Wing-body junction optimisation with CAD-based parametrisation including a moving intersection

Shenren Xu[a,*], Sebastian Timme[a], Orest Mykhaskiv[b], Jens-Dominik Müller[b]

[a] *University of Liverpool, Liverpool L69 3GH, United Kingdom*
[b] *Queen Mary University of London, London E1 4NS, United Kingdom*

Abstract

Over the past decades significant progress has been made with adjoint computational fluid dynamics solvers, which are an essential part of efficient high-fidelity aerodynamic shape optimisation. Shape parametrisation is much less mature, in particular the field is lacking efficient and automatic CAD-based parametrisation methods. The paper proposes a novel CAD-based parametrisation with CAD in the design loop such that the CAD shape can ultimately serve as a datum surface in multi-disciplinary optimisation. Wing and fuselage are modelled with B-spline surfaces. The intersection line is calculated using an in-house implementation of a B-spline surface modeler and its derivative is efficiently calculated via finite differences. The proposed parametrisation method is applied to the redesign of the wing-fuselage junction of the DLR-F6 model using an adjoint solver based on Reynolds-averaged Navier–Stokes equations. The moving intersection line capability enables the fuselage surface to be deformed and the resulting intersection line to move along the fixed wing during optimisation. The flow separation in the wing-body junction is substantially suppressed by an improved fuselage shape, at the cost of $O(10)$ steady-state flow and adjoint solutions. The proposed parametrisation method represents an important step toward automated CAD-based optimisation for fully-featured aircraft characterised by complex intersecting surfaces.

Keywords: B-spline, intersection, wing-fuselage junction, adjoint, RANS

1. Introduction

Over the past few decades, adjoint-based aerodynamic shape optimisation using Computational Fluid Dynamics (CFD) has been widely applied to the design and optimisation of aircraft [1, 2, 3]. The adjoint approach was proposed in [4, 5] to efficiently compute the design gradient for aerodynamic shape optimisation. The method was later extended to more realistic cases such as multi-element aerofoils and modern transport configurations [6, 7, 8, 9, 10].

The aerodynamic drag of an aircraft mainly consists of pressure/wave drag (due to either flow separation or shock wave), induced drag, skin-friction drag and interference drag [11]. Modern aircraft are usually designed to be free of massive flow separation and thus adjoint based optimisations initially have mainly focused on reducing the wave drag at transonic flow conditions [5]. Induced drag can also be reduced for either planar [12] or nonplanar geometries [13]. To reduce skin-friction drag, the main design philosophy is to ensure that the flow remains laminar for as large an area as possible [14]. Adjoint-based aerofoil optimisation is performed for natural laminar flow in [15], incorporating an iterative laminar-turbulent transition prediction methodology.

Optimisation aimed at the final contributor to the overall drag, interference drag, has been much less studied. An example of the interference drag is the side-of-body separation found at the wing-body junction of the DLR-F6 model [16, 17, 18]. Manual modification to the wing-body junction of DLR-F6 was proposed in [19] and it was later confirmed in wind tunnel experiments that the modification indeed reduces the separation and improves the lift-to-drag ratio [18, 20]. Shape optimisation using the adjoint method on this configuration has been performed by Brezillon et

*Corresponding author
Email address: shenren.xu@liverpool.ac.uk (Shenren Xu)

al. [21], where it was found that to effectively reduce the separation, the most critical region for shape modification is the wing-body junction rather than the wing. However, the mesh-based parametrisation approach combined with free form deformation finds an optimised shape with a narrower fuselage and a distorted wing, apparently not suitable to manufacturing. Recognising that the parametrisation method plays an important role, especially for cases with complex geometries such as the DLR-F6 model, in this work we perform shape optimisation for the DLR-F6 case using a CAD-based parametrisation approach with more practical geometric and flow constraints.

The commonly used mesh-based parametrisation approach uses the coordinates of the computational mesh as design variables, it is thus straightforward to set up and permits a rich design space. However, it also has several drawbacks. First, the optimised shape exists only as a mesh and is not trivial to transfer this to a CAD format for further engineering analysis and/or manufacturing [22]. Secondly, surface sensitivities, i.e. the gradient of the objective function with respect to surface mesh coordinates, are usually highly oscillatory for flows at high Reynolds number and using them directly to perturb the mesh will result in an oscillatory shape. To circumvent this issue, one could either smooth the gradient [23, 24] or the shape [25, 26, 27]. However, the amount of smoothing, which affects how fast the optimum is reached, is difficult to determine a priori. The oscillatory gradient can also be implicitly removed by a using lower-dimensional design space. One typical method to achieve this is to use free form deformation (FFD) technique [28] where a coarse 'lattice' is wrapped around the surface mesh that is subject to optimisation. The surface mesh can be seen as a two-dimensional manifold embedded in the three-dimensional Cartesian space defined by the coarse 'lattice'. The lattice could be either a simple linear box [21] or a more sophisticated form such as a B-spline volume [29]. However, the simplicity of FFD is to some extent offset by its lack of a good control of the local deformation [21].

To overcome the shortcomings of the mesh-based parametrisation method, CAD-based approaches use parameters available with the CAD model as the design variables. One could take the brute-force approach to include the CAD modeller and mesher in the optimisation loop and calculate geometry-to-mesh sensitivity via finite differences [30]. This is very time-consuming as for each design variable an expensive remeshing and interpolation operation needs to be performed. As a alternative to using a full commercial CAD system, one could also build one's own geometry modelling kernel whose derivative then can be more efficiently computed either with automatic differentiation [31, 32], or through finite differences. The CAD geometry then can be used to drive both the aerodynamic and structural analysis and optimisation [33]. We propose a light-weighted CAD-based parametrisation that uses the control points of the B-spline surfaces as the design variables. The perturbation of the intersection line is considered; its derivative with respect to the design variables is computed using finite differences. Instead of remeshing the surface and volume meshes every time the design is perturbed, the surface mesh is automatically synchronised with the geometry via parametric coordinates in the B-Spline surfaces, and the volume mesh is efficiently deformed using the radial basis functions (RBF) approach. The method is applied to modify the DLR-F6 wing-fuselage junction to improve its aerodynamic performance.

The remaining part of the paper is structured as follows. In Section 2, the flow and adjoint solvers, the parametrisation method using B-splines, handling of the intersection line, surface and volume mesh movement and finally the optimiser are explained. The optimisation test case is described in Section 3, with an introduction to the test case, analysis of the baseline flow solution, the formulation of the optimisation problem, followed by the optimisation results. Conclusions are given in Section 4.

2. Governing equations and methodology

2.1. The flow solver

The nonlinear flow solver and the adjoint solver are both part of the DLR-TAU code suite, a CFD software package widely used as a production code in the European aerospace industry as well as a research code for method development [34, 35]. The Reynolds-averaged Navier–Stokes (RANS) equations are solved with a finite-volume discretisation on unstructured grids with various options of spatial and temporal discretisation schemes and turbulence models. In this paper, the mean flow is discretised with the Jameson–Schmidt–Turkel (JST) scheme [36] with matrix dissipation [37]. The Spalart–Allmaras model [38] is discretised using first-order accurate Roe scheme [39]. The nonlinear flow equations are pseudo-time marched using the first-order backward Euler implicit scheme. At each pseudo-time step, agglomeration multigrid is used to accelerate the convergence with lower-upper symmetric-Gauss–Seidel [40, 41] as the multigrid smoother.

2.2. The adjoint solver

The cost function for optimisation $\mathbf{J}$ is a function of both the flow solution $\mathbf{U}$ and the volume mesh $\mathbf{X}$ which is itself a function of the design variable $\boldsymbol{\alpha}$. To compute the design gradient, the cost function is linearised as

$$\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial \mathbf{J}}{\partial \boldsymbol{\alpha}} + \mathbf{v}^T \mathbf{f} \tag{1}$$

where

$$\mathbf{f} := -\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}$$

with $\mathbf{R}$ being the nonlinear residual vector, and $\mathbf{v}$ the solution to the adjoint equation

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \mathbf{v} = \mathbf{g}. \tag{2}$$

The right-hand side in Equation (2) is the linearisation of the cost function with respect to the flow variables

$$\mathbf{g} := \left(\frac{\partial \mathbf{J}}{\partial \mathbf{U}}\right)^T.$$

The adjoint equation only needs to be solved as many times as the number of cost functions. For aerodynamic applications, the cost functions are usually limited to a handful, such as lift, drag and moment, while the number of design variables could be hundreds or even thousands. The adjoint approach thus provides an efficient method of computing the design gradient for aerodynamic shape optimisation.

In the DLR-TAU code, the adjoint equation is solved using the Jacobian-forming Newton–Krylov method. The Jacobian matrix $\partial\mathbf{R}/\partial\mathbf{U}$ is formed, transposed and stored in memory using the existing residual calculation subroutines differentiated by hand. A Krylov solver enhanced by subspace recycling [42, 54] and preconditioned using incomplete lower-upper factorisation is used to iteratively solve the linear system. Once the adjoint solution is computed, it needs to be multiplied with the source term $\mathbf{f}$, which depends on the design surface parametrisation, $\delta\mathbf{X}_s(\boldsymbol{\alpha})$, and the volume mesh deformation method, $\delta\mathbf{X}(\delta\mathbf{X}_s)$. The metric derivative, $\mathbf{f}$, then can be assembled via the chain rule as

$$\mathbf{f} := \frac{\partial \mathbf{R}}{\partial \delta\mathbf{X}} \frac{\partial \delta\mathbf{X}}{\partial \delta\mathbf{X}_s} \frac{\partial \delta\mathbf{X}_s}{\partial \boldsymbol{\alpha}}. \tag{3}$$

2.3. Surface parametrisation using B-spline

The geometry shown in Figure 1 is retrieved from the second AIAA CFD Drag Prediction Workshop (DPW II) website. The wing-body model consists of a large number of B-spline surfaces. Parametrisation using B-spline surfaces or NURBS has been explained in [31] and was later extended to deal with multiple NURBS surfaces with continuity constraint in [32]. Previous work on the inclusion of a moving intersection line has been reported for a wing-body model where both the wing and the fuselage are formed by a single NURBS surface [43]. In this work, we extend the method to be applied to a realistic problem with reasonable complexity. In the geometry shown in Figure 1, the inboard part of the wing consists of 21 B-spline surfaces (in green), among which ten each are for the upper and lower surfaces and one is for the trailing edge. The fuselage itself is another B-spline surface (in grey) with its control points shown on the right in Figure 1. In the following, any of the 21 B-spline surfaces on the wing (green) will be referred to as Surf-W while the grey part on the fuselage will be referred to as Surf-F. Since we are interested in the wing-body junction optimisation, the parts in blue and green, i.e. the front and rear parts of the fuselage and the entire wing, will not change during the design iterations. Only the control points in the yellow box in Figure 1 are allowed to move in the x, y and z directions during optimisation. Surf-F has 25 control points, denoted by $\mathbf{P}_f$, in both parametric directions, u and v, and the control points subject to design are the 8th through 20th in the u direction and the 13th through 28th in the v direction. This results in a total of 208 control points that are free to move in all three directions. The whole design thus has 624 degrees of freedom. The main reason we confine the design control points inside a small region is that the optimisation should only introduce a localised change to the geometry at the wing-body junction.

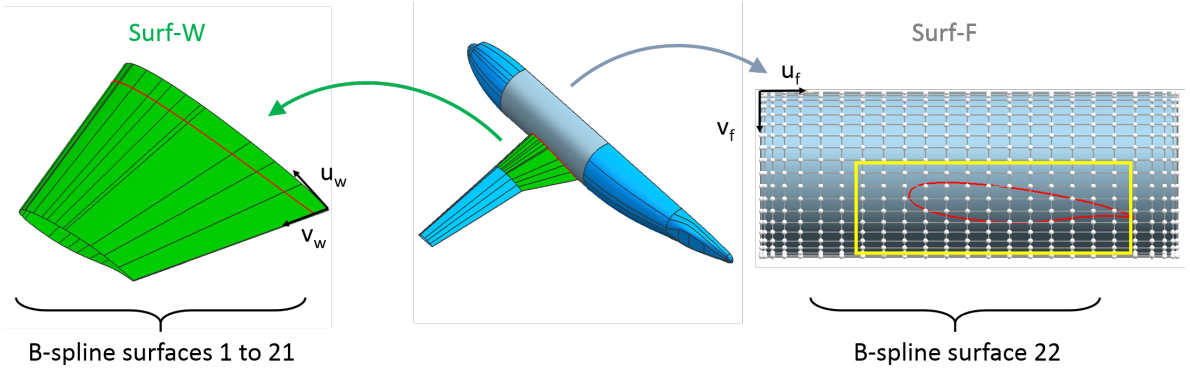

Figure 1: The geometry of the entire DLR-F6 wing-body model (centre) with the 21 B-spline surfaces forming the inner part of the wing (left) and the 22nd B-spline surface forming the middle part of the fuselage. The wing-body intersection line is marked in red in all three panels. The control points on Surf-F that are allowed to move are highlighted in the yellow box.

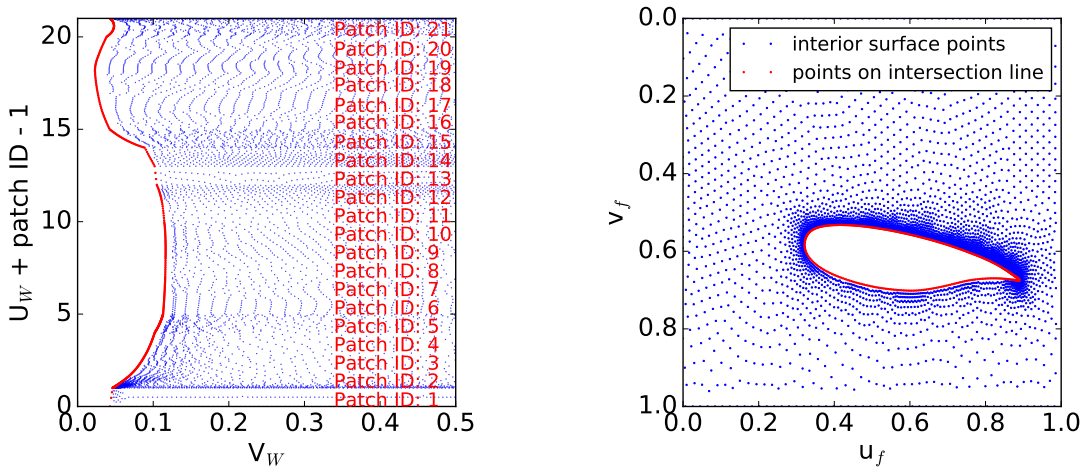

Figure 2: Surface mesh points on Surf-W (left) and Surf-F (right) projected into parametric space.

2.4. Linking surface mesh and geometry

In this section, we introduce the method to map surface mesh points to the B-spline surfaces via the parametric coordinates, which provides a mechanism to link and synchronise the surface mesh with the geometry. The computational mesh used is an unstructured mesh with prism elements in the boundary layer and mixed-type elements away from the viscous walls, consisting of a total of 2.4 million mesh points. The wall-normal spacing is around 1, in wall units, and the wall-normal cell growth ratio is limited to be no larger than 1.25. The far field boundary is located 100 reference chord lengths away. The surface mesh points that will move in this study are those that lie on Surf-W and Surf-F, and the link between these points and the underlying geometry needs to be established by finding the (u, v) coordinates of each mesh point in the parametric space of the B-spline surface it belongs to. In the mesh file, the surface points belonging to the fuselage, the wing and the intersection can be distinguished, which simplifies the point inversion process. The computed (u, v) coordinates of all the relevant surface mesh points are shown in Figure 2, in which the points corresponding to the intersection line are coloured in red. Shown on the left in Figure 2 are the point inversion results for the 21 B-spline surfaces that form Surf-W. Note that the x-axis values are u_w coordinates augmented with the patch ID (patch IDs of 0 through 20 are assigned to the 21 B-spline surfaces around the wing surface, starting from the trailing edge), so that all 21 B-spline surfaces can be shown together. Points with $v_w > 0.5$ are not shown for better illustration of the intersection.

2.5. Handling the moving intersection line

A point $\mathbf{S}$ on the intersection line between Surf-F and Surf-W (points that are marked red in Figure 2) is one that satisfies

$$\mathbf{S}(u_w, v_w, \mathbf{P}_w) = \mathbf{S}(u_f, v_f, \mathbf{P}_f),$$

where (u_w, v_w) and (u_f, v_f) are the parametric coordinates of the same point on the respective B-spline surfaces Surf-W and Surf-F, which are defined by the control points P_w and P_f respectively. Based on the definition of an intersection point, it can be found as the solution to

$$\mathbf{r}^T\mathbf{r} =: d = 0 \tag{4}$$

where the distance vector $\mathbf{r}$ between two points on the two B-spline surfaces is defined as

$$\mathbf{r} := \mathbf{S}(u_w, v_w, \mathbf{P}_w) - \mathbf{S}(u_f, v_f, \mathbf{P}_f).$$

Since d is a non-negative C^1 function, a necessary condition for Equation (4) to be satisfied is that the derivative of d should be zero

$$\left(\frac{\partial d}{\partial \mathbf{u}}\right)^T =: \mathbf{d_u} = \mathbf{0} \tag{5}$$

where $\mathbf{u} := (u_w, v_w, u_f, v_f)$. Equation (5) is solved using Newton's method iteratively as

$$J_\mathbf{u}(\mathbf{u}_{n+1} - \mathbf{u}_n) = -\mathbf{d_u}(\mathbf{u}_n) \tag{6}$$

until $\|\mathbf{d_u}\|_2$ converges below a threshold. The 4-by-4 Jacobian matrix $J_\mathbf{u}$ is defined as

$$J_\mathbf{u} := \frac{\partial \mathbf{d_u}}{\partial \mathbf{u}} = \begin{bmatrix} \partial^2 d/\partial u_w \partial u_w & \partial^2 d/\partial u_w \partial v_w & \partial^2 d/\partial u_w \partial u_f & \partial^2 d/\partial u_w \partial v_f \\ \partial^2 d/\partial v_w \partial u_w & \partial^2 d/\partial v_w \partial v_w & \partial^2 d/\partial v_w \partial u_f & \partial^2 d/\partial v_w \partial v_f \\ \partial^2 d/\partial u_f \partial u_w & \partial^2 d/\partial u_f \partial v_w & \partial^2 d/\partial u_f \partial u_f & \partial^2 d/\partial u_f \partial v_f \\ \partial^2 d/\partial v_f \partial u_w & \partial^2 d/\partial v_f \partial v_w & \partial^2 d/\partial v_f \partial u_f & \partial^2 d/\partial v_f \partial v_f \end{bmatrix}$$

and is inverted using Gaussian elimination. Usually only two or three Newton steps are required to find the solution to very good accuracy if a good initial guess is given, which is always possible in this work, because we start from the initial geometry where the intersection points can be found easily and during design steps, these intersection points only move within the neighbourhood of their original locations.

During the initial step, point inversion is done for the mesh points on the intersection line onto both Surf-W and Surf-F. Therefore, for all surface mesh points on the intersection line, two pairs of parametric coordinates (u_w, v_w) and (u_f, v_f) exist, corresponding to the initial control points $\mathbf{P}_w$ and $\mathbf{P}_f$. Once the control points for Surf-F are perturbed by a small amount to $\delta\mathbf{P}_f$ (note that we do not perturb Surf-W), we can compute the perturbation to the parametric coordinates of the intersection points, both $(\delta u_w, \delta v_w)$ and $(\delta u_f, \delta v_f)$ via Newton's method, which are then used to approximate the derivative term $\partial\mathbf{u}/\partial\mathbf{P}_f$ via finite differences.

It can be seen that the parametric coordinates found depend on the initial guess since the solution is not unique and it has one extra degree of freedom along the intersection line. To make the algorithm more robust, we found it useful to fix coordinate u_w from the initial state and only update the other three parameters when recomputing the coordinates of the mesh points on the intersection lines. Regarding implementation in the Newton step, this is done by modifying Equation (6) to be

$$(I - B + BJ_\mathbf{u})(\mathbf{u}_{n+1} - \mathbf{u}_n) = -B\mathbf{d_u}(\mathbf{u}_n)$$

where I is a 4-by-4 identity matrix and the projection matrix B is defined as

$$B := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This significantly simplifies the surface mesh movement on the wing due to a perturbed intersection line because all the

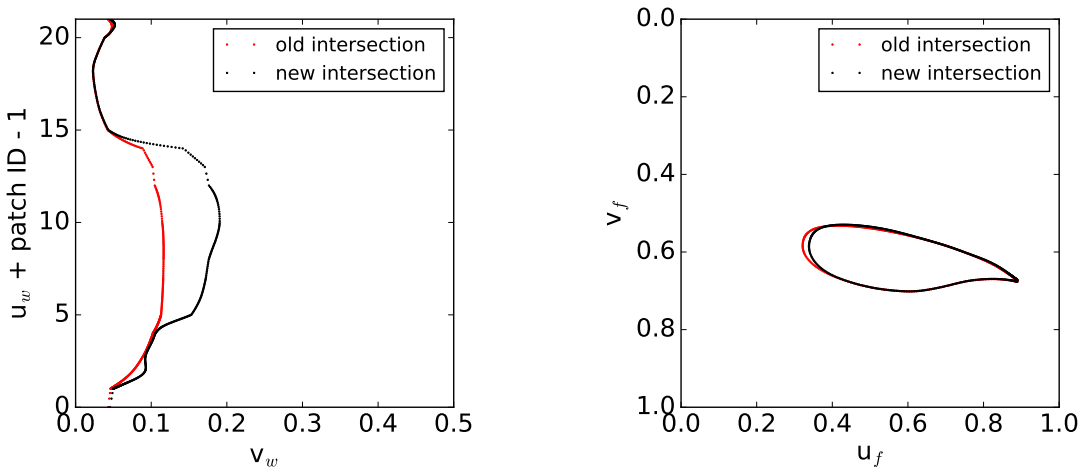

Figure 3: Parametric coordinates of mesh points on the original ('old') and perturbed ('new') intersection lines on Surf-W(left) and Surf-F(right).

mesh points on the wing essentially move in one-dimensional manner along the spanwise direction as the chordwise coordinate u_w is fixed. This implies that no mesh point on the wing, consisting of 21 chordwise connected B-spline surfaces, will cross the border of the patch it belongs to. To be more specific, in Figure 2 (left), the red points are perturbed only in the horizontal direction, while the red points in Figure 2 (right) can move in both u and v directions. Shown in Figure 3 are the parametric coordinates of the mesh points on the original and perturbed intersection lines for both Surf-W (Figure 3 (left)) and Surf-F (Figure 3 (right)). The perturbed intersection line used for illustrative purpose in the figure is the one corresponding to the optimised shape, which will be explained in the results section. The perturbed parametric coordinates of the intersection line will then be used as the boundary condition for the surface mesh movement on both Surf-W and Surf-F, explained in the following subsection.

2.6. Surface mesh movement

At the end of each optimisation step, the optimiser returns a value for the perturbation of the control points of the fuselage B-spline, $\Delta\mathbf{P}_f$. In general, we have

$$\mathbf{S}(u_w, v_w, \mathbf{P}_w) \neq \mathbf{S}(u_f, v_f, \mathbf{P}_f + \Delta\mathbf{P}_f),$$

as the intersection line has moved. Upon the re-computation of the intersection line, the surface mesh points originally lying on the intersection line now have updated parametric coordinate due to a perturbation Δv_w, Δu_f and Δv_f. Next step is to update the parametric coordinates of the remaining points on the surfaces so that the perturbation to the points on the intersection line smooths out into the surrounding mesh points.

This is done in the two-dimensional parametric space using inverse distance weighting (IDW) [44]. The algorithm of IDW is well documented in literature and it is only briefly discussed here. For Surf-F, the perturbation of the parametric coordinates of i-th interior mesh point (i.e., neither on the intersection line nor on the outer boundary) is governed by

$$\Delta u_f^i = \frac{\sum_{j=1}^N w_{i,j}\Delta u_f^j}{\sum_{j=1}^N w_{i,j}} \quad \text{and} \quad \Delta v_f^i = \frac{\sum_{j=1}^N w_{i,j}\Delta v_f^j}{\sum_{j=1}^N w_{i,j}} \tag{7}$$

where N is the total number of surface mesh points on Surf-F and the weighting coefficient is defined as

$$w_{i,j} := \begin{cases} \dfrac{1}{d_{i,j}}, \text{ if } d_{i,j} \leq 5d_{te} \\ 0, \text{ if } d_{i,j} > 5d_{te} \end{cases}$$

where the distance function is defined as

$$d_{i,j} := \|\mathbf{S}(u_f^i, v_f^i, \mathbf{P}_f) - \mathbf{S}(u_f^j, v_f^j, \mathbf{P}_f)\|_2$$

The weighting coefficient vanishes if the distance is larger than 5 times the trailing edge height d_{te}. This not only reduces the computational cost but also allows the mesh to locally deform in rigid body manner. Our numerical experiments show that this compact stencil allows the high quality in the boundary layer mesh to be preserved during mesh smoothing. Dirichlet boundary condition is used with known Δu_f^j and Δv_f^j at the intersection and zero displacement at the outer boundary. Equation (7) needs to be solved iteratively. We formed the Jacobian matrix and solved the resulting linear system using GMRES preconditioned by incomplete lower-upper factorisation.

The surface mesh movement on the wing is slightly different and therefore warrants further clarification. As mentioned above, during the re-computation of the intersection line, u_w is kept constant and only v_w is updated. The v_w coordinates for the remaining interior mesh points are updated according to

$$\Delta v_w^i = \frac{\sum_{j=1}^N w_{i,j} \Delta v_w^j}{\sum_{j=1}^N w_{i,j}}. \tag{8}$$

2.7. Volume mesh deformation

Mesh deformation based on RBF is used in DLR-TAU to perturb the volume mesh with given displacement of the surface mesh points. The methodology is documented in detail in [45] and has previously been used, for instance, to deform a mesh due to control surface deflection [46] and for volume deformation in a gradient-based shape optimisation [47]. RBF-based mesh deformation essentially interpolates the volume mesh displacement using radial basis functions that center around the surface mesh points whose displacement is known. In this work, thin plate spline is chosen as the radial basis function due to its accuracy and robustness. To limit the size of the RBF matrix to invert, the approach proposed in [47] is used to reduce the number of interpolation centres to 5000, resulting to a 5004-by-5004 matrix to be inverted per volume mesh deformation.

2.8. The optimiser

The optimiser used in this work is SLSQP, which is a sequential least squares quadratic programming algorithm [48] provided in SciPy [49]. SLSQP optimiser uses the quasi-Newton method with a BFGS update of the Hessian matrix with line search. Both equality and inequality constraints can be imposed as well as a prescribed bound for each design variable.

2.9. Summary of work flow

A summary of the entire automated optimisation process is illustrated in Figure 4. Given a CAD file for the original geometry and a CFD mesh generated from the same CAD file, the key boundary representation information is extracted from the CAD file and stored in a STEP file. The surface mesh points in the CFD mesh are linked with the B-spline surface information from the STEP file via point inversion. This concludes the preparation step. The flow and adjoint solvers are then run to compute the flow and adjoint solutions. To assemble the complete design gradient $\mathrm{d}\mathbf{J}/\mathrm{d}\boldsymbol{\alpha}$, finite difference is used to perturb each design control point on Surf-F and produce a perturbed volume grid using steps described in Subsections 2.5, 2.6 and 2.7. The perturbed meshes due to $\pm\Delta\alpha_i$ are then used for a residual-only calculation to get

$$\frac{\partial \mathbf{R}}{\partial \alpha_i} \approx= \frac{\mathbf{R}(\mathbf{U}, \mathbf{X} + \Delta\mathbf{X}(\Delta\mathbf{X}_s(\Delta\alpha_i))) - \mathbf{R}(\mathbf{U}, \mathbf{X} + \Delta\mathbf{X}(\Delta\mathbf{X}_s(-\Delta\alpha_i)))}{2\Delta\alpha_i}$$

The post-processing subroutine that computes the objective functions is also run on the deformed meshes to compute

$$\frac{\partial \mathbf{J}}{\partial \alpha_i} \approx \frac{\mathbf{J}(\mathbf{U}, \mathbf{X} + \Delta\mathbf{X}(\Delta\mathbf{X}_s(\Delta\alpha_i))) - \mathbf{J}(\mathbf{U}, \mathbf{X} + \Delta\mathbf{X}(\Delta\mathbf{X}_s(-\Delta\alpha_i)))}{2\Delta\alpha_i}$$

Compared to differentiating the CAD system and an the automated remeshing procedure, a major saving in computational cost can be achieved in our approach due to the inexpensive volume perturbation algorithm described in

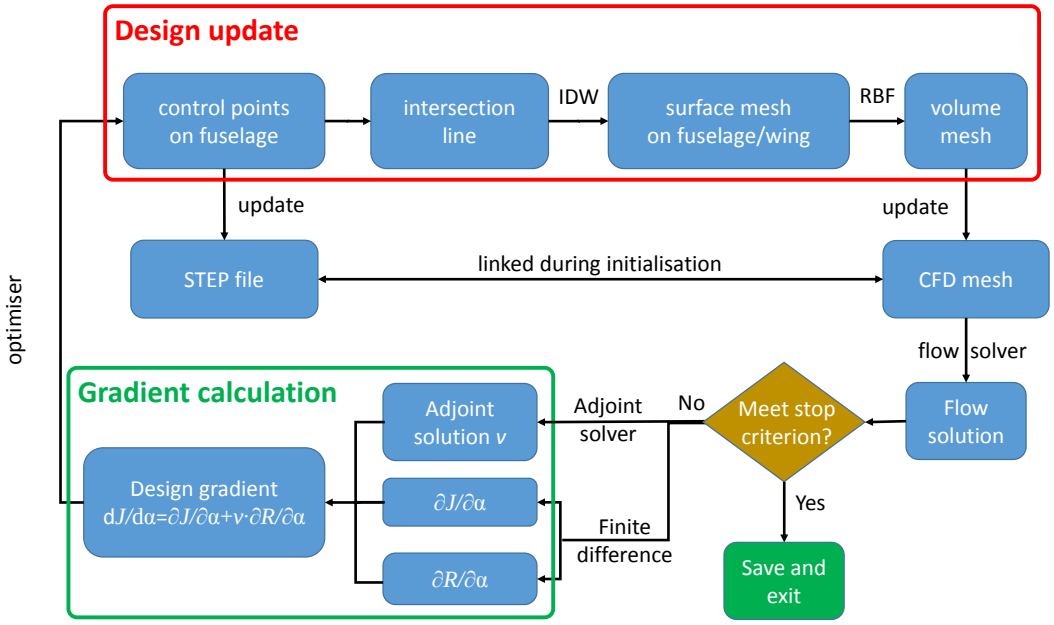

Figure 4: Flowchart for automated optimisation process.

Section 2.7. The design variable is defined as the deviation of the control points from their original coordinates, the volume mesh is always deformed from the original mesh. Therefore, the RBF matrix does not vary, and thus does not need to be recomputed and inverted. The inversion of the RBF interpolation matrix will incur some computational cost, one flow solution, however this needs to be performed only once throughout the entire optimisation. Every volume perturbation is then one matrix-vector product. As a result, the computational cost of generating the deformed meshes for finite differences is still order of magnitude smaller than that of solving the flow and the adjoint. Finite-differencing or tangent-linearisation hence lead to acceptable computational cost.

Using Equation (1), the final gradient can be computed and fed to the optimiser along with the function values from the flow solver run. The optimiser returns a new design in terms of a perturbation to the control points of the fuselage and the angle of attack. The control point perturbation is used to update the design and deform the CFD mesh. The angle of attack perturbation is incorporated in the free stream boundary condition. The automated process terminates when the stopping criterion is met.

3. DLR-F6 optimisation

The DLR-F6 case is a half wing-body model with mean aerodynamic chord (MAC) of 141.2 mm and projected half-span of 585.647 mm. The wing is defined by four aerofoil sections. The DLR-F6 model is available both as wing-body and wing-body-nacelle-pylon configuration in order to assess the accuracy of the computed interference drag due to engine installation. In this work, we focus on the flow separation at the wing-body juncture, rather than the nacelle-pylon-wing interference. Nacelle and pylon are thus not included in this work. All computations are performed at Mach number of 0.75 and Reynolds number of 3 million.

3.1. Flow solution for baseline geometry

The nonlinear flow solutions are computed for angles of attack of -4°, -2°, 0°, 0.5°, 1°, 2°, 3°, 3.5°, 4°, 4.5° and 5°. The steady-state solutions are found when the density residual has converged seven orders of magnitude. For angle of attack of 5.5°, the flow solver converges to a limit cycle at relative residual level of 10^{-2} and thus we were not able

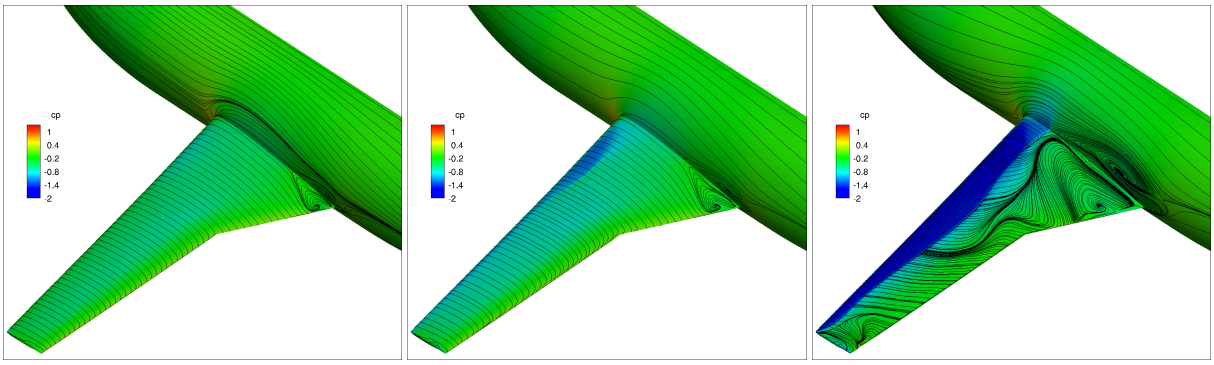

Figure 5: Pressure coefficient contour and skin friction lines for the original DLR-F6 at angle of attack of -2°, 0° and 4° (from left to right).

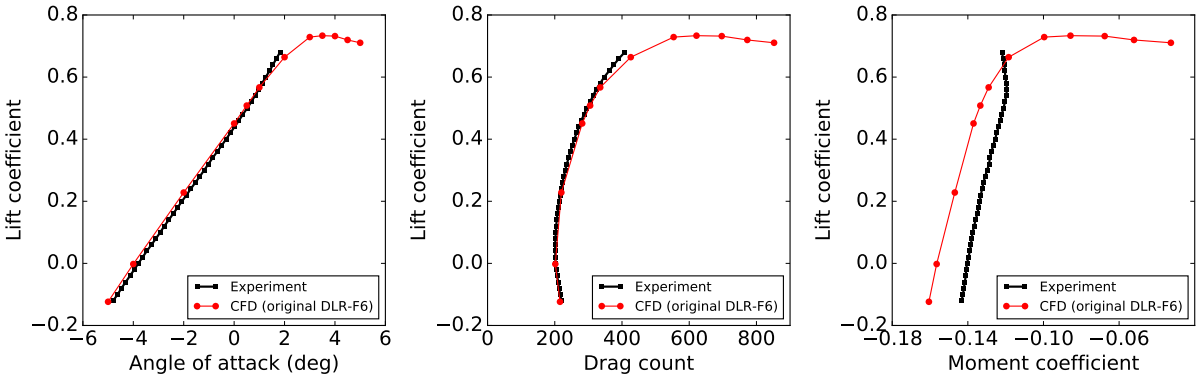

Figure 6: Computational vs. experimental results for the original DLR-F6.

to obtain a fully converged solution. This indicates that a stable steady-state solution probably does not exist at this condition. The converged steady-state solutions at several angles of attack are visualised in Figure 5. A side-of-body separation zone is present in all the results, with increasing size of the separation zone for larger angles of attack. For angle of attack larger than 1°, there is massive separation along the wing span and the separation patterns varies in the large range of angles of attack considered. Despite the various locations of flow separation, the lift, drag and moment curves agree reasonably well with experimental results that are available only for angle of attack of -4.8° to 1.82°(Fig. 6). Nevertheless, we have some confidence that the CFD results beyond 1.82° are representative of the kind of flows expected from this wing-body configuration. Consistent with the results of other participants in DPW II, the pitching moment in our study is also underestimated by CFD [16].

3.2. Optimisation problem formulation

The aerodynamic shape optimisation problem is formulated as summarised in Table 1. The control points of the B-Spline surface for the fuselage together with the angle of attack are used as the design variables to minimise the drag coefficient with the constraints that the lift and moment coefficient do not change. In addition, only a positive, i.e., outward, deformation of the fuselage is permitted. A single-point optimisation is performed at near cruise condition, with Mach number of 0.75 and an angle of attack of 0° with a lift coefficient of 0.4506. To obtain a more robust optimum, however, one should ideally perform a multi-point optimisation by including conditions at different lift coefficients and Mach numbers [50, 51, 52]. This is not pursued in this work as the focus is on the geometric parametrisation itself. Nevertheless, some additional technical aspects need to be considered if a multi-point optimisation was to be performed for this case. First, as can be seen from the flow solution results, for low angles

Table 1: Aerodynamic shape optimisation problem.

	Function/variable	Description	Quantity
minimise	C_D	Drag coefficient	1
with respect to	α	Angle of attack	1
	P_f	B-spline control points	624
subject to	$C_L = 0.4506$	Lift coefficient constraint	1
	C_M = -0.1370	Moment coefficient constraint	1
	$\Delta P_f \geq 0$	Positive deformation constraint	624

Table 2: Aerodynamic shape optimisation results.

	$C_D \times 10^4$	C_L	C_L/C_D	C_M	AoA (°)
original	281.6	0.4506	16.00	-0.1370	0
optimised	269.2	0.4506	16.74	-0.1370	-0.31

of attack, there is already an area of flow separation present next to the wing-fuselage junction. At the higher angle of attack of 4°, a massive flow separation appears on the entire wing. Our experience is that in order to find a steady adjoint solution, a strong implicit solver [53] or even a Newton solver is needed. Indeed, a deflated Krylov subspace solver has recently been used to improve the robustness and efficiency of a Newton adjoint solver at challenging off-design conditions [54]. Secondly, even if steady state nonlinear flow and adjoint solutions could be found by improved numerics, it is still up to the optimisation practitioner to judge the appropriateness of the steady state approach itself, as when the flow becomes increasingly unsteady, the steady state solution may no longer be able to reflect the relevant flow physics [55]. In those cases, an unsteady approach is more appropriate.

3.3. Optimisation results

After 24 flow evaluations and 39 gradient evaluations (13 for each cost function), the performance of the optimised shape is compared with the original one in Table 2. A reduction of 12.4 drag counts is achieved with only 0.009% change to lift and and 0.010% to moment, while the angle of attack is decreased by 0.31° for the same lift. The optimisation process was terminated when the objective functions had converged to what appears to be a local optimum and the flow constraints were satisfied. The evolution of lift, drag and moment over the optimisation steps are shown in Figure 7 and the drag count is separately shown in Figure 8 with red and blue markers indicating where the flow and adjoint solutions are computed at a particular optimisation step. Also shown in Figure 8 is the evolution of the angle of attack (AoA). The improved fuselage shape would result in increased lift at constant AoA, to maintain the target lift AoA is continuously reduced due to an improved fuselage shape. Shown in Figure 9 are streamline plots for both the original and the optimised shapes. It can be seen that the large separation zone in the original shape has been reduced significantly. The deformation of the fuselage, scaled by MAC, is visualised on the CFD mesh in Figure 10. It can be seen that despite the rich design space the optimised shape is very smooth with two distinctive bumps created, one near the wing leading edge and the other near the trailing edge above the wing.

To examine whether the single-point optimised shape is a robust one, the angle of attack is again varied from -5° to 5° including -0.31°. The lift, drag and moment coefficients are shown in Figure 11 for the optimised and original shapes. It can be seen that both the lift curve slope and the maximum lift have increased (although the latter has to be taken with some caution as CFD is known to be less reliable at large angle of attack condition). The lift-to-drag ratio for $C_L = 0.4506$ has increased for over 4.5% from 16.0 to 16.74.

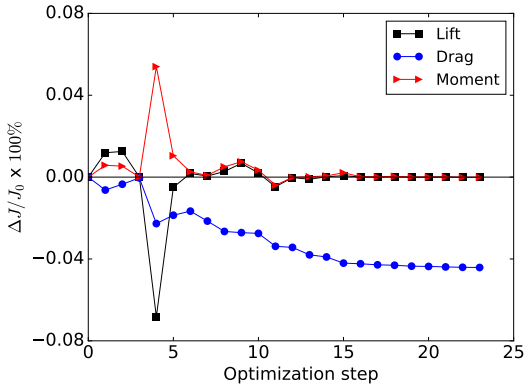

Figure 7: Normalised change of lift, drag and moment over optimisation steps.

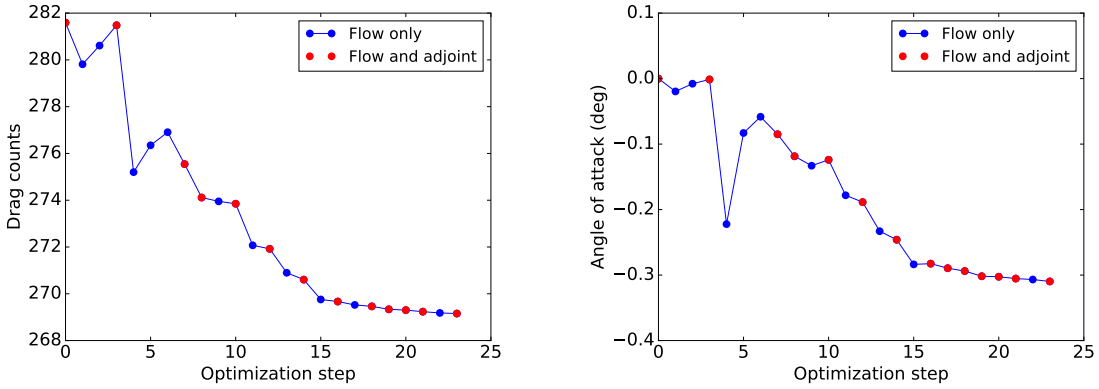

Figure 8: Evolution of drag counts and the angle of attack over optimisation steps. Blue: only the flow solution is evaluated; red: both the flow and adjoint solutions are computed.

4. Conclusions

Aerodynamic shape optimisation has been presented which uses adjoint flow sensitivities and a CAD-based parametrisation inside the design loop. A novel CAD-based parametrisation method has been proposed that uses as design variables the control points of the B-spline boundary representation and thus avoids computationally expensive differentiation of the complete CAD system. Computational cost of the optimisation is further reduced by using inexpensive inverse distance weighting for surface mesh relaxation, and pre-computed Radial Basis Functions for the volume mesh. Shape derivatives are computed for the surface, as well as the wing-fuselage intersection line. The capability of dealing with a moving intersection allows the parametrisation method to automatically incorporate the intersection line in the optimisation process.

The parametrisation method is applied to the DLR-F6 model, a typical wing-body configuration. At the cost of 24 steady-state flow and 39 adjoint solutions, an optimised shape is found using the SLSQP optimiser. The optimised shape features a deformed fuselage with two distinctive bumps, one around the wing leading edge and the other above the trailing edge. The junction separation is significantly suppressed due to this shape change, demonstrating the effectiveness of the methodology. The proposed CAD-based parametrisation method is a step-change toward enabling automated shape optimisation of fully-featured aircraft, which typically consist of complex intersecting surfaces.

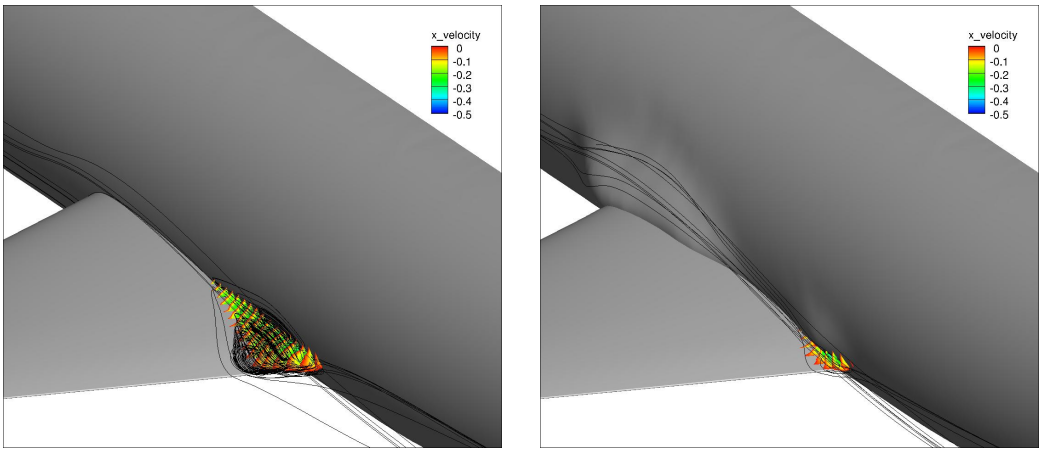

Figure 9: Region with reverse flow and three-dimensional streamlines for original (left) and optimised (right) shape.

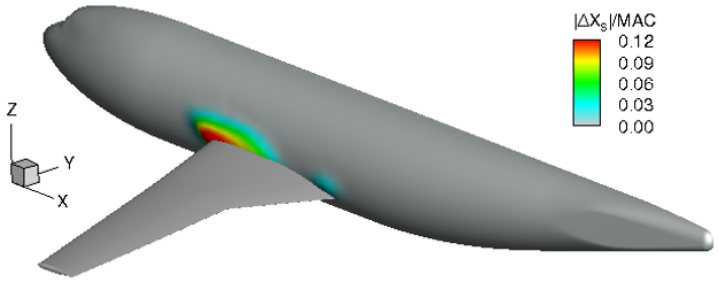

Figure 10: Deformation contour of fuselage. The deformation is scaled by MAC and visualised as a contour plot on the original fuselage.

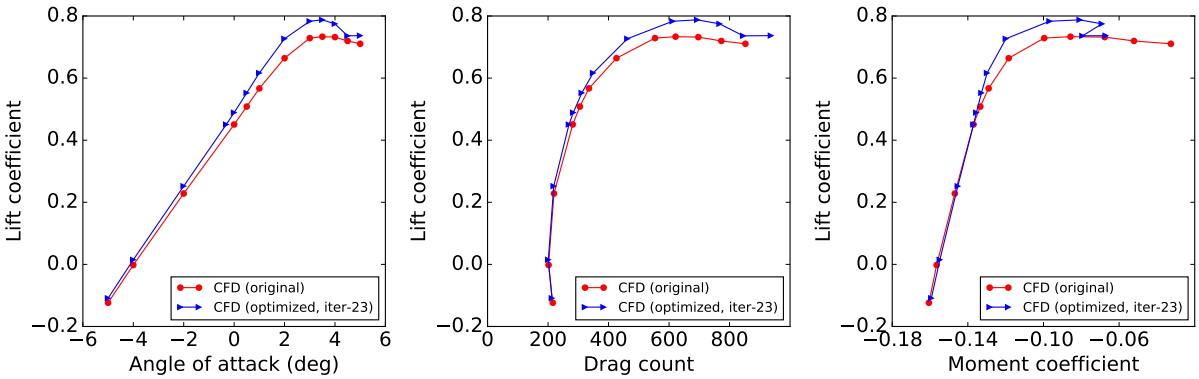

Figure 11: Lift curve, drag and pitching moment polars for original and optimised shapes.

Acknowledgements

The research leading to the results in this work is co-funded by Innovate UK, the UK's innovation agency, as part of the Enhanced Fidelity Transonic Wing project, and the European Union Horizon 2020 Framework Programme

for Research and Innovation under Grant Agreement No. 642959. This first author would like to thank Giampaolo Pagliuca for the help on developing the optimisation script using SciPy and resolving high performance computing related issues.

References

[1] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. *Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation*, volume 96. NASA Ames Research Center, Research Institute for Advanced Computer Science, 1996.

[2] M.B. Giles and N.A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3-4):393–415, 2000.

[3] N. Kroll, N.R. Gauger, J. Brezillon, R. Dwight, A. Fazzolari, D. Vollmer, K. Becker, H. Barnewitz, V. Schulz, and S. Hazra. Flow simulation and shape optimization for aircraft design. *Journal of Computational and Applied Mathematics*, 203(2):397–411, 2007.

[4] O. Pironneau. On optimum design in fluid mechanics. *Journal of fluid mechanics*, 64:97–110, 1974.

[5] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.

[6] A. Jameson, L. Martinelli, and N.A. Pierce. Optimum aerodynamic design using the Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10(1-4):213–237, 1998.

[7] W.K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28(4):443–480, 1999.

[8] E.J. Nielsen, J. Lu, M.A. Park, and D.L. Darmofal. An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids. *Computers & Fluids*, 33(9):1131–1155, 2004.

[9] D.J. Mavriplis. Multigrid solution of the discrete adjoint for optimization problems on unstructured meshes. *AIAA Journal*, 44(1):42–50, 2006.

[10] D.J. Mavriplis. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA Journal*, 45(4):741–750, 2007.

[11] S.F. Hoerner. *Fluid-dynamic drag: practical information on aerodynamic drag and hydrodynamic resistance*. Hoerner Fluid Dynamics, 1965.

[12] K. Leoviriyakit. *Wing planform optimization via an adjoint method*. PhD thesis, Stanford University, 2005.

[13] J.E. Hicken and D.W. Zingg. Induced-drag minimization of nonplanar geometries based on the Euler equations. *AIAA Journal*, 48(11):2564–2575, 2010.

[14] R.D. Joslin. Aircraft laminar flow control 1. *Annual review of fluid mechanics*, 30(1):1–29, 1998.

[15] R. Rashad and D.W. Zingg. Aerodynamic shape optimization for natural laminar flow using a discrete-adjoint approach. *AIAA Journal*, 54(11):3321–3337, 2016.

[16] K.R. Laflin, S.M. Klausmeyer, T. Zickuhr, J.C. Vassberg, R.A. Wahls, J.H. Morrison, O.P. Brodersen, M.E. Rakowitz, E.N. Tinoco, and J.-L. Godard. Data summary from second AIAA computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, 42(5):1165–1178, 2005.

[17] M. Hemsch. Statistical analysis of CFD solutions from the drag prediction workshop. *AIAA paper*, 842:2002, 2002.

[18] G.M. Gatlin, M.B. Rivers, S.L. Goodliff, R. Rudnik, M. Sitzmann, et al. Experimental investigation of the DLR-F6 transport configuration in the national transonic facility. *AIAA Paper*, 6917:2008, 2008.

[19] J.C. Vassberg, A.J. Sclafani, and M.A. DeHaan. A wing-body fairing design for the DLR-F6 model: A DPW-III case study. In *AIAA Paper 2005-4730, 2005*.

[20] R. Rudnik, M. Sitzmann, J.-L. Godard, and F. Lebrun. Experimental investigation of the wing-body juncture flow on the DLR-F6 configuration in the ONERA S2MA facility. In *27th AIAA Applied Aerodynamics Conference*, pages 22–25, 2009.

[21] J. Brezillon and R.P. Dwight. Applications of a discrete viscous adjoint method for aerodynamic shape optimisation of 3D configurations. *CEAS Aeronautical Journal*, 3(1):25–34, 2012.

[22] G. Becker, M. Schäfer, and A. Jameson. An advanced NURBS fitting procedure for post-processing of grid-based shape optimizations. In *49th AIAA Aerospace Sciences Meeting, Orlando, FL, USA*, 2011.

[23] A. Jameson and A. Vassberg. Studies of alternative numerical optimization methods applied to the brachistochrone problem. 9(3), 2000.

[24] A. Stück and T. Rung. Adjoint RANS with filtered shape derivatives for hydrodynamic optimisation. *Computers & Fluids*, 47(1):22–32, 2011.

[25] A. Jaworski and J.-D. Müller. Toward modular multigrid design optimisation. In C. Bischof and J. Utke, editors, *Lecture Notes in Computational Science and Engineering*, volume 64, pages 281–291, "New York, NY, USA", 2008. Springer.

[26] M. Hojjat, E. Stavropoulou, and K.-U. Bletzinger. The vertex morphing method for node-based shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:494–513, 2014.

[27] E. Stavropoulou, M. Hojjat, and K.-U. Bletzinger. In-plane mesh regularization for node-based shape optimization problems. *Computer Methods in Applied Mechanics and Engineering*, 275:39–54, 2014.

[28] J.A. Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal*, 39(5):877–884, 2001.

[29] J.E. Hicken and D.W. Zingg. Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement. *AIAA Journal*, 48(2):400–413, 2010.

[30] T.T. Robinson, C.G. Armstrong, H.S. Chua, C. Othmer, and T. Grahs. Optimizing parameterized CAD geometries using sensitivities based on adjoint functions. *Computer-Aided Design & Applications*, 9(3):253–268, 2012.

[31] G. Yu, J.-D. Müller, D. Jones, and F. Christakopoulos. CAD-based shape optimisation using adjoint sensitivities. *Computers & Fluids*, 46(1):512–516, 2011.

[32] S. Xu, W. Jahn, and J.-D. Müller. CAD-based shape optimisation with CFD using a discrete adjoint. *International Journal for Numerical Methods in Fluids*, 74(3):153–168, 2014.

[33] J.T. Hwang, G.K.W. Kenway, and J.R.R.A. Martins. Geometry and structural modeling for high-fidelity aircraft conceptual design optimization. In *Proceedings of the 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Atlanta, GA, USA*, 2014.
[34] D. Schwamborn, T. Gerhold, and . Heinrich. The DLR TAU-code: Recent applications in research and industry. In *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006.
[35] T. Gerhold. Overview of the hybrid RANS code TAU. In *MEGAFLOW-Numerical Flow Simulation for Aircraft Design*, pages 81–92. Springer, 2005.
[36] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. AIAA-CP 81-1259, 1981.
[37] R.C. Swanson and E. Turkel. On central-difference and upwind schemes. *Journal of Computational Physics*, 101(2):292–306, 1992.
[38] S.R. Allmaras and F.T. Johnson. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, pages 1–11, 2012.
[39] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.
[40] S. Yoon and A. Jameson. Lower-upper symmetric-Gauss–Seidel method for the Euler and Navier–Stokes equations. *AIAA Journal*, 26(9):1025–1026, 1988.
[41] R. Dwight and J. Brezillon. Efficient and robust algorithms for solution of the adjoint compressible Navier–Stokes equations with applications. *International Journal for Numerical Methods in Fluids*, 60(4):365–389, 2009.
[42] S. Xu, S. Timme, and K.J. Badcock. Enabling off-design linearised aerodynamics analysis using Krylov subspace recycling technique. *Computers & Fluids*, 140:385–396, 2016.
[43] M.J. Martin-Burgos. *NURBS-Based Geometry Parameterization for Aerodynamic Shape Optimization*. PhD thesis, Universidad Politecnica de Madrid, 2015.
[44] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.
[45] A. De Boer, M.S. Van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures*, 85(11):784–795, 2007.
[46] A.K. Michler. Aircraft control surface deflection using RBF-based mesh deformation. *International Journal for Numerical Methods in Engineering*, 88(10):986–1007, 2011.
[47] S. Jakobsson and O. Amoignon. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, 36(6):1119–1136, 2007.
[48] D. Kraft. *A software package for sequential quadratic programming*. DFVLR Obersfaffeuhofen, Germany, 1988.
[49] E. Jones, T. Oliphant, and P. Peterson. SciPy: Open source scientific tools for Python, 2001. *URL http://www. scipy. org*, 73:86, 2015.
[50] Z. Lyu and J.R.R.A. Martins. Aerodynamic design optimization studies of a blended-wing-body aircraft. *Journal of Aircraft*, 51(5):1604–1617, 2014.
[51] Z. Lyu, G.K.W. Kenway, and J.R.R.A Martins. Aerodynamic shape optimization investigations of the common research model wing benchmark. *AIAA Journal*, 53(4):968–985, 2014.
[52] G.K. Kenway and J. Martins. Aerodynamic shape optimization of the CRM configuration including buffet-onset conditions. In *54th AIAA Aerospace Sciences Meeting*, page 1294, 2016.
[53] S. Xu, D. Radford, M. Meyer, and J.-D. Müller. Stabilisation of discrete steady adjoint solvers. *Journal of Computational Physics*, 299:175–195, 2015.
[54] S. Xu and S. Timme. Robust and efficient adjoint solver for complex flow conditions. *Computers & Fluids*, 148:26–38, 2017.
[55] J.A. Krakos and D.L. Darmofal. Effect of small-scale unsteadiness on adjoint-based output sensitivity. AIAA-CP 2009-4274, 2009.