Shape optimisation using CAD linked free-form deformation

A. Nurdin **N. W. Bressloff** **A. J. Keane**
andy.keane@soton.ac.uk

University of Southampton
Southampton
England, UK

C. M. E. Holden
Airbus Operations Limited
Bristol
England, UK

ABSTRACT

Free-form deformation (FFD) is a method first introduced within the graphics industry to enable flexible deformation of geometric models. FFD uses an R3 to R3 mapping of a deformable space to the global Cartesian space to produce the geometry deformation. This method has been extensively used within the design optimisation field as a shape parameterisation technique. Typically it has been used to parameterise analysis meshes, where new design geometries are produced by deforming the original mesh. This method allows a concise set of design variables to be used while maintaining a flexible shape representation. However, if a computer aided design (CAD) model of the resulting geometry is required, reverse engineering techniques would need to be utilised to recreate the model from the deformed mesh. This paper extends the use of FFD within an optimisation routine by using FFD to directly parameterise a CAD geometry. Two methods of linking the FFD methods with the CATIA V5 CAD package are presented. Each CAD integration technique is then critiqued with respect to shape optimisation. Finally the set-up and initialisation of a case study is illustrated. The case study chosen is the aerodynamic optimisation of the wing-fuselage junction of a typical passenger aircraft.

Paper No. 3603. Manuscript received 24 June 2010, revised version received 25 February 2011, second revision received 17 November 2011, accepted 25 January 2012.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

NOMENCLATURE

AoA	angle of incidence, degrees
AR	wing aspect ratio
[**B***]	matrix of Bernstein tensor products at object points
[$\mathbf{B}^{+}$]	pseudoinverse of the matrix of Bernstein tensor products at object points
$B_i(t)$	Bernstein polynomial
$b/2$	semi-span, m
c	Mean aerodynamic chord, m
C_D	drag coefficient
C_L	lift coefficient
C_M	pitching moment coefficient
C_P	pressure coefficient
DPW	Drag Prediction Workshop
D_{var}	vector of design variable values
l, m, n	number of lattice control points in the S,T and U directions respectively
M	Mach number
N	Number of mesh nodes
Ob	object point in Cartesian co-ordinates
P	control point
S	wing reference area, m^2
S,**T**,**U**	unit vectors of the local co-ordinate system
s, t, u	local co-ordinates of a geometry point
W	vector of control point weights
X	arbitrary geometry point
X_{ffd}	displaced arbitrary point in Cartesian co-ordinates
X_0	origin of local co-ordinate system
X_{ref}	moment coefficient reference point x co-ordinate
Y_{ref}	moment coefficient reference point y co-ordinate
S_{ref}	moment coefficient reference point s co-ordinate
ΔOb	displacement of object points in Cartesian co-ordinates
ΔP	displacement of the lattice control points in Cartesian co-ordinates
Δ	difference of quantity
η	fraction of wing semi-span

Subscript

c, i, f	coarse, intermediate and fine
cruise	cruise conditions
i, j, k	control point indices
l, t	lattice position, target position
L, U	lower bound, upper bound

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

1.0 INTRODUCTION

Over recent years the advances made in computing power have revolutionised the design process. Traditional trial and error methods have been replaced by automated optimisation routines. The general idea is that by parameterising a design with a series of design variables and then automating design generation and assessment, a search process can be used to find optimum designs. During such an optimisation process hundreds/thousands of designs can be produced that have to be evaluated. It is therefore important to keep the computing resources needed per design to a minimum. However, it is now also commonplace within the aerodynamics industry to employ high fidelity computational fluid dynamics (CFD), a computationally expensive method, as a design quality assessment tool. For aerodynamic shape optimisation a flexible and parsimonious parameterisation method is therefore required to generate the new aerodynamic surfaces.

Within the field of shape optimisation the method of shape parameterisation is critical to the overall success of the optimisation process. The parameterisation technique defines the design space and hence determines the possible designs the optimisation process can produce. The suitability of a particular parameterisation technique is based on its ability to balance the need for a flexible representation and a manageable number of design variables. If the parameterisation is based on a limited number of design variables the resulting range of possible shapes is limited, therefore restricting the outcome of the optimisation process. Conversely, a very flexible parameterisation is likely to require a large number of design variables. If the number of design variables is too large the design space becomes unmanageable and the search process will struggle to find the optimum design[1]. Samareh[2] presents a survey of possible shape parameterisation techniques, which discusses in detail the advantages and disadvantages of each process.

One popular parameterisation technique is to use splines to define the shape. However, for complex geometries this method typically requires a large number of design variables. Free-form deformation (FFD) is also highlighted by Samareh as a potentially effective shape parameterisation technique. He states: "MSO [multidisciplinary shape optimisation] must rely on the parameterisation of the analysis grids, for which the FFD algorithm is ideal." FFD has been successfully implemented as a shape parameterisation method for aerodynamic surfaces within an optimisation process[3,4]. However, current usage deforms either an analysis mesh or a two-dimensional spline geometry. A CAD geometry is generally not present within the optimisation loop. This paper outlines a methodology to utilise FFD techniques within a CFD-based optimisation framework while maintaining a CAD geometry. The method employs directly manipulated FFD, CAD integration, response surface optimisation and CFD. The aim is to ensure that at each step in the optimisation process a complete CAD geometry is directly available for every configuration considered, without direct manipulation of the underlying CAD NURBS control points, but also so that there is no need to re-engineer any FFD generated geometry back into the CAD world. The approach is generic to any CAD or geometry kernel-based model although here the commercial Catia V5 framework is adopted throughout. The application of the method with respect to a practical design process is discussed via the optimisation of a wing-fuselage junction for a typical passenger aircraft. The following paper is set out in three broad sections. First, FFD methods and their usage within optimisation are discussed. Second, the optimisation framework and CAD integration methods are examined. Finally, the setup and initialisation of a case study is outlined, including geometry selection and generation, the optimisation work-flow, details of the mesh generation and CFD assessment.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

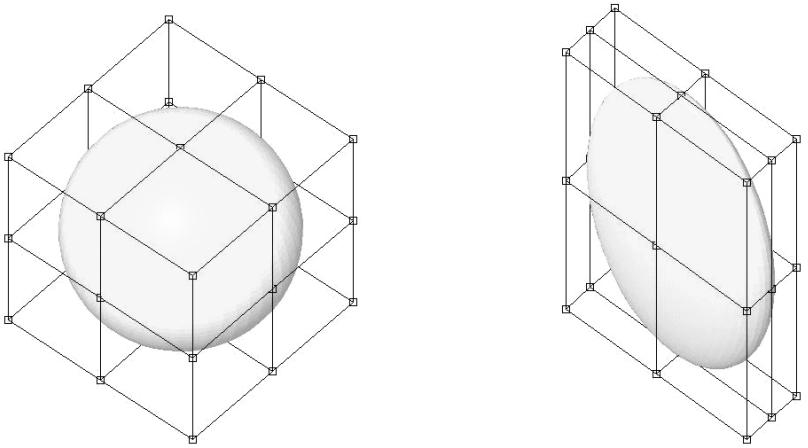


Figure 1. Sphere deformed using FFD.

2.0 FREE-FORM DEFORMATION TECHNIQUES AND UTILISATION

FFD was first introduced by Sederberg and Parry[5] in the field of computer graphics as a method for deforming geometric models in a flexible manner. In general, free-form deformation techniques encompass the following processes: first a volume definition is required and positioned such that it encloses the geometry to be manipulated, a method is then used to deform said volume and finally the geometry inside the volume must deform as part of the volume that encloses it. Sederberg and Parry's FFD method achieves this by using a tricubic Bezier volume that is defined using a lattice of control points. Movements in these control points provide the deformation of the control volume needed. This deformation occurs in an analogous fashion to deforming a Bezier curve through its polygon points. Deformations applied to the Bezier volume are passed to the geometry through its parametric co-ordinates. The following physical analogy, proposed by Sederberg and Parry is useful to understand the general FFD method. Imagine the geometry to be deformed is manufactured from a flexible material, for example a rubber ball. This geometry is embedded within a clear flexible plastic block. As the block is deformed the embedded geometry is forced to deform in a similar fashion, see Fig. 1. FFD is the mathematical equivalent of this process.

The Sederberg and Parry FFD method can be described in three steps: Control volume/lattice construction; freezing the lattice; lattice and geometry deformation. The three general steps are next discussed in turn.

2.0.1 Control volume/lattice construction

As previously mentioned Sederberg and Parry define the control volume using a tensor product trivariate Bezier volume, which is defined by a regular uniformly subdivided 3D lattice of control points. The lattice consists of l, m and n control points in the **S**, **T** and **U** directions, respectively. Within the control volume a local co-ordinate system is defined such that any point X within the control volume is defined by

$$X = X_0 + s\mathbf{S} + t\mathbf{T} + u\mathbf{U}, \quad \ldots (1)$$

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

where X_0 is the origin of the local co-ordinate system in Cartesian co-ordinates. Note that for any point within the control volume $0 < s < 1$, $0 < t < 1$ and $0 < u < 1$. Although Bezier volumes were used in the discussion by Sederberg and Parry, it is also possible to use other basis functions such as B-splines or non-uniform rational B-splines (NURBS)[6], this is discussed further in Section 2.1.1.

Sederberg and Parry restrict the initial lattice shape to be parallelepiped and aligned with the global co-ordinate system. While this limits the types of deformations achievable it has two major advantages. First, because of the convex hull property of Bezier volumes, the parallelepiped condition is necessary to ensure the control volume completely fills the defining lattice. The implication of this is that any geometry enclosed by the lattice can be defined using the Bezier volume parametric co-ordinates. If the initial lattice shape is not parallelepiped the user would need a good knowledge of Bezier curves and their behaviour to form the control volume. The second advantage, which is as a consequence of the first is that the 'lattice freezing' process is drastically simplified.

2.0.2 Freezing the lattice

The next step is to 'freeze' the lattice. This requires the calculation of the position of each geometry point in the local co-ordinate system. For a parallelepiped control volume this can be accomplished through simple linear algebra as

$$s = \frac{\mathbf{T}\times\mathbf{U}(X - X_0)}{\mathbf{T}\times\mathbf{U}\cdot\mathbf{S}}, t = \frac{\mathbf{S}\times\mathbf{U}(X - X_0)}{\mathbf{S}\times\mathbf{U}\cdot\mathbf{T}}, u = \frac{\mathbf{S}\times\mathbf{U}(X - X_0)}{\mathbf{S}\times\mathbf{T}\cdot\mathbf{U}} \quad \ldots (2)$$

2.0.3 Lattice and geometry deformation

Deformations are applied to the lattice by moving the control points away from the original positions. The geometry deformed position is then defined by a trivariate tensor product Bernstein polynomial as

$$X_{ffd} = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} B_i(s)B_j(t)B_k(u)P_{i,j,k} \quad \ldots (3)$$

In the case of the control volume definition using a Bezier volume the general form of the Bernstein polynomials is

$$B_i(s) = \binom{l}{i}(1-s)^{l-i}s^l \quad \ldots (4)$$

2.1 FFD advances

Although the FFD method presented by Sederberg and Parry is a powerful deformation tool it does have a few limitations. The two most significant of these are the restrictions on the lattice shape, and the indirect nature of the control mechanism. Sederberg and Parry restrict the initial lattice shape to be regular, parallelepiped aligned with the global co-ordinate system. These restrictions limit the types of deformations achievable. The indirect nature of the FFD control mechanism means it is not entirely

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

obvious to the user what deformation can be achieved by moving any given control point. With respect to shape optimisation this increases the complexity of choosing the design variables needed to achieve the required deformation types. This is further complicated if the control lattice contains internal control points. These limitations have been addressed by numerous subsequent studies.

2.1.1 Basis functions

The flexibility of the initial control volume representation was increased first by Griessmair and Purgathofer[6] and then Lamousin and Waggenspack[7], who extended the FFD method to use B-spline and NURBS bases, respectively. Lamousin and Waggenspack illustrate their method by deforming a model of a human leg. They show that the use of NURBS as the basis function enables the use of a non-uniformly spaced lattice and deformation through control point weights. These abilities increase the flexibility of the original FFD method. However, the control lattice is still restricted to a parallelepiped shape.

When using a NURBS basis with the FFD methods the lattice freezing process is complicated as a numerical search technique is needed. Lamousin and Waggenspack[7] use a golden section search routine, whereas Coquillart[8] utilises a Newton approximation. Whichever search process is used it can be very computationally expensive. However, if the grid is orthogonal and aligned with the global Cartesian co-ordinates the problem can be separated into three smaller independent searches, one for each of the local co-ordinate directions[8]. It is important to note that typically during an optimisation process the lattice creation and freezing processes only occur once. New designs are generally created by deforming a 'base' design and lattice.

2.1.2 Extended free form deformation (EFFD)

EFFD was first introduced by Coquillart[8] and expands the FFD methods to use non-parallelepiped shaped lattices. Coquillart uses a tensor product piecewise tricubic Bezier volume to define the geometry deformation. Within each subdivision of a standard $l \times m \times n$ FFD lattice, termed a 'chunk' by Clark[9], Coquillart places an arbitrary tricubic Bezier volume. This obviously extends the original lattice to a $(3l + 1) \times (3m + 1) \times (3n + 1)$ lattice. Each chunk is equivalent to a single basic FFD volume as presented by Sederberg and Parry[6]. Within Coquillart's EFFD implementation only the corner points of the chunks can be deformed. The positions of the intermediate lattice points are automatically calculated from the chunk point positions and are positioned so as to maintain continuity between the chunk volumes. This is achieved by maintaining a constant tangent over the chunk boundaries. Freezing the lattice requires first locating the lattice chunk that the geometry resides within and then using numerical search techniques to find the local point co-ordinates. Each geometry point within a chunk is defined by;

$$X_{ffd} = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} B_i(s)B_j(t)B_k(u)P_{i,j,k} \qquad \ldots (5)$$

where in this case B_i, B_j ,B_k are fourth order Bernstein polynomials. If the chunk is not parallelepiped a numerical search technique is utilised to solve Equation (5).

Coquillart states that the shape of the FFD lattice has a large effect on the deformations achievable, however, the use of lattices which are too complex can produce unpredictable results. The implication of this with respect to shape optimisation is that the choice of the lattice shape is critical to the shape of the design space and consequently the result and efficiency of the optimi-

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

sation process. Knowledge of the required type of deformation and/or approximate optimum shape aids lattice selection.

While the EFFD method does enable a greater range of deformations when compared to the original FFD methods, it loses some of the flexibility due to the automatic calculation of intermediate lattice points. It is also clear that the numerical search techniques needed to freeze the EFFD lattice are time consuming and computationally expensive. The EFFD lattice by its very nature will rarely be aligned with the global Cartesian co-ordinate system. Due to this fact, the numerical search cannot be reduced to three line searches, which can be used when a NURBS basis is utilised. Instead a more complicated three variable volume search is required. However, during a design optimisation process freezing the lattice still only needs to occur once to freeze the base design. All subsequent designs will use the frozen base co-ordinates.

2.1.3 Directly manipulated free form deformation (DMFFD)

DMFFD was first introduced by Hsu *et al*[10] as a method of providing a more intuitive interface for the FFD routines. In order to do this a new type of point is defined. An 'object point' is a point placed within the global co-ordinate system on to a particular part of the geometry that can be used as a manipulation handle. The user defines the required position of the object point after deformation. The deformed control point locations needed to produce the required deformation are then calculated and are subsequently used with Equation (3) to determine the deformed geometry positions. In order to calculate the required control point locations the following systems of equations have to be solved;

$$\Delta Ob = [\mathbf{B}^*]\Delta P. \quad \ldots (6)$$

Within the literature there are two distinct methods for solving Equation (6) for the control point displacements. Hsu *et al* utilised a pseudoinverse of [**B***] enabling Equation (6) to be rearranged to produce;

$$\Delta P = [\mathbf{B}+]\Delta Ob. \quad \ldots (7)$$

If the number of object points is less than or equal to the number of lattice control points Equation (7) is solvable. However, if the number of object points is greater than the number of lattice control points the system of equations is over-determined and hence there is no unique solution. If the system of equations is under-determined, i.e., the number of object points is less than the number of control lattice points, multiple solutions exist. In this situation the pseudoinverse gives the least squares solution and so defines the solution with the minimum control point displacement. Hu *et al*[11] outlined a method that uses a constrained optimisation method to derive an explicit solution without the use of a pseudoinverse. The resulting solution is

$$\Delta P_{i,j,k} = \frac{B_{i,j,k}(s_l,t_l,u_l)}{\sum_{i,j,k=0}^{l,m,n} B^2_{i,j,k}(s_l,t_l,u_l)}(Ob_t - Ob_l) \quad \ldots (8)$$

Equation (8) calculates the control point displacements needed to achieve one objective point deformation. Hu *et al* showed that for multiple objective points the required deformation can be achieved by a series of sequential single objective point calculations, as long as the objective points

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

are mutually independent, i.e., only one object point is present in an individual lattice 'chunk'.

With respect to shape optimisation, when DMFFD is used the design variables are the objective point deformed co-ordinates. This method can be used to reduce the number of design variables while maintaining flexibility within the geometry representation.

2.2 FFD implementation

The present study uses an implementation of a three-dimensional directly manipulated extended FFD method (DM-EFFD) using a B-spline basis. This method has been chosen as the EFFD grid allows a greater flexibility of shape parametrisation. It is clear from the present FFD usage detailed in the literature that the selection of the FFD lattice is critical to the outcome of the optimisation process. During the case study set out in Section 5 the EFFD grid enables the lattice to be wrapped around the wing profile, producing surface deformation normal to the wing. A direct manipulation scheme has been chosen to reduce the number of design variables and to make the design space more manageable. The B-spline basis is used as it provides the necessary local control without the added complexity of using a NURBS basis.

The EFFD implementation used by this study uses a piecewise tricubic B-spline volume to define the deformable space. In order to determine the chunk within which a geometry point resides, a three-dimensional Delaunay tessellation is used. The Delaunay tessellation subdivides each chunk into simplex elements. This is implemented using the Matlab function *delaunay*. The convex-hull properties of each of the simplex elements can then be used to determine if a geometry point lies within any of the simplex elements and subsequently within that lattice chunk.

Sequential quadratic programming (SQP) is used to solve the constrained optimisation problem of calculating the local co-ordinates of each geometry point. An initial estimate of the local co-ordinate of [0·5, 0·5, 0·5] is used to start the optimisation. This is the same seed point as used by both Lamousin and Waggenspack[8] and Coquillart[9], neither of whom found any cases that did not converge.

The EFFD lattice is directly manipulated using a slight alteration to the explicit direct manipulation solution Hu *et al*[11] developed. When freezing the initial lattice, the local co-ordinates of the direct manipulation points are calculated in the same manner as the geometry points, i.e., first the chunk they reside within is found and then a numerical search of the volume is used. During this process each direct manipulation point is tagged with a flag to the chunk it resides in. This speeds up the FFD routine as the chunk search process is only executed once. Once the chunks that the direct manipulation points reside in are known, the Hu *et al* explicit direct manipulation solution is used to determine the chunk inner lattice node deformations required. The chunk corner nodes of the outer lattice are then calculated from the inner lattice corner node positions.

In Sections 3 and 5, details are given of how the DM-EFFD method is integrated with a CAD geometry into an optimisation framework.

3.0 CAD INTEGRATION

The ability to maintain a CAD geometry within an optimisation framework underpins the current work. Although CAD changes can be carried out by direct manipulation of the NURBS controls points within the CAD model, here this is achieved by integrating the FFD techniques set out in Section 2 alongside the chosen CAD package. The advantage is that the manipulations carried out can be directly targeted at the design changes required rather than the geometry construction process used to define the initial geometry itself. The CAD software used throughout this study is CATIA

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

V5, although the methodologies presented should be applicable with any of the major CAD packages/geometry kernels. Attempts to use CAD directly in optimisation are not new of course, for example Fudge *et al*[12] produced both a CAD-based and a CAD free optimisation system. Fudge *et al* communicated with CATIA V5 through the use of CAPRI (the computational analysis programming interface). This study does not use this middle-ware but communicates with the CAD software through internal CAD functions/methods.

To integrate the FFD techniques with a CAD package it is necessary to be able to manipulate the baseline surface construction points. However, within the CATIA package while the surface construction points are available within the Surface Sculptor workbench, this approach cannot be automated and hence negates its usefulness for integration with an optimisation process. It is therefore necessary to identify alternative methods of CAD integration. At the start of this study two possible options were identified. Either the surface can be exported from CATIA using the initial graphics exchange specification (IGES) standard[†], through which access can be gained to the surface construction points, or if the surface is created by lofting through spline profiles, CATIA design tables can be used to gain access to the spline construction points. In this section the two identified methods are considered along with the practicalities of utilising them.

3.1 CATIA design tables

CATIA design tables allow the values of CATIA geometry constraints to be linked with an external file. This can take the form of an Excel sheet or a tab delimited text file. The design table can be edited externally and the design can then be updated with the new values. Any CATIA constraints applied within a CATIA model can be linked with a design table, although typically design tables are used with numerical constraints.

Although CATIA design tables cannot be used to export surface construction points from CATIA, they can still be used to integrate FFD methods. If the surface to be deformed/optimised is created using the 'loft' command, parameterising the loft profiles in turn parameterises the desired surface. It is commonplace for lofting profiles to consist of splines constructed through points. If the position of the spline construction points are constrained, a design table can be used to export the constrained co-ordinates. This re-parameterisation is particularly useful if there is a large number of lofting profiles as the number of construction point co-ordinates will be too large to use directly as design variables.

The method of constructing the loft profiles is particularly important when using this method. The loft profiles can be constructed either within the CATIA global environment using three-dimensional points, or within a CATIA sketch using two-dimensional points. A CATIA sketch is a planar two dimensional environment with an independent origin. There are two reasons why this is significant when linking the FFD methods. First, any point co-ordinates use the sketch origin as the reference point. It is therefore important that the sketch origin is the projection of the global origin onto the sketch plane. The second reason why profile construction using a CATIA sketch is significant is that any profile constructed using a CATIA sketch can only be deformed within the sketch plane. This has to be taken into account when setting up any subsequent FFD operations. A three-dimensional lattice can be used to encapsulate all the profiles. However, the lattice needs to be aligned with the sketch planes. Additionally, lattice node movements need to be restricted to lie within these planes. As an alternative to using a three-dimensional lattice, it would also be possible to use a sequence of two-dimensional FFD operations in parallel (one for each loft profile). This would eliminate the restrictions on the lattice node movements. The disadvantage is that this would produce an increase in the number of design variables.

[†] The Initial Graphics Exchange Specification (IGES) defines a neutral data format that allows the digital exchange of information among Computer-aided design (CAD) systems. The official title of IGES is Digital Representation for Communication of Product Definition Data, first published in January, 1980 by the U.S. National Bureau of Standards as NBSIR 80-1978.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

If the geometry is constructed via only two lofting profiles, such as the geometry used by Menzel *et al*[13], it is more efficient to use two two-dimensional FFD parameterisations. However if the construction consists of more than two profiles, then using a 3D lattice allows any deformation to be smoothly spread over the construction profiles, whereas the use of 2D FFD produces independent deformations.

The main limitation of this type of parameterisation is that the design process allows no control over the surface between the lofting profiles. Splines can be used as guide curves between lofting profiles, but in practice the limitations on the degree of curvature that the guide curves can have limits any usefulness except to maintain a smooth edge transition. The restrictions on the 3D lattice node movement limits the type of geometries that it is possible to create. As will be seen this limitation is addressed with the IGES file integration approach. An advantage that design table integration has over the use of IGES files is that one design table can contain numerous designs, whereas an individual IGES file is needed for each design. Each row of the design table is able to contain the co-ordinates of all the spline construction points within a design. The advantage of this is that it enables the engineer to quickly visually compare the types of designs that the optimisation process is generating.

3.2 IGES integration

The IGES standard is a commonly used data format that enables the exchange of information from one CAD system to another. IGES files take the form of text files in ASCII format. The IGES format can be exploited in order to integrate the CAD package with the FFD methods. Within the CAD software just the surface to be optimised can be exclusively exported in the IGES format. Within CATIA this is achieved by hiding all but the target surface: only geometry in the 'visible' space is exported. Using knowledge of the IGES structure, a script can be written to read an IGES file and extract the information relating to any rational B-spline surfaces. Within the IGES format, rational B-spline surfaces take the following form;

$$X(s,t) = \frac{\sum_{i=0}^{l}\sum_{j=0}^{m} W(i,j)P(i,j)B_i(s)B_j(t)}{\sum_{i=0}^{l}\sum_{j=0}^{m} W(i,j)B_i(s)B_j(t)} \quad \ldots (9)$$

The information captured by the script developed here comprises first and second knot sequences, weights and control point co-ordinates. Using Equation (9), it is possible to recreate the surfaces within an external program such as Matlab, however this is not necessary to integrate the FFD methods. Deformation of the surface can be achieved by using a three-dimensional lattice to encapsulate the surface construction points. In this way only the surface construction point co-ordinates are altered. Neither the construction point weights nor the knot vectors are considered. As the FFD lattice is deformed the construction points move. The resulting construction point positions can be written back to the IGES file using a similar script to the IGES import script described above. The use of FFD techniques to parameterise the surface construction points in this manner is similar to how the design table CAD integration method parameterises the spline construction points. If the IGES files are used in this way with a lofted surface, it is essentially equivalent to increasing the number of lofting profiles and thereby increasing the amount of control of the surface.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

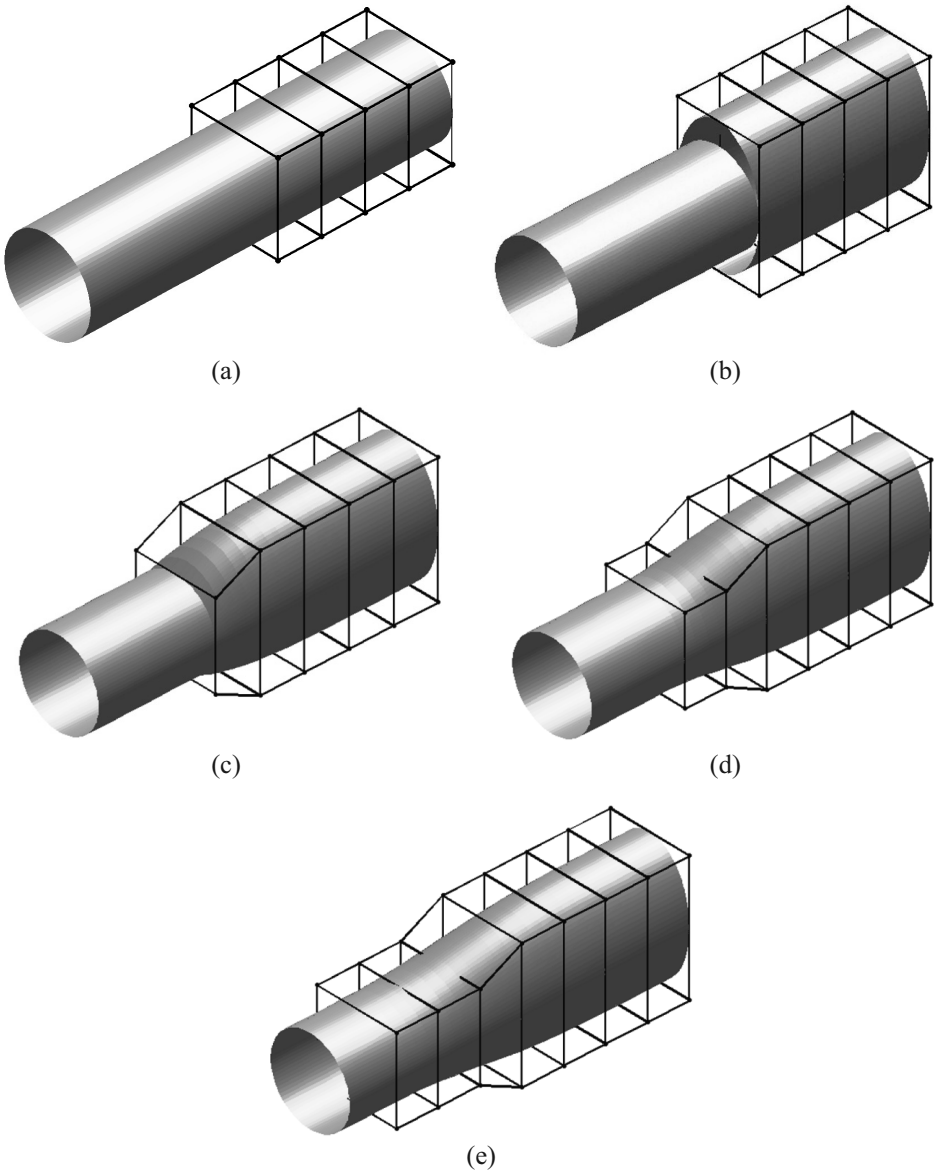


Figure 2. Continuity control – continuity over the FFD lattice boundary is achieved by fixing the lattice node on k adjacent planes to the boundary. (a) Initial geometry (b) C^{-1} boundary (c) C^{0} boundary (d) C^{1} boundary (e) C^{2} boundary.

The limitations on the lattice node movements inherent in the design table integration method are not present when using IGES files, as the surface construction points are free to move within the 3D space. This enables a greater range of geometries to be produced. However, maintaining continuity with the undeformed CAD surfaces is more complicated and boundary node movement constraints are needed.

The main advantage of this method over the design table integration is that the deformed surface is independent of the construction method. The FFD lattice can also be laid over CAD surface joins

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

allowing the optimisation of arbitrary surface selections. However the more IGES surfaces are used, the more complicated it becomes to manipulate the data set. It is also important to note that for ruled surfaces it is the boundary curves that need to be parameterised, not the surface construction points themselves.

Of the two methods outlined, it is recommended that for simple geometries built with two spline profiles the design table integration method coupled with 2D FFD parameterisation is the most appropriate solution. For complex geometries the IGES integration with 3D FFD techniques provides a more flexible representation, and therefore a more successful parameterisation. It is notable that both integrations can be used with any of the FFD techniques discussed in Section 2 and that the selection of the most appropriate method is problem specific.

4.0 CONTINUITY

An important aspect when considering a local shape parameterisation technique is its ability to maintain continuity between the surface to be optimised and any adjoining surfaces. Sederberg and Parry[6] showed that the conditions required to maintain C_k continuity over the boundary are that the lattice nodes on k planes adjacent to the boundary are suitably controlled. Figure 2 shows an example of how this can be adopted to achieve a C^2 continuous connection along one co-ordinate direction. A simple base geometry of a cylinder is deformed to enlarge the diameter of one half (see Fig. 2(a)). Figure 2(b) shows the simplest FFD lattice to achieve this, but disregards the boundary continuity and hence a separation occurs. Figures 2(c)-(e) show the FFD lattices required to produce C^0 to C^2 continuous boundaries and the resulting geometries. To achieve this a plane of undeformed control points is added to each FFD grid respectively. This is equivalent to appending the original FFD lattice (here termed the 'active' lattice), with another undeformed lattice, (here termed the 'continuity' lattice), at the boundary requiring continuity control. Obviously this method can be used to achieve continuous boundaries with surfaces to all faces of the FFD lattice.

5.0 CASE STUDY – AERODYNAMIC OPTIMISATION

In order to examine the practicalities of implementing the proposed FFD parameterisation and CAD integration methods, the set-up and initial search of a simple case study is presented. The aim of this study is the aerodynamic optimisation of a fairing placed around the wing-fuselage junction (here termed the 'wing-fillet') of a typical transonic passenger aircraft. This case study has been chosen as it requires the manipulation of a complex geometry with numerous geometrical constraints to test the proposed methodology. It is also an area of the flow field within which improvements may be possible. It must be stressed however, that relatively modest size meshes have been used in the CFD analysis which has been of a simplified model. This has enabled a useful number of design geometries to be considered within a modest computational budget. It is fully accepted that in commercial work, CFD cell counts of 100m and more would quite normally be used to adequately resolve wing-fuselage junction drag effects. Throughout the aim here has not been to come up with optimised geometries *per se*, rather to demonstrate a process that allows FFD schemes to be closely coupled to CAD which in turn yield geometries compatible with automated meshing and analysis. Given more powerful computational facilities it would be a simple matter to invoke far more detailed CFD meshes over the full geometry and thus generate more accurate flow regimes.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

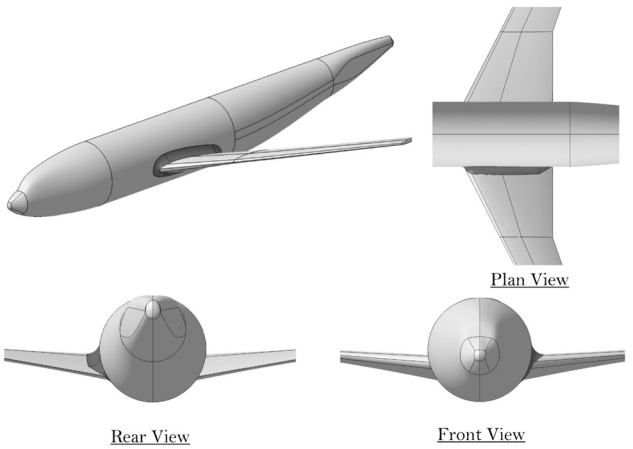


Figure 3. Wing-body configuration of the DLR-F6 geometry with a simple circular arc wing-fillet shown shaded.

Table 1
DLR-F6 geometry details

Property	Value
$S/2$	72·700mm^2
c	141·2mm
$b/2$	585·6mm
AR	9·5
X_{ref}	157·9mm
Y_{ref}	0mm
Z_{ref}	–33·9mm
M_{cruise}	0·75
CL_{cruise}	0·5
Re_{cruise}	3×10^6

5.1 Optimisation objectives and constraints

The case study aims to minimise the total drag experienced by the aircraft while maintaining a fixed lift by only manipulating the shape of the wing-fillet. Each design is subject to the following aerodynamic constraints:

1. Angle-of-attack, $AoA = 1°$,
2. Mach number, M = 0·75.

These constraints are the design cruise conditions for the unfaired configuration of the selected geometry (see Section V.B). Fixing the angle-of-attack and requiring the lift to also remain unchanged of course drives the fairing to be neutral with regard to its impact on lift and just to reduce interference drag. Geometric constraints are applied to the problem via the upper and lower bounds of the design variables such that:

$$0 \leq \{Dvar\} \leq 1, \quad \ldots (10)$$

where *Dvar* defines a vector of normalised design variables.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

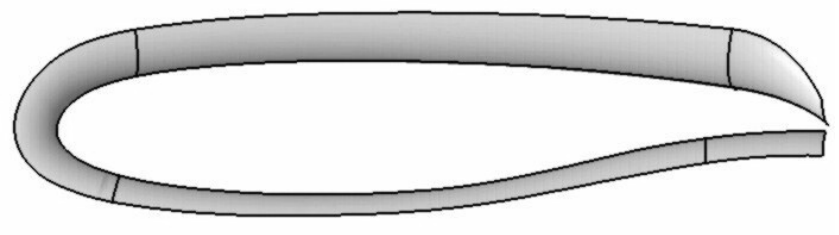


Figure 4. Initial blend shape consisting of five separate sections.

5.2 Geometry

The base geometry selected for this study is the wing-body configuration of the DLR-F6 as used in the AIAA second drag prediction workshop[14] (see Fig. 3). The DLR-F6 geometry was selected as it represents a fairly modern aircraft geometry for which experimental wind tunnel data is widely available. It represents a wide-body aircraft of the Airbus type. The base CATIA model was constructed by reconfiguring the IGES model made publicly available by ONERA for the second drag prediction workshop to the wing body configuration. The nacelles and pylons were removed to simplify the CFD analysis within the optimisation process. The DLR-F6 base geometry does not contain any type of wing-fuselage junction fairing. The properties of the geometry are shown in Table 1.

Raymer states that: "for improved aerodynamic efficiency, the wing-fuselage connection of many aircraft is smoothly blended using a wing-fillet. A wing-fillet is generally defined by a circular arc of varying radius, tangent to both the wing and fuselage[15]". The combination of the initial wing-fillet geometry and deformation method needs to be able to generate such a shape whilst maintaining the flexibility to generate new radical geometries. As an initial fillet geometry a circular arc of constant radius is used, which can also be seen in Fig. 3. As discussed in Section 3, the construction

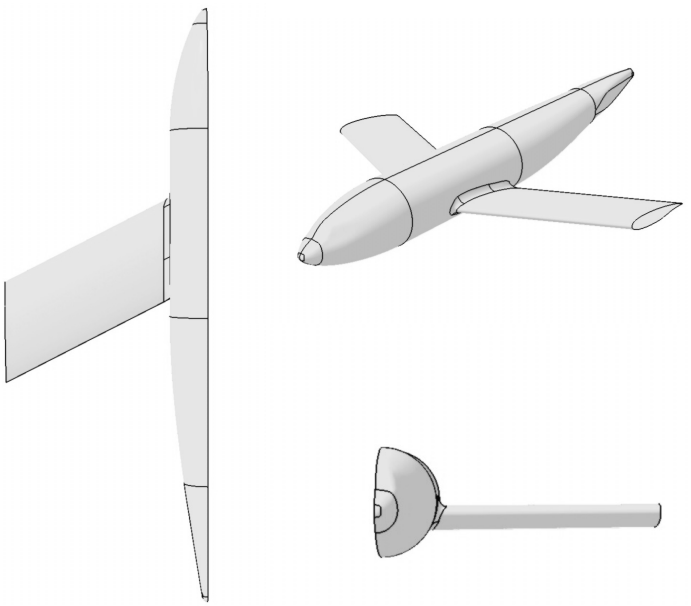


Figure 5. DLR-F6 geometry with simplified wing.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

method of the wing-fillet has an impact on the most applicable integration method and consequently on the outcome of the optimisation study. To create this baseline fillet geometry, the CATIA 'loft' function was selected as the most appropriate method.

In order to generate the flexible representation needed for optimisation, the wing-fillet was constructed from surfaces superimposed on top of the base geometry. This enables a greater range of deformations than the alternative of embedding the wing-fillet within the base geometry. It was decided to construct the wing-fillet in five distinct sections to allow independent control of the upper and lower surfaces and the leading and trailing edges. This also improves the resilience of the geometry reconstruction processes – in any geometry manipulation scheme it is always possible to produce nonsensical shapes and this breakdown of the problem helps minimise such problems. It also increases the flexibility of the design and enables a more detailed individual analysis of each section if necessary. The five separate surfaces are as follows: the upper surface, lower surface, leading edge, upper surface trailing edge and lower surface trailing edge, see Fig. 4.

The wing-fillet is constructed using spline profiles that are equally spaced around the wing-fuselage junction line and are normal to the wing-surface. The spline profiles are each defined by four construction points and are constrained to be coincident and tangential to both the wing-surface and the plane tangential to the fuselage. The initial spline construction points are set to lie on an arc of constant radius, $R = 5$mm (model scale). The wing-fillet surfaces are created by lofting through the spline profiles using two guide splines to ensure that connection with the wing and fuselage is achieved. The two guide curves are created by interpolating the profile start and end points, respectively, using the appropriate surface as a basis. This ensures a water-tight surface is always achieved.

5.3 Simplified model

After completing an initial CFD assessment study (See Section 5.8) on the DLR-F6 model which showed that at least six million cells would be needed to adequately resolve the flow fields, with 11m needed to gain full convergence, it was clear that any subsequent optimisation study would be limited by the CFD evaluation time and so it was therefore decided to develop a simplified model to reduce the CFD costs. The simplified model consists of a modified wing attached to the DLR-F6 fuselage. The fuselage is unaltered to maintain realistic boundary-layer growth, which is a significant factor in the fluid behaviour in the wing-fuselage junction region. The new wing has been created with no twist, taper or dihedral (see Fig. 5) and is generated by extruding the original wing profile at the wing-fuselage junction to a distance of twice the section chord length. During the CFD process a mirror plane is placed at the wing tip to eliminate any wing-tip effects[‡]. The refined region of the CFD mesh normally needed to capture wing tip fluid behaviour is then redundant and can be omitted from the simplified model CFD mesh. As a consequence of simplifying the model and adopting the intermediate meshing parameters from the assessment study, the size of the CFD mesh required following mesh independency checks is approximately 3·3m cells, producing a significant reduction in the computing time needed. The mesh in the vicinity of the wing-fuselage junction is essentially unchanged by this simplification and this provides a balance between detail and speed of the optimisation study to conduct initial studies.

5.4 Geometry parameterisation

To begin with only the wing-fillet upper surface has been manipulated. This enables the outlined methods to be examined without the added complication of the geometry restrictions that need to

[‡]The flow becomes parallel to the inflow at this plane because, in effect, a 'ghost' body is placed an equal distance beyond the aircraft that enforces the required symmetry.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

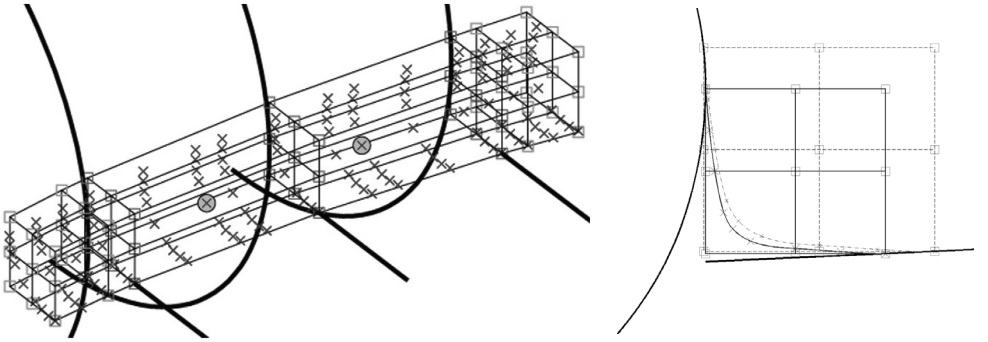


Figure 6. (a) Initial 3D lattice with wing upper surface embedded and constraint curves (× wing-fillet upper surface IGES surface construction points; □ FFD lattice nodes; ○ direct manipulation points); (b) *x*, *z* section at one active lattice plane showing lattice, lattice node and wing-fillet positions before (solid) and after (dashed) FFD shape change.

be applied to the leading and trailing edges to maintain continuity. This initial study selects a relatively coarse active lattice with three planes in each of the *x*, *y* and *z* directions. The active lattice is appended at either end in the *y*-direction by a $3 \times 3 \times 3$ continuity lattice (see Fig. 6). As discussed in Section 2, the nodes of the continuity lattices are fixed in the 3D space in order to maintain C^2 continuity with the other wing-fillet surfaces. To ensure a continuous connection between the fuselage, wing and wing-fillet, the lattice nodes on these boundaries' movements are suitably restricted. To achieve this, the fuselage main section and wing surfaces are generated within Matlab and are used to create boundary curves on the surfaces that the FFD lattice boundary node movements are restricted to.

Figure 6 shows the initial FFD lattice with the appropriate boundary curves (bold curves). The fuselage boundary curve is the 2D lofting profile used to produce the fuselage main section. The wing boundary curve is produced by interpolating points projected onto the wing surface to ensure full closure of the distorted surface. The design variables are associated with the upper left and lower right boundary lattice nodes on each lattice plane and are the distances along the boundary curves. Note that each of these nodes is aligned with the initial position of the wing-fillet patch edge and the final target position for each node is calculated via the parametric equations of the appropriate boundary curve. As well as maintaining continuity, this method also reduces the number of design variables for the problem. The final deformed geometry produced by this process is then always fully closed and smoothly continuous between the wing and fuselage, meaning that no geometry-healing processes are required during mesh generation.

In addition to the position of the boundary nodes along their constraining curves, the other design variables for this study are the co-ordinates of the target position of direct manipulation points placed on the wing-fillet surface. The position and number of direct manipulation points are also part of the FFD parameter survey. Initially two direct manipulation points are used that are equally spaced along the mid-chord of the surface. The direct manipulation is free to move within 3D space. The movement of the direct manipulation points is restricted by the design variable range set for each co-ordinate during the optimisation process. These ranges are selected such that, in the *z*-direction, the direct manipulation points cannot breach the boundary of the adjacent FFD lattice chunk. This is a restriction imposed by the direct manipulation technique. In the *x* and *y* directions the movement is limited by the fuselage and wing surfaces and hence the range minima are set as such.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

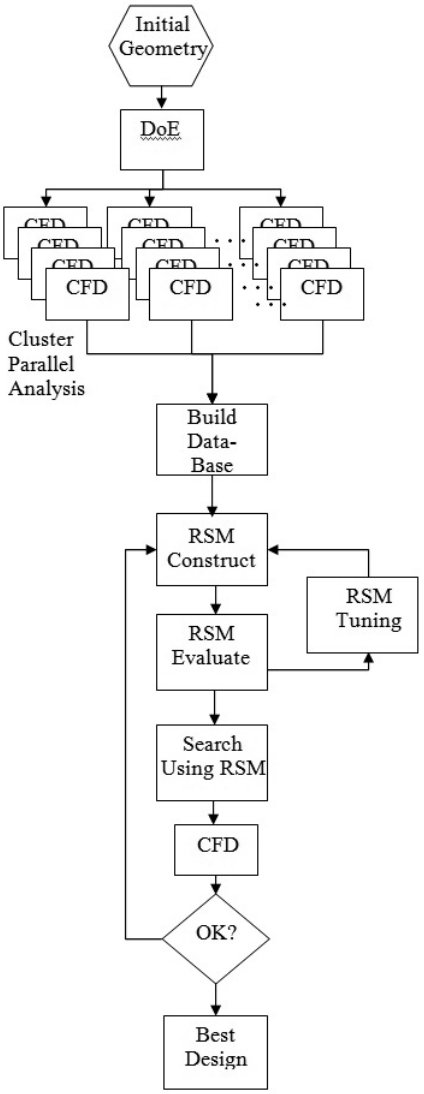


Figure 7. Response Surface Method (RSM) based optimisation strategy.

The initial study thus has a total of 12 design variables (six boundary node positions and three for each direct manipulation point). The six boundary node positions enable the size of the wing-fillet along the surface to vary. The direct manipulation points enable bumps/divots to be induced along the surface.

Using this type of FFD parameterisation, each optimisation design point analysis consists of the following procedures: decoding of the design variables, calculation of the inner lattice node deformation, calculation of the outer lattice node deformation, deformation of the geometry, CATIA model update, CFD mesh generation and finally CFD analysis. The FFD-based geometry manipulation stages of this process are extremely rapid when compared to the mesh generation which in turn is much faster than CFD analysis.

5.5 CAD communication

CATIA V5 is an object-orientated CAD package that can expose a large number of objects to external applications. CAA-API is the interface through which access to these objects can be gained. This enables external scripts or applications to be produced that can control most CATIA objects. CAA-API can be accessed through C++, JAVA and the Visual Basic programming languages with assistance from CAA-RADE (Rapid Application Development Environment).

In order to integrate CATIA within an optimisation loop it must be run in batch mode. Using Visual Basic to control CATIA requires CATscript files which set out the changes required. For this study a CATscript file has been developed that opens an existing product (assembly) containing the DLR-F6 geometry as a component. The deformed IGES wing-fillet is then imported as a new component, the CATIA 'Split' command removes the excess wing and fuselage surfaces and finally the entire product is exported as an IGES file ready for mesh generation. The excess surface is the fuselage and wing surface that the new wing-fillet is laid over.

5.6 Optimisation strategy

The geometry parameterisation technique outlined above is integrated within the OPTIONS optimisation framework[16]. The optimisation strategy used in this study is shown in Fig. 7. The

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

strategy incorporates a design of experiments (DoE) to select an initial database of designs such that it maximises the ability of this database to approximate the design space encompassing all design points. Many types of DoE techniques are available with varying space filling properties. Here the Latin Hypercubes method is used as it has been shown to provide a near uniformly distributed set of control points with a degree of stochastic positioning. On top of this initial database a response surface model is created. This is a type of curve fitting used to approximate the design space. Evaluation of the response surface can then be used in lieu of the full CFD analysis with a numerical search technique to locate a possible optimum design. However before an optimum design is located the response surface is subjected to a series of updates to ensure the design space is adequately searched and reliable results are produced.

Here the response surface-based optimisation is set up as a two objective problem, with one objective being the reduction of drag while the second is the minimisation of any change in the lift coefficient from its baseline value at an angle-of-attack of 1° (for example, with the simplified design and the baseline fillet this gives a $C_L = 0{\cdot}467$ and C_D is 0·0719). This allows all the CFD calculations to be run at a fixed angle-of-attack while also ensuring that drag reductions are not achieved by simply reducing the overall lift. The optimiser used is the response surface enhanced non-dominated Sorting genetic algorithm version II (NSGA2) described by Voutchkov and Keane1[7]. For the final optimisation described 120 initial geometries were analysed to establish the initial response surfaces followed by a further 90 updates positioned according to the NSGA2 process. Further details of the approach are to be found in the references.

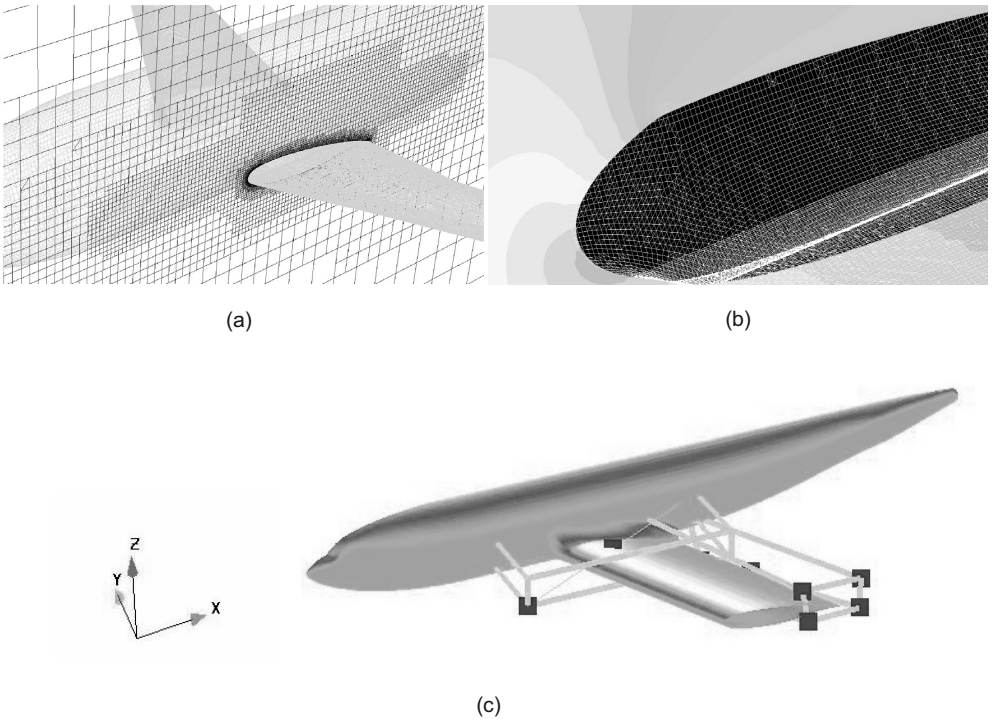


Figure 8. (a) and (b) Harpoon mesh and surface mesh for the simplified DLR-F6 geometry with fillet, (c) Harpoon mesh refinement areas for the simplified DLR-F6 geometry.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

5.7 Mesh generation

The mesh generation package Harpoon v3.0a[18] is used here to produce the CFD mesh for each new design. Harpoon is an automated application, which can produce hex dominant meshes at up to two million cells/minute on P4-based PCs. Although it is possible to use FFD methods to manipulate the CFD mesh, therefore eliminating the need to produce a new CFD mesh for each design, to do this in parallel with using FFD to manipulate the geometry is not trivial. The complication arises as it is necessary to ensure the geometry produced by the CAD package is accurately described by the CFD mesh. In particular the ability to vary the global size of the wing-fillet would be complicated, as this would require an FFD grid to be produced around the entire CFD mesh rather than just the wing-fillet. The speed at which Harpoon is capable of creating an adequate 3D CFD mesh when compared to the time for the CFD analysis is however insignificant for three-dimensional studies. A five million cell mesh can be created in 230 seconds, whereas the CFD solve takes approximately 10 hours. It is therefore simpler to re-mesh the geometry as each variation is created and exported from the CAD-based process.

Harpoon generates the new CFD mesh while running in batch mode from Harpoon configuration files. Matlab is used to update the configuration file for each new design to point to the exported IGES geometry. Within Harpoon, mesh size control is achieved by setting a global base size, surface size, an expansion rate and if necessary a maximum cell size. Harpoon creates a mesh by initially meshing the geometry surface using the surface base sized elements and then grows the volume mesh away from this surface towards the outer boundaries expanding the element size as it is grown, see Fig. 8(a) and (b). If a boundary layer is required it is applied after the initial volume mesh is created. The boundary layer is grown from the surfaces, pushing the existing volume mesh out. Since this distorts the existing mesh, Harpoon restricts the boundary-layer height to a maximum of three times the initial local cell height, however it is preferable to keep it within two times the local cell height. If more detail is required within a certain area a refinement region can be used to increase the local mesh density. For this case study refinement regions are placed around the trailing edge and more importantly the wing-fuselage junction, see Fig. 8(c). It is expected that the drag changes seen between designs is likely to be relatively small. It is therefore important that the flow within this region is captured with as much detail as possible. The refinement region around the fuselage-wing junction is sized so as to enclose any of the possible wing-fillet designs.

5.8 CFD Analysis

All CFD simulations within this study are executed using the commercial code Fluent 6.3.26 and involve solving the three-dimensional steady-state Reynolds averaged Navier-Stokes equations. The turbulence model selected for this problem is the Spalart-Allmaras (SA) one equation model. This was chosen as it has been proven to give accurate results for external flow problems. More details of the turbulence model can be found in Spalart and Allmaras[19]. The SA turbulence model was also one of the models examined during the second drag prediction workshop (12 out of the 29 submitted cases used SA). The boundary conditions used for the full DLR-F6 geometry analyses are set to simulate the design cruise conditions as already shown in Table 1 using Fluent far-field pressure boundary conditions. Details of the Fluent setup used are given in Table 2.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

Table 2
Details of the initial Fluent setup

Solver	Coupled density based with energy equation
Turbulence model	Spalart-Allmaras single transport equation
Flow medium	Air – Ideal gas
Solver Courant number	5
Discretisation scheme	second order
Flow mach number	0·75
Reynolds number	3×10^6

To establish the effects of mesh dependency three meshes sizes were studied using the full, unfaired geometry at a fixed lift coefficient and compared to the published experimental results made available by the second drag prediction workshop. Table 3 contains details of the mesh sizes and associated boundary layers. Significant regions of boundary layer separation at the upper wing surface trailing edge were reported during the wind tunnel testing. A large separation bubble was also seen at the rear wing-fuselage junction. The accuracy of any CFD analysis on this geometry is therefore reliant on its ability to predict this type of flow behaviour. This ability is largely dependent on the mesh density/quality and to a lesser extent the turbulence model selected.

Of particular interest from the second drag prediction workshop is the first case study[14]. This consisted of a single point mesh dependency study at cruise conditions (see Table 1 for cruise condition details). Table 4 summarises the lift and drag data found during the DPW-2 and from the three CFD meshes examined here. It is clear that the C_D values achieved by all the meshes are close to the experimental value. The DPW showed that generally CFD under predicted *AoA* and C_D. With respect to *AoA* this is also true for the three analysis meshes considered here, whilst the C_D values are slightly over predicted by a few drag counts.

On comparing the C_P profiles for the wing-lower surface achieved by each mesh it was found that generally the CFD solutions were in good agreement with the experimental data. This is expected, given that during the wind tunnel testing, no major flow features were seen on the lower surface[‡‡]. The upper wing surface C_P profiles did not show such a good correlation with

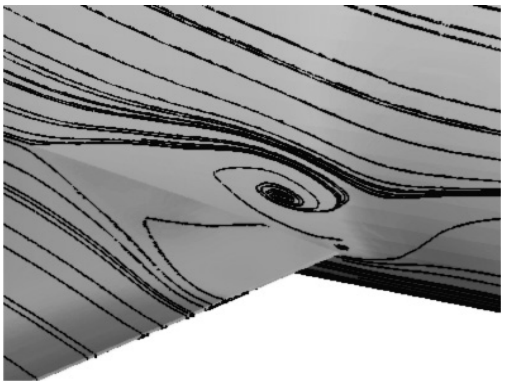


Figure 9. Surface restricted flow stream lines predicted by coarse mesh.

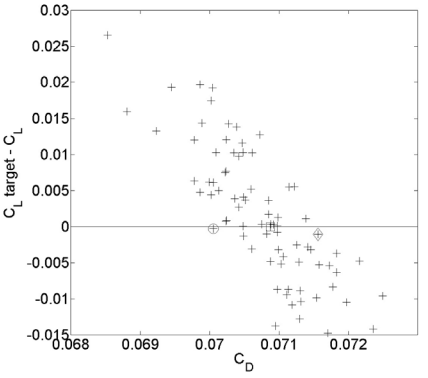


Figure 10. Results of an 80 point DoE over the upper surface patch of the simplified wing; base-line fillet design marked □, design 9 marked ○, design 39 marked ◊.

[‡‡]The flow becomes parallel to the inflow at this plane because, in effect, a 'ghost' body is placed an equal distance beyond the aircraft that enforces the required symmetry.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

Table 4
Experimental and CFD Data summary for varying mesh qualities

Parameter	**Experimental Data**	
AoA, degrees	0·52	
C_L	0·500	
C_D	0·0295	
Parameter	**DPW Av. Coarse**	**Coarse**
AoA, degrees	0·131	0·091
C_L	0·5001	0·4990
C_D	0·02985	0·031
number of Nodes, *N*	1·3 million	1·8 million
Parameter	**DPW Av. Intermediate**	**Intermediate**
AoA, degrees	0·161	0·091
C_L	0·5000	0·4999
C_D	0·02902	0·032
number of Nodes, *N*	4 million	6.0 million
Parameter	**DPW Av. Fine**	**Fine**
AoA, degrees	0·1529	0·092
C_L	0·4999	0·5001
C_D	0·02883	0·0321
number of Nodes, *N*	12 million	11·2 million

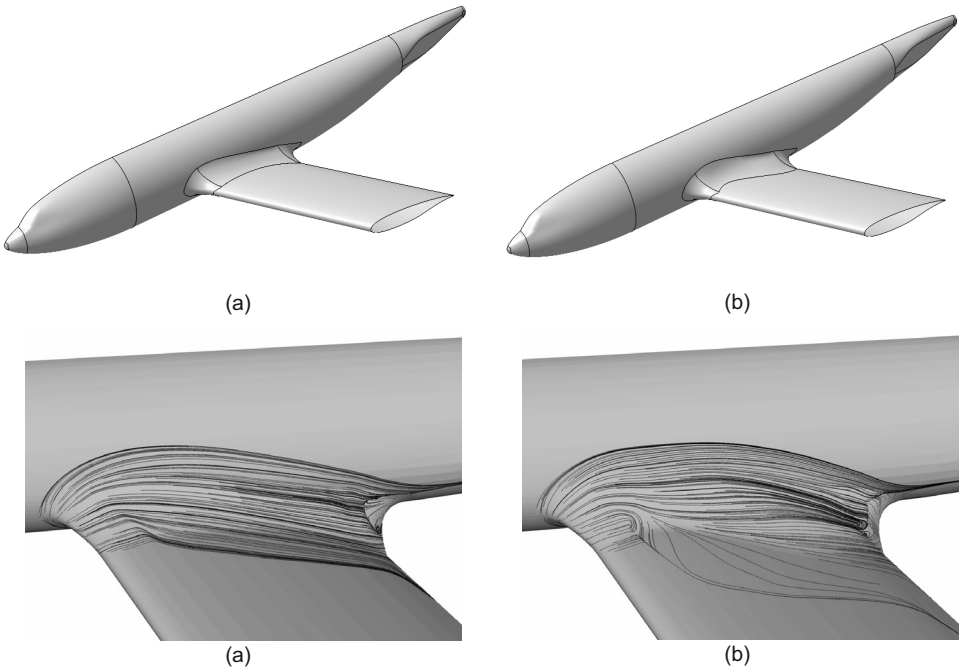


Figure 11. geometry and flow fields for designs 9 and 39 of the upper surface patch DoE using the simplified wing.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

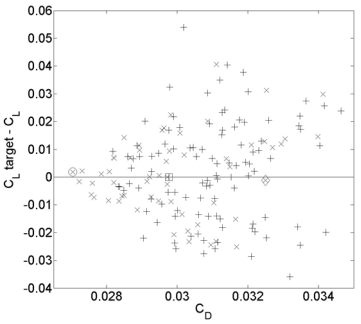


Figure 12. Results of an 125 point DoE (+) followed by 90 updates (×) over the entire wing-fuselage junction of the original DLR-F6 geometry; base-line fillet design marked □, design 146 marked ○, design 161 marked ◊.

the experimental data and the CFD analysis did not capture the position of the shock on the upper surface with complete accuracy. The largest deviation from the experimental data was seen on semi-span profile $\eta = 0{\cdot}150$ towards the upper surface trailing edge. This is within the predicted separation bubble region. During the DPW similar C_P profiles at $\eta = 0{\cdot}150$ were achieved by Rumsey *et al*[20] and Scheidegger and Stuckert[21]. It is thought that the reason for the deviation is the variation in the size of the separation bubble predicted by the CFD code. Figure 9 shows the separation bubble predicted by the coarse mesh analysis. Rumsey *et al* and Scheidegger and Stuckert both saw this deviation reduce with an increase in mesh size. With respect to the mesh resolution tested here, little difference could be seen between the profiles predicted for each mesh size, and this is also seen in the C_D values predicted.

Overall the C_D values achieved are close enough to the experimental value to suggest that the CFD setup with the coarse mesh would produce results realistic enough for the purposes of the optimisation studies presented here, particularly since the main aim of the work reported is to demonstrate a geometry manipulation process rather than to generate optimal wing fairings. In fact, to aid in dealing with the new geometries being generated the final optimisation studies have been run with meshes with around four million cells, lying somewhere between the coarse and intermediate meshes, but with $y+$ values in the range 30-80.

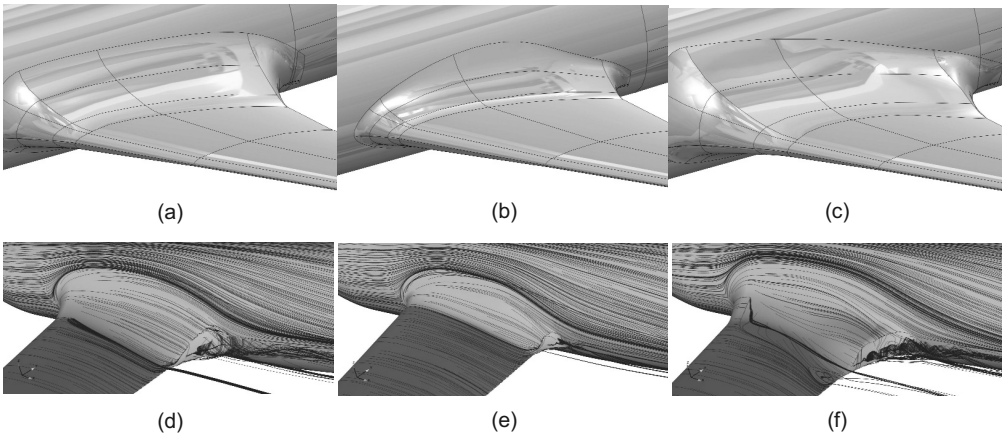


Figure 13. geometry and flow fields for designs 1, 146 and 161 of the entire wing-fuselage junction optimisation using the original DLR-F6 geometry (it should be noted that although a series of lines are shown on these CAD images, the patches are, in fact, single lofts in CATIA with a series of guide curves).

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

Table 3
Mesh dependency details

Mesh	Coarse	Intermediate	Fine
number of Nodes, N	1·8 million	6·0 million	11·2 million
Boundary layer initial height	0·03mm	0·03mm	0·01mm
Number of Boundary cells	5	5	5
Typical value of y^+	30-80	5	1
Boundary layer growth rate	1·22	1·22	1·22
Boundary layer total height	0·081mm	0·081mm	0·027mm
Boundary box size	106c	106c	106c

5.9 Initial results

Figure 10 shows the distribution of results obtained during an initial 80 point DoE where only the upper surface fairing patch is allowed to change using 12 design variables as already described in Section 5.4. The results display a variation in CD of approximately 30 drag counts, as well as a maximum distance of 0·025 from the target lift coefficient. This distribution contains a Pareto-like front upon which the best performing designs reside[‡†]. There is an obvious trade-off between lift and drag as might be expected. Design 9 marked on the figure lies on the Pareto front and also shows minimal variation in lift from the target – it offers a reduction of eight drag counts at fixed lift simply by changing the shape of the upper part of the fairing; moreover this has been achieved from the initial DoE without any optimisation. Design 39 also marked on the figure again shows minimal variation in lift but now has significantly worse drag than the base-line design. Figure 11 illustrates these geometries and their respective flow fields and it is clear that producing a smooth upper surface fillet is important to controlling the separation bubble at the rear of the wing-fuselage junction and hence the overall drag, although the overall variations in drag caused by just varying the upper surface fillet at fixed lift are not great.

5.10 Optimisation results

Having set up an automated process that allows geometries to be changed using CAD-linked FFD, meshed and then solved via Fluent attention is finally returned to the full DLR-F6 geometry. Figure 12 shows the results of applying the NSGA2 optimisation strategy to the DLR-F6 geometry without any simplification, i.e., including twist, taper, camber, tip flows, etc. 12 design variables have again been used but now spread over the entire wing-fillet, giving less control in each patch but more overall control of the shape – one could use more variables but this makes any optimisation harder since more variables have to be controlled. Notice that the NSGA2 optimiser has added designs to the original DoE points that significantly reduce the drag coefficient while concentrating on designs with fixed lift coefficients. In this case design 146 shows an improvement over the base-line fillet design of 27 drag counts with a lift coefficient deviation of only 0·0018. The final C_D is 0·027 compared to that for the base-line fillet of 0·030 and that for the un-faired geometry of 0·032. Design 161 shows one of the poor fillets but one which still gives the desired lift coefficient – in this case the C_D is 0·033.

[‡†] A Pareto set of designs is one whose members are all optimal in some sense, but where the relative weighting between the competing goals is yet to be finally fixed (see for example, Fonseca and Fleming[22]. More formally, a Pareto set of designs contains systems that are sufficiently optimiSed that, to improve the performance of any set member in any one goal function, its performance in at least one of the other functions must be made worse. Moreover, the designs in the set are said to be 'non-dominated' in that no other set member exceeds a given design's performance in all goals. It is customary to illustrate a Pareto set by plotting the performance of its members against each goal function as in Figure 10.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

Clearly, adding a fairing tends to reduce drag, but adding a well optimised fairing more than doubles this gain, while a very poor fairing can in fact be worse than no fairing at all. Figure 13 illustrates the geometries and flow fields for the base-line fillet and for designs 146 and 161. Design 146 gives a significant drag reduction while design 161 clearly makes things notably worse. It is apparent from these results that the key to reducing drag is control of the separation bubble near the trailing edge of the wing, where the wing meets the fuselage, a point made by many authors during the drag prediction workshop – design 146 has almost completely removed this bubble which is why it shows such a good drag improvement. Conversely, design 161 not only has a large separation bubble at the rear of the wing-fuselage junction, but the poor shape of the fillet at the leading edge also leads to further separations outboard along the span of the wing. It is also worth noting that the good design has a far from obvious fillet shape – one that would not easily be derived without the use of formal optimisation-based search schemes. It would no doubt be possible to further refine these designs with longer searches and more design variables – using only 210 analysis runs in total it is extremely unlikely that a fully converged shape has emerged. Nonetheless very worthwhile gains have been achieved.

6.0 SUMMARY AND CONCLUSION

The use of computer aided design (CAD) packages is central to aerospace design processes; however their manipulation during optimisation studies can be difficult, especially if they have not been explicitly constructed with optimisation parameterisations in mind. On the other hand free-form deformation (FFD) tools, while eminently suitable for design optimisation, have mainly been applied to analysis meshes rather than CAD geometry directly. This means that any final optimised shape has to be re-engineered back into the CAD world if it is created using FFD applied to a mesh. The aim of this paper has to been to demonstrate an approach that applies FFD directly to the basic elements of a CAD model so as to allow the flexibility of FFD-based approaches while still tightly coupled to a CAD model. Although only at a preliminary stage and used with strictly limited size CFD meshes, the work presented demonstrates that such an approach can be set up and applied to a realistic design geometry. Here free-form deformation is linked with the CAD package CATIA V5 and used to optimise a complex wing-fuselage junction shape. It has been shown that this method can be successfully integrated within a multi-objective optimisation framework. Details of the optimisation process are given, along with an analysis of the initial results achieved. This case study shows that after just the initial set of CFD runs a significant reduction in CD was achieved which could be further improved using response surface-based search, albeit based on rather limited CFD mesh sizes. Finally, an optimisation study on the full DLR-F6 geometry has shown that FFD optimised wing-fuselage fillets can reduce interference drag by more than twice the amount of simple un-optimised fillets by careful control of separation features.

ACKNOWLEDGEMENTS

The support of Airbus and the EPSRC in funding this work is gratefully acknowledged.

REFERENCES

1. TOAL, D., KEANE, A. and BRESSLOFF, N. Kriging hyperparameter tuning strategies, *AIAA J*, May 2008, **46**, (5), pp 1240-1252.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press

2. SAMAREH, J.A. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization, *AIAA J*, May 2001, **39**, (5), pp 877-884.
3. SARAKINOS, S.S., AMOIRALIS, E. and NIKOLOS, I.K. Exploring freeform deformation capabilities in aerodynamic shape parameterization, 2005, EUROCON 2005 – International conference on 'Computer as a Tool', **1**, pp 535-538, November 2005, Belgrade, Serbia and Montenegro.
4. ANDREOLI, M., JANKA, A. and DESIDERI, J. Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics, November 2003, INRIA Research report 5019, Sophia Antipolis, France.
5. SEDERBERG, T.W. and PARRY, S.R. Free-form deformation of solid geometric models, *ACM SIGGRAPH Computer Graphics*, August 1986, **20**, (4), pp 151-160.
6. GRIESSMAIR, J. and PURGATHOFER, W. Deformations of solids with trivariate B-spline, 1989, Eurographics conference, pp 20-27, September 1989, North-Holland, Hamburg.
7. LAMOUSIN, H.J. and WAGGENSPACK, W.N. NURBS-based free-form deformations, *IEEE Computer Graphics and Applications*, November 1994, **14**, (1), pp 59-65.
8. COQUILLART, S. Extended free-form deformation: a sculpturing tool for 3D geometric modeling, *ACM SIGGRAPH Computer Graphics*, August 1990, **24**, (4), pp 187-196.
9. CLARK, J.H. Parametric curves, surfaces and volumes in computer graphics and computer-aided geometric design, 1981, Technical Report CSL-TR-81-221, Stanford University, Stanford, CA, USA.
10. HSU, W.M., HUGHES, J.F. and KAUFMAN, H. Direct manipulation of free-form deformations, *Computer Graphics*, July 1992, **26**, (2), pp 177-184.
11. HU, S.-M., ZHANG, H., TAI, C.-L. and SUN, J.-G. Direct manipulation of FFD: Efficient explicit solutions and decomposible multiple point constraints, *The Visual Computer*, July 2001, **17**, (6), pp 370-379.
12. FUDGE, D.M., ZINGG, D.W. and HAIMES, R. A CAD-free and a CAD-based geometry control system for aerodynamic shape optimization, *AIAA*, January 2005, **51**, (4), pp 1-15.
13. MENZEL, S., OLHOFER, M. and SENDHOFF, B. Direct manipulation of free form deformation in evolutionary design optimisation, 2006, Parallel Problem Solving from Nature conference (PPSN IX), pp 352-361, September 2006, Reykjavik, Iceland.
14. AIAA, Second drag prediction workshop, June 2003, http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop2/, NASA, Orlando, FL, USA.
15. RAYMER, D.P. *Aircraft Design: A Conceptual Approach*, Fourth edition, 2006, AIAA Education Series, p 159, New York, NY, USA.
16. KEANE, A.J. *The Options Design Exploration System Reference Manual and User GuideVersion B3.*1, 2005, University of Southampton, UK, http://www.soton.ac.uk/~ajk/options.ps, pp 10.1-10.8.
17. VOUTCHKOV, I.I. and KEANE, A.J. Multiobjective optimization using surrogates, 2006, Seventh International Conference on Adaptive Computing in Design and Manufacture, (ACDM 2006), pp 167-175, May 2006, Bristol, UK.
18. BARDWELL, R. Harpoon, 2007, http://www.sharc.co.uk/downloads/guidelines.pdf, Sharc, Manchester, UK.
19. SPALART, P.R. and ALLMARAS, S.R. A one equation turbulence model for aerodynamic flows, 1992, AIAA Paper 92-0439, **1**, pp 5-21.
20. RUMSEY, C., RIVERS, S. and MORRISON, J. Study of CFD variation on transport configurations from the second drag-prediction workshop, *Computers and Fluids*, August 2005, **34**, (7), pp 785-816.
21. SCHEIDEGGER, T. and STUCKERT, G. DLR-F6 Wing-Body-Nacelle Simulations, June 2003, http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop2/pdf/54Scheidegger_dpw2.pdf, NASA, Orlando, FL, USA.
22. FONSECA, C.M. and FLEMING, P.J. An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computing*, **3**, pp 1-16, 1995.

https://doi.org/10.1017/S0001924000007351 Published online by Cambridge University Press