W. Splines for curve and surface representation. Comput Graphics Appl 1983;3:61–9.
[4] Schumaker LL. Spline functions: basic theory. New York: John Wiley and Sons; 1981.
[5] Sobieczky H, Dulikravich G, Dennis BH. Parameterized geometry formulation for inverse design and optimization. In: Proceedings of the 4th international conference on inverse problems in engineering. Brazil: Rio de Janeiro; 2002.
[6] Klein M, Sobieczky H. Sensitivity of aerodynamic optimization to parameterized target functions. In: Proceedings of the international symposium on inverse problems in engineering mechanics. Japan: Nagano; 2001.
[7] Sarakinos S, Valakos I, Nikolos I. A software tool for generic parameterized aircraft design. Adv Eng Software 2007;38(1):39–49.
[8] Coiro DP, Nicolosi F. Design of low-speed aircraft by numerical and experimental techniques developed at DPA. Aircraft Des 2001;4:1–18.

[9] van Tooren MJL, Nawijn M, Berends JPTJ, Schut EJ. Aircraft design support using knowledge engineering and optimisation techniques. In: AIAA structures, structural dynamics and materials conference; 2005.

[10] Cerulli C, Meijer PB, van Tooren MJL. Parametric modelling of aircraft families for load calculation support. In: Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference; 2004.

[11] Rodriguez D, Sturdza P. A rapid geometry engine for preliminary aircraft design. In: Proceedings of the 44th AIAA aerospace sciences meeting and exhibit. Nevada: Reno; 2006.

[12] Raymer D. RDS-professional in action: aircraft design on a personal computer. SAE/AIAA paper 965567; 1996.

[13] Raymer D. RDS:A PC-based aircraft design, sizing, and performance system. AIAA paper 92-4226; 1992.

[14] Ugail H, Bloor MIG, Wilson MJ. Techniques for interactive design using the PDE method. ACM Trans Graphics 1999;18(2):195–212.

[15] Ugail H, Bloor MIG, Wilson MJ. Manipulation of PDE surfaces using an interactively defined parameterization. Comput Graphics 1999;23(4):525–34.

[16] Sevant NE, Bloor MIG, Wilson MJ. Aerodynamic design of a flying wing using response surface methodology. AIAA J Aircraft 2000;37(4):562–9.

[17] Sevant NE, Bloor MIG, Wilson MJ. Cost effective multipoint design of a blended HSCT. AIAA J Aircraft 1998;36(2):642–50.